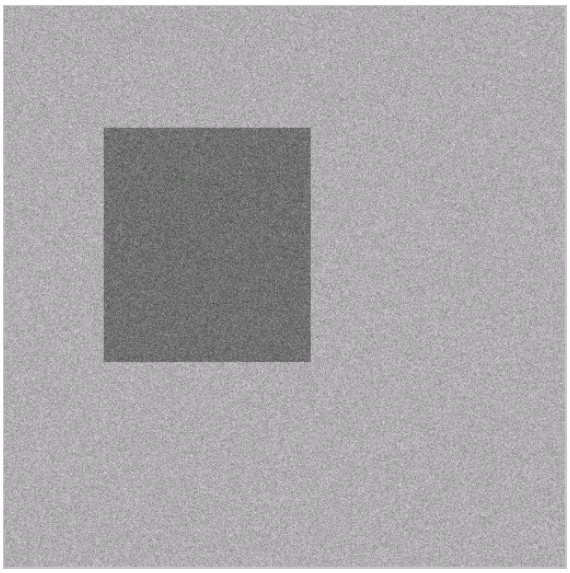
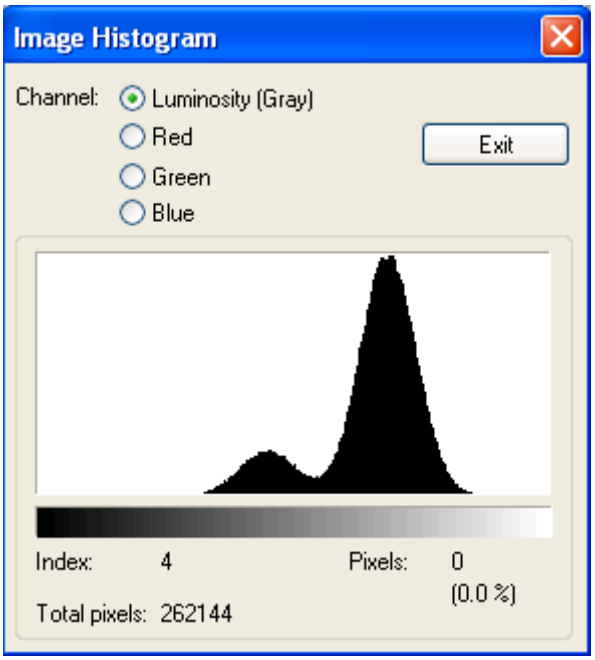
EECS101: HOMEWORK #3 SOLUTION

1. image1
   a) Display



   b) Histogram



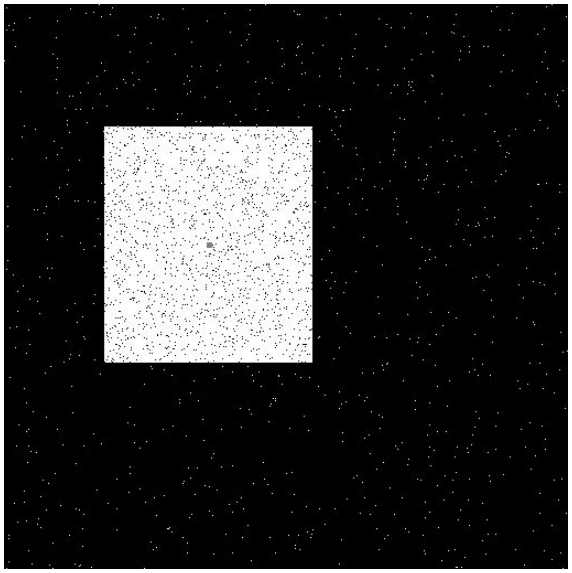
   c) Threshold = 139

   d) $A = \sum_i \sum_j b_{i,j}$ =39605

$$\bar{x} = \frac{\sum_i \sum_j j b_{i,j}}{\sum_i \sum_j b_{i,j}} = 186$$

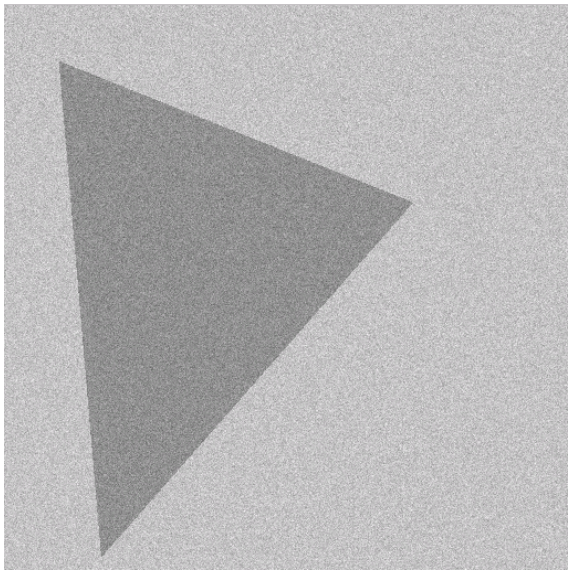
$$\bar{y} = \frac{\sum_i \sum_j i b_{i,j}}{\sum_i \sum_j b_{i,j}} = 218$$

   e) Binary image

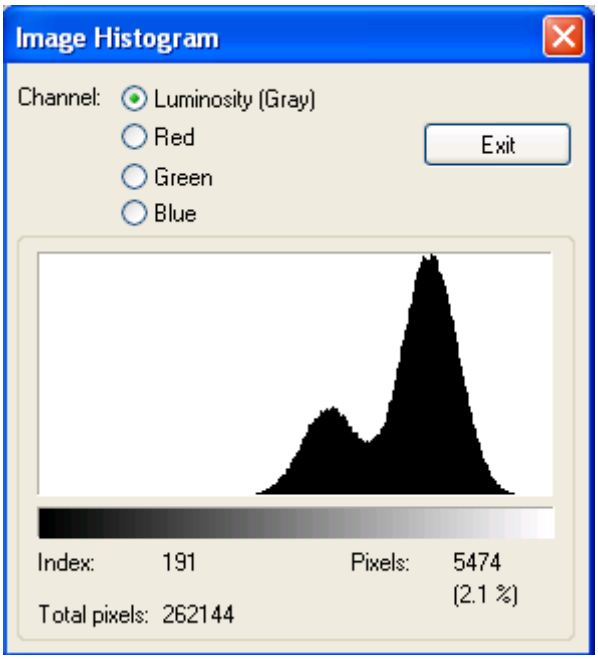


The gray point indicates the center.

2. image2
    a) Display



    b) Histogram



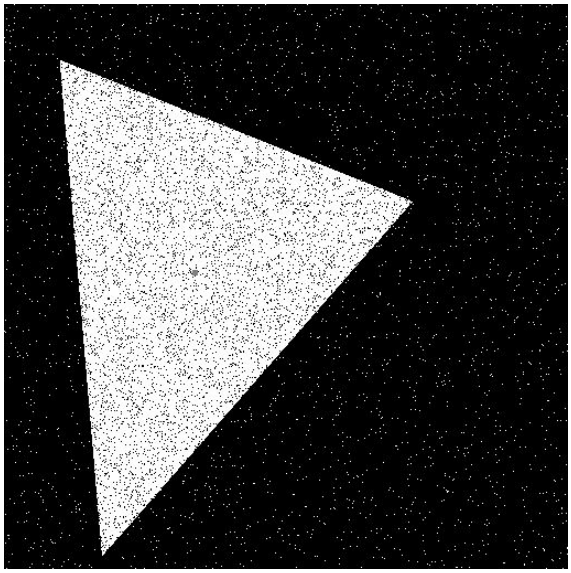
c) Threshold = 164

d) $A = 65843$

$\bar{x} = 173$

$\bar{y} = 243$

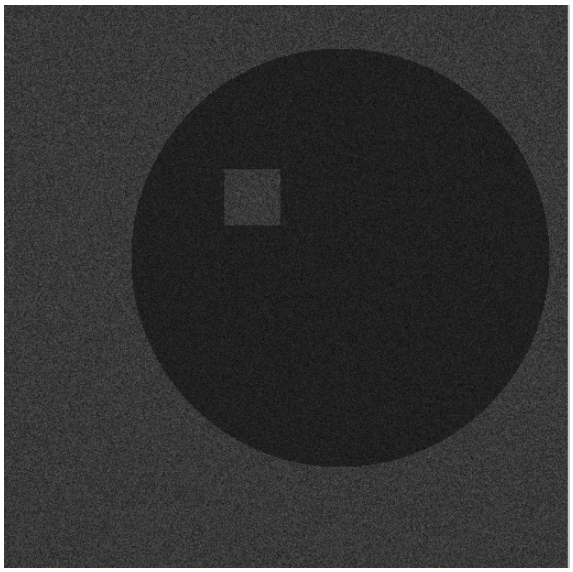
e) Binary image



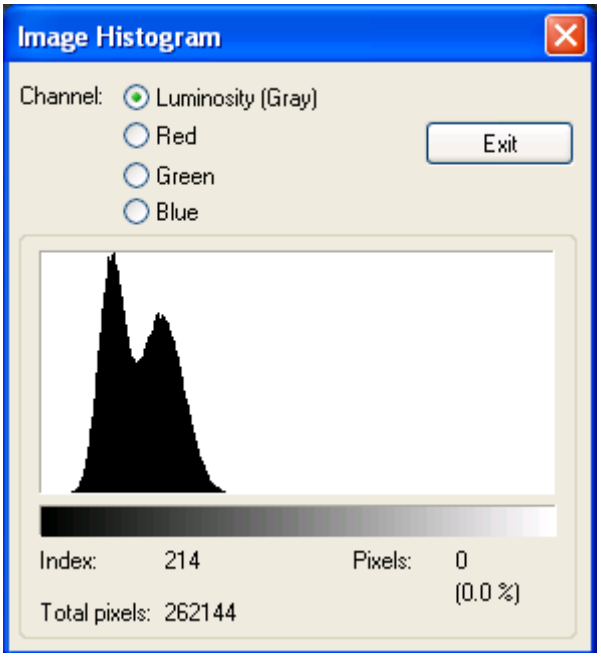
The gray point indicates the center.

3. image3

a) Display



b) Histogram



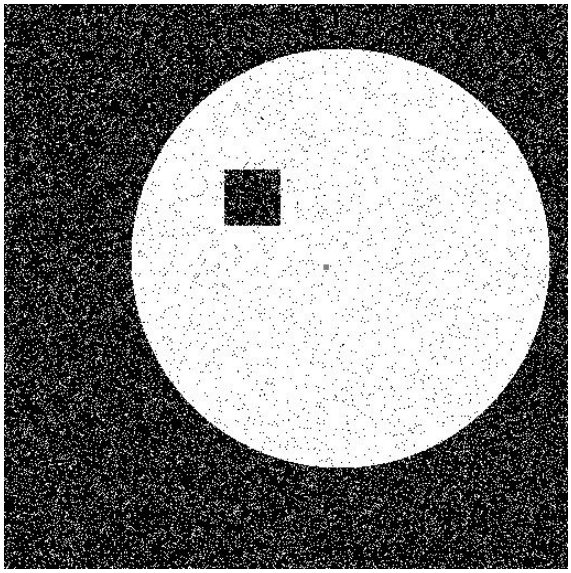
c) Threshold = 48

d) $A = 129498$

$\bar{x} = 292$

$\bar{y} = 238$

e) Binary image



The gray point indicates the center.

4. 8-connected component

The algorithm uses two 2D arrays, i.e., I(x,y) and label(x,y). I contains the image and label has the same size as I and contains the label for each pixel. It also has a variable K which is the number of components seen so far. It works by looking at the eight neighbors of the current one. To handle cases where components are connected to the bottom (as shown below), a second pass from bottom to top is needed.

1										1
	1								1	
		1						1		
			1				1			
				1		1				
					1					

Therefore, the algorithm passes the image twice. One from top to bottom, left to right and the other from bottom to top, left to right. In the first pass, it checks if the current pixel is 1. If so, it then looks at its eight neighbors. If none of its value 1 neighbors has been labeled, increase K and label itself K. If some neighbors are 1 and have been labeled, use the smallest label of these neighbors, say L. After this, it looks at the neighbors again. If any of the neighbors is 1 and has not been labeled or has a label larger than L, label it L. If the current pixel is 0, nothing happens. In the second pass, the job is to correct wrongly labeled pixels. It is done by using a 3x3 window. If the current pixel is 1, then every pixel of value 1 inside the window will be given the smallest label of those pixels.

The algorithm works as follows:

```
Set label(x,y) to zero
Set K to zero
For each row in I from top to bottom
        For each column in I from left to right
                If the current pixel I(x,y) is 1
                        If any of its eight neighbors is 1 and has been labeled
                                L = the smallest label of these neighbors
```

```
                label(x,y) = L
            Else
                Increase K by 1
                Label it K
                L = K
            If any of its eight neighbors is 1 and has not been labeled
                Label those neighbors L
            If any of its eight neighbors have label larger than L
                Label those neighbors L

For each row in I from bottom to top
    For each column in I from left to right
        If the current pixel I(x,y) is 1
            L = the smallest label of all value 1 pixels inside the 3x3 window
            Label all value 1 pixels L
```