MTRN4230 Robotics Asst1 Description

Asst1 requires you to work individually to process images of blocks in the robot cell and detect their position. It builds on the fundamentals of image processing in PSE1. It is worth 20 course marks in total.

Learning Outcome

- 4. Apply and evaluate image processing techniques in robotics

Method

The robot cell contains two rigidly mounted cameras which are used to identify the locations of blocks on the table and the conveyor. Blocks will either contain a letter or coloured shape and will be placed in any position or orientation with the largest side flat on the table. This means blocks are not stacked, but may be adjacent. This assignment requires you to use images from the camera mounted over the table, detect all blocks visible on the table and identify the following properties of each block within view:
{x, y, theta, colour, shape, letter, 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. For the letters, theta will be in the range of -pi to pi. For the shapes, theta will be in the range of -pi/4 to pi/4.
- colour = {0 = letter (white), 1 = red, 2 = orange, 3 = yellow, 4 = green, 5 = blue, 6 = purple}
- shape = {0 = letter, 1 = square, 2 = diamond, 3 = circle, 4 = club, 5 = cross, 6 = star}
- letter = {all upper case letters of the alphabet represented as numbers with 1 = A, 2 = B, …, 26 = Z, 0 = shape (not a letter)}
- 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.r

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. Code to capture images is on Moodle (currently under “Important Documents”).

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

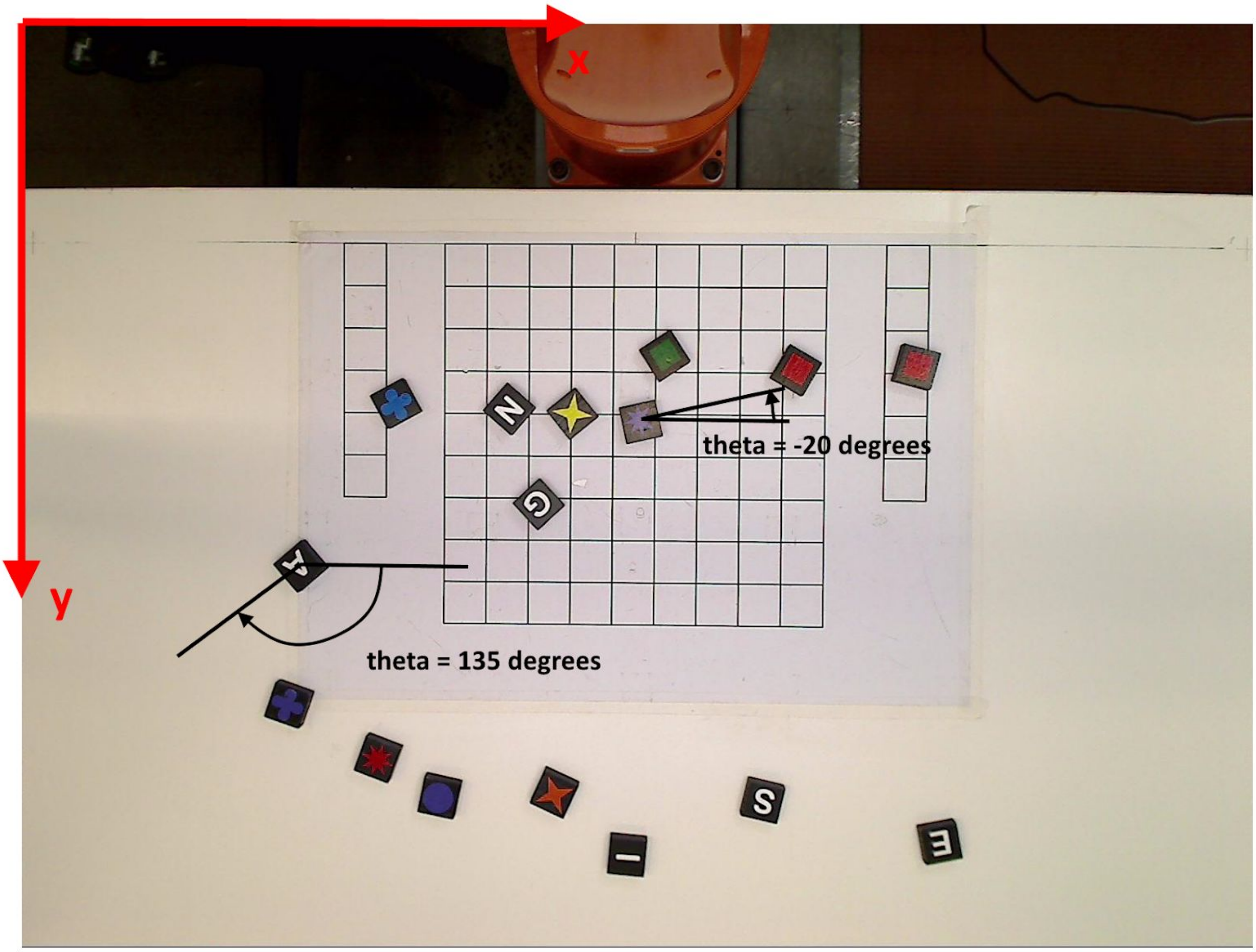

Figure 1: Coordinate system for defining block orientations, with the origin at the centroid. Multiple orientations are considered equivalent due to symmetry. Note that actual values for orientation need to be specified in radians.

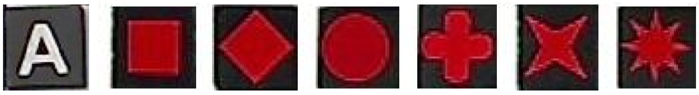

Figure 2: Shape definitions {0 = letter, 1 = square, 2 = diamond, 3 = circle, 4 = club, 5 = cross, 6 = star}

Resources

Use the course discussion forum for getting help along with the computer vision lecture notes. Previous students who have attempted similar assignments have had the most success when convolutional neural networks are used for classification.

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
rectangles:
x y theta colour shape letter reachable
…
<<<

For example

```
image_file_name:
IMG_001.jpg
rectangles:
421.743582 598.845489 -2.525727 0.000000 0.000000 25.000000 1.000000
470.706507 466.645607 -1.457361 0.000000 0.000000 17.000000 1.000000
728.811077 435.868910 0.863727 0.000000 0.000000 5.000000 1.000000
753.354512 572.507366 -0.249312 0.000000 0.000000 23.000000 1.000000
835.387949 673.955530 -1.534449 0.000000 0.000000 14.000000 1.000000
```

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 100 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. 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.

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.

Marking Criteria

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

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

1 point - colour correctly identified

1 point - shape correctly identified

1 point - letter correctly identified

1 point - reachable correctly identified

This total number of points will be divided by 7 and by the actual number of blocks in the image and expressed as a percentage score for that image. A maximum of 40 blocks will be visible in a single image. The automarker will only award points for the first N blocks, where N are the actual number of blocks in the image.

Due date

11:55pm Friday 31 August (week 6) to Moodle.

No late submissions will be accepted without special consideration - see the course outline.