HOMEWORK 4 U1
CONTOUR SEGMENTATION

Héctor Mauricio Mendoza Xicoténcatl

Abstract—In this report, we documented our coding process. This process involved two different segmentations in one program: Color segmentation and Contour Segmentation. In addition, we used an OpenCV function to calculate the perimeter of the contour segmentation in our figures.

Index Terms—Thresholds, Color Segmentation, OpenCV2, Contour Segmentation, Perimeter Segmentation.

1 PROBLEM TO SOLVE

In a piece of paper draw different figures (at least 5) and paint them using different colors. Obtain:

- Segmentation by colors.
- Contour of each figure.
- Perimeter of each figure.

2 CODING PROCESS

In this activity, we drew 5 different figures in a blank paper and colored them with two different colors, red and green. With this figure, we were able to apply different methods of shape detection. In the following sections we will discuss the coding process of each one of them.

2.1 CONTOUR DRAWING & PERIMETER

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

import numpy as np
import cv2

Then, we started to use a new command in order to use our webcam:

cap = cv2.VideoCapture(0)

Due to we needed to find each of the contours corresponding to each geometric figure, we obtained a binarized image, for this we used edge detection with cv2.Canny. After this made smoother the binary image so we will use dilation and erosion.
We used the following command to draw the contours in our window frame:

cv2.drawContours(frame2, cnts, -1, (0,255,0), 2)

Following the program, the next part is the figure detection. For this we used a for condition, in order to detect the different figures. In order to do this, we used an if condition on the approx variable. With this, we are able to distinguish figures. As an example, here is the code for the triangle figure:

if len(approx)==3:
cv2.putText(frame2,'Triangulo: ' + str(int(perimeter)), (x,y-5),1,1.5,(0,255,0),2)

With the command **"perimeter = cv.arcLength(cnt,True)"** we are able to get the perimeter of our figure in screen.

2.2 SEGMENTATION BY COLORS

The color segmentation coding starts with the use of two smoothing techniques:

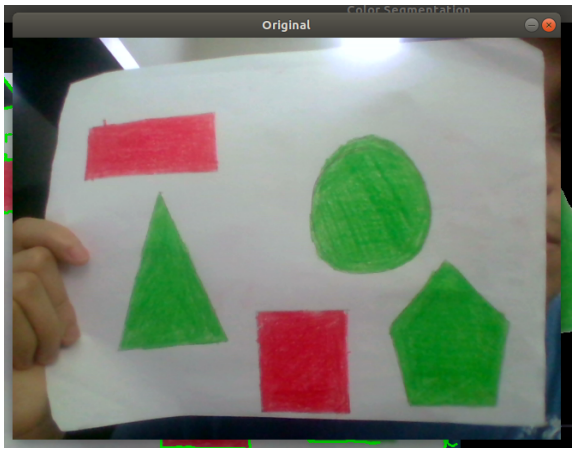
blurred_frame = cv2.GaussianBlur(frame, (5, 5), 0)
hsv_frame = cv2.cvtColor(blurred_frame, cv2.COLOR_BGR2HSV)

Then we applied two ranges of colors and save them into a mask in order to finish our color segmentation. This ranges were applied to the red and green colors. Finally, we used the bit-wise operation in order to only detect the predetermined color. Example:

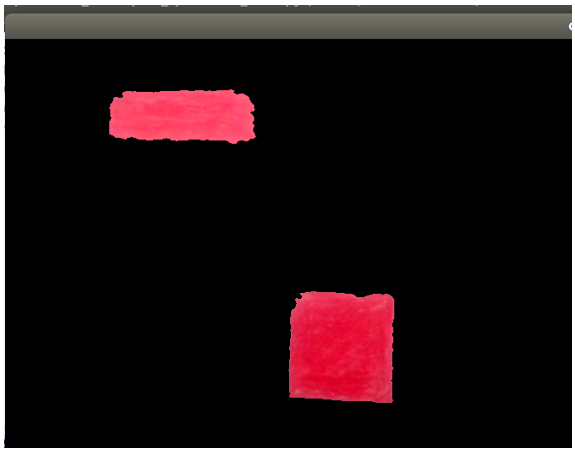
lower_blue = np.array([161,155,84])
upper_blue = np.array([179,255,255])
red_mask = cv2.inRange(hsv_frame, lower_blue, upper_blue)
red = cv2.bitwise_and(frame, frame, mask = red_mask)

Finally, we print our results.

3 RESULTS

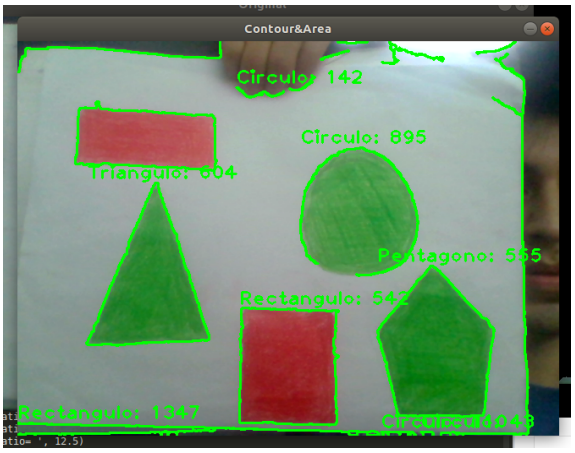


Original Image



Color segmentation - Red Color

Color segmentation - Green Color



Contour Detection Perimeter

4 CONCLUSIONS

I consider the most complicated part of this exercise was the realization of how to distribute each one of the segmentations. It was complicated for me what to do first. Once I started to read the documentation, it become easier to use the functions to each one of the tasks requested. As a final comments for this activity, I must emphasise that the results seen in the video are not one hundred percent exact due to some reasons: The type of webcam used, the stability of the sheet of paper, and the figures draw are not completely straight.

REFERENCES

[1] G. Solano, "Detectando FIGURAS GEOMÉTRICAS () con OpenCV - Python omes-va.com", OMES, 2020. [Online]. Available: https://omes-va.com/detectando-figuras-geometricas-con-opencv-python/. [Accessed: 30-Sep- 2020]

[2] ”OpenCV: Contour Features”, Docs.opencv.org, 2020. [Online]. Available: https://docs.opencv.org/master/dd/d49/tutorial_py_contour_features.html. [Accessed: 30- Sep- 2020]

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