PA1 Problem 1

Michael Fielder

October 11 2021

Below is an explanation of my methods for creating a Canny Edge Detector for Problem 1 of the first programming assignment. Note that I read in and display the image using the matplotlib library in Python. Also note that the Canny algorithm is executed using the function `canny()` function, which takes in the image name, σ value, and the size of the kernel to use for the convolution. Note that the image must be present in your current directory for the program to recognize it. I believe that my implementation has issues but I don't quite know where. My image does not blur effectively even though I'm pretty sure I implemented the convolution and kernel correctly. In order to calculate `G`, I map the 1-D Gaussian function over the array, `x = [-2, -1, 0, 1, 2]` for a kernel size of 5. The same mapping process occurs for the derivative Gaussian kernel and that produces the gradients correctly. The only time I can get a more significant blur is if I increase the kernel size. I also made sure to normalize the G filter, but it didn't really alleviate the problem.

A couple notes on the program. I add padding to image to ensure that each intermediary representation is the same size as the original image. This is to make calculations easier and simplify my methods.

Note that the three intermediary images can be found in the attached images, `Figure_1.png`, `Figure_2.png`, `Figure_3.png` as I couldn't find a good way of displaying them in the document at a high quality. I used a $\sigma = 1$ and a kernel size of 5. I believe that I implemented the derivative kernels correctly as you can see from the provided directional gradients for each image. The images sadly does not blur correctly, and I don't know why.

The sigmas picture can be found in the attached picture, `Sigmas.png`. Where I use $\sigma = 1, 3, 6$ from left to right. As you can see, there isn't a huge difference between the three pictures, which leads me to believe that I did something incorrectly. If I had to choose a sigma value, I would most likely choose $\sigma = 2$ or 3 as a sigma of 1 is too low and causes too much noise and a sigma any higher than 3 will break the image.