lab 07

Syed Mustafa and Hazika Farooq

April 2023

1 Task 7.3

The object detection strategy useds in this lab is based on RGB colour spectrum. We used RGB threshold values of colours to detect our object. Initially we planned to incorporate the depth camera to get the top of the cube only, but since the depth function was not giving sccurate enough values, we did not use it in object detection solution.
the second reason why we don't need the middle location precisely is because we could get the pincher over it from the obtained coordinates and since the end-effector can expand to size much greater than the cube's size, so while closing the claw, the cube will somehow end up in the center of the claw and it would pick it up.

The solution is 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 yellow cubes.

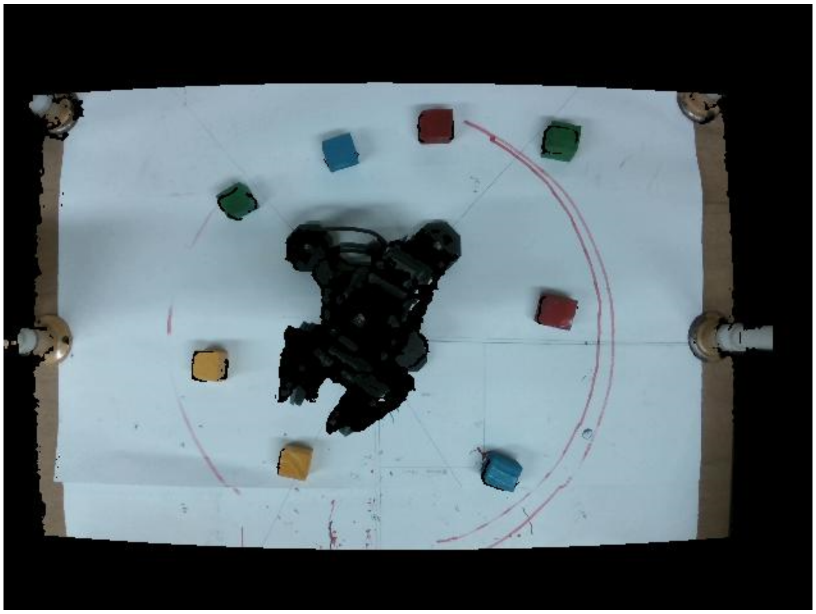

Figure 1: Demo Image

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. We also had to define the area of the image we want to work with and we removed the "cubes" the algorithm found outside of our workspace in the dark or in the center where our robotic arm lies.

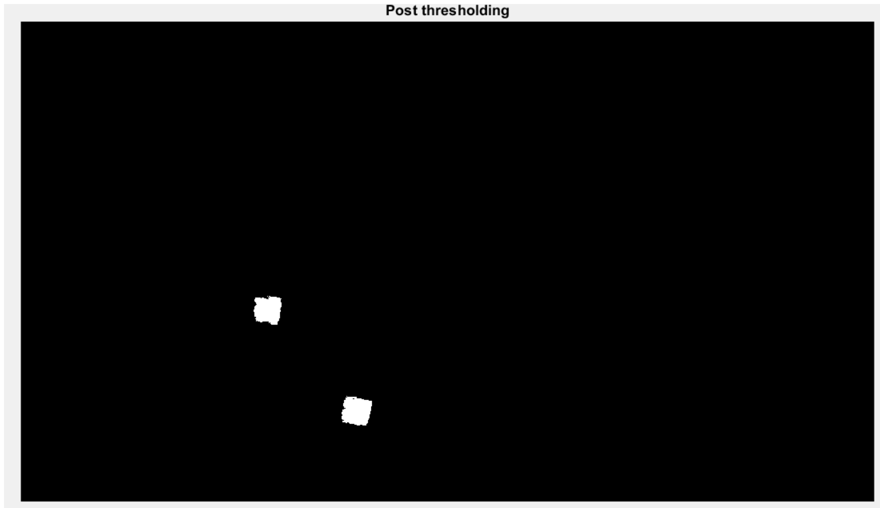

Figure 2: Setting the image threshold for Yellow

the final output we got framing these thresholding values on our image,

Figure 3: Caption

2 Code

```
% img = imread('testo1.png');
% subplot(3,2,1);
% imshow(img);
```

```
%
% img1 = rgb2gray(img);
% subplot(3,2,2);
% imshow(img1);
% min = [0.32 0.32 0.32];
% max = [0.38 0.38 0.38];
% red = img(:,:,1);
% green=img(:,:,2);
% subplot(3,2,3);
% imshow(red);
% % red_mask = (red(:,:,1) >= min(1)) & (red(:,:,1) <=
    max(1)) & (red(:,:,2) >= min(2)) & (red(:,:,2) <=
   max(2)) & (red(:,:,3) <= max(3)) & (red(:,:,3) >=
   min(3));
%
% final = imsubtract(green,img1); %take out red object
%
% final1 = im2bw(final, 0.2);
% subplot(3,2,4);
% imshow(final);
% subplot(3,2,5);
% imshow(final1);
% subplot(3,2,6);
% se = strel('square', 10);
% red_mask = imclose(imopen(final1, se), se);
%
% % Find connected components in the mask
% cc = bwconncomp(red_mask);
%
% % Extract the location of the red 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 red cubes marked
% imshow(img);
% hold on;
% plot(red_cubes(:,1), red_cubes(:,2), 'ro', '
   MarkerSize', 10, 'LineWidth', 2);



% img1 = imread('pic1.jpeg');
%
% red = img1(:,:,1);
```

```
% blue = img1(:,:,2);
% green = img1(:,:,3);
%
% d = impixel(img1);
%
% out = red>160 & blue <101 & blue >0 & green<110 &
    green>0;
% figure;
% imshow(out);
%
% out2 = imfill(out,'holes');
% out3=bwmorph(out2,'dilate',3);
% out3 =imfill(out3,'holes');
% figure;
% imshow(out3);

x = imread('pic1.jpeg');

% d=impixel(x)

% yellow final

yellow_x = ((x(:,:,1)>140) & (x(:,:,2)> 90) & (x
    (:,:,3)< 65));

figure;
% figure,imshow(yellow_x);
se = strel('square', 10);
yellow_mask = imclose(imopen(yellow_x, se), se);

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

% Extract the location of the red 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 red cubes marked
imshow(x);
hold on;
plot(yellow_cubes(:,1), yellow_cubes(:,2), 'yo', '
    MarkerSize', 10, 'LineWidth', 2);
```

```
%  green cube final

green_x = ((x(:,:,1)<30) & (x(:,:,2)> 60) & (x(:,:,3)<
     72));
% figure,imshow(green_x);
se = strel('square', 10);
green_mask = imclose(imopen(green_x, se), se);

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

% Extract the location of the red 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 red cubes marked

plot(green_cubes(:,1), green_cubes(:,2), 'go', '
    MarkerSize', 10, 'LineWidth', 2);
hold on

% blue final
blue_x = ((x(:,:,1)<70) & (x(:,:,2)< 115) & (x(:,:,3)>
     90));
% figure,imshow(blue_x);
se = strel('square', 10);
blue_mask = imclose(imopen(blue_x, se), se);

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

% Extract the location of the red 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 red cubes marked

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

```
% red final
red_x = ((x(:,:,1)>70) & (x(:,:,2)< 40) & (x(:,:,3)
    <60));
% figure,imshow(red_x);se = strel('square', 10);
se = strel('square', 10);
red_mask = imclose(imopen(red_x, se), se);

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

% Extract the location of the red 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 red cubes marked

plot(red_cubes(:,1), red_cubes(:,2), 'ro', 'MarkerSize
    ', 10, 'LineWidth', 2);
hold on;

% displaying all color images
maksyellow = im2bw(imcomplement(yellow_x));
mask3 = cat(3,maksyellow,maksyellow,maksyellow);
yellow_x_new = x;
yellow_x_new(mask3)= 0;
figure,
subplot(2,2,1)
imshow(yellow_x_new);
title('yellow blocks')

maksred = im2bw(imcomplement(red_x));
mask3 = cat(3,maksred,maksred,maksred);
red_x_new = x;
red_x_new(mask3)= 0;
subplot(2,2,2);
imshow(red_x_new);title('red blocks')

maksgreen = im2bw(imcomplement(green_x));
mask3 = cat(3,maksgreen,maksgreen,maksgreen);
green_x_new = x;
green_x_new(mask3)= 0;
subplot(2,2,3);

imshow(green_x_new);
```

```
title('green blocks')

maksblue = im2bw(imcomplement(blue_x));
mask3 = cat(3,maksblue,maksblue,maksblue);
blue_x_new = x;
blue_x_new(mask3)= 0;
subplot(2,2,4);
imshow(blue_x_new);
title('Blue blocks')
```

Different blocks displayed from the image using concatenate function.

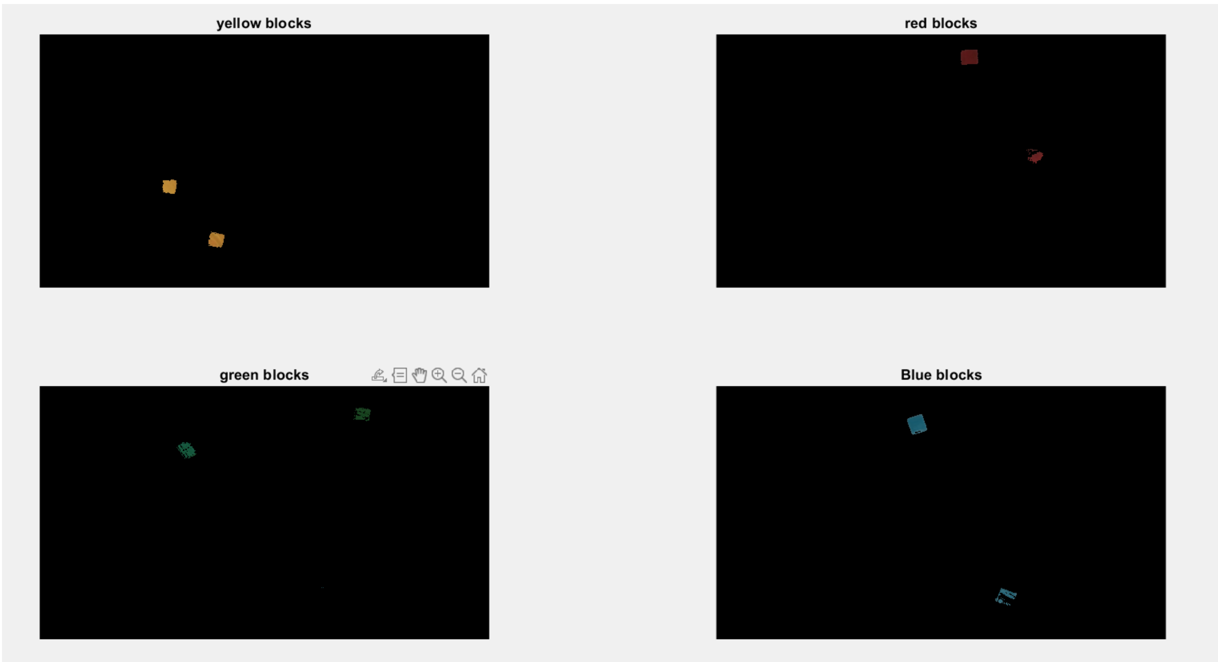

Figure 4: Extracted colored blocks in the image

3 Statistics

1. the code works regardless of how many objects of each colour were in the workspace as long as the same colour was not too close to another
2. The number of objects did not influence colour detection.
3. 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