CMSC 426 - P1

Gudjon Einar Magnusson

October 3, 2016

1 Gaussian Filter Bank

The Gaussian derivative filter is generated by first creating a simple Gaussian filter an then get the derivative by running a Sobel filter on it. The pixel size of the filter is defined to be $2\lceil sigma \times 3 \rceil + 1$.

For my filter bank I generated 24 different versions, 8 orientations and 3 scales, $sigma = \{1, 2, 4\}$.

2 Half-disk Masks

To generate a half-disk with radius r I first generate square matrix of size $2\lceil r \rceil + 1$. Cells that have a square distance from the center less than r^2 get a value of 1 others are 0. Next I give every cell with a row index greater than $\lceil r \rceil + 1$ a value of 0. To rotate the half-disk I used the Matlab function *imrotate*.

For my mask bank I chose to use 8 different orientations and 3 different radii, $r = \{2.8, 4.8, 6.8\}$. I found that using a radius that is a fraction instead of a whole number produces a cleaner looking circle.

3 Image Gradients

I generate gradient maps by first generating version of the image that emphases specific features and the comparing the histogram of values surrounding each pixel.

In my implementation I combine 3 different gradients, texture gradient, brightness gradient and color gradient.

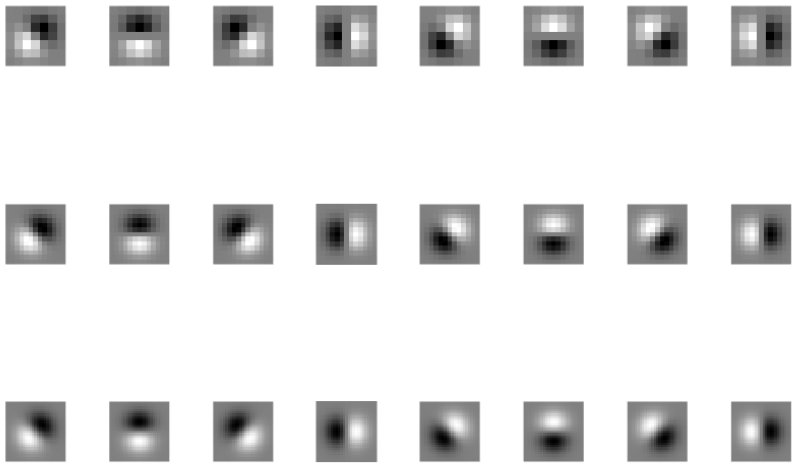


Figure 1: Gaussian Filter Bank

3.1 Texture Gradient

The texton map is generated by first filtering the image with each of the Gaussian filters and then running kmeans on the result. I chose to use 16 clusters for for the texton map.

3.2 Brightness Gradient

A brightness map is generated by grouping pixels based on the luminance value in the Lab color space. To do this I normalize the values in Lab color space to $0-1$ range, multiply by the number of groups and round up. I chose to use 32 groups for the brightness map. As you can see in figure 6a an 6b, sometimes this produces a nice separation between objects, but sometimes it does not.

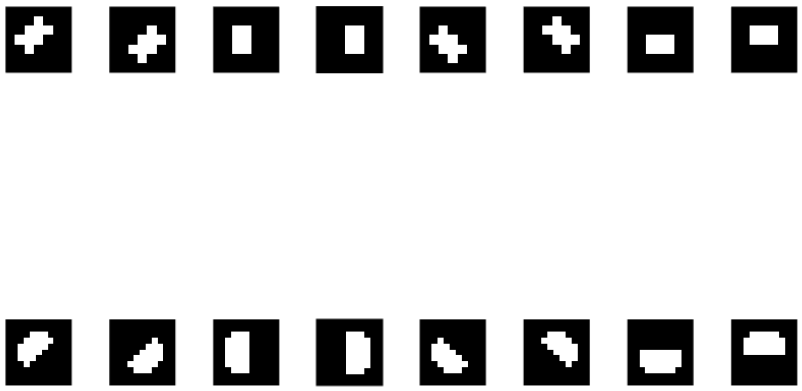


Figure 2: Half-disk Masks

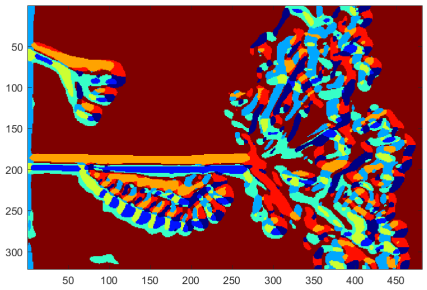
3.3 Color Gradient

To generate a color map I run kmeans on the a ans b channels of the Lab color space. Since the Lab color space is specially designed to make colors that are visually close have a small euclidean distance, kmeans should be able produce meaningful color clusters.

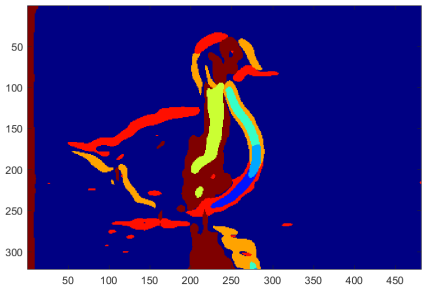
4 Results

Tweaking the size of the half-disk masks and picking a good cluster number for each of the maps turned out to be very difficult. I needed large half-disk to the be able to generalize textures that repeat over a large area but that makes the outlines very thick and fuzzy, which brings down the precision.

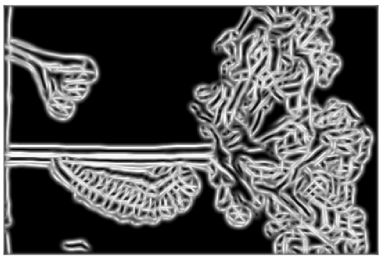
My strategy was to use low cluster numbers to find large flat clusters and small half-disks to try to find sharp edges. Large clusters causes each of the gradient maps to lose some edges but between all of them they usually find all the important edges.



(a) Texton map



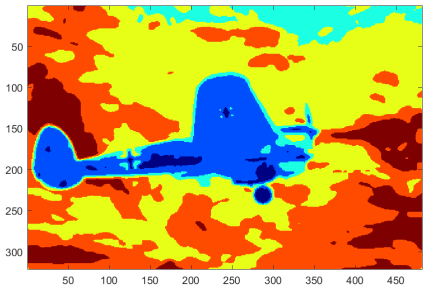
(b) Texton map



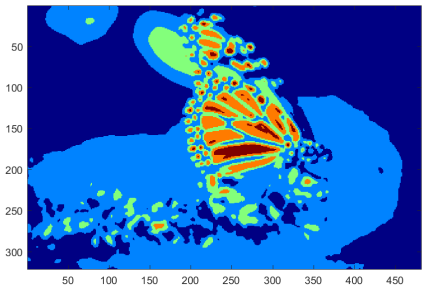
(c) Texture gradient

(d) Texture gradient

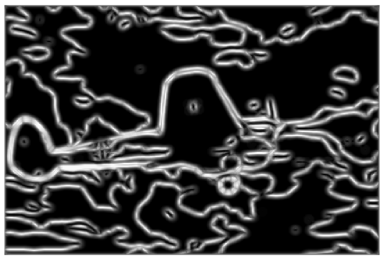
Figure 3: Texton map and the resulting texture gradient



(a) Brightness map



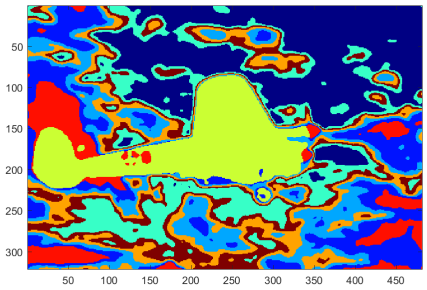
(b) Brightness map



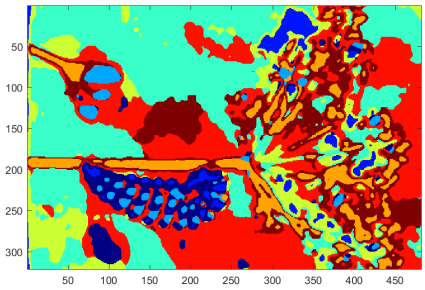
(c) Brightness gradient

(d) Brightness gradient

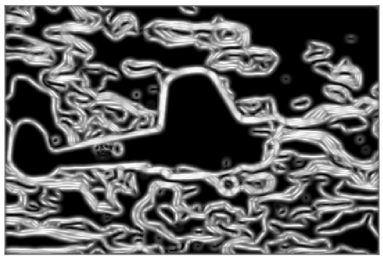
Figure 4: Brightness map and the resulting brightness gradient



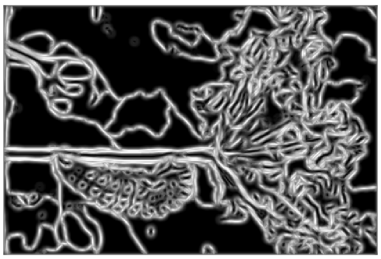
(a) Color map



(b) Color map

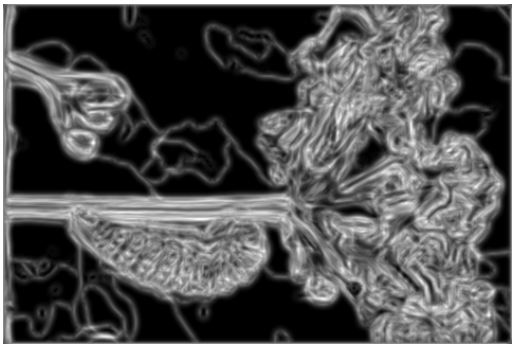


(c) Brightness gradient

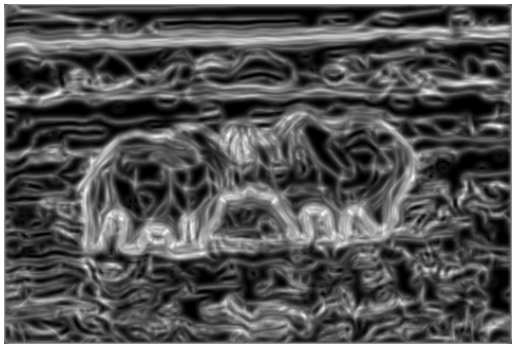


(d) Brightness gradient

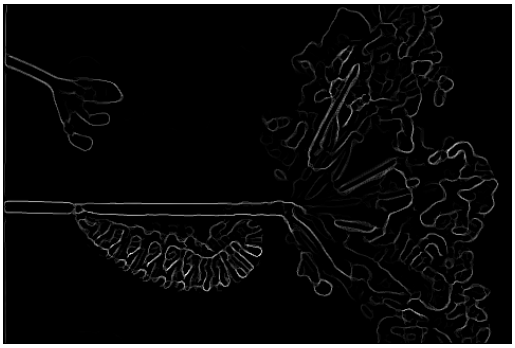
Figure 5: Brightness map and the resulting brightness gradient



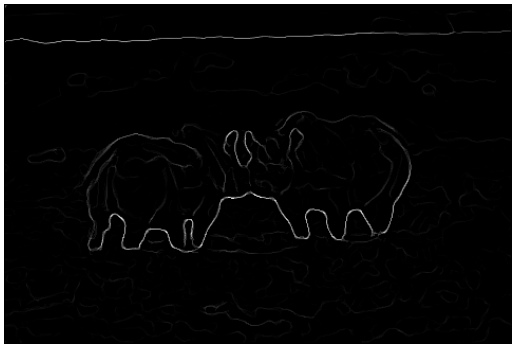
(a) Color map



(b) Color map



(c) Brightness gradient



(d) Brightness gradient

Figure 6: Brightness map and the resulting brightness gradient

In the my edges ended up being very fizzy and with a lot of noise around them. My multiplying my results with the canny base line I get cleaner results but the PR curve does not look much better than with just canny.

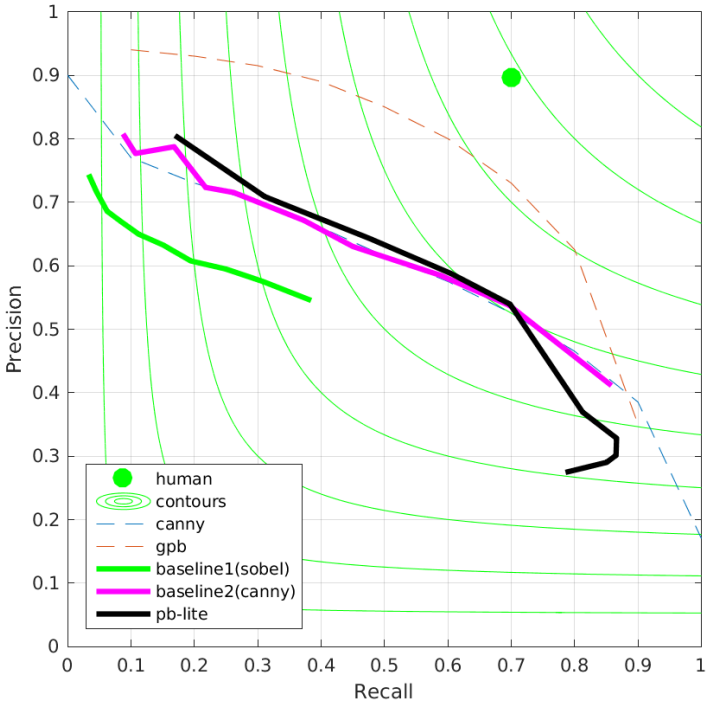


Figure 7: My PB result

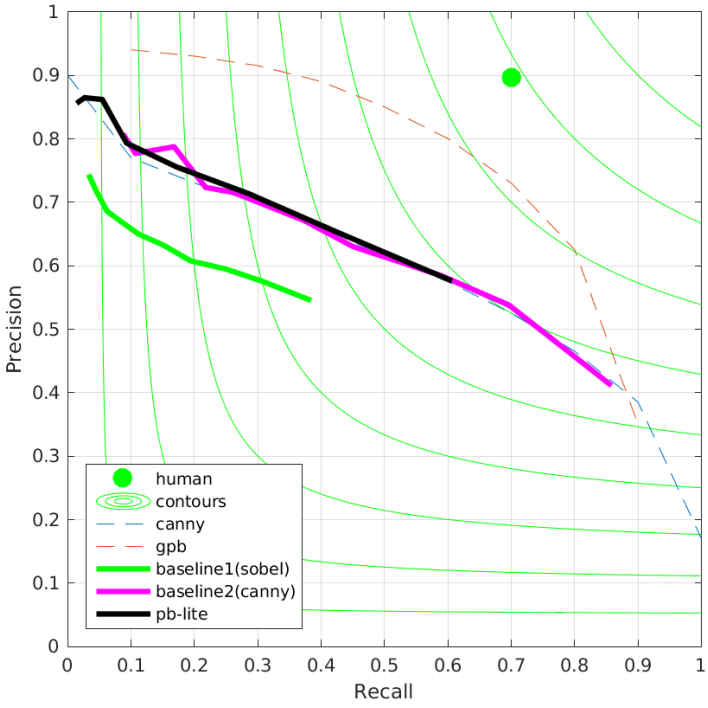


Figure 8: My result multiplied to canny