Batuhan Tosun

2017401141

EE 576 MACHINE VISION | HOMEWORK 2

a) This project consists of two main stages. In the first stage, our goal is to remove the green region from the given image. To achive this, the given image is first transformed to HSV color space using OpenCV's cvtColor function with parameter "COLOR_RGB2HSV". Then, the three channels of the resulting image are displayed separately as gray images. The HSV channels of both "bot1_1206_1.jpg", "bot1_sot23_1.jpg" and "bot1_soic.jpg" images can be seen below.

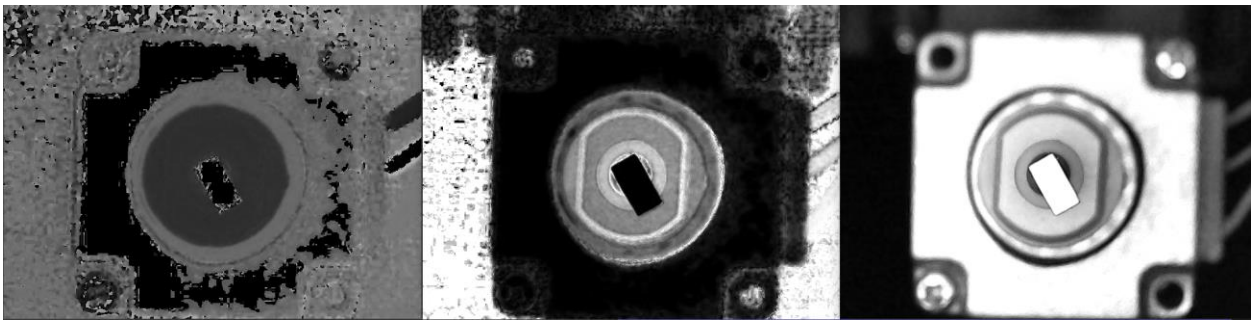

Fig1: The Hue, Saturation and Value Channels of the image "bot1_1206_1.jpg"

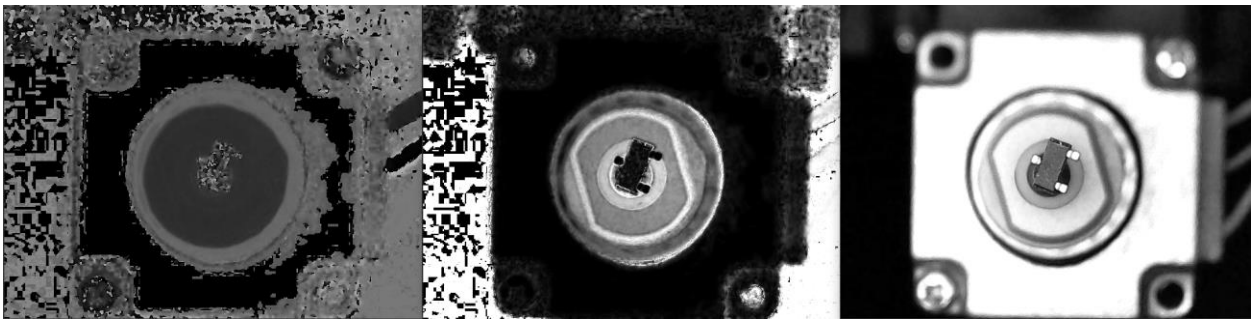

Fig2: The Hue, Saturation and Value Channels of the image "bot1_sot23_1.jpg"

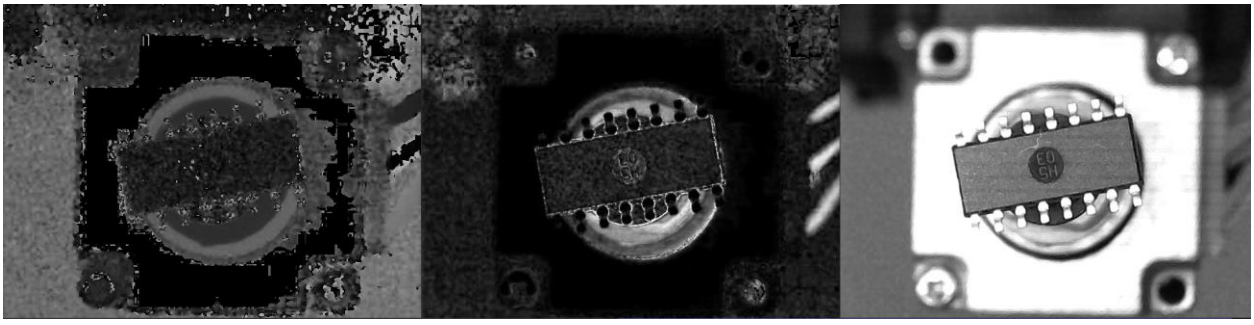

Fig3: The Hue, Saturation and Value Channels of the image "bot1_soic.jpg"

Based on the "bot1_1206_1.jpg" image, due to its simplicity in finding the desired green region, I have determined the lower and upper bounds for H, S and V values, where the "green" color lies in the HSV space. These lower and upper bounds then given to OpenCV's inRange function as parameters to generate

the binary mask. Since the green region should appear black and all the remaining regions should appear white, the generated mask is subtracted from 255 and the desired mask is obtained. The original, thresholded and masked versions of the images “bot1_1206_1.jpg”, “bot1_sot23_1.jpg” and “bot1_soic.jpg” can be seen below.

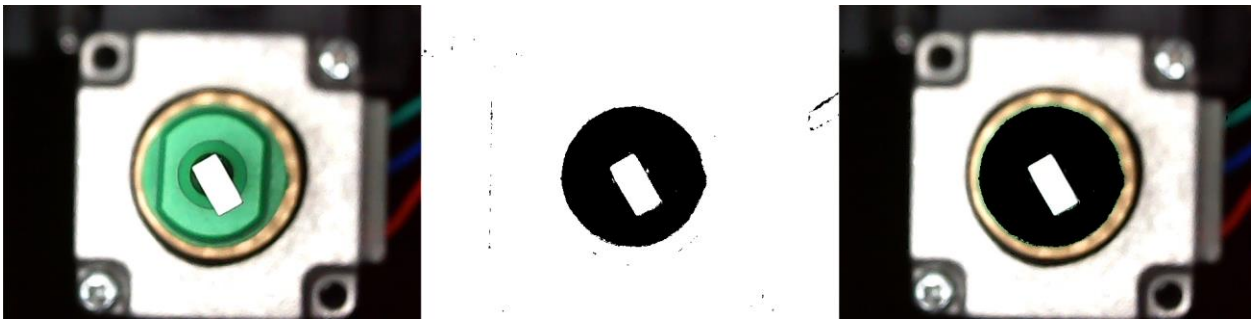

Fig4: The original, thresholded (binary mask), and mask-applied versions of the image “bot1_1206_1.jpg”

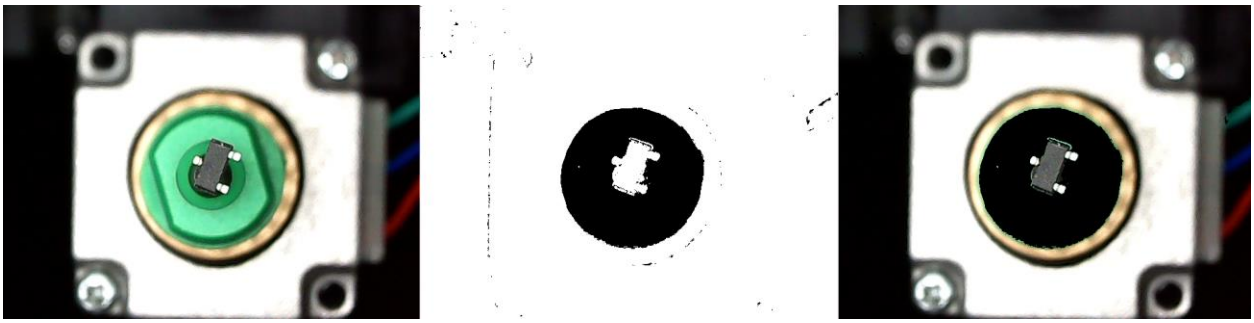

Fig5: The original, thresholded (binary mask), and mask-applied versions of the image “bot1_sot23_1.jpg”

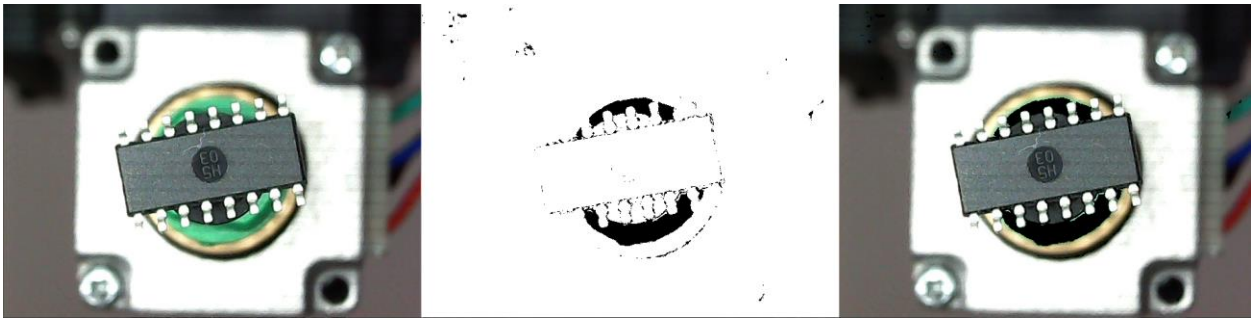

Fig6: The original, thresholded (binary mask), and mask-applied versions of the image “bot1_soic.jpg”

From the figure, it can be observed that the binary masks generated by thresholding the images are not perfect. There are some undesired regions which appear black after thresholding even though they are outside the green region. Also, there are some small undesired regions which appear white after thresholding within the green region.

b) In the second stage of the project, our goal is to get the boundaries of the central object within the green region as continuous as possible. To achieve this purpose, we need to implement some necessary image enhancement methods and high pass filtering to the enhanced image for detecting the edges/boundaries. The first thing to do for image enhancement is to eliminate those outlier regions/pixels mentioned previously. Appropriate morphological transformations are applied to achieve this. After a few

trials, I have decided to implement opening (for removing the small objects on the black background) followed by closing (for removing the small holes on the main image) on the binary mask image using OpenCV's morphologyEx function [1].

The resulting binary mask is enhanced but there are some sharp points around the edges due to the shape of structuring element used in the morphological transformations. To reduce the effect of those sharp points, I have applied Gaussian filter using OpenCV's GaussianBlur function [2]. Then, I have implemented Canny Edge Detector algorithm using OpenCV's Canny function on the enhanced binary mask [3]. The original and the boundary detected versions of "bot1_1206_1.jpg", "bot1_sot23_1.jpg", and "bot1_soic.jpg" images are given below side-by-side.

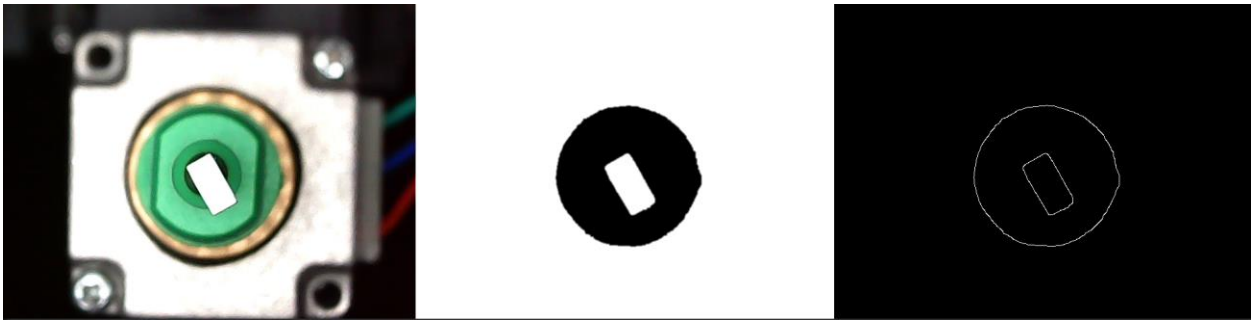

Fig7: The original image of "bot1_1206_1.jpg", its enhanced binary mask, and the borders of the main object

Fig8: The original image of "bot1_sot23_1.jpg", its enhanced binary mask, and the borders of the main object

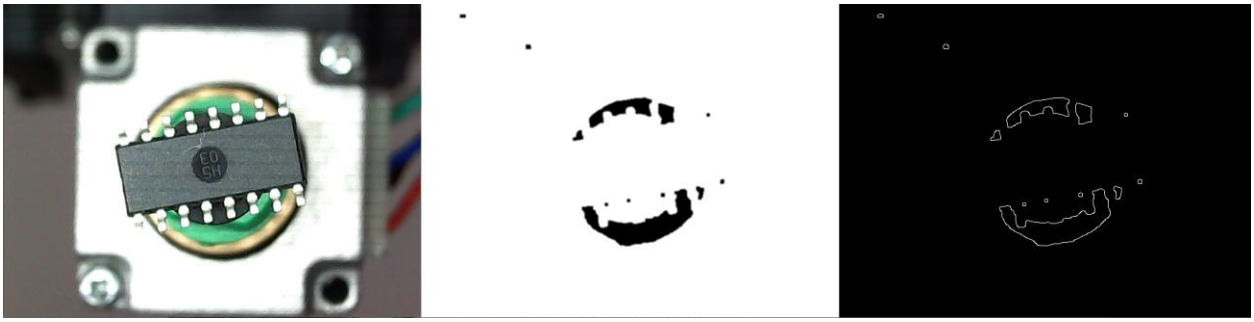

Fig9: The original image of "bot1_soic.jpg", its enhanced binary mask, and the borders of the main object

References

[1] https://docs.opencv.org/3.4/d3/dbe/tutorial_opening_closing_hats.html

[2] https://docs.opencv.org/3.4/dc/dd3/tutorial_gausian_median_blur_bilateral_filter.html

[3] https://docs.opencv.org/3.4/da/d5c/tutorial_canny_detector.html