ICV Assignment
Circle Detection

Ryan Wong

August 2015

1 Introduction

In order to apply circle detection on an image, the image needs to be processed using Gaussian filter, Canny Edge Detection and Hough Circle Detection. The process of these three stages will be explained in this report.

2 Processing of Image

2.1 Greyscale

The image given is first greyscaled and normalised so that it fits over the entire grey scale (from 0 to 255).

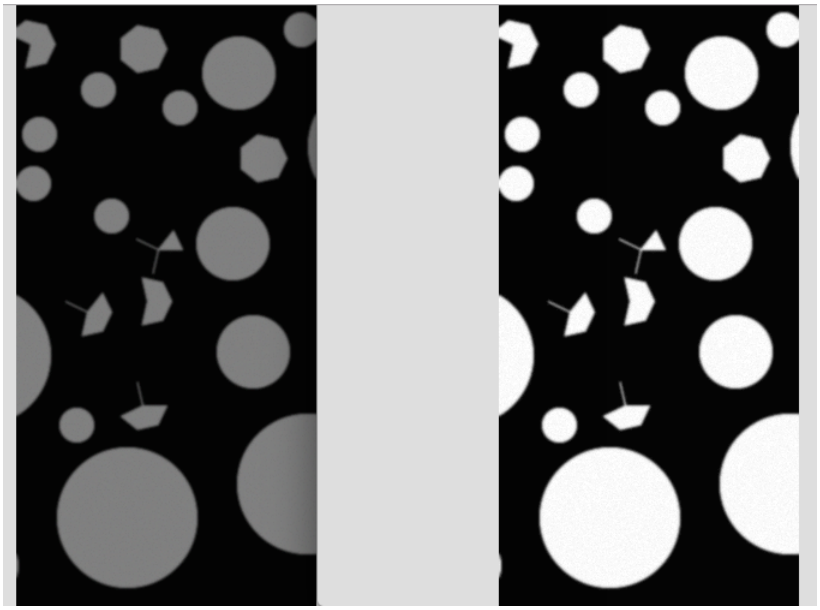

Figure 1: Left: input image, right: greyscale image

2.2 Gaussian Filter

The greyscaled image is the blurred using a Gaussian filter. This is achieved by using the Gaussian function $G(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$. The Gaussian kernel size can be entered by the user which is the parameter for 'gwin' and the sigma is the parameter entered with 'sigma'.
The greater the sigma the greater the blur on the image.

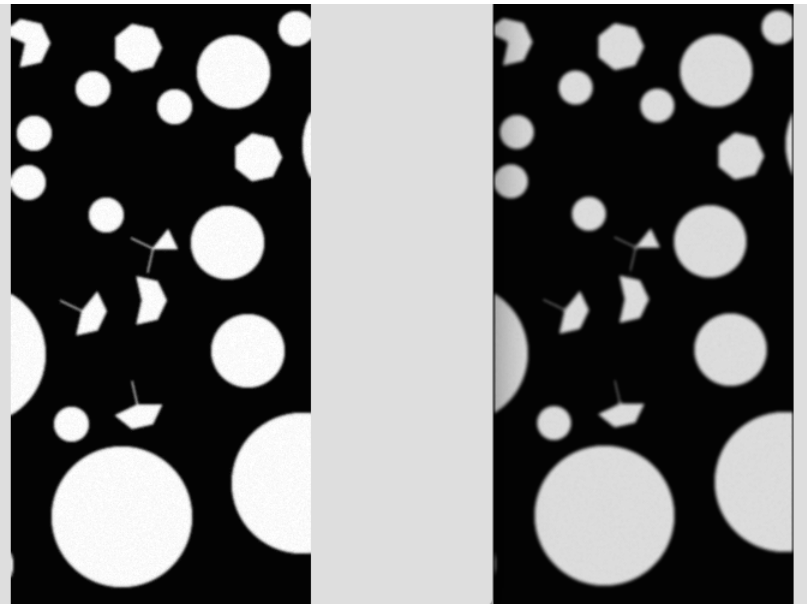

Figure 2: Left: greyscaled image, right: Gaussian image with gwin=5 and sigma = 2

2.3 Canny Edge Detection

2.3.1 Sobel operator

The sobel operators were then applied to find the edge strength by taking the gradient of the image.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figure 3: Sobel Operators

The edge magnitude and direction (angle) were also calculated to be applied in non-maximum suppression. The magnitude was calculated using $G = \sqrt{{G_x}^2 + {G_y}^2}$ and the angle was calculated using $\theta = tan^{-1}(\frac{G_y}{G_x})$. The angle was then categorised into the group it was closest to. The angle groups were 0, 45, 90, 135 degrees.

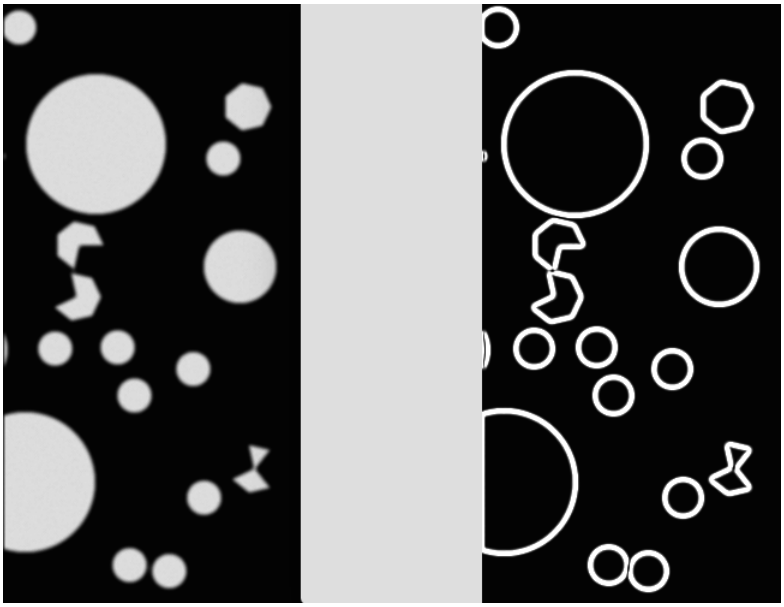

Figure 4: Left: Gaussian Image, Right: Sobel Magnitude Image

2.3.2 Non-Maximum Suppression

After the Sobel Operators were applied, non-maximum suppression was applied using the magnitudes and angles to detect the edges.

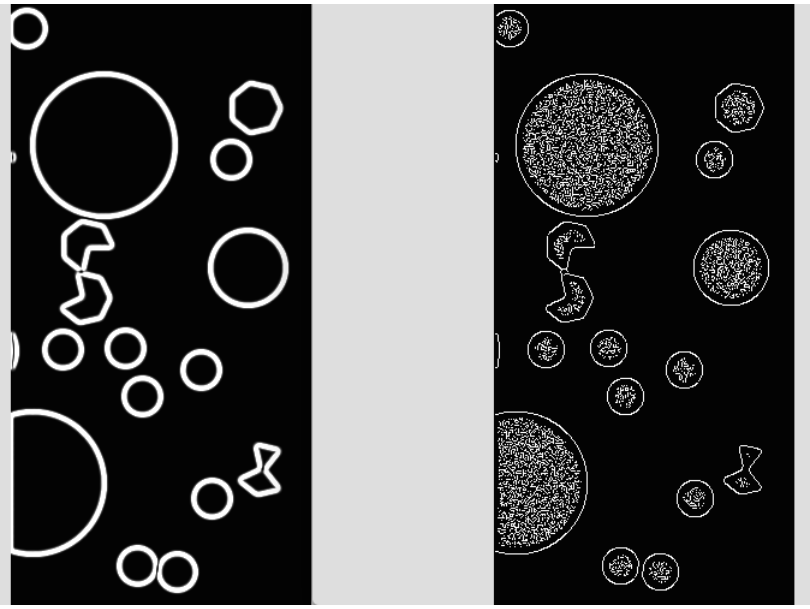

Figure 5: Left: Sobel Magnitude Image, Right: Non-maximum suppression image

2.3.3 Hysteresis

Next a hysteresis was applied to eliminate the noise. The hysteresis used 2 parameters, one for the low threshold and one for the high threshold. The low threshold is the parameter 'lthresh' and 'hthresh' is the high threshold which can be entered when running the program. These values need to be tweaked depending on the image to find good edges.

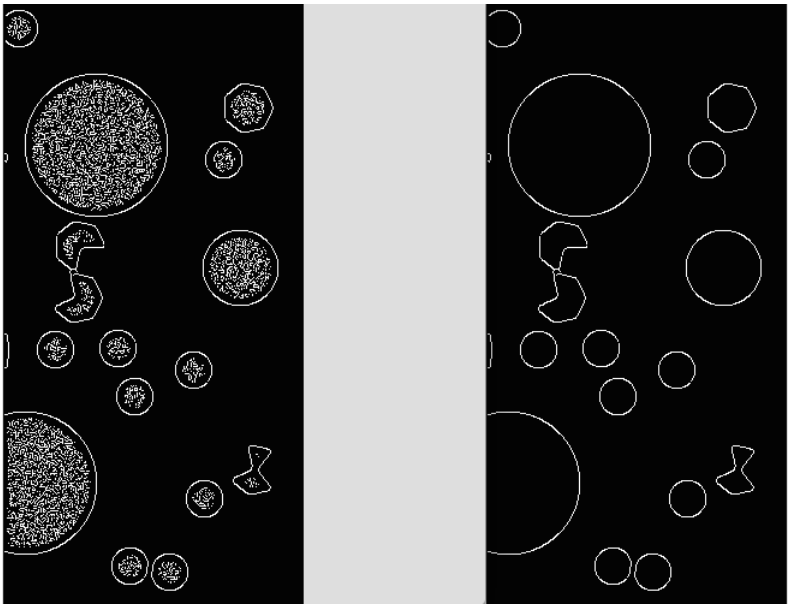

Figure 6: Left: Non-maximum suppression image, Right: Hysteresis Image using low threshold of 100 and high threshold of 200

2.4 Hough Circle Detection

Finally Hough Circle Detection algorithm was applied to find the circles. The Bresenham's circle algorithm was used to create the circles needed for Hough Circle Detection.
The Hough Circle Detection algorithm takes in two parameters. One being the minimal radius length and the other is a accuracy factor for the accuracy of the circles to be found.

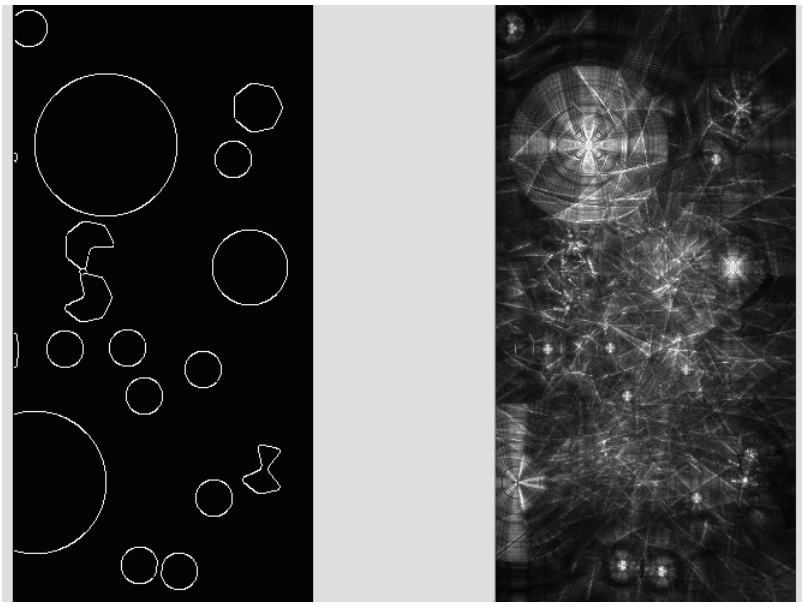

Figure 7: Left: Hysteresis Image, Right: Accumulated value image for all radii greater than 4

From the accumulated values the circles can now be detected using the accuracy factor. Accumulated values that are accepted to be center of circles are the values that are greater than the number of pixels for the

circumference of the radius divided by the accuracy factor (which will be referred to as Hough Threshold). i.e
$Hthresh = accumulator[x][y]/accuracyfactor$
In order to cater for the circles that go over the edge a factor is introduced to scale the accuracy factor. This is achieved by creating a block around a potential circle and then calculate the area of this block. Then divide this value by the area of the block of a full circle. i.e
$factor = \frac{((min(width,x+radius)-max(0,x-radius))*(min(height,y+radius)-max(0,y-radius)))}{(4*radius*radius)}$
This factor is then multiplied by the Hough Threshold. This method will only find circles that have their centers within the image.

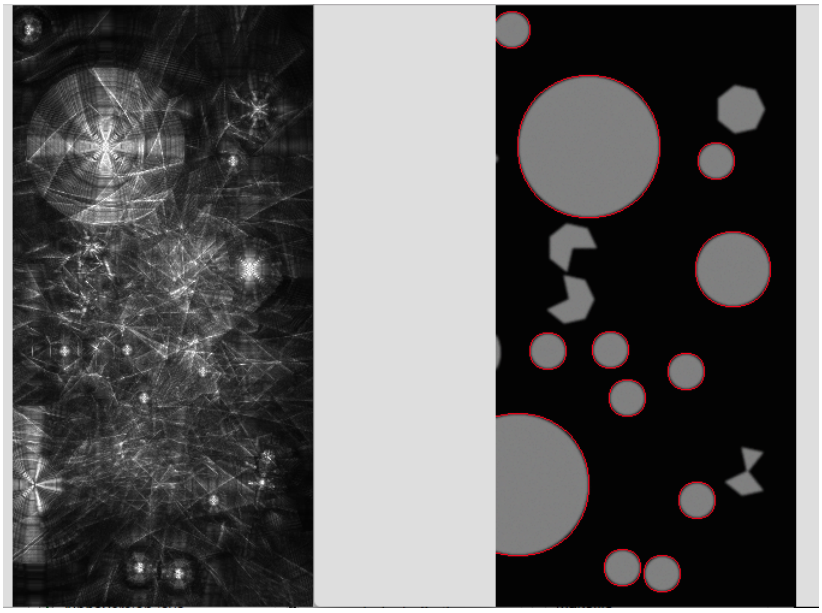

Figure 8: Left: Accumulated Image, Right: Circles Detected using the accuracy factor of 1.6 indicated in red

3 Program Application and Parameters

In order to run the program first compile it with 'make'.
Then to run the program type " **make** file=<file name> gwin=5 sigma=2 lthresh=100 hthresh=200 mrad=5 houghthresh=1.6 **run**"
Depending on the image these parameters need to change.

- **gwin** indicates the window size for the Gaussian Filter. This should preferably be at 5.
- **sigma** indicates the sigma in the Gaussian function. The greater this value the greater the blur.
- **lthresh and hthresh** indicates the thresholds in the hysteresis. The greater the hthresh and lthresh the less edges that are going to be picked up. hthresh should always be greater than lthresh.
- **mrad** indicates minimum circle radius that is being searched for. Preferably should be 5.
- **houghthresh** is the accuracy factor for Hough Circle Detection. The greater the number of the accuracy factor the less accurate the algorithm is. Therefore it will pick up more noisy circles.

For the gif files given, the following parameters in the command have found to be the most accurate **make** file=<file name> gwin=5 sigma=2 lthresh=100 hthresh=200 mrad=5 houghthresh=1.6 **run**

3.1 Tweaking Parameters

The program creates 7 output files for each major stage in circle detection.

- **greyscale**
- **Gaussian filter**
- **Canny magnitude** if this image has too few edges then decrease sigma.
- **non-minimal suppression**
- **hysteresis** if this image has too few edges then decrease lthresh and hthresh.
- **accumulator**
- **circles detected** if there are many small unneeded circles detected increase mrad. If the circles in the image are noisy circles then increase houghthresh.

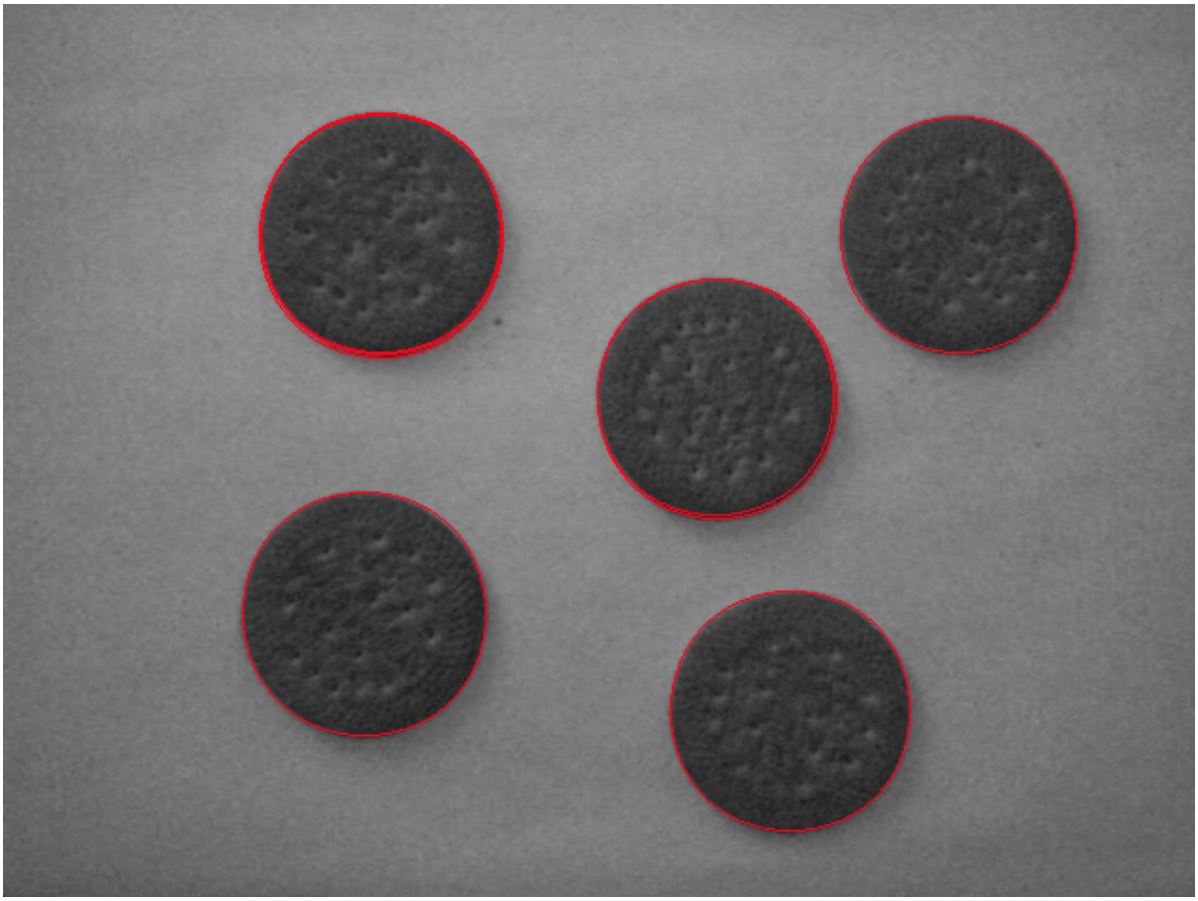

Figure 9: Sample output using the command "make file=circle0.png gwin=5 sigma=2 lthresh=10 hthresh=100 mrad=5 houghthresh=2.4 run"

4 Conclusion

The algorithms applied proved to identify circles but required varied parameters to get accurate circle detection.