Canny Edge Detection Project Report

Nick Kallfa

In this project we implemented the Canny Edge Detector using ImageJ. Below are the results for the two images included in the project file:

Figure 1A

Figure 1B

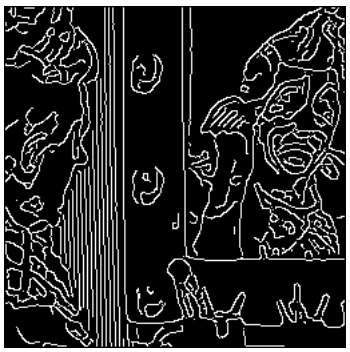
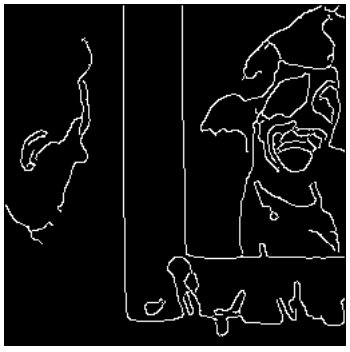


Figure 1B was obtained by setting a Gaussian blur of radius 5 and standard deviation 2 with a low threshold of 20 and a high threshold of 40. One thing I was surprised about was that I needed to set a high threshold that actually seemed quite low. In other variations I had seen online of the Canny Edge Detector in class I read that the high threshold could be set quite high often to obtain good results. For my particular program this was not the case. If I set the threshold too high then it did not seem to trace out as many of the edges, however it seemed to work best when both the low and high threshold were low. I am including in figure 2 the output of the same image under the same parameter controls, but this time using a high threshold of 150.

Figure 2



Its clear that not all of the edges were traced out. I was unable to determine a clear reason why this was occurring, but I believe the output from non-maximal suppression found the edges, but the pixel intensities of those edges varied quite largely so in order to actually trace out an edge well enough we needed to set the high threshold low in order to do that. I am including the other image in figure 3 with its result from the Canny Edge Detector.

Figure 3A

Figure 3B

The above image was computed using the parameters 9 for size, 2 for standard deviation, 20 for a low threshold, and 40 for a high threshold. Again, the same effect occurred to this image as it did for the previous image if I kept the parameters the same, but increased the high threshold.

The following three images show the effects of the Canny Edge Detector if we change the standard deviation keeping all of the other parameters the same. We used a kernel size of 13, a low threshold of 20, and a high threshold of 40 for each of the following three figures, however, we changed the standard deviation from 1 to 5 to 10 each time.

Figure 4 STD: 1

Figure 5 STD: 5

Figure 6 STD: 10

We see that increasing the standard deviation causes less edges to be found. This is because the standard deviation determines how many values will be weighted in some neighborhood of a targeted pixel. If the standard deviation is low, then only pixels very close in some neighborhood will be weighted and actually matter in the Gaussian blur. Pixels further away will either have a smaller weight or not matter at all in determining the blur. On the other hand, if we increase the standard deviation then not only will pixels that are close to the targeted pixel be weighted but so will pixels that are further away from the targeted pixel. This means that the image will be blurrier with a higher standard deviation than a lower standard deviation. Thus, if its blurrier there arent as many sharp changes in local intensities that are occurring. Local intensity changes give rise to edges and as a result less edges will be detected.