MTRN4230 Robotics Asst1 Description

Asst1 requires you to work individually to process images of the Qwirkle blocks in the robot cell and detect their position. It builds on the fundamentals of image processing in PSE1. It is worth 15 course marks in total.

Aim
To understand how to design a robust image processing algorithm.

Method
The robot cell contains two rigidly mounted cameras which are used to identify the locations of objects on the table and the conveyor. This assignment requires you to use images from the camera mounted over the table and identify the following properties of each object within view:
{x, y, theta, colour, shape, upper_surface, reachable}

- (x, y, theta) = position of centroid of block in image coordinates with the orientation shown in Figure 1 defined in radians in image coordinates as shown in Figure 2. Units of pixels and radians (range of -pi/4 to pi/4).
- colour = {1 = red, 2 = orange, 3 = yellow, 4 = green, 5 = blue, 6 = purple, 0 = inverted}
- shape = {1 = square, 2 = diamond, 3 = circle, 4 = club, 5 = cross, 6 = star, 0 = inverted}
- upper_surface = indicate which side of the block is facing up. The coloured side is side 1, the black underside (inverted) is side 2. Blocks will not be placed on their edges.
- reachable = is the centroid able to be reached by the robot gripper? The three points marked on the table with coordinates specified as robtargets (see Moodle) lie on the arc which defined the maximum reach of the gripper. Any object with x and y coordinates within that arc is considered reachable, irrespective of the z coordinate. 1 = true, 0 = false.

Ignore camera calibration for the purpose of this assignment.

You are welcome to take more sample images in the robot cell for your own testing purposes. The following lines of Matlab code will achieve this, but may not display the images in the same orientation as shown in Figure 1.

```
vid = videoinput('winvideo', 1, 'RGB24_1600x1200');
frame = getsnapshot(vid);
imshow(frame); % To display a single frame
preview(vid); % To display a live video stream
```

The images will be 1600x1200 resolution and a training / sample set will be available on Moodle soon.

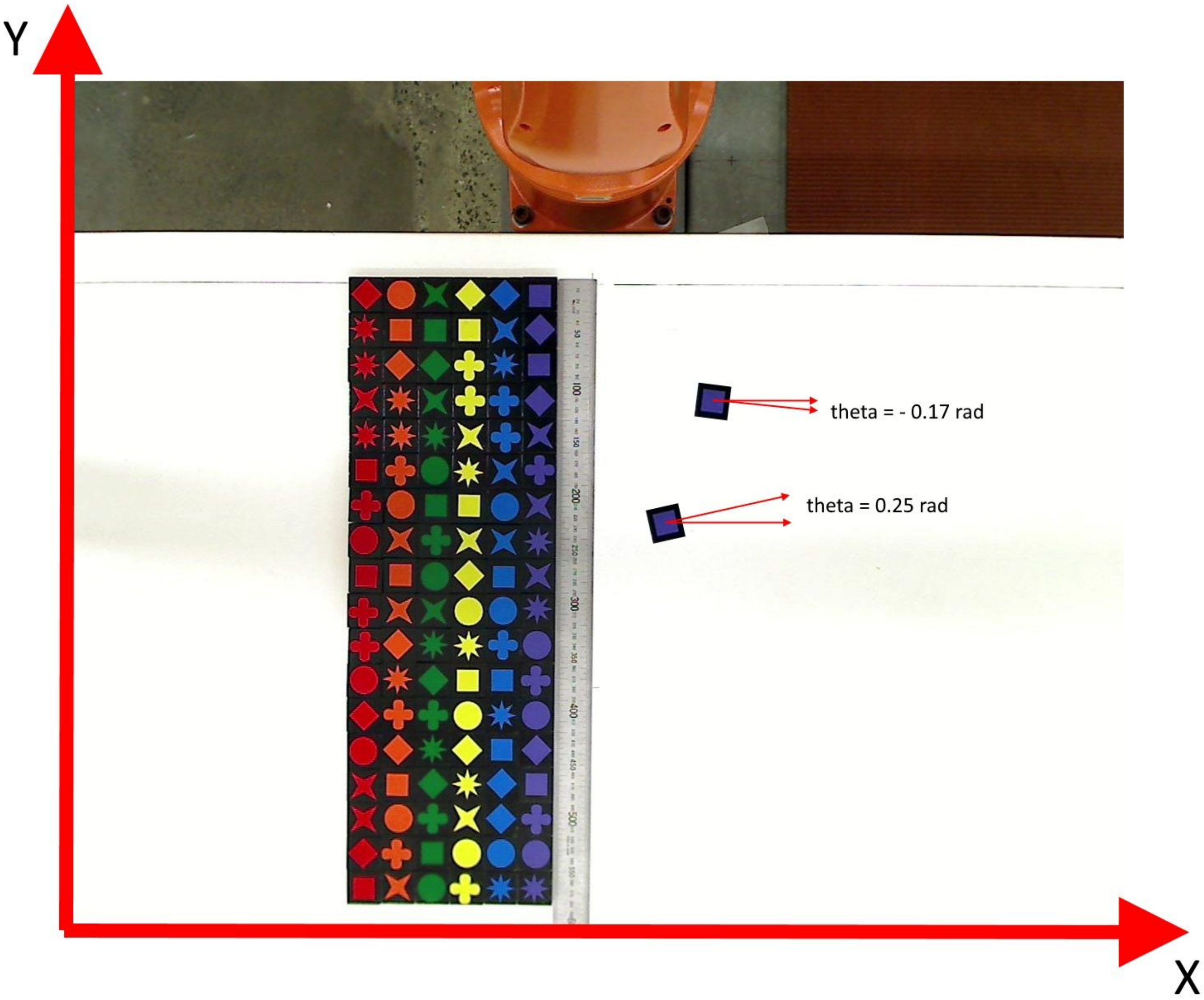

Figure 1: Coordinate system for defining block orientations, with the origin at the centroid. Multiple orientations are considered equivalent due to symmetry.

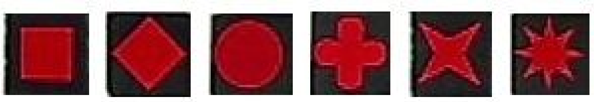

Figure 2: Shape definitions {1 = square, 2 = diamond, 3 = circle, 4 = club, 5 = cross, 6 = star, 0 = inverted}

Resources

Use the course discussion forum for getting help along with the computer vision lecture notes.

Assessment

Submit a single .zip file named zXXXXXXX_MTRN4230_ASST1.zip to Moodle containing all Matlab files you need. When unzipped this should contain a function called zXXXXXXX_MTRN4230_ASST1.m which takes four string arguments. The first argument is a full file path and name for an input image. The second is just the input file image file name. The third is a full file path and name to write the results to. The fourth is the folder that your function is running in. When called, this function should open the input image, process it to calculate the information, open the results file and write the results before closing both files. See the skeleton function on the course homepage.

The results should be written in plain text in the following format, again see the skeleton function on the course homepage for the exact format.

```
>>>
image_file_name:
filename.jpg
blocks:
x y theta colour shape upper_surface reachable
…
<<<
```

For example

```
image_file_name:
IMG_0002.jpg
blocks:
210.00 300.00 -0.170 6.000 1.000 1.000 1.000
200.00 250.00 0.250 6.000 1.000 1.000 1.000
```

Computation time

The automarking script will operate by calling your function once per test image and comparing the output with a manually labelled output. Up to 150 test images will be used for which a total maximum processing time of 10 minutes will be allowed per student, at which time the marking process will be stopped and your mark calculated from any completed function calls. The marking process will progress from images at the lowest level of difficulty to those at the highest level of difficulty and will be carried out on one of the lab machines in J18-212. Make sure your code works on these machines.

The automarking script will seed the random number generator in Matlab to ensure consistency of results by calling `rng(0, ‘twister’)`. More help is available here. This will aid solutions using RANSAC or other randomised methods.

Make sure that you make use of `try`/`catch` commands in Matlab, because if your script throws an error on one image, the automarking script may not be able to continue to test your code on further images.

Marking Criteria

Images will be classified into the following categories

Difficulty level 1 - images with no blocks touching each other, but with the possibility of some inverted blocks. 50% of the total marks for this assignment will be available in this category.

Difficulty level 2 - images with blocks with the coloured side facing up or inverted, placed in any position or orientation but with the largest side flat on the table. This means blocks are not stacked, but may be adjacent. 50% of the total marks for this assignment will be available in this category.

The mark for each category will be the average from all images in that category.

For each block your result claims is in a single image:

4 points - accurate position and orientation. The multiple coordinate frame orientations that are possible for each block due to symmetry are considered equivalent.

1 point - colour correctly identified

1 point - shape correctly identified

1 point - upper_surface correctly identified

1 point - reachable correctly identified

This total number of points will be divided by 8 and by the actual number of blocks in the image and expressed as a percentage score for that image. For inverted blocks, the colour and shape should be set to 0.

A maximum of 36 blocks will be visible in a single image.

Due date

11:55pm Friday 1 September (week 6) to Moodle.