Digital Image Processing Lab

Mini Project

Hough Transform implementation

Group Members

Rahul Kumar Meena-18EC35023

SN Ramanathan – 18EC35028

Shreya Kumari-18EC35027

Aim:

To implement Hough Transform from scratch without the help of any prebuilt libraries like OpenCV and using python.

Theory:

Hough Transform:

The Hough transform is a technique that can be used to isolate features of a particular shape within an image. Because it requires that the desired features be specified in some parametric form, the *classical* Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, *etc.* A *generalized* Hough transform can be employed in applications where a simple analytic description of a feature is not possible. Due to the computational complexity of the generalized Hough algorithm, we restrict the main focus of this discussion to the classical Hough transform. Despite its domain restrictions, the classical Hough transform retains many applications, as most manufactured parts contain feature boundaries that can be described by regular curves. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.

Sobel Filter for Edge Detection:

The Sobel operator sometimes called the Sobel–Feldman operator or Sobel filter is used in image processing and computer vision, particularly within edge detection algorithms where it creates an image emphasising edges. It is named after Irwin Sobel and Gary Feldman. Sobel and Feldman presented the idea of an "Isotropic 3 × 3 Image Gradient Operator" at a talk at SAIL in 1968. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel–Feldman operator is either the corresponding gradient vector or the norm of this vector. The Sobel–Feldman operator is based on convolving the image with a small, separable, and integer valued filter in the horizontal and vertical directions and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation that it produces is relatively crude, in particular for high-frequency variations in the image.

Procedure:

1. Edge detection part has already been done in DIP lab experiments and just needed to be translated in python.

2. To do Hough Transform first thing is to create Rho and Thetas values to be used in accumulator from given specification. These values are given through trackbar.

3. Binarization of image is done for better processing since the absolute value of pixel is of no use (because we only need the positional information of the pixel) and binarization is done to decide is certain pixel needs to be transformed.

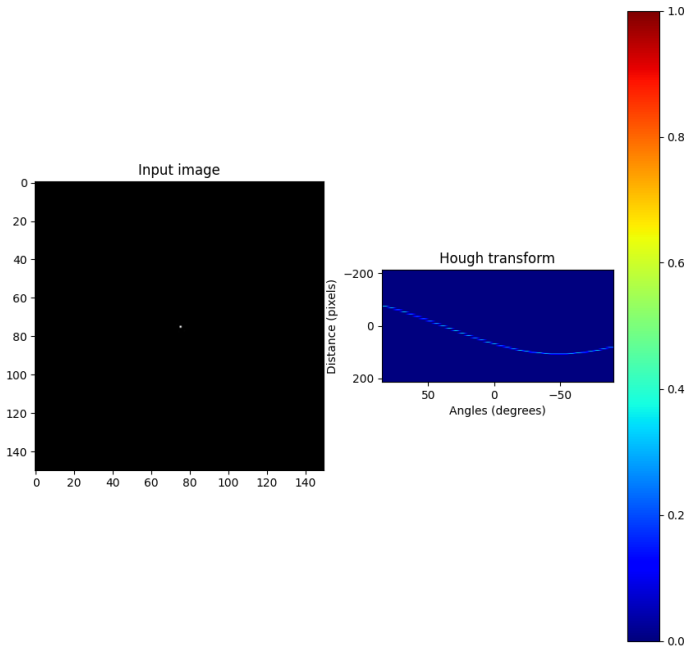
4. Using these Rho and Theta values we iterate over every pixel in binarized image and find value of rho for every theta for that pixel position and increase the count of these rho and theta in accumulator.

5. Hough transform is obtained by plotting accumulator while we can use accumulator values detect line by setting a certain variable threshold. If accumulator value is higher for some particular theta and rho then it forms a line (though not perfect since we approximate using quantization).

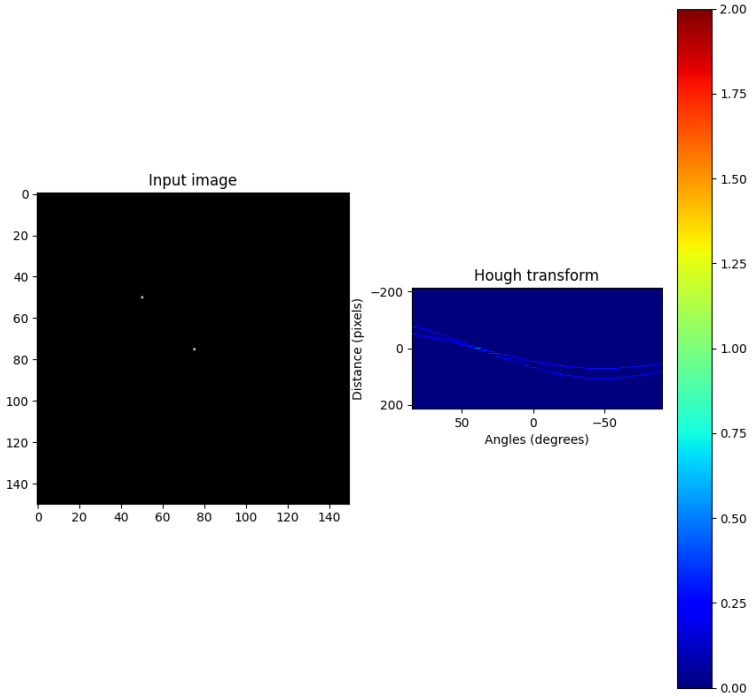
6. To obtain output image we create a blank image and iterate over every pixel in binarized image(used to get accumulator) and get rho and theta and check if the accumulator value is higher than threshold set for that particular rho and theta. If condition is satisfied then we mark that pixel as highest intensity in output image.

Result:

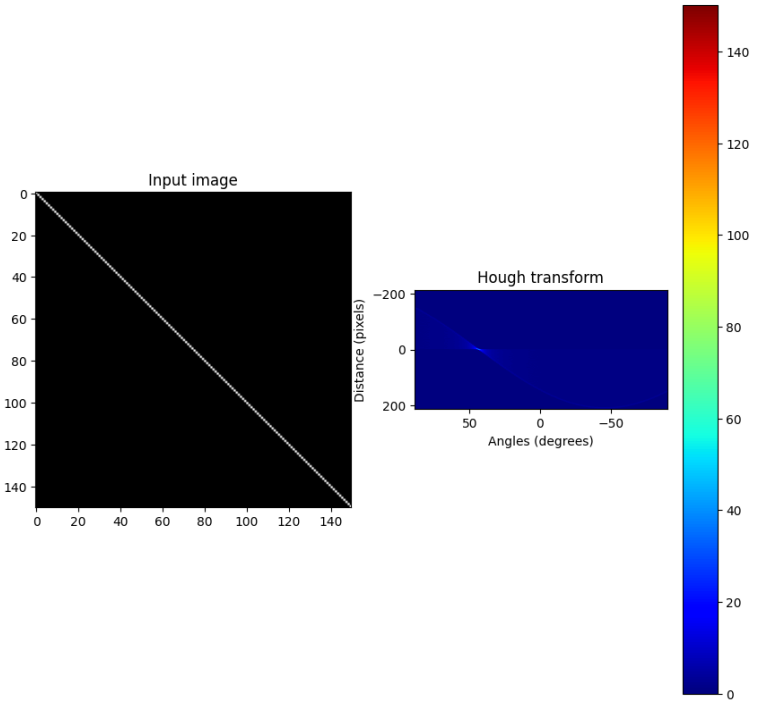
Hough Transform on Basic Images:



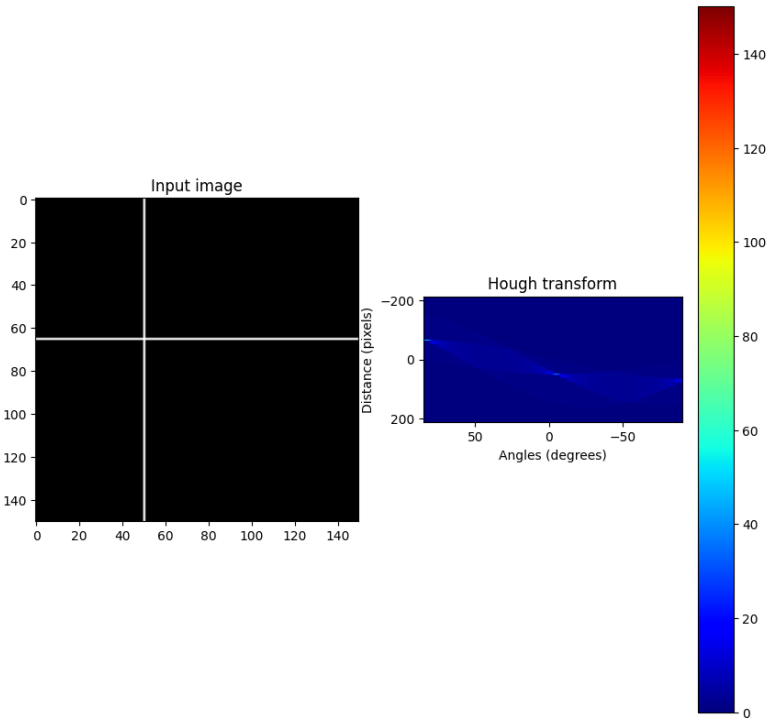
One point in picture plane <-> Sinusoidal curve in parameter plane



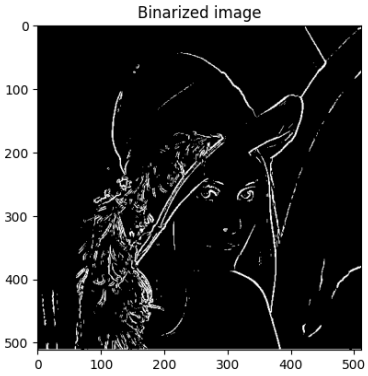
Two point in picture plane <-> Two Sinusoidal curve in parameter plane

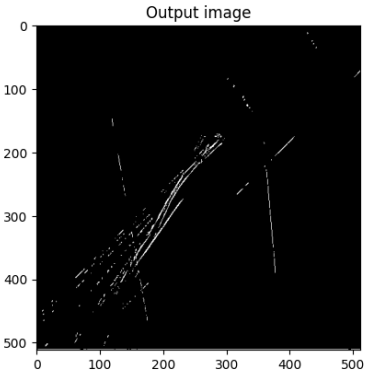


One Line in picture plane <-> Sinusoidal curves through single point in parameter plane

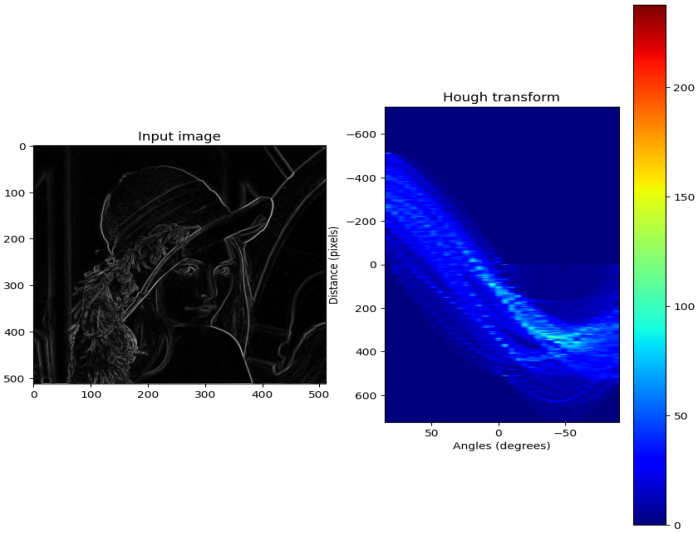


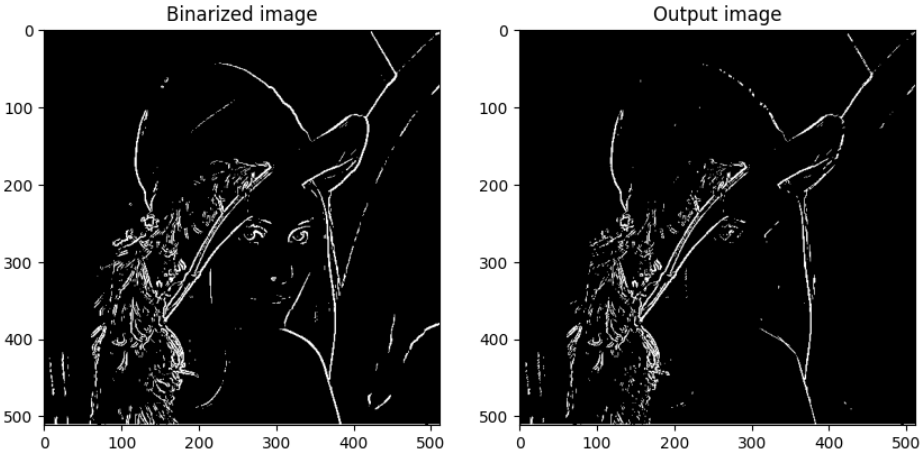
Hough Transform on Normal Images:



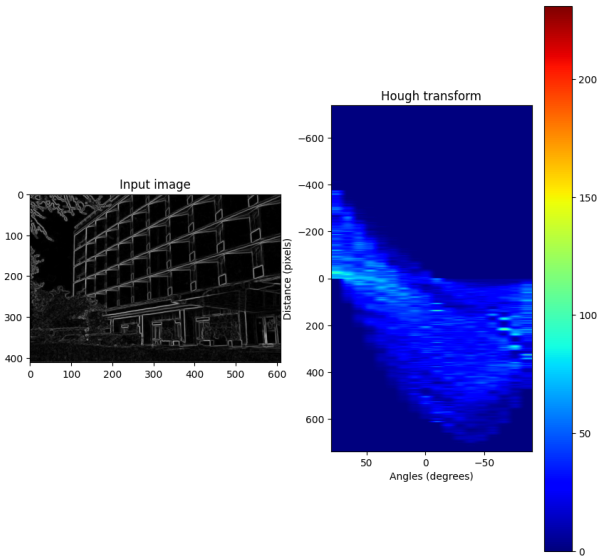


Input Image Binarized Image Final image



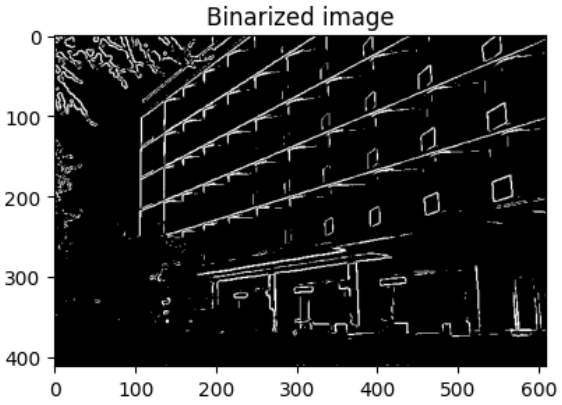


Binarized Image Final image(decreased Accum. threshold)

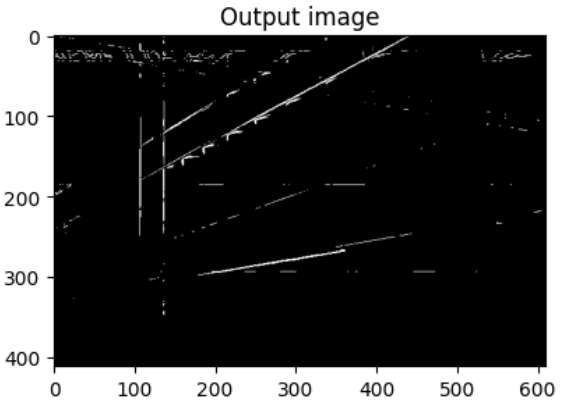


Input Image

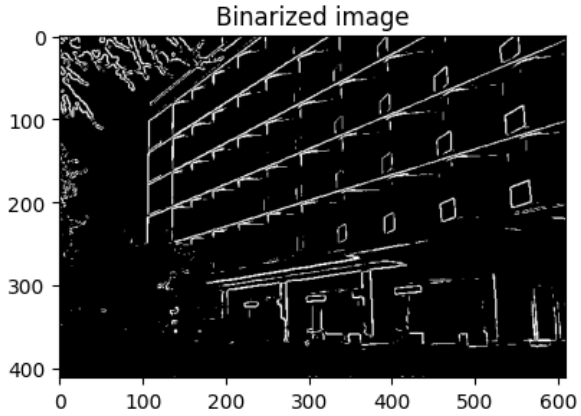
Edge detected Image and Hough Transform



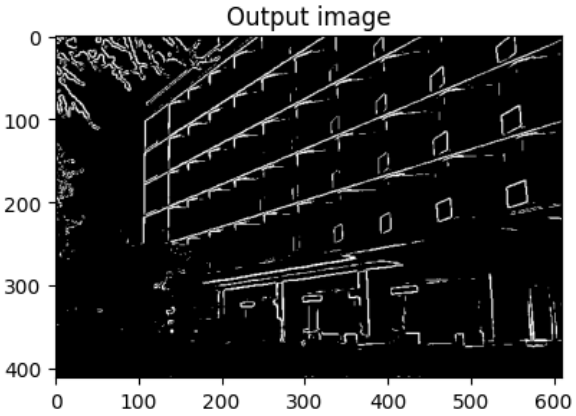
Binarized Image



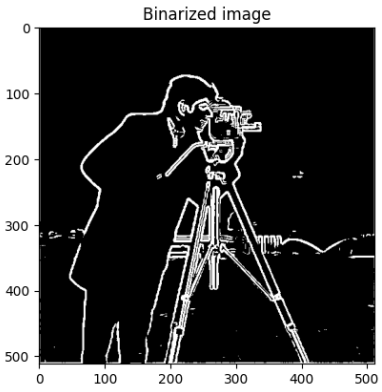
Final image(Acc. Threshold =88)

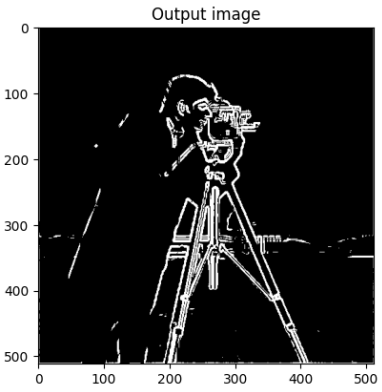


Binarized Image

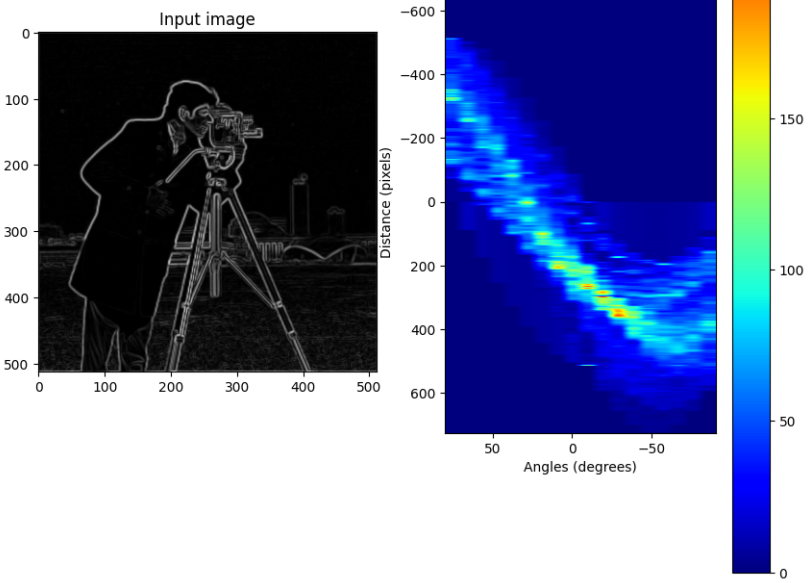


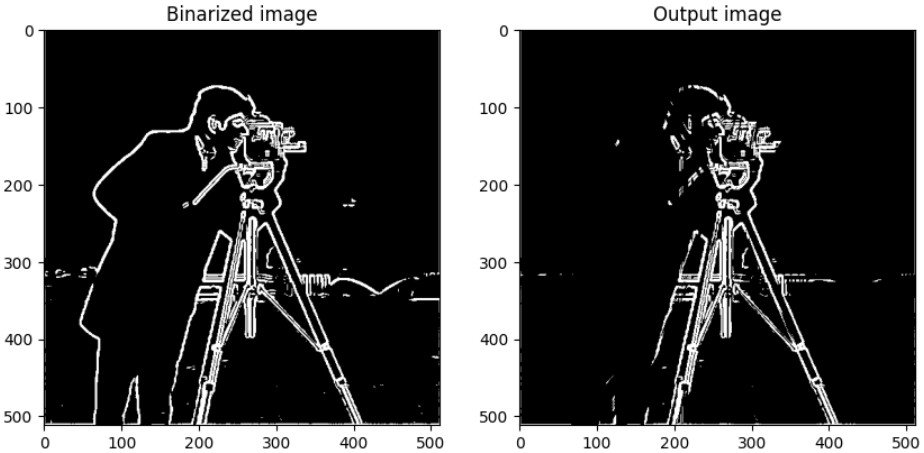
Final image(Acc. Threshold =39)





Input Image





Discussion:

- Hough Tranform has several property such as
    1. A point in the picture plane corresponds to a sinusoidal curve in the parameter plane.
    2. A point in the parameter plane corresponds to a straight line in the picture plane.
    3. Points lying on the same straight line in the picture plane correspond to curves through a common point in the parameter plane.
    4. Points lying on the same curve in the parameter plane correspond to lines through the same point in the picture plane.

- These properties can easily be checked for implementation in hough transform for basic images in results above. For example a dot image produces a sinusoidal curve etc.

- Rho (distance parameter) values are quantized at interval of one. While theta quantization can be varied through trackbar 3. A smaller values of angle step gives higher number of total angles and vice versa.

- A smaller value of Angle step gives better result but results in higher computation. Also edge detection part if done with filter size of 9x9 takes too much time. Computation time can also be reduced by increasing image threshold(trackbar1) but may compromise the result.

- Results for lena image and building image are both shown with sobel diagonal filter of kernel size 3x3. While results for cameraman are with kernel size of 5x5. Using a bigger filter size than this produces image with blurry edges. Therefore smaller kernel size is preferred.