EE 576 Project: # 3

Musa ALMAZ

03.04.2024

I. Segmentation

1) Write a function that segments the green region in the given images and then finds it outer and inner boundaries. You may write your own connected components algorithm or use an available algorithm (ie. OpenCV, external). If you use an external segmentation code, make sure you given the reference for it.

The **segmentGreenRegion** function in the Project3 class performs a crucial task in image processing: the segmentation of green regions.

In this function, segmentation is achieved through color-based filtering in the HSV (Hue, Saturation, Value) color space, a method known for its effectiveness in dealing with variations in lighting and shade compared to the traditional RGB space. By precisely defining the range of green hues and applying this criterion to the original image, the algorithm efficiently highlights green regions while dimming the rest.

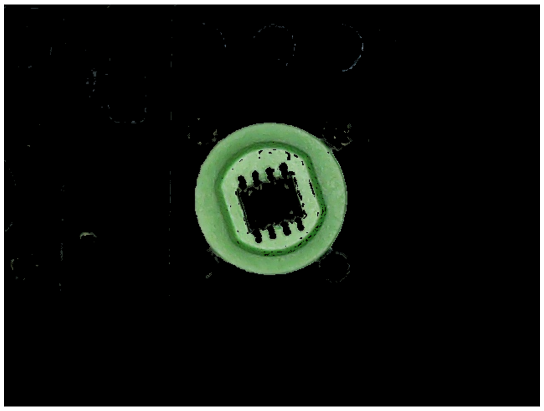
Fig. 1: Segmented Image

II. Finding Boundaries

1) Now write a function that finds the inner and outer boundaries of the green segment.

The algorithm for finding boundaries in the Project3 class consists of two key methods: **findContoursInRegion** and **findLargestObject**. Initially, **findContoursInRegion** takes a binary image as input, where the regions of interest are segmented from the background. Utilizing OpenCV's findContours method, it identifies and stores all the contours in the image along with their hierarchical relationships. Contours are effectively the boundaries of connected regions in the binary image.

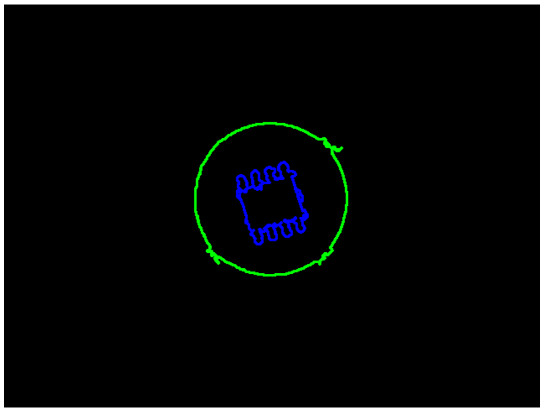
Fig. 2: Boundary Drawed Image

III. Finding Object

1) Write a function that finds the object inside the green region. Considering the horizontally aligned object to have 0 degrees rotation, find the orientation of the object and undo the rotation.

The **findLargestObject** focuses on identifying the most significant contour, which in many applications corresponds to the primary object of interest. To achieve this, it first checks if there are enough contours detected. If so, it then proceeds to sort these contours based on their area, singling out the two largest. From these, the method picks the second largest contour as the largest one typically represents the outer boundary of the entire image or a background region. The chosen contour is then used to form a RotatedRect, which is the minimum area rectangle encapsulating the contour. This rectangle provides valuable information about the orientation and spatial distribution of the object within the image, allowing for further image processing tasks such as object alignment or region analysis.

IV. Results

As shown in the Figure 4, the overall result is depicted.

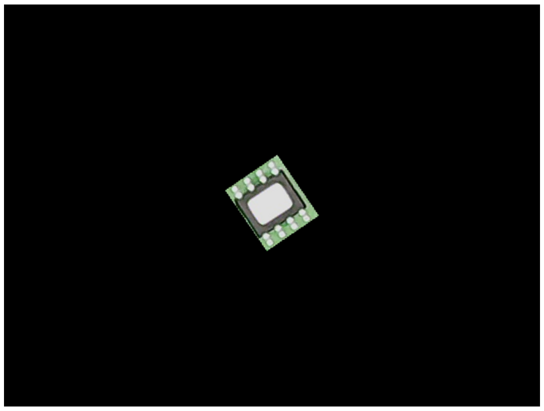
Fig. 3: Rotated Image

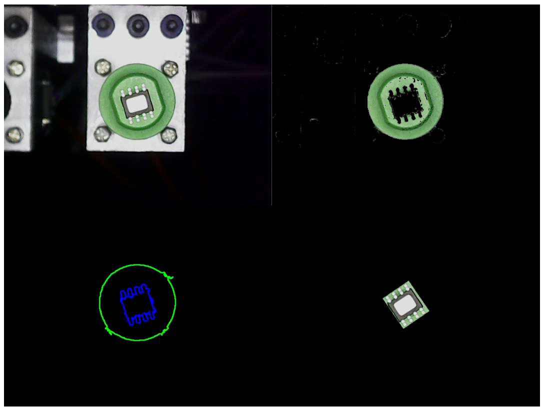
Fig. 4: Overall Result