School of Computer Sciences, Universiti Sains Malaysia

Semester 1, Academic Session 2022/2023

CPC453: Computer Vision & Robotics

Project: Path Planning for A Mobile Robot with Object Recognition for Obstacle Avoidance Purposes

Prepared by: Group S

Presentation Video: Link

Submission Date: 24 January 2023

No.	Name	Matric Number	Email
1	Divenesh A/L Shamugam	146311	diveneshbeez@student.usm.my
2	Katheeravan A/L Balasubramaniam	147744	katheeravan305@student.usm.my
3	Thanish A/L Natarajan	149156	thanish@student.usm.my

Abstract

Obstacle avoidance and path planning is the essential task in the autonomous mobile robot since its aim is to reach the goal without any collision with obstacles. Computer vision is proposed in this project to sense the environment of the virtual robot and carry out image segmentation to extract the information from the environment. This paper explores the BiRRT algorithm to determine the shortest path from any current position to the goal in an environment with static obstacles. We discuss the algorithm used for path planning, the map environment, robot architecture, and obstacle avoidance. We also discussed the implementation vision sensor that used to take image frames from the environment and theoretically explained on how the image frames are processed for further analysis and implementations. Example images for several tasks are provided for justifying the discussion.

Table of Content

1. Introduction

Autonomous robot is basically a machine that can carry out a certain task without human intervention. These robots are able to navigate and interact with the environment with the help of sensors, actuators and processors. Nowadays, different industries are implementing this technology because autonomous robots can be programmed for a wide range of tasks. One of the main tasks that a robot needs to carry out is navigation. Robot navigation is where the robot moves from a starting point to goal point. It is made up of mainly three components which are object detection, obstacle avoidance and path planning. Object detection involves analyzing the environment and differentiating obstacles from paths. As for obstacle avoidance is where the robot avoids the obstacle detected and continues to move towards its goal. Object avoidance can be done both statically and dynamically. The final component is path planning where the robot tries to determine the most optimum path to reach its goal point. There are many algorithms that can be implemented for path planning.This includes Bug algorithms, A* algorithm, Potential Field algorithm and Wandering Standpoint algorithm. In order to determine the best path, several factors such as map borders and obstacles need to be taken into account. Simultaneous localization and path planning (SLAM) is a technique that can be implemented in robots to navigate an unknown environment. Robots will create maps of the environment during navigation. This paper covers obstacle detection using computer vision, obstacle avoidance using proximity and ultrasonic sensor and path planning using Bidirectional Rapid Exploring Random Tree (BiRRT) algorithm.

2. Problem Statement

Autonomous mobile robots have become more prominent in recent times and widely used in many fields, especially in warehouses. Path planning and computer vision is the most essential task in mobile as its aim is to reach specific goals from any current position. Hence, a smooth and continuous path should be calculated by the mobile robot to reach the assigned goal. This problem is further complicated in the case of multiple static and dynamic obstacles as the path generated by the mobile robots may intersect with obstacles. Thus, the bidirectional rapidly exploring random trees (BiRRT) algorithm is used in this project as it analyzes the entire map, locates the static obstacles and generates a geometrical path that could avoid the static obstacles. The mobile robot used in the project is Pioneer P3DX as it comes with a pre-installed bunch of 16 sonar sensors. The sensors transmit sound waves and recollect to calculate the distance of the obstacle in the area sensor face. As the Pioneer P3DX comes with sonar sensors that cover the entire P3DX, the robot can detect the obstacles from any angle. Hence, the robot can avoid the dynamic or static obstacle if it is the colliding distance with the mobile robot and recalculate its path to reach the goal. The proximity sensor is used in this project where it helps to return out from the all-sided covered static obstacles and recalculates the path to reach the goal. The computer vision works along with the obstacle avoidance where it triggers the vision sensor to take the image frames during the trip of the robot and carry out image segmentation to identify the obstacles.

3. Object Detection

3.1 Introduction

Object detection is the process of analyzing an environment and identifying and locating obstacles. This approach is closely related to path planning as it is very important to identify all obstacles in an environment in order to plan the efficient path for mobile robots to navigate. There are mainly two ways to perform object detection. One way is to use a sensor based method which utilizes sensors like ultrasonic, infrared and lidar sensors while another method is a computer-vision based method where object detection will be done using images captured from either camera or vision sensor. In this project a computer-vision based approach was implemented for object recognition in an environment.

Computer-vision based approach involves retrieving image from a robot through sensors and then analyzing the image to detect obstacles. There are many object detection algorithms such as You Only Look Once (YOLO) and Faster R-CNN. In this project, objects will be detected with computer vision techniques image segmentation and edge detection but object classification will not be focused.

3.2 Methodology

First the image that has been retrieved will be converted to grayscale. This is done to perform histogram equalization on the image to make sure that the intensity level of the image is evenly distributed. There are two ways to perform histogram equalization on coloured image, either by converting the image to grayscale or separating each band of coloured and performing histogram equalization separately. For this project the first method has been chosen. Figure 2 shows the intensity level graph of grayscale image before performing histogram equalization. The intensity level of the pixel is higher at grayscale level 150 to 225. This indicates that the image is on the lighter side. As can be seen in the image, the environment and floor is on the light side. At grayscale level 40 to 110 the intensity level is high as well which indicates that some part of the image is on the darker side. As we can see in the image, the shapes are the darker side which represent the high intensity at level 40 to 110. Figure 3 shows the intensity level histogram after histogram equalization has been performed. Now the intensity level is evenly distributed but the image overall has become darker compared to the original image.

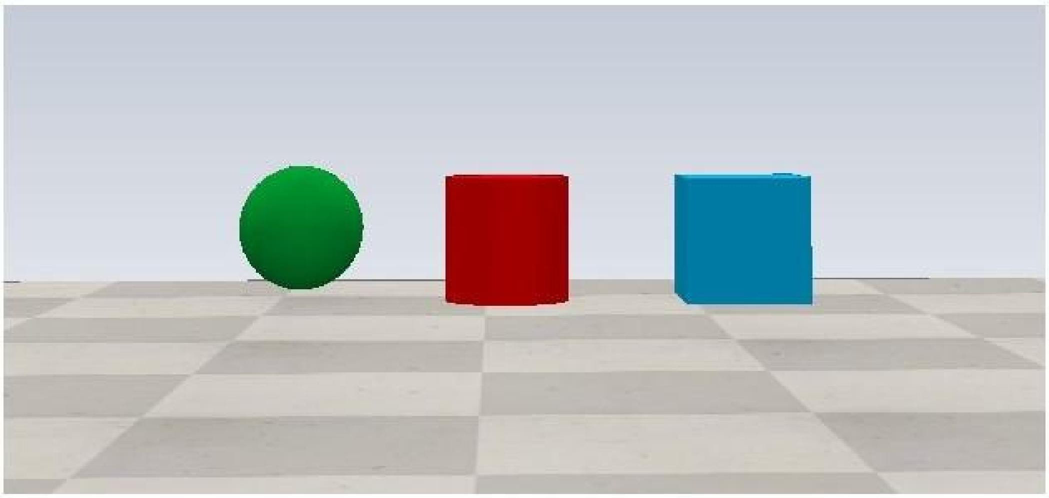
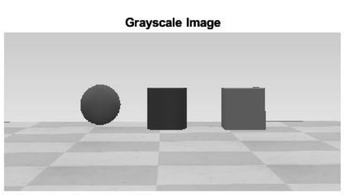


Figure 1: Original Image for Object Detection



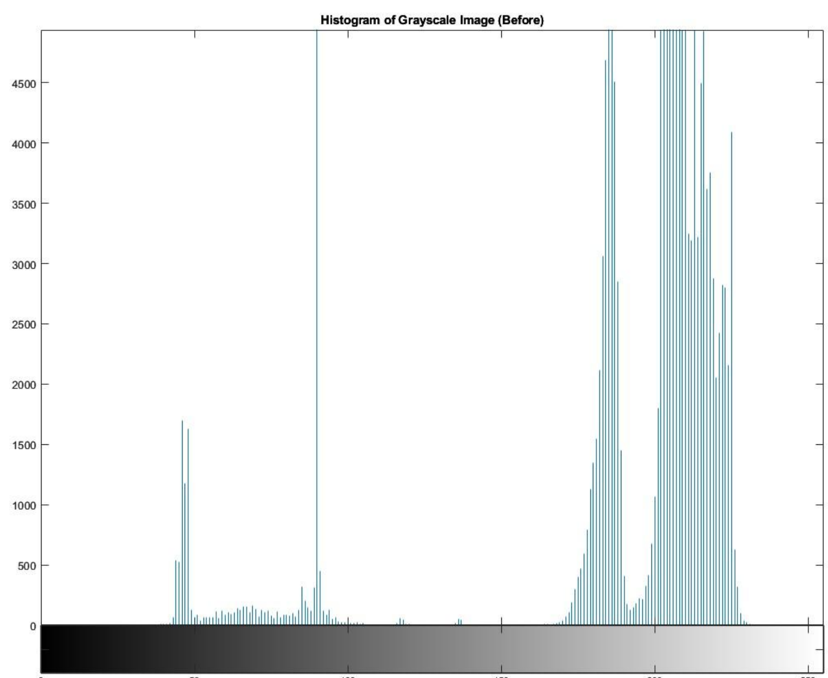


Figure 2: Histogram of Grayscale Image before Histogram Equalization

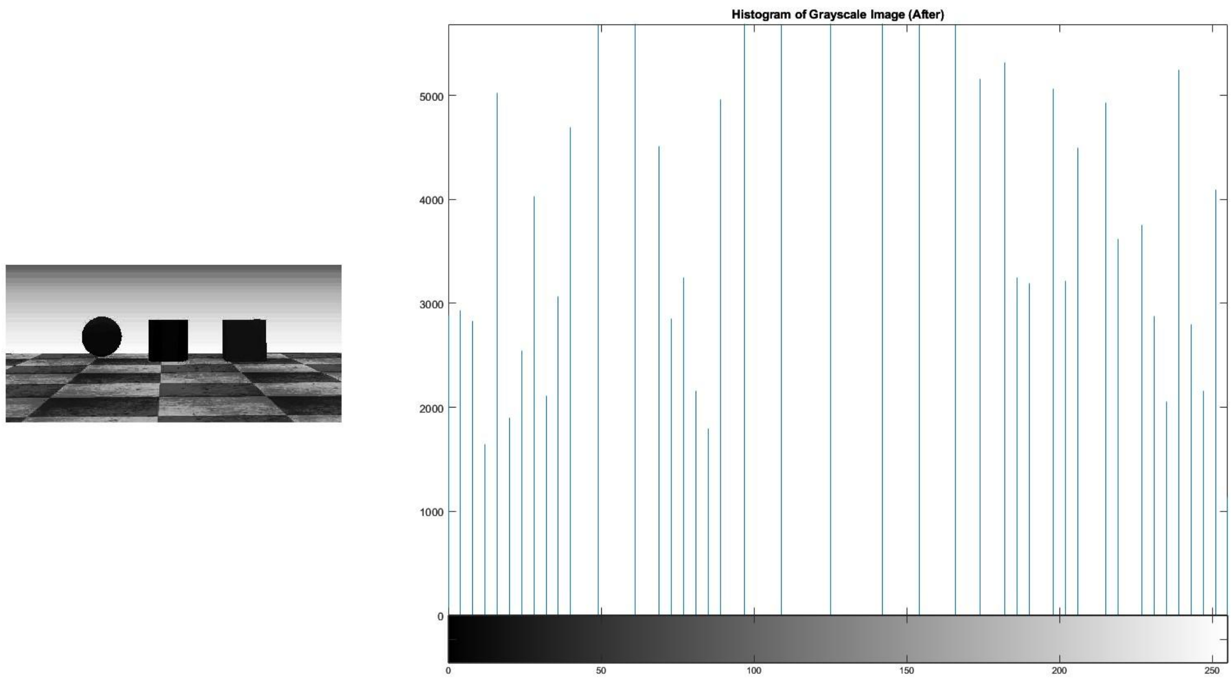


Figure 3: Histogram of Grayscale Image after Histogram Equalization

Next comes image denoising. Image denoising is done in order to remove unwanted noises from the image before performing image segmentation. Two filtering methods have been implemented which are average filtering and median filtering. After performing these two filtering methods on both normal image and histogram equalization image, it can be said that median filtering performed better in both of the images. The average filtering method causes both the images to become blurry as shown in Figure 4 and Figure 6.

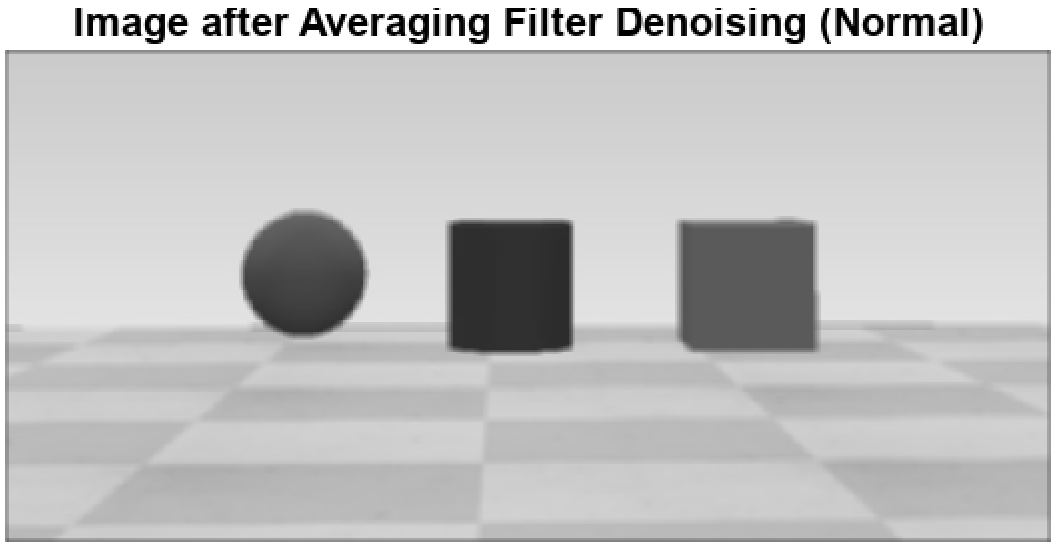


Figure 4: Normal Image with Average Filter

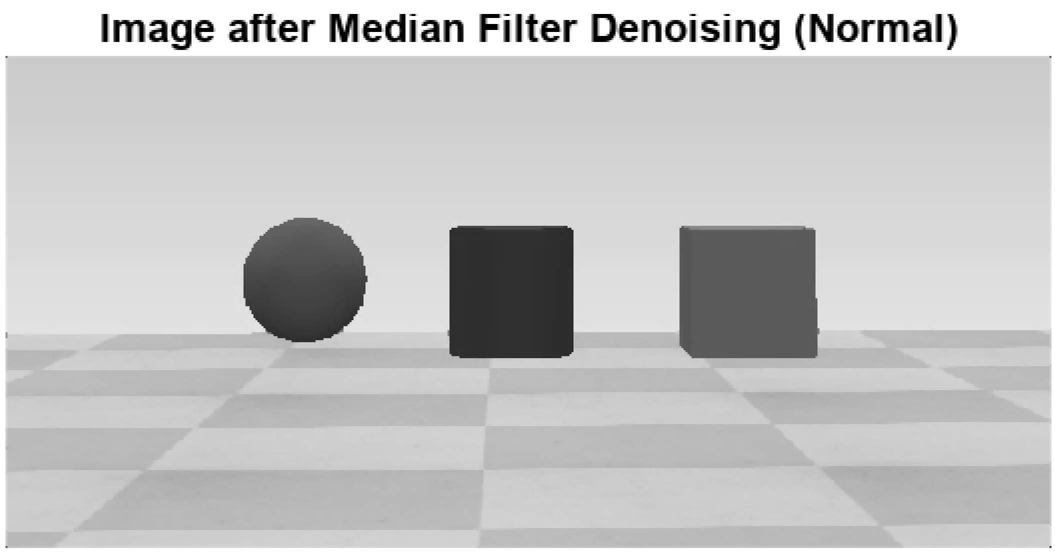


Figure 5: Normal Image with Median Filter

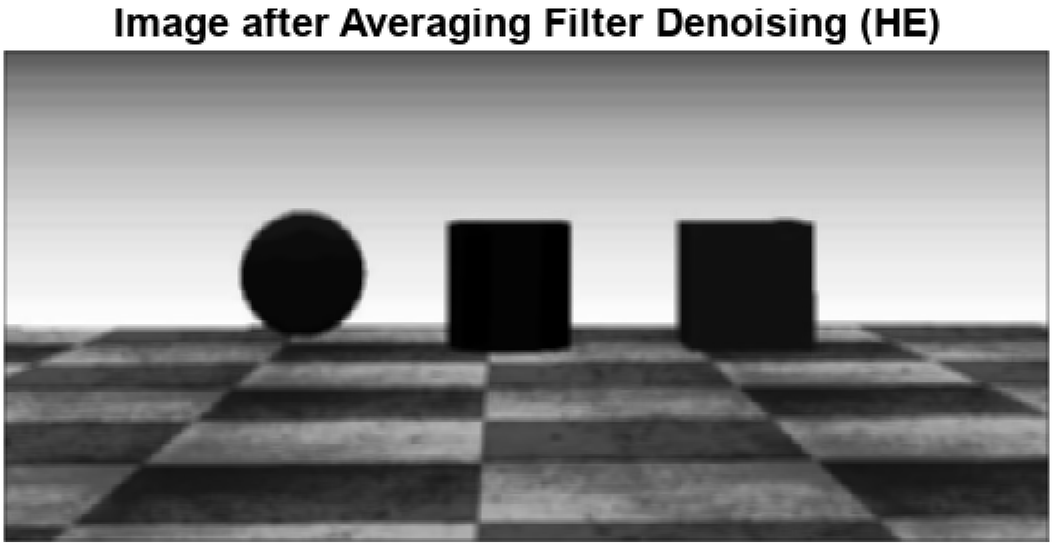


Figure 6: Histogram Equalized Image with Average Filter

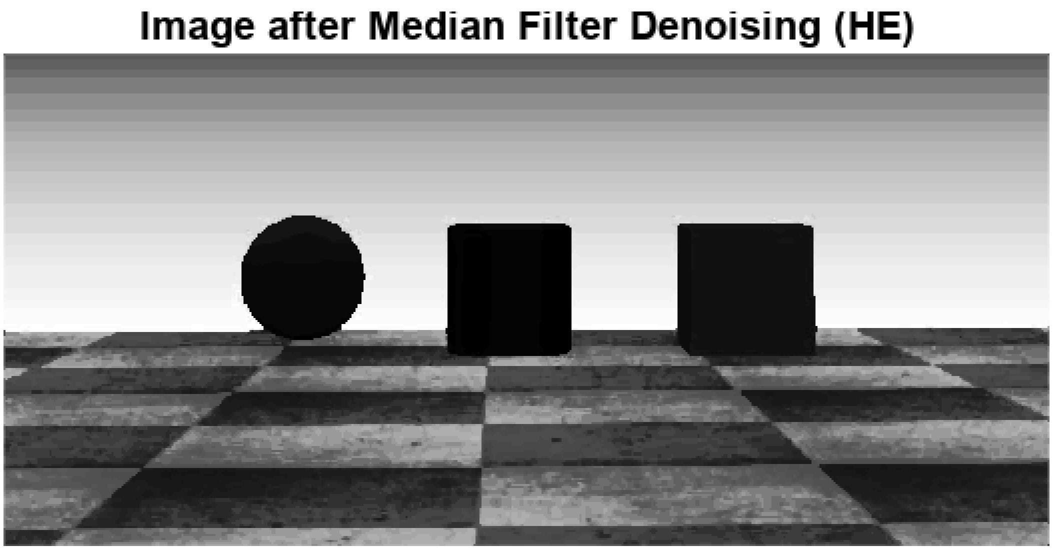


Figure 7: Histogram Equalized Image with Median Filter

Following that would be the image segmentation part. Two different image segmentation techniques have been implemented for this project, edge detection and OTSU thresholding.

For edge detection, Sobel filter and Canny filter have been implemented. Compared to the Canny filter, the Sobel filter performs better. As it can be seen in Figure 8 Sobel Filter on normal image can separate the image accurately but Sobel filter on normal image can detect the shapes accurately as there are some missing edges and it is connected to the floor. Generally, Sobel filter outperforms Canny filter in edge detection in both images.

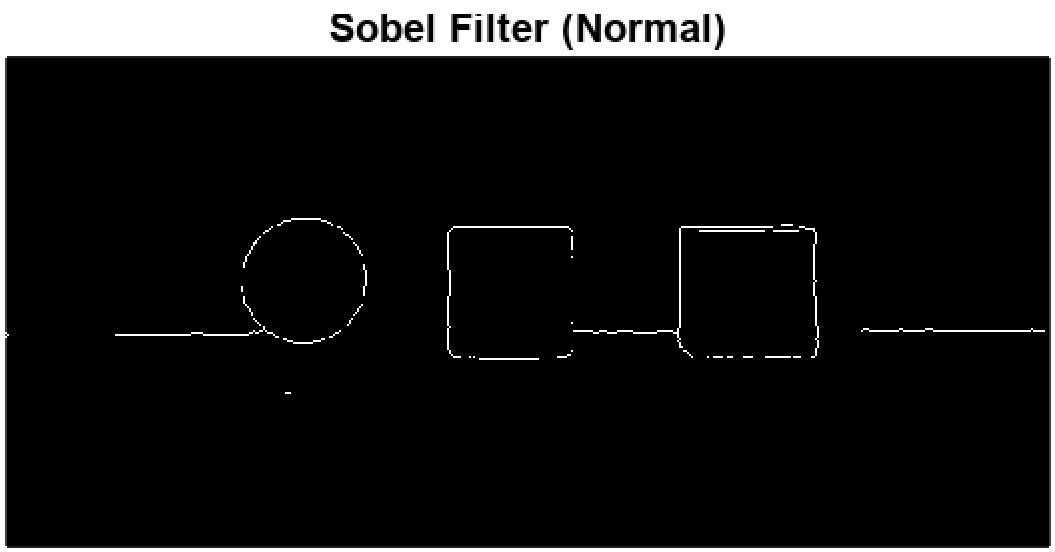


Figure 8: Normal Image with Sober Filter

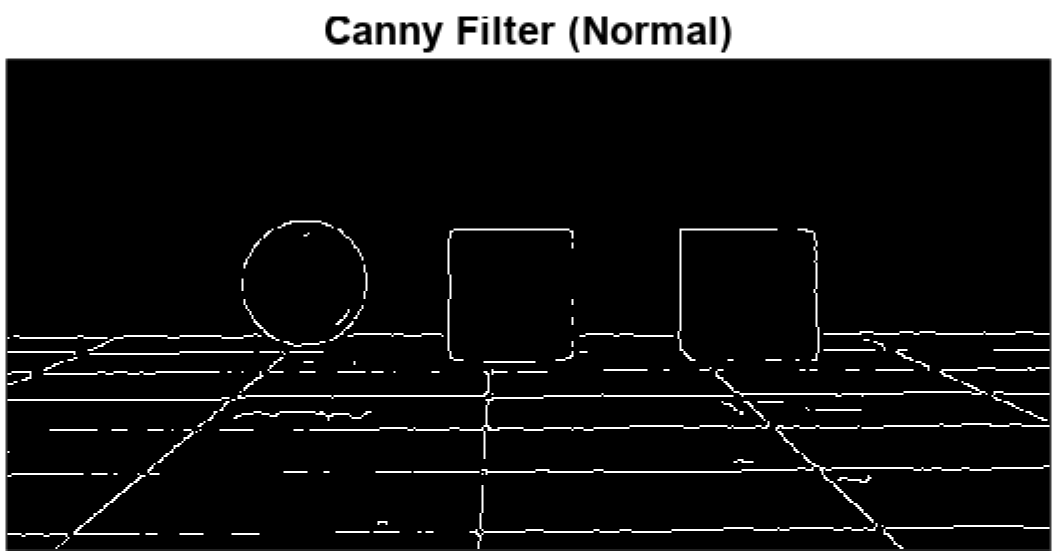


Figure 9: Normal Image with Canny Filter

Figure 10: Histogram Equalized Image with Sober Filter

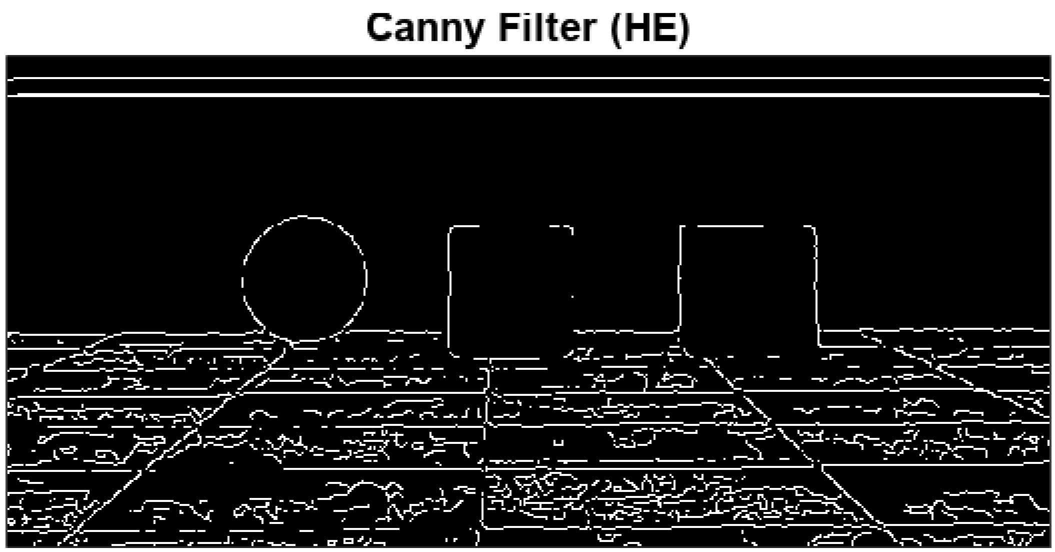


Figure 11: Histogram Equalized Image with Canny Filter

The final part will be image segmentation using thresholding methods. The OTSU thresholding method was chosen because the histogram of the original grayscale image has bimodal distribution between two peaks. So, OTSU thresholding will be the most suitable. OTSU thresholding can perform well on a normal image as shown in Figure 12. All the shapes were segmented from the environment accurately. OTSU thresholding on histogram equalized image is not performed well. The shapes could not be segmented properly some of them are still connected to the floor as connected segments. Overall image segmentation on normal image can be done accurately compared to histogram equalized image. For the denoising of image, median filtering is better and for edge detection Sobel filter is chosen.

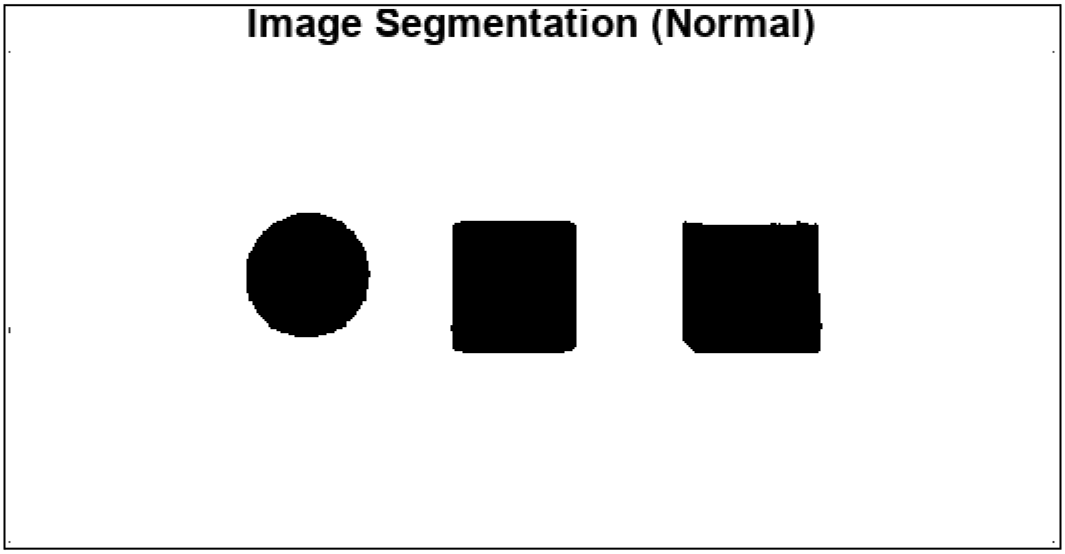


Figure 12: OTSU Thresholding Normal Image

Figure 13: OTSU Thresholding Histogram Equalized Image

3.3 Computer-Vision based Object Detection in Path Planning

The vision sensor in the mobile robot will use a specific function to capture the image and store it in a specific folder. Then, all the images that are stored in that folder will be read by MATLAB code and processes each image by detecting the objects. When there is an object detected, the MATLAB will use an API to integrate with CoppeliaSim software and connect the mobile robot and make the robot move by using the detected image in real time.

4. Obstacle Avoidance

4.1 Introduction

Obstacle avoidance is an action performed by mobile robots or vehicles to sense an obstacle in its environment and can successfully avoid the obstacles along its navigation towards a target. This ensures that there is no collision between the obstacles and the mobile robots. Obstacle avoidance requires a variety of sensors such as lidar, ultrasonic sensors, vision sensors, and infrared sensors which reads the data from the robot environment and processes the data using suitable algorithms to create a successful navigation without collision with obstacles. The actuators such as electric are used to move the wheels of the mobile robots away from the obstacles. The main objective of obstacle avoidance is to make sure the safety of mobile robots is always protected.

In this project, there are two sensors used to sense the obstacles that are dynamically placed on the path that has been created to reach the target. The first sensor is the ultrasonic sensor which is used to detect the obstacles and try to avoid the obstacles by changing the motor speed. There are a total of 16 sensors used from the left side to right side of the mobile robot. The second sensor is proximity sensors with ray type which is used to detect the obstacles in the front as well as makes the mobile robot create a new path to avoid the obstacles. There are only 3 sensors used which are placed in front of the mobile robot.

4.2 Methodology

The purpose of object avoidance in this project is to make the mobile robot able to avoid the obstacles that are placed on the path of the robot that is needed to navigate towards the goal, which is a cylinder. The methodology of this object avoidance is explained by using two different scenarios. The map contains several obstacles such as spheres and cubes. The first scenario is placing a cube near the path generated which slightly blocks the robot from reaching its target. When the robot comes, the ultrasonic sensor will read the value and perform the Braitenberg algorithm to calculate the speed of the left and right motor to avoid the cube obstacle. When the speed of motors is adjusted, the robot will turn and overcome the obstacle and at the same time maintain the path created initially. The figure 14 and figure 15 show the mobile robot opposing the cube obstacle and the way the robot avoids that obstacle without colliding with it.

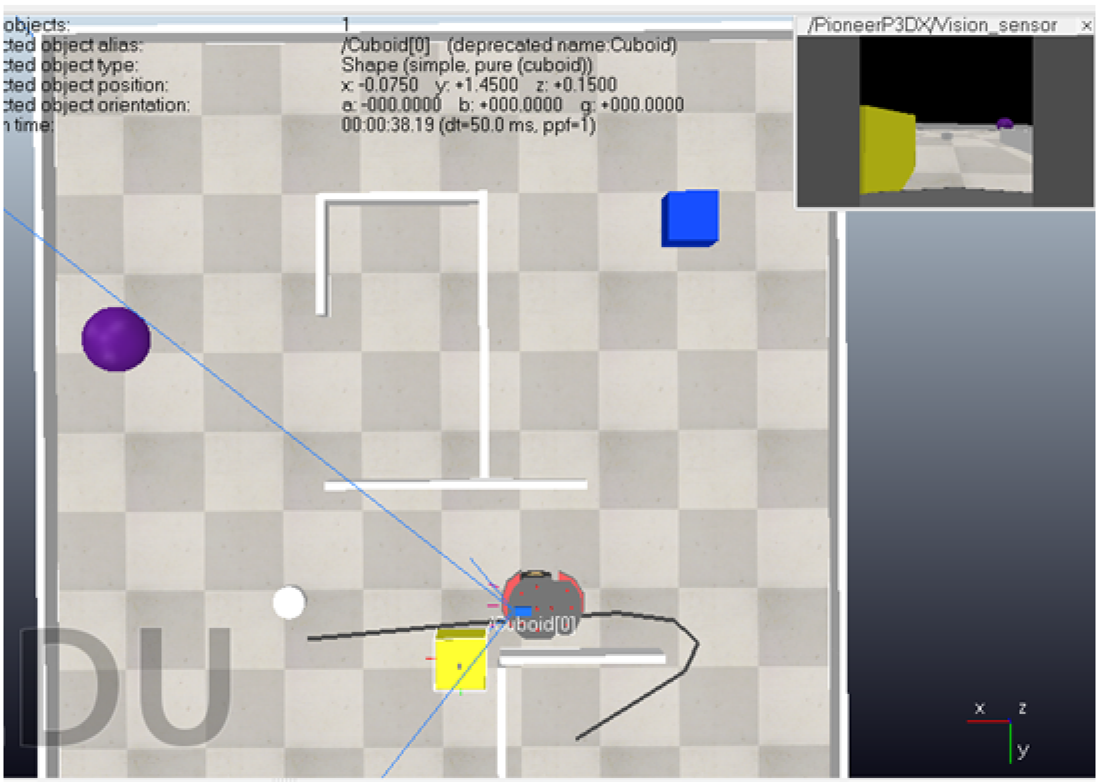


Figure 14 shows mobile robot before avoiding square obstacle

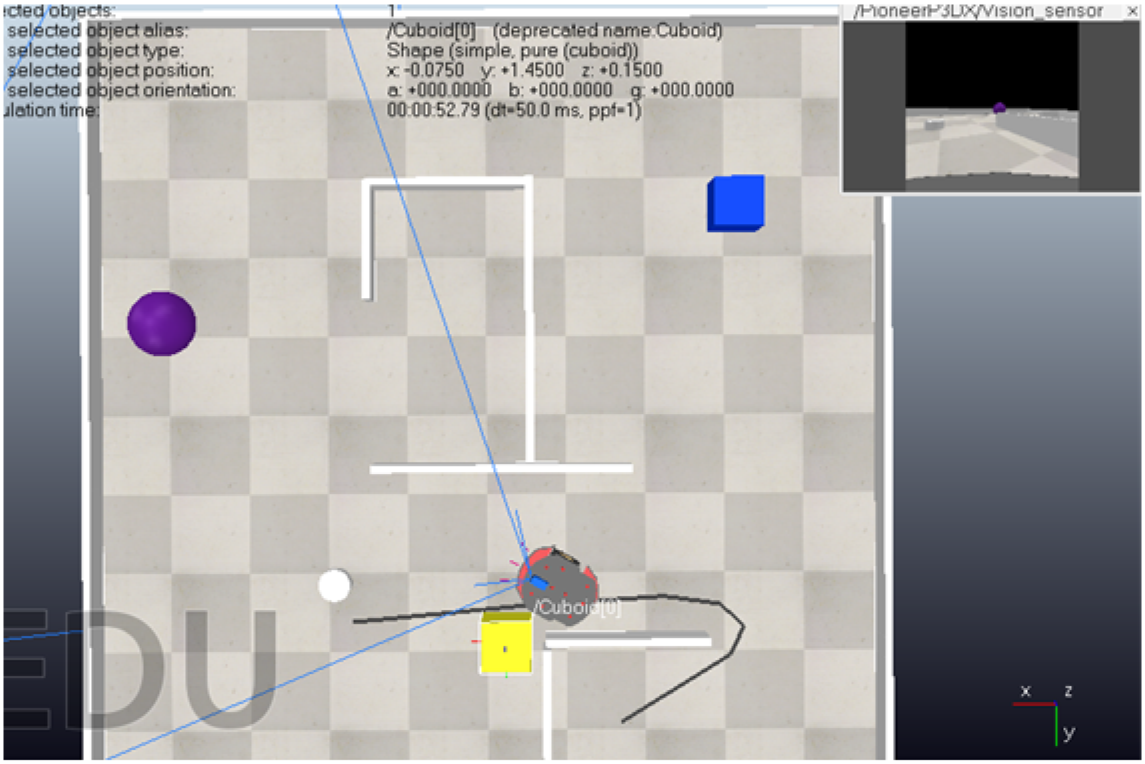


Figure 15 shows mobile robot after avoiding square obstacle

. In the second scenario, another obstacle, which is a sphere, is placed on the path of the robot which fully blocks it from going towards the target. When the mobile robot moves, the ultrasonic sensors sense the sphere and try to avoid it without breaking the path, but it cannot. So, the proximity sensor is used to detect the sphere. The result of proximity will become 1 when there is an object detected. So, when the sphere is detected, the path created will be destroyed and no path will be created at that particular moment. During that time, the ultrasonic sensors will try to avoid the sphere obstacle by moving the front side of the robot away from the sphere. This allows the proximity sensors to have the result value as zero. The value zero means that there are no obstacles detected in front of the robot. So, now a new path will be generated to continue its journey towards the goal. The figure 16 and figure 17 show the mobile robot path before and after detected by proximity sensors.

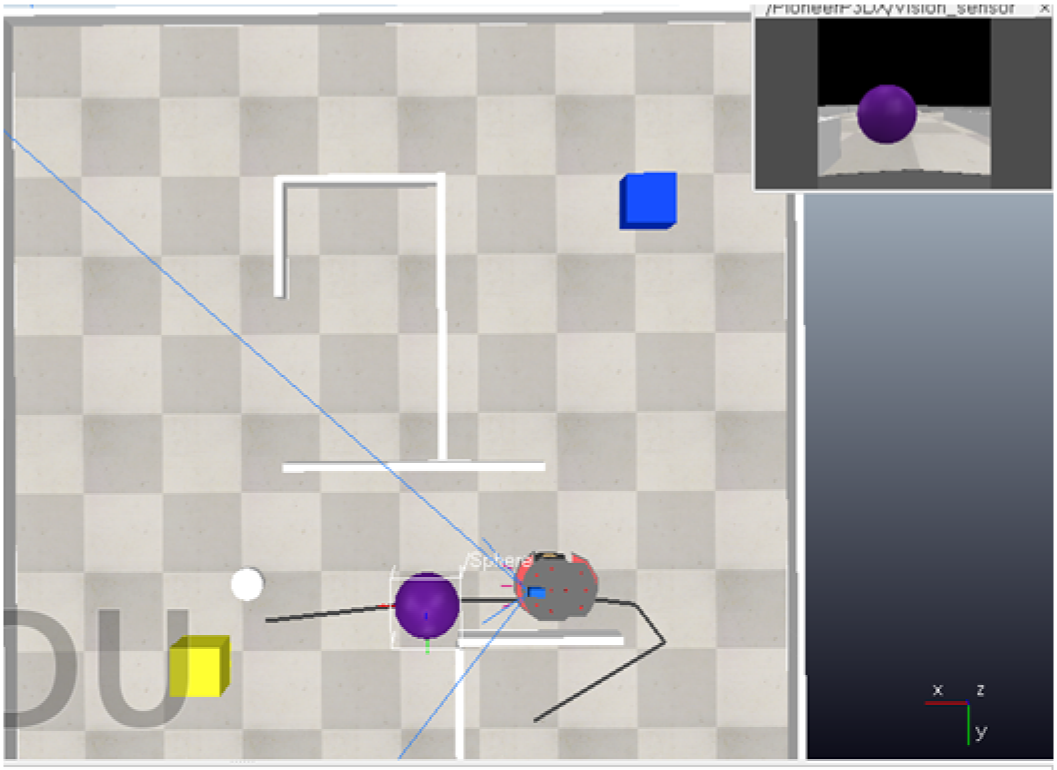


Figure 16 shows mobile robot before detect proximity sensors.

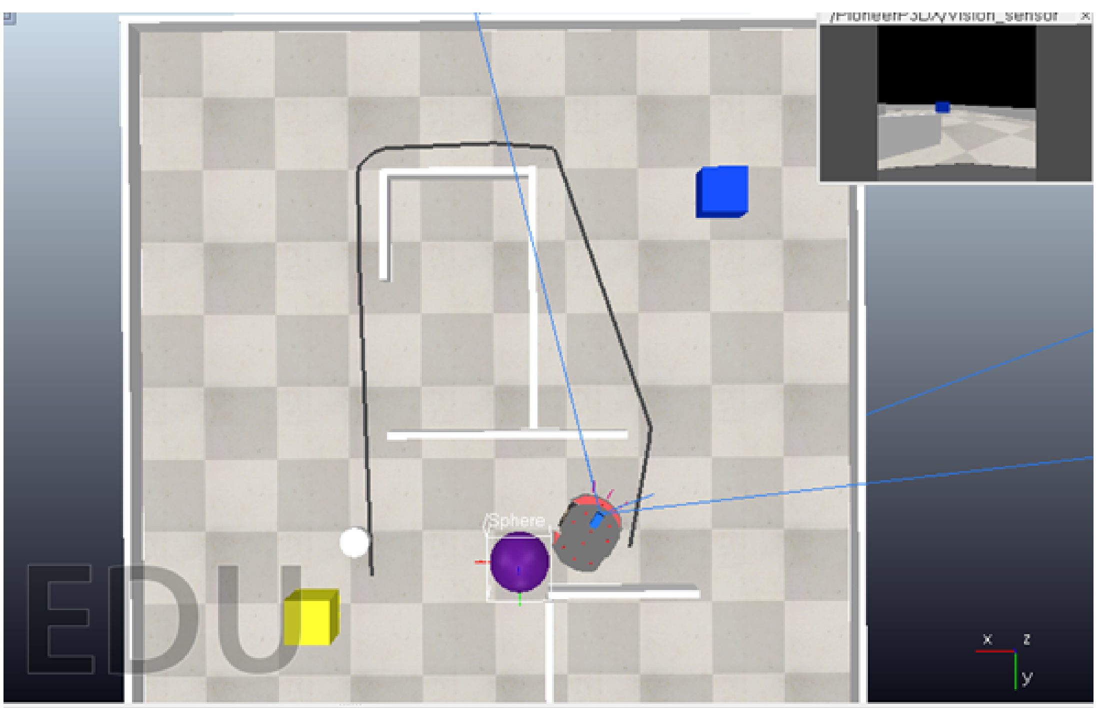


Figure 17 shows mobile robot after detect proximity sensors.

5. Path Planning

5.1 Introduction

The path planning is the essential part of the mobile robots to generate its own shortest path from any current state to the goal state. Path planning requires the map of the entire location of the robot including the starting point and the goal point. The maps can be in any form such as state spaces, topological maps and grid maps. The path planning is divided into two techniques whether global path planning or local path planning. The global path planning analyzes the entire map, gets all the static obstacles, goal of the robot and initial point of the robot, then calculates the path. The local path planning senses the environment and controls the robots actuators based on the environment. The path planning technique varies based on the algorithm whether Grid-Based Search Algorithm and Sampling Based Search Algorithm. Grid based search algorithm memory requirements are based on the dimension, higher dimension such as 4DOF or 6DOF, the higher the memory requirement. It finds the least amount of time if grid-map. Sampling-based algorithms can be used in search spaces with low or high dimensions.With sampling-based search algorithms, a state space's new nodes or robot configurations are randomly sampled to produce a searchable tree.

5.2 Methodology

The algorithm used in the project is Bidirectional rapidly-exploring random trees (Bi-RRT). The Bi-RRT algorithm derived from the Rapidly-exploring random trees (RRT) algorithm where it solved some of the problems in the RRT algorithm. The RRT is the most well-known sampling-based algorithm. RRT works by creating two trees at the beginning point and another tree at the end point. Then, the trees decides its own roles where one creates nodes and another tree tries to link the node until the path is found. Although this process is not efficient as it is not finding the shortest path, this works rapidly which can be implemented to the robots which need to identify their path quickly. The major drawback is the algorithm characteristics will interfere in determining the sampling heuristics. Hence, the Bi-RRT algorithm becomes a better approach for the drawback found in the RRT. Asymptotic optimality is not guaranteed when connecting two-directional trees using the greedy connect heuristic. Before attempting to join the trees together using the RRT-Connect heuristic, the Bi-RRT first searches for neighboring vertices in the tree that is being processed. Bi-RRT based path planning algorithms have received more attention as a result of their efficiency and effectiveness in producing viable paths.

5.3 Implementation

Initially, all the variables that are used in the global path planning are initialized. The algorithm planned to be used is Bi-RRT hence the algorithm is set into an variable. Then, the "coroutine" which is the main function for the entire path planning is started. In the main function, at first it set the goal variable as not reached. Then it starts a loop until the goal has to be reached. In the loop function, it will check if the robot is colliding with the static obstacles or not. Then, it calculates the starting point of the robot and goal and length between the starting point and the goal point. It will check whether the goal is moved from its initially calculated position or not, if true, it will calculate the path. Then, it will use the algorithm to create the path, set the starting point of the robot and the goal point. It draws a path line to show how the robot is going to travel to reach the goal point. Then it will calculate the collision of the robot with the entire static obstacles in the map. It will initiate the robot actuators, calculate the position and orientation of the robot and drive the robot based on the calculated path. Once the robot reaches the goal, it will set the variable of goal as reached and break the entire loop. Once the loop is breaked, the system will remove the lines drawn to navigate the mobile robot and remove the nodes, and clean up the memory used.

5.4 Sample Images

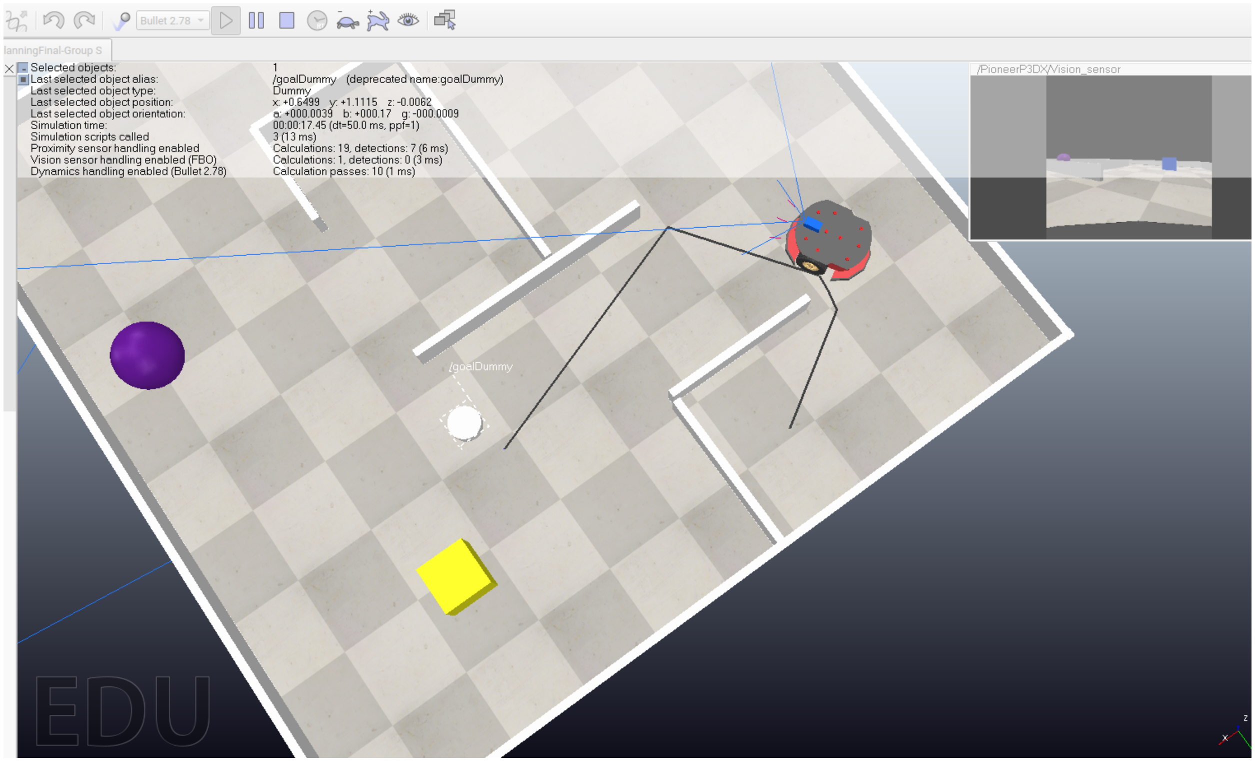


Figure 18: Global Path Planning

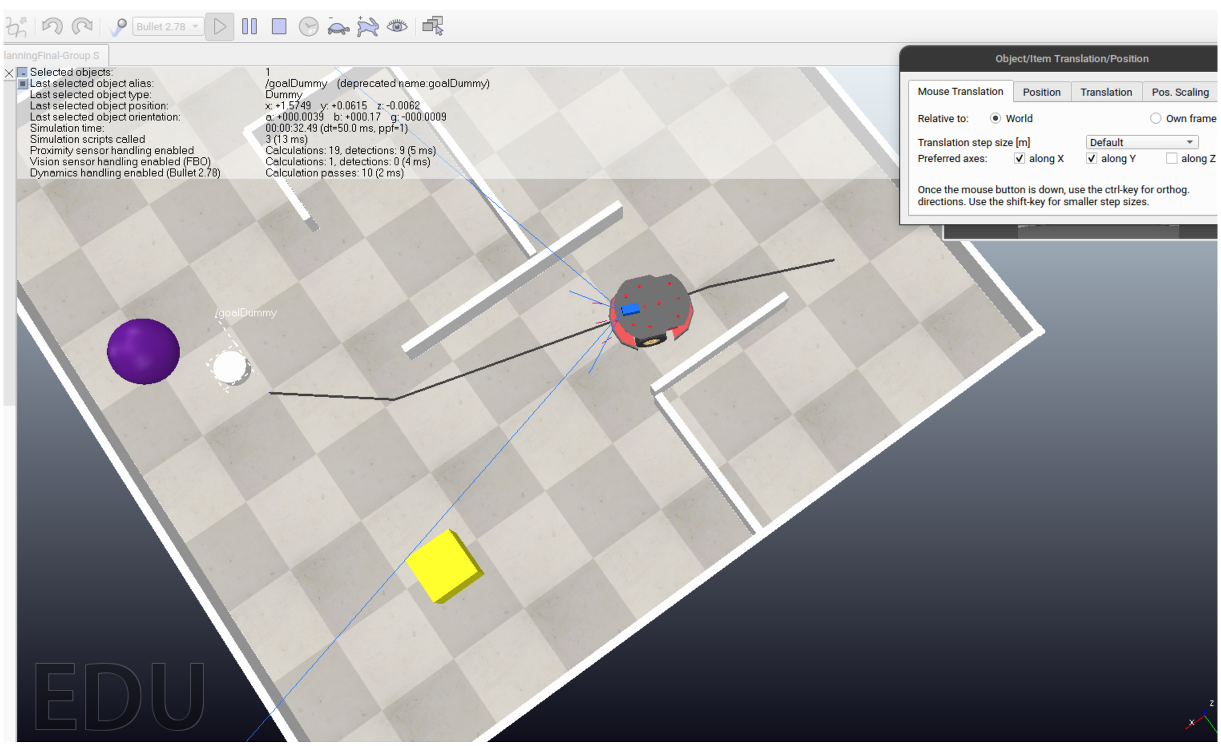


Figure 19: Dynamic Path Planning

6. Conclusion

In summary, autonomous robots are helpful for humans in decreasing their burdens in many ways, especially in fields such as industries, logistics and exploration. The path planning for the robot is very essential as it impacts the cost, time and scalability. If the path chosen is longest, then the mobile robot will need more resources to reach its goal. This is also time consuming and less efficient. Thus, a proper algorithm which suits the robot's surroundings is needed and since the path can be in both ways which is static and dynamic, it needs to detect obstacles using vision and avoid the obstacles by using the sensors and actuators. In our project, the Pioneer P3DX mobile robot is able to target its goal and creates the path. Then, it moves along the path until it reaches the goal. If an obstacle is present along the path, the robot senses them and moves away without losing its path. Moreover, if an obstacle completely blocks the path in a dynamic, then the robot detects it using proximity sensors and moves away from the obstacles. After successfully moving away from that obstacle, the robot creates another new path towards its goal and continues its journey. Finally, the robot stops its movement when it reaches the goal. However, this path planning is complex to build and requires a lot of effort in order to create more efficient path planning with obstacles avoidance.

Reference

1. Muhammad, A.; Ali, M.A.H.; Turaev, S.; Abdulghafor, R.; Shanono, I.H.; Alzaid, Z.; Alruban, A.; Alabdan, R.; Dutta, A.K.; Almotairi, S. A Generalized Laser Simulator Algorithm for Mobile Robot Path Planning with Obstacle Avoidance. *Sensors* **2022**, *22*, 8177. https://doi.org/10.3390/s22218177
2. Qureshi, A. H., & Ayaz, Y. (2015). Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments. *Robotics and Autonomous Systems*, *68*, 1–11. https://doi.org/10.1016/j.robot.2015.02.007
3. *Path Planning*. (n.d.). MATLAB & Simulink. https://www.mathworks.com/discovery/path-planning.html
4. Ghosh, D., Nandakumar, G., Narayanan, K., Honkote, V., & Sharma, S. (2019). Kinematic Constraints Based Bi-directional RRT (KB-RRT) with Parameterized Trajectories for Robot Path Planning in Cluttered Environment. *2019 International Conference on Robotics and Automation (ICRA)*. https://doi.org/10.1109/icra.2019.8793896