Introduction to Robotics Lab

Laiba Ahmed and Muhammad Suleiman Qureshi

March 2023

Lab 07

Task 7.3

Our object detection strategy relies on just the RGB camera. This is mainly because of 2 reasons, our initial strategy was to also incorporate the depth camera to get the top face of the cube to get the most accurate location but since the depth camera was not giving us accurate enough values, we did not utilize it to detect the object. The second reason was that we thought if we could get the location, albeit not incredibly precise, through just the RBG camera, we could get the pincher over it and since the end-effector can expand to a size much greater than the cube, when the claw was closing it would somewhat push the cube into the center of itself and pick it up.

The strategy explained below will include visuals and the steps applied on the image below, with our detection running for all cubes individually but the steps will be shown for the green cubes.

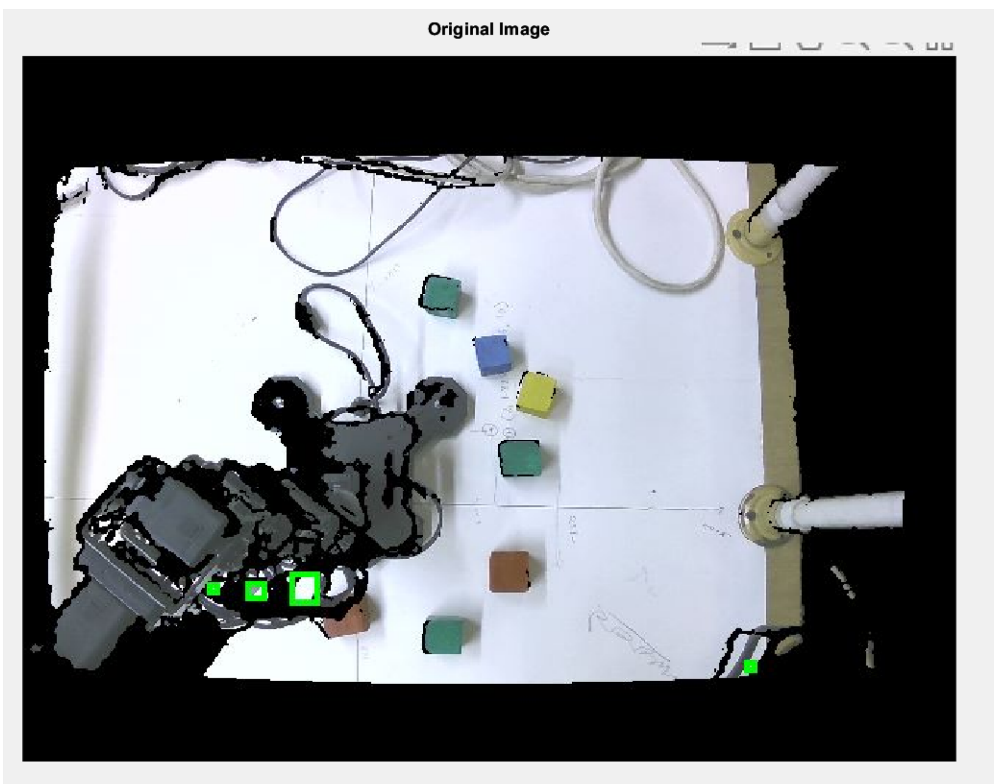

Figure 1: Test Image

Our strategy was to get a threshold for every colour (gotten through trial and error). This threshold was then applied over the image, and everything in this threshold would be considered the colour and whitened and everything not in the threshold would be black. Then to remove stray pixels, we used morphological operations.

Then, we had the squares/cubes in the image and find out the connections between the pixels. Then we found out the center of the said connected pixels. We then plotted the center of the connected pixels on top of the original image to depict where the algorithm found the cube.

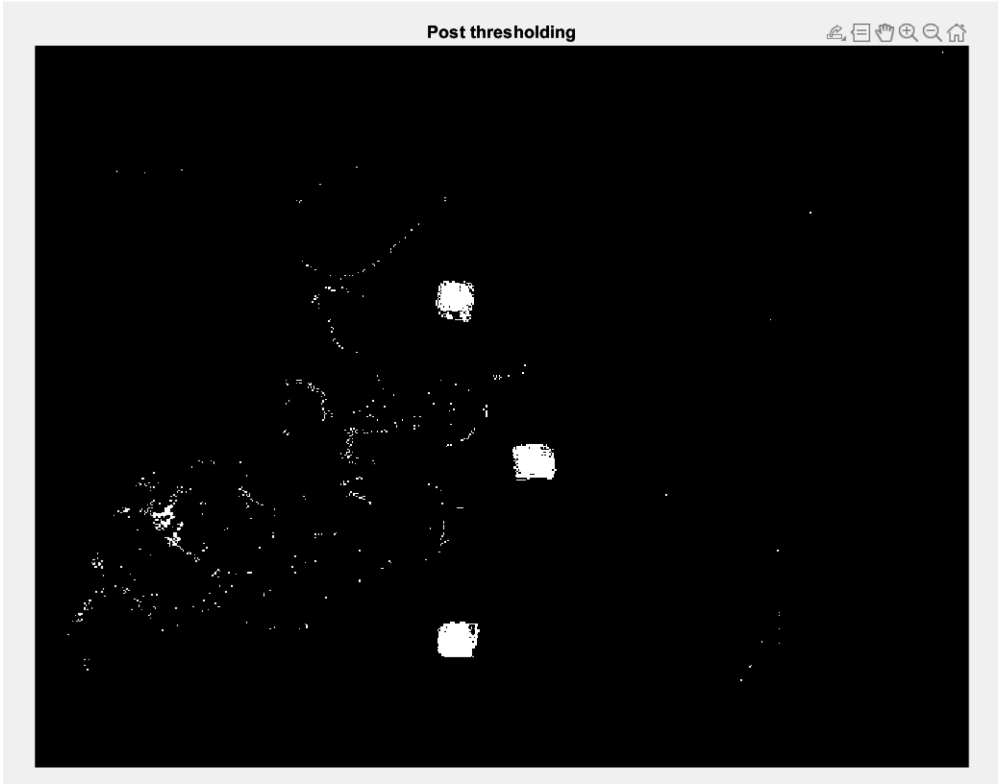

Figure 2: After getting the image within threshold values

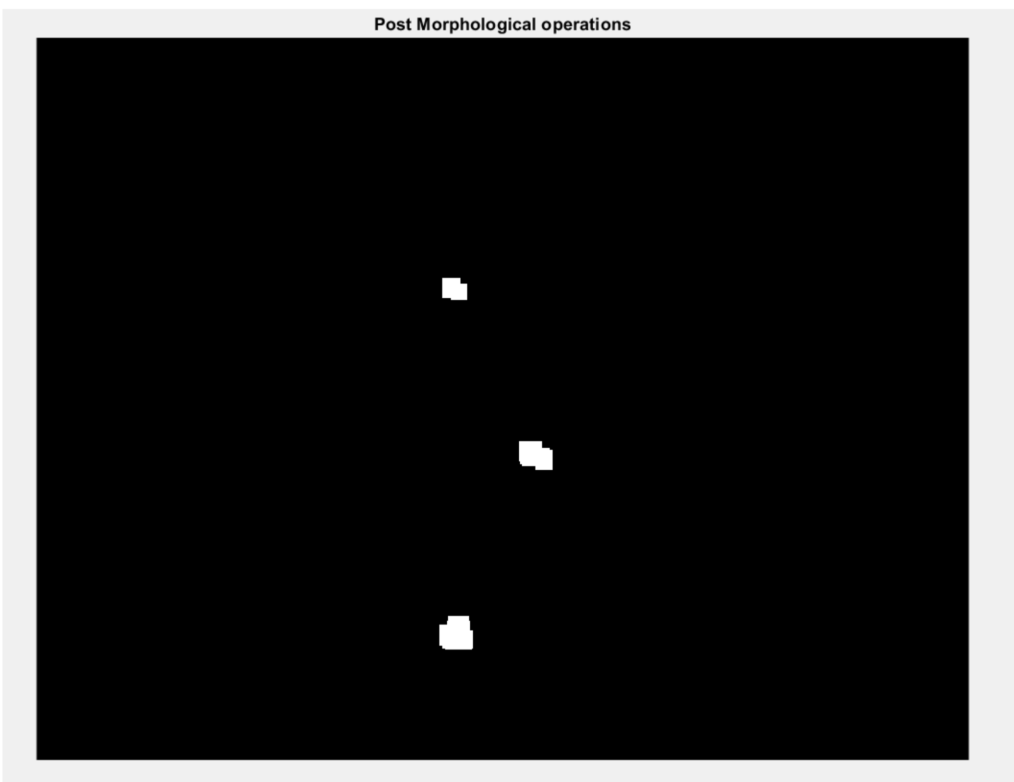

Figure 3: After morphological operations to remove stray pixels

Statistics

Number of objects of each color in the workspace to be detected

Our code was only working as long as there was an object of every colour within the view of the camera.

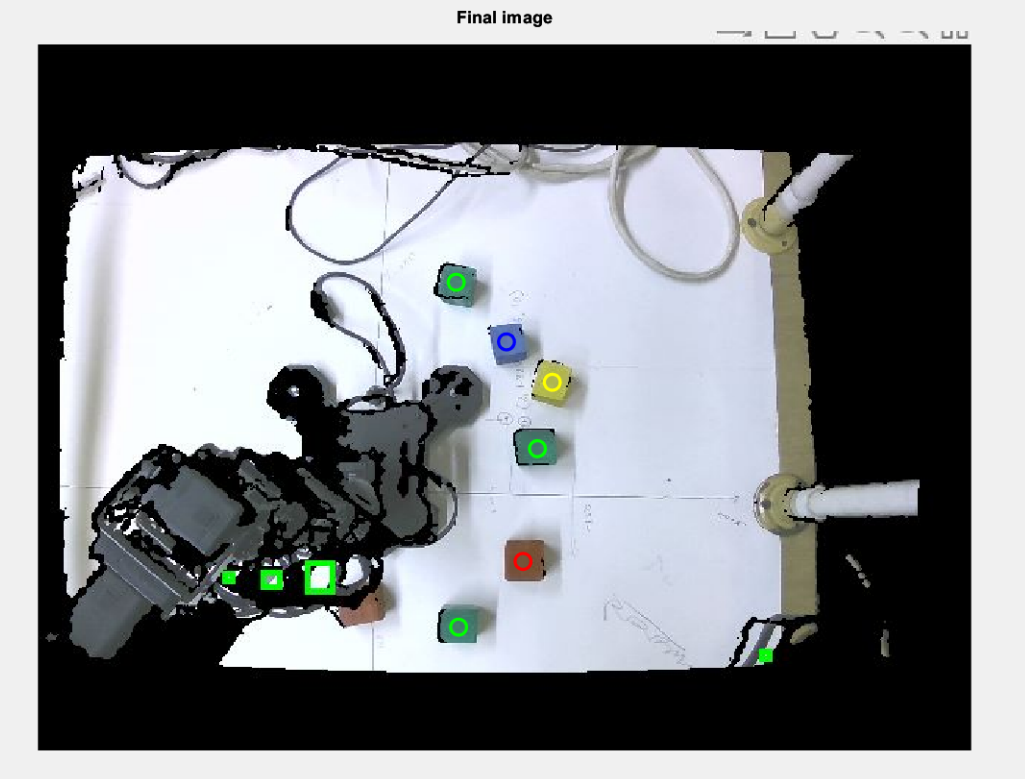

Figure 4: After applying all steps for all colours

Number of objects of each color correctly detected

The number of objects did not influence colour detection.

Accuracy rate for each color

There was about an 85% rate for all the colours we used, the only issue was that we sometimes had to tweak the threshold values to be able to get all the visible cubes registered.

0.1 Code

```
im = imread('testo.jpeg');
img = im2double(im);
figure;
imshow(im);
title("Original Image")
figure;
% green
min = [0 0.4 0.4];
max = [0.4 1 0.55];

%Threshold to extract the green regions
green_mask = (img(:,:,1) >= min(1)) & (img(:,:,1) <= max(1)) & (img(:,:,2) >= min(2)) & (img(:,:,
imshow(green_mask)
title("Post thresholding")
figure
% Clean up the mask using morphological operations
se = strel('square', 10);
green_mask = imclose(imopen(green_mask, se), se);

imshow(green_mask)
title("Post Morphological operations")
figure
```

```
% Find connected components in the mask
cc = bwconncomp(green_mask);

% Extract the location of the cubes as a list of centroids
stats = regionprops(cc, 'Centroid');
green_cubes = cat(1, stats.Centroid);
disp(green_cubes);
disp(size(green_cubes));
% Display the image with the cubes marked
imshow(img);
title("Final image")
hold on;
plot(green_cubes(:,1), green_cubes(:,2), 'go', 'MarkerSize', 10, 'LineWidth', 2);
hold on;

%red range
min = [0.4 0 0];
max = [1 0.45 0.45];

red_mask = (img(:,:,1) >= min(1)) & (img(:,:,1) <= max(1)) & (img(:,:,2) >= min(2)) & (img(:,:,2

% Clean up the mask using morphological operations
se = strel('square', 10);
red_mask = imclose(imopen(red_mask, se), se);
% figure;
% imshow(red_mask)
% figure;
% Find connected components in the mask
cc = bwconncomp(red_mask);

% Extract the location of the cubes as a list of centroids
stats = regionprops(cc, 'Centroid');
red_cubes = cat(1, stats.Centroid);
disp(red_cubes);
disp(size(red_cubes));
% Display the image with the cubes marked
% imshow(img);
hold on;
plot(red_cubes(:,1), red_cubes(:,2), 'ro', 'MarkerSize', 10, 'LineWidth', 2);


%blue
min = [0 0 0.55];
max = [0.4 0.5 1];
blue_mask = (img(:,:,1) >= min(1)) & (img(:,:,1) <= max(1)) & (img(:,:,2) >= min(2)) & (img(:,:,2

% Clean up the mask using morphological operations
se = strel('square', 10);
blue_mask = imclose(imopen(blue_mask, se), se);

% Find connected components in the mask
cc = bwconncomp(blue_mask);

% Extract the location of the cubes as a list of centroids
stats = regionprops(cc, 'Centroid');
blue_cubes = cat(1, stats.Centroid);
disp(blue_cubes);
disp(size(blue_cubes));
% Display the image with the cubes marked
```

```
plot(blue_cubes(:,1), blue_cubes(:,2), 'bo', 'MarkerSize', 10, 'LineWidth', 2);
hold on;

% yellow
min = [0.6 0.6 0];
max = [1 1 0.4];
yellow_mask = (img(:,:,1) >= min(1)) & (img(:,:,1) <= max(1)) & (img(:,:,2) >= min(2)) & (img(:,:

% Clean up the mask using morphological operations
se = strel('square', 10);
yellow_mask = imclose(imopen(yellow_mask, se), se);

% Find connected components in the mask
cc = bwconncomp(yellow_mask);

% Extract the location of the cubes as a list of centroids
stats = regionprops(cc, 'Centroid');
yellow_cubes = cat(1, stats.Centroid);
disp(yellow_cubes);
disp(size(yellow_cubes));
% Display the image with the cubes marked

plot(yellow_cubes(:,1), yellow_cubes(:,2), 'yo', 'MarkerSize', 10, 'LineWidth', 2);
hold on;
```