Alexandria University
Faculty of Engineering
Computer and Communication
Program

CCE 490N Computer Vision
Assigned: Sunday, Nov. 4, 2018
Due: Sunday, Nov. 18, 2018

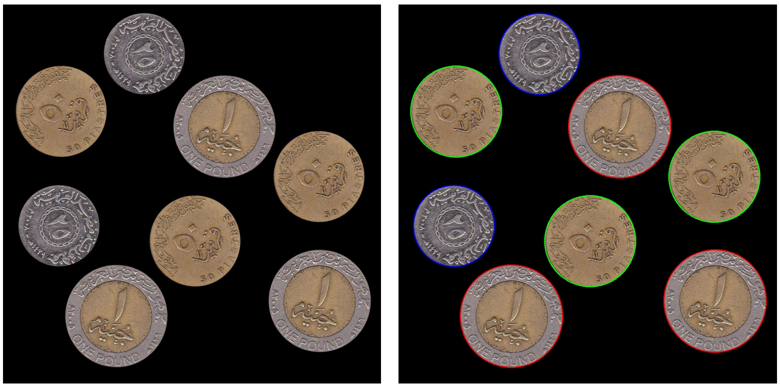
Problem Set 3
Coins Detection Using Hough Transform

The objective of this problem set is the detection of circular coins in an image using Hough Transform. This task can be subdivided into the following procedures:

1. The image is smoothed in order to reduce the amount of noise.
2. An edge detection procedure is implemented.
3. A circular Hough transform is used for circle detection.
4. Coins classification based on the radius of the detected circle.
5. The monetary value of the coins in the image is returned.

The following screenshots show the final output for the coin detection program:

Figure 1: Final output after coin detection



(a) Original image (b) Image after coin detection

1 Circle Hough Transform (CHT)

Circles are a common geometric structure of interest in computer vision applications. The use of the Hough transform to locate circles will be explained and demonstrated in this problem set. This is a particular example of the use the Hough transform to search a parameter space.

The Hough transform can be used to determine the parameters of a circle when a number of points that fall on the perimeter are known. A circle with radius R and center (a, b) can be described with the following equations:

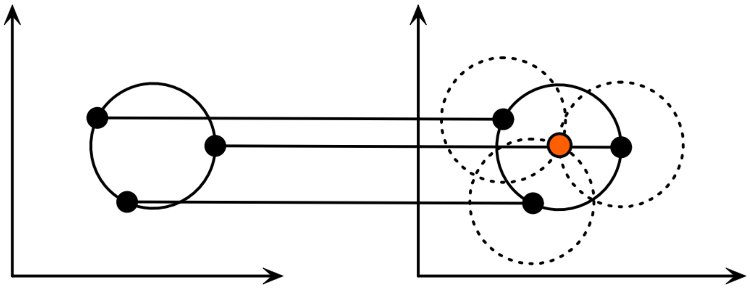
Radius for Different Coin Types	
One Egyptian Pound	135 Pixels
50 Piastres	120 Pixels
25 Piastres	107 Pixels

Table 1:
Radius for Different Coin Types

$$(x-a)^2 + (x-b)^2 = R^2 \tag{1}$$

The locus of (a, b) points in the parameter space fall on a circle of radius R centered at (x, y). The true center point will be common to all parameter circles, and can be found with a Hough accumulation array.

Figure 2: Each point in geometric space (left) generates a circle in parameter space (right). The circles in parameter space intersect at the (a, b) that is the center in geometric space.

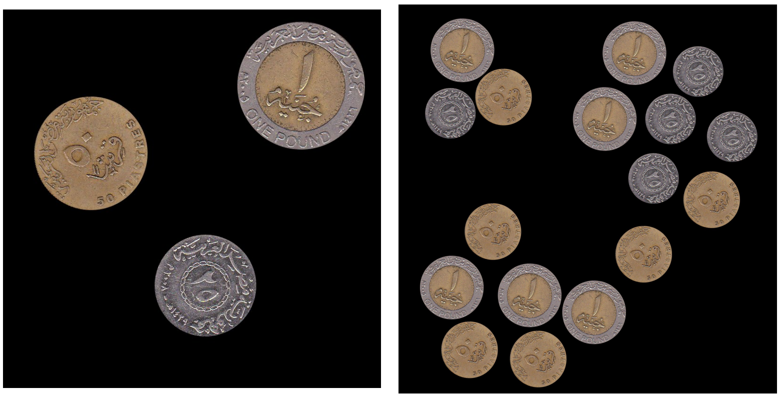


2 Data Set

Your program shall be tested on several images for Egyptian coins. The data set has been pre-processed and most of the noise has been removed. You could assume the following about the data set:

- Images include the following coins only: one Egyptian pound, 50 piastres and 25 piastres.
- There are at most 20 coins per image.
- Each image has at least one coin of each type.
- For simplicity, all the coins of the same type have the same radius.

Figure 3: Sample Images from the Egyptian Coins Data Set

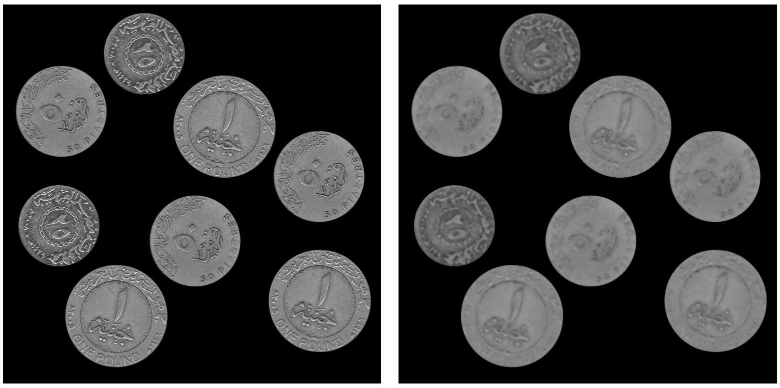


3 Implementation Details

3.1 Smoothing the image

Smoothing the image is accomplished using a 2-dimensional median smoothing filter. A window size of seven will be suitable for the provided data set.

Figure 4: Smoothing the image with median filter



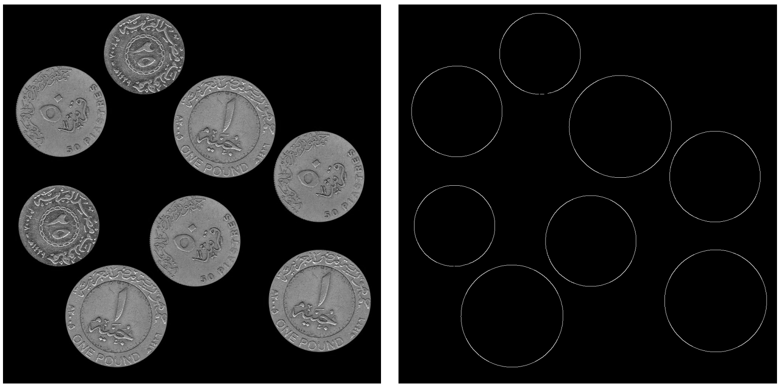
(a) Gray Scale Image

(b) Smoothed Image

3.2 Edge Detection

After smoothing the image, Canny's algorithm has to be used for edge detection. We will use relatively high values for thresholding to remove most of the noise. Suitable threshold values for the provided data set are 250 for the low threshold and 280 for the high threshold.

Figure 5: Edge Detection Using Canny's Algorithm

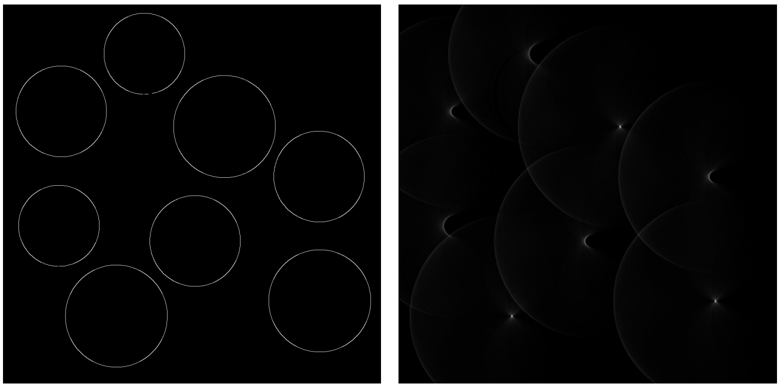


(a) Smoothed Image (b) Detected Edges

3.3 Accumulation into (a, b)-space using circular Hough transform

```
procedure CHT(Edges, R)
    for each edge point $\in$ Edges do
        for each possible value of $b$ do
            Substitute in (1) and solve for $a$
            Increment $H(a, b)$
        end for
    end for
end procedure
```

Figure 6: Accumulation into (a, b)-space using CHT



(a) Detected Edges Using Canny's Algorithm (b) Rendering of Transform Results for R=135px

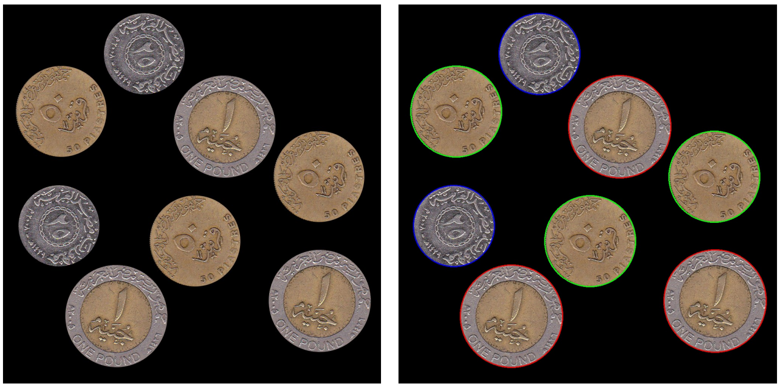
3.4 Refining Coordinates and CHT Post-Processing

During the whole process of finding the centers, some inaccuracies could occur. This could be due to choosing a large bin size for CHT or due to noise in the detected edges. Therefore, after finding the coordinates, a search in the (a, b)-space is executed. We look for the highest peak of the accumulator function and perform non-maximum suppression for lower values.

3.5 Coin Detection

The data set has been processed so that different types of coins could be easily detected. You could benifit form the radii values in $table(1)$ to differenciate between Pounds, 50 Piastres and 25 Piastres.

Figure 7: Final output after coin detection



(a) Original image (b) Image after coin detection

4 Requirements

1. Explain your implementation in concise steps.
2. For one of the images, display and briefly comment on the Hough space accumulator array.
3. Experiment with ways to determine how many circles are present by postprocessing the accumulator array.
4. For one of the images, demonstrate the impact of the vote space quantization (bin size).
5. The detected coins should be highlighted in the output image with a discriminative color.
6. Return the monetary value of the coins in the image.

5 Deliverables

You are required to deliver the following:

- Your code.
- Output for some test images.
- Report including explanation of your code and representative results on sample test images.

6 Submission Instructions

1. You have to work in groups of two.
2. Please send your assignment to:
   - **Email:** csd.delivery@gmail.com
   - **Subject:** [CV] Assignment 3
   - **Body:** Your Name / Your ID
   - **Attachment:** One compressed file including a folder with your ID number, please include all your *.py files in this folder, don't use any subfolder for your source files.
   - **Deadline:** Monday, 21 April 2014, 11:59 PM, Editing the source code after submission is not allowed.

Good Luck