Hough Transform

(a) Hough transform of the image.Using hough image.png

The Hough transform1 is designed to find lines in images, but it can be easily varied to find other shapes.Suppose (x,y) is a point in the image (which we shall assume to be binary). We can write y=ax+b, and consider all pairs (a,b) which satisfy this equation, and plot them into an “accumulator array”. The(a,b) array is the “transform array”.

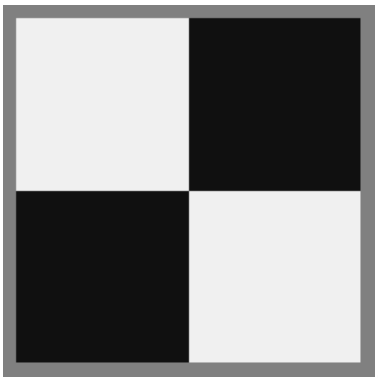

Fig:Image

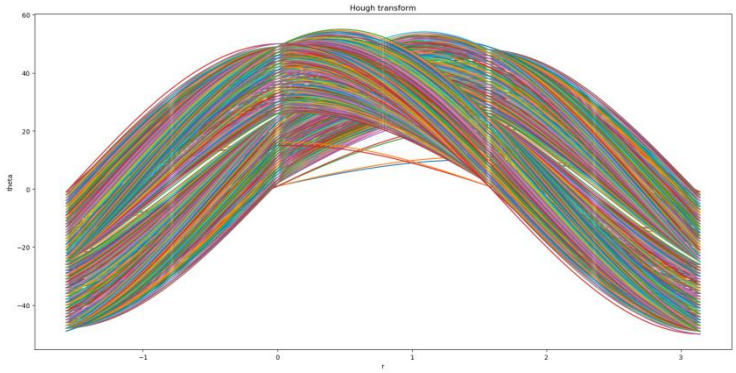

Fig: Hough transformed image

Inference:

- The second image is the Hough transformed of the given image (first).
- It shows all the lines corresponding to the all the points in the image.
- As seen all lines passes from the intersection point.

(b) Now we're going to add some noise. Use Hough image, but with noise. The goal is to adjust the filtering, edge finding, and Hough algorithms to and the lines.

Steps:

- Reading the Hough noise image

- Applying the Gaussian filter to reduce the noise. For better filtering the kernel size of the filter used is of 9x9
- Thresh holding is done to convert the image into binary image, values less than 80 are 0 and greater are 1.
- Canny edge detection is applied for detecting the edges.
- Resizing of the image Is done(50X50).
- Adjusting is done for the edges.
- Mapping of edge points to the Hough space and storage in an accumulator
- Interpretation of the accumulator to yield lines of infinite length. The interpretation is done by thresholding and possibly other constraints.
- Conversion of infinite lines to finite lines.
- R and theta is calculated by the formula.

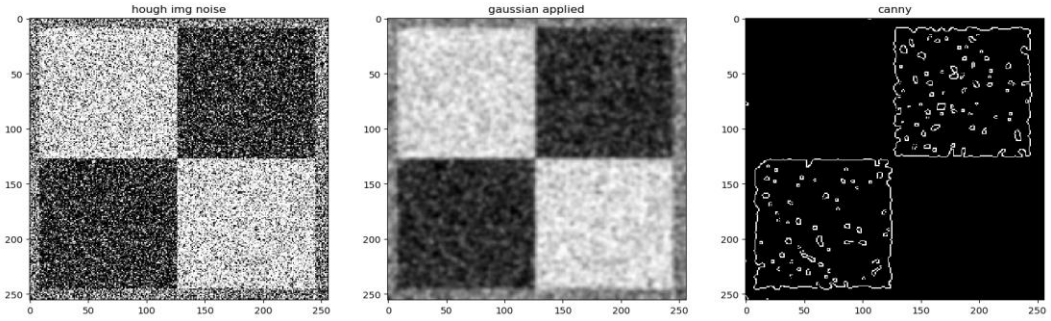

Fig: left-Hough noise image, centre-blurring done using Gaussian filter, right-image after canny edge detection

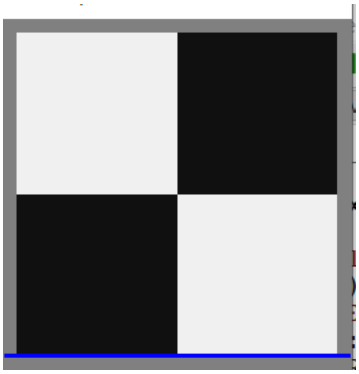

Fig: Prominent line

Inferences:

- Taking greater kernel size help to remove noise better.
- There are many values of r which are highest(same)
- Taking different values gives different prominent lines.