Computational Photography

Assignment 5 - Hough Transform
Winter 2024

Introduction

In this assignment you will demonstrate your ability to implement utilize the *Hough Transform* for finidng parameterizable objects from edge pixels. In particulater, you will demonstrate your ability to:

- Generate “fake” edge data.
- Apply hough transform to edge data for various parameterizable shapes.
- Find local maxima in hough transform.
- Generate shapes based on parameterized shapes.

In this assignment, as with all of our assignments, you shouldn’t be using built-in functions that violate the “spirit” of the assignment. For instance, any sort of functions to generate circles or lines, or to compute hough transforms, are forbidden. *However*, since we have already implemented edge detection in a prior assignment, you **may** use a function like *edge* to obtain edge pixels in an image. In general, use your intuition, and when in doubt ask the instructor or TA.

Grading

Theory Questions	10pts
Generating lines and circles	10pts
Hough transform for a line	25pts
Hough transform for a circle	25pts
Single circle detection on real image	30pts
(Extra Credit) Multiple circle detection on real image	10pts
TOTAL	100pts

Table 1: Grading Rubric

Dataset

For this assignment we're going to use two pieces of data:

1. Synthetically generated data
2. A provided grayscale image, *circles1.gif*

1 Theory Questions

For the following questions, show the computations to support your answers.

1. (5pts) Given an image with a width of 200 pixels, and a height of 100 pixels, what is the size of the hough transform accumulator for lines if the step side for each parameter is 1.0 (assume that the angle is in degrees)?

2. (5pts) If the probability of an edge pixel is actually on the object we are attempting to detecting is 0.2, and the desired accuracy of our model is 0.99, how many independent tests must we run $RANSAC$ on, if we are attempting to detect a line?

2 Generate Fake Data

To start off with, let's generate some fake edge data so that we know what the "solution" is.

Create a 400×400 binary image that has two objects on it:

- A line with slope $m = 1$ and y-intersept $b = -100$.
- A circle with center $x = 100, y = 200$ and radius $r = 50$.

You can generate the line by starting varying x from its minimium to maximum value, generating y values along the way according to the formula $y = mx + b$.

You can generate the circle by varying $\theta = [0, 2\pi]$ and generating the (x, y) coordinates according to:

$$x = x_0 + rcos\left(\theta\right)$$

$$y = y_0 + rsin\left(\theta\right)$$

where (x_0, y_0) is the center of the circle and r is its radius.

Note that since the origin of an image's coordinate system is the top-left, its positive y-axis is *down*, and therefore your image is a reflection about the x-axis from what you might imagine it to be.

3 Hough Transform for a Line

Next let's try to find the parameters of that line! Apply the *Hough Transform for a Line* to your binary image. Use the *polar* form, varying the of the parameters θ, r according to the slides, and incrementing them by one for each bin.

NOTE: You CANNOT use the Matlab function *hough* or anything related to do this for you. However, you MAY want to use a function like *ind2sub* to help you get subscript indices from a linear index.

In your report, provide:

- The value of (θ, r) that cooresonds to the maximum value in the Hough Transform.
- The corresponding values for (m, b) where m is the slope and b is the y-intercept. Show the equations/formulas that allow you to compute those from (θ, r).
- A plot of the Hough Transform as an image.

4 Hough Transform for Circle

Next let's try to find that circle! To find all the parameterize your circle, you'll need a 3D Hough Transform (x_0, y_0, r, where x_0 and y_0 are the x and y coordinates of the center of the circle, respectively, and r is its radius).

You may make the following assumptions:

- The circle's center is within the bounds of the image.
- The circle's radius is less than the diagonal length of the image.

Much like with the line, in your report, provide:

- The value of (x_0, y_0, r) that cooresonds to the maximum value in the Hough Transform.
- A plot of the Hough Transform as an image. **However**, since this is a 3D histogram just plot x vs y for the *slice* where $r = rmax$ where $rmax$ is the value of r find in the max bin.

5 Apply to a Real Image

Now let's apply this stuff to a real image!

For the problem we'll use the provide image *circles.gif*. However, this is a grayscale image, not a binary image. In your previous assigment you implemented elements of a Canny Edge Detector. For this one you'll just use Matlab's *edge* function to do this for you. Feel free to play with any parameters of that function, but if you deviate from the defaults, put in your report what parameters you changed.

Once you have your binary image, apply your Hough Circle detection to it. Display as subplots:

- Original image
- Binary image
- Original image with dominant circle superimposed in *red*.

Iin your report, provide:

- The value of (x_0, y_0, r) that cooresonds to the maximum value in the Hough Transform.
- The subplot image described earlier.

6 Extra Credit: Additional Circles

Of course there was more than one circle in that image! For the last part, attempt to identify all the coins in the image (and nothing else). To do this:

1. Apply a threshold to all the bins, such that any bin below the threshold is not considered a potential circle.
2. Apply *3D non-maximum suppression* to the remaining candidates. You **CANNOT** use a function like *findlocalmaxima* or related. You must do so yourself.

In your report provide a subplot like you did in the previous part as well as the parameters of the circles/coins you detected.

Submission

For your submission, upload to Blackboard a single zip file containing:

1. PDF writeup that includes:
   (a) Your answer to the theory question(s).
   (b) For Part 2, your generated binary image.
   (c) For Part 3, the values for θ and r that provide the maximum in your Hough transform, their corresponding values for the slope and y-intercept of that line, and an image of the Hough transform.
   (d) For Part 4, the values for the center and radius of the circle that provides the maximum in your Hough transform as well as an image of a *slice* of the Hough transform where r has its maximum value.
   (e) For Part 5, three images: original image, binary edge image, and original image with detected dominent circle superimposed on it. In addition, provide the parameters of that dominant circle.
   (f) For Part 6, (if applicable) A list of the parameters of the detected circles, the subplot image, similar to in the previous part.
2. A README text file (**not** Word or PDF) that explains:
   (a) Any unique features of your program (if applicable).
   (b) Any instructions on how to run your script to reproduce your results.
3. Your source file(s).

Part I
Answer to Theory Questions

For the following questions, show the computations to support your answers.

1. (5pts) Given an image with a width of 200 pixels, and a height of 100 pixels, what is the size of the hough transform accumulator for lines if the step side for each parameter is 1.0 (assume that the angle is in degrees)?

2. (5pts) If the probability of an edge pixel is actually on the object we are attempting to detecting is 0.2, and the desired accuracy of our model is 0.99, how many independent tests must we run $RANSAC$ on, if we are attempting to detect a line?

Given:

- Image width = 200 pixels
- Image height = 100 pixels
- Step size for each parameter = 1.0

Steps:

1. Calculate the range for θ. θ ranges from 0 to 180 degrees, giving 181 possible values.

2. Calculate the maximum distance $r_{\max}$:

$$r_{\max} = \sqrt{\left(\frac{200}{2}\right)^2 + \left(\frac{100}{2}\right)^2} = \sqrt{10000 + 2500} = \sqrt{12500} = 111.8$$

3. Determine the range for r. r can take values from -112 to 112, yielding 225 possible values.

4. Calculate the size of the Hough transform accumulator:

$$181 \times 225 = 40725$$

Thus, the size of the Hough transform accumulator for lines is 40725. Given the parameters:

- Probability that a pixel is on the object, $p = 0.2$,
- Number of pixels needed for our model, $D = 2$ (for a line),
- Desired probability of success, $P = 0.99$.

We use the RANSAC formula to calculate the number of independent tests (N) required:

$$1 - P = (1 - p^D)^N$$

Solving for N, we get:

$$N = \frac{\log(1 - P)}{\log(1 - p^D)}$$

Substituting the given values into the formula:

$$N = \frac{\log(1 - 0.99)}{\log(1 - (0.2)^2)}$$

After performing the calculation, we find:

$$N \approx 112.811044$$

$$N \approx 113$$

Thus, to achieve a 99% probability of successfully detecting the model, approximately 113 independent tests are required using the RANSAC method.

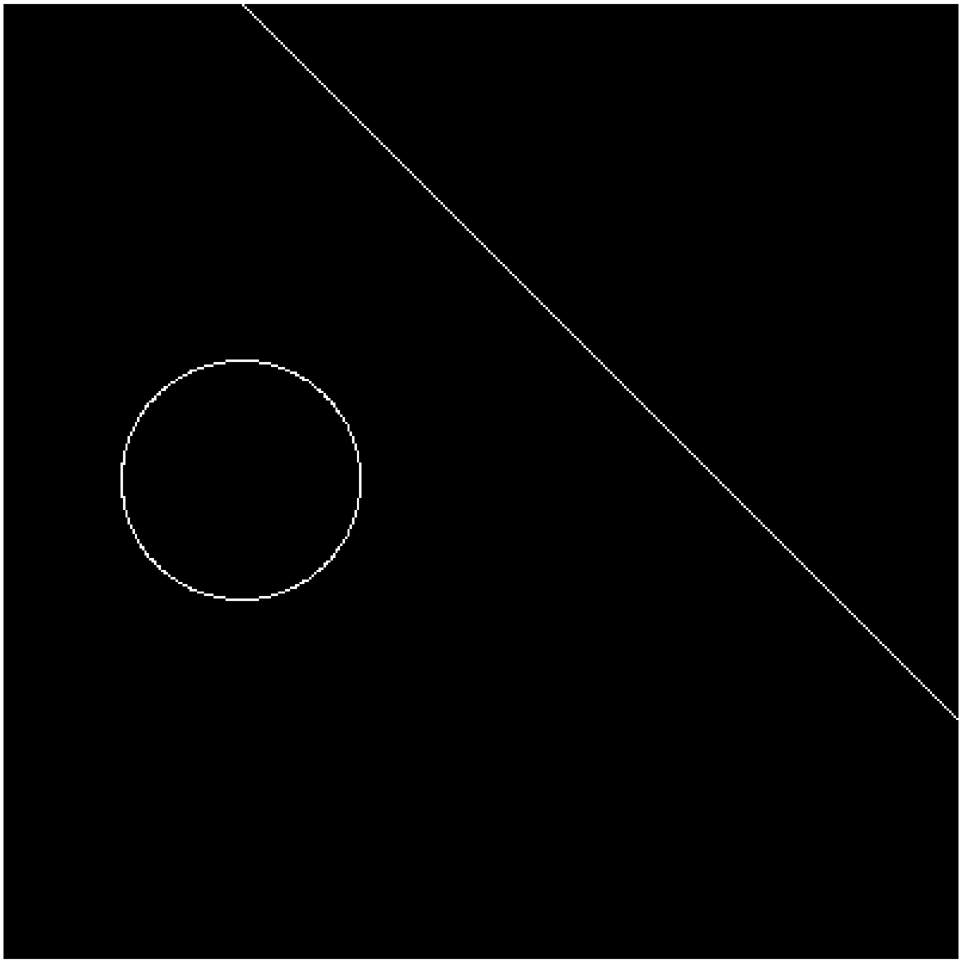

Figure 1: Part 2 - Generated Binary Image.

Part II

Part 2-6 Images

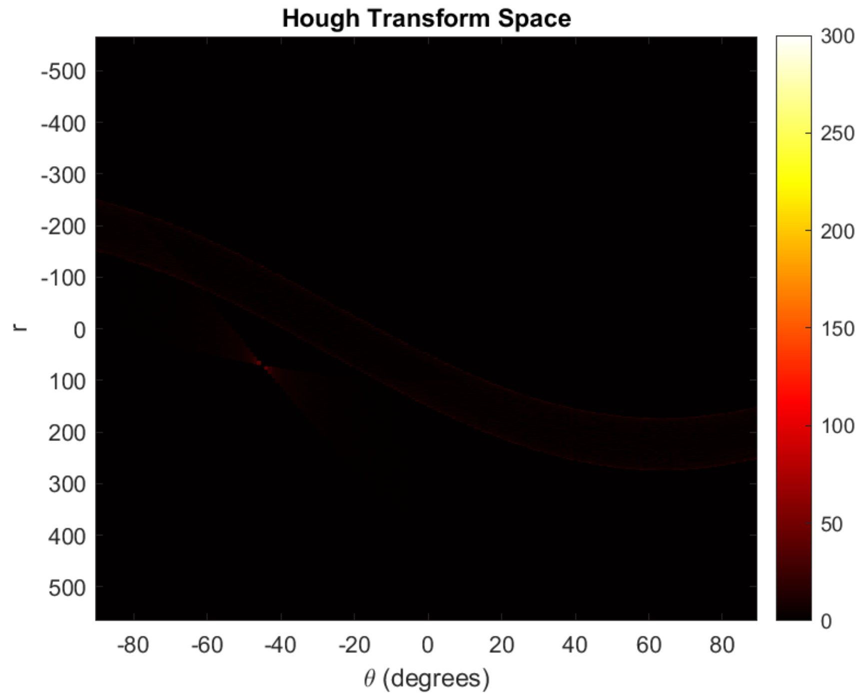

Figure 2: Part3 - Image of the Hough transform.
Detected line: y = -1.04x + -101.22 Detected Line Parameters: Theta (degrees): -136.00 R: 70.31
Slope (m): -1.04 Y-intercept (b): -101.22

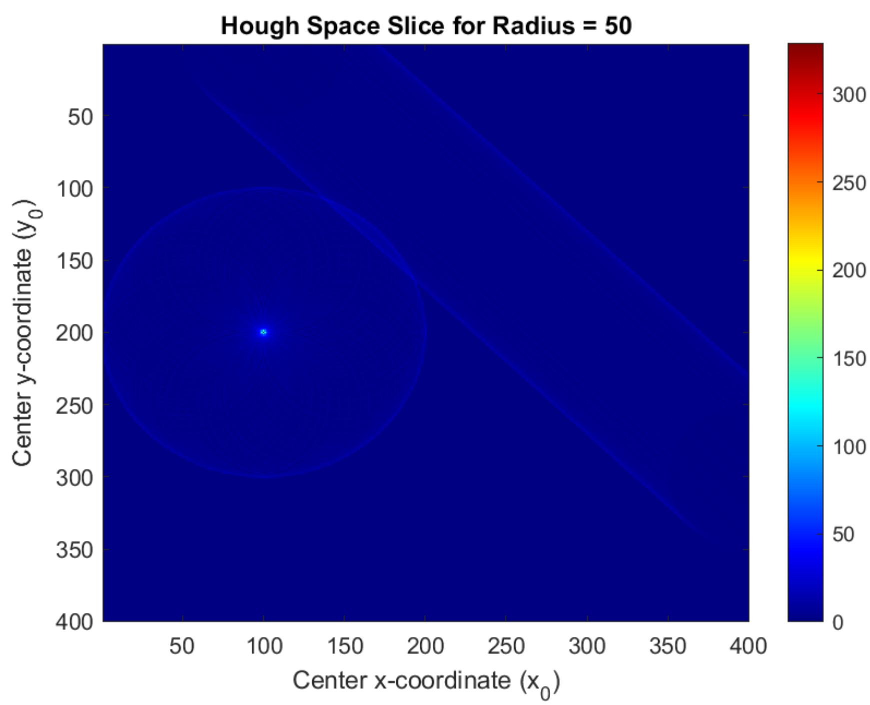

Figure 3: Part 4 - Image of a *slice* of the Hough transform where r has its maximum value.
Detected Circle Parameters: Center: (100, 200) Radius: 50 Maximum value in Hough Space: 329

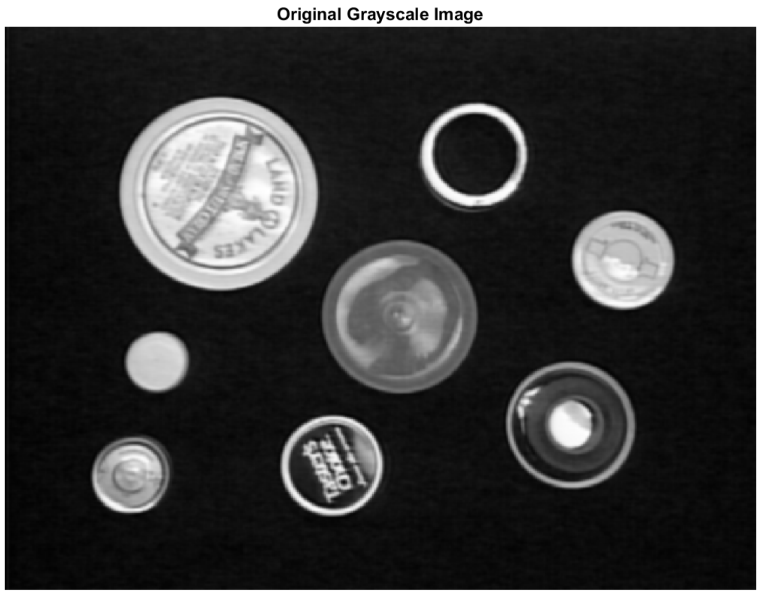

Figure 4: Part 5 - Original Grayscale Image $\sigma = 1$ w. Gaussian Filter

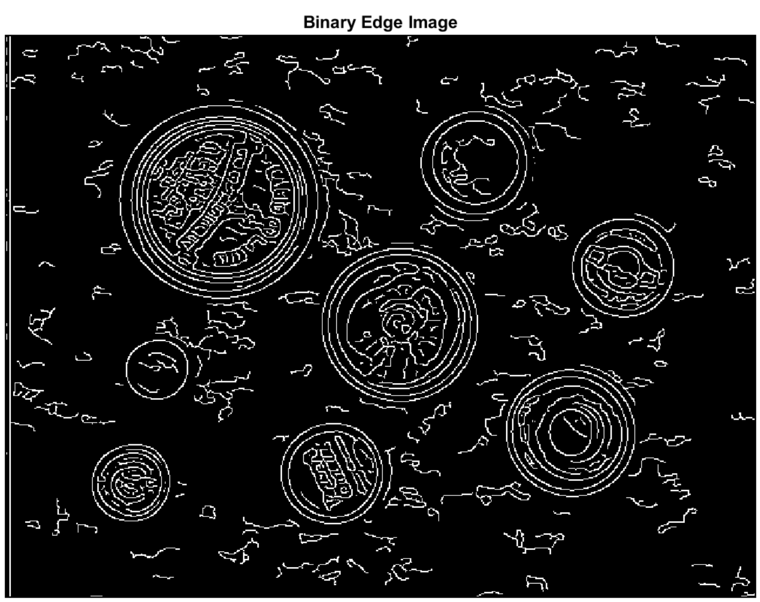

Figure 5: Part 5 - Binary Edge Image $\sigma = 1$ w. Gaussian Filter

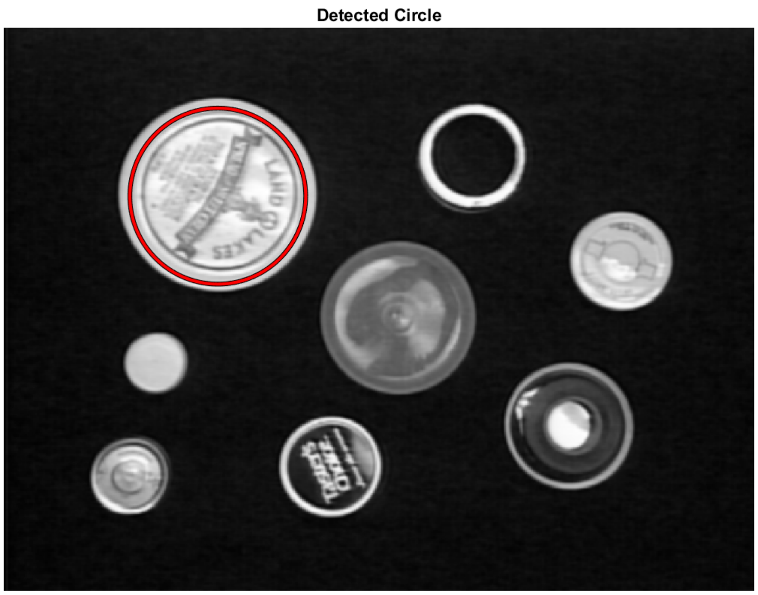

Figure 6: Part 5 - Original image with detected dominant circle superimposed on it

Detected circle parameters:
Center: (183, 144)
Radius: 75
Hough Space maximum value: 223
Hough Space maximum location: (183, 144, 12)
Hough Space size: 640 x 480 x 17

Figure 7: Part 6- Original Grayscale Image $\sigma = 1$ w. Gaussian Filter

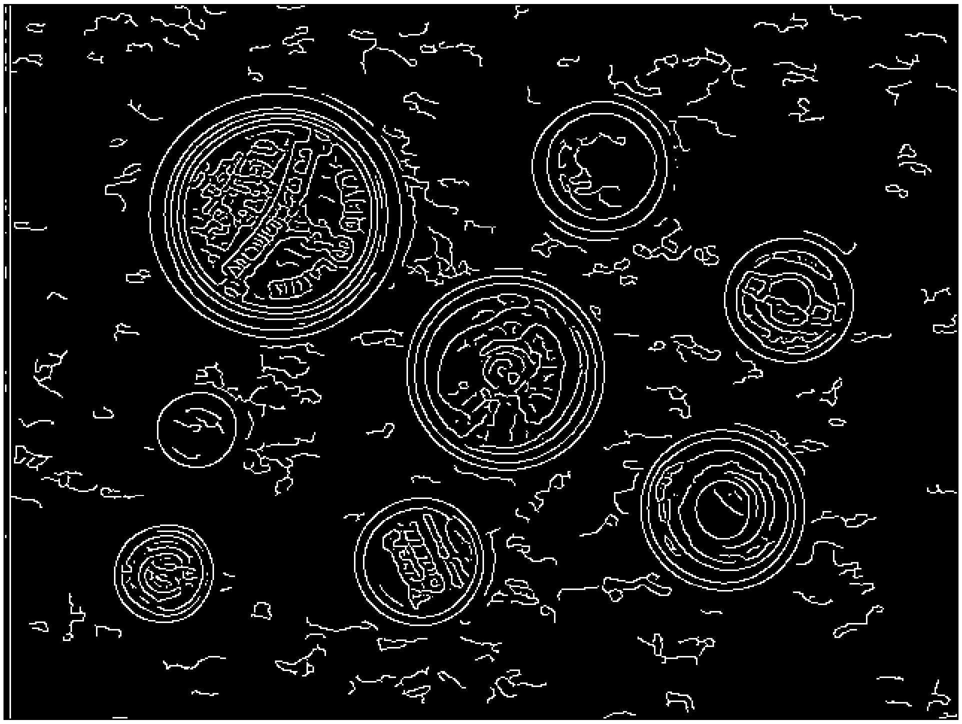

Figure 8: Part 6 - Binary Edge Image $\sigma = 1$ w. Gaussian Filter

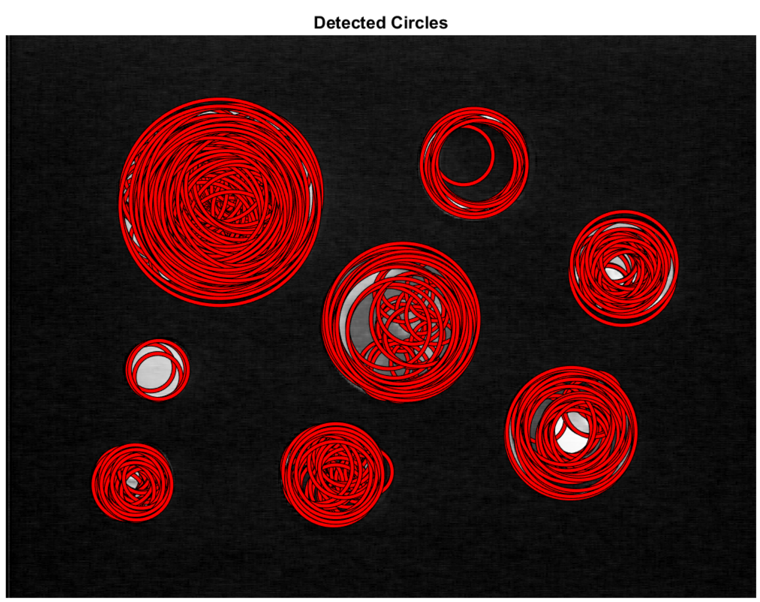

Figure 9: Part 6- All detected circles Threshold = 0.5

Figure 10: Part 6- Original Grayscale Image $\sigma = 1$ w. Gaussian Filter

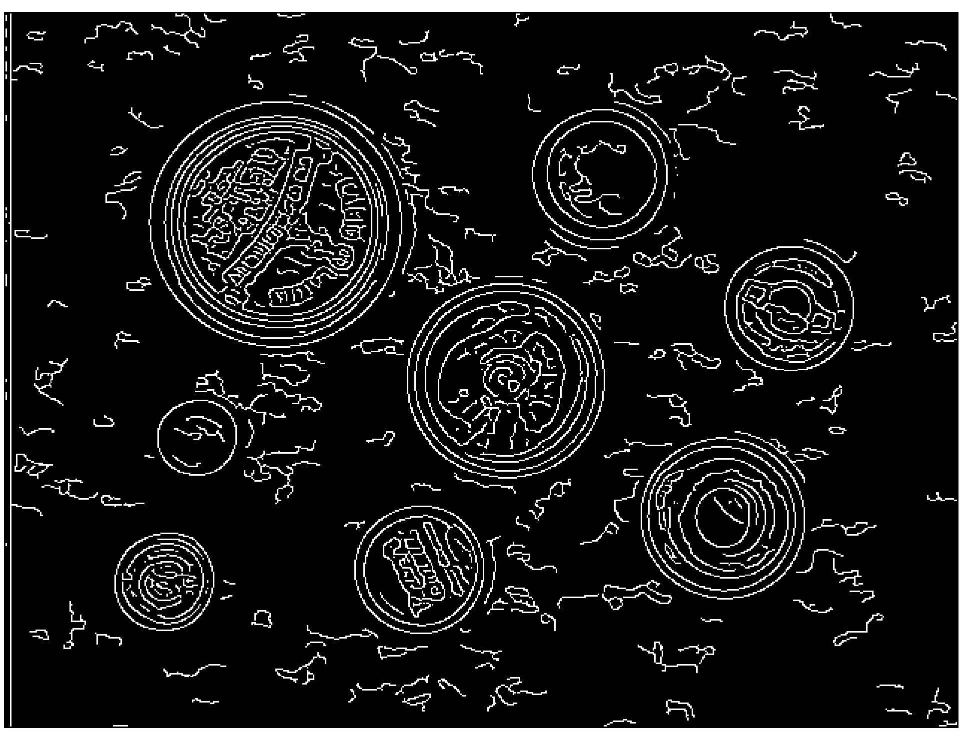

Figure 11: Part 6 - Binary Edge Image $\sigma = 1$ w. Gaussian Filter

Figure 12: Part 6 - All Detected Circles. A threshold value of $0.6 \times \max(\text{HoughSpace})$ was used to identify the most prominent circles in the image. This higher threshold ensures that only the most significant accumulations in the Hough Space, which correspond to strong circle candidates, are considered for detection.

Part III

A list of the parameters of the detected circles Threshold value of $0.6 \times \max(\mathbf{HoughSpace})$

Center_X	Center_Y	Radius
111	387	20
133	134	20
142	107	20
151	169	20
154	141	20
157	155	20
158	126	20
159	146	20
160	160	20
166	140	20
167	129	20
167	185	20
168	129	20
169	127	20
185	105	20
285	375	20
285	376	20
286	376	20
292	394	20
295	363	20
295	364	20
338	250	20
484	339	20
484	340	20
516	199	20
525	198	20
525	199	20
526	200	20
527	200	20
527	201	20
528	201	20
109	382	25
110	386	25
129	286	25
131	286	25
153	104	25
156	150	25
158	125	25
160	152	25
165	124	25
166	145	25
199	182	25
486	340	25
516	200	25
521	196	25

Center_X	Center_Y	Radius
526	210	25
531	191	25
105	384	30
110	384	30
111	380	30
158	112	30
529	196	30
173	110	35
277	375	35
400	109	35
403	109	35
491	341	35
527	201	35
277	373	40
282	372	40
395	110	40
405	107	40
486	340	40
524	199	40
400	108	45
400	111	45
482	337	45
333	247	50
333	248	50
482	341	55
483	339	55
190	138	60
332	244	60
337	248	60
343	245	60
179	146	65
182	142	65
183	139	65
186	139	65
187	142	65
189	147	65
179	141	70
180	149	70
183	144	75
178	142	80
183	140	85