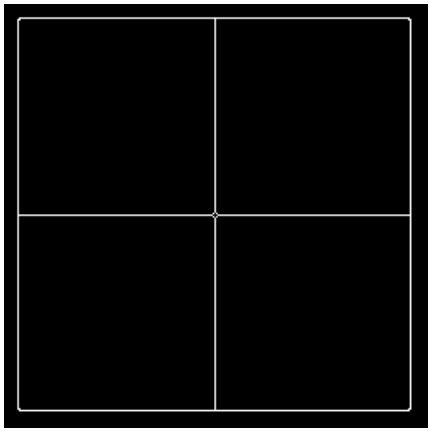
Problem Set 1
CS4495 - Fall2014
Cristina Chu (cchu43)

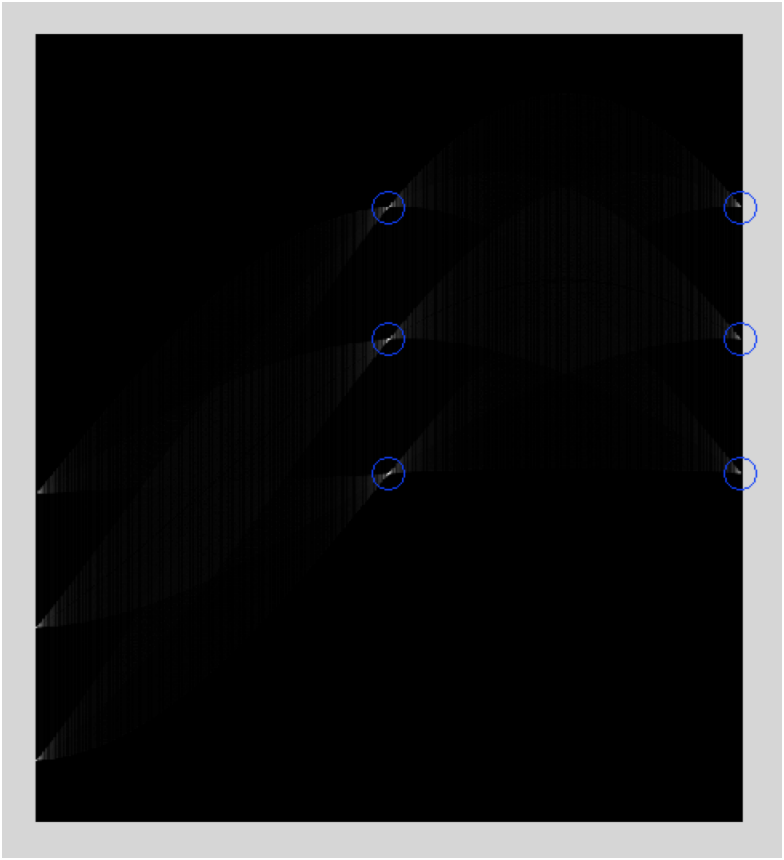
Part 1

a. Edge Image

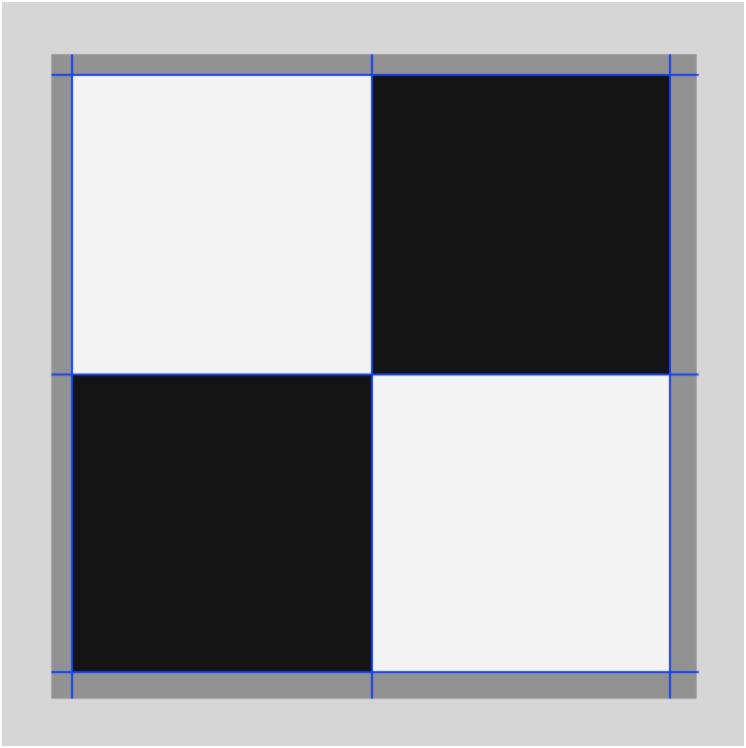


Part 2

Hough accumulator array with peaks labeled



Intensity image with lines drawn

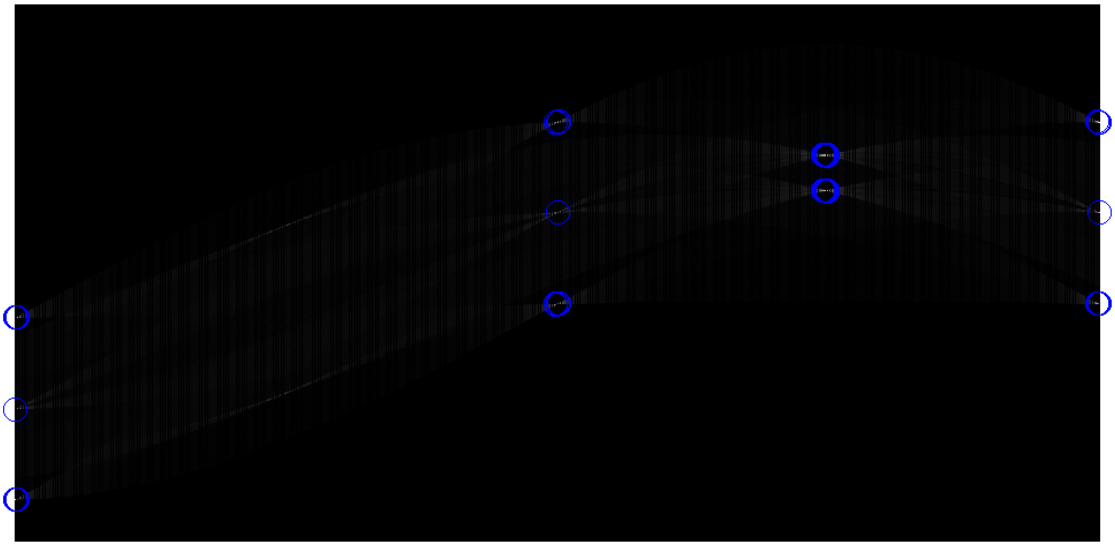


Response: (Describe accumulator bin sizes and why/how picked those)

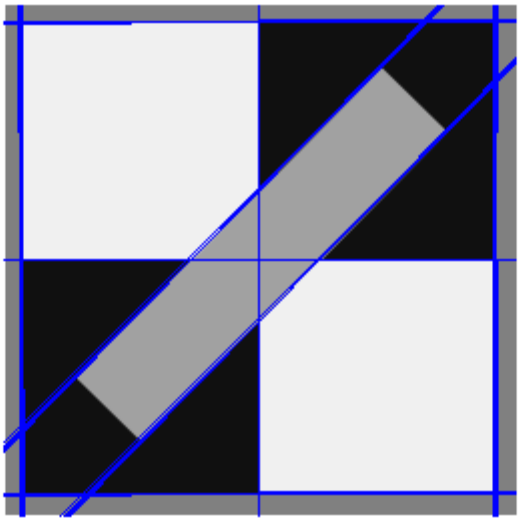
My Hough accumulator array size is 725x629.
I basically started to play around with numbers, simply taking into consideration that I wanted a large number of small bins to be able to detect lines easier/more accurate, as well as numbers big enough to be able to deal with low values of radians for theta. I ended up using the integer value of the diagonal of the input image multiplied by 2 (size of d) and the integer maximum value of theta multiplied by 200 (size of theta).

Part 3

Hough accumulator array image with peaks labeled



Intensity image with lines drawn on them



Response: Describe your accumulator bin size and why/how you picked those.

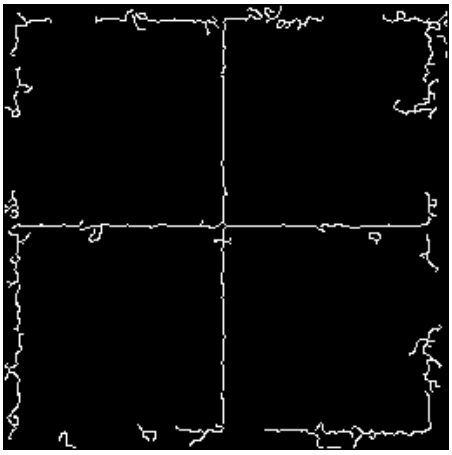
My Hough accumulator array size is 725x1006.
I started by using the same array size of the previous part, but realized that I needed a bigger size for the values of theta since it wouldn't properly detect the diagonal lines. Therefore I ended up using the integer value of the diagonal of the input image multiplied by 2 for the size of d and the integer maximum value of theta multiplied by 320 for the size of theta.

Part 4

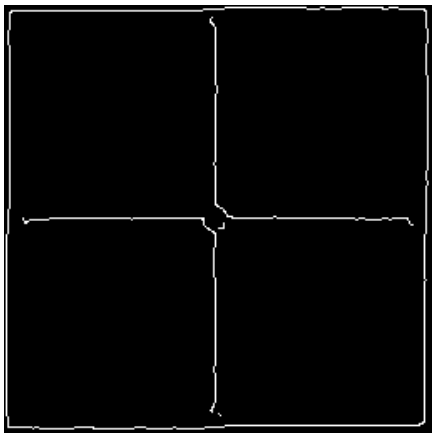
a. Smoothed image

b. Edge Images

Monochrome image version

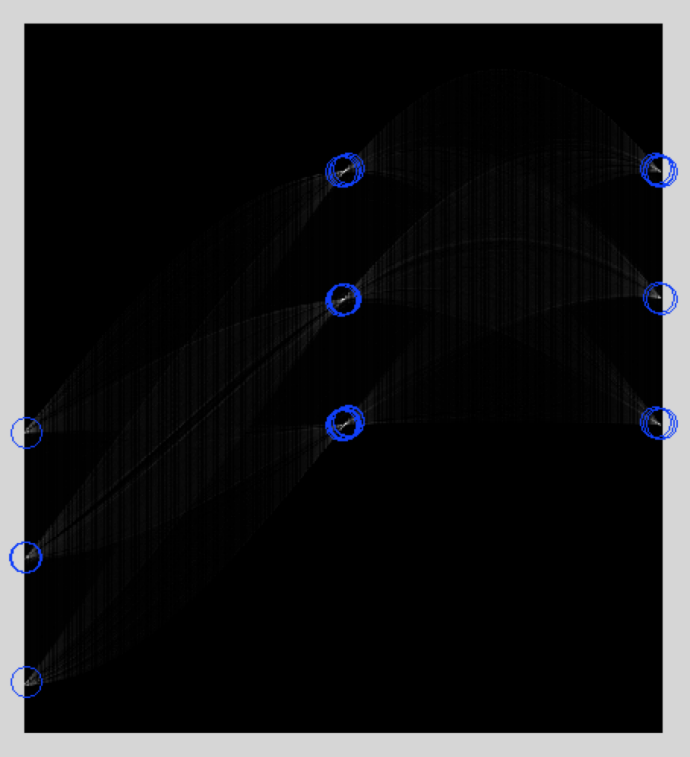


Smoothed image version

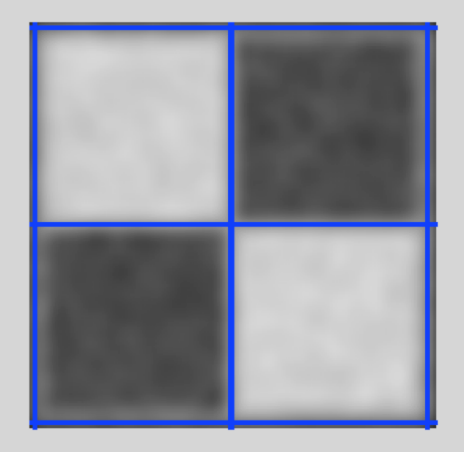


c. Final results

Hough accumulator array image with peaks labeled



Intensity image with lines drawn

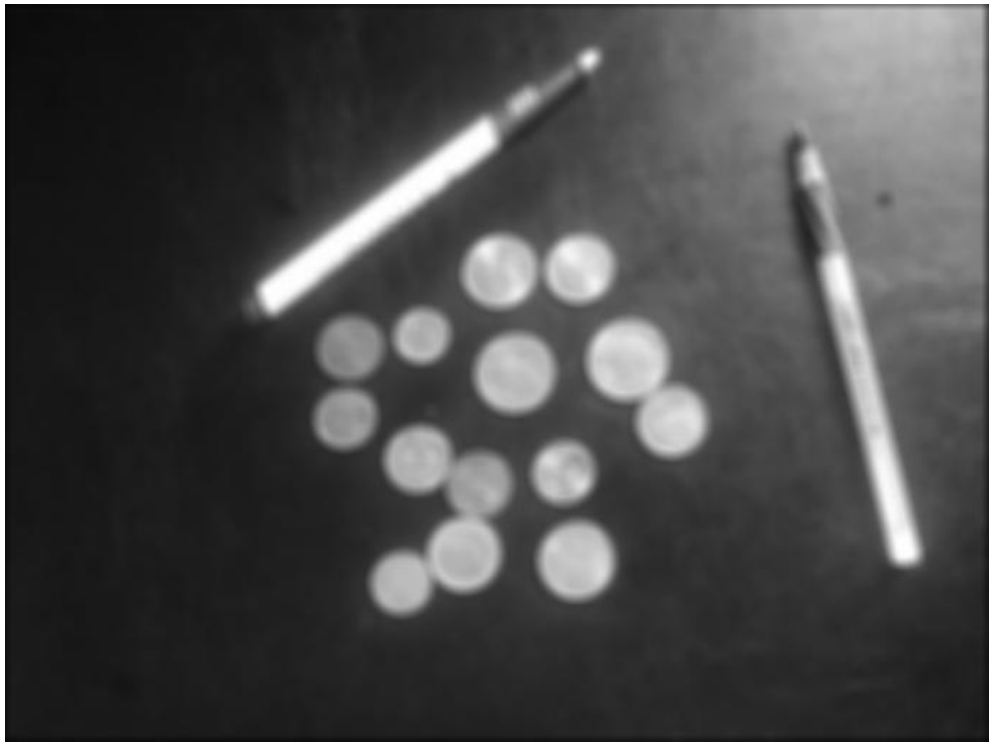


Response: What did you do to get the best result you could?

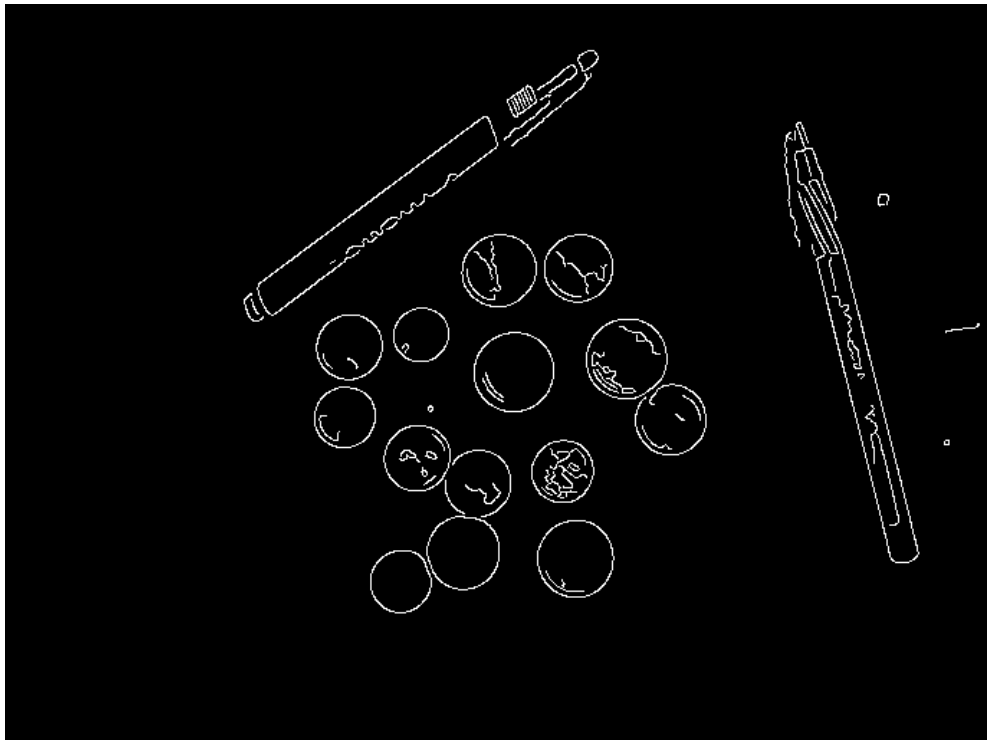
First, I experimented with a lot of different edge detectors to see which one would give me a good edge image. After being satisfied enough with the edge image, I experimented with the threshold value to better detect peaks on the Hough accumulator array.

Part 5

a. Smoothed image

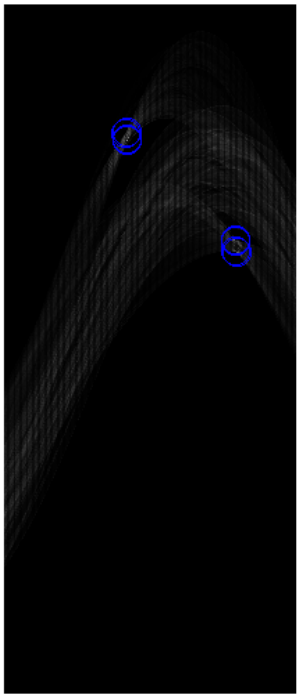


b. Edge image of smoothed image

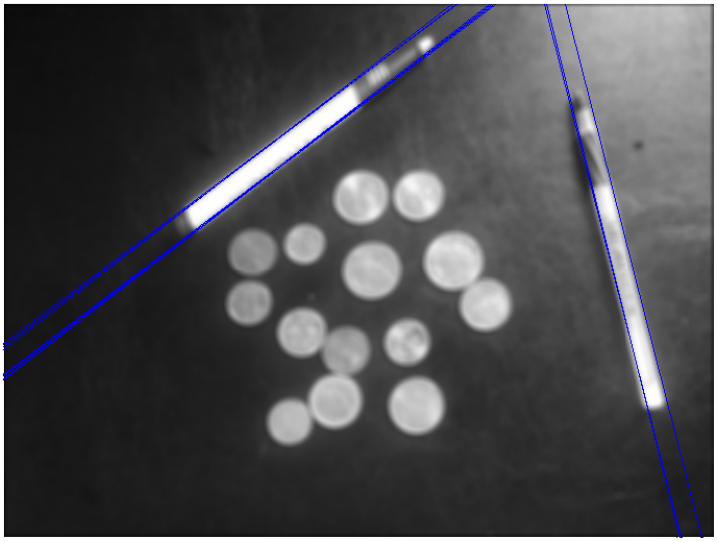


c. Final results

Hough accumulator array image with peaks labeled



Intensity image with lines drawn

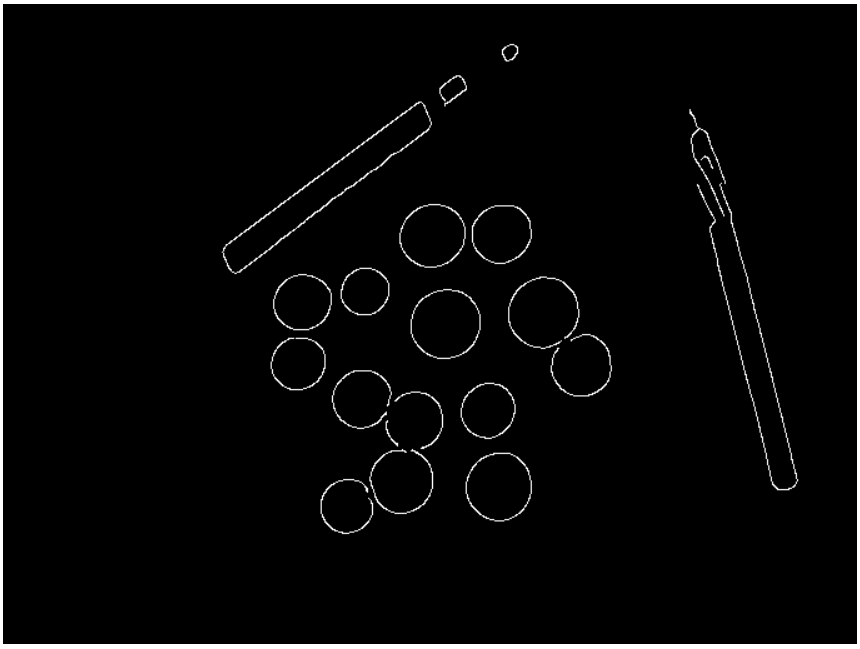


Response: What did you do to get the best result you could?

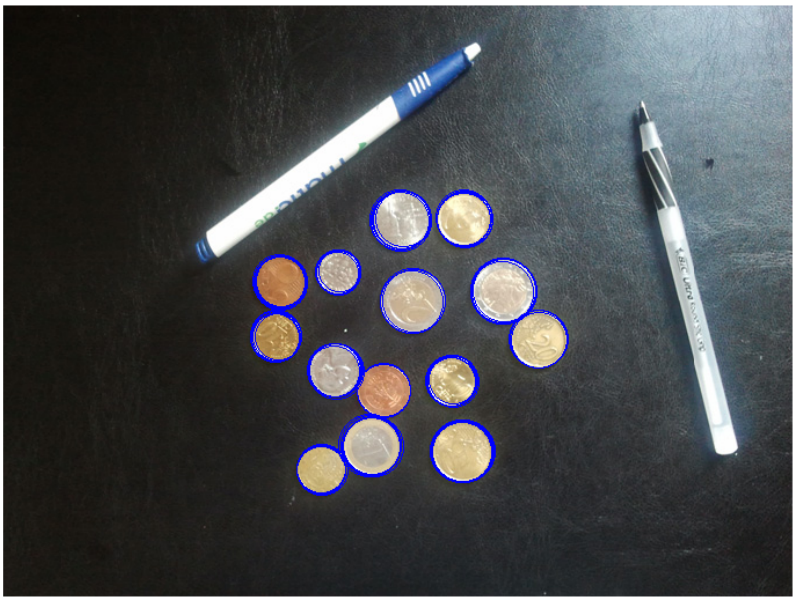
Again, I experimented with edge detectors trying to get an edge image that would give me a somewhat complete edge of the pens. Although I did get lines through the edges of the pens, there a several lines rather than just one which could be explained by some of the edge noise that appears on the edge image. I also played around with the threshold value for the peaks to get less lines.

Part 6

Edge image



Final result image with circles drawn

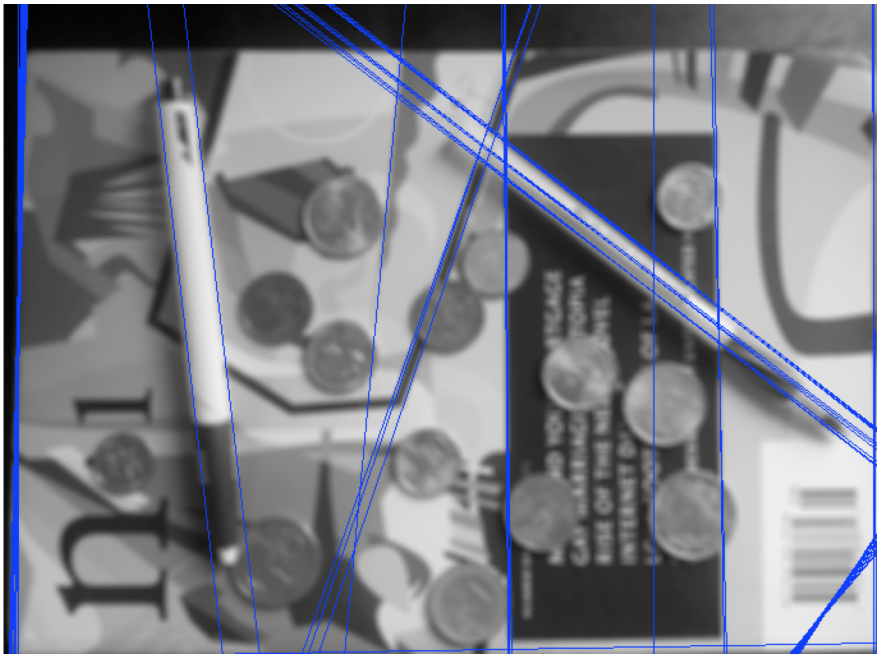


Response: What did you do to find the circles?

The Hough accumulator array needed to be a 3D array instead of a 2D in order to count for the center (x-Value and y-Value) as well as the radius of each circle.

Part 7

a. Smoothed image with lines drawn



b. Response: Problems present in trying to get only boundaries of pens?

The edge image contains many edges from the book where the pens are resting. Some edges in the book are even more prominent that the edges of the pen, so trying to isolate the pens is hard.

c. Smoothed image with lines drawn ONLY for pens

This can be done by finding parallel lines. In other words, lines whose theta only changes by a couple degrees and distance is around 40 pixels afar.
(*was not able to get it to work)*

Part 8

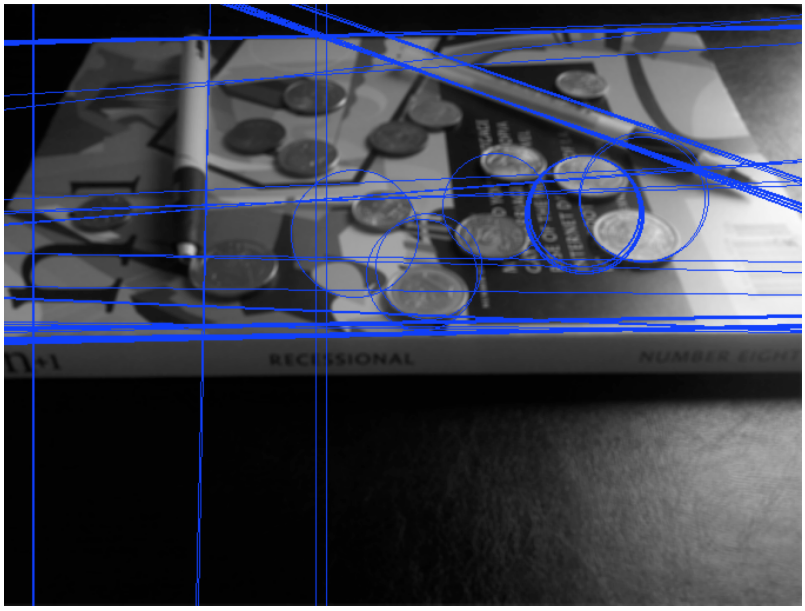
a. Smoothed image with circles drawn

b. Response: (Are there any false alarms? How would/did you get rid of them? Mention where in code they are if done)

There are a couple of false alarms, as well as some edges that are not detected. In order to get rid with some of the false alarms, I lowered my threshold as well as tried to get an edge image with no extra edges (noise) but didn't work too well. Basically, for the threshold part, I simply used a higher threshold value when looking for peaks in the Hough accumulator array. For the edges part, on the other hand, I played around with edge detection functions and the smoothing of the image.

Part 9

a. Image with lines and circles



Can you find lines? The circles?

The pens, although a little hard due to the noise, can be detected. However, there are a lot of lines that shouldn't be there which might come from using a low threshold to try to find the pens. The circles, on the other hand, are not so easy to detect thanks to the busy pattern of the book, as well as the fact that the perspective of the shot makes the coins appear more oval than circles.

b. Response: (What might fix the circle problem?)

The perspective of the picture makes it hard to obtain the correct circles mainly because they appear more oval than perfect circles. Because the Hough transform implemented is targeted for circles, the results are not very good.

In order to fix the problem, the best solution is to implement a Hough transform that is more generalized and does not heavily relies on a specific/certain shape. Basically, a Hough implementation that would be able to detect more oval shapes as well as circles.