Machine Vision 1 Exercise Documentation

Calvo Salazar Santiago, 12450801

October 2025

1 Experiment

1.1

σ represents the standard deviation of the gaussian distribution. Therefore, in order to have a representative gaussian bell to use for filtering, the width of the Gaussian filter should depend on the value of σ, the higher σ, the greater the kernel width should be to correctly represent a complete gaussian bell in the kernel. If we have a high sigma and a small kernel width, our kernel is only going to contain the "top" of the gaussian bell, and as a consequence all the pixels are going to be weighted similarly.
By decreasing the sigma and the kernel width we have a thinner gaussian bell so the filtering effect would be light; on the other hand, increasing these values increases the filtering effect, hence the edge detector shows fewer details as shown in figures 1 and 2.

In practice, having a small kernel size with a high σ, decreases the filtering effect causing as a result the increase of the noise in the edge detector results as shown in figure 3

Figure 1: edge detection with $\sigma = 2$

Figure 2: edge detection with $\sigma = 6$

Figure 3: edge detection with $\sigma = 3$
and kernel width of 5

1.2

Blurring effectively smooth the pixel intensities and suppress higher peaks by averaging out rapid intensity changes, as it can be seen in the figure 4

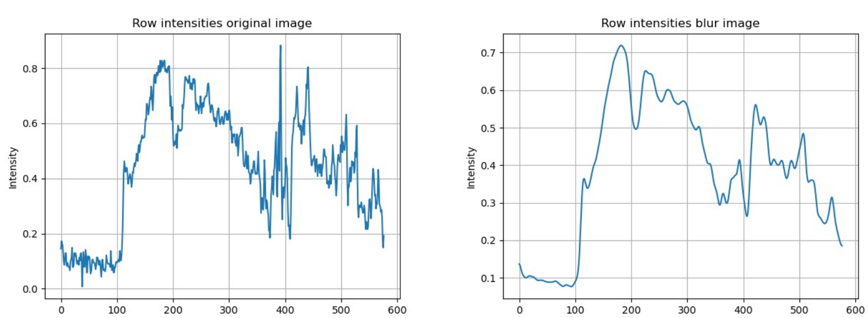

Figure 4: Effect of blurring on pixel intensities.

2 Experiment

2.1

By changing the Hysteresis Thresholding parameters we are effectively suppressing previously detected edges. depending on the threshold values we are suppressing real edges or false edges, as it can be seen in the figure 5 we have less edges than in figure 6 as a result of higher threshold values for the first image.

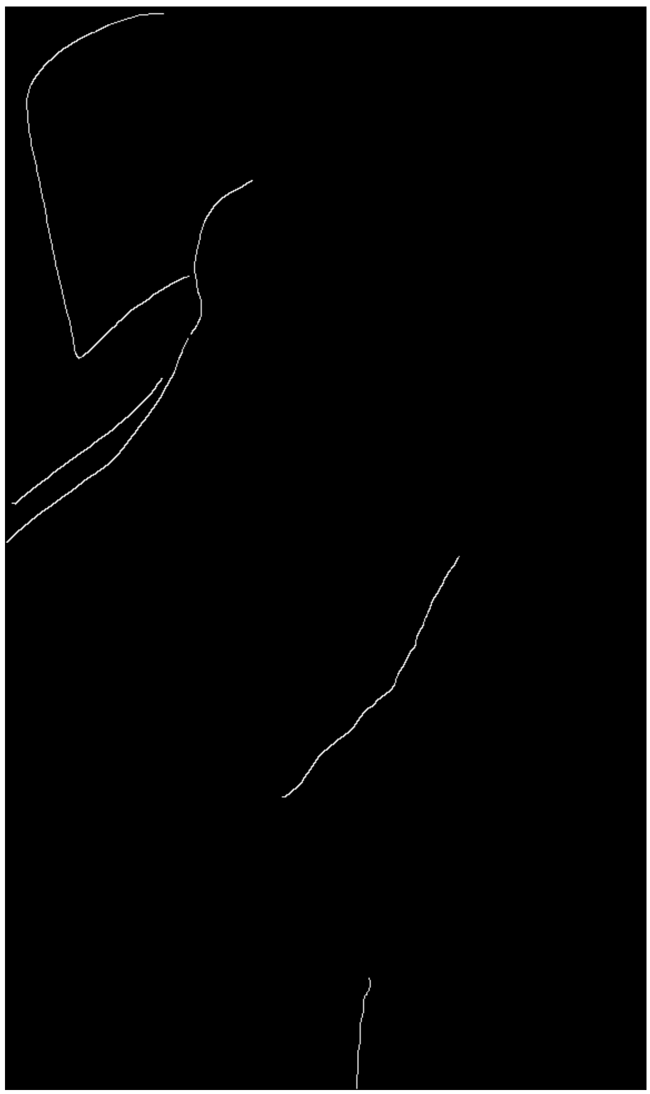

Figure 5: Edges with $threshold_{low} = 0.3$ and $threshold_{high} = 0.5$

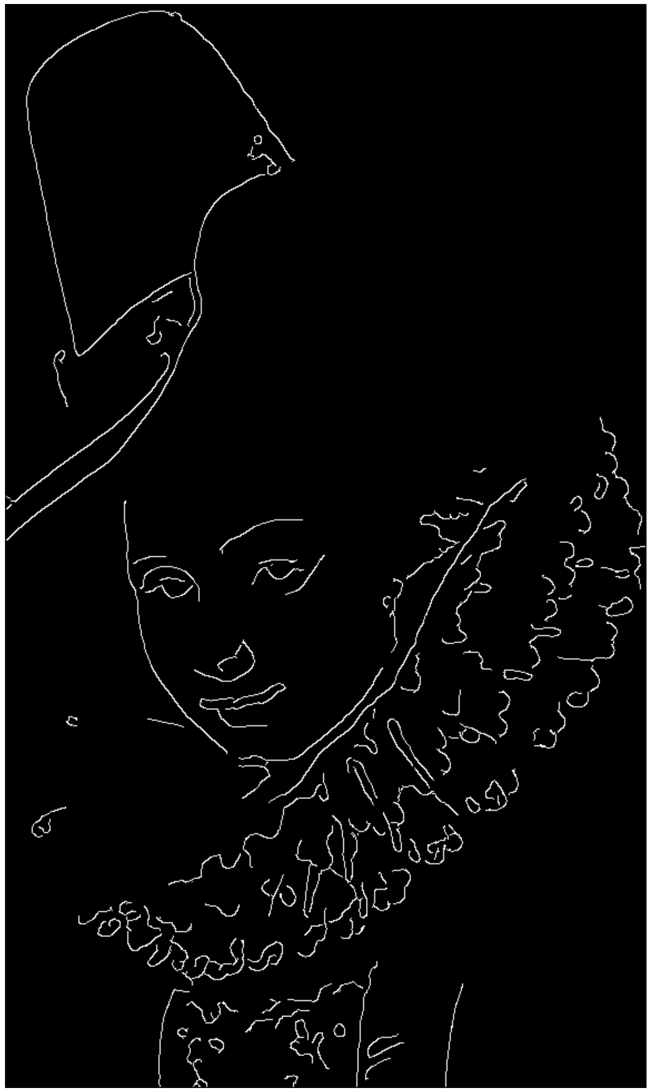

Figure 6: Edges with $threshold_{low} = 0.1$ and $threshold_{high} = 0.2$

2.2

Using the same thresholds on different images give inconsistent results, since different images can have a completely different pixel intensity distribution. For instance, the non-zero pixels of one image can follow a gaussian distribution center around 0.3 while for other image it can be centered around 0.7. (as it can be seen in the figure 7 using the same thresholds for different images changes the results). If the images where took under the exact same conditions (illumination, same camera, etc) the same thresholds can be used for 2 different images.

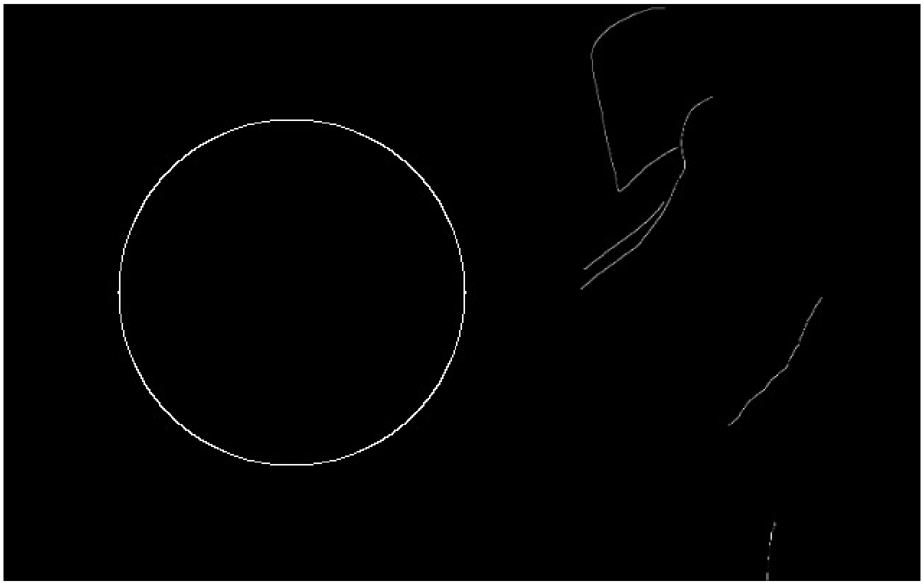

Figure 7: edges using $threshold_{low} = 0.3$ and $threshold_{high} = 0.5$ for different images

3 Experiment

3.1

Figure 8: Noise effect on image edges, first row $\sigma = 0$, second row $\sigma = 0.2$, third row $\sigma = 0.5$

As it is shown in the figure 8, by adding noise to the image the resulting edges are more noisy as well. this is due to the fact, that the gradient operation using the Sobel operator maximizes the noise of the image, introducing artificial new edges in the result.

3.2

Increasing σ effectively increases the filtering effect of the gaussian filter which reduces more the noise but also can lead to the smoothing of real edges, so with higher σ we are also losing information about the edges, therefore there is a upper limit for σ.
In addition, by decreasing the hysteresis thresholds, specially the higher threshold, we get more edges on the result, as we allow more values to pass the threshold. Hence, it is possible to tune the σ and the thresholds to get reasonable results (as shown in figure 9), but these results hardly are going to be as good as the results that can be obtained with a noiseless image.

Figure 9: Noisy image (standard deviation = 0.2), gaussian filter with $\sigma = 4$ and $threshold_{low} = 0.01$ and $threshold_{high} = 0.05$

4 References

As main Reference for the implementation of the exercise, the following article was used: `https://medium.com/@soham.phanse/canny-edge-detection-in-python-4cbc1209adbc`