Xichen Liu

CS415 MP3

Line detection by Hough transform

First, here is my idea of the algorithm. Hence, to do the line detection by Hough transform, the input should be an edge detected image. For the part, I used the build-in functions, such as GaussianBlur(), and Canny().

As for the first step of the algorithm, I computed the input image's length and width, and so to calculate the maximum rho value. Then I set the interval of rho axis and theta axis. My understanding to quantization level is modify the interval of the axis in Hough space. And then I established an accumulator and initialized to all zeros.

Then the voting part. For every point in the input image, check if the value of that pixel is zero or not. Because the input image is an edge detected image, which means if the value of that pixel is 0, it is the background. If it is zero, pass; if not, for every theta value, which means every direction around the current-processing point, vote from every point on the line in the Hough transform.

After that, I extract the edges that greater than the threshold I set. As I tried many times with different value of the threshold, I found 45 might be a good one. For every curve I got for Hough space, the rho and theta are known, so it is not hard to find the corresponding line in normal space. Draw it then.

Here we got the lines detected from the edge detected image.

I tried to modify the sigma of Gaussian filter, and the intervals of rho and theta axis in Hough space. Here are the results.

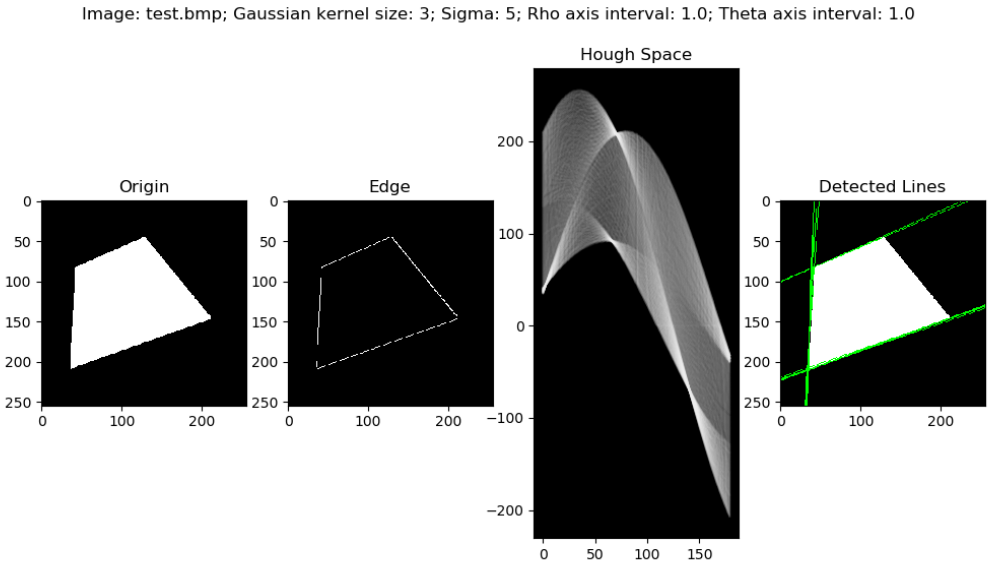


Image: test.bmp; Gaussian kernel size: 3; Sigma: 5; Rho axis interval: 2.0; Theta axis interval: 2.0

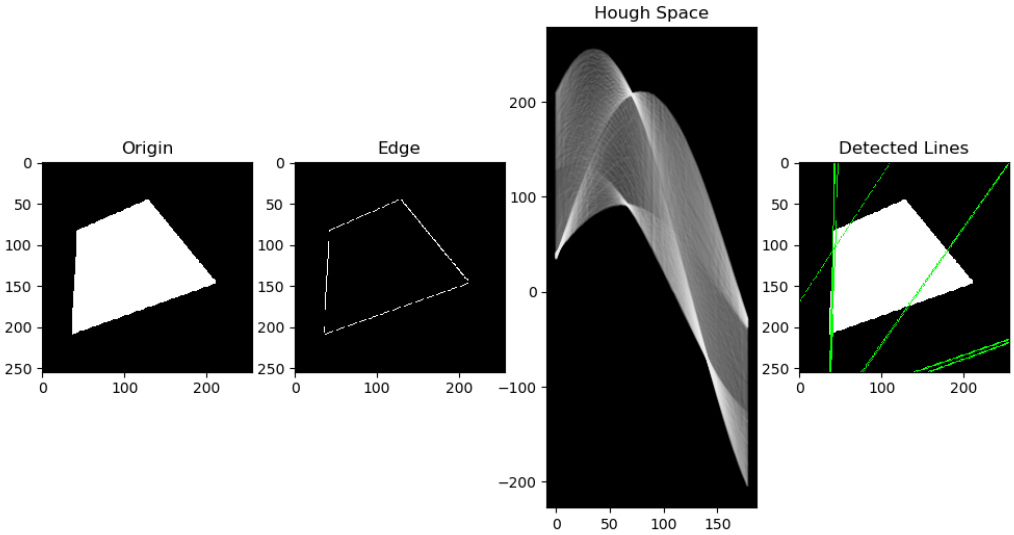


Image: test.bmp; Gaussian kernel size: 3; Sigma: 10; Rho axis interval: 1.0; Theta axis interval: 1.0

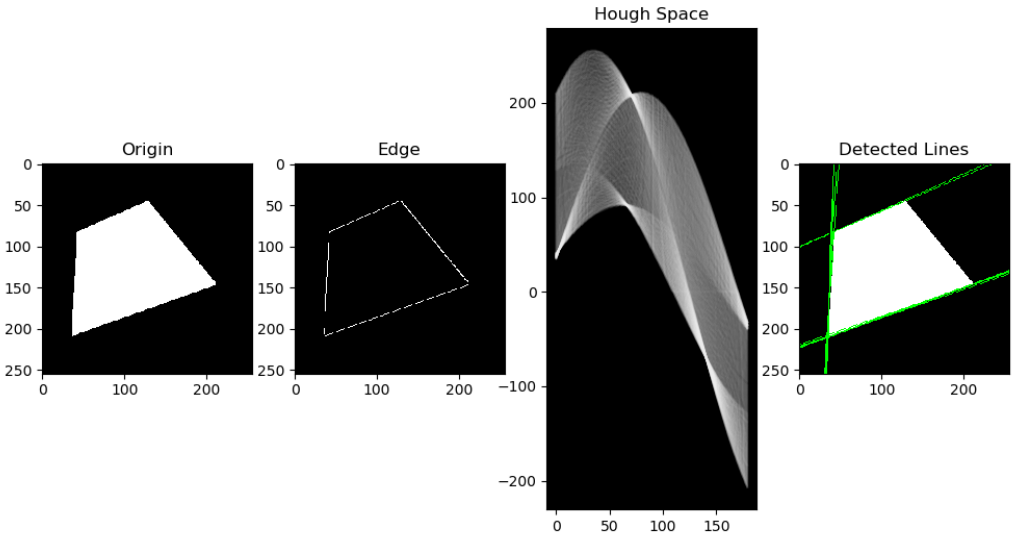


Image: test.bmp; Gaussian kernel size: 3; Sigma: 10; Rho axis interval: 2.0; Theta axis interval: 2.0

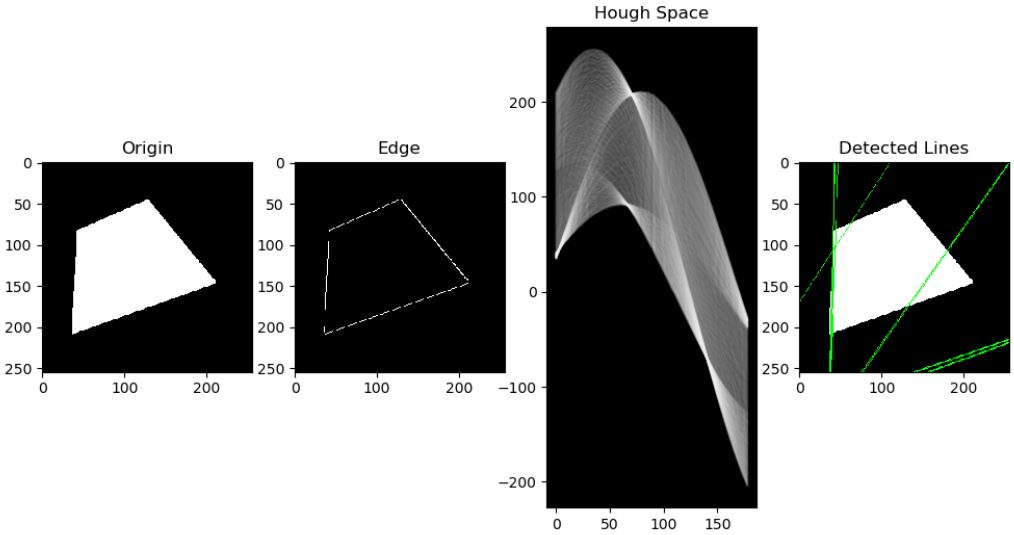


Image: test2.bmp; Gaussian kernel size: 3; Sigma: 5; Rho axis interval: 1.0; Theta axis interval: 1.0

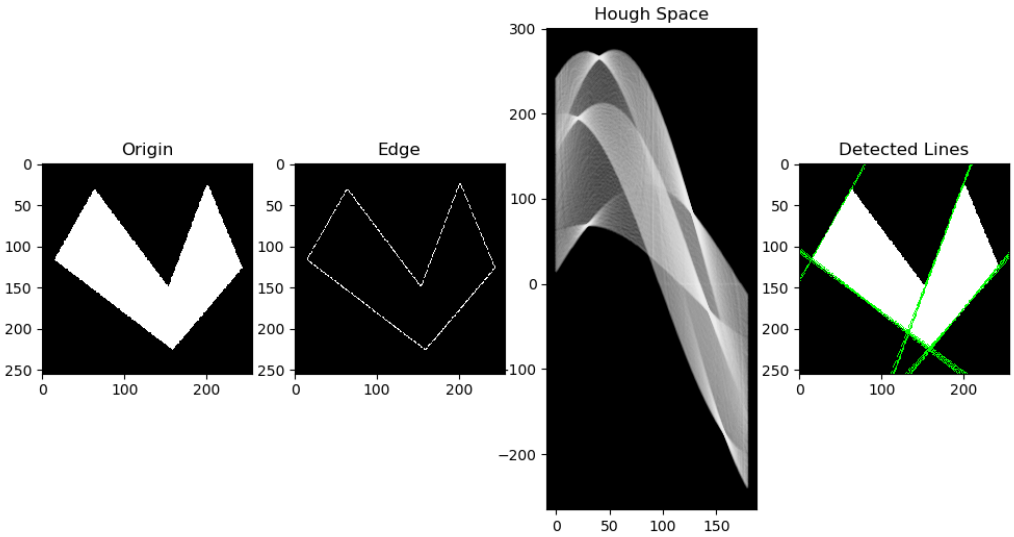


Image: test2.bmp; Gaussian kernel size: 3; Sigma: 5; Rho axis interval: 2.0; Theta axis interval: 2.0

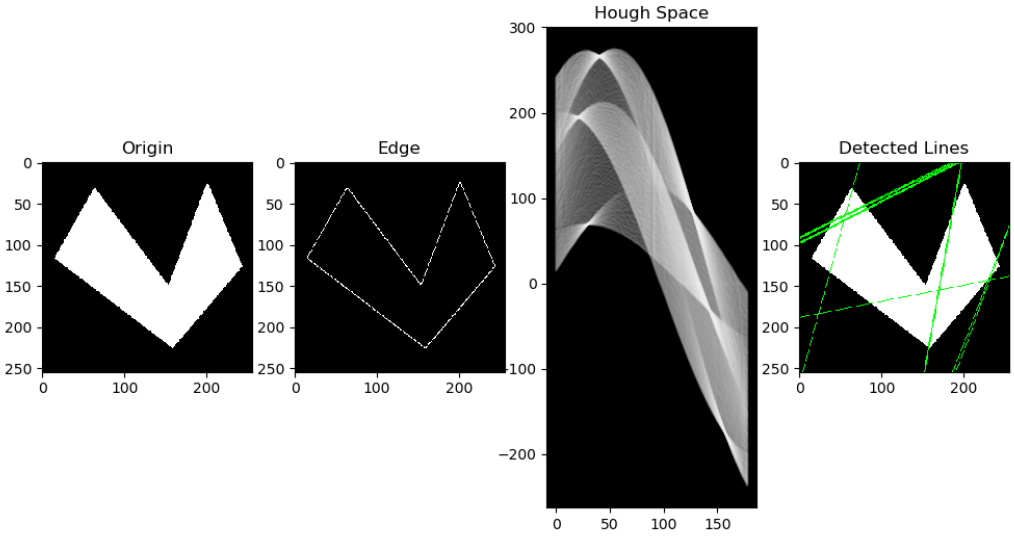


Image: test2.bmp; Gaussian kernel size: 3; Sigma: 10; Rho axis interval: 1.0; Theta axis interval: 1.0

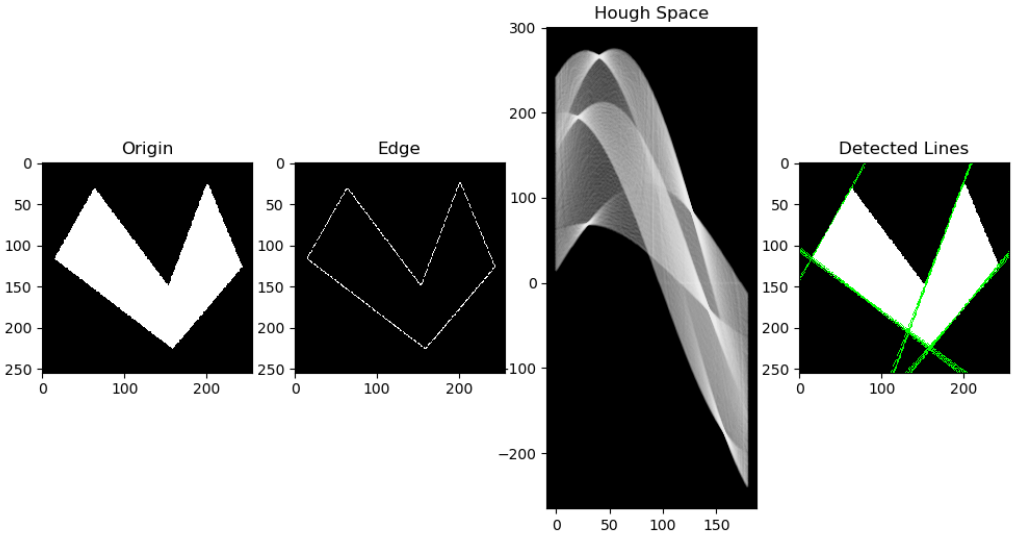


Image: test2.bmp; Gaussian kernel size: 3; Sigma: 10; Rho axis interval: 2.0; Theta axis interval: 2.0

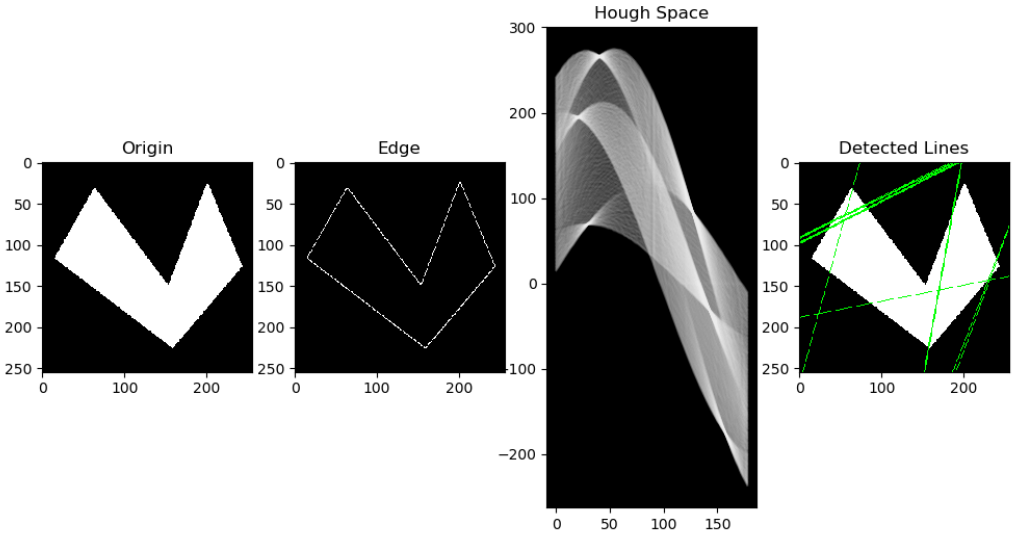


Image: input.bmp; Gaussian kernel size: 3; Sigma: 5; Rho axis interval: 0.5; Theta axis interval: 0.5

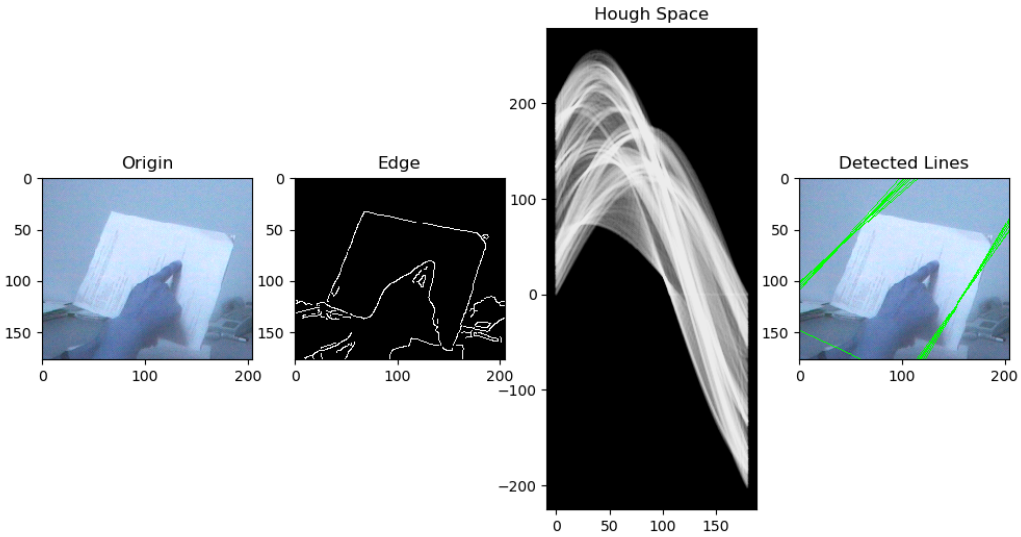


Image: input.bmp; Gaussian kernel size: 3; Sigma: 5; Rho axis interval: 1.0; Theta axis interval: 1.0

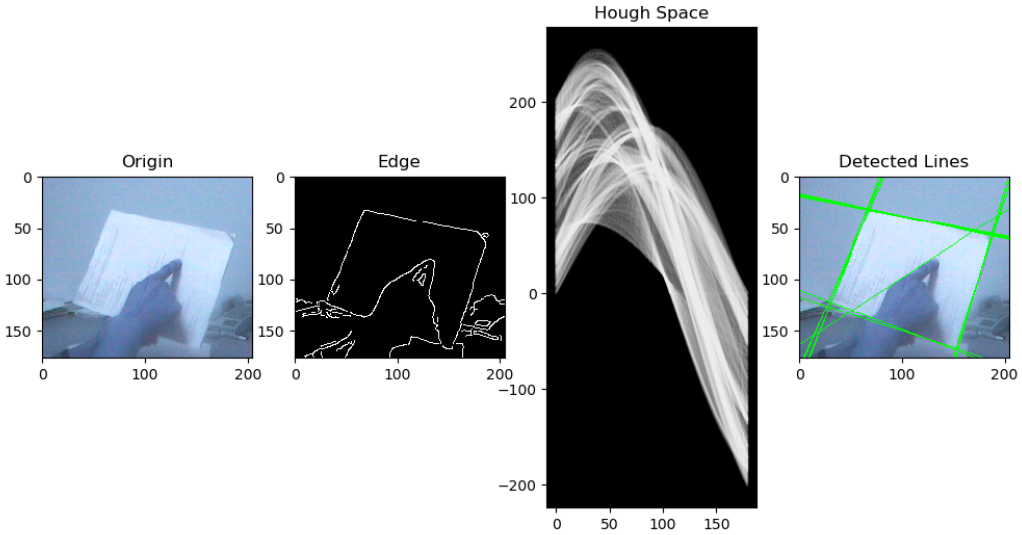


Image: input.bmp; Gaussian kernel size: 3; Sigma: 5; Rho axis interval: 2.0; Theta axis interval: 2.0

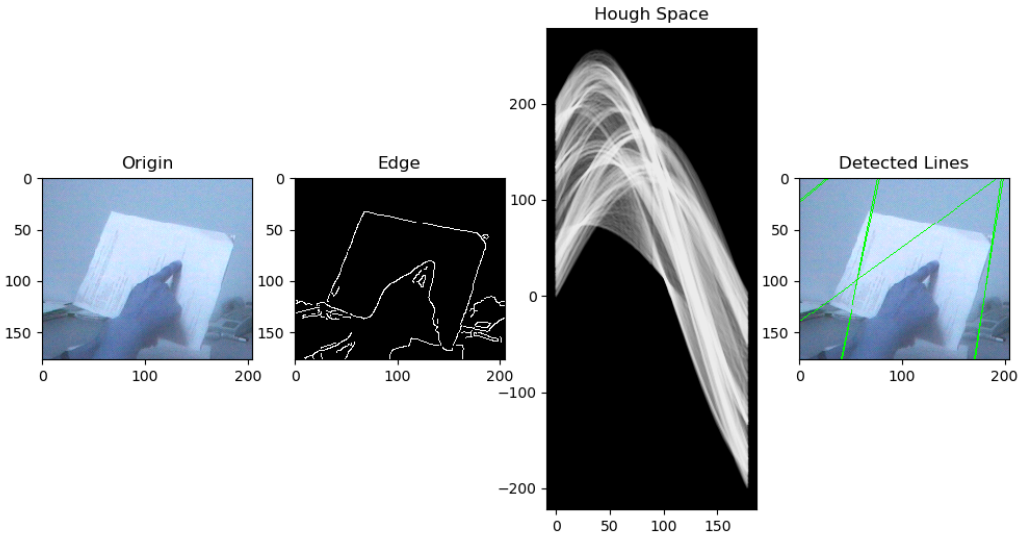


Image: input.bmp; Gaussian kernel size: 3; Sigma: 10; Rho axis interval: 0.5; Theta axis interval: 0.5

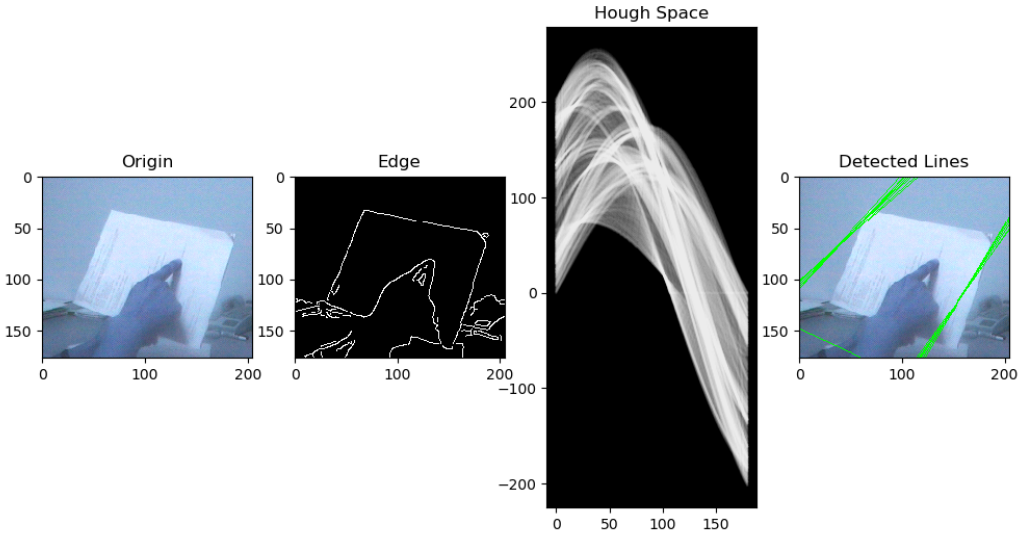
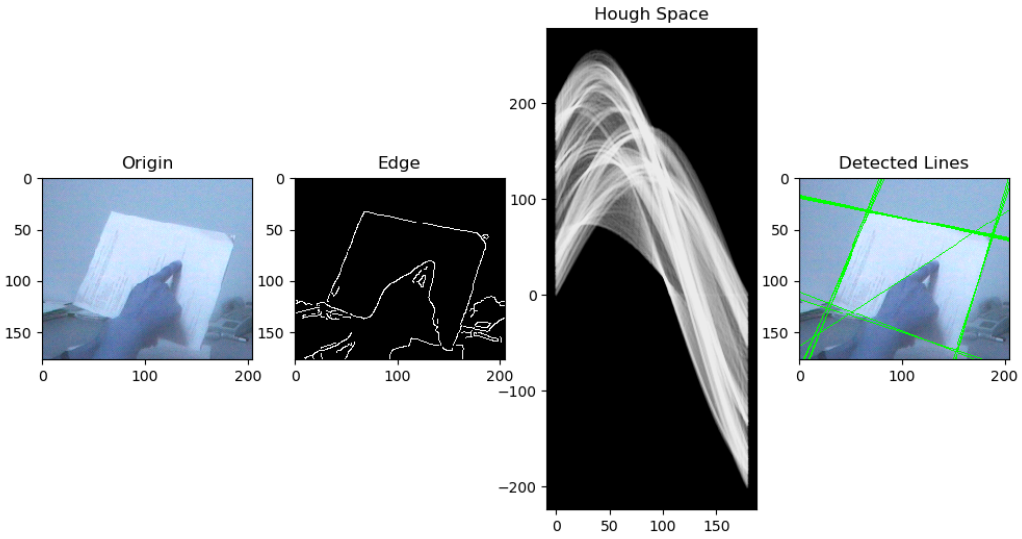
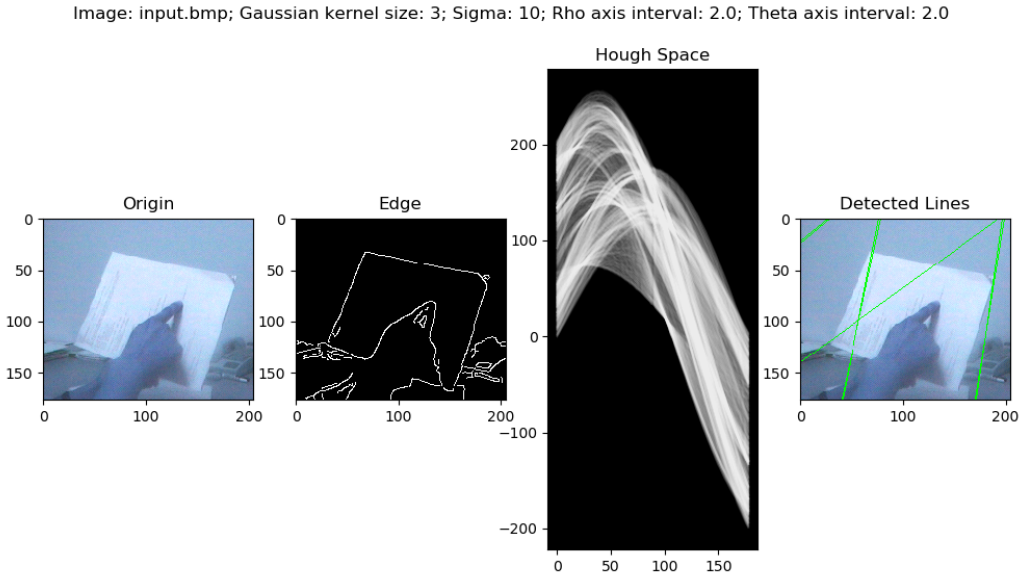


Image: input.bmp; Gaussian kernel size: 3; Sigma: 10; Rho axis interval: 1.0; Theta axis interval: 1.0





Analysis: Since I modified the sigma of Gaussian filter, and the intervals of rho and theta axis in Hough space, I found that the changes of sigma that influence the output image not that significant. I think the reason is that the sigma affects the image after GaussianBlur(), but the given images' edges are already significant enough to detect the edges. And after all, the input image is the edge detected image, so the change to sigma will not cause a significant change to the line detected image.

Secondly, I tried different intervals of rho and theta axis in Hough space to find if the density of the intervals in Hough space affects somehow. As the result shows, the ones which always set the value 1 to the interval in Hough space perform better. Other than these two variables, I also have tried different thresholds for Canny(), and different thresholds for detecting edges from accumulator. As I predicted, the edge detected image will have more edges left and will cause more lines detected in the final output, while the thresholds changed for Canny(). And different thresholds preform the same.