41014 Proposal for Project

Sensors and Control for Mechatronic Systems (University of Technology Sydney)

41014

Sensors and Control for Mechatronic Systems

Project Proposal

Group Project 6: Robot Following a Straight Line by Observing a Square using a Turtlebot

Mohammad Sulaiman Khan (13092489)
Sy Duy Nguyen (12699070)
Huy Nhat Minh Nguyen (13734569)

Date: 31st March 2021

1. INTRODUCTION

The aim of this project is to use data from the sensor attached to a Turtlebot to keep it moving in a straight line. The project tasks including utilising the sensor's abilities to observe four corner positions of a square in the robot's coordinate frame, a square will be situated in front of the robot, to move the robot in a straight path perpendicular to the square while observing noise in the environment (UTS Canvas 2021).

2. SCOPE

The project is defined as the range of tasks and delivery processes undertaken by the team to design a robot that follows a Turtlebot that follows a straight line using an on-board sensor. Therefore, the scope of this project is to demonstrate the following objectives:

1. Visually display sensor data during the motions by using the camera to view the orientation and position of the square from the Turtlebot's perspective.
2. Use sensor data from the robot's sensor as a control feedback (visual servoing)
3. Develop code to calibrate robot's camera to recognise the four corners of the square.
4. Develop code and structures to allow robot to move in a straight line through the environment by observing a square.
5. Construct video demonstration to capture movement of robot in the simulation moving in a straight line in 3D.
6. Develop written documentation of the processes involved with developing the Turtlebot simulation.

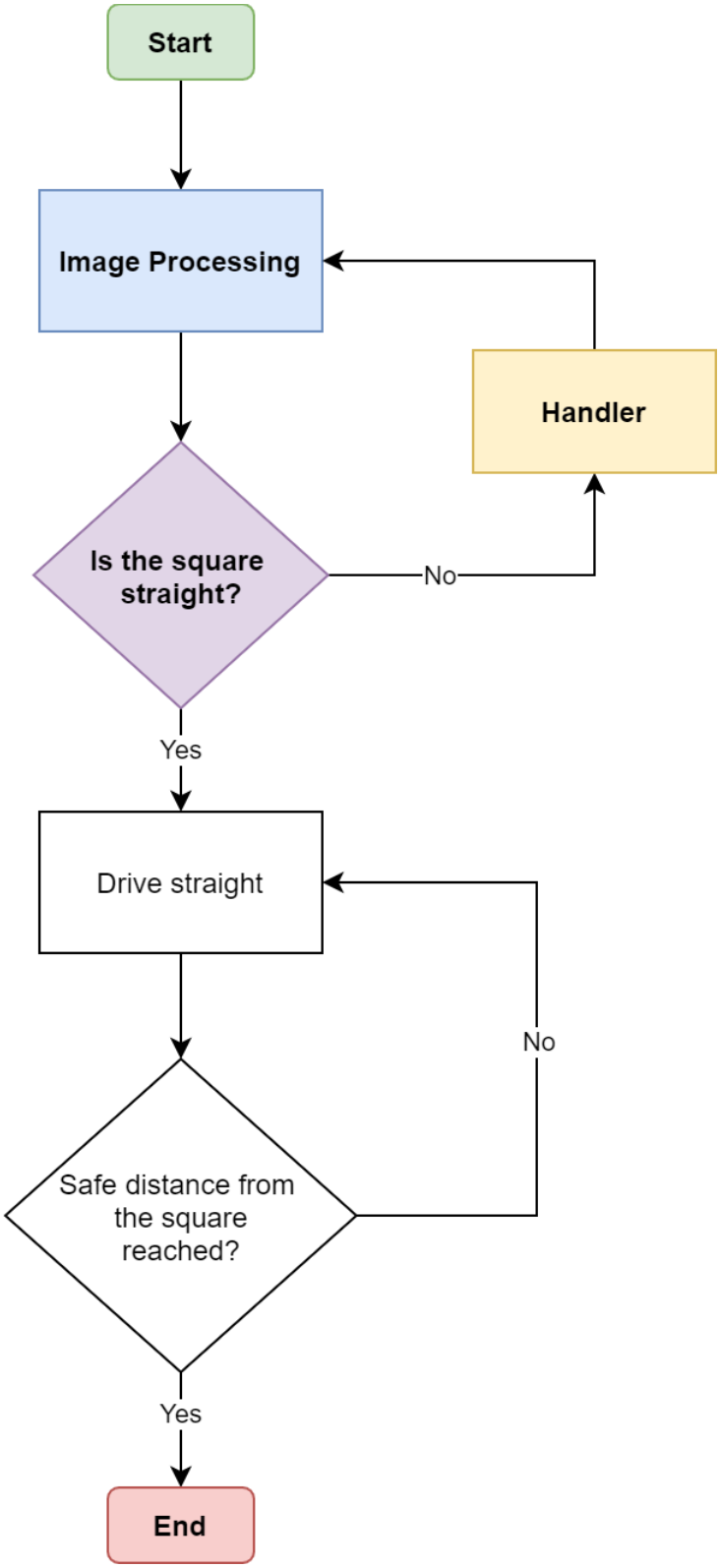
3. DELIVERABLES

The following outlines the deliverable to be handed over to the supervisor throughout the project at the approximate times stated in the table below.

Deliverables	Delivery Date
Submit proposal to Dr. Liang Zhao	31/03/2021
Mid-review report	22/04/2021
Set up the simulation environment for the Turtlebot to move in – using square shape	09/04/2021
Develop starting code to calibrate robot's camera to extract the square shape.	16/04/2021
Develop a condition to allow the robot to move forward in the main code.	02/05/2021
Develop control feedback algorithm to adjust orientation and location of robot depending on position/orientation error.	08/05/2021
Do final debugging and finalise code.	10/05/2021
Finalise video demonstration of simulation in Gazebo	13/05/2021
Deliver presentation to supervisor	20/05/2021
Deliver written report to supervisor	21/05/2021

4. PROPOSED APPROACHES

4.1 Conceptual design



4.2 Proposed Approaches

4.2.1 Image Processing to locate the square (Blue process, see Section 4.1):

Use RGB-D camera to obtain image data and depth info of the environment.

- Obtain depth info of the square for localisation (where the robot is with respect to the square)
- Poll square shape feature recognition (based on projected edges and corners)
- Define and project out lines from the corners of the square, where the robot should stay in between while driving straight (more on this in Section 4.2.3)

4.2.2 Condition to check if the square is straight (Purple condition, see Section 4.1):

With the position of the 4 corners, and the depth info obtained, the square is straight when the depth value of all horizontal pixels (from the robot's perspective) are equal. All of these values must fall between the limits (the depth values of 4 corners) to ensure the background depth is disregarded. The following figure visualises this condition:

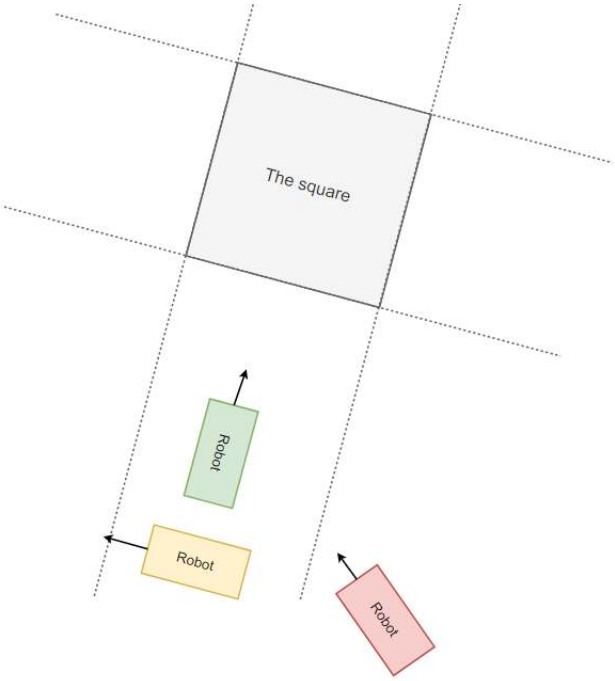
999	999	999	999	999	999	999	999	999	999
999	999	999	999	999	999	999	999	999	999
999	999	999	999	999	999	999	999	999	999
999	999	999	999	999	999	999	999	999	999
999	999	999	10	10	10	10	999	999	999
999	999	999	10	10	10	10	999	999	999
999	999	999	10	10	10	10	999	999	999
999	999	999	10	10	10	10	999	999	999
999	999	999	10	10	10	10	999	999	999

The figure above visualises the expected depth values received in the scenario where the square is straight from the robot perspective. The pixels (or values) inside of the 4 corners (highlighted yellow) will be the region of interested, this will be defined by the feature recognition and geometry of the square.

With the interested region defined, background depth values or noises (being 999 in the figure) can be filtered out in the code.

4.2.3 Handler for when the square is not straight (Yellow process, see Section 4.1):

Using the depth info obtained, the orientation of the square can be determined (from the robot perspective). The robot can then align itself perpendicular to the square by using trigonometry (with lines projected from the corners or the square).



The perpendicular alignment with the square is done in 3 stages (red, yellow, green – shown in the figure above).

Stage 1 (red): Project lines from the corners of the square (dashed lines) and work out the orientation differences to turn the robot parallel to the square.

Stage 2 (yellow): Drive the robot to inside the projected lines.

Stage 3 (green): When Stage 1 and 2 are done, the robot is now parallel with the square and inside the limit area. By turning it 90 degrees (clockwise or counter-clockwise, depending on where the robot begins), the robot will be facing the square perpendicularly. It can then proceed to drive straight until it reaches a safe distance away from the square. While driving straight, a visual feedback (see Section 4.2.2) with a P-Controller will be implemented to anticipate small errors.

5. References

UTS Canvas 2021, *41014 Sensors and Control for Mechatronic Systems*, viewed 30 March 2021 <https://canvas.uts.edu.au/courses/16744/files/1088499?module_item_id=495064>