

ENGINEERING PRODUCT DEVELOPMENT

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

30.119 Intelligent Robotics

Robotics Computer

Vision Project

Team 12

Barry Tee Wei Cong | 1001549

Choo Jia Hui| 1001790

Mandia Joaquin Inigo Manas | 1001809

Chua Tze Chong | 1001758

# Introduction

The aim of this report is to describe the steps to demonstrate the capabilities of a wheeled robot (ie TurtleBot) in avoiding obstacles in a mapped course by use of cameras. This is to simulate the ability of an autonomous vehicle to avoid obstacles on mainstream traffic which can be a hazard if not done in a proper and safe way.

# Concept

By mounting a Kinect on the robot, we will be able to leverage on its 3D mapping capability and program the robot to avoid obstacles while travelling. The Kinect contains an RGB color VGA video camera, a depth sensor and a multi-array microphone. For the 3D mapping, the camera and the depth sensor had to work in tandem to generate the 3D map. First, the infrared projector will illuminate and paint the object with a series of markers. Next, the depth sensor and the camera will determine the distance of the object using triangulation. In our demonstration, we programmed the robot to automatically avoid any obstacles that is 30cm in front of the robot. We have to generate a 3D map of the area that the robot will be traversing along manually and upon running the program, the robot is able to detect any new obstacles in front of it and will initiate its obstacles avoidance algorithm.

# Equipment

1. HP Laptop with Linux OS 14.04 Installed
2. Kinect Camera V1
3. TurtleBot

# Steps to Execute the Movement of a Robot

1. Turn on the HP Laptop.
2. Open a New Terminal Window. Key in the following command. This command is to interface with the TurtleBot.

roslaunch turtlebot\_bringup minimal.launch

1. Open a New Terminal Window. Key in the following command. This command is for the TurtleBot to start the mapping process.

roslaunch turtlebot\_navigation gmapping\_demo.launch

1. Open a New Terminal Window. Key in the following command. This command is for the User-Interface (UI) of the mapping window as shown in Figure 1.

roslaunch turtlebot\_rviz\_launchers view\_navigation.launch

|  |
| --- |
|  |
| Fig 1. User Interface of the Mapping Window |

1. Open a New Terminal Window. Key in the following command. This command is for the user to use the computer keyboard as a controller to move the robot and map the place as shown in Figure 2. So, the process is for the Computer Vision software to learn its surroundings in this process and map the boundaries that it can travel freely in.

roslaunch turtlebot\_teleop keyboard\_teleop.launch

|  |
| --- |
|  |
| Fig 2. Keyboard Window to Move the Robot |

1. In the Terminal Window for Step 5, press and hold Ctrl and C keys on the keyboard. It will deactivate Step 5 process. After which, key in the following command. This command is to save the map that was mapped out as shown in Figure 3.

rosrun map\_server map\_saver -f my\_map # launch saving of map

|  |
| --- |
|  |
| Fig 3. Saved Image of the Map |

1. Close all terminals except the Terminal for Step 1. Now we can proceed with the navigation and obstacle avoidance aspects of the demonstration. Open a New Terminal Window. Key in the following command. This command will load the map for navigation.

roslaunch turtlebot\_navigation amcl\_demo.launch map\_file:=/home/arms14/my\_map.yaml

1. Open a New Terminal Window. Key in the following command. This command is for the UI of the navigation window as shown in Figure 4.

roslaunch turtlebot\_rviz\_launchers view\_navigation.launch

|  |
| --- |
|  |
| Fig 4. User Interface of the Navigation Window |

1. In the UI, just set the starting position and heading of the TurtleBot with the 2D Pose Estimate Position option. Next, set the target position and heading with the 2D Nav Goal option. The robot will be able to route a path on the map that it mapped previously. Any new obstacles found within its path will be rerouted and avoided based on the obstacle avoidance algorithms used in the library. Below is the link on YouTube of our demonstration.

<https://youtu.be/9De0VB1HQ8w>

# Conclusion

The continual development of autonomous vehicles banks on the main factor of safety. Given a high rate of 30 frames per second (FPS) used by the camera on the autonomous robot, the collision of autonomous vehicles with obstacles is not completely zero. As such, further development will be needed to ensure fool-proof safety.