CS5680/CS6680 – Fall Semester 2024

Assignment 5 – Morphological Operations

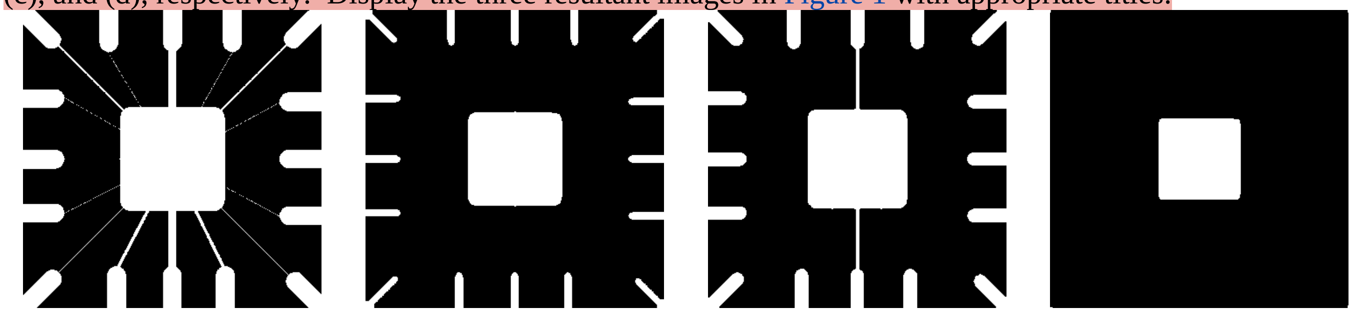
Due: 11:59 p.m., Tuesday, October 29, 2024

Total Points: 45 points

You can call built-in morphological operations to solve all sub-problems of Problem I.

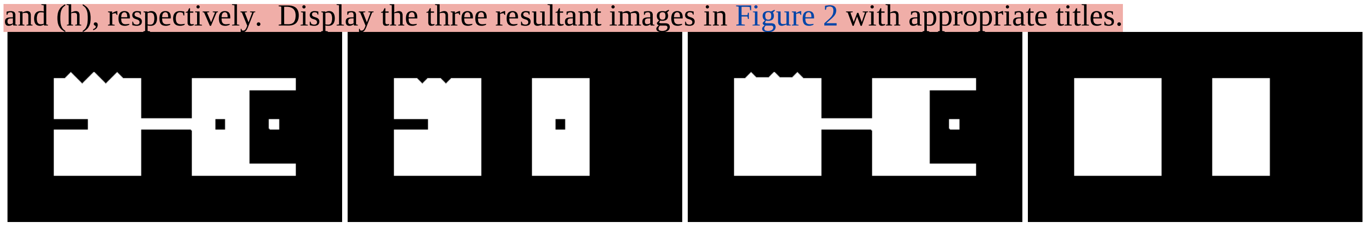
Problem I: Problem Solving Using Morphological Operations [Total: 20 points]

1. [5 points] Apply morphological operation(s) on *Wirebond.tif* to obtain three desired images as shown in (b), (c), and (d), respectively. Display the three resultant images in Figure 1 with appropriate titles.



(a) Original image (b) Desired image 1 (c) Desired image 2 (d) Desired image 3

[5 points] Apply morphological operation(s) on *Shapes.tif* to obtain three desired images as shown in (f), (g), and (h), respectively. Display the three resultant images in Figure 2 with appropriate titles.



(e) Original image (f) Desired image 1 (g) Desired image 2 (h) Desired image 3

Note: Here, I list the original image for your reference so you can visually compare the differences between the original image and the expected resultant images. As long as your resultant images are similar to the ones posted on the assignment sheet, you will get full credit.

2. [5 points] Apply an open-close operation on *Dowels.tif* using a disk structuring element of radius 5 (i.e., an open operation followed by a close operation is applied to the original image using the same structuring element). Apply a close-open operation on *Dowel.tif* using a disk structuring element of radius 5 (i.e., a close operation followed by an open operation is applied to the original image using the same structuring element). Display two resultant images in Figure 3 side-by-side.

Apply a series of open-close operations on *Dowel.tif* using a series of structuring elements of increasing size (i.e., a disk structuring element of a radius of 2, 3, 4, and 5). Similarly, apply a series of close-open operations on *Dowel.tif* using a series of structuring elements of increasing size (i.e., a disk structuring element of a radius of 2, 3, 4, and 5). Display two resultant images in Figure 4 side-by-side. You have to use "loop" to include the series of open-close operations or close-open operations. In other words, each loop will perform one open-close or close-open operation with a disk structuring element of a certain radius.

3. [5 points] Apply morphological operation(s) on *SmallSquares.tif* to locate foreground pixels that have east and north neighbors and that have no northwest, west, southwest, south, or southeast neighbors. Display the final resultant image in Figure 5. Use the console to show the number of foreground pixels that satisfy the above conditions.

Problem II: Applications of Morphological Operations [Total: 25 points]

A preprocessing step in the application of microscopy is to isolate individual round particles from similar particles that overlap in a group of two or more particles (see image "**Ball.tif**"). Assuming that **all particles are of the same size**, use the "**extraction of connected components**" morphological operation as described below to solve the following sub-problems.

Let Y be a connected component contained in image A and assume that a point p of Y is known. Then the following iterative expression yields all the elements of Y.

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \ldots\ldots$$

where $X_0 = p$ and B is a suitable structuring element.

The algorithm terminates at iteration step k if $X_k = X_{k-1}$. The algorithm converges to $Y = X_k$.

1. [8 points] Using the idea of "extraction of connected components" to implement a **FindComponentLabels** function to label connected objects (i.e., connected components with the value of 1's) in a binary image. This function has two input parameters, where **im** is the original binary image and **se** is the structuring element. It has two output parameters **labelIm** and **num**, where **labelIm** is the labeled image that contains the labeled connected objects **(Please sequentially label the connected objects with 1 being the starting label)** and **num** is the number of connected objects (i.e., **the largest label for a connected object**) found in the binary image. It should be noted that different connected objects have different labels and the labels for all the elements in a connected object are the same. In addition, the smallest label for a connected object is **1** and the largest label for a connected object is **num**.

Call the **FindComponentLabels** function to label the connected particles and return the total number of connected particles in a given image **Ball.tif.** Use the console to display the total number of connected particles in **Ball.tif** found by **FindComponentLabels**. Display these connected objects in Figure 6 using appropriate visible gray-level intensities or color intensities.

2. [2 points] Call an appropriate built-in function, which accomplishes the same functionality as the **FindComponentLabels** function, to label connected particles and return the total number of connected particles. Use the console to display the number of connected particles in **Ball.tif** found by the built-in function. Display these connected particles in Figure 7 using appropriate visible gray-level intensities or color intensities. Note: *Depending on the implementation, FindComponentLabels function and built-in function may label the same connected particles using different labels. However, the total number of connected particles should be the same.*

3. [5 points] Produce an image **A** containing only **connected particles** residing on the border (i.e., four sides) of the image. Display the original image and image **A** side-by-side in Figure 8 with appropriate titles. Use the console to show the number of connected particles residing on the border. You have to write your solution to this problem without calling a built-in function.

4. [8 points] Produce an image **B** containing only visually **overlapping particles** not residing on the border of the image and an image **C** containing only visually **individual particles** not residing on the border of the image. Display the original image, image **B**, and image **C** side-by-side in Figure 9 with appropriate titles. Use the console to show the number of overlapping particles that do not reside on the border and the number of individual particles that do not reside on the border. Hint: Since all particles are of the same size, you may need to design a strategy to estimate the size of the individual particle. **Please DO NOT hardcode any values to solve the problem.**

5. [2 points] Produce an image **D** containing only visually **partial overlapping particles** residing on the border of the image and an image **E** containing only visually **partial individual particles** residing on the border of the image. Display the original image, image **D**, and image **E** side-by-side in Figure 10 with appropriate titles. Use the console to show the number of visually partial overlapping particles that reside on the border and the number of visually partial individual particles that reside on the border.