MTRN4230 Robotics
Assignment 2
Runkai Zhao, z5146927
18/6/2020

# 1 PART 1

## 1.a

The height, width, and the number of channels of the image can be obtained from the "size" functiom or directly checking the "Workplace" window in MATLAB.

| | |
|---|---|
| Height | 684 pixels |
| Width | 912 pixels |
| Depth | 3 layers |

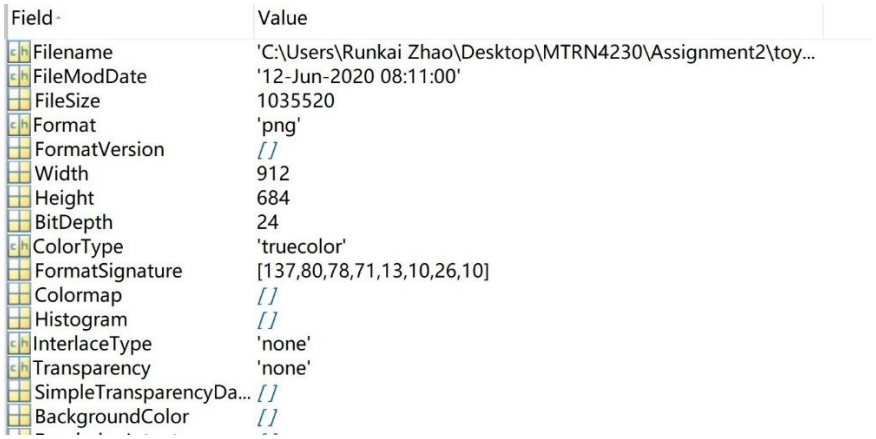


*Figure 1.a.1 Image Information*

## 1.b

By using MATLAB's Color Threshold tool, the RGB threshold value of the blue cup can be obtained. Then, transferring it into a binary mask as shown below.

*Figure 1.b.1 Binary mask of cup*

## 1.c

By using MATLAB's Color Threshold tool, the RGB threshold value of the white ball can be obtained. Then, transferring it into a binary mask as shown below.

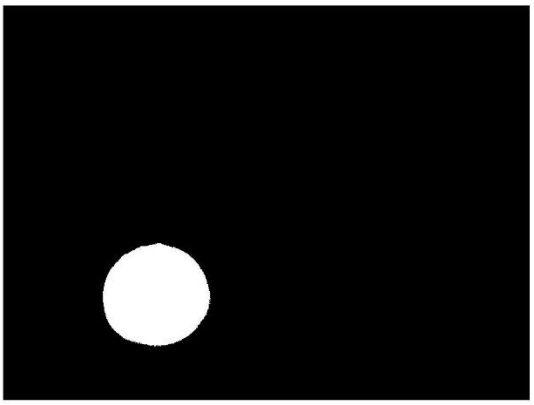

*Figure 1.c.1 Binary mask of ball*

## 1.d

After the step B and C, it is straightforward to get a combined binary mask of cup and ball through bitwise operation "OR", then directly multiplying the binary mask with the image matrix. The part the image with logical "1" will have remained as the below figure, and other parts will be turned to dark.

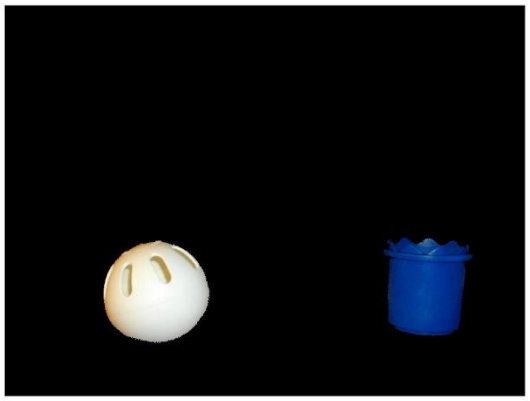

*Figure 1.d.1 Color mask of ball and cup*

# 2 PART 2

## 2.a

*Figure 2.a.1 SURF descriptor of "chess_knights_run.png"*

## 2.b

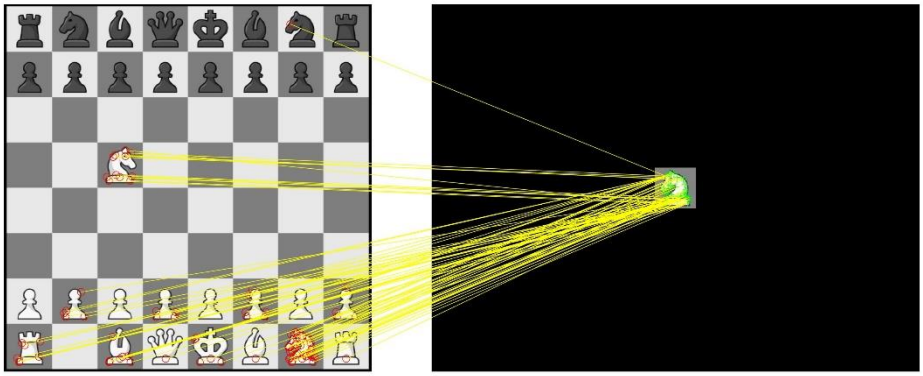

*Figure 2.b.1 Matched pointers*

## 2.c

In order to determine the approximate locations of these two knights, the distance between two matched features, returned by “matchFeatures” functions, can be applied to remove those disinterested matched points. After selection and filtering, the new feature matching figure is displayed as below,

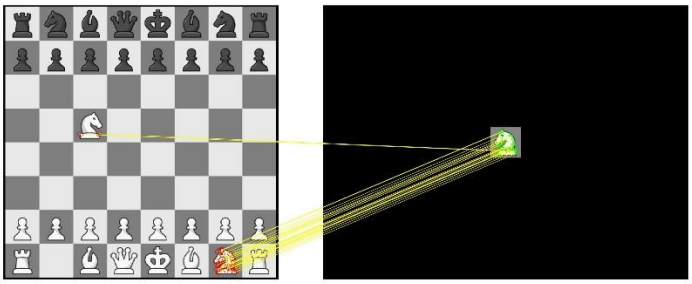

*Figure 2.c.1 New Matched Points*

Then, get the x and y locations of each matched points around these two knights. Calculating the average values of them which are the approximate coordinate of two knights. Finally, the true distance by using the distance formula is 693.4435.

# 3 PART 3

## 3.a

The steps are listed to highlight the centers of the orange’s slices:

1. Loading image from the local file.
2. Converting the image into a binary image.
3. Modifying the binary image by the “imerode”, “imclose”, “bwareaopen”, and “bwareafilt” functions, changing the parameters of functions to get a figure displaying the centers only.
4. By the “Circularity” of image regions returned from the “regionprops” function, to remove some disinterested area and to identify the locations of the white centers.
5. Marking out the white centers as green stars.

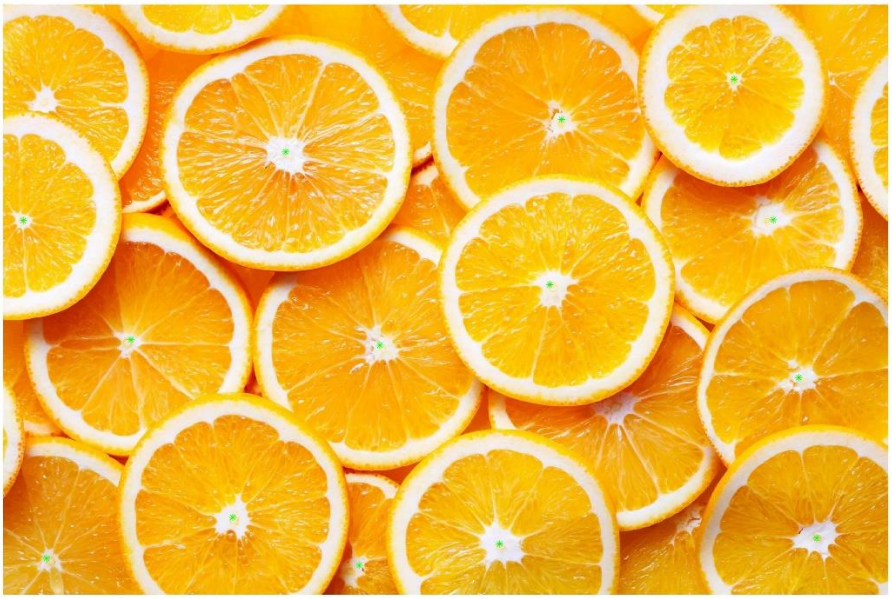

*Figure 3.a.1 Marking out the white centers*

## 3.b

The steps are listed to highlight the boundaries of the orange's slices:

1. Loading image from the local file.
2. Converting the image into a binary image.
3. Modifying the binary image by the "imerode", "imclose", and "bwareaopen" functions, then changing the parameters of "strel" function to display a better image.
4. By the "Circularity" and "Area" of image regions returned from the "regionprops" function, to identify the locations of the white centers, then setting the pixels around the centers to dark.
5. Removing some noises through the "imopen" functions.
6. Reversing the black and white in this image by the "imcomplement" function to get a mask.
7. Covering the real image with the binary mask.

*Figure 3.b.1 Marking the boundaries of the orange slices*

## 3.c

The steps are listed to create the masks for the top slices:

1. Applying the RGB threshold value of the white ball to get an image showing the interested regions.
2. Modifying the binary image by the “imerode”, “imclose”, and “bwareaopen” functions, then changing the parameters of “strel” function to display a better image without the white noise points.
3. Using the “padarray” functions to adjust the structure of image matrix.
4. Filling the holes of the image by the “imfill” function and resize the image matrix.
5. Removing some noise points, then get the mask.
6. Covering the real image with the binary mask.

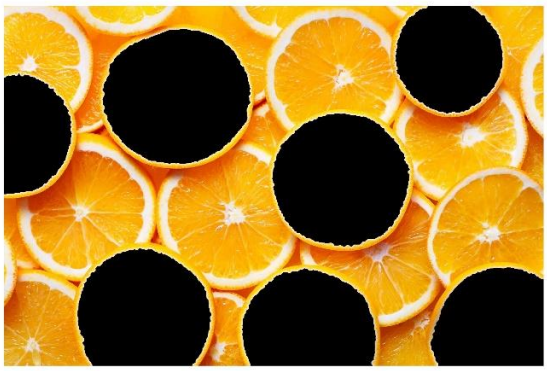

*Figure 3.c.1 Masks for the top slices*