James Coleman

CS 3150

Homework 2 part 2

September 27th, 2020

My very first step is to write out an outline of what I plan to code in comments. I don’t always inevitably have to add more code that I didn’t think of but it helps me to organize my process and to stay focused on the goal. After that I added some header material, my imports and was ready to code.

I grabbed the image with cv2’s imread and was done with that section of the code, or so I thought. I later had to return to convert the image to gray. Even though the image is black and white imshow still reads it in as a color image and this will cause issues later if not properly taken care of. I think it’s a good habit to do color conversion and dimension extractions right away when acquiring an image. You’ll need the dimensions later anyway. And while we’re at it we might as well display the original:

A picture containing chart

Description automatically generated

Next I built and applied an average filter. I started with 15 by 15 but that was way too blurry in my opinion so I tried a couple different sizes and settled on 7 by 7. Below are 15 by 15 (left) and 7 by 7 (right) averaged images:

A picture containing graphical user interface

Description automatically generated A picture containing bird, device

Description automatically generated

After setting up both my vertical and horizontal Sobel filters I applied them. At this point I decided to try out the scipy convolve2d function and that’s when I discovered that I had not properly converted the original to gray scale. The convolve2d function complained that I was not passing it a 2d array, a little digging and the solution became clear. From that point I built the gradient image manually using the square and sqrt functions from numpy. The gradient image turned out very dark initially until I added vmin and vmax specifications to the parameters of my pyplot.imshow function. I find it very cool that you can see the inner contour of the iris in this image:

A picture containing application

Description automatically generated

Building a ring-shaped filter was quick work. I borrowed a bit of my own code from homework 0 and quickly had a working filter. There was an issue however. Though I tried several different border parameters from both cv2.filter2d and scipy.convolve2d I could not get this odd border issue to go away. Below where two of my best results:

Graphical user interface

Description automatically generated A picture containing graphical user interface

Description automatically generated

I finally decided to just work with what I was getting. So, in order to detect the pupil center, I just limited my scan to roughly the central rectangular area. Once I located that bright point in the middle and recorded its coordinates, I drew a small circle on my gradient image to make sure I was on the right path:

A picture containing application

Description automatically generated

The point turns out to be ever so slightly low and to the right. I suspect this is caused by the fact that several points at the focus of the ring filtered image had the same max intensity and instead of detecting the central one I found either the first or last. I would have to double check my code and do some trouble shooting to know exactly what is happening but no matter. With a little nudge of my focus, more borrowed code from the disc creation section and trying out a couple different radii, I came up with a very accurate boundary for the pupil. I then applied it to our original and here it is:

A picture containing chart

Description automatically generated