CS540/440 – Digital Image Processing

Lab 4 – Exercises on Morphological Operations

Due: 16:00 p.m. Wednesday, March 28th, 2018

General Assignment Instructions: The same as Lab 2.

Problems: Problem Solving Using Morphological Operations

1. [2 points]

Load in *City.jpg* image. Apply the morphological gradient operation $g = (f \oplus b) - (f \ominus b)$ where f is the original image and b is the 3×3 square structuring element. Display the resultant image in figure 1. Use the Matlab console to describe the appearance of the resultant image and the reason for such an appearance.

Matlab hint: *strel(), imdilate(), imerode(), imsubtract()*

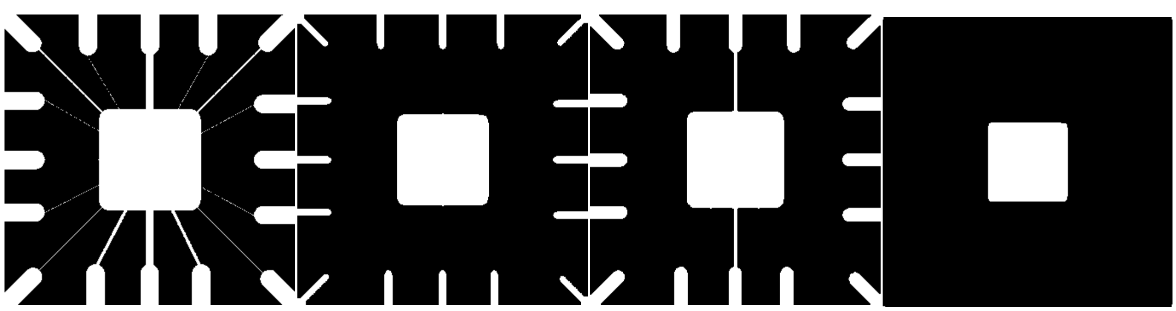
2. [3 points]

Load in *SmallSquares.tif* image. Apply the appropriate morphological operation(s) on the original image to locate foreground pixels that have east and south neighbors and that have no northeast, north, north-west, west, or southwest neighbors. Display the final resultant image in figure 2. Use the Matlab console to show the number of foreground pixels that satisfy the above conditions.

Matlab hint: *strel(),bwhitmiss()*

3. [5 points]

Load in *Wirebond.tif* image. Apply appropriate morphological operation(s) on the **original image** to obtain the following three desired images (i.e., (b), (c), and (d)), respectively. Display the three resultant images in figure 3 with appropriate titles. 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.

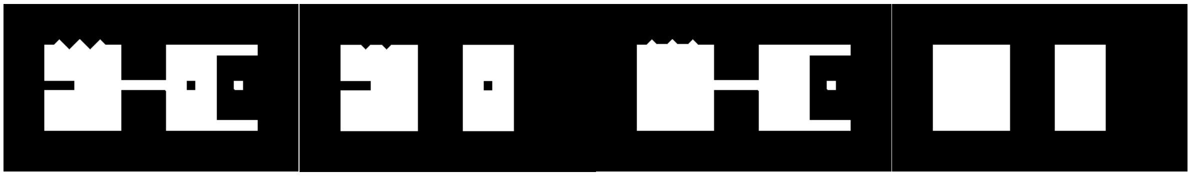


(a) Original image (b) Desired image 1 (c) Desired image 2 (d) Desired image 3

Matlab hints: *strel(),imerode(im,se)*

4. [5 points]

Load in *Shapes.tif* image. Apply appropriate morphological operation(s) on the **original image** to obtain the following three desired images (i.e., (b), (c), and (d)), respectively. Display the three resultant images in figure 4 with appropriate titles. 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.



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

Matlab hint: *strel(),imopen(im, se), imclose(im,se)*

5. [5 points]

Load in *Dowels.tif* image. 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 5 side-by-side. Use the Matlab console to explain the appearance of two resultant images and the reason(s) for such an appearance.

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 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 radius of 2, 3, 4, and 5). Display two resultant images in figure 6 side-by-side. Use the Matlab console to explain the appearance of two resultant images and the reason(s) for such an appearance.

Matlab hint: *strel(),imopen(im, se), imclose(im,se),*

6. [15 points]

A preprocessing step in an application of microscopy is to isolate individual round particles from similar particles that overlap in 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 (pp. 536–539 of the second edition of the Digital Image Processing textbook or pp. 645–647 of the third edition of the Digital Image Processing textbook) to solve the following sub-problems. Here is the basic idea of the "extraction of connected components" algorithm:

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

$$X_k = (X_{k-1} \oplus B) \cap A \qquad \mathrm{k} = 1, 2, 3, \cdots$$

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 $Y = X_k$.

6.1 [2 pts] Call an appropriate Matlab built-in function to label connected particles and return the total number of connected particles. Use the Matlab console to display the number of connected particles in **Ball.tif** found by the Matlab built-in function. Display these connected particles in figure 2 using appropriate gray-level intensities.

6.2 [5 pts] Produce an image **A** containing only **connected particles** not residing on the border of the image. Display the original image and the image **A** side-by-side in figure 3 with appropriate titles. Use the Matlab console to show the number of connected particles that do not reside on the border.

6.3 [8 pts] Produce an image **B** containing only **overlapping connected particles** not residing on the border of the image. Display the original image and the image **B** side-by-side in figure 4 with appropriate titles. Use the Matlab console to show the number of the overlapping connected particles that do not reside on the border. **Please DO NOT hard code any values to solve the problem.**

Matlab hint: *strel(), bwlabel(), regionprops(),*

Close all figures and all variables in the workspace.