**Programming Assignment 2**

**Ran Tian 4070588**

**Task1:**

1: Use the edge detection code from programming assignment 1, we get the result binary image after edge detection and thresholding.

2: For every edge element in the image(pixel value is 1), we get the x coordinate and y coordinate, then by the equation:

we map the original point to hough space(parameter space)

3: The hough space matrix is size of (m,n), hough\_space(i,j) represent how many points in the line of original image with parameter (i,j). Here I set the maximum and minimum bound for each parameter, and then set the number of steps for each parameter.

4: After counting, we will select the parameters for the lines. Here I set the threshold value to extract the parameter.

5: After step 4, there are repeated lines we have detected, to eliminate these duplicated lines, I also set the threshold value for . In other words, if the

|<=3, then I regard these two lines are the same and I take the average the same logic apply to determine p.

6: Finally, we get the line parameters. And I plot the result lines and compare with original lines.

**Task2:**

The logic of this task is similar to the task 1, but we get 4 parameters in the hough space, the ellipse representation is:

the edge detection and thresholding part is the same as task1.

1: For every edge element in the image(pixel value is 1), we get the x coordinate and y coordinate, then by the equation above. We can count the number of appearance in the hough matrix. But the time complexity of compute this matrix is O(n^4), where n is the step sizes of four parameters, which is pretty huge. And step size affect the accuracy of our detection results.

2: For this task, in order to check the correctness of my algorithm, I am sort of “cheating” -- I reduce the running time by setting the range of a,b,x0,y0 manually, which in other words, I only detect the ellipses with whose parameters are in my range.

**Notice:** Even with this reduction, the running time to compute hough matrix is about 7~10 minutes, so if you don’t want to spend time on generating this matrix, just skip section 1 since I have save the hough matrix in **my\_running\_hough\_matrix.mat** and I load it in the start of section 2.

3: Like task1, I use a threshold value to extract the all possible parameters.

4: After step 3, I get repeated parameters for the same ellipse. To solve it, I set the threshold value to x0 to determine if two results are the same, in other words:

20, for different i,j, I regard them the same ellipse and take the average of those two for the final result. The threshold value is set manually.

5: Finally, I plot the my detected ellipses.

Conclusion: Two tasks share the same logic for Hough transform. The base line is to convert the original space to hough space(parameter space) . Here comes my results:

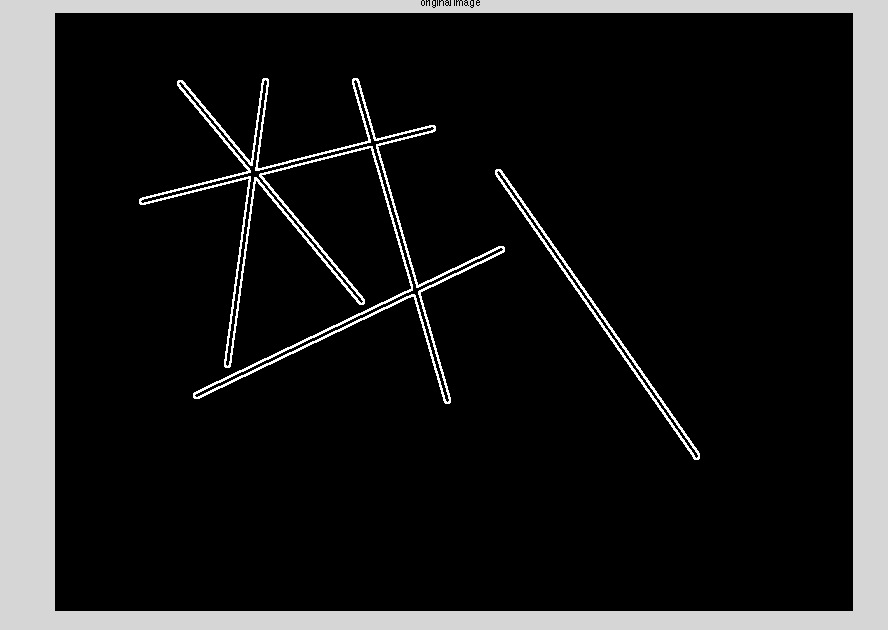
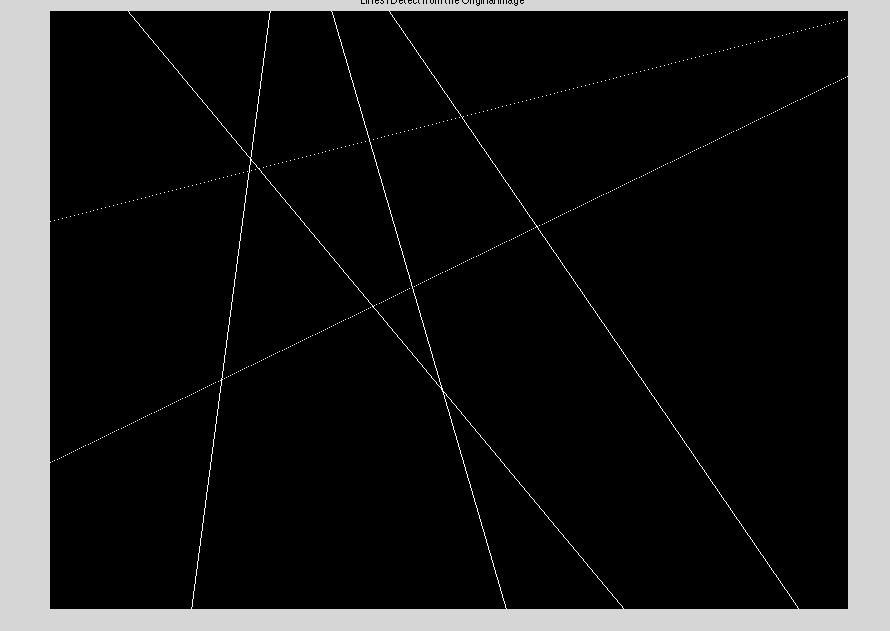


Image After Edge Detection Image from my detection results

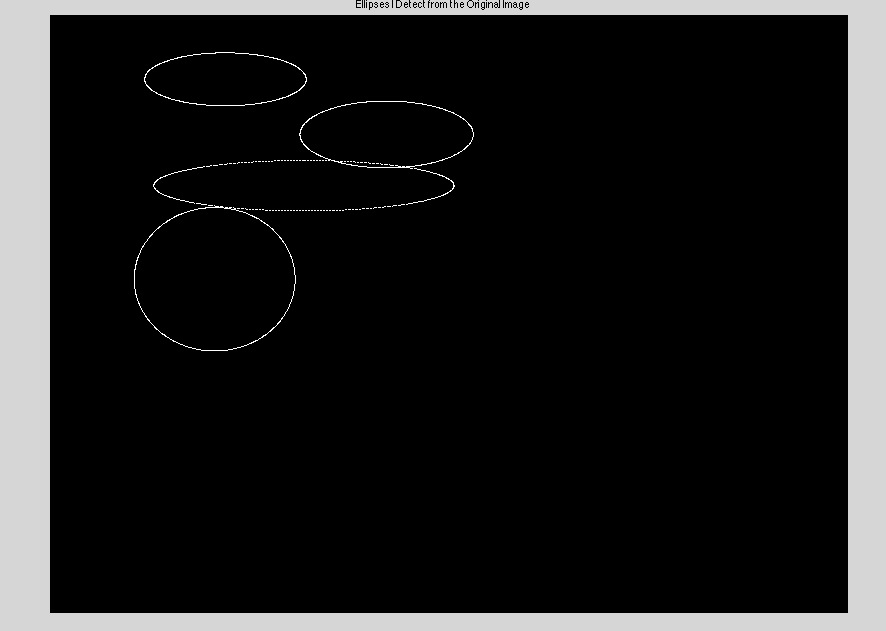
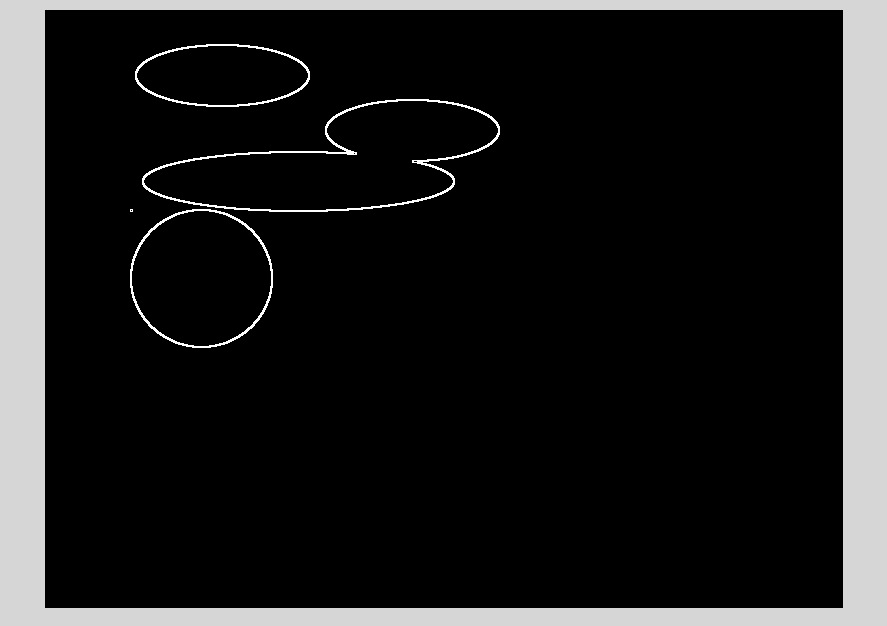


Image After Edge Detection Image from my detection results