

# HOMework 6 U1

## HOUGH TRANSFORM

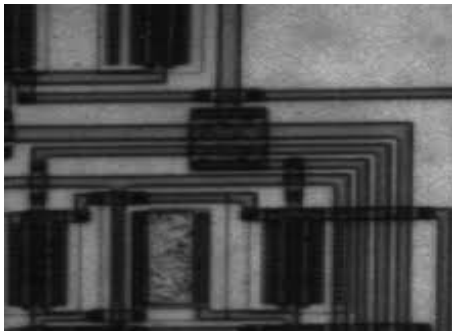
Héctor Mauricio Mendoza Xicoténcatl

**Abstract**—In this report, we documented our coding process. This process involved the use of the Hough Transform image processing technique.

**Index Terms**—OpenCV, Hough Transform, Hough Linear, Hough Transform Circle.

### 1 PROBLEM TO SOLVE

Use the following circuit image for the following:



Circuit Image - Input

- Apply Hough Transform to try find the lines

Take a picture of coins on any surface and:



Coins Image - Second Input

- Segment the coins from the background
- Apply Hough Transform to find the circles

### 2 CODING PROCESS

Our coding process is divided into two different codes. The first one corresponds to the solution of the lines for the first input image. The second corresponds with the drawing of the circular contours of the second input image.

#### 2.1 HOUGH TRANSFORM

Similar to previous exercises, first we imported the libraries necessary for the program with the commands:

```
import numpy as np
import cv2
```

We detect the edges using the Canny operator:

```
dst = cv.Canny(source_img, 50, 200, None, 3)
```

Then we apply the transform using the following code line:

```
lines = cv.HoughLines(dst, 1, np.pi / 180,
150, None, 0, 0)
```

Next, we display the result by drawing the lines. For this, we decided to use a for condition (Short example):

```
if lines is not None
—for i in range(0, len(lines)):
——rho = lines[i][0][0]
```

```

———theta = lines[i][0][1]
————— continue....

```

Finally, we print the results:

```
cv.imshow("Original Image", source_img)
```

## 2.2 COIN SEGMENTATION

As usual, first we imported the libraries necessary for the program with the commands:

```

import numpy as np
import cv2
import numpy as np

```

Next, we convert our input image to gray-scale:

```

gray = cv.cvtColor(src,
cv.COLOR_BGR2GRAY)

```

Next we Blur the image to reduce noise:

```
gray = cv.medianBlur(gray, 5)
```

The following command applies the Hough transform on the image:

```

circles = cv.HoughCircles(gray,
cv.HOUGH_GRADIENT, 1, rows / 8,
param1=100, param2=30, minRadius=1,
maxRadius=100)

```

We can say that this activity is another tool we have learned to use in OpenCV. With this we were able to detect circles and lines within different figures. I consider the most difficult challenge was the.

## 3 RESULTS



Hough Result



Hough Circular Result

## 4 CONCLUSIONS

We can say that this activity is another tool we have learned to use in OpenCV. I consider the most complicated task this time involved the use of the operators to find the input result desired.



**Héctor Mendoza** Héctor Mauricio Mendoza Xicoténcatl is a Computational Robotics Junior student, currently studying at Universidad Politécnica de Yucatán.