

Programming Assignment 1

CAP- 5415

Canny Edge Detection

In this assignment we had to implement Canny edge detection without the help of libraries except some basic one, to implement functions such as Gaussian filters, convolutions, non-maximum suppression. So, to test our code we take 3 sample images from Berkeley Segmentation dataset, and we provide final output also we showcase output at three different standard deviations values when smoothing.

Canny Edge Detection Algorithm

1. We read the Image, and then to remove the noise the image is smoothened with the kernel size of 3 and $\sigma = 1$ and then we normalize it.
2. Then we calculate the derivative gradient of filtered Image. That is when the image has been already smoothened. The derivatives I_x and I_y are calculated by convolving I with 1-D gaussian kernels.
3. We then find the Magnitude of the edge response by combining x and y components.
4. For NMS i.e., Non maximum suppression is basically means if the current magnitude of the pixel is greater than neighboring pixels nothing changes else it is set to zero.
5. For hysteresis thresholding we define two thresholds high and low respectively. The high threshold is used to start the edge and low threshold to continue them.

Results

Sigma (0.2)



Fig: Bird



Fig: I_x



Fig: I_y



Fig: I_x dash



Fig: I_y dash

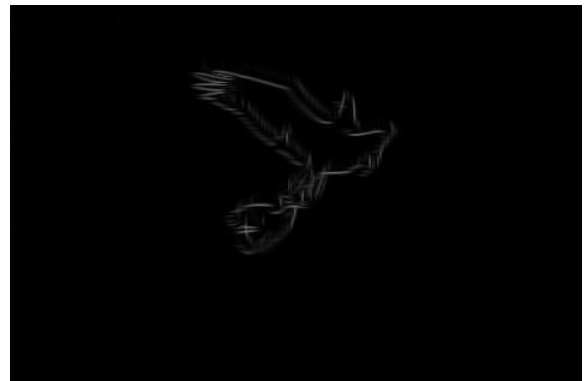


Fig: M_{xy}

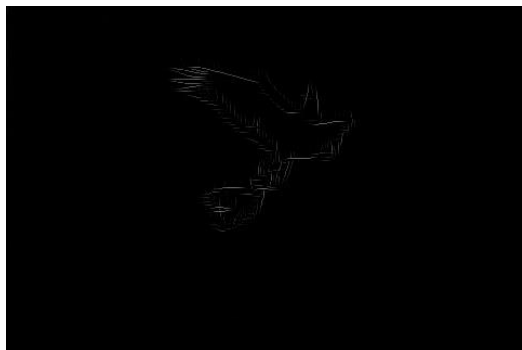


Fig: NMS

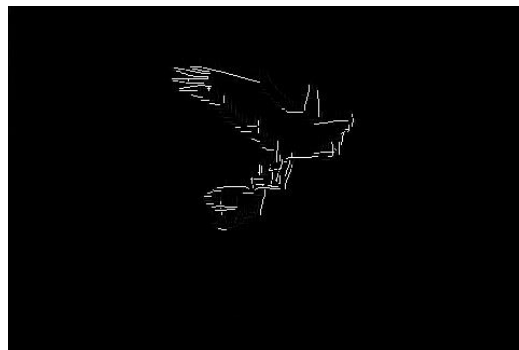


Fig: Threshold

Sigma (1.5)



Fig: Input Image



Fig: I_x



Fig: I_y



Fig: I_x dash



Fig: I_y dash

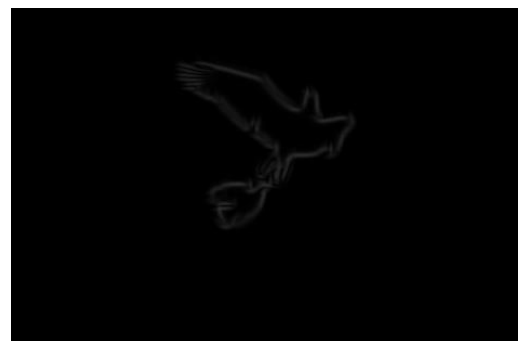


Fig: M_{xy}

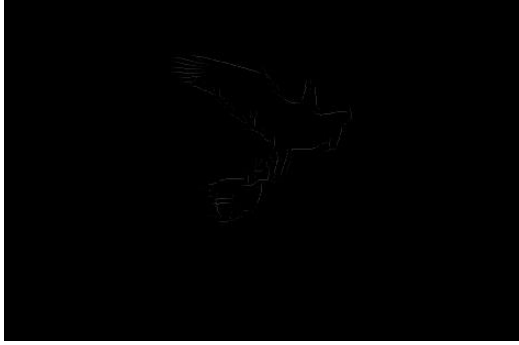


Fig: NMS output

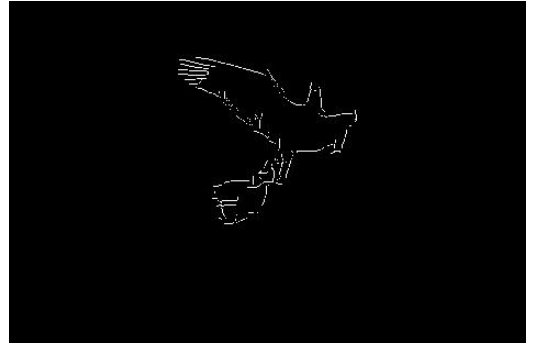


Fig: Threshold output

Sigma (3)



Fig: Input Image



Fig: I_x



Fig: I_y



Fig: I_x dash

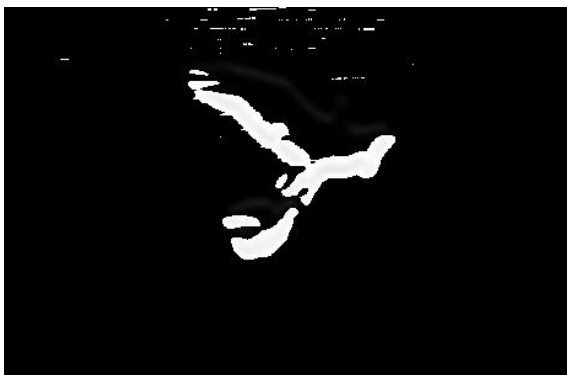


Fig: I_y dash

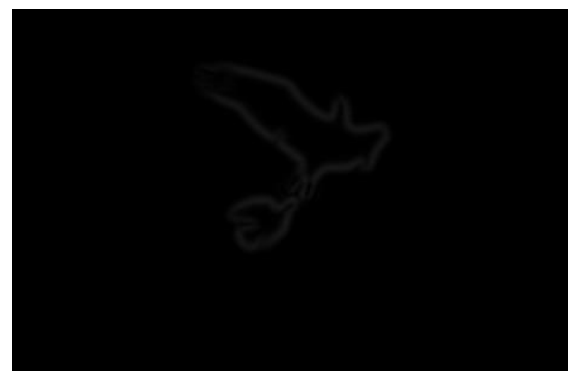


Fig: M_{xy}



Fig: NMS output

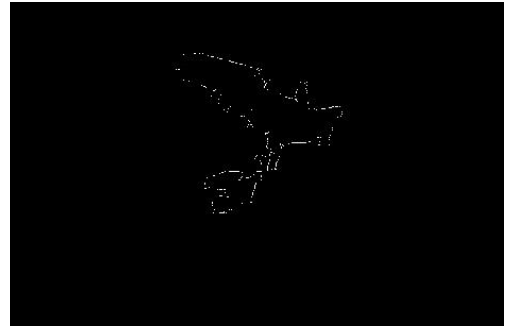


Fig: Threshold output

Conclusion

For the other two examples I have created separate folders in which all the results are stored separately in different folders for different σ values.

In conclusion higher the sigma values the output will be coarse and lower the sigma values the edges will be finer.

[GitHub link](#)