Computer Vision assignment 3

Name: Sai Teja Karnati

ID: 659 365 999

Basic Concepts, Clarity and Algorithm Descriptions

Hough Transform

- Sometimes when we want to detect straight lines, curves or some other simple shapes. A simple edge detector can be used to check points on a certain edge or a simple curve but when we miss some points present on the shapes due to irregularity of image data or some noise we don't get a proper representation of the shapes.
- To solve this problem we can use a hough transform which groups together point explicitly by a voting procedure which is discussed below.
- Here we are simply detecting lines using the hough transform. There are also complex algorithms where we can detect other shapes like curves and ellipses using the hough Transform

Algorithm description

The simplest case of Hough transform is detecting straight lines. In general, the straight line **y = mx + b** can be represented as a point (b, m) in the parameter space. However, vertical lines pose a problem. They would give rise to unbounded values of the slope parameter m
First I am creating an accumulator to detect the line of the form
rho = x*cos(theta) + y*cos(theta) so here (rho,theta) are the parameters
We associate a line with a single pair of (rho,theta)
Here the dimension of the Accumulator is two since there are two unknown parameters. The accumulator is incremented every time we expect a line to pass through at a certain r and theta Then by finding the local maxima applying a threshold value in the accumulator space we find out the most likely lines in the image. By varying the quantizing levels of rho and theta different outcomes can be achieved. We can see that the accumulator is intersected everytime there is an edge. So the number of bright points in the accumulator is nothing but the number of lines here. Below are different values of rho and theta tried out on all three images.

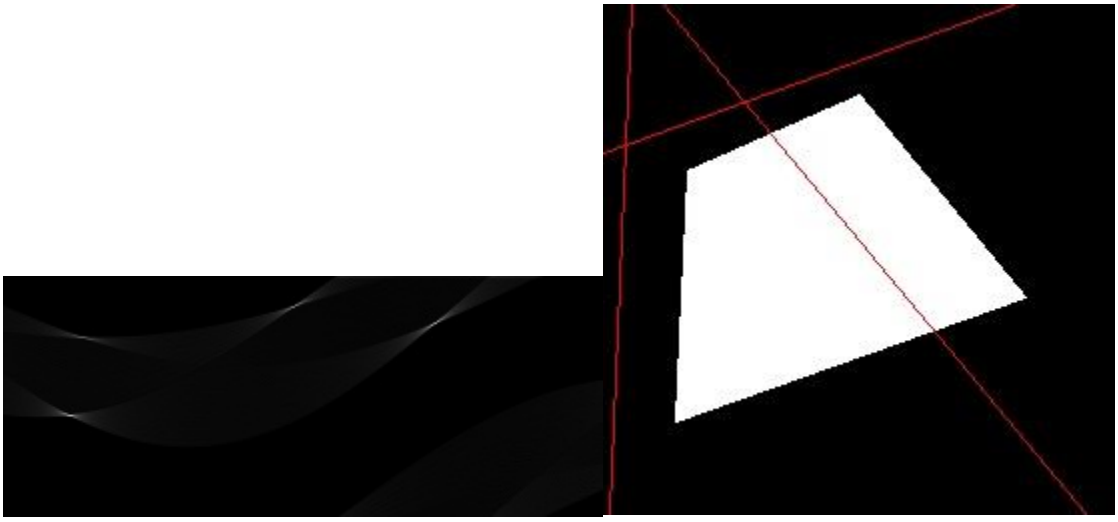
We will test the following parameter for each of the given 3 test images. We explore one best set of parameters. We will also discuss the observations at the end of each test case.

Parameters experimenting

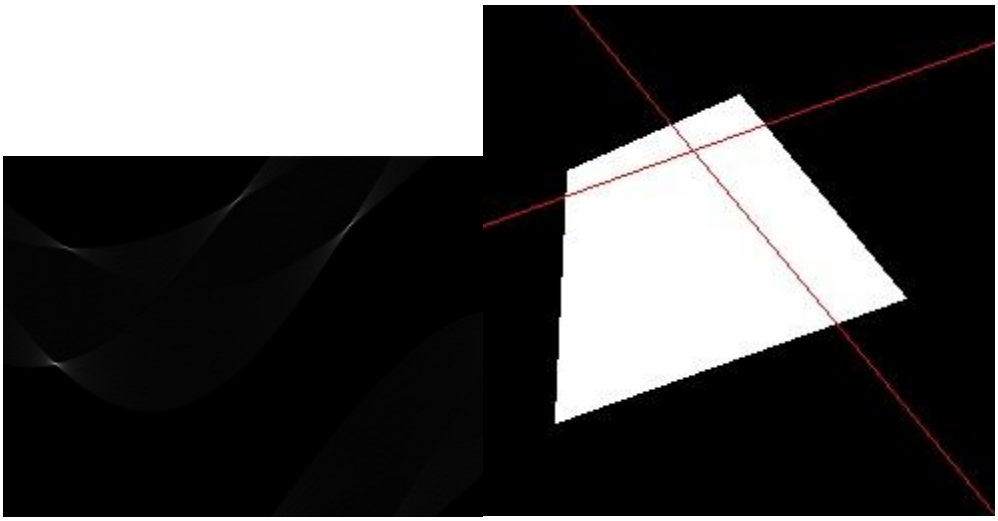
#	Rho Quant	Theta Quant	Threshold
1	120	300	120
2	180	240	150
3	360	460	40
4	200	500	30
5	300	120	40
6	60	200	35
7	240	100	50
8	Exploration-Best	Exploration-Best	Exploration-Best

Experimenting with Test Image

Case 1: Image = Test, Rho Quant =120, Theta Quant = 300, Threshold = 120

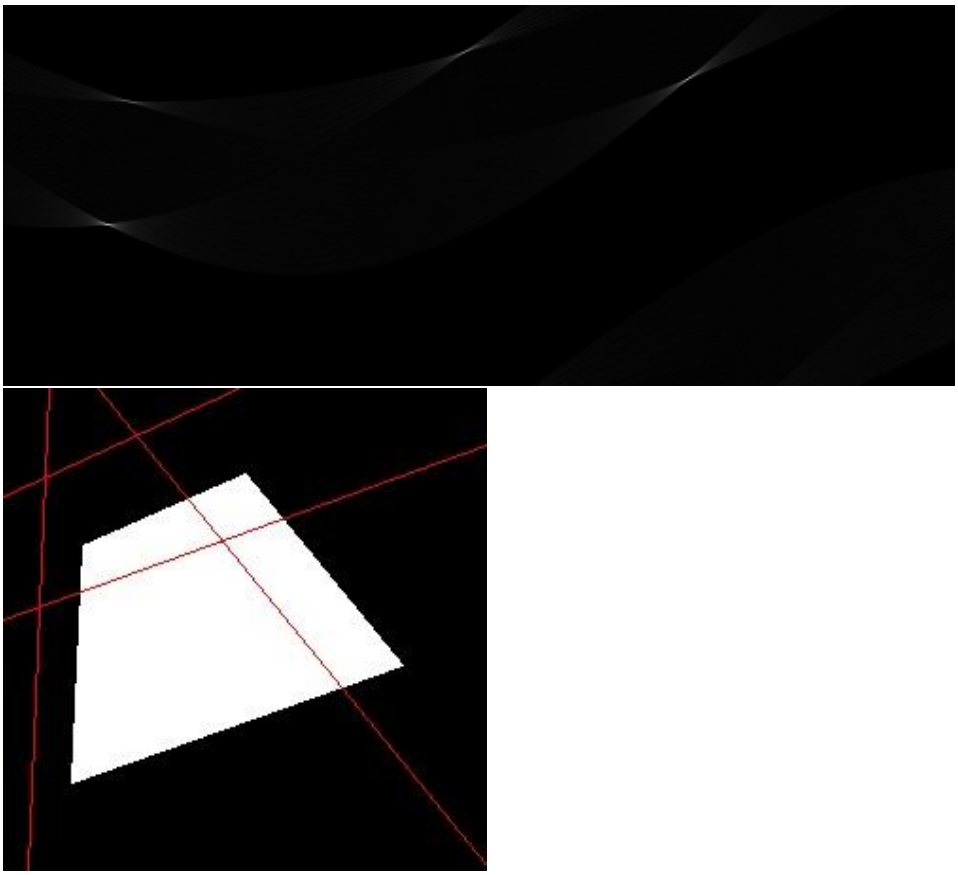


Case 2: Image = Test, Rho Quant =180, Theta Quant = 240, Threshold = 150

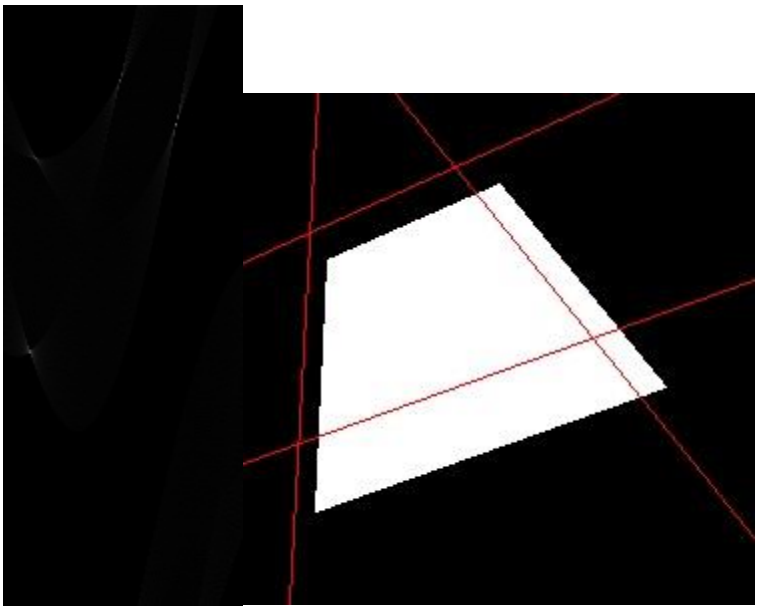


Case 3: Image = Test, Rho Quant =360, Theta Quant = 460, Threshold = 40

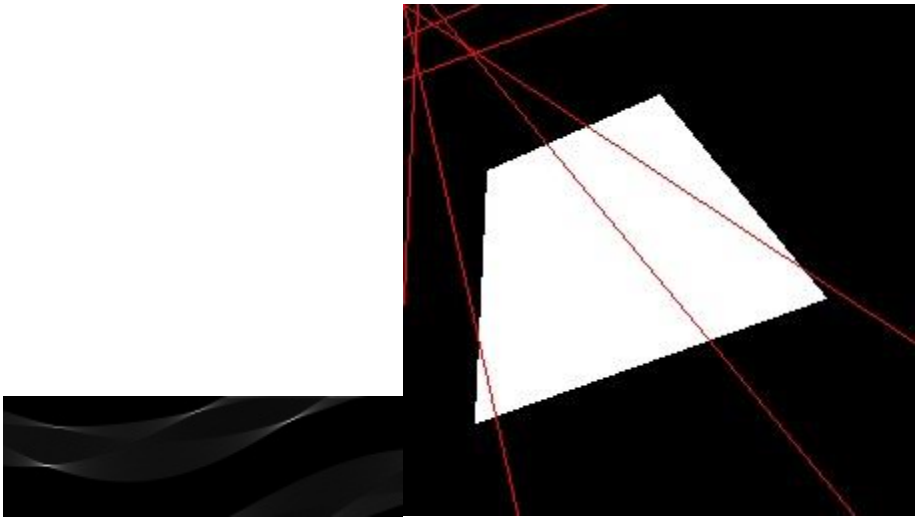
Case 4: Image = Test, Rho Quant =200, Theta Quant = 500, Threshold = 30



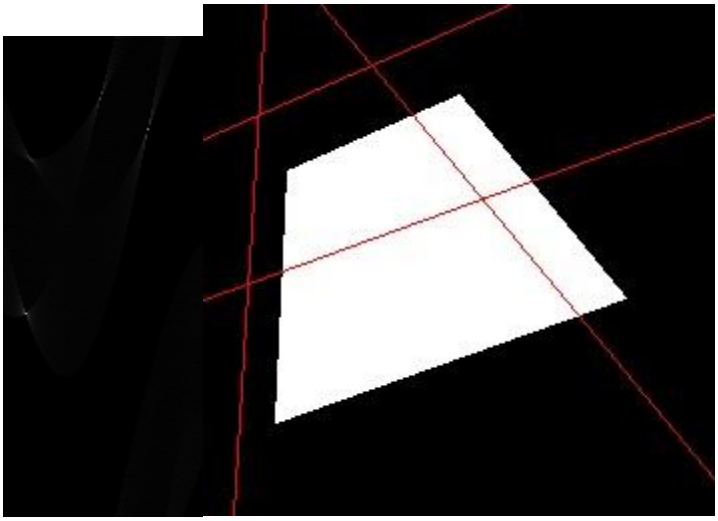
Case 5: Image = Test, Rho Quant =300, Theta Quant = 120, Threshold = 40



Case 6: Image = Test, Rho Quant =60, Theta Quant = 200, Threshold = 35



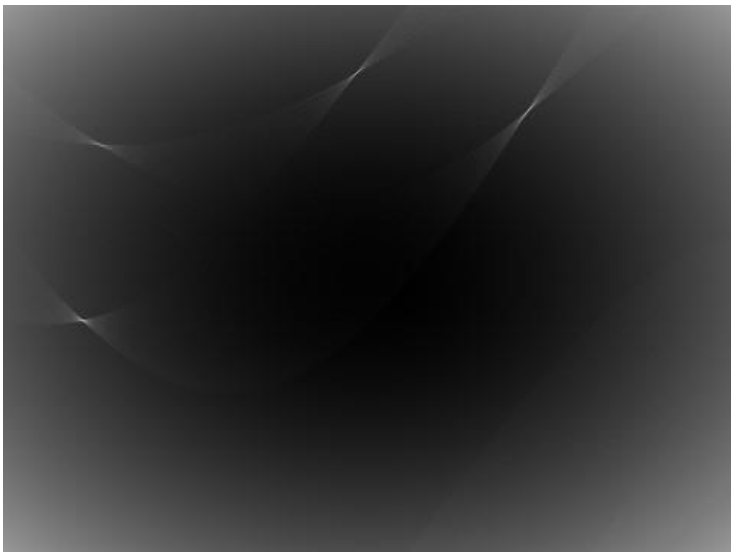
Case 7: Image = Test, Rho Quant =240, Theta Quant = 100, Threshold = 50



Some exploration:

Case 8: Image = Input, Rho Quant =360, Theta Quant = 480, Threshold = 50

Added a filter for better visualization.



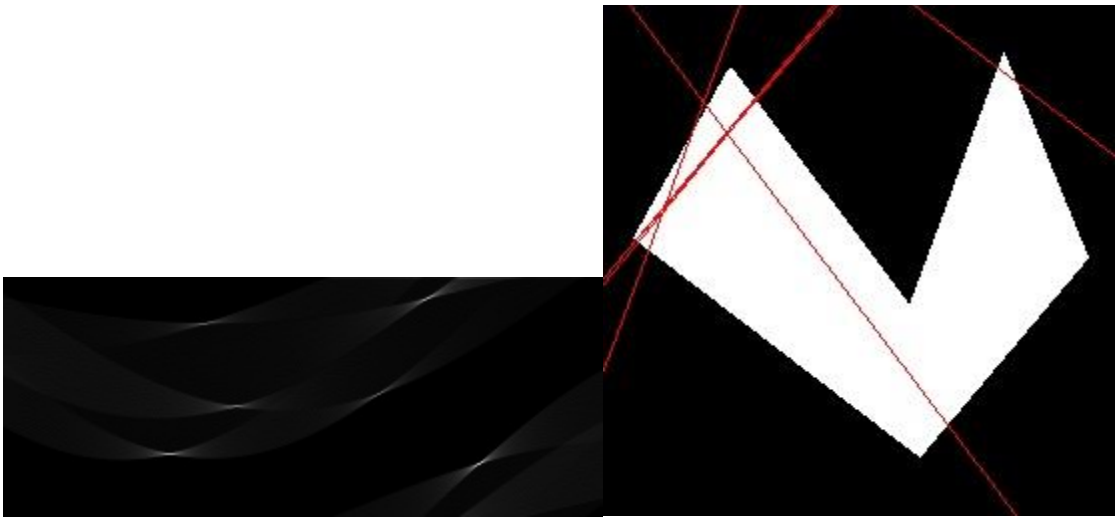
Comparison and Analysis for different Quantization values

- We Observe the best Line Detection on Test Image 1 for rhoquant = 360 and thetaquant= 480 and threshold = 50. We also observe that as the rhoquant value changes the shapes shift away from the original value.
- Also keeping the threshold around 30 - 50 is the best choice
- The edges are detected quite cleanly since this is a simple image but we might have to use canny edge detector when it comes to complex images.

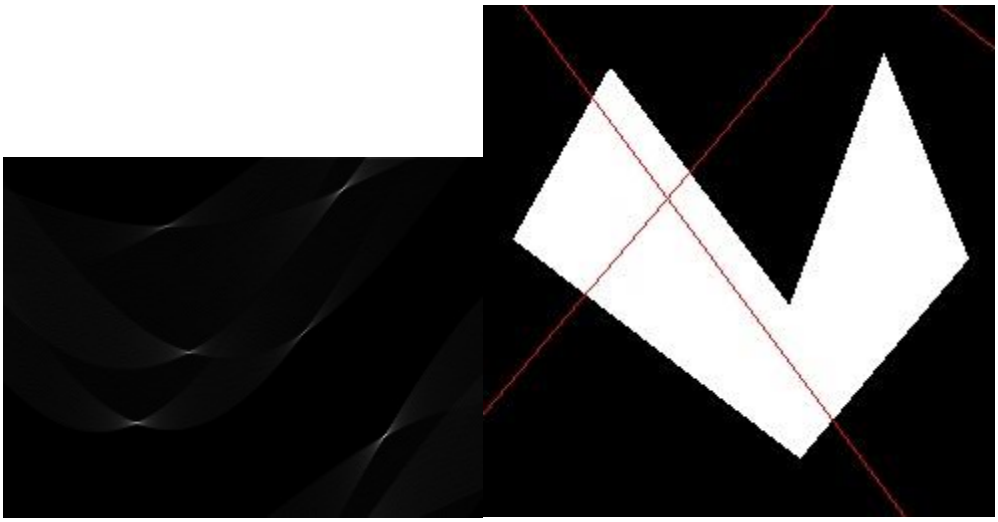
.

Experimenting with Test2 Image

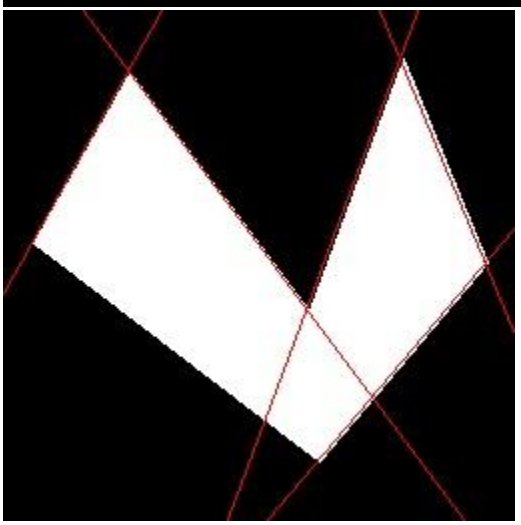
Case 1: Image = Test2, Rho Quant =120, Theta Quant = 300, Threshold = 120



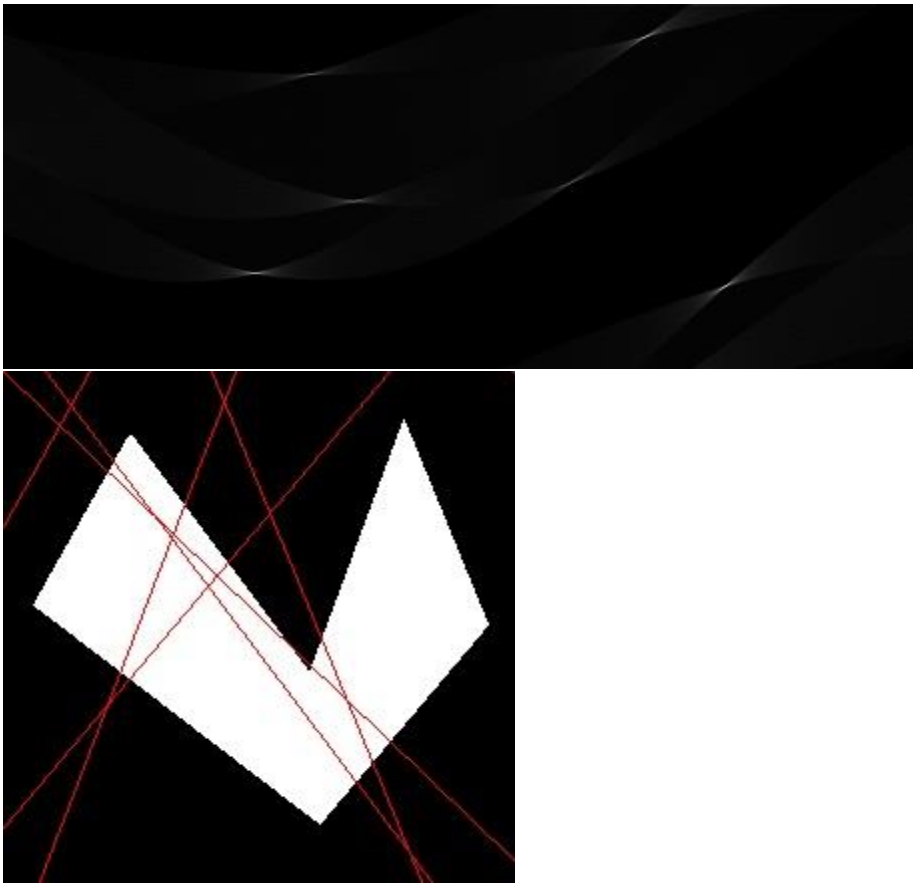
Case 2: Image = Test2, Rho Quant =180, Theta Quant = 240, Threshold = 150



Case 3: Image = Test2, Rho Quant =360, Theta Quant = 460, Threshold = 40

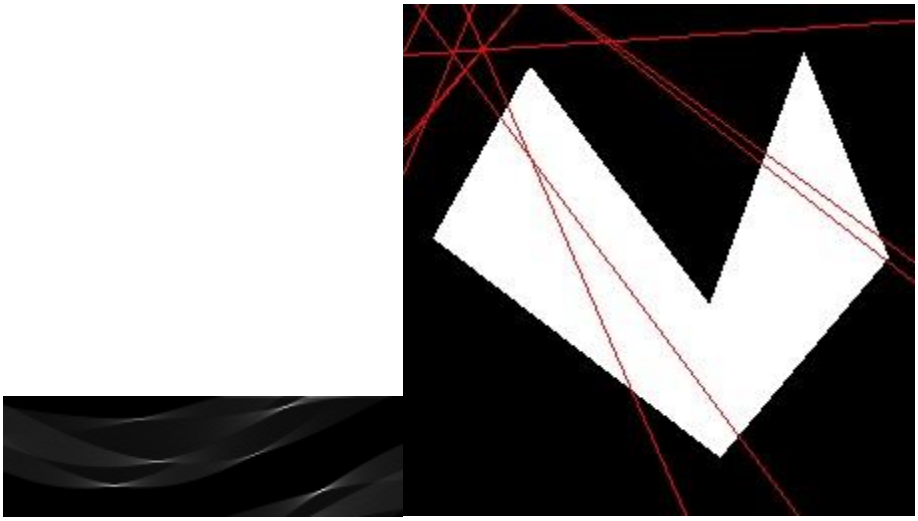


Case 4: Image = Test2, Rho Quant =200, Theta Quant = 500, Threshold = 30

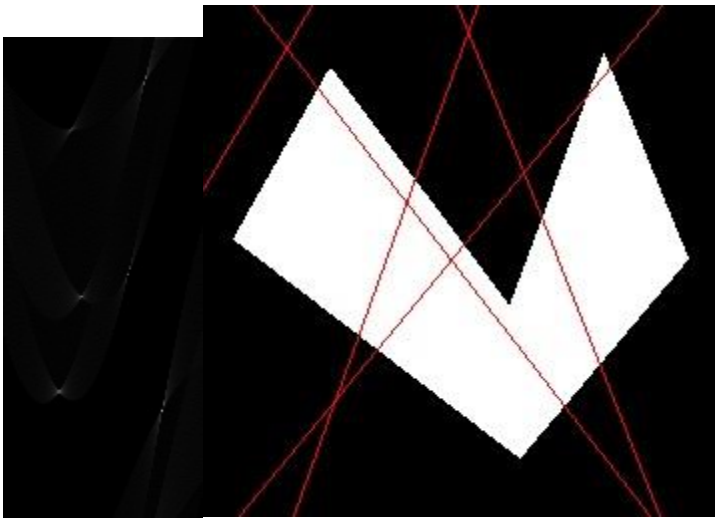


Case 5: Image = Test2, Rho Quant =300, Theta Quant = 120, Threshold = 40

Case 6: Image = Test2, Rho Quant =60, Theta Quant = 200, Threshold = 35

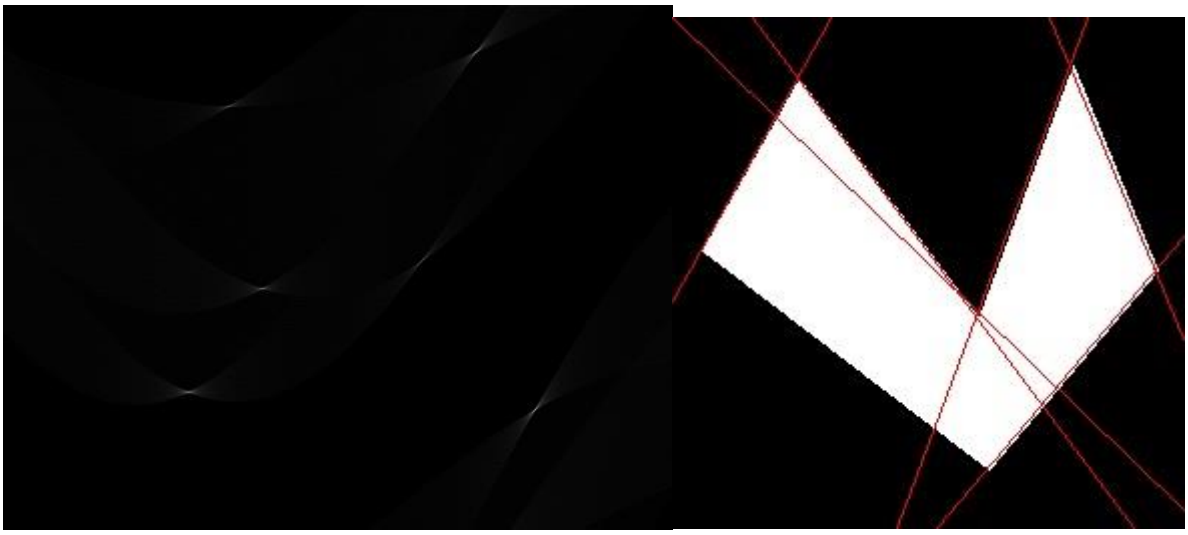


Case 7: Image = Test2, Rho Quant =240, Theta Quant = 100, Threshold = 50



Some exploration:

Case 8: Image = Input, Rho Quant =360, Theta Quant = 460, Threshold = 20



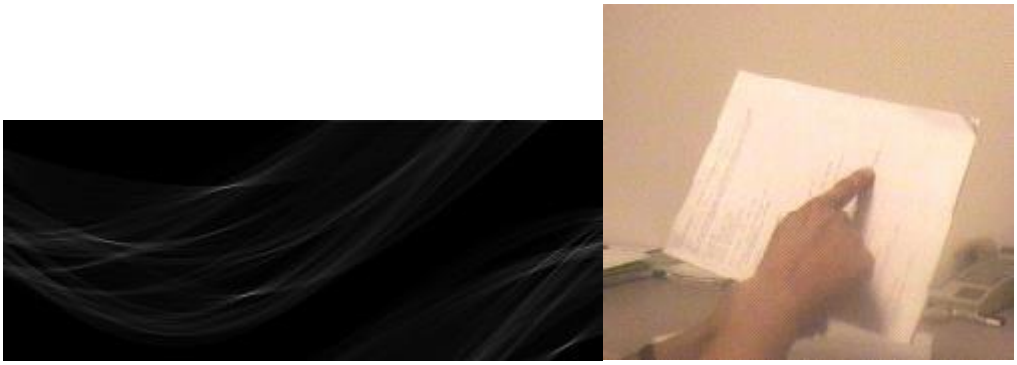
Comparison and Analysis for different Quantization values

- We Observe the best Line Detection on Test Image 2 for rhoquant = 360 and thetaquant = 460 and threshold = 20-40. We also observe that as the rhoquant value changes the shapes shift away from the original value.

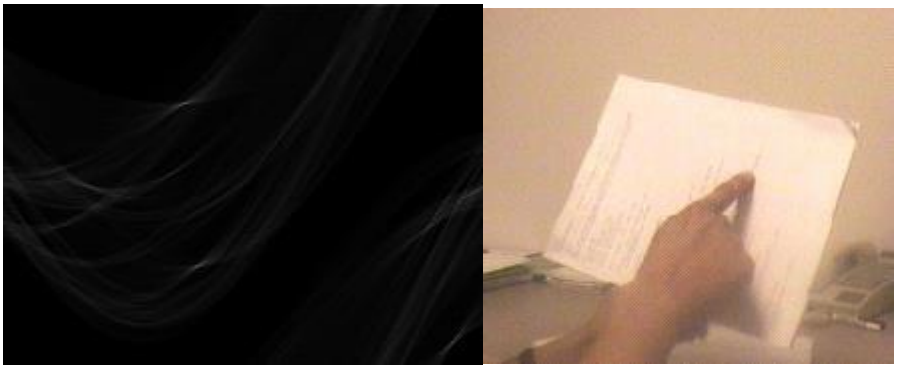
- Also keeping the threshold around 20 – 40 is the best choice. As the threshold decreases we observe unnecessary lines also in the image.
- Even in the best choice I see that a line is missed from detection. I think sometimes hough transform can miss the edges in the boundaries. They might show up when different threshold levels are set up.

Experimenting with Real Input Image

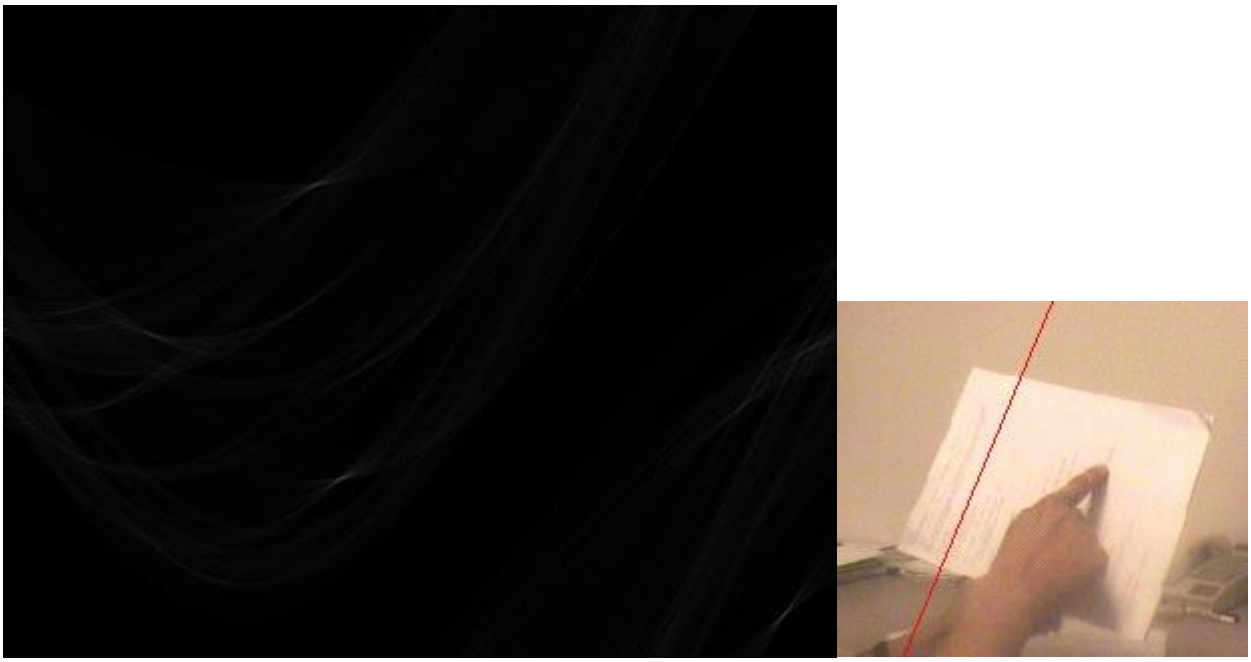
Case 1: Image = Input, Rho Quant =120, Theta Quant = 300, Threshold = 120



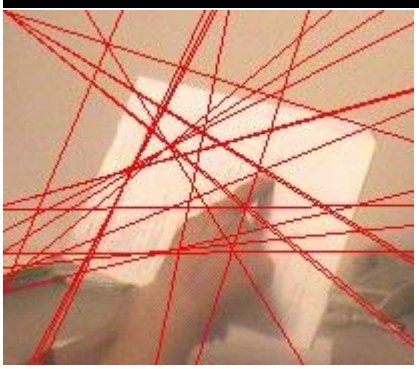
Case 2: Image = Input, Rho Quant =180, Theta Quant = 240, Threshold = 150



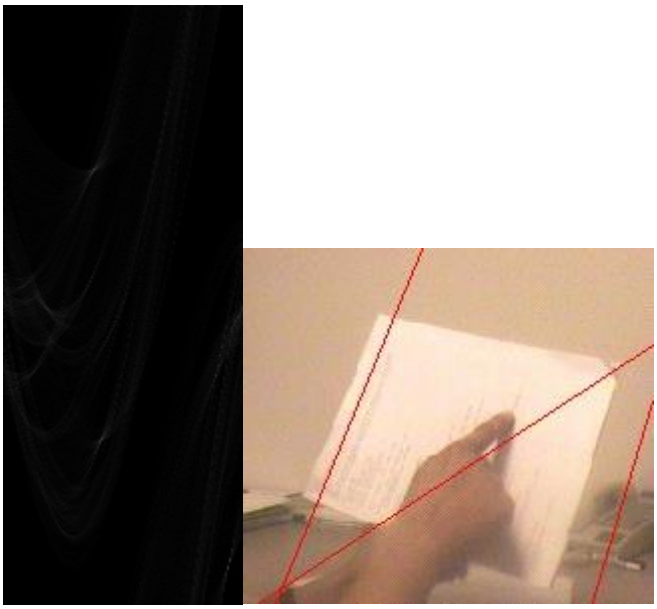
Case 3: Image = Input, Rho Quant =360, Theta Quant = 460, Threshold = 40



Case 4: Image = Input, Rho Quant =200, Theta Quant = 500, Threshold = 30

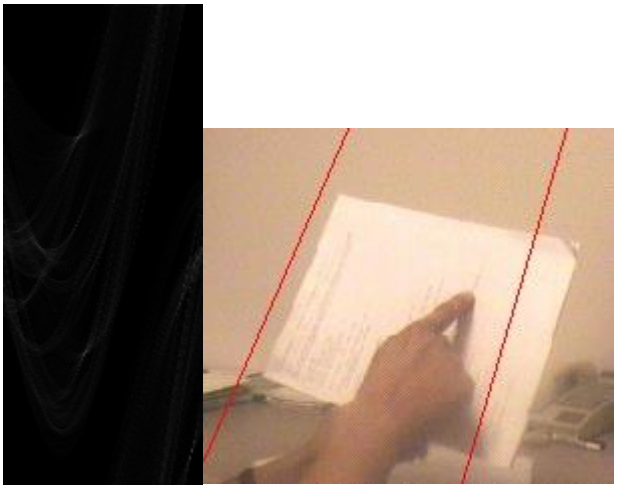


Case 5: Image = Input, Rho Quant =300, Theta Quant = 120, Threshold = 40



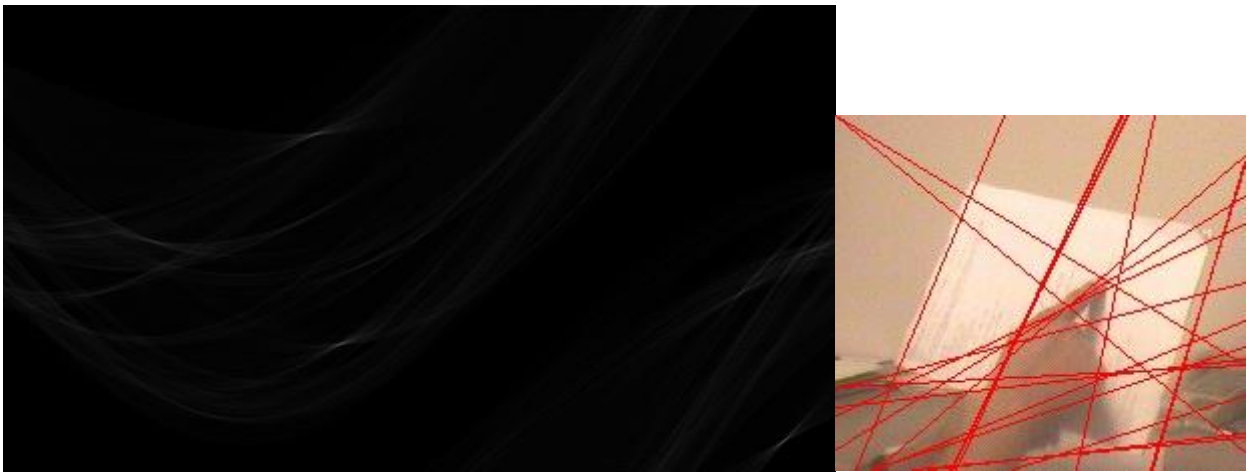
Case 6: Image = Input, Rho Quant =60, Theta Quant = 200, Threshold = 35

Case 7: Image = Input, Rho Quant =240, Theta Quant = 100, Threshold = 50



Some exploration:

Case 8: Image = Input, Rho Quant =280, Theta Quant = 500, Threshold = 25



Comparison and Analysis for different Quantization values

- We observe the best Line Detection on input Image for rhoquant = 280 and thetaquant = 500 and threshold = 25. We also observe that as the rhoquant value changes the shapes shift away from the original value
- Even in the best choice I see that a line is missed from detection. I think sometimes hough transform can miss the edges in the boundaries. They might show up when different threshold levels are set up.

- We observe that the size of the rhoquant values is somewhat correlated. Because the previous two images had a good out at rhoquant 360 when the size of them was around 250. While this has a good one at around 280 when its size is around 190.

- **Significant Intersections in parameter space** occur at the corners because at corners there is possibility for multiple lines to be present. Also at curves which have different tangential slopes at different points like here the curve in the hand. Sometimes we observe many lines near the curve of the hand.