Image Processing Lab

Batch 9:

Ajay Viswanathan [09EC3509]

Sourin Sutradhar [09EC3514]

Experiment 4

Binary and Grayscale Morphology

Problem Objective:

Write C++/Image-J modular functions to perform the following operations on the given test image, ricegrain.bmp. All functions must support 8-bit grayscale image formats and binary images.

1. 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 – center pixel).

2. Make separate functions for erosion, dilation, opening, and closing of grayscale images

a. ErodeGray, DilateGray

Input: Grayscale image, structuring element

Output: Eroded/dilated image

b. OpenGray, CloseGray

Input: Grayscale image, structuring element

Output: Opened/closed image

Brief 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 dilation operator takes two pieces of data as inputs. The first is the image which is to be dilated. The second is a (usually small) set of coordinate points known as a structuring element (also known as a kernel). It is this structuring element that determines the precise effect of the dilation on the input image.

The mathematical definition of dilation for binary images is as follows: Suppose that X is the set of Euclidean coordinates corresponding to the input binary image, and that K is the set of coordinates for the structuring element. Let Kx denote the translation of K so that its origin is at x. Then the dilation of X by K is simply the set of all points x such that the intersection of Kx with X is non-empty.

The mathematical definition of grayscale dilation is identical except for the way in which the set of coordinates associated with the input image is derived. In addition, these coordinates are 3-D rather than 2-D. As an example of binary dilation, suppose that the structuring element is a 3×3 square, with the origin at its center, as shown in Figure 1. Note that in this and subsequent diagrams, foreground pixels are represented by 1's and background pixels by 0's.

1	1	1
1	1	1
1	1	1

Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),
(-1, 0), (0, 0), (1, 0),
(-1, 1), (0, 1), (1, 1) }

Figure 1 A 3×3 square structuring element

To compute the dilation of a binary input image by this structuring element, we consider each of the background pixels in the input image in turn. For each background pixel (which we will call the input pixel) we superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel position. If at least one pixel in the structuring element coincides with a foreground pixel in the image underneath, then the input pixel is set to the foreground value. If all the corresponding pixels in the image are background, however, the input pixel is left at the background value.

For our example 3×3 structuring element, the effect of this operation is to set to the foreground color any background pixels that have a neighboring foreground pixel (assuming 8-connectedness). Such pixels must lie at the edges of white regions, and so the practical upshot is that foreground regions grow (and holes inside a region shrink). Dilation is the dual of erosion i.e. dilating foreground pixels is equivalent to eroding the background pixels.

Let A be the image undergoing analysis, B be the structuring element, then Dilation is described by:

$$\boxed{A \oplus B = \{c \in Z^2 \mid c = a + b \quad for \quad some \quad a \in A, b \in B\}}$$

The erosion operator takes two pieces of data as inputs. The first is the image which is to be eroded. The second is a (usually small) set of coordinate points known as a structuring element (also known as a kernel). It is this structuring element that determines the precise effect of the erosion on the input image.

The mathematical definition of erosion for binary images is as follows:

Suppose that X is the set of Euclidean coordinates corresponding to the input binary image, and that K is the set of coordinates for the structuring element. Let Kx denote the translation of K so that its origin is at x. Then the erosion of X by K is simply the set of all points x such that Kx is a subset of X.

The mathematical definition for grayscale erosion is identical except in the way in which the set of coordinates associated with the input image is derived. In addition, these coordinates are 3-D rather than 2-D. As an example of binary erosion, suppose that the structuring element is a 3×3 square, with the origin at its center as shown in Figure 1. Note that in this and subsequent diagrams, foreground pixels are represented by 1's and background pixels by 0's.

1	1	1
1	1	1
1	1	1

Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),

(-1, 0), (0, 0), (1, 0),

(-1, 1), (0, 1), (1, 1) }

Figure 2. A 3×3 square structuring element

To compute the erosion of a binary input image by this structuring element, we consider each of the foreground pixels in the input image in turn. For each foreground pixel (which we will call the input pixel) we superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel coordinates. If for every pixel in the structuring element, the corresponding pixel in the image underneath is a foreground pixel, then the input pixel is left as it is. If any of the corresponding pixels in the image are background, however, the input pixel is also set to background value.

For our example 3×3 structuring element, the effect of this operation is to remove any foreground pixel that is not completely surrounded by other white pixels (assuming 8-connectedness). Such pixels must lie at the edges of white regions, and so the practical upshot is that foreground regions shrink (and holes inside a region grow).

Erosion is defined as:

$$\boxed{A \Theta B = \{ x \in Z^2 | (B)_x \subseteq A)}$$

Very simply, an opening is defined as an erosion followed by a dilation using the same structuring element for both operations. See the sections on erosion and dilation for details of the individual steps. The opening operator therefore requires two inputs: an image to be opened, and a structuring element.

Graylevel opening consists simply of a graylevel erosion followed by a graylevel dilation.

Opening is the dual of closing, i.e. opening the foreground pixels with a particular structuring element is equivalent to closing the background pixels with the same element.

It is defined as:

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

Closing is opening performed in reverse. It is defined simply as a dilation followed by an erosion using the same structuring element for both operations. See the sections on erosion and dilation for details of the individual steps. The closing operator therefore requires two inputs: an image to be closed and a structuring element.

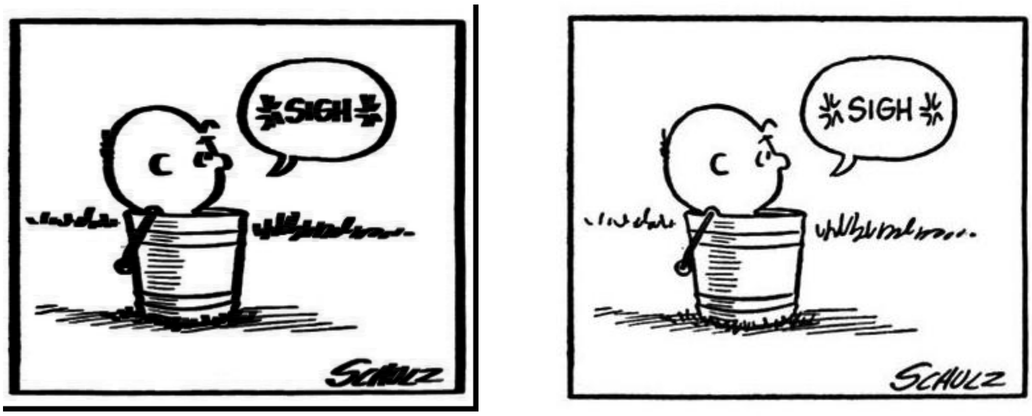
Graylevel closing consists straightforwardly of a graylevel dilation followed by a graylevel erosion.

Closing is the dual of opening, i.e. closing the foreground pixels with a particular structuring element, is equivalent to closing the background with the same element.

It is defined as:

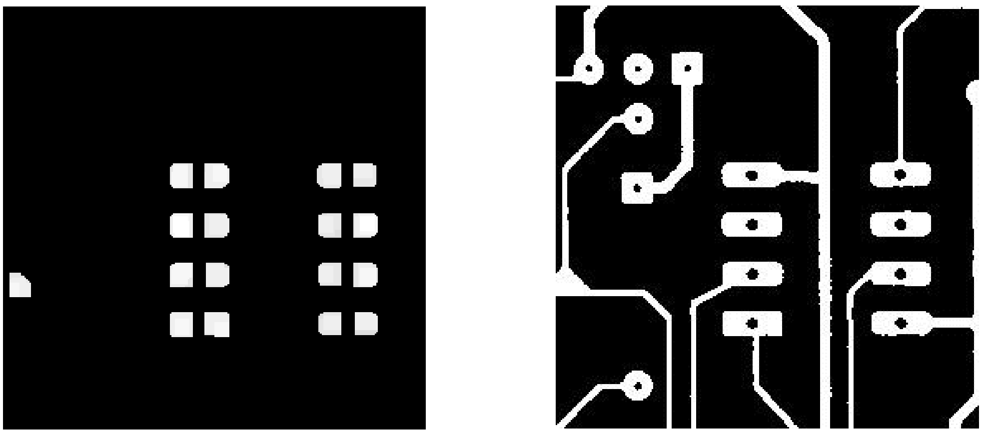
$$A \bullet B = (A \oplus B) \ \Theta \ B$$

Results:



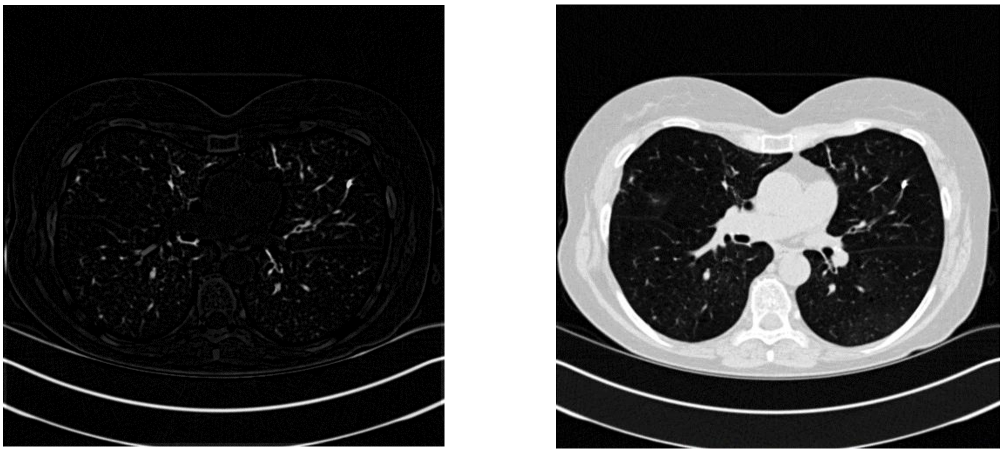
Fountain-pen effect – Erosion, Line element, 5x5 mask size

Line thickening effect – Erosion, Diamond element, 5x5 mask size

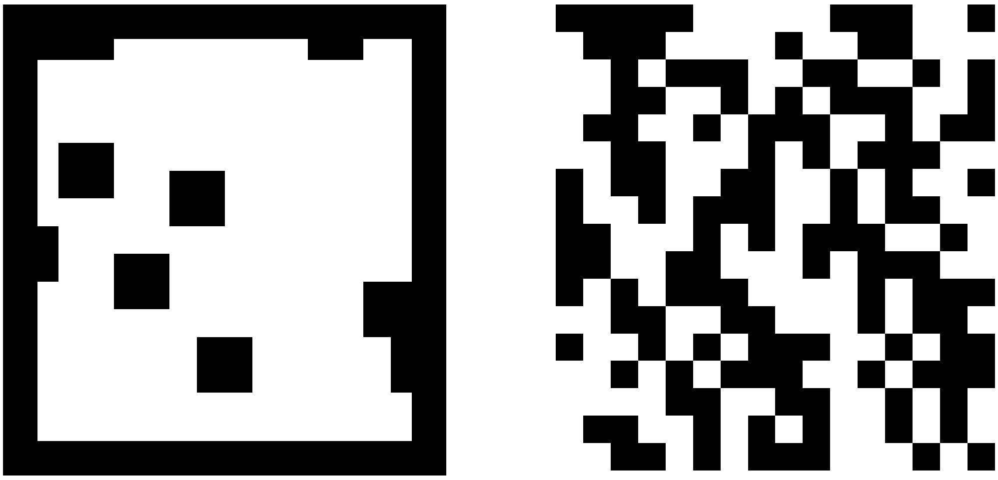


Isolating elements – Opening, Square element, 9x9 mask size

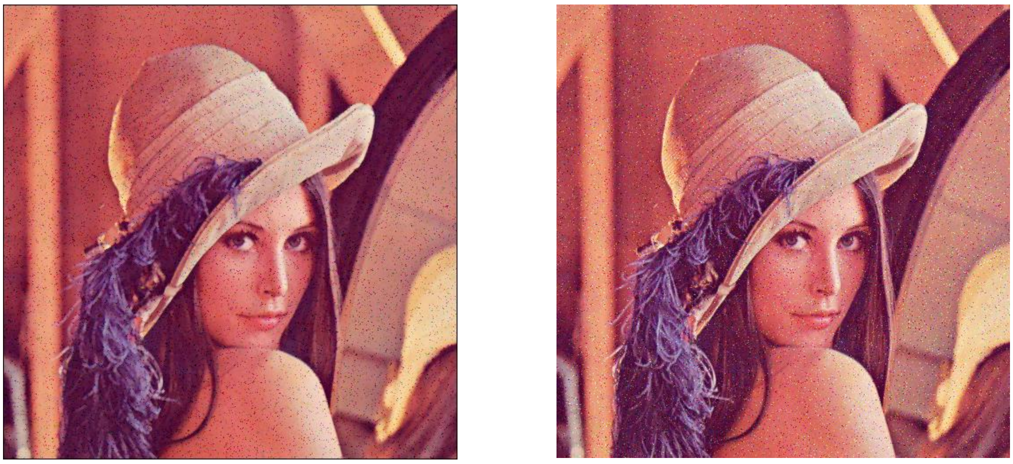
Image shrinking effect – Erosion, Line element, 21x21 mask size



Highlighting finer details of image – White top-hat, Diamond element, 9x9 mask size

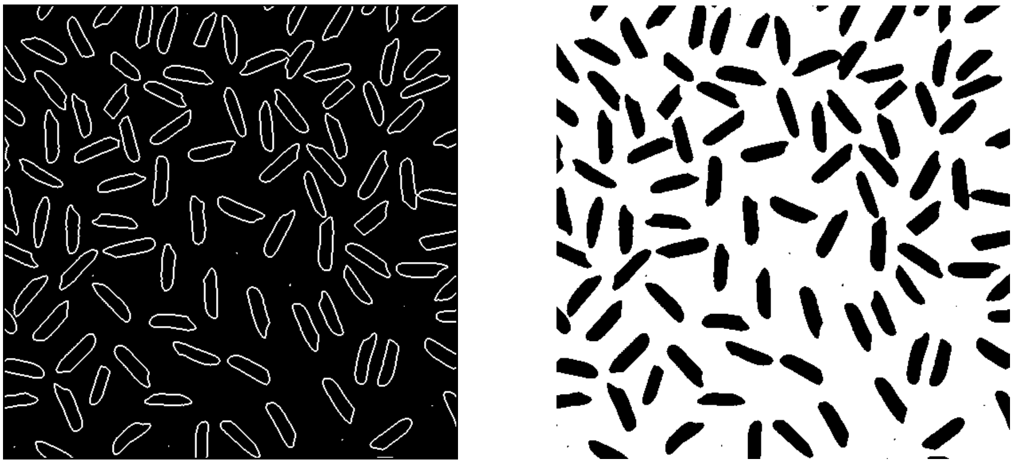


Detecting largest gaps – Closing, Square element, 21x21 mask size



Removing random noise – Opening, Diamond element, 3x3 mask size

Dilation, Square element, 3x3 mask size



Edge-detection, Diamond element, 5x5 mask size

Observation and Inference:

1. Opening and closing are idempotent operations. This means, performing them on an already processed image won't change the result. Opening and closing are duals of each other.
2. For erosion we calculate the infimum (or minimum value from the mask) and for dilation we calculate the supremum (or maximum value from the mask).
3. Top-hat transform is used to extract small elements and details from given images.
4. White-top-hat returns elements that are smaller than structuring element and brighter than surroundings
5. Black-top-hat returns elements that are smaller than structuring element and darker than surroundings.
6. Other structuring elements are cross-shaped, and other transforms are watershed, hit-and-miss and skeletonization.
7. We can count number of items in a thresholded image by running an opening transform and counting elements by connect component labelling.