CSSE463 Image Recognition

Lab 7 Circle Finder

Zesun Yang and Yuan Zhou

[yangz@rose-hulman.edu](mailto:yangz@rose-hulman.edu)

zhouy4@rose-hulman.edu

January 29, 2018

# Introduction:

In this lab, we experienced and practiced Hough transform, and utilize the idea of Hough transform to perform detection of circular objects with arbitrary radii in grayscale images. Overall our circle finder was very successful. It identified and marked most of the circular objects in images.

# Experimental procedures:

We first read the image and converted them to edge images using the MATLAB edge function and set the parameter be “canny” because it gave a clearer result than ‘Roberts’ and ‘Sobel’ edge detector we wrote in lab 3.

Although we noticed that there was a MATLAB function called imfindcircle which could return the centers and the radius given an image and a radius range, however we believe it is best for us to develop our own algorithm on circle finder and obtain better understanding Hough transform.

In our circle finder, we wrote a nested for loop so that for every row, column, angle (from 1 to 360) and radius number (estimated from 15 to 70), we calculate the x and y position, and then check if the point is part of the circular edge. If the point is a part of some circular edge, and it is within the image range, then we detected it to be a circular edge. The runtime of the nested for loops are slow, but it checked every point in the image and found all candidates of circular edges.

These intersection points characterize the straight line-segments of the original image. Then we tried different values for the threshold, and eventually set the threshold value to 150 to separate out the regions we are interested in, based on the different intensities in the foreground and background.

Then we used the MATLAB function ind2sub to find the x and y positions of the center. After that, we used “rectangle” function to outline circular edges we found in images.

We have include our result images on next page.

# Results:

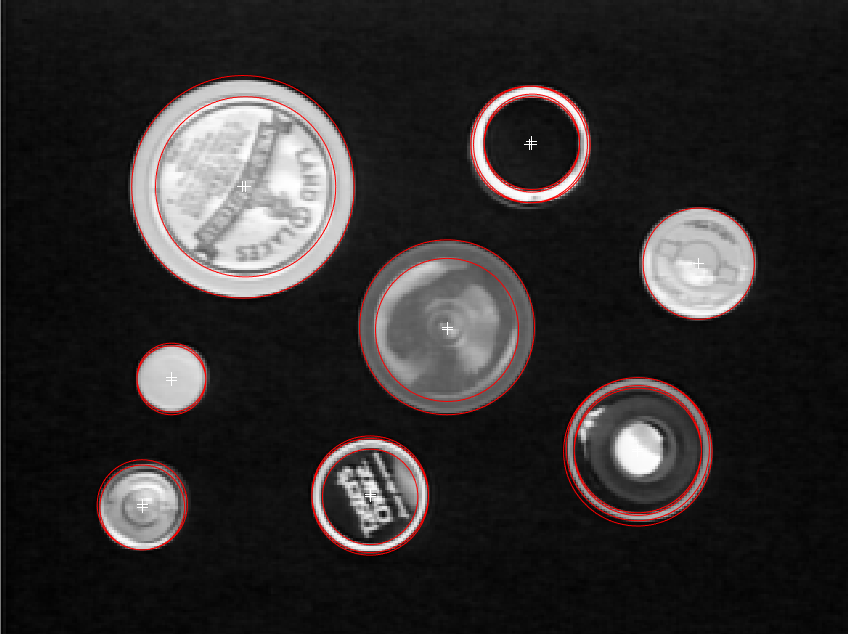
 Figure 1. c1.tiff mask in parameter space Figure 2. c1.tiff result

Figure 3. c2.tiff mask in parameter space Figure 4. C2.tiff result

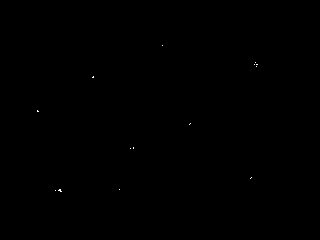
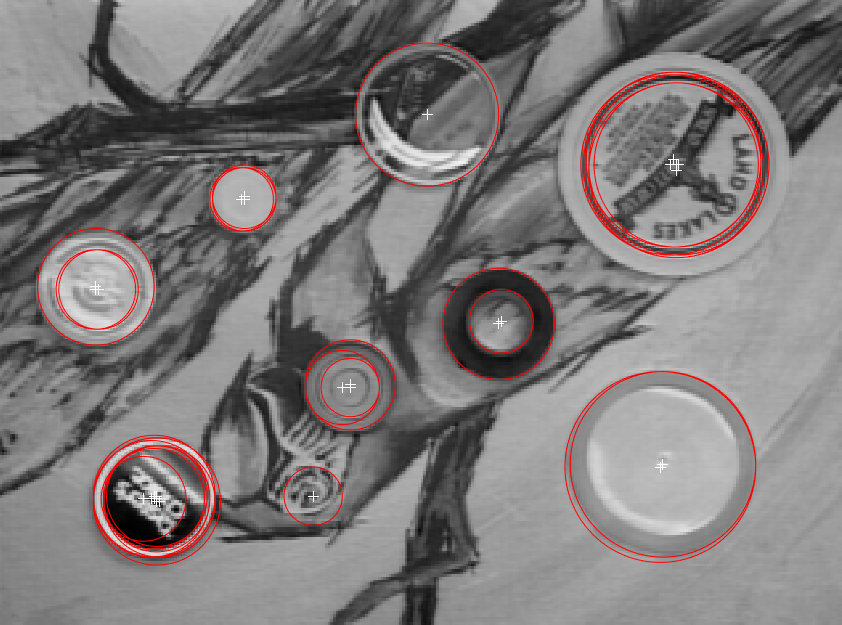


Figure 5. c3.tiff mask in parameter space Figure 6. C3.tiff result

Parameter settings:

Estimated minimum radius:15 Estimated maximum radius: 70 Threshold:190

Peak threshold:0.7

Statistics:

|  |  |  |
| --- | --- | --- |
|  | True Positive Rate | Precision |
| C1.tiff result | 87.5% | 58.3% |
| C2.tiff result | 100% | 66.7% |
| C3.tiff result | 100% | 62.8% |

Table 1. Result Image statistics

C1: TPR:7/8=87.5% Precision:7/12=58.3% number of false positive: 4

C2: TPR:8/8=100% Precision:8/12=66.7% number of false positive: 4

C3: TPR:9/9=100% Precision: 9/14=62.8% number of false positive: 5

# Discussion:

Overall our circle detector is fairly successful on identifying circular objects. For c1.tiff , it has a true positive rate of 87.5% (because of failure to identify one of the circle which is overlaid by two other circular objects). C2.tiff and C3.tiff both have true positive rates of 100% since it correctly identified and marked all objects with circular edges. However, our circle detector is not good at identifying internal circular edges and external circular edges, and that caused our precision to be low. For example, in c1.tiff, our circle detector falsely identified the internal edge of a coin to be an independent circular object. In addition, in c3.tiff, our circle identified the one of the bird’s eye in the background as an independent object as well.

One way to have our detector to not mark internal edges is to increase the threshold, however, by doing so we will lose some objects with lighter edges (intensity) and have a lot of noises around them. This is a tradeoff and we decided to not increasing our threshold and hold our true positive high. We were thinking our detector missed some circles in c1.tiff is because the circles are covered by other objects and not having continuous circular edges, thus confused our detector. In c2.tiff and c3.tiff, there is no circles covered by other objects, and they were all correctly identified. In c3.tiff, the bird’s eye is in fact a circle, and we cannot avoid identifying the eye as a circle even though it is a part of the background.