CSE-573: Computer Vision and Image Processing

Project-3

Name: Anunay Rao

UBPerson#: 50291493

UBIT: anunayra

Task 1: Morphology Image Operations

(a) The two morphology image algorithms to remove the noise from image are:

(A ∘ B) • B : This is a series of compound operations namely, opening followed by closing the image A with the structural element B. (res_noise1.jpg)
Here, the structural element used is of size 3x3. With center most element as the origin.

(A • B) ∘ B : This is also a series of compound operations namely, closing followed by opening the image A with the structural element B.(res_noise2.jpg)
Here, the structural element used is of size 3x3. With center most element as the origin.

Opening is defined as carrying out erosion on image A with structural element followed by dilation with the same structural element. Whereas, closing is defined as carrying out dilation on Image A with structural element followed by erosion with the same structural element.

The erosion morphological operation generates an output image ‘g’ from an input image ‘f’ using a structuring element ‘h’ where

g(x,y) = 1, if h completely fits f
0, otherwise
The dilation morphological operation generates an output image ‘g’ from an input image ‘f’ using a structuring element ‘h’ where
g(x,y) = 1, if h hits or intersects f
0, otherwise

Figure 1:res_noise1.jpg

Figure 2: res_noise2.jpg

(b) The two results are exactly same as shown above.

(c) To extract the boundary from an image A the following operation is performed:
O = A – (A!B) which is subtract the image eroded with the structural element B from the original image. O is the output image showing boundary.
B = [[1,1,1],[1,1,1],[1,1,1]]

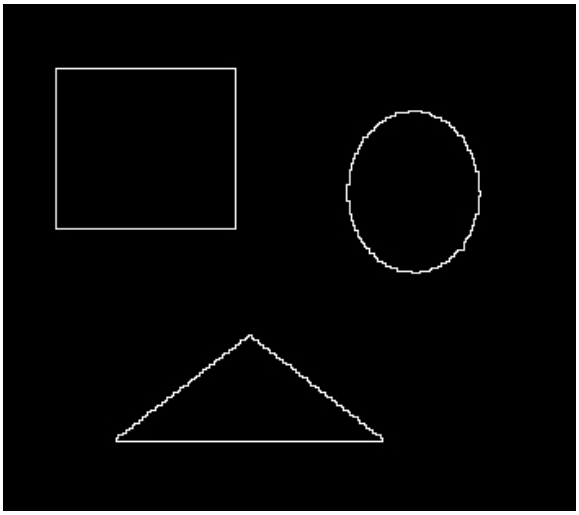

Figure 3: res_bound1.jpg

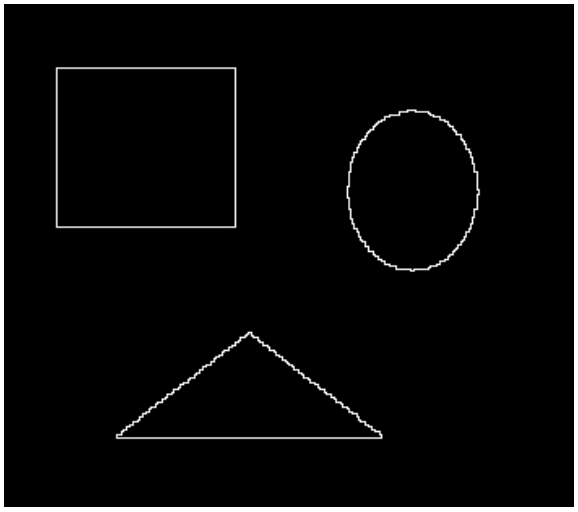

Figure 4: res_bound2.jpg

Task 2: Image Segmentation and point detection

(a) Point Detection:
Here the formulation measures the weighted difference between the centre point and its neighbours. A point has been detected at the location on which the mask is centred if $|R| \geq T$ where T is a non-negative threshold
R is the sum of products of the coefficients with the gray levels contained in the region encompassed by the mark. Here R = [[-1,-1,-1],[-1,8,-1],[-1,-1,-1]]
Threshold was found to be T = 1000.
Coordinates of the detected point are : 445,-249 (considering the top left corner of the image as origin and the image in fourth quadrant).

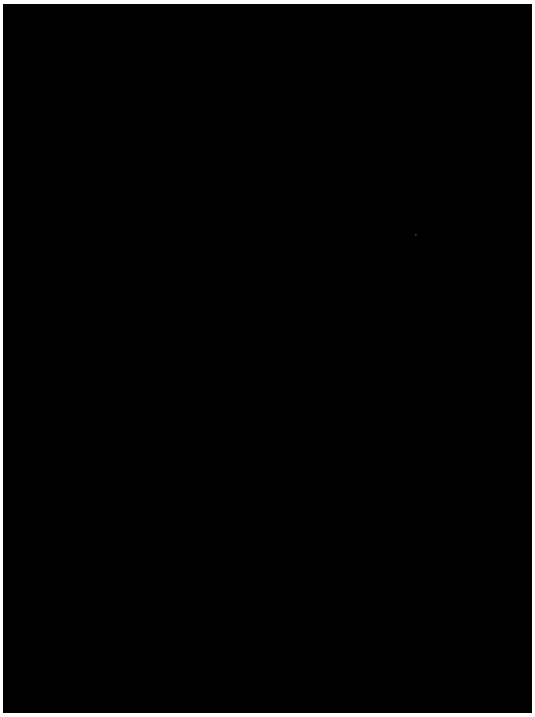

Figure 5: Image after thresholding

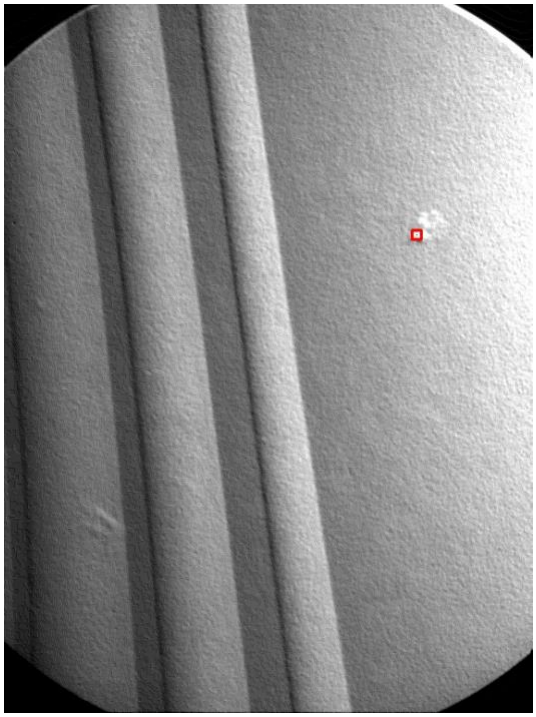

Figure 6: Detected point in Red box

(b) Segmentation: In order to separate the foreground from background we segment the image at some threshold value. By observing the histogram, the threshold was found to be at 204. After thresholding, the image with T=204. I have applied erosion on it to get the components to significant size.

Figure 7: Thresholded Image at T=204

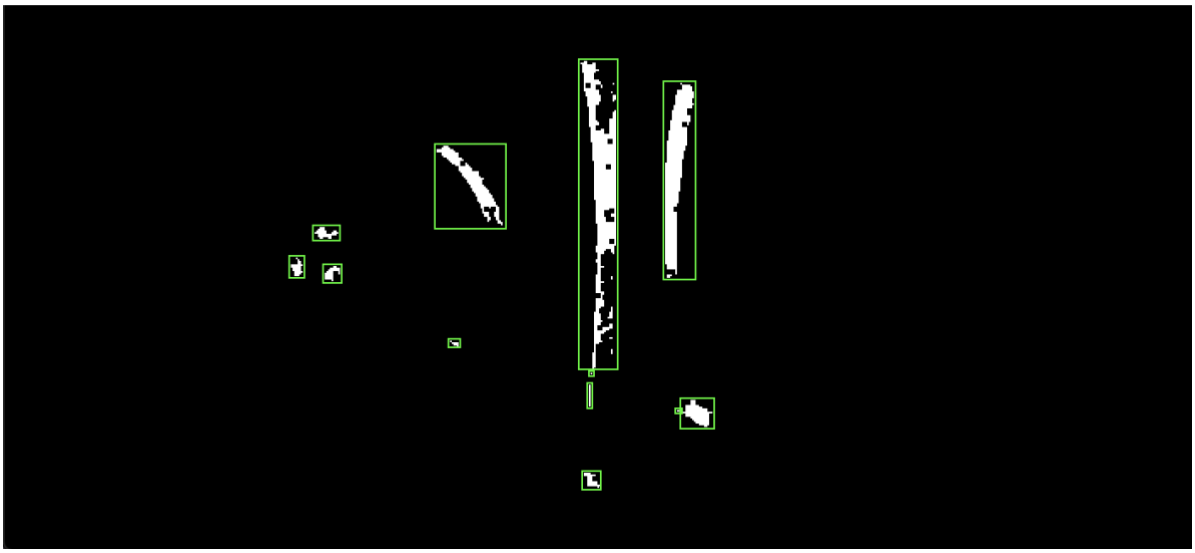

Figure 8: Detected components of significant size. (Eroded with 3x3)

Task 3: Hough Transform

In order to detect blue lines and red lines on the image, I first applied Sobel operator on it to extract the edges from the image. Then thresholding the image to get the binary image. Now creating an accumulator array [rho,theta] to store the votes from the edge points for each theta and row with some thresholding on the rho.
Then calculating which angle got the maximum votes and it was found that 3°and 36° got the maximum votes which implies that most of the edge points belong to these angles for some rho. Now calculating the local maxima to get the rho values.

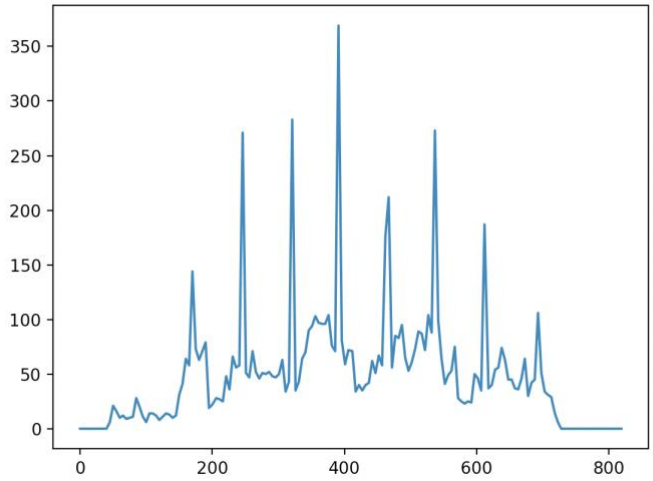

Figure 9: Votes(y-axis) for Blue lines v/s Rho value(x-axis)

From the figure 9, we can select the suitable thresholding for finding the local maximum value of rho. Here, peaks corresponds to the blue lines in the image with 36° theta value.

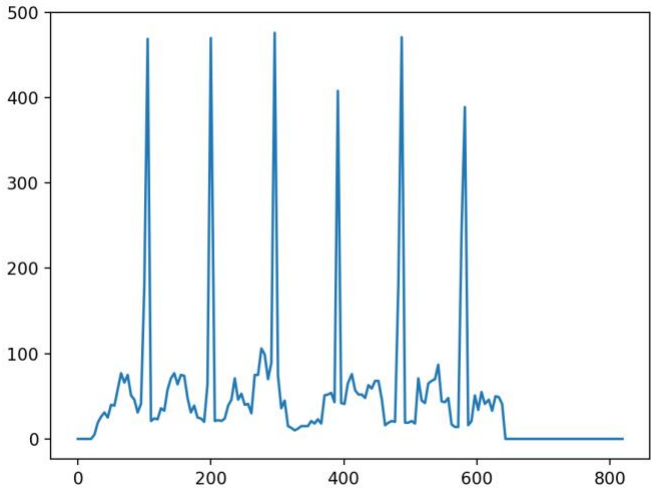

Figure 10: Votes(y-axis) for red line v/s rho value (x-axis)

Similarly from the figure 9, we can select the suitable thresholding for finding the local maximum value of rho. Here, peaks corresponds to the red lines in the image with 3° theta value.

Similarly, for detecting coins taking radius range from 20 to 30 and theta from 0° to 360° and using parametric form.

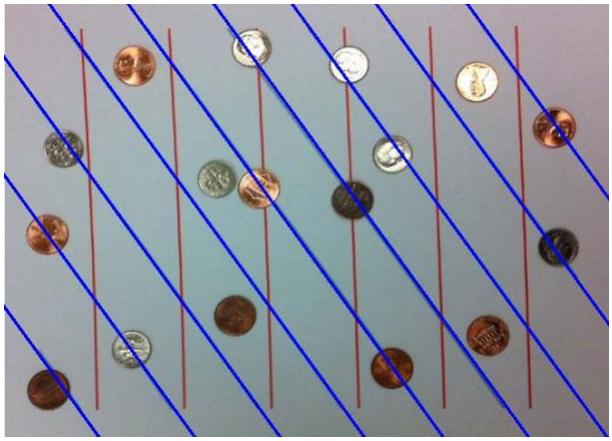

Figure 11:blue_lines.jpg

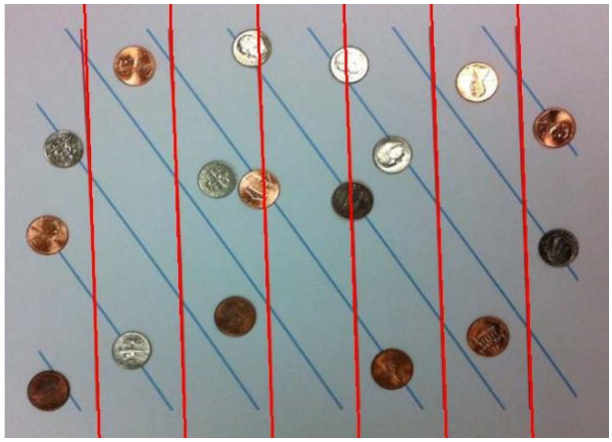

Figure 12: red_lines.jpg

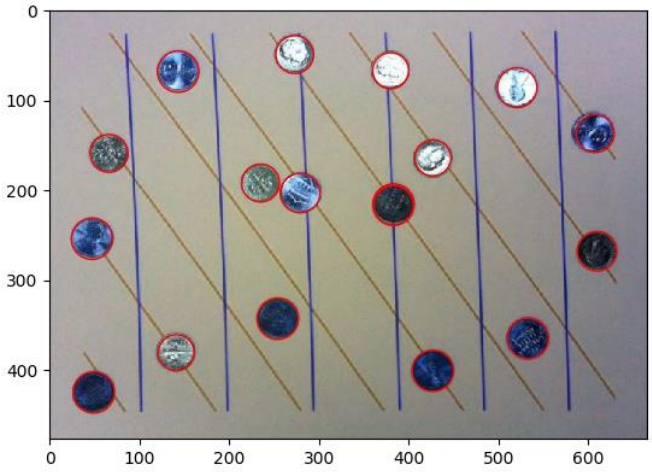

Figure 13: coins.jpg