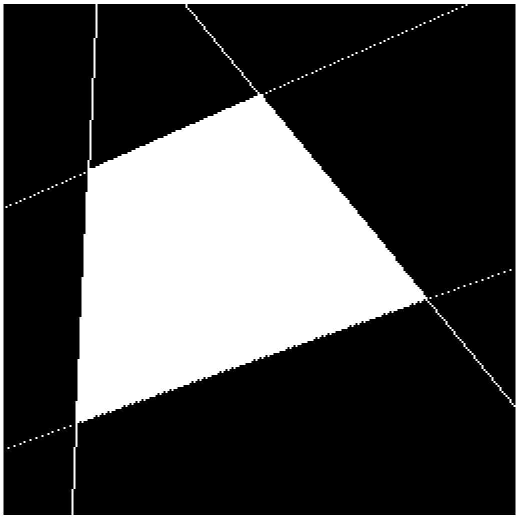
MP6 Hough Transform

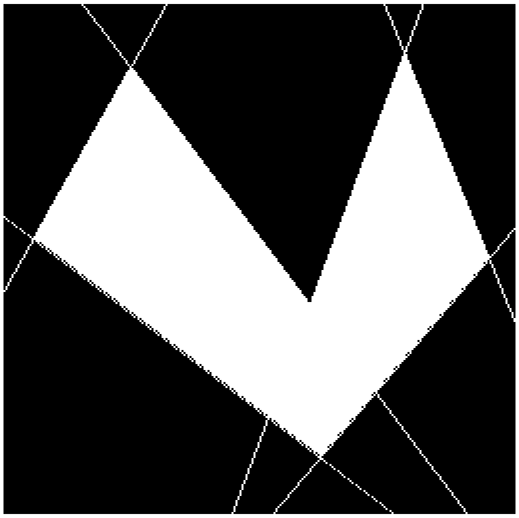
In this homework, a Hough Transform is implemented to detect the line inside an image along with edge detection.

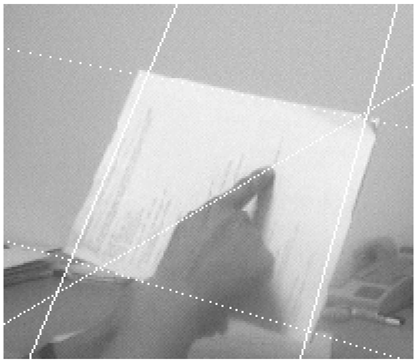
Hough Transform

The Hough Transform function takes the processed edge image that comes from the edge detection function (Canny Edge detection in this case) and detects possible linear edges. First, the function collects all edge pixel position information to estimate the possible lines which pass through the point. All possible lines are expressed by functions in polar coordinates. Then a voting map (parameter space) is created. The parameter theta and rho which are used to express the line in polar coordinates are the axes of the voting map. One point will be represented by a sinusoid curve in the voting space. When all points are expressed in the voting space, the intercept point will be the possible line parameter that can express the linear edges. It is hard to extract the intercept point directly from the voting map, but normally the intercept point will get high votes. A vote filter is used to collect all possible intercept point coordinates and remove all points with low votes. This will generate several clusters of points. Then a K mean clustering algorithm is used to find the center of each cluster and provide the most likely intercept points. The intercept points will provide the parameter of the line which can be used to generate the lines in the image.

Results:



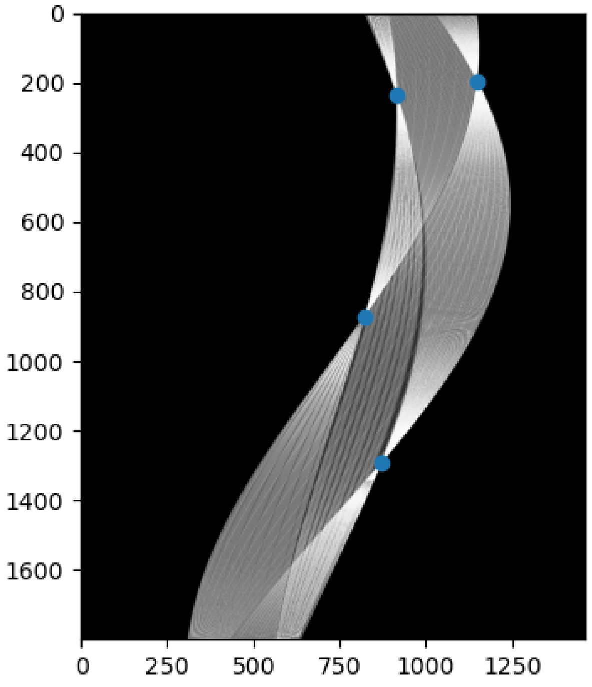




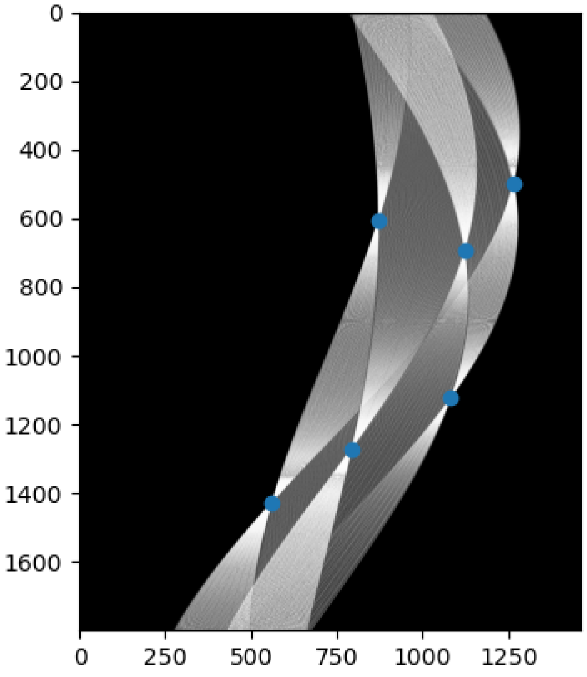
Results should have lines inside to indicate the straight lines in the image. According to all the results, we can see the edge detector can detect most edges in the image.

detect significant intersections in the parameter space:
(all images are processed by Histogram Equalization function to increase the contract)

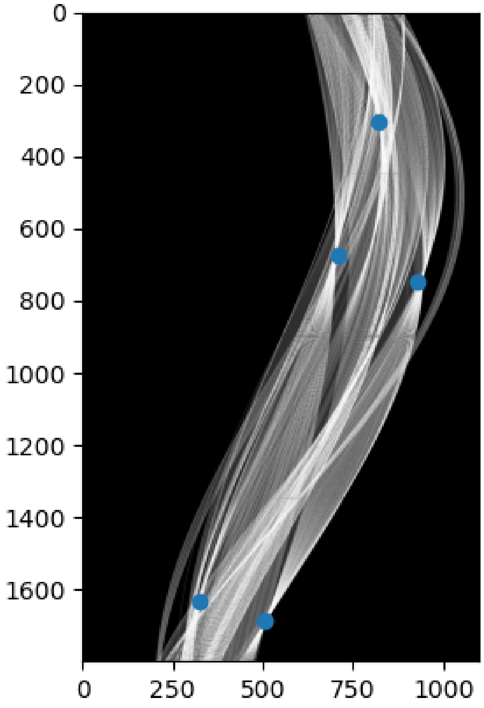
Test parameter space:



Test2 parameter space:



Input parameter space:

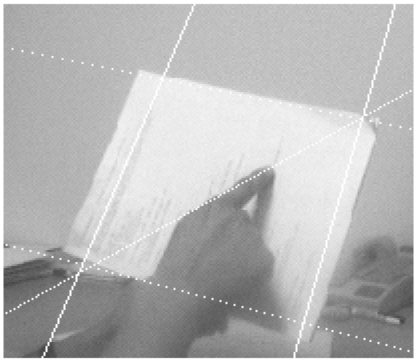


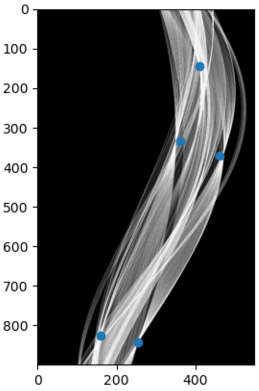
comparison of different quantization in the parameter space:

In this test, image “input” is used for the input image. Three different size of spaces are selected: 900 x sqrt(r^2+c^2), 1800 x sqrt(r^2+c^2)*2, 4500 x sqrt(r^2+c^2)*5

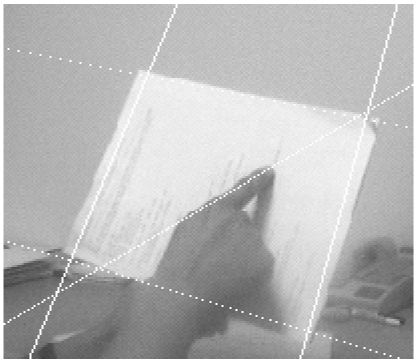
The Voting space and final results show below:

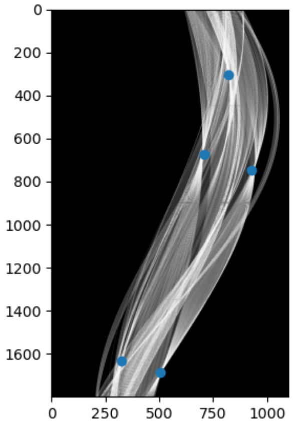
900 x sqrt(r^2+c^2):



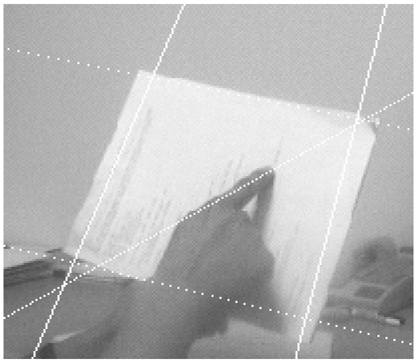


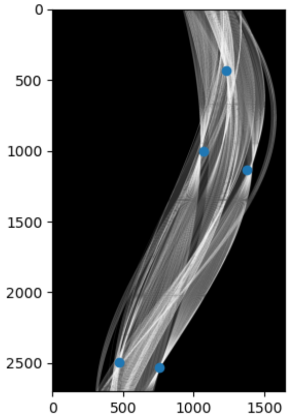
1800 x sqrt(r^2+c^2)*2:





4500 x sqrt(r^2+c^2)*5





Even though the result looks similar, if the parameter space is too big or too small, the function will have very low performance. If the space is too big, all the votes will have their own point to represent, which will be hard for the filter to remove non intercept points. If the space is too small, a lot of the votes will be on the same point, which will reduce the resolution of the result.