Color Edge Detection in Low Contrast Images: Midterm Status Progress Report

Brief Description:

For my final project, I am developing various color edge detection algorithms to be used on low contrast images to see if they perform better than a classic grayscale canny edge detector.

Achievements:

I have implemented the regular canny detector (same as the one from class) in addition to testing its output on images compared to opencv's canny detector. Additionally, I have implemented two RGB edge detectors, one of which is a version of a pre-existing algorithm with modifications that improve the output[1]. This color edge detector was based on an adaptive median kernel for smoothing, using a weighted average to convert from RGB color space to grayscale, a maximum directional difference calculation, and an adaptive thresholding component. The second implemented color edge detector is one that I proposed in my earlier project proposal. It envolves crisp pre-aggregation[2], which is simply averaging the values of the different RGB channels to convert to one single gray channel. I have preliminary results for these two edge detection algorithms (additionally in comparison with traditional grayscale canny edge detectors) that can be seen in Figure 1 and Figure 2. Algorithm 1 has had some bugs that I am still working out which is why it's performance is not as good as I hoped but algorithm 2 is displaying good results that I hope to improve on with the algorithms I have detailed later in this paper. Although not completed, I am currently working on a crisp pre-aggregation model for using the HSV color space that envolves hierarchical weighting.

Deviations from Original Plan and Updated Timeline:

I have expanded the color spaces that I will be implementing. I will use RGB since it is the classic color space that images are represented in and a good starting point. I have also expanded into using the HSV color space due to its use in computer graphics and expanding use in computer vision. I am also

---

[1] https://ieeexplore.ieee.org/document/5329404
I removed edge thinning from the algorithm because the operators they used were outdated and the outputs were worse when using modern operators.

[2] https://link.springer.com/article/10.1007/s44196-022-00137-x#Sec5

exploring the opponent color space due to its more human replicating approach which I think will be super interesting.

3/9-3/15: Finish crisp pre-aggregation for HSV, crisp post-aggregation for RBG and HSV.

3/15-3/20: Compare results, finish slides, presentation, and report.

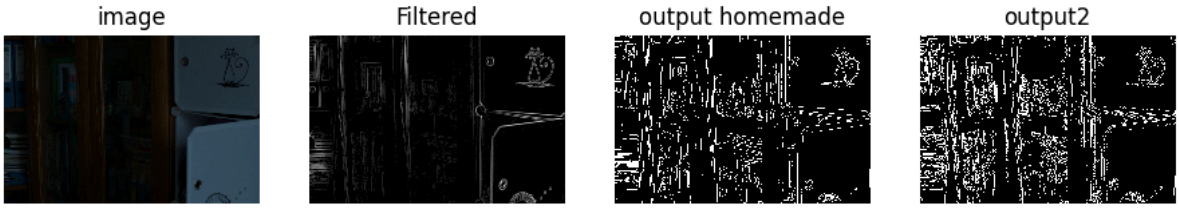

Figure 1. Image depicts the original low light image, algorithm 1's edge detection, the class' canny detector, and finally opencv's canny detector.

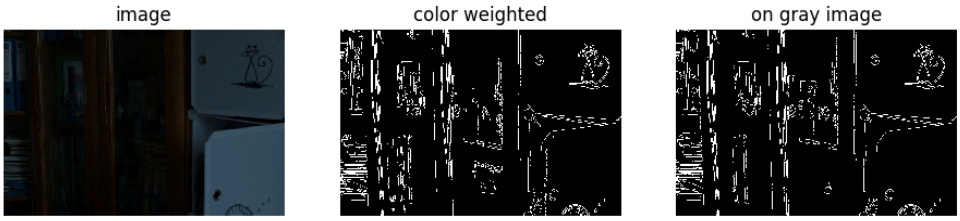

Figure 2. Image depics the original low light image, algorithm 2's edge detection and opencv's canny detector on the original image shifted to grayscale.

Future Work:

After completing my crisp pre-aggregation model for the HSV color space, I will implement the cirsp post-aggregation model for both the RGB and HSV color spaces. This involved applying an edge detection algorithm over each channel separately which produces 3 edge maps, and then combining them into a singular edge map. If I have time, I will pursue implementations of opponent color spaces.

Note:

I currently have a concussion (and have for the past week) which has made progress slower than expected/hoped due to my limited screen time as I try to heal and decrease my pain. I hope to implement opponent color space algorithms and should be able to.