Automatic Coin detection, diameter recognition and coin counting

Sree Nirmillo Biswash Tushar
Department of Electrical and Computer Engineering, Texas State University,
San Marcos, TX-78666, United States of America

***Abstract*-Coin detection and diameter recognition is a good way to determine the value of the coin and counting the coin along with this can help to determine the bill. Currently, most of the coin detection method follow Hough circle transformation. In Hough circle transformation, minimum radius, maximum radius, and minimum distance between the center point of the circles are used to suppress redundant circle and find optimum circle for the coin. Minimum distance parameter will give inaccurate circle when none of the circles are perfect. Therefore, when multiple circles are detected for a single coin in the range of the defined radius, instead one keeping one and removing others, all the circles' centers and radius have been averaged since it is difficult to say which circle is close to the coin. This method has provided good results for the coin detection, diameter prediction and coin counting. Lastly, a graphical user interface has been developed to tune the minimum radius and maximum radius parameter.**

Keywords –Spiral Antenna, Substrate, Excitation

I.INTRODUCTION

Bill estimation through image processing and computer vision is a research that has been studied to automate the process. Computer vision-based coin detection and the diameter calculation automates the process of billing and saves a lot of human effort of time. Generally, most of the coin are detected are recognizing the coin as a circle by [1] [2]. Hough Circle based circle detection is a tricky task since it is a rigorous search algorithm. Often, it ends up with multiple circles for a single coin because of a lot of edges present in the coin's body. Therefore, a circle suppression algorithm is necessary even after proving the minimum radius and maximum radius value.

In this project, instead of using default minimum distance between the circle as a circle suppression method, I have averaged all the circles detected for a single coin. The reason was that it is difficult to say which circle among the multiple circles for a single coin resembles the coin mostly and the averaging method can lessen the error of all false circle detection. The method provides satisfactory result for different scenarios such as densely populated coin, low density coin and overlapping coin. After the coin detection, diameter of the coin that has been found from Hough circle detection has been shown in the image with the number of coins detected. Lastly, a GUI has been developed to fine tune the radius search parameter. The rest of the paper is arranged like this: Section (II) discusses necessary preliminary concepts that has been to develop the methodology, Section (III) illustrates the methodology of the proposed work, Section (IV) shows the result found implementing the proposed method, Section (V) proposes some suggestion for future researcher and lastly, Section (IV) concludes with some observations.

II.BACKGROUND

The coin detection task can be divided into 5 steps:

(1) Low pass filtering of the image with gaussian blur
(2) Initial Circle detection with Hough Circle Detection on blurred image
(3) Circle suppression for Efficient detection

A. Gaussian blur Filter:

Generally, a coin has a lot of edges in its body. Therefore, to detect only the edge at the circumference, edge at the body should be blurred. In this project, gaussian blur filter from open cv [3] of kernel size (11,11) has been used to do the task. SigmaX and SigmaY has been kept 1. All those values have been determined though experimentation.

B. Hough Circle Detection:

Hough circle detection is a standard method for circle detection in image processing [4]. In Hough circle detection. canny edge filter is used to determine the edges and a rigorous search tries to fit a circle of radius r with center at the edges found. For an instance, a circle with center (x_0, y_0) and radius r can be expressed as equation (1):

$$(x - x_0)^2 + (x - x_0)^2 = r^2 \quad (1)$$

The algorithm will create circle with parameter (x_0, y_0, r) where x_0,y_0 are the edge point found by canny edge detector. Then an accumulator will give a score to every intersected

978-1-5386-9111-3/19/$31.00 ©2019 IEEE

point as the number of circles passed through the point. Then, a threshold will determine will either the point a circle or not [2]. This process will be continued for radius of different size.

C. Circle suppression for Efficient detection:

In this process, multiple circles are detected for a single coin. The reason behind that the algorithm detects circles by maximum voting and the presence of unnecessary edges may cause to detect some fake circles. In order to tackle this issue, a circle suppression algorithm is necessary to suppress multiple for same coin. In Hough Circle detection algorithm, minimum distance between circles works as a suppressor. But, in this project, I have developed an algorithm to get optimum circle.

III.METHODOLOGY

Coin detection, diameter demonstration and coin counting task can be assembled as a process that has been discussed in Section (II). The methodology that has been followed is shown as a flow chart in figure 1.

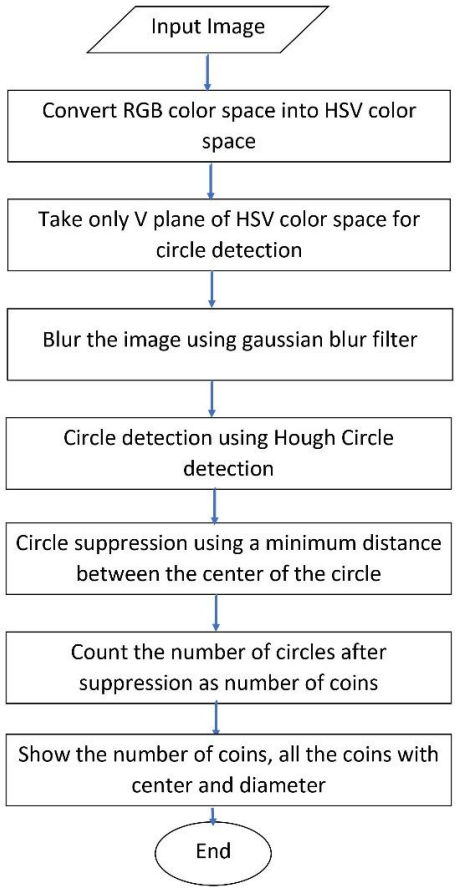


Figure 1: Flow chart for coin detection and coin counting

Firstly, the input image is converted to HSV color space since V plane of HSV color space shows good characteristics while edge detection [1]. Before circle detection, image needs to be blurred by gaussian filter. Circle suppression provides better coin detection capability to the model. Eventually, the output of the system will be image with detected circle with diameter denoted and the total number of coins detected.

In this project, a very simple circle suppression algorithm is used in this project instead of minimum distance parameter of the Hough circle detection [5]. The algorithm relies on the fact that when a coin is detected by multiple circles, those circles' center will be very close to each other. Therefore, if we group all the centers that are less than some threshold apart and average all the centers and diameters for the same group of circles, there will be lesser possibility detecting multiple circles for a single coin. In this project, this approach has helped to suppress a lot of redundant circle. The algorithm is shown as a flow chart in figure 2.

As it is seen from figure 2, first an array is created with number from 0 to N-1 to avoid doing same calculation for all the neighbors. For an example, when a circle, x finds it neighbor – say at k, l, m index – the iterator needs to avoid doing same calculation for value k, l and m. This can be done by checking a condition before proceeding to the calculation. The condition can be formed by replacing all the neighbors' value with their host (i in this case) which will make all neighbors index and value unequal and then check the condition for the circles either their index and value equal or not. Here is an example to clarify what has been said. Consider, 5 circles have been detected initially. So, A will be array array of [0,1,2,3,4]. Now, circle at 0 finds its neighbor at index 2,3 and the value of the A will be updated as A = [0,1,0,0,4]. Now the neighbor at index 2,3 has a value 0 and therefore, the condition is not met for iterator at index 2 and 3. And therefore, calculation at index 2 and 3 will be avoided. These neighbors will be determined by checking either the distance between the center of two circles is less than some predefined threshold or not. This process will make some clusters with circles with their neighbors and the final circles will be average of the centers and diameter of all circles in an individual cluster.

IV.RESULT AND DISCUSSION

Generally, grayscale image is used for edge detection. But, in [1], V plane of HSV color space was suggested to use for circle detection. Figures 3(a), 3(b) and 3(c) show the RGB image, corresponding grayscale image and V plane of the image in grayscale colormap respectively [6]. As it is seen that the edge is clearer in V plane, edge detection by V plane should outperform edge detection by grayscale image. Therefore, I have followed [1] and used V plane of HSV color space for rest of the work.

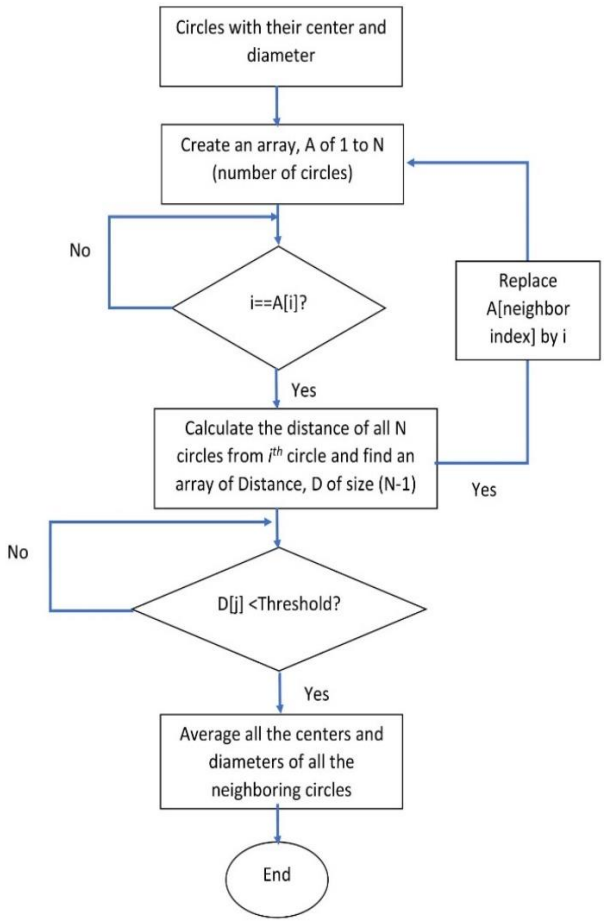


Figure 2: Flow chart suppressing the redundant circles

Then, a gaussian blur filter of kernel size (11,11) and σ_x and σ_y with value 1 and 1 respectively is used for blurring. The blurring effect is shown in figure 3(d). Then, a Hough circle detection algorithm searches circle with the predefined parameter and find appropriate circle. Hough circle detection has been performed by OpenCV. In OpenCV, there are total 7 parameters in Hough circle detection algorithm.

Method: This is the method for circle detection using edge information. Gradient method has been used in this project.

dp: dp is the inverse of the resolution of the accumulator with respect to input size. Accumulator collects how many circles from different edge points have been passed through each accumulator grid point. This has been set to 1. So, in our project, it has the same size as the input image.

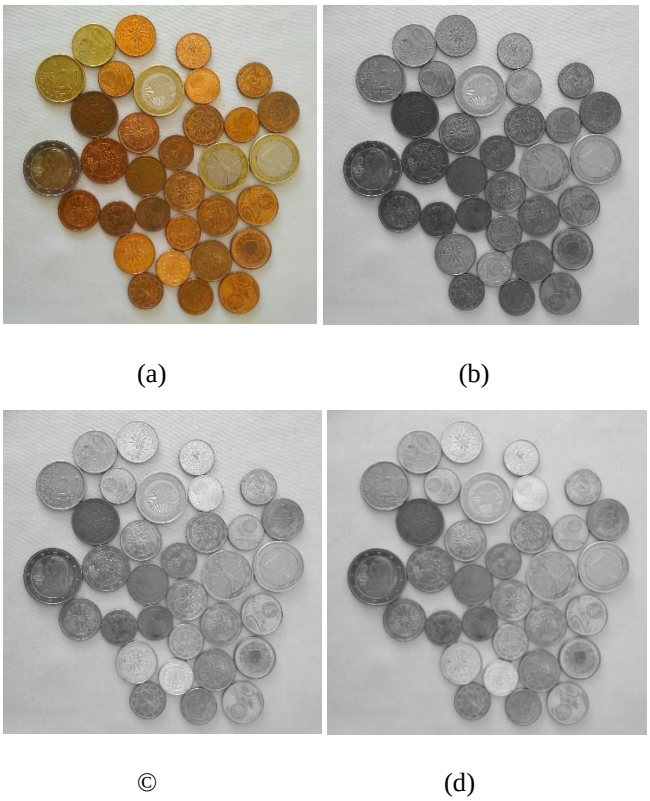
MinDist: Minimum Distance between the detected circles. This parameter is not used in this project.

Param1: Upper threshold value of the canny edge detector beyond which everything is edge. For the mentioned image, param1 was set as 80.

Param2: Lower threshold value of the canny edge detector below which everything is non-edge. For the mentioned image, param2 was set as 30.

Radius Minimum, Radius Maximum: Minimum and Maximum radius of the detected circles. This was tuned by two parameters a and b. The parameter "a" is the ratio of the number of rows and minimum radius and the parameter "b" is the ratio of the number of rows and maximum radius. Therefore, a and b represent how wide the image is with respect to minimum and maximum radius. For every image, user will have to guess a and b by considering the ratio of the width of the image to minimum and maximum radius.

For the demonstrated image, the value of a = 20 and b = 10 gave good results. That means that image is 20^{th} time wider than the minimum radius and 10^{th} times wider than the maximum radius. Circle suppression has been done by declaring a threshold as described in methodology and threshold value has been set as 0.70 times of the minimum radius of the image. In figure (4), (a) shows the result without circle suppression and (b) with circle suppression.



(a) (b)

© (d)

Figure 3: Image of a coin in (a) RGB (b) Grayscale and (c) V plane of HSV color space

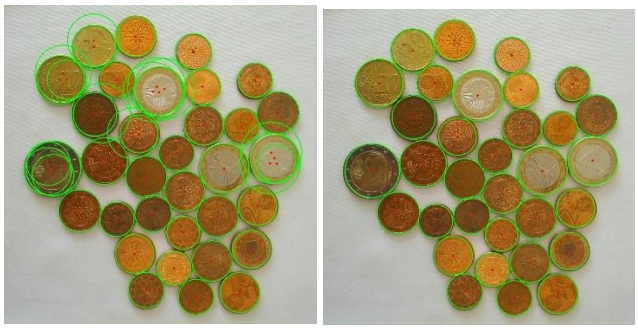


Figure 4: Coin Detection (a) without circle suppression (b) with circle suppression

Since the circle has been detected properly, the circle information can be used to count the number of coins (the number of circles detected) and the diameter of every circle as the diameter of the coin. Figure (5) shows the coin detected image with diameter and the number of circle information. After that, using Tkinter, a GUI has been developed that can open a file from the PC, a and b value can be tuned and output two images – on the right original image and on the left coin detected image. Figure (6) shows the application with output.

Figure 5: Coin detection, coin counting and diameter demonstration

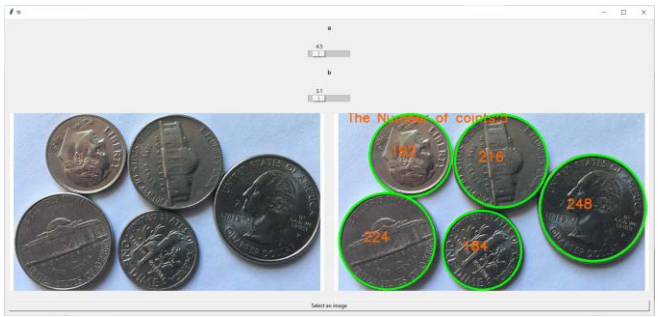
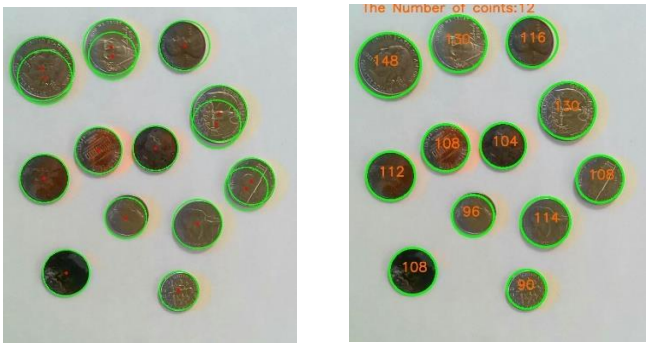
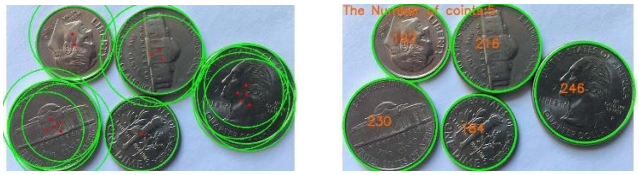


Figure 5: Coin detection, coin counting and diameter demonstration by a tkinter GUI

So far, coin detection with large number of coins and without overlap has been shown. Figure (7) shows coin detection with moderate (7(a)), small number of coins (7(b)) and coins overlapping with each other (7(c)). For each types of circle, coin detection without circle suppression and with suppression have been illustrated. It is very difficult to find a universal value of minimum radius and maximum radius for all size of coin in image. Therefore, a and b value need to be set by the user. Figure (7) gives the value of a and b for all sets of images.



(a) Moderate number of coins, a=20, b=10



(b) Small number of coins, a=4.5, b=3

(c) Overlapping coins, a=7, b=6

Figure 7: Coin Detection for (a) moderate number of coins in image (b) small number of coins in image (c) overlapping coins. For all image duals, left image is the coin detection without circle suppression and right image is the coin detection with circle suppression.

V.FUTURE WORK

In proposed method, coin detection needs to be done by a and b. In future, I will try to develop a method so that the algorithm itself sets the value by recognizing edges. Also, coin value will be recognized for bill estimation.

VI.CONCLUSION

Coin detection with varying size of coin is a tricky task. In this project, an effort has been made to develop a GUI that can detect coins by setting some parameters. In GUI, the diameter of the coin and the number of coins has been shown along with the detected coin. Neighbor detection-based circle suppression algorithm have been implemented which has seen good accuracy for all the images used. The method can be improved by auto-parameterize circle detection.

VII.REFERENCE

[1] Gomółka, Zbigniew, et al. "The use of the Circular Hough Transform for counting coins." *Measurement Automation Monitoring* 61 (2015).

[2]https://web.stanford.edu/class/ee368/Project_Autumn_1617/Reports/report_delgado.pdf

[3]https://docs.opencv.org/master/d4/d86/group__imgproc__filter.html

[4] Yuen, H. K., et al. "Comparative study of Hough transform methods for circle finding." *Image and vision computing* 8.1 (1990): 71-77.

[5]https://docs.opencv.org/3.4/d3/de5/tutorial_js_houghcircles.html

[6]https://github.com/j05t/coin_detector/blob/master/input.jg