

MTRN4230 Robotics ASSIGNMENT 2

AIMS

The following assessment aims are derived from course learning outcome 4:

1. Knowledge of basic image processing techniques for robotics.
2. Demonstrate the ability to combine basic image processing techniques to achieve a given outcome.

DUE DATE

<i>Assessment criteria</i>	<i>Due date and submission requirements</i>	<i>Deadline for absolute fail</i>	<i>Marks returned</i>
<i>Upload completed assignment + code</i>	<i>Week 3: 11:59pm, Friday via Teams</i>	<i>Week4:11:59pm, Wed.</i>	<i>1 week after submission</i>

INSTRUCTIONS

1. Download all images relevant to this assignment from course resources.
2. Assignment submission should consist of two files:
 - a. A report (in **.pdf format**) describing your step by step approach for each problem, answers, and corresponding image results,
 - b. A single Matlab script which contains all necessary code used for the assignment.
3. Develop your Matlab script in the same folder as the images and avoid using absolute paths (i.e.: Always use **imread('image.jpg')** instead of **imread('C:/absolute/path/image.jpg')**). This allows tutors to check your code efficiently.

ACTIVITIES

1. Part 1: Complete the following tasks using 'toysflash.png':
 - a. Use a Matlab function to find the height, width, and number of channels of the image. (1 point)
 - b. Write a code to generate a binary mask which covers the blue cup in the image. Display this binary mask in a separate image. You can use Matlab's **Color Thresholder** tool **ONLY** for the purpose of identifying the suitable color threshold values. You must write your own code to generate the final image. (2 point)
 - c. Similarly create a binary mask covering the plastic ball at the front of the image. Display this binary mask in a separate image. (1 point)
 - d. Generate a colour mask containing both objects in (b) and (c) as shown in Fig. 1. Briefly describe how you achieved this goal. (1 point)

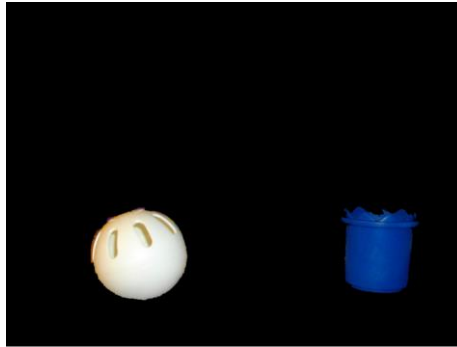


Figure 1 Colour mask for the cup and the plastic ball.

2. Part-2: Complete the following tasks using 'chess_knights_run.png' and 'knight.png':
 - a. Plot all SURF descriptors overlaid on 'chess_knights_run.png'. (1 points)
 - b. Match SURF descriptors from 'knight.png' to 'chess_knights_run.png'. Visualise the matched pairs. (3 point)
 - c. Using these two images and the feature matching concept, write an algorithm to find the distance between the two white knights on the chess board. The algorithm should work on any image where both white knights are on dark squares. Any answer within +/- 30 pixels from the true distance is accepted. In the report, describe how you solved this problem with corresponding images results. (3 points)
3. Part 3: Complete the following tasks using 'oranges.jpg' image: (8 points)
 - a. Write a code to identify the centres of each orange slice. Display markers on the centre of each orange slice. For full marks, you should identify at least 12 centres with no more than 2 incorrect centres. In the report, briefly write down the steps you followed to achieve this task with corresponding image outputs. (3 point)
 - b. Detect the boundaries of the orange slices. Display the boundaries overlaid on the original image. Figure 2 shows answers accepted as a correct boundary. A boundary is considered correct, if the boundary covers at least 80% of the visible boundary of a slice. For full marks, boundaries corresponding to at least 11 slices should be identified with no more than 7 incorrect boundary lines. See Figure 3 for an example solution. Briefly write down the steps you followed to achieve this task along with corresponding images. (3 points)



Figure 2: Any result closer to the edge (in green) of the slice is considered correct. To be accepted as a correct boundary, at least 80% of the visible border should be identified.



Figure 3: A sample solution. Incorrect boundary lines are circled. Question marks show missing boundary lines.

- c. Create a binary mask for the seven slices on the top layer (The seven slices are marked in Figure 4). For full marks, masks for at least 5 slices should be displayed with no more than one incorrect mask. Masks should fully cover the slice. Describe your approach briefly along with corresponding images. (2 points)

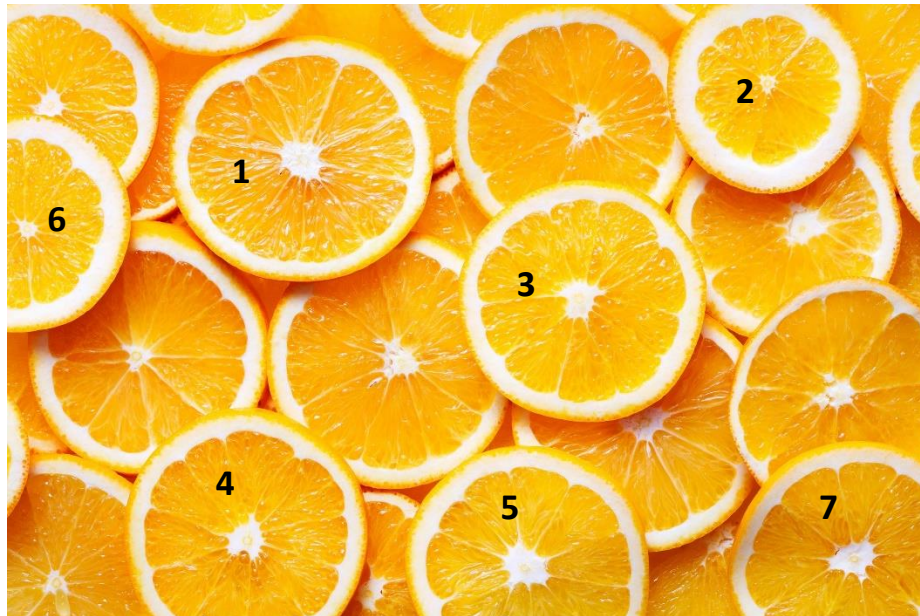


Figure 4: The seven slices on the top layer.

MARKING CRITERIA

Overall mark for this item is 20%. It has been distributed as below:

Part1	5%
Part2	7%
Part3	8%

Resources

Use **Color Thresholder** tool to identify thresholding values for each colour channel.

Hints: **imread, imshow, imerode, imclose, imfill, edge, detectSURFFeatures, extractFeatures, matchFeatures, bwareaopen, imbinarize, rgb2gray, hist3, find, imadjust, bwareaopen, strel, regionprops, bwconncomp**

IMPORTANT: Above function list does not include everything you need. You are free to use any other function as well.

Use the course discussion forum for clarifications and getting help.