Hough Transform

Objective:

To implement the Hough transform and test on sample images

Language used: Python

Running the code (mp06.py): ***(requires mp05.py)***

In a terminal, enter:

```
./mp06.py [image name] [theta_factor] [rho_factor]
```

Example usage:

```
./mp06.py input.bmp 1.0 1.0
```

Description of Hough Transform:

- A line can be represented as 'y = mx + c'
- The coordinate space is transformed from Cartesian space to Hough space (ρ & θ) as shown in Figure 1

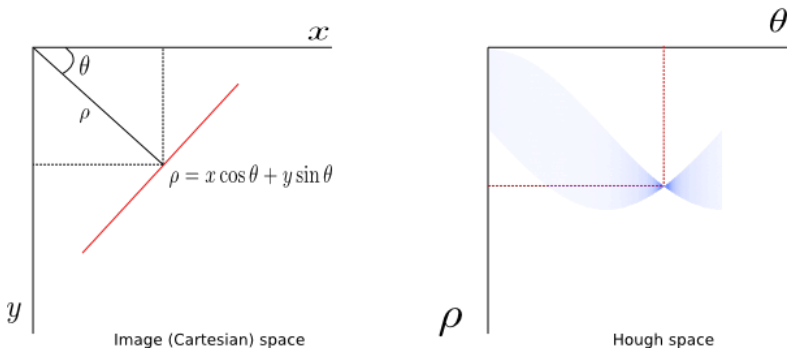
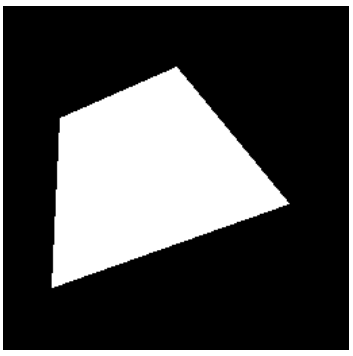
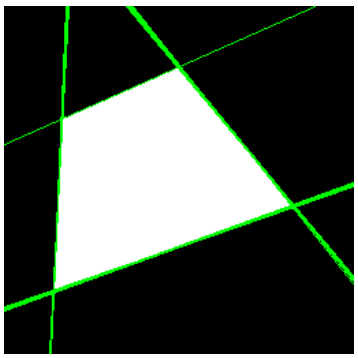
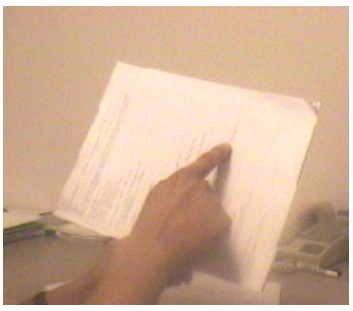
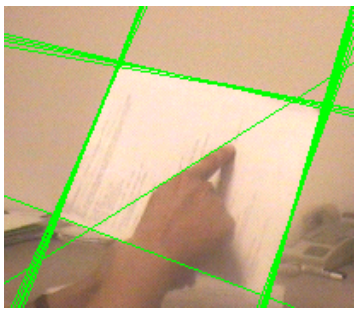


Figure 1: Representation of a line in Cartesian (left) and Hough (right) coordinates

- Here ρ is the distance from line to origin and θ is the angle of the vector of least magnitude to the line
- To obtain these lines to be mapped into points in the Hough space, various methods can be used to identify the lines in an image; in this implementation, the Canny edge detection from MP5 is used
- Once points in Hough space have been detected, a inverse transform is carried out to map the corresponding lines in the image
- Two factors are used as 'resolution' of ρ and θ for quantization of points in the Hough space

Results Analysis:

The following table summarizes the results obtained using two test images:

Image No.	Input Image	Output Image	ρ Factor	θ Factor
1 (test)			1.0	1.0
2 (test2)			0.5	0.5
3 (input)			0.5	0.5

All the input images produced the desired output images. It is possible to make the lines on the output images more optimal by tuning ρ and θ factors as well as the sigma and thresholds for Canny Edge Detection.