MTRN4230 T2 2023 Project 2: Path Planning using sensor feedback!

1. Learning Outcomes

- Apply concepts in computer vision to enable the robot to interact with its environment.
- Apply coordinate transformations to convert between camera and robot frames.
- Utilise path planning to enable the robot to complete tasks successfully.

2. Due Date

- Demonstration during your normal lab time in week 12 (see assignment release announcement if you have an exam clash)

3. The Brief

The objective of this task is to integrate computer vision with the robotic arm to achieve a more advanced pick and place operation. You will use the top-down view provided by the webcam to:

1. Identify the location of the game board in reference to the robot base joint.
2. Identify the obstacle pieces and the player piece and their positions in reference to the robot base joint. (They will be colour coded to simply identification)
3. You will need to move the obstacle pieces from their current position to the correct positions based on a provided occupancy grid / matrix. Various difficulties of challenges can be completed.
4. You must use the bug2 algorithm to move the player piece from the source location to the target location.

Part A: Computer Vision

You will need to use the Computer Vision Toolbox from MATLAB and the supplied USB webcam (see here for MATLAB interface) to complete the following:

1. Use the top-down view provided by the webcam to identify the four corners of the game board. Please refer to Appendix A for the orientation of the corners, and an example of the gameboard and overall layout.
2. Calculate the homography matrix and compete a projective transform. This is required to match up the perspective of the 2-dimensional image of the game board with that of the real-world gameboard.

3. Obtain the pose of each of the corners in reference to the robot's base joint. You have been provided with 4 ground truth coordinates (see appendix A) to complete the perspective transform. You can instruct the robot to move to each of the corners to confirm that you have correctly calculated the pose.
4. Using colour space image segmentation techniques to identify the location of each of the obstacle and player piece on the gameboard. An example image of the obstacles can also be seen in the Appendix A.

By the end of this part of the assessment you should be able to successfully output the pose (and move to) of every game piece on the board.

You have been provided with some sample images that you can use to develop your computer vision solution at home. These images are from the webcam.

A note on colour spaces for colour thresholding has been added in Appendix B. This should be reviewed as an addition to the approach taught in lectures.

Part B: You're A-Maze-Ing!

Part B has varying degrees of difficulty. You are required to attempt and submit a solution of one of the following challenges. You can also start from the beginner challenge and work their way up to the most difficult challenge if they so wish (they build on top of each other).

Maze Novices (Credit - Distinction)

For this challenge, you will be provided with an occupancy grid containing the game board state. The obstacles on the game board will be placed in the correct location. You are required to use the bug2 algorithm to solve the maze and move the player piece from the source location to the target location.

An example grid you could be provided:

```
   grid = [
1 1 1 1 1 1 1 1 1 1
1 3 1 4 0 0 0 0 0 1
1 0 0 1 2 2 0 0 0 1
1 0 0 0 0 0 0 0 0 1
1 0 0 0 0 1 2 0 0 1
1 0 0 0 0 0 1 0 0 1
1 1 1 1 1 1 1 1 1 1]
```

Table 1 Specification of the Digits and their meanings

Digit	Meaning
0	Empty Space
1	Wall or Red Obstacle
2	Blue Obstacle
3	Player Piece
4	Destination

Maze Masters (Distinction)

For this challenge, the obstacles on the physical gameboard are not in their correct locations. You will be provided with an occupancy grid specifying the correct location of the obstacles. The occupancy grid may need to be rotated for it to align perfectly with what is visible on the table. You will need to move the obstacle pieces to their correct locations, before using the bug2 algorithm to solve the maze.

The gameboard has 3 different classes of obstacle pieces. Each class has its own restriction, and it must be followed when completing the challenge.

Table 2 Classes of Obstacles and their movement restrictions

Obstacle Class	Restriction
Red	Obstacles cannot be moved.
Blue	Obstacles can be lifted and moved.

Maze Monarchs (High Distinction)

This challenge builds up from the Maze Master challenge. The only addition is that your solution should be robust to any dynamic changes of the gameboard. This means that the demonstrator will be allowed to move obstacles during the operation of your program. Your program should be able to identify these changes and move the obstacles back to their correct positions. The player piece will not be moved, only the obstacle pieces. If an obstacle piece is moved while the player piece is being moved (i.e bug2 algorithm is running), the obstacle piece must still be moved back to its correct location. How this will be achieved is up to you.

Part C: Discussion

During the final demonstration of the assessment, you will need to answer any questions asked regarding the implementation of your solution. These questions can be about specifics in your implementation, challenges you faces or the general understanding of theory about computer vision or robot kinematic techniques applied.

Part D: Extension (Optional)

Maze Conquerors (BONUS CHALLENGE for those who dare (2 marks))

For this challenge you are only provided with the destination position of the player piece. you must be able to accept this coordinate as input into your program from the command terminal. The positions of the obstacles may allow for the player piece to be moved to the destination position using the bug2 algorithm or they may not allow it. You will decide which obstacle pieces to move and where such that they can move the player piece to its destination.

The same restrictions to the movement of the obstacles as specified in Table 2.

This challenge is worth 2 bonus marks. Including the bonus marks total marks for this assessment is capped at 21 Marks.

GOOD LUCK!

4. Marking Criteria

For the in-person demonstration assessment, you are not required to complete the individual components of part A below IF you have completed part B Maze Masters or Maze Monarchs. This is due to the necessity of implementing Part A completely for these to work.

If you are unable to demonstrate Part B successfully, demonstrators will ask you to demonstrate tasks from Part A to be assessed as described below.

This assignment is worth a total of 20 % of your course mark. The marking criteria for both Parts A – D are provided below.

Part A – 6 %:

Part A will only need to be explicitly demonstrated if you are unable to successfully complete Part B.

Objective 1: Identify the Four Corners and of the Game Board

1. Open up a window in MATLAB of image taken by the webcam and annotate (letters/colours/symbols etc) the four corners.
   a. **Full Marks:** Identify all 4 corners.
   b. **Partial Marks:** 2 or 3 corners successfully identified.
   c. **No Marks:** Fewer than 2 corners have been identified.
2. The orientation and label of each corner outlined in appendix A must match the label given in the student's code. I.e., Corner 3 of the gameboard in appendix A must match corner 3 in the students code.
   a. **Full Marks:** Correctly labelled all 4 corners.
   b. **Partial Marks:** 2 or 3 corners correctly labelled.
   c. **No Marks:** Fewer than 2 corners have been correctly labelled.

Objective 2: Identify Location of Obstacle and Player Pieces on the Gameboard Using Colour Space Image Segmentation Techniques

1. On the same window opened for "Objective 1" use colours/symbols to identify the game pieces
   a. **Full Marks:** All obstacle and player pieces are correctly identified.
   b. **Partial Marks:** Some but not all obstacle and player pieces are correctly identified.
   c. **No Marks:** No or very few obstacle and player pieces are correctly identified.

Objective 3: Calculate Homography Matrix and Perform Projective Transform

1. Open up a window in MATLAB after having applied a projective transform on the image taken by the webcam. The following will be assessed.
    a. **Full Marks:** (All must be satisfied)
        i. Corner 1 must be in top left of the screen.
        ii. Corner 2 must be in top right of the screen.
        iii. Corner 3 must be in the bottom left of the screen.
        iv. Corner 4 must be in the bottom right of the screen.
        v. The displayed projective transformed gameboard must be square with the screen, with its edges relatively parallel to the screen. (> 5 degrees of rotation)
        vi. The obstacle and play pieces must also be correctly projective transformed.
    b. Partial Marks:
        i. The displayed projective transformed gameboard has some rotation, but the identified corners themselves are reasonably within the correct regions.
        ii. The displayed projective transformed gameboard is square with the screen, with its edges relatively parallel to the screen but the identified corners are not in the correct locations. (I.e flipped upside down)
        iii. The projective transform applied to the obstacle and/or player piece has some misalignment in the perspective.
    c. No Marks:
        i. Projective transform has not been implemented correctly. Corners are not in their correct locations and the displayed gameboard has significant rotation or skew.

Objective 4: Obtain Poses

1. Validate the calculated corner poses by instructing the robot to move to each corner and confirming their correctness.
    a. Full Marks:
        i. The robot successfully moves to each corner and confirms the correctness of the calculated poses. The movement can be issued manually, without the use of an algorithm if needed.
    b. Partial Marks:
        i. The robot successfully moves to some of the corners and confirms the correctness of some of the calculated poses. The movement can be issued manually, without the use of an algorithm if needed.
    c. No Marks:
        i. The robot fails to move to any corner, or the calculated poses are incorrect.

2. Validate the calculated obstacle and player piece poses by instructing the robot to move to selected pieces as specified by the demonstrator.
    a. Full Marks:
        i. The robot successfully moves to each piece specified by the demonstrator and confirms the correctness of the calculated poses.
    b. Partial Marks:
        i. The robot successfully moves to some of the pieces specified by the demonstrator and confirms the correctness of some of the calculated poses.
    c. No Marks:
        i. The robot fails to move to any game piece, or the calculated poses are incorrect.

Part B – 10 %

Note: students will be awarded only be awarded with the percentages for one of the following subsections to make up the 10 % for part B. They will be assessed on the criteria for the maximum spec they have completed (this will be asked at the start of the assessment).

Maze Novices Marking Criteria (5 %):

Objective 1: Implement the Bug2 algorithm to solve the maze and navigate the player piece from the source location to the target location.

a. **Full Marks:**
    i. The Bug2 algorithm is correctly implemented, and the player piece successfully reaches the target location.
    ii. The player piece successfully reaches the target location without colliding with any obstacles.
b. **Partial Marks:**
    i. The Bug2 algorithm is partially implemented, and the player piece partially navigates towards the target location but does not reach it.
    ii. The player piece partially reaches the target location but does collide with some obstacles.
c. **No Marks:**
    i. The Bug2 algorithm is not implemented, or the player piece does not make any progress towards the target location.
    ii. The player piece does not reach the target location or collides with too many obstacles.

Maze Master's Marking Criteria (7 %):

Mazes master's marking criteria builds up on the marking criteria specified for the Maze Novices component of the task. Please also refer to that criteria as well.

Objective 1: Move the Obstacle Pieces to Their Correct Locations

a. **Full Marks:**
    i. All obstacle pieces are correctly moved to their respective correct locations, following the movement restrictions for each obstacle class.
b. **Partial Marks:**
    i. Some but not all obstacle pieces are correctly moved to their correct locations, or the movement restrictions for some obstacle classes are partially followed.
c. **No Marks:**
    i. The obstacle pieces are not moved to their correct locations, or the movement restrictions for the obstacle classes are not followed.

Maze Monarch's Marking Criteria (10 %):

Mazes Monarch's marking criteria builds up on the marking criteria specified for the Maze Novices and Maze Master's component of the task. Please also refer to both other criteria.

Objective 1: Solution is responsive to any dynamic changes within the gameboard.

a. **Full Marks:**
    i. The algorithm is able to move every obstacle piece moved by the demonstrator back to the correct location.
b. **Partial Marks:**
    i. The algorithm is able to move some but not all obstacle pieces moved by the demonstrator back to the correct location.
c. **No Marks:**
    i. The algorithm is able to no obstacle piece moved by the demonstrator back to the correct location.

Part C – 4 %

Poor (0 – 1 %)	Insufficient (1 – 2 %)	Developing (2 – 3 %)	Accomplished (3 – 4 %)
(i) The student demonstrates a lack of understanding of their solution and its implementation. (ii) The student's responses are vague, irrelevant, or show a significant misunderstanding of key concepts.	(i) The student demonstrates a limited understanding of the solution and its implementation. (ii) The student's responses lack depth and fail to provide comprehensive explanations.	(i) The student demonstrates a satisfactory understanding of the solution and its implementation. (ii) The student's responses are generally accurate but may lack some details or clarity.	(i) The student demonstrates a thorough understanding of the solution and its implementation. (ii) The student's responses are comprehensive, clear, and provide in-depth explanations.

Part D – 2 %

Reminder: this is a bonus 2 % that caps the total assignment mark at 21/20

Maze Conquerors Marking Criteria (BONUS CHALLENGE)

Objective 1: Decide Which Obstacle Pieces to Move and Where

a. **Full Marks:**
   i. The student makes the correct decisions regarding which obstacle pieces to move and their new placements, considering the movement restrictions for each obstacle class.

b. **Partial Marks:**
   i. The student partially makes correct decisions regarding which obstacle pieces to move and their new placements, or some movement restrictions are partially followed.

c. **No Marks:**
   i. The student does not effectively decide which obstacle pieces to move or does not consider the movement restrictions for the obstacle classes.

Objective 2: Move the Obstacle Pieces to Their Correct Locations

a. **Full Marks:**
   i. All obstacle pieces are correctly moved to their respective correct locations, following the movement restrictions for each obstacle class.

b. **Partial Marks:**
   i. Some but not all obstacle pieces are correctly moved to their correct locations, or the movement restrictions for some obstacle classes are partially followed.

c. **No Marks:**
   i. The obstacle pieces are not moved to their correct locations, or the movement restrictions for the obstacle classes are not followed.

Objective 3: Implement the Bug2 algorithm to solve the maze and navigate the player piece from the source location to the target location.

a. **Full Marks:**
    i. The Bug2 algorithm is correctly implemented, and the player piece successfully reaches the target location.
    ii. The player piece successfully reaches the target location without colliding with any obstacles.
b. **Partial Marks:**
    i. The Bug2 algorithm is partially implemented, and the player piece partially navigates towards the target location but does not reach it.
    ii. The player piece partially reaches the target location but does collide with some obstacles.
c. **No Marks:**
    i. The Bug2 algorithm is not implemented, or the player piece does not make any progress towards the target location.
    ii. The player piece does not reach the target location or collides with too many obstacles.

Appendix A

Game Board Specifications:

An image of the gameboard is shown below. It is an 8 x 5 grid with each piece being an 60 mm by 60 mm square and an overall board size of 380 mm by 590 mm. The corners of the gameboard will be yellow. When viewing the webcam image, the corners of the gameboard and ground truth locations will lie in the respective image quadrant.

Five different colours are used for completion of computer vision: purple (ground truth), orange (board corners), green (player piece), red, and blue (obstacles).

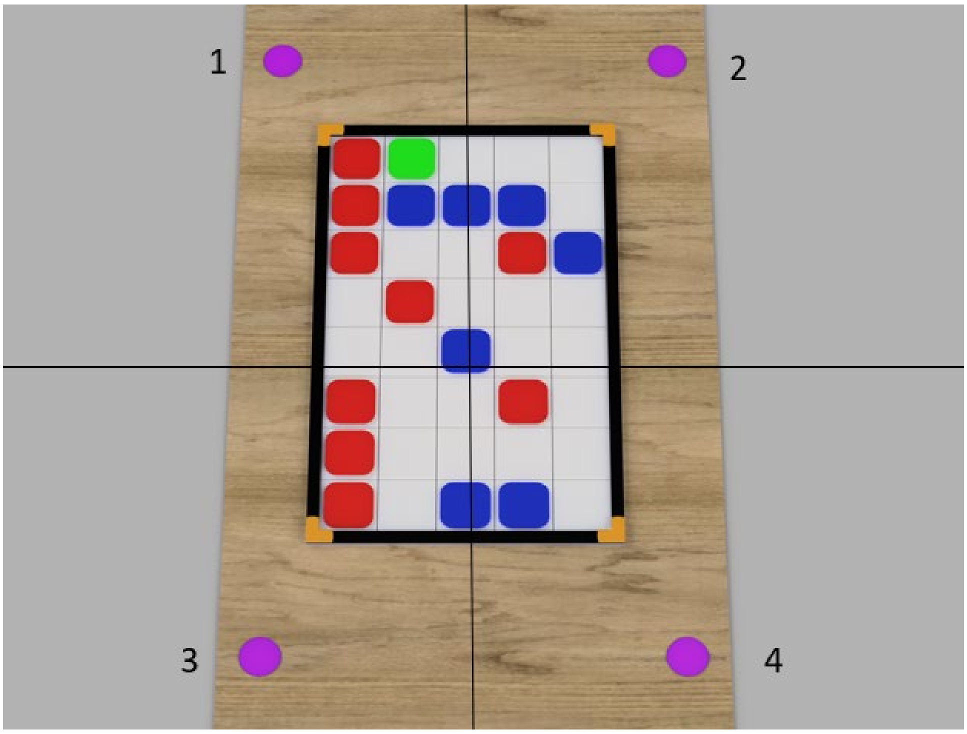

Figure 1: Rendering of sample game board layout on table as viewed from webcam

Ground Truth Coordinates:

Four ground truth coordinates translate directly to global coordinates as defined in the table below. They will all be of the same purple colour. There will be one ground truth coordinate visible in each quadrant of the webcam.

Table 3 Pose information for Ground Truth points

Coordinate (from labelled in image above)	Coordinates (x, y – from robot's base frame)
1	-250, 75
2	-250, -525
3	-900, 75
4	-900, -525

Appendix B

Colour processing:

When representing the pixels in an image, colours may be represented in multiple ways. We have learnt thus far about the standard RGB (red, green, blue) representation. This, while often being the simplest representation and also the one that cameras directly capture can have some drawbacks when performing image processing.

Another common representation is HSV (hue, saturation, value) where the hue represents the number (either a value out of 360 or a percentage), the saturation is its intensity, and the value is lightness or darkness.

When lighting conditions vary across an image (as is often the case), it is often easier to perform colour thresholding in the HSV colour space. This would mean that rather than putting a range over each of your R, G, and B values, instead you would give a range to each of your H, S, and V values for each pixel. This can be performed in an identical fashion to what was covered in lectures.

It is very simple to convert a given RGB image to HSV in MATLAB using the built-in `rgb2hsv` function (https://au.mathworks.com/help/matlab/ref/rgb2hsv.html).

Please note that the concepts taught in lectures can be used for this assignment, however it may be simpler to tune your colour thresholding if you use HSV representation.