EE417 - Homework Assignment - Report

Baris Sevilmis
Due Date: 28/10/2018

Edge Detection

This assignment will focus on most used edge detection algorithms on various images. Three images were used for this process, such that effect of edge detection algorithms can be discussed on different type of images. Algorithms that were used for the assignment were provided by Image Processing ToolBox of MatLab. These algorithms are the following: Prewitt, Roberts, Canny, LoG and lastly Sobel. MatLab code for the work is provided below in Listing 1. Discussion will begin in the next section of the report.

Listing 1: MatLab Code for Various Edge Detection Algorithms

```
function J = barissevilmis_edgedetect(img, k)
    [row, col, ch] = size(img);

    if(ch == 3)
        rgbimg = rgb2gray(img);
    end

    rgbimg = double(rgbimg);
    J = zeros(size(img));

    if(k == 1)
        J = edge(rgbimg,'Prewitt');
        S = Prewitt;
    elseif(k == 2)
        J = edge(rgbimg,'Roberts');
        S = Roberts;
    elseif(k == 3)
        J = edge(rgbimg,'Canny');
        S = Canny;
    elseif(k == 4)
        J = edge(rgbimg,'log');
        S = LoG;
    elseif(k == 5)
        J = edge(rgbimg,'Sobel');
        S = Sobel;

    end

    subplot(1,2,1);imshow(img);
    title('Original Image');
    subplot(1,2,2);imshow(J);
    title(['Edge Filtered Image with ', S]);
end
```

Sobel is a well known edge detection technique, that uses a mask that is formed from first order derivative. Sobel filters can be seen in Figure 1. Although it is a successful method in terms of detecting most of the edges, it is obvious that there are weak points. Some of the real edges are not considered as edges by the Sobel. Mainly, most obvious edges are detected. Therefore, conclusion would be that Sobel achieves success, but remains lacking in performance. Results of Sobel are depicted in Figure 2, 3 and 4.

-1	-2	-1
0	0	0
1	2	1

G_x

-1	0	1
-2	0	2
-1	0	1

G_y

Figure 1: Sobel Filter

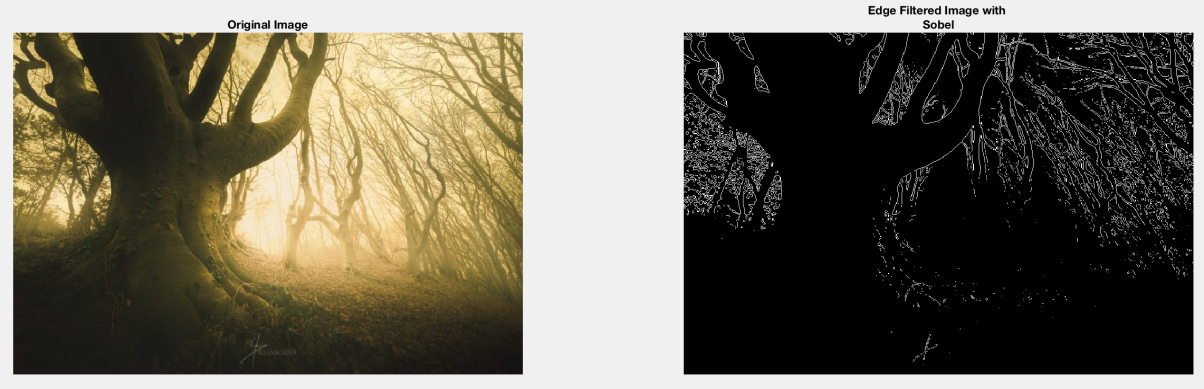

Figure 2: Sobel Edge Detection-1

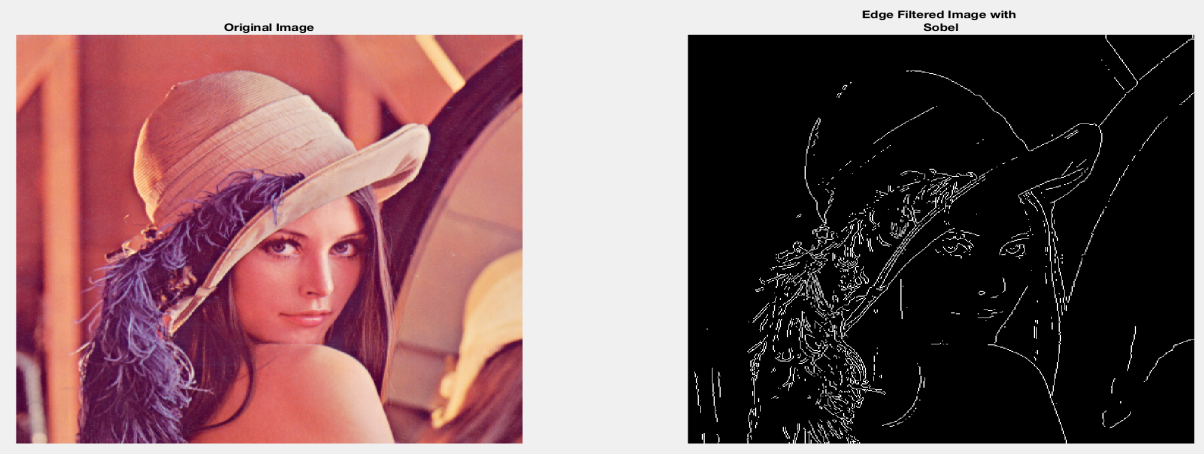

Figure 3: Sobel Edge Detection-2

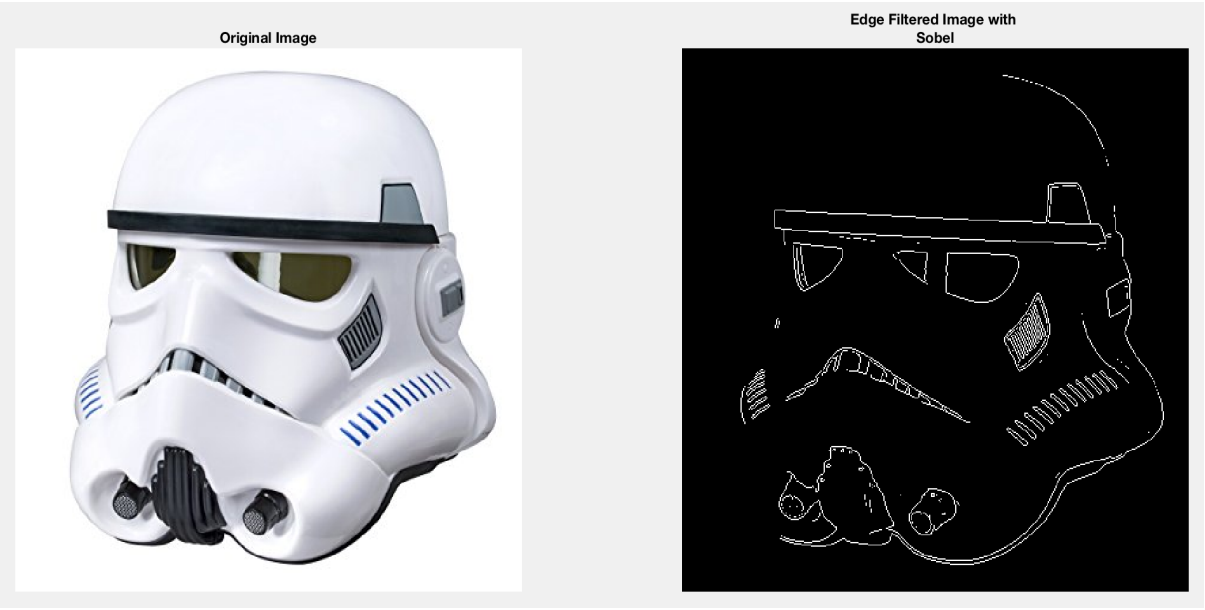

Figure 4: Sobel Edge Detection-3

Prewitt edge detection is very similar to Sobel, such that Prewitt uses a very similar filter to Sobel. Prewitt filters are depicted in Figure 5. Therefore, results of Prewitt and Sobel are almost the same, as a matter of fact it would be very hard for us to spot the differences. Perhaps, inspecting results in a very detailed fashion would enable us to notice differences. To conclude, as said previously in Sobel, Prewitt filter remains lacking though its success in detecting most of the edges. There are still many edges undetected, that can be clearly seen. Results of Prewitt Edge Detection are demonstrated in Figure 6, 7 and 8. These results should be compared with previous corresponding Sobel results for better understanding.

-1	0	+1
-1	0	+1
-1	0	+1

G_x

+1	+1	+1
0	0	0
-1	-1	-1

G_y

Figure 5: Prewitt Filter

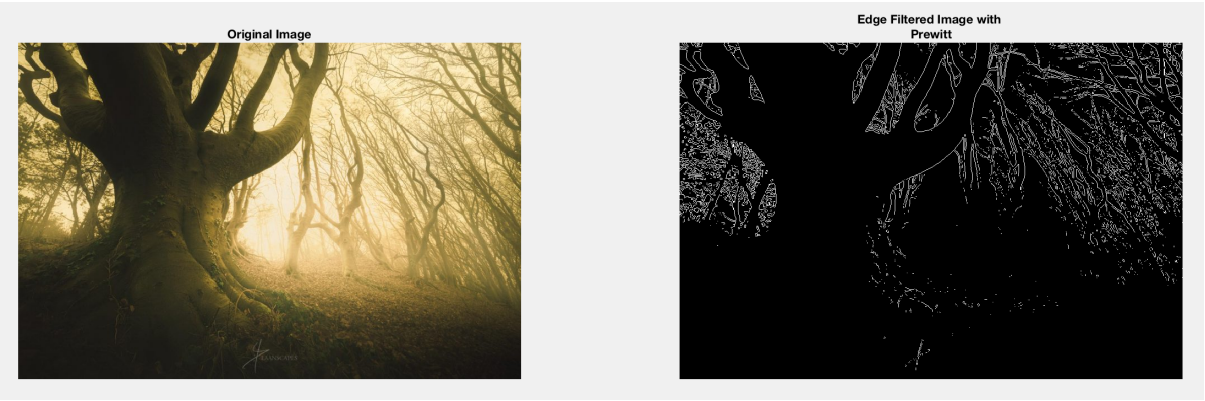

Figure 6: Prewitt Edge Detection-1

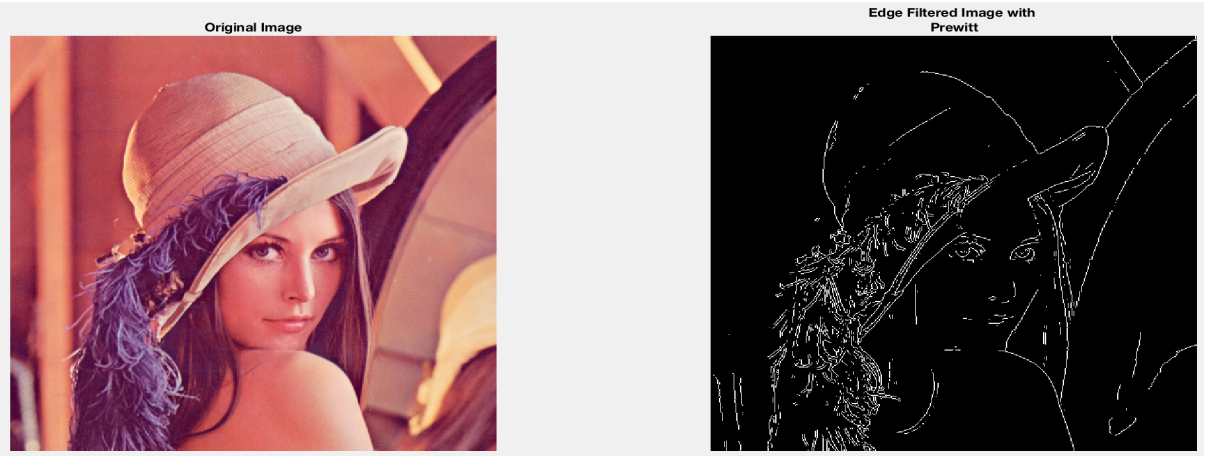

Figure 7: Prewitt Edge Detection-2

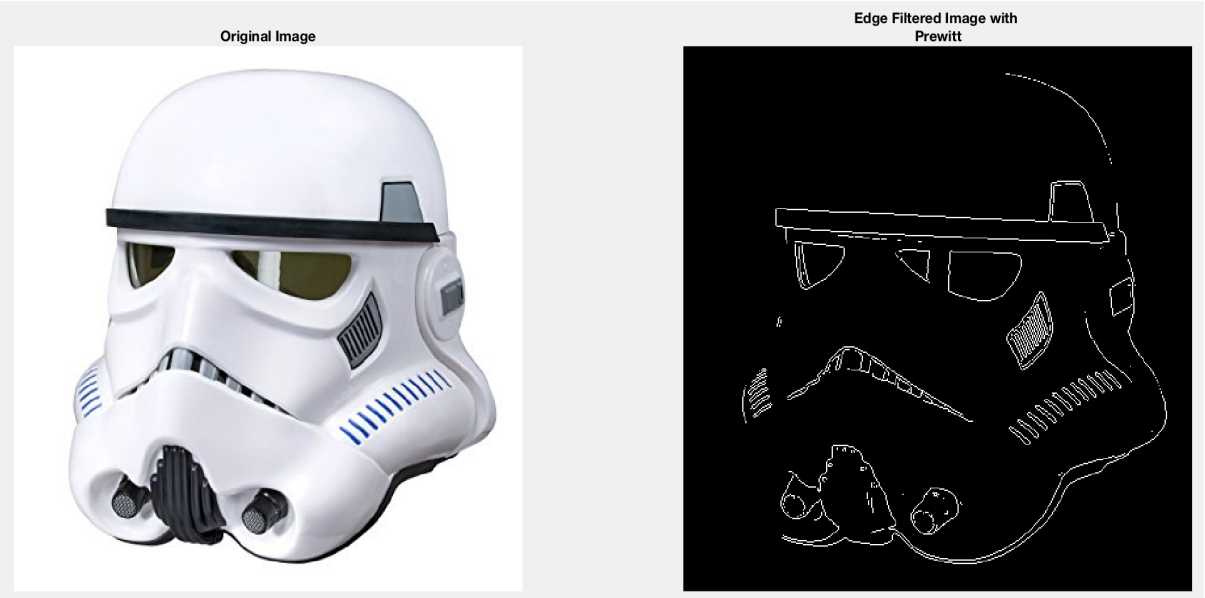

Figure 8: Prewitt Edge Detection-3

Roberts Edge Detection method is also similar to Sobel and Prewitt with different horizontal and vertical masks. Only difference is that Roberts filters, namely Robert Cross Edge Detectors, are 2x2 filters instead of 3x3 filters. Roberts filters are shown in Figure 9. Roberts filters achieves success in detecting edges as previous techniques, yet if looked closely results of Roberts Filters are more dim in comparison to Sobel and Prewitt. In other words, there are less edges detected even difference is minimal, additionally detected images are less distinct again in comparison to Sobel and Prewitt. In conclusion, Sobel and Prewitt may yield very similar results, though Roberts Cross Edge Detection does not achieve the same level of edge detection as them. Figure 10, 11 and 12 depicts the results of Roberts Cross Edge Detection.

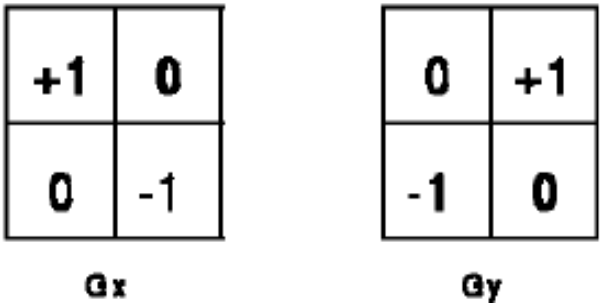

Figure 9: Roberts Cross Edge Detectors

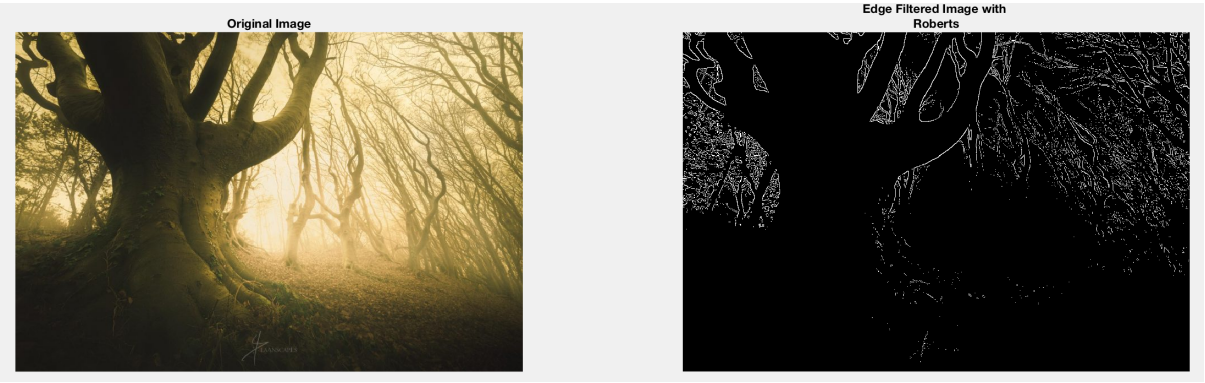

Figure 10: Roberts Edge Detection-1

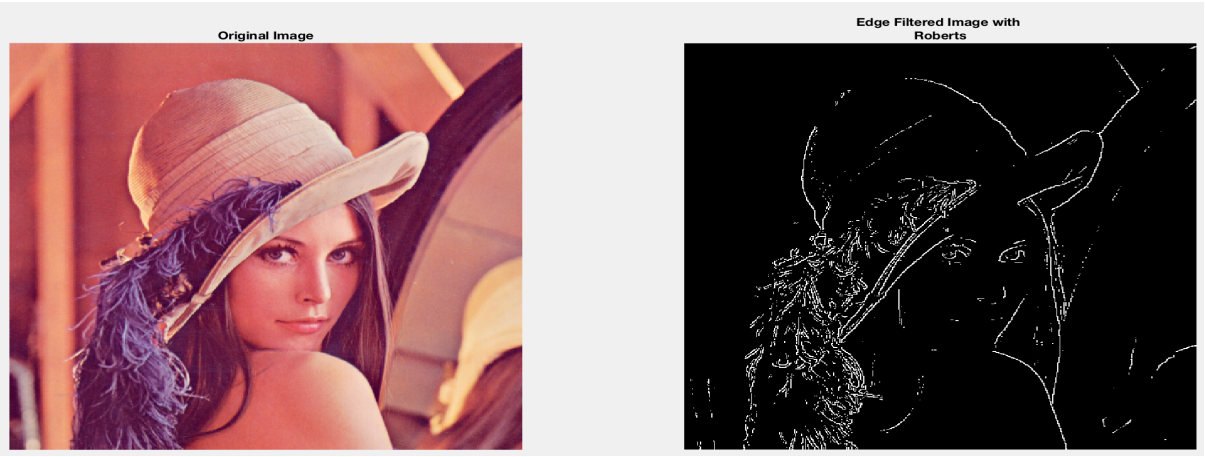

Figure 11: Roberts Edge Detection-2

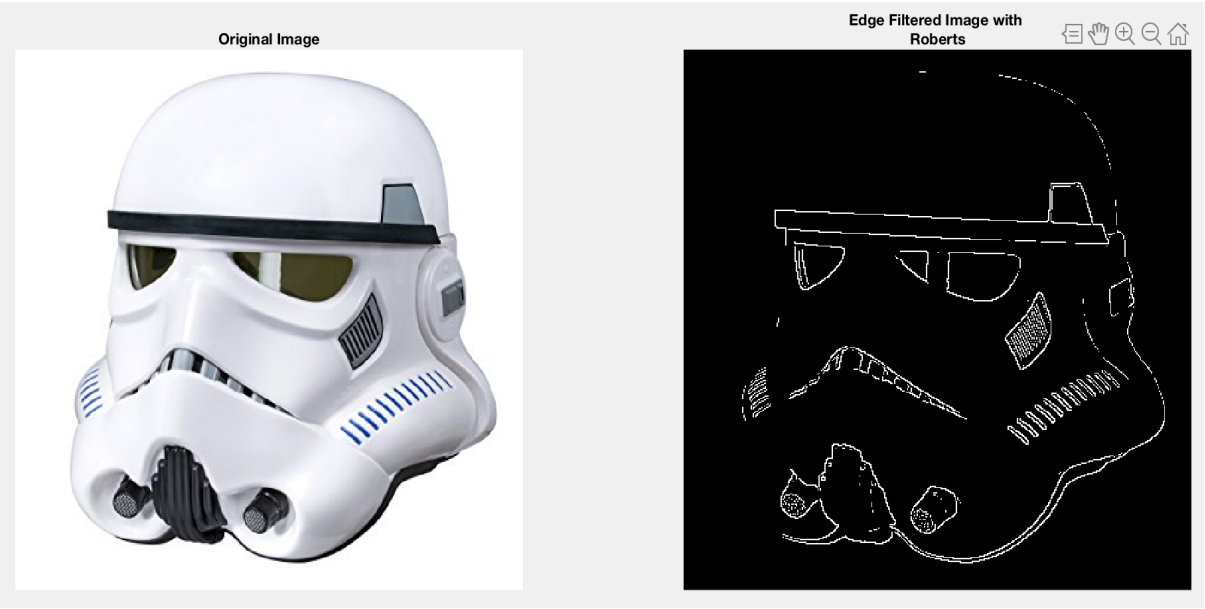

Figure 12: Roberts Edge Detection-3

Canny Edge Detection is not anymore a technique for edge detection as Sobel, Roberts and Prewitt, but it can be considered as an advanced algorithm. Canny edge detection requires some crucial steps before detecting edges such that results would be optimal. These steps, namely Canny Edge Detection algorithms phases, are demonstrated in Figure 13 in most simple way. Firstly, image is processed by Gaussian filter for noise removal. Then, gradient calculation takes place, in which first order derivatives such as Sobel are used. In this step, both gradients, edge strengths and edge orientations are computed for later on both removing the unrelated image features and connecting corresponding edges. Then, Nonmax-Supression takes place in which only the high intensity pixels are selected. In other words, non-maximum intensity values are suppressed in the given edge orientation, and maximum intensity values, namely edges, are filtered. Lastly, 2 thresholds are used to distinguish exact edges by removing the extra pixels around images. To conclude, last two steps separate the edges and thin them out. There are some unique

advantages of this algorithm that are the following:

- Less sensitivity noise
- Removal of streaking by 2 level thresholding
- Decent localization and gradient calculation for generating thin, one-pixel wide images

Results of canny edge detection are depicted in Figure 14, 15 and 16.

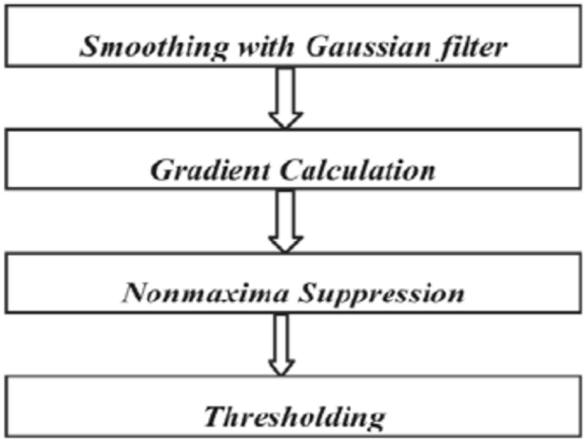

Figure 13: Canny Edge Detection

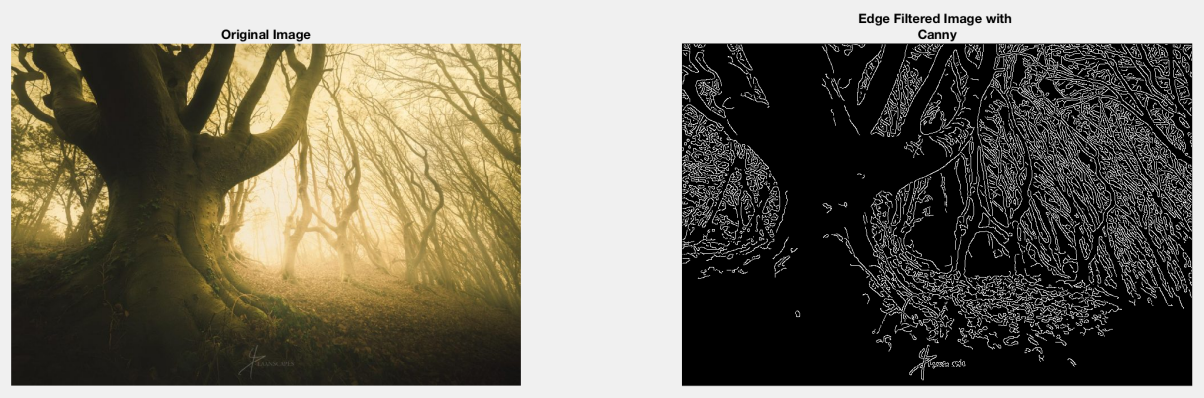

Figure 14: Canny Edge Detection-1

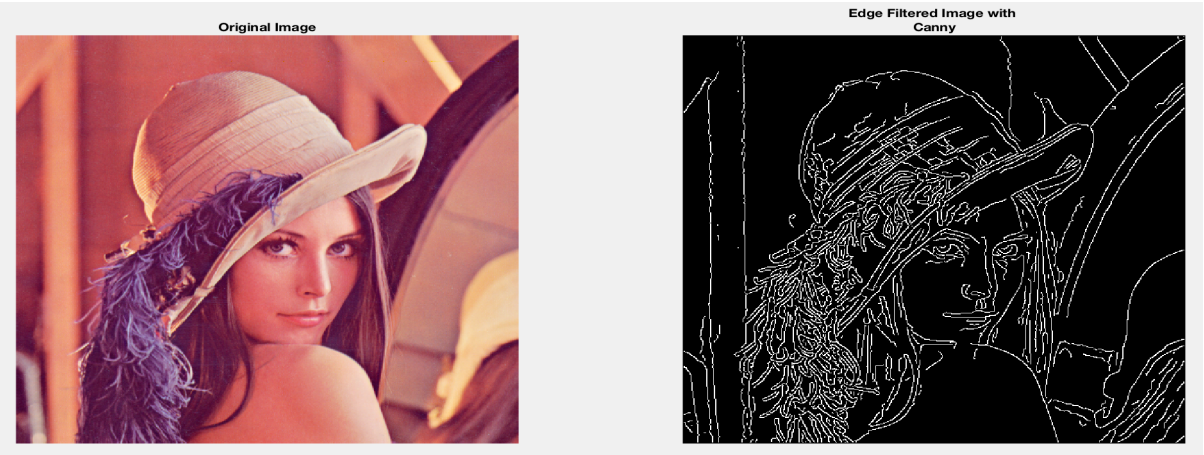

Figure 15: Canny Edge Detection-2

Figure 16: Canny Edge Detection-3

Laplacian of Gaussian (LoG) will be the last algorithm to be discussed, as it is another unique edge detection technique and additionally, it is an advanced algorithm as Canny in comparison to methods such as Sobel, Prewitt and Roberts. LoG algorithm is shown in Figure 17 in most basic sense. Basically, image is smoothed out as in Canny Algorithm with a method such as Gaussian Smoothing. After, smoothing out the image, resulting image is processed with Laplacian Filter to find Zero-Crossings. Laplacian filter is a second order derivative filter unlike first order derivative filters. Therefore, a disadvantage of this algorithm is its extreme sensitivity to noise. Reason is as already known taking derivatives results in increased noise. However, end result of Laplacian Smoothing is almost as smooth as Canny Edge Detection results meaning that it is much better than first order derivative methods such as Sobel. To conclude, results of LoG Edge Detection are depicted in Figure 18, 19 and 20. Noise effect can be clearly seen in results, even better when compared with Canny results.

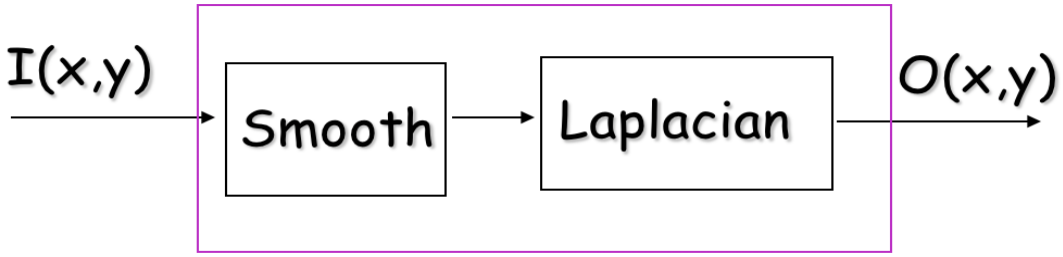

Figure 17: LoG Edge Detection

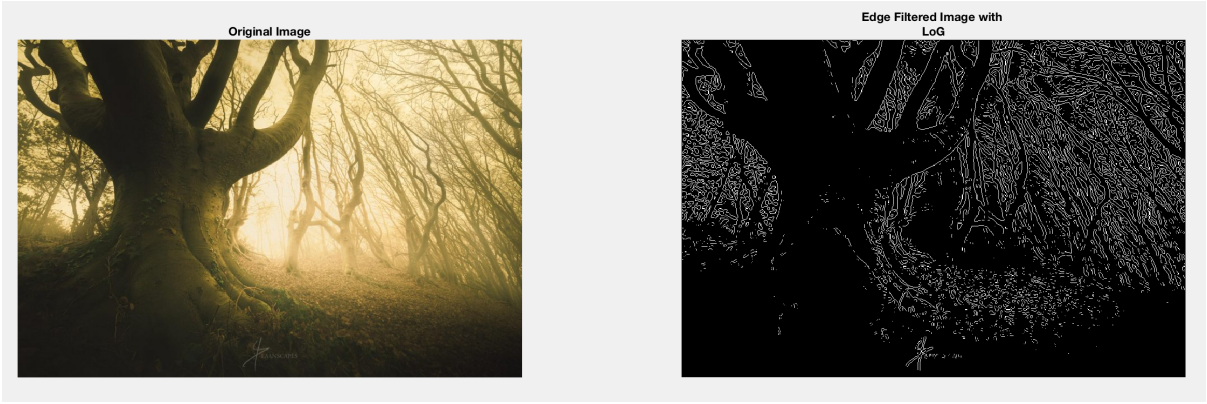

Figure 18: LoG Edge Detection-1

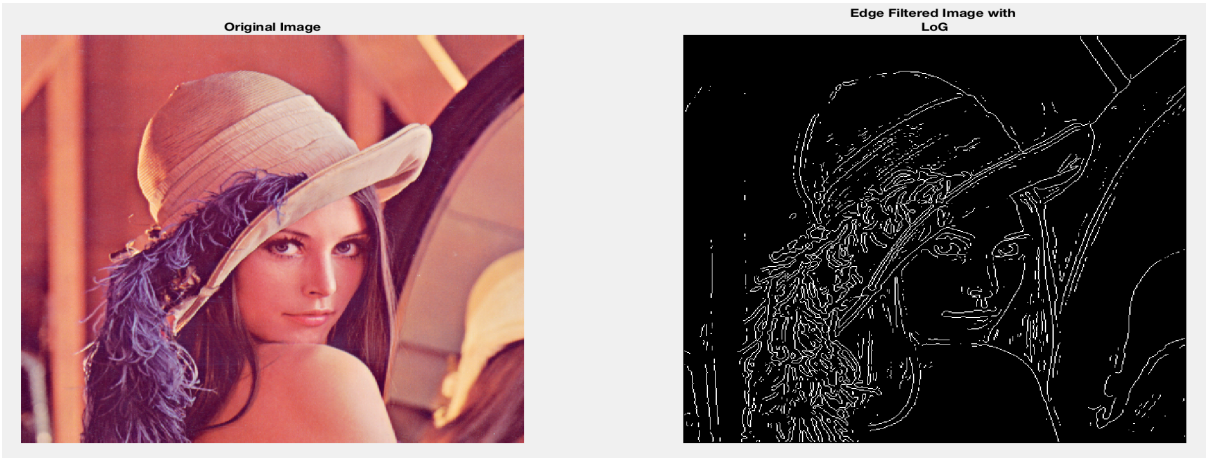

Figure 19: LoG Edge Detection-2

Figure 20: Log Edge Detection-3

Discussion

Details for five different edge detection techniques were explained above with their advantages and disadvantages. Algorithms such as Sobel, Prewitt and Roberts can be implemented easily. Their edge detection performance prove successful but remain lacking in terms of missing edges and relatively thick edges. On the other hand, Canny's algorithm provides a superior edge detection by smoothing, non-max supression and two level thresholding, though it would be definetly harder to implement it in an efficient mannor. LoG filter proves also great success, and may be perhaps implemented easier than Canny. However, LoG filter is extra sensitive to noise in images because it is second order derivative, but still it provides better edge detection on most images in comparison to Sobel, Prewitt and Roberts.

To conclude, success of these algorithms in decreasing order is the following:

1. Canny's Edge Detection
2. LoG Edge Detection
3. Sobel Edge Detection
4. Prewitt Edge Detection
5. Roberts Edge Detection