

Introduction:

Hough Transform is a popular algorithm for line detection in computer vision. It is a feature extraction technique that is widely used to identify and detect lines in an image. The Hough Transform works by converting the image into a parameter space, where the parameters are the slope and the intercept of the lines in the image. By doing this, it is possible to detect lines that are not necessarily straight or of a certain length.

In this report, we will analyze the code for doing line detection using Hough Transform. We will explain the working of the Hough Transform algorithm, analyze the code line by line, and provide an output of the algorithm.

Analysis:

The code starts by importing the necessary libraries: NumPy, OpenCV, Math, PIL (Python Imaging Library), and Matplotlib. These libraries are used for performing mathematical operations, image processing, and visualization.

The next block of code defines the **hough_line** function, which is used for performing the Hough Transform on an input image. This function takes the input image and some parameters as inputs and returns the accumulator array, the theta array, and the rho array. The theta array contains the angles at which the lines are detected, and the rho array contains the distance from the origin to the line.

The **plot_line** function is defined to visualize the output of the Hough Transform algorithm. It takes the input image, the accumulator array, the theta array, and the rho array as inputs and plots the output using the Matplotlib library.

In the main section of the code, the input image is read using the OpenCV library, and then converted into grayscale using the PIL library. The **cv2.Canny** function is used to detect the edges in the grayscale image.

The **hough_line** function is then called, and the output is stored in the **accumulator**, **thetas**, and **rhos** variables. These variables are then passed to the **plot_line** function to visualize the output. The threshold and angle bin values are set, and the **argwhere** function is used to find the peak indices in the accumulator array that are greater than the threshold value. These peak indices are then used to calculate the slope and intercept of the lines using the theta and rho arrays. Finally, the **ImageDraw** module is used to draw the lines on the original image.

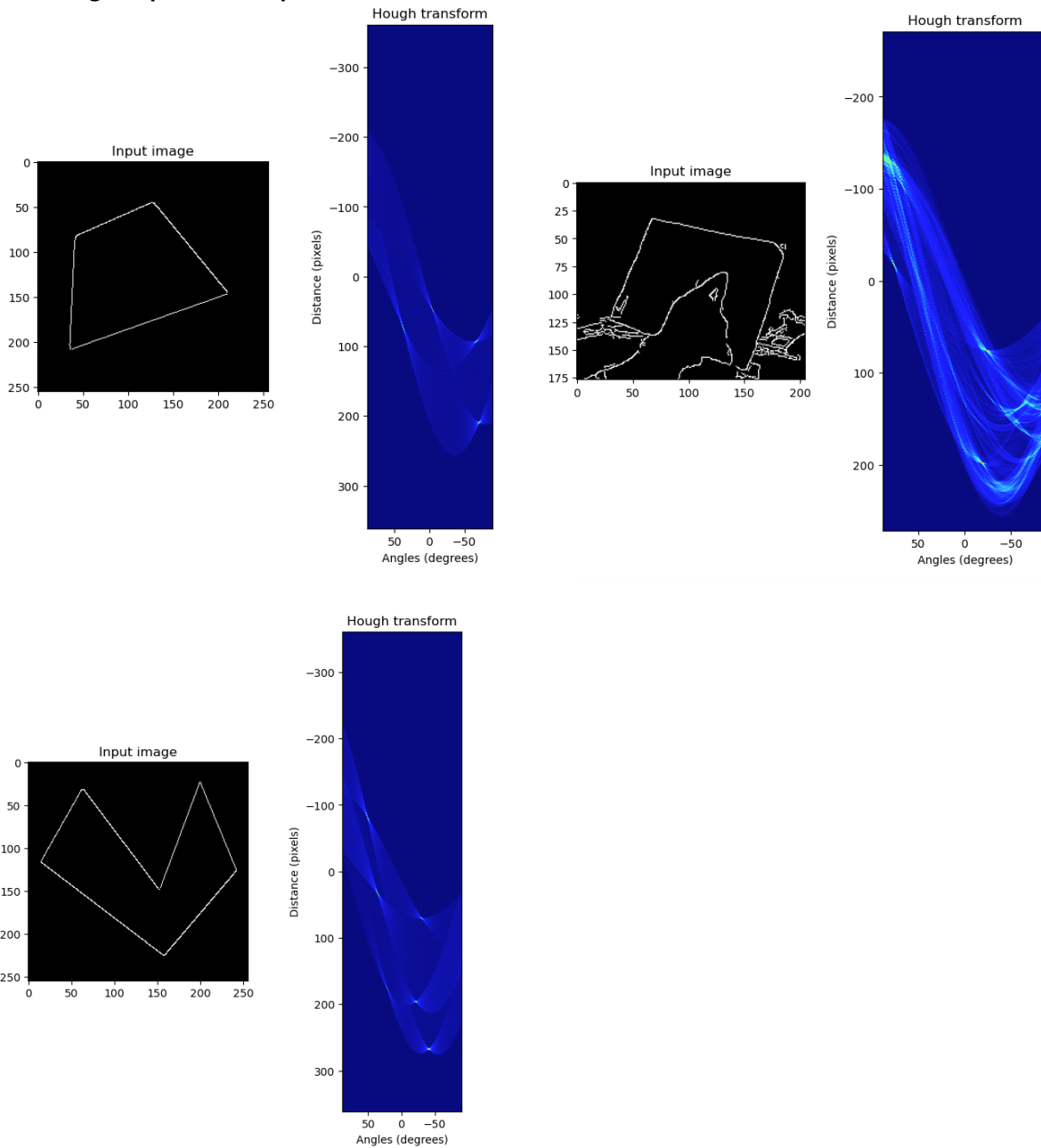
Following improvements can be made,

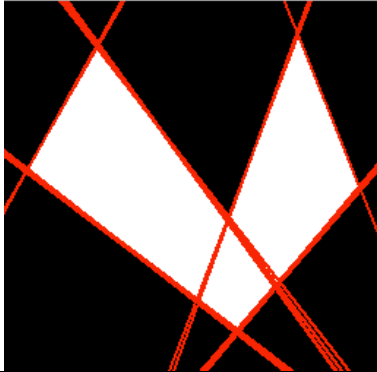
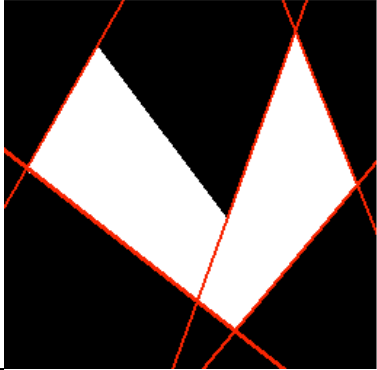
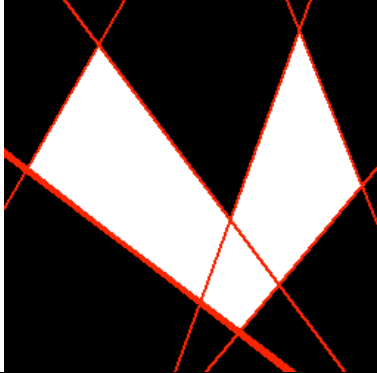
1. Apply non-maximum suppression to eliminate duplicate lines that are detected at different points along the same line.
2. Implement a post-processing step to connect fragmented line segments into longer, more coherent lines.

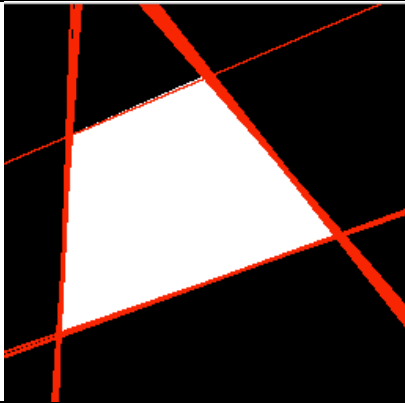
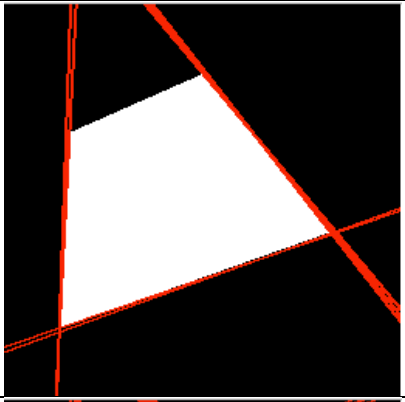
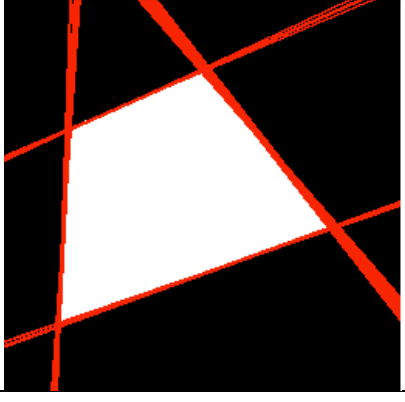
Conclusion:

The Hough Transform algorithm is a powerful tool for detecting lines in an image. The code provided in this report demonstrates how to perform line detection using Hough Transform in Python. By understanding the code and the algorithm, it is possible to apply Hough Transform to other image processing tasks as well.

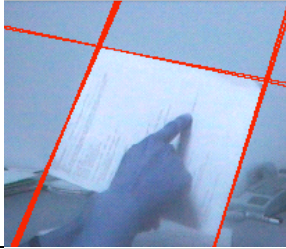

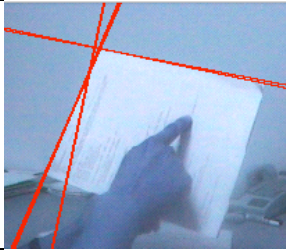
Visualizing the parameter space:



Parameters	Result
Threshold = 60 Angle Bins = 360	
Threshold = 80 Angle Bins = 180	
Threshold = 80 Angle Bins = 360	

Parameters	Result
Threshold = 45 Angle Bins = 360	
Threshold = 50 Angle Bins = 360	
Threshold = 40 Angle Bins = 360	

Venkatakishnan Ranganathan
Computer Vision MP#6

Parameters	Result	
Threshold = 60 Angle Bins = 360		
Threshold = 50 Angle Bins = 360		
Threshold = 65 Angle Bins = 360		
Threshold = 60 Angle Bins = 180	