FINAL PROJECT U1 GEOMETRICAL FIGURES

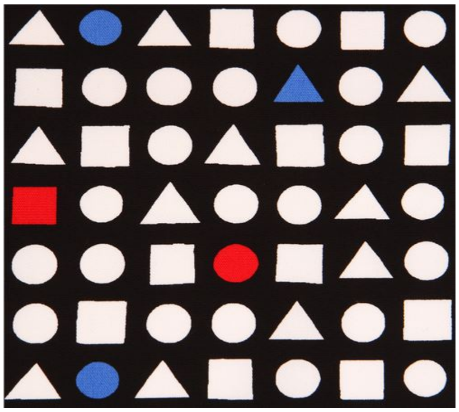
Héctor Mauricio Mendoza Xicoténcatl

Abstract—In this report, we documented our coding process for the final project of ”Artificial Vision” subject. This process involved many different operations involving three geometric figures: Triangles, Squares, and Circles.

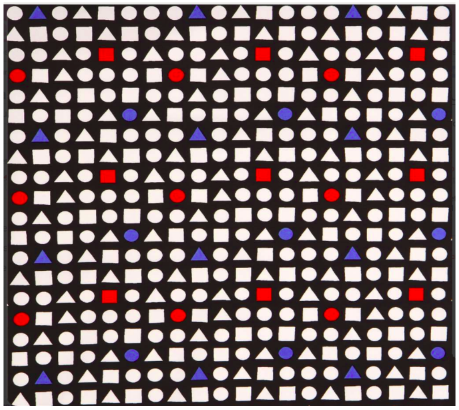
Index Terms—OpenCV, Color segmentation, Contour segmentation, Thresholds, Area calculation of a figure.

1 PROBLEM TO SOLVE

FRom the attached images find:



First Input



Second Input

- Total area.
- The total area of circles / triangles / squares.
- Total number of figures:

1) Small figure : 49
2) Big figure : 400

- Number of circles / triangles / squares.
- Amount of Triangles:

1) Blue
2) Red
3) White

- Amount of Circles

1) Blue
2) Red
3) White

- Amount of Squares

1) Blue
2) Red
3) White

2 CODING PROCESS

In order to solve this activity, we used all the knowledge learned throughout the unit one of Artificial Vision. We used a great variety of OpenCV functions to be able to get the desired results.

2.1 COLOR SEGMENTATION

This section of the coding process is one of the most essentials parts of the project. Here we are able to detect the shapes on the figure, classify them into figures and write the results on the figure itself. Similar to previous exercises, first we imported the libraries necessary for the program with the commands:

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

Here we apply some filters which will allow us to detect the contours on the figure, and make the code available to make manipulate.

```
imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
imgBlur = cv2.GaussianBlur(imgGray,(7,7),1)
imgCanny = cv2.Canny(imgBlur,50,50)
```

```
hsv_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
```

Then we apply some maks processing to the detect only the colors we want. In this case red and blue. Blue example:

```
lower_blue = np.array([100,110,110])
upper_blue = np.array([130,255,255])
blue_mask = cv2.inRange(hsv_img, lower_blue, upper_blue)
blue = cv2.bitwise_and(img, img, mask = blue_mask)
```

2.2 GEOMETRIC SEGMENTATION & AREA CALCULATION

In this section, we decided to apply a python function that will allow us to apply a geometrical segmentation to any figure we want. Not restricted as a path in the first coding lines, for example.

```
def getContours(img):
```

One key command to this project is cv2.findContours(). This command allow us to write the contours on the image, that we will later segment.

```
contours,hierarchy = cv2.findContours(img,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
```

Next, we use a for condition to calculate the type of figures and their corresponding areas.

```
—– for cnt in contours:
——– area = cv2.contourArea(cnt)
```

In this section we are able to identify the type of geometrical figure.

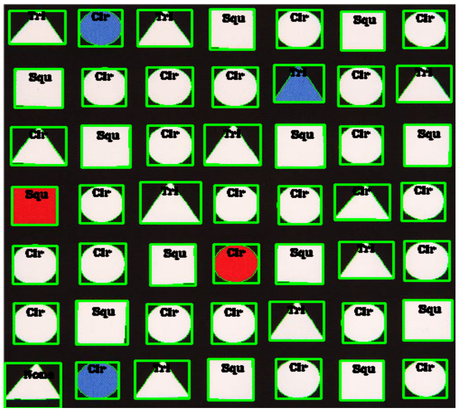
```
————– if len(approx)==3:
————— tri+=1
————— area_tri+=area+area
————— objectType=”Tri”
```

In the section above we calculate the length of the object, if it is 3 we define that it will be a triangle.
Next we print the type of figure on the image:

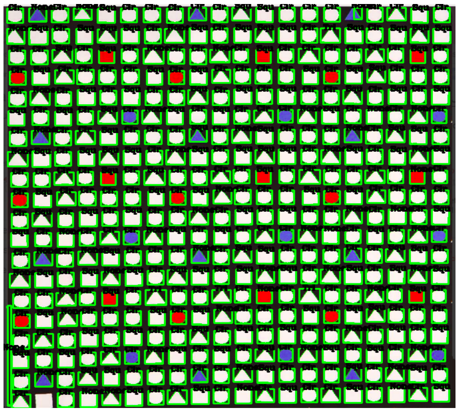
```
cv2.putText(imgContour,objectType,
(x+(w//2)-10,y+(h//2)-
10),cv2.FONT_HERSHEY_COMPLEX,0.4,(0,0,0),2)
```

We assign a variable to the geometrical figures and we print them.
´print(”————————————— Number of figures ————————————–”)
print(”Total Number of Triangles: ”, totaltri)
print(”Total Number of Squares: ”, totalsqr)
print(”Total Number of Circles: ”, totalcir)
print(”Total Number of Figures”, totalfig)

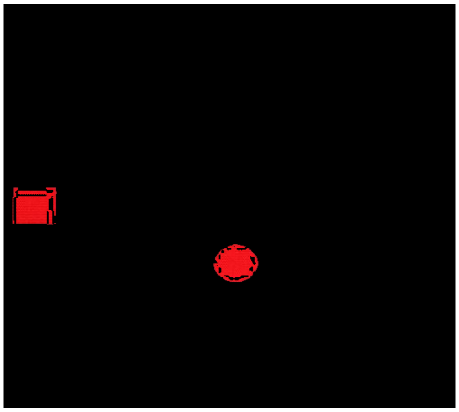
3 RESULTS


Tri Cir Tri Squ Cir Squ Cir
Squ Cir Cir Cir Tri Cir Tri
Cir Squ Cir Tri Squ Cir Squ
Squ Cir Tri Cir Cir Cir Cir
Cir Cir Squ Cir Squ Tri Cir
Cir Squ Cir Cir Tri Cir Squ
None Cir Tri Squ Cir Squ Cir

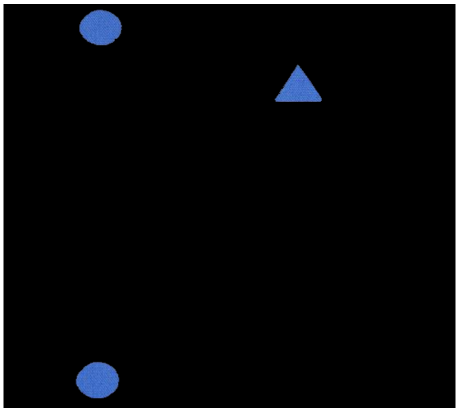
Small Figure Contour



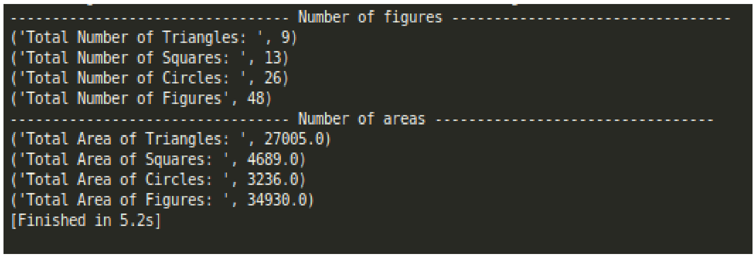
Big Figure Contour



Color Segmentation - Red



Color Segmentation - Blue



```
-------------------------------- Number of figures --------------------------------
('Total Number of Triangles: ', 9)
('Total Number of Squares: ', 13)
('Total Number of Circles: ', 26)
('Total Number of Figures', 48)
-------------------------------- Number of areas --------------------------------
('Total Area of Triangles: ', 27005.0)
('Total Area of Squares: ', 4689.0)
('Total Area of Circles: ', 3236.0)
('Total Area of Figures: ', 34930.0)
[Finished in 5.2s]
```

4 CONCLUSIONS

I consider this was one of the most challenging projects I have faced involving programming. The lack of time is the answer for my results. I consider that this project involved all the things learned throughout this first unit of Artificial Vision. One of the most challenging sections in this project was the logic behind each one of the processes. I consider also the detection of colors, is until now, very challenging for me. I was not able to finish it one hundred percent but I think the knowledge used to solve this project is the most important aspect for me.

Héctor Mendoza Héctor Mauricio Mendoza Xicoténcatl is a Computational Robotics Junior student, currently stu-diying Universidad Politécnica de Yucatán.