Görkem Yılmaz

21601927

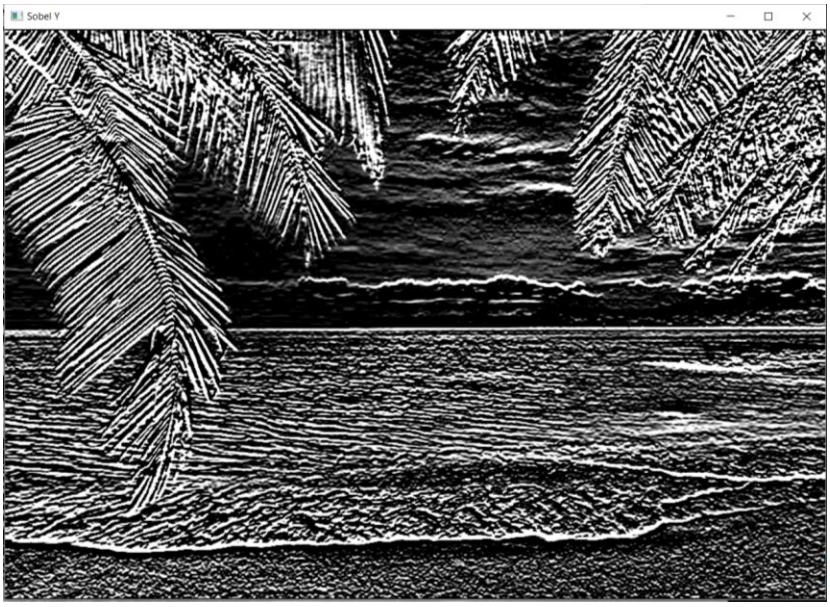
Introduction to Computer Vision

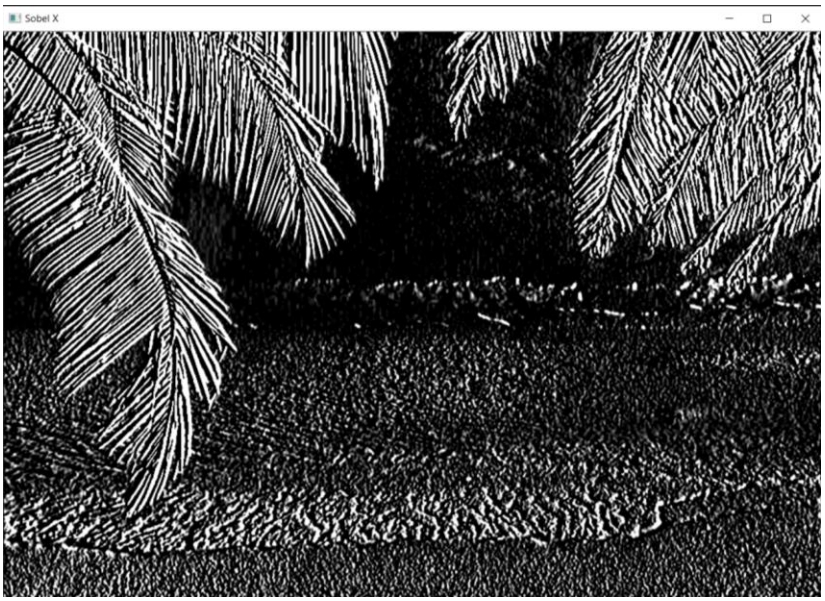
Homework 3 Report

Part 1)

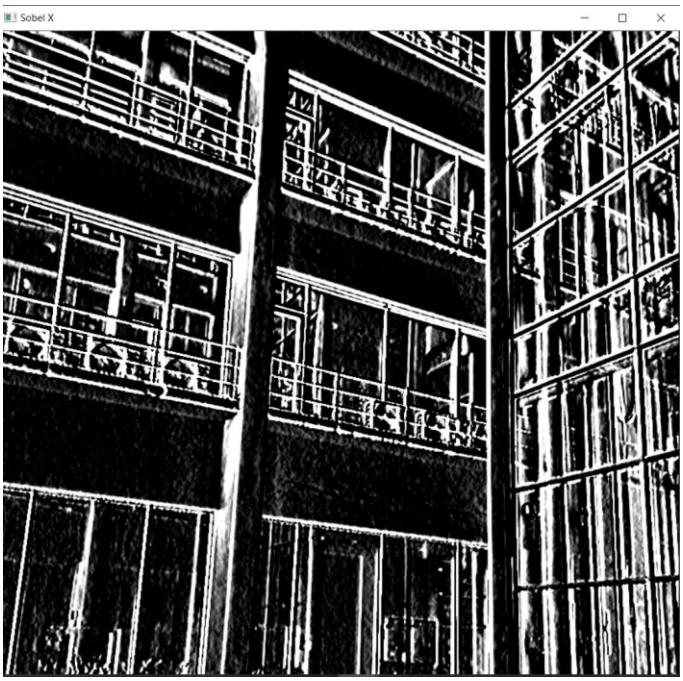
1.1.1) Sobel Edge Detection

For edge1.png,

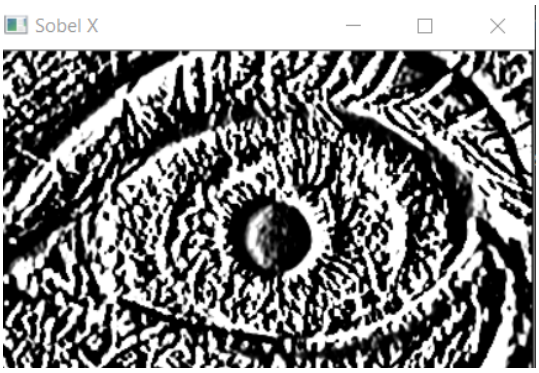




For edge2.png,



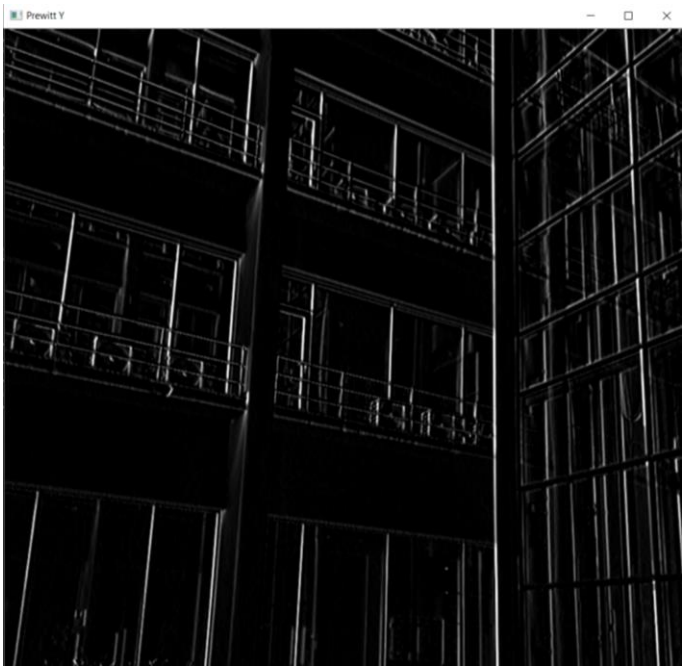
For edge3.png,



1.1.2) Prewitt Edge Detection

For edge1.png,

For edge2.png,



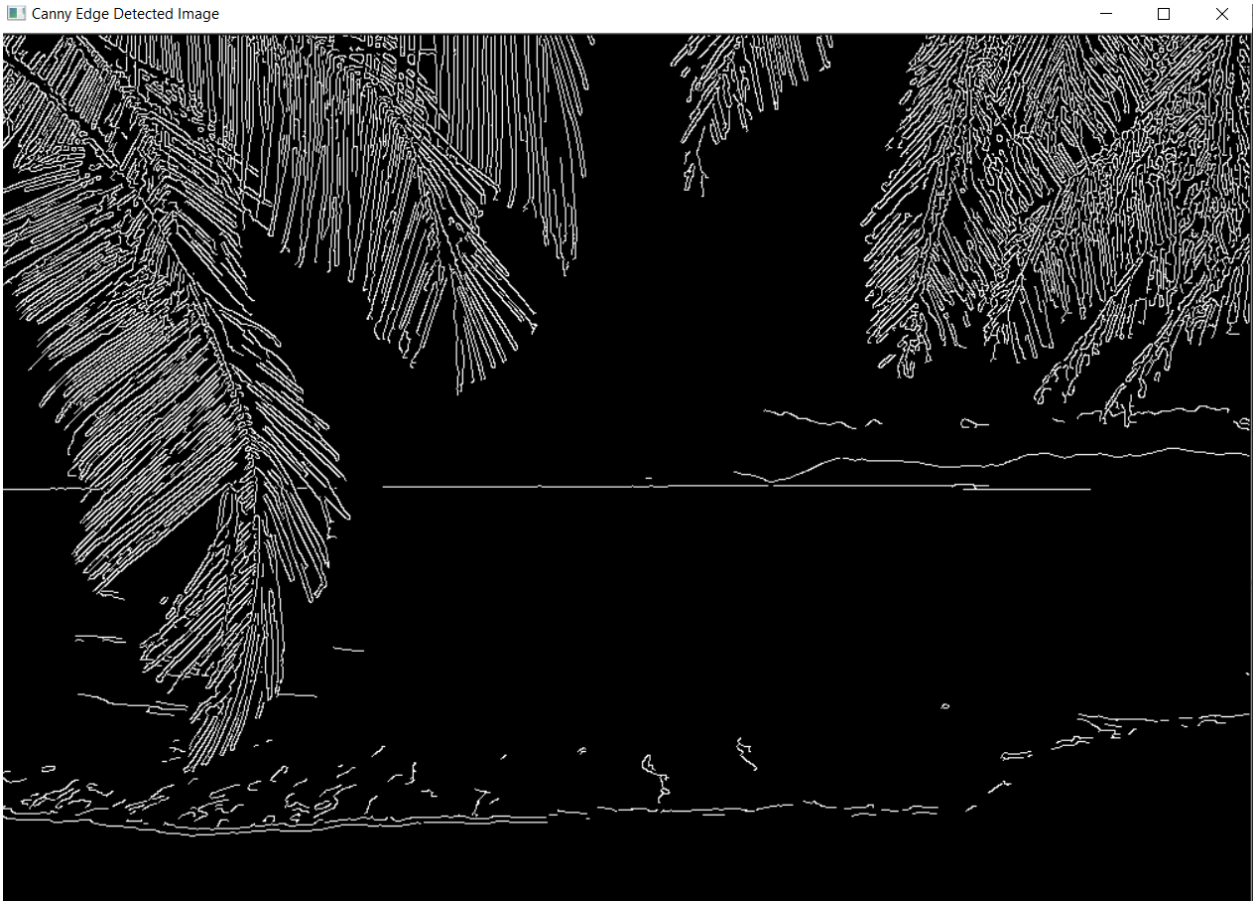
For edge3.png,

1.1.3) As a result, I have observed that with both Sobel and Prewitt Edge Detection the main edges are detected successfully. As for the differences that I have seen, first of all, result of using a horizontal filter is different from using a vertical filter. If we use a horizontal filter, horizontal lines are more likely to be precisely detected and the opposite is valid for vertical filter.

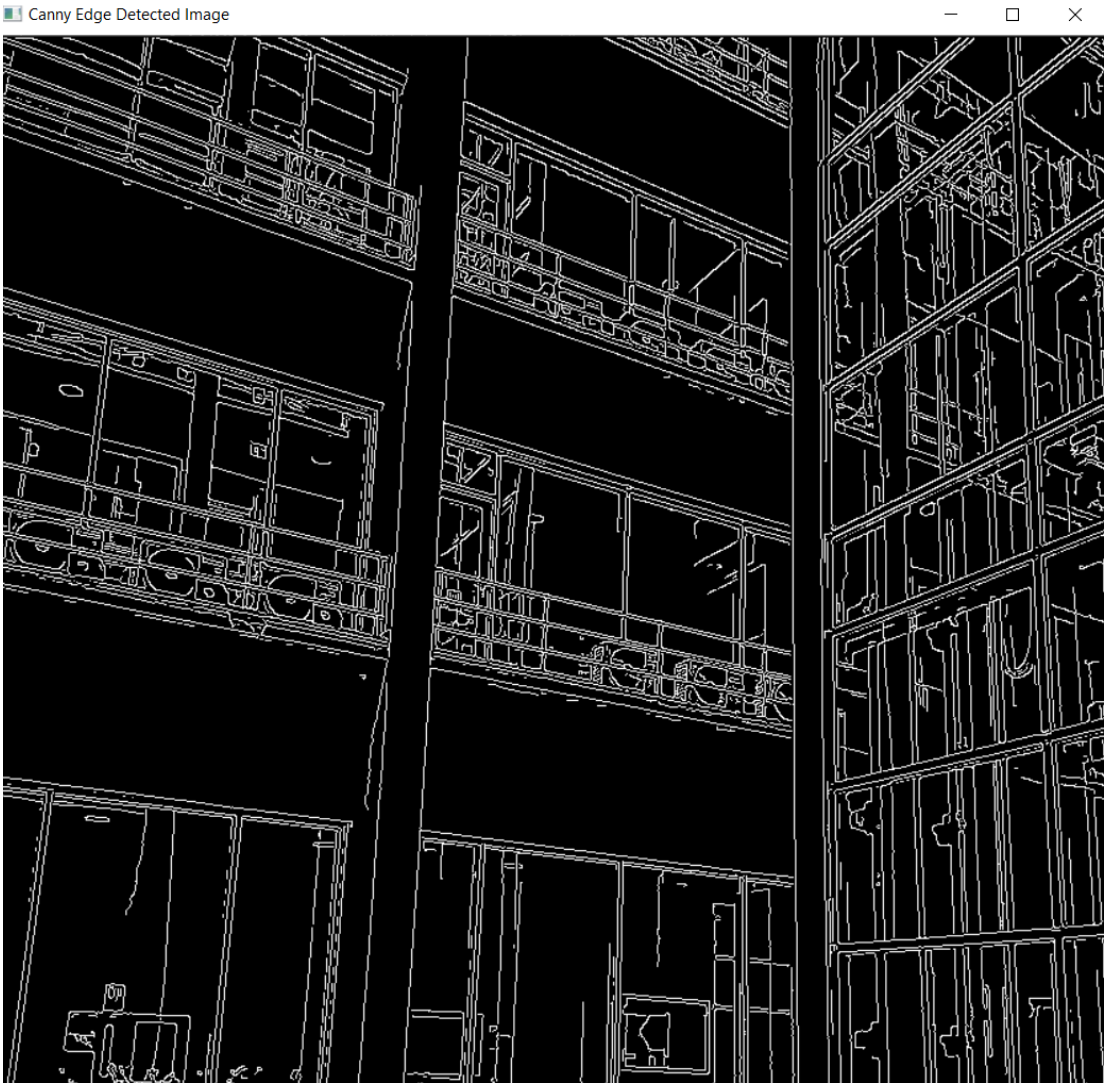
The difference between Sobel Edge Detection and Prewitt Edge Detection is that the filters they are using. That results in having a noisier detection with Sobel Edge Detection (almost every small edge is detected) but with Prewitt Edge Detection, only the main edges are detected and they are clearer compared to Sobel results. Also, a little research about this subject reveal that in sobel operator the coefficients of masks are not fixed and they can be adjusted according to our requirement unless they do not violate any property of derivative masks.

1.2.1) In Canny Edge Detection, I've observed that the level of detail increases if we give a smaller sigma value as a parameter.

For the first image, I used 0.05 as a low threshold value and changing that value didn't make any significant changes. I used 1 as sigma value because l figured it gives enough detail about the edges. For the high threshold, I used 40 because when I give a smaller value, background (sky and sea) is also detected and it gives a noisy result. When I used 40, only the leaves, horizon and sea line were detected and it gave a more precise result. Here is my result:



For the second image, I used 20 as the high threshold to reduce the level of detail but when I did that, some edges that should be detected disappeared. Therefore, I decrease the sigma value in order to gain those details back. I used 0.2 as sigma and I did not change the low threshold. Here is the resulting image:

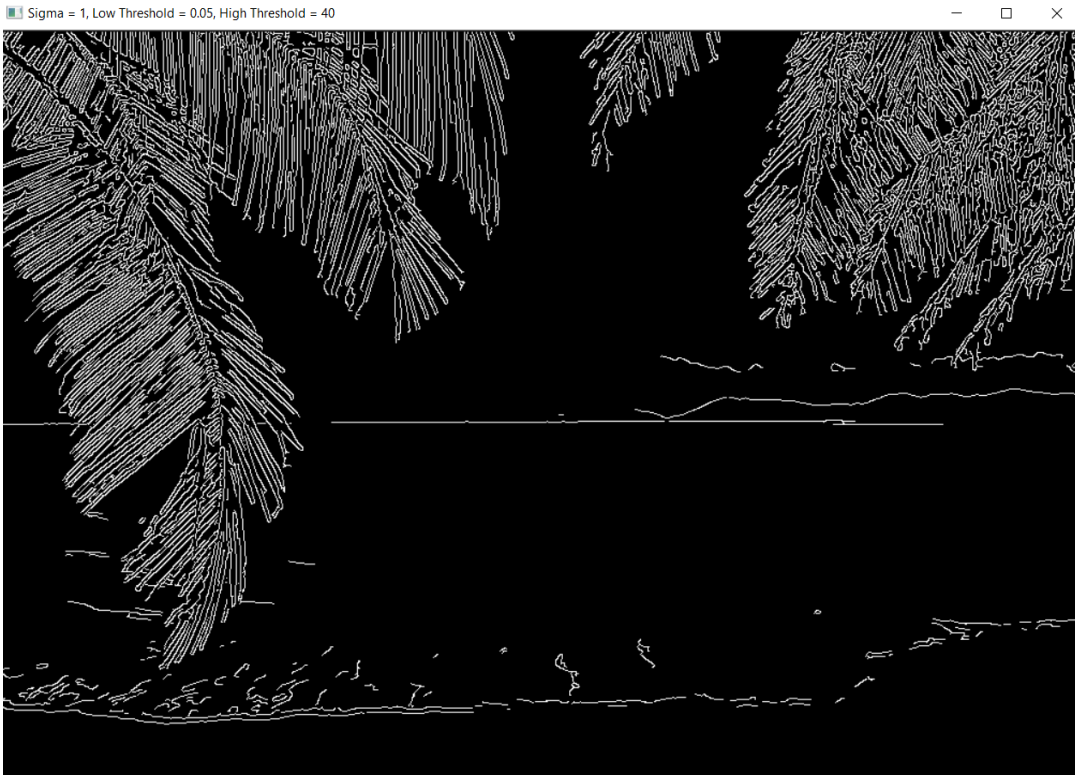


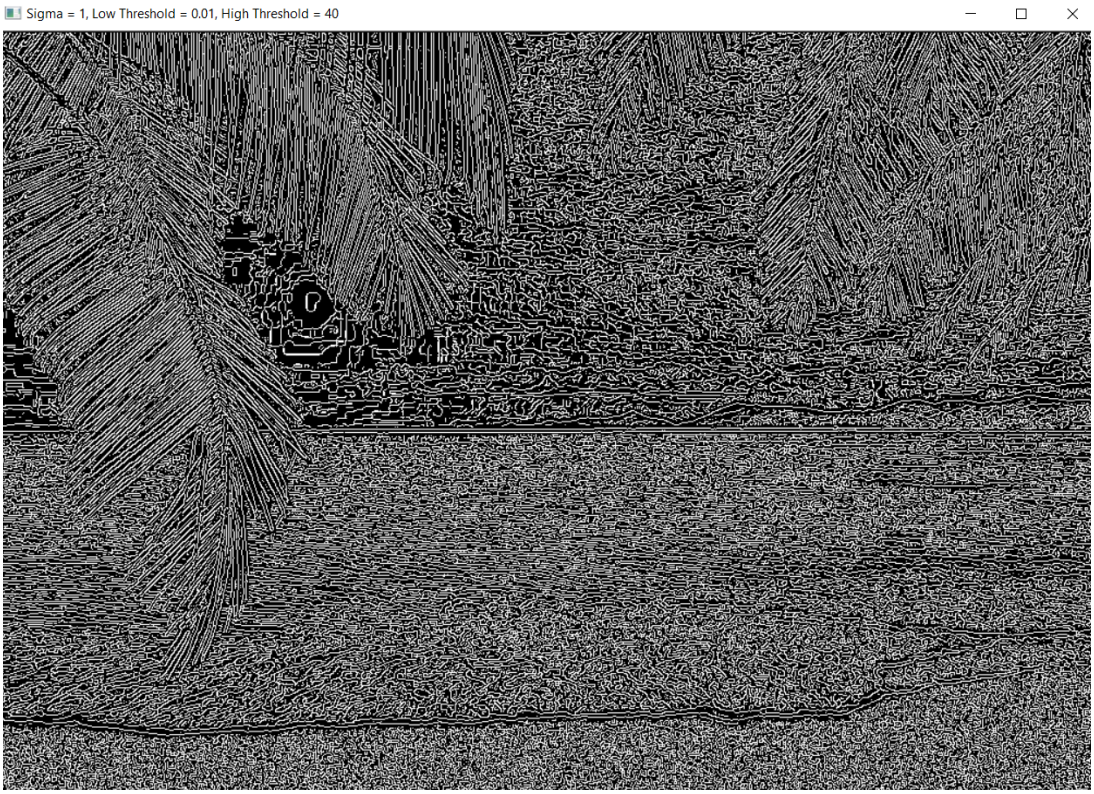
For the last image, I used 2 as sigma, 0.005 for low threshold and 30 as the high threshold. When I used a higher value for high threshold, I discovered that the edges start to disappear from the left side of image. So, I thought that 30 is the optimal value for high threshold. Here is the result I got:

1.2.2)

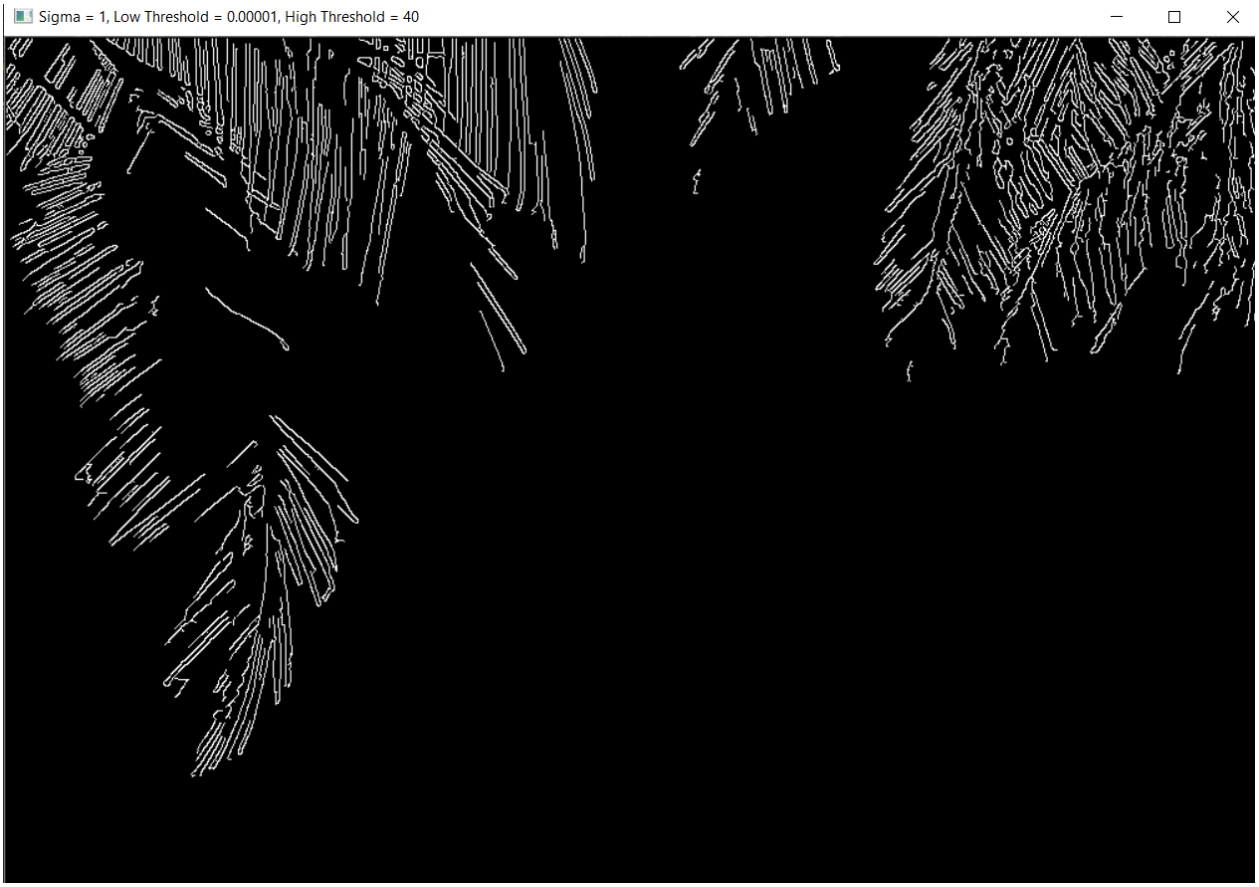
Parameters are listed in the image titles.

4 different result for edge1.png:

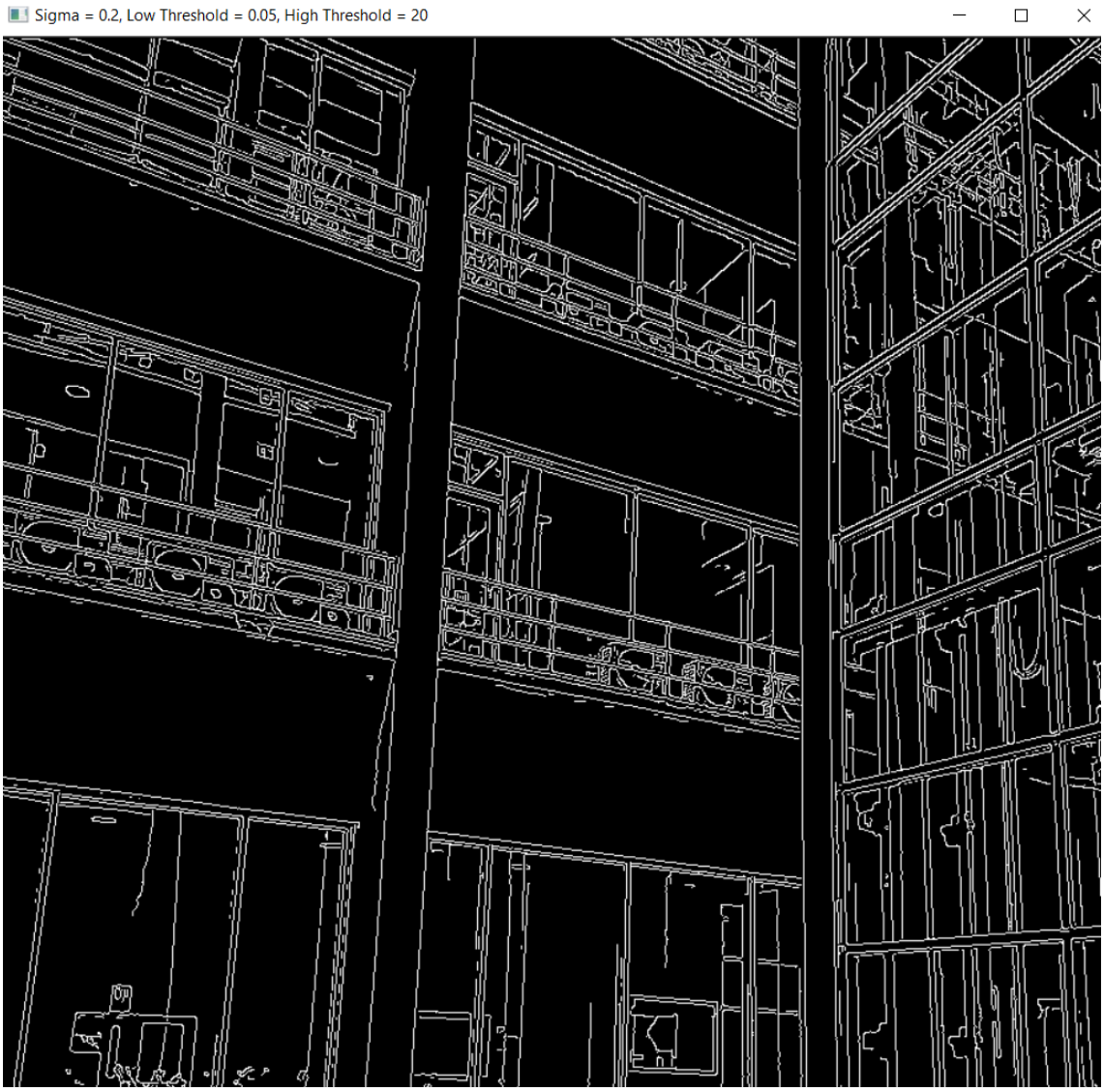


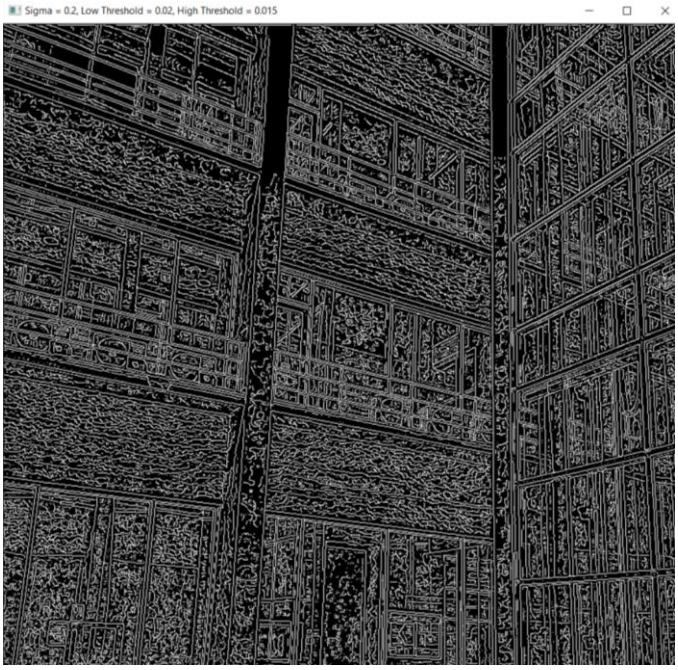

Sigma = 1, Low Threshold = 0.01, High Threshold = 40

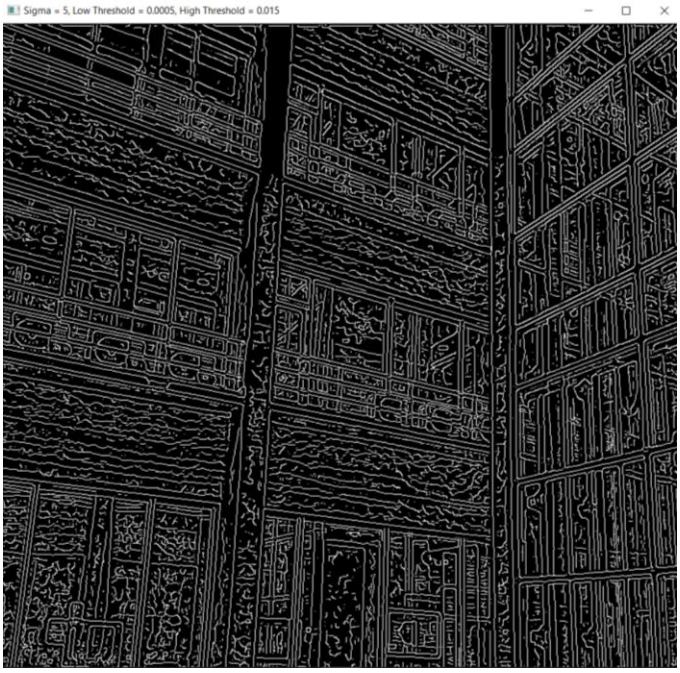

Sigma = 1, Low Threshold = 0.005, High Threshold = 40



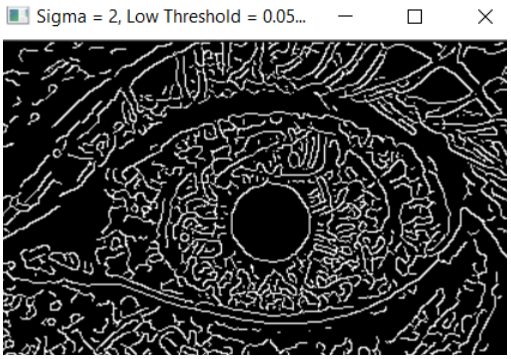
4 different results for edge2.png:

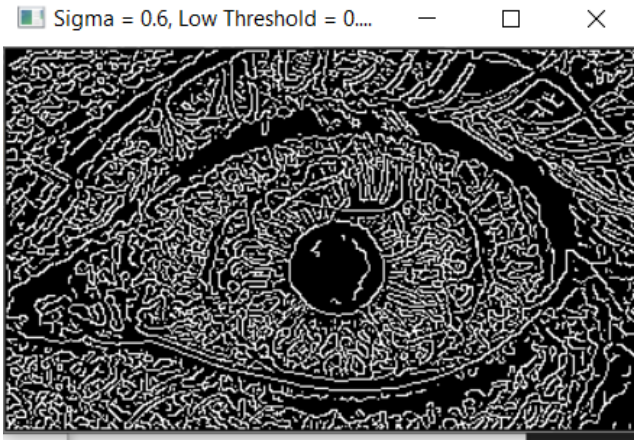

Sigma = 0.2, Low Threshold = 0.05, High Threshold = 20

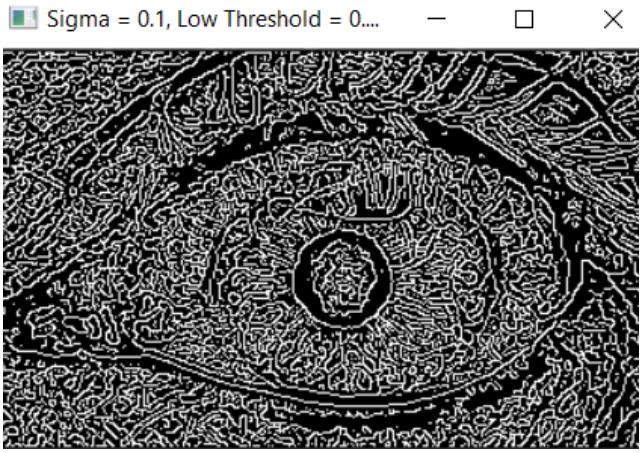

Sigma = 0.2, Low Threshold = 0.02, High Threshold = 0.015


Sigma = 5, Low Threshold = 0.0005, High Threshold = 0.015

4 different results for edge3.png:







1.2.3) Compared to Sobel Edge Detection and Prewitt Edge Detection, Canny Edge Detection does not use 3x3 mask. Instead, it uses adjustable masks. In Canny Edge Detection, before applying Sobel filters we do convolution. Therefore, we remove speckle noise with a low pass filter beforehand. After the application of Sobel filtering, we do non-maximum suppression to pick out the best pixels for edges when there is more than one possible location. As a result, Canny makes us get more precise edge detection outcomes.

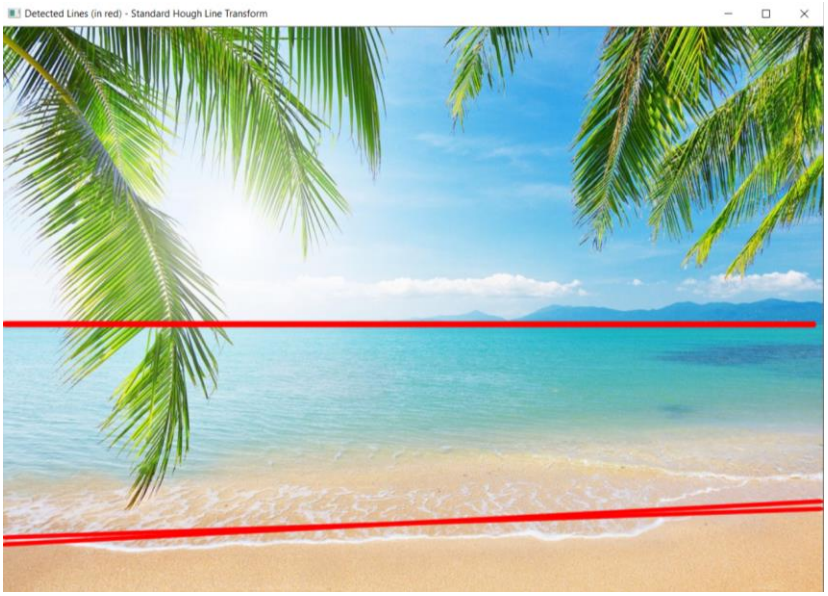
Part 2)

2.1) After applying Hough Line Transform, I get the following result for hough.png:

2.2)

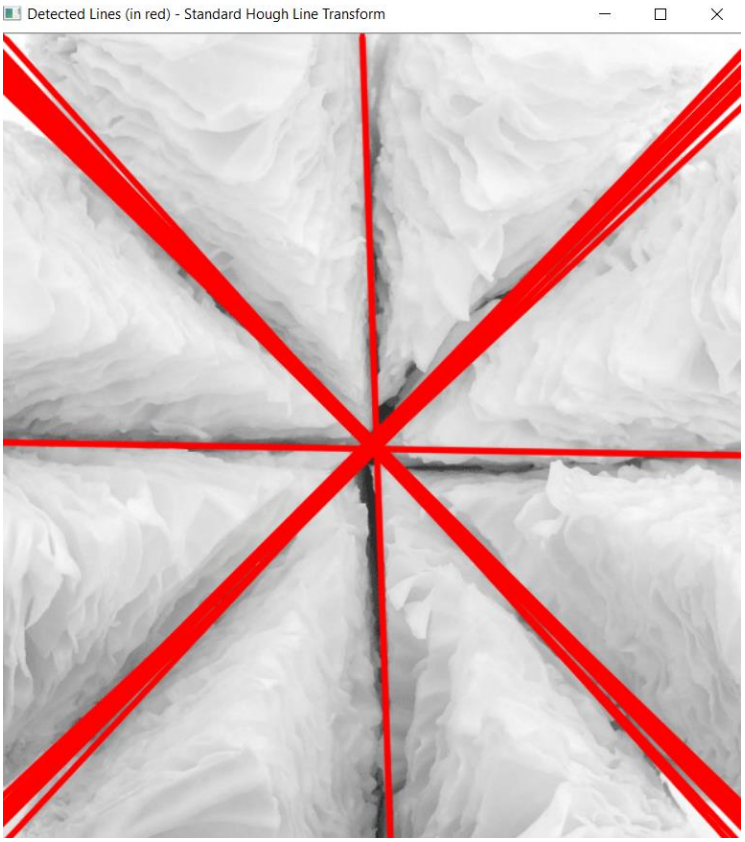
I wrote lines into the original image in order to give a better vision.

For edge1.png, result of applying Hough Line Transform is:



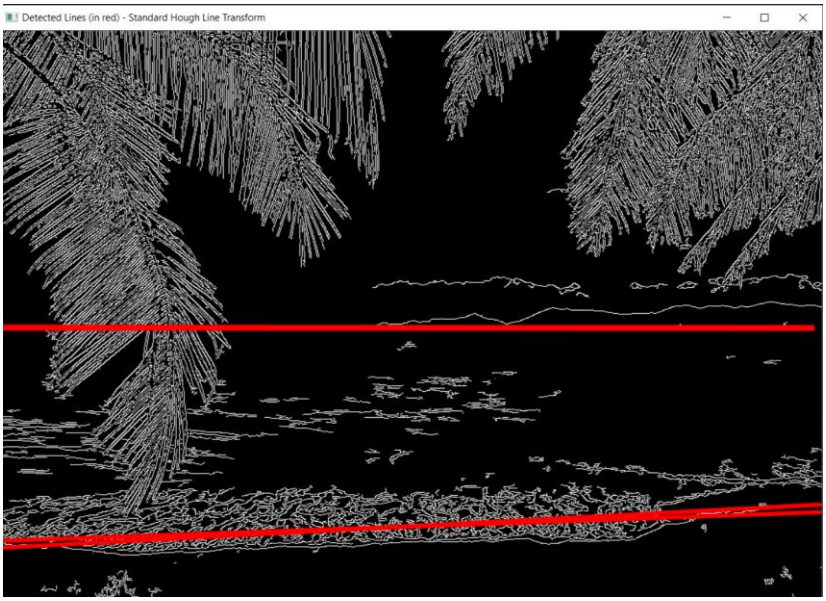
I used 300 as the threshold value.

For hough3.png, result of applying Hough Line Transform is:

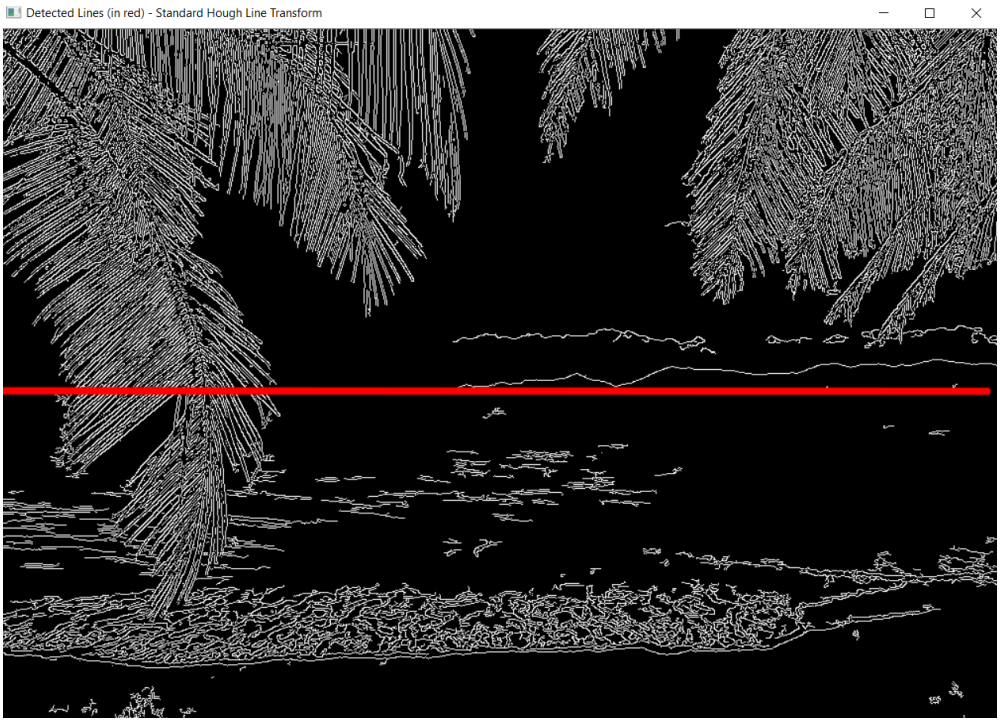


For this image, I had to use 144 as threshold value because using a higher number made the vertical red line lost. Using lesser values also resulted in not having precise lines.

2.3) For this part, assignment states I have to use edge.png but there is no edge.png. Thus, I assume it means edge1.png because the other images do not have horizons.



As it can be seen, I had managed to line the horizon and beach level with 1 as rho, pi/180 as theta and 300 as threshold. But to find only the horizon, threshold should at least be 317.



I think the overall performance of this procedure was adequate.