TOUR GUIDE BOT

BLG456E Project Final Report

Team

Marvin

Recep Can Babaoğlu -150130112

Eray Mert KAvuk - 150130133

23 December 2018

Abstract

This project is about designing the model and software of a turtlebot that guides the students in an exhibition environment from a base point to a route with predefined location. In our day and age the importance of both the hardware and software of a robot continues to rise, making the research and development of such areas worthwhile. Our project is fully completed.

1. Introduction

Exhibition guide TurtleBot will design for deal with a real issue. As we all see finding a guide or reaching the necessary information is a problem for many people when they visit an exhibition, art gallery or a museum. In this project we would like to do something in this area.

In this project, we are going to design and develop Exhibition Guide Robot for primary school children who are just start learning about math and geometry. In our exhibition we will exhibit some geometric shapes such as triangle, circle etc…Our aim is make children love math and geometry. That's why we are trying to make our exhibition funnier and use a turtlebot instead of a math teacher.

The project is interesting because it is a solution for a real issue and also solve it in a funny way that children love it. Our other motivation is also it can be used for many different area as well and it can be improved by others. Not only for children or in a funny way but also it can be used for an art gallery. Especially, if the robot supported with enough data that contain information about artworks. It can be wonderful with the language and speech support also. This points are also increase our motivation about our projects.

The project we have selected is the path-following of an turtlebot in an mapped environment, when the robot reaches the destinations, it performs the assigned task by giving some information about the shapes hanging on the wall. Our robot can recognize artworks in the exhibition. When robot recognize an artwork. It will go to it and stop there for couple seconds to give information about artwork.
Our project aims to to design the model and software of a turtlebot that can achieve such a task. In this report we will be discussing the background, design, importance, challenges and results of our project as well as the many approaches we have taken and the final design of our code.
We also captured a **video** for our project and uploaded to youtube: https://youtu.be/0K-PRETNnPg

2. Background

Various advancements in the robotics industry has taken place in the recent years. Many researchers and engineers have focused on interaction between human and robots. One of the emerging application is tour guide robot. Also, many different company previously developed such a guide robot. The implementations of robots varied in many different approaches such as sensors used, map representations, and user interactions. We have investigated some previous tour guide robots. Some of them are Rhino, Minerva, Asimo, Tawobo, Toyota, Skycall [1].

Name of robots	Type of map used	Sensors used	User interactions
Rhino	Given map	Sonar, Tactile, Laser range finder, and Infrared	Website to track robot
Minerva	Learned map	n/a	Website to track robot
Asimo	n/a	Six different sensors	Given screen to write feedback of tour
Tawobo	n/a	n/a	n/a
Mobile Robot	Learned map	Gyro	Users can stand on it and control it.
Toyota	n/a	Image recognition sensors	Able to read name tags. Engages with audience.
Skycall	Learned map	Sonar sensors	Mobile App.

Table 2.1. Summary of previous implementations.

Like many guide robots we were also inspired from RHINO which is a first popular interactive tour guide robot, it was deployed in the “Deutsches Museum Bonn” in May 1997. It has been created to assist and entertain people in museums etc. The first commercial attempt reached 2000 people in 6 day [2]. But our environment not dynamic as a real environment, robot will designed for a public places but we ignored dynamic factors such as people behaviors etc. There is no interaction from human to robot.

3. Design and Implementation

The main idea for this project is to design a turtlebot that goes to predefined points by following a route and then detecting the shapes on given location. We divided the project into two work package. The work package 1 includes environment, mapping and localisation. We presented work package 1 in interim demo, all the parts are completely satisfactory. The work package 2 includes path following and image processing. After the interim demo we have short time for implementing Work Package 2. All packages prepared according to proposal order. Also the group member prepared whole project together. Simulation will be done with ROS~Kinetic[3], Gazebo[4] and Rviz[5] on Ubuntu 16.04. We are using turtlebot[6] for our simulations.

3.1 Environment

The preparing environment is a first part from work package 1. We design a new world using Gazebo for our exhibition area. We started from empty world then inserted different models from gazebo model database also we edited model shape, size etc.
We faced our first problem, model database is not enough for our project. We need walls with different shapes on it. Following “Color and Texture” tutorials from gazebosim.org we have overcome the creating new models. We designed a template wall model taken from database then we embedded our shape (jpg image) on the surface. When we placing the shapes on the wall, we considered the turtlebot vision range. Because we need appropriate photos for image processing.

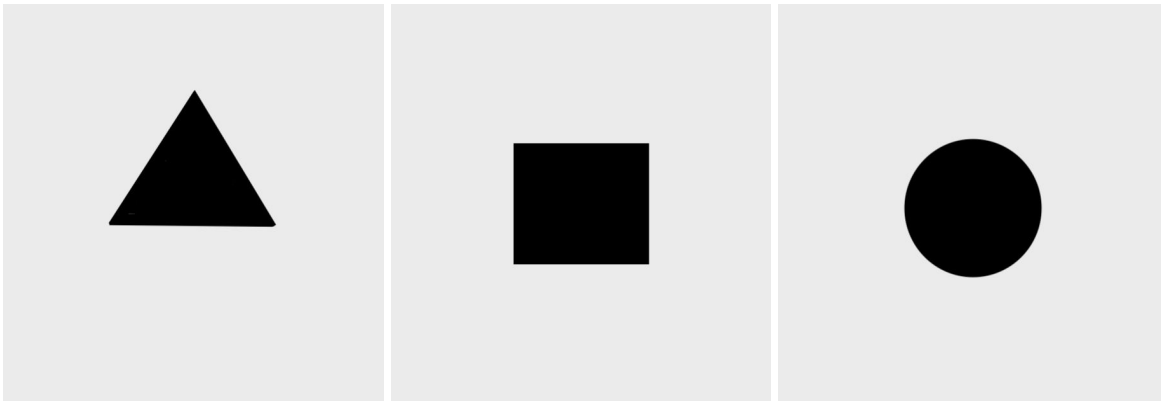


Figure 3.1.1. Textures on the wall

Figure 3.1.2. Exhibition area

The exhibition area is given in figure 2. It is a simple environment because it was designed for primary school students. The turtlebot takes the tour participants from the entrance. There are some table, bookshelf, chair etc. There are 3 different shape on the wall. The environment is fully suitable for the implementation of other packages.

3.2 Mapping

The mapping is next part from work package 1. We used templates from our assignments for mapping the environment. The mapping completed following the gmapping tutorial from wiki.ros.org. The package provides laser-based Slam. We use the Rviz for visualizing the mapping process. Then mapping completed using teleop, we drive the robot using keyboard on the map. Checking map on the Rviz we visited everywhere. Lastly, we saved the map to map server using map saver argument.

3.3 Localisation

The last and easiest part of the work package 1 is Localisation. We just tested the result taken from mapping. The Amcl package from wiki.ros.org was used. The Amcl tries to match the laser scans to the map. It tries to detect if there is any drift occurring in the pose estimate based on the odometry. We launched the navigation demo (our yaml file) and Rviz. Then tested the localisation sending some 2D navigation goals etc. Work package 1 finished with the end of localization. It has finished the according to sche
duled time.

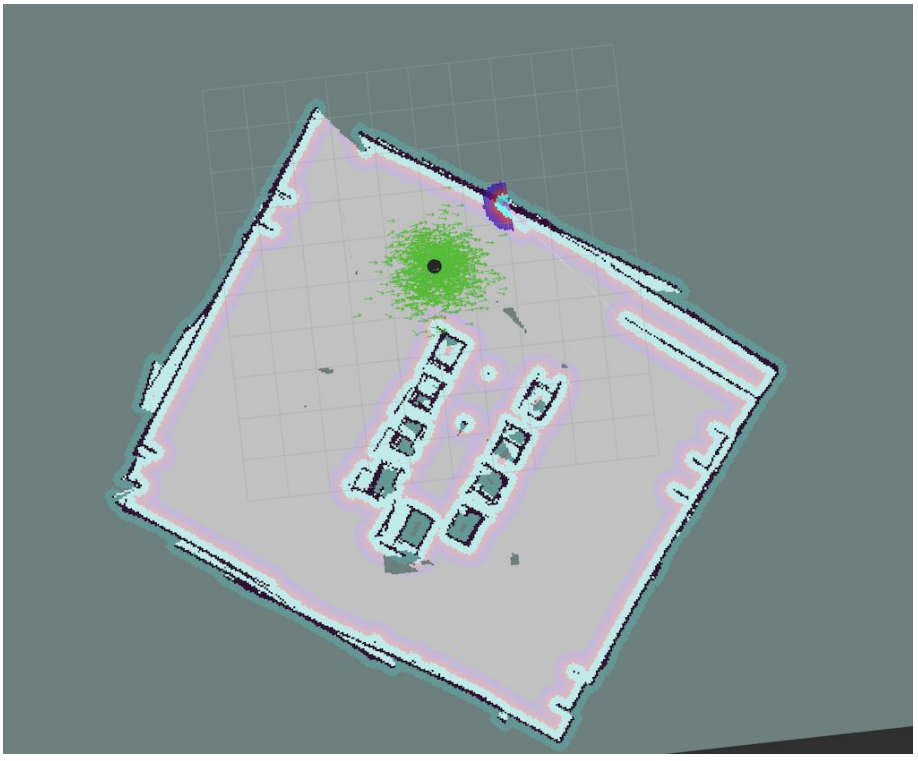


Figure 3.3.1. Localization on Rviz

3.4 Path Following and Path Finding

The crucial part of the project is movement. We are familiar with making movement (following wall, following waypoints, escaping from obstacles etc.) from assignment. We study on move_base package. It attempts to reach given goal with a mobile base. It related to local planner and global planner, it consider local cost and global cost.

Our waypoints determined from Rviz, using publish point button we get the coordinates. Also we find the quaternion values setting some navigation goals from rviz, quaternion values are used for determining the orientation. Then we put them into a Yaml route file. We examined official Yaml documentation from https://yaml.org/spec/1.2/spec.html for saving jpg photos from given specific coordinates.

Also we faced another problem when the turtlebot was going to waypoint it turns around too much. After some search, we realized that this was due to more rotation acceleration.
When the robot turns to the right orientation, it can't stop because the acceleration is too much. Because of the acceleration it lost the right orientation it tries to turning again. Default ros param of the max_rot_vel is 5. We decreased the velocity param to 1.

When we defining route we should consider camera view. Some of the waypoints setted for good view for image processing. Firstly robot turns the wall to be able to detect the shape then goes to there. Gives some basic information while going to wall. It does same thing for every shape then goes the initial point that is near the entrance.

3.5 Area Detection and Image Processing

The turtlebot follows the given route then taking photos when it comes to predefined points. We work with /camera/rgb/image_raw. We use image_view package. It is a simple image viewer for ROS image topics. We have determined photo capture points from the camera in the rviz. We have added a camera then choose rgb camera. When the rgb camera good point (figure 4) for processing we note the coordinate of point. Using cv_bridge[7] images converted from ROS images to OpenCV images. Then captured images are formatted as jpg. After applying the gaussian filter for denoising the image, we applied the threshold for getting 2 color image which are black and white. cv2.findContours function finds the contours then our shape detector method defines the shapes by counting edges. We set the Rviz displays, added a camera /camera/rgb/image_raw then saved our Rviz configuration file. We load Rviz with our configuration file from launch file.

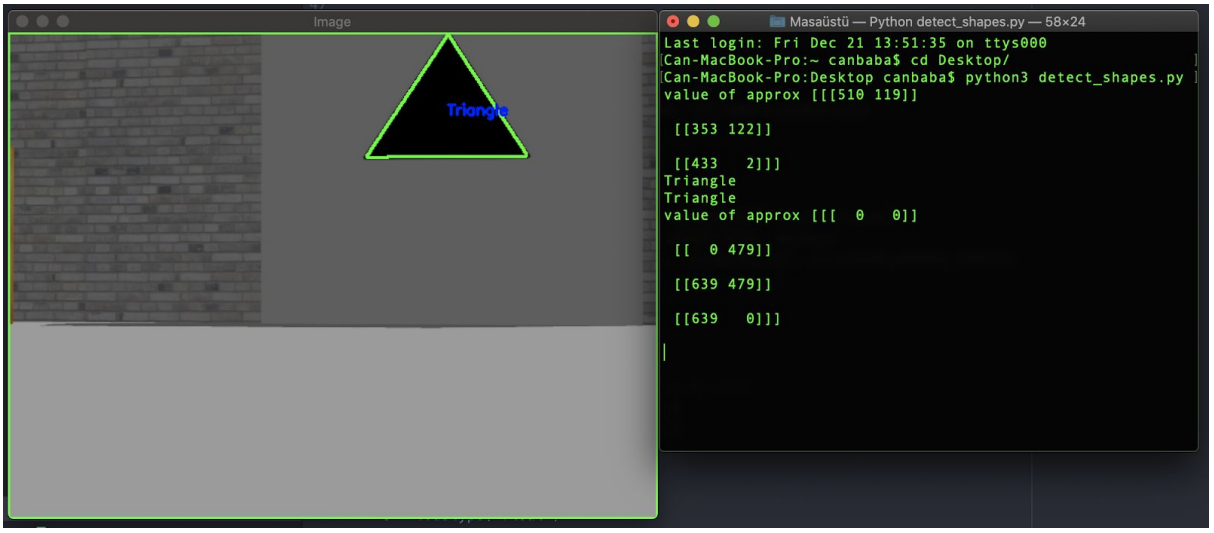


Figure 3.5.1. Detecting triangle and finding edge points.

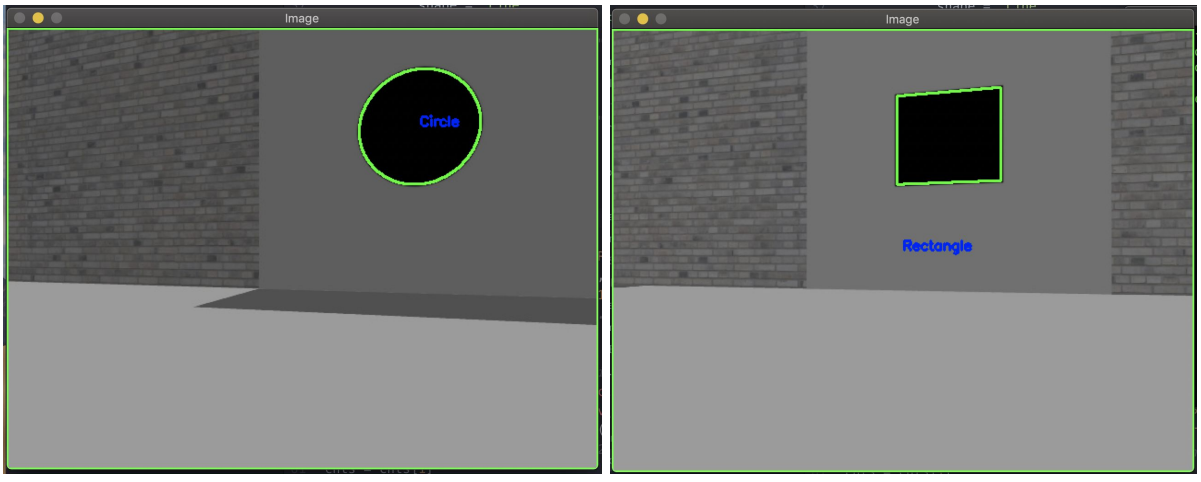


Figure 3.5.2. Detecting circle and rectangle.

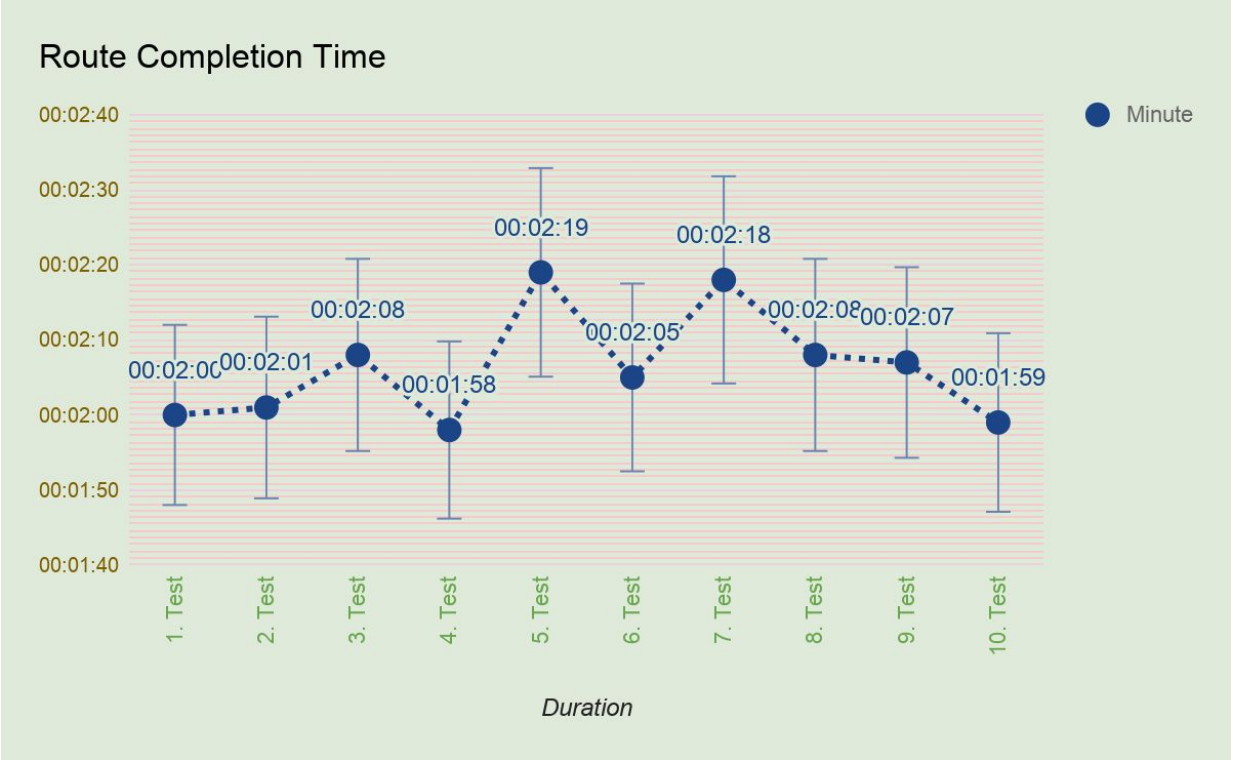
4. Analysis and Results

Task	Start date	Estimated Finish date	Finish date	Estimated Amount of time (hours)	Amount of time	Share of marks (%)
WP1:Work Package 1	25 Nov	7 Dec	10 Dec	24	24	50%
WP1.1: Environment	25 Nov	28 Nov	1 Dec	8	**12**	10%
WP1.2: Mapping	30 Nov	7 Dec	10 Dec	8	8	30%
WP1.3: Localisation	30 Nov	7 Dec	10 Dec	8	4	10%
WP2:Work Package 2	10 Dec	23 Dec	23 Dec	40	38	60%
WP2.1: Path finding	10 Dec	21 Dec	23 Dec	10	10	15%
WP2.2: Path following	10 Dec	21 Dec	23 Dec	10	10	10%
WP2.3: Area detection	15 Dec	20 Dec	20 Dec	5	3	5%
WP2.4: Image detection	15 Dec	20 Dec	20 Dec	15	**8**	30%
Report Preparation & Video Demo	21 Dec	23 Dec	23 Dec	5	5	-
Total	25 Nov	23 Dec	23 Dec	69	65	**110%**

Table 4.1. Main Tasks and Schedule

The project schedule prepared before the proposal meeting, after the first meeting with professor we edited schedule. The proposal defence meeting very useful.

Most part of the project follow the scheduled time plan. Preparing the environment took more than estimated hours, because following the tutorials about creating new models took more time. On the other hand image processing part took less time, because we have learnt image processing from computer vision course this term.



Graph 4.1. Rote Completion Time Comparison

Most of the finishing time around the 2 minute. We tested turtlebot 10 times (Graph 4.1) to see if it works stable. The average completion time is (02:06) 2 minutes 6 seconds.
Sometimes (10%) turtlebot reach the first waypoint longer than usual time. If it happens, It does not change the process all the route completed with some delay.

Also we tested many times shape detection part of the project to understand in what angle or view we need to take pictures. If the turtlebot sees the table at an angle of 45 degrees, it can process the image. To understand we captured different jpg images from turtlebot camera then try to shape.

5. Conclusion

In our project we are trying to help people in exhibition areas. Traditional methods might take human effort and other solutions such as headphones and etc is can be distractor. Turtlebot will go to exhibition area then will give information about exhibited things and it makes it more interactive and funny. Robot will do it our path finding algorithms and object recognition processes.

Not only a fix a real world problem and also start something new make us more excited about project. When we will see that on the news reel robots guides people in the galleries and the museums, probably we will watch it with a great smile on our faces.

The teaching assistant Mr. Kıvrak really helped us a lot when we stuck. Thanks to his referrals we were able to complete the packages. We met him 2 times, first about the mapping part, the second time we have received advice about move base.

Through the project we learned how to work with the simulator through ros. By doing a simple project, we gained experience for later. We learned how to do making environment, mapping, localization, path finding and path following. The project completed in accordance with the time schedule. We have worked together.

6. Reference List

[1] Al-Wazzan, A., Al-Fahran, R., Al-Ali, F. & El-Abd, M. (2016). Tour guide robot, 2016 International Conference on Industrial Informatics and Computer Systems (CIICS), Sharjah, United Arab Emirates: IEEE.

doi: 10.1109/ICCSII.2016.7462397

[2] Burgard Wolfram et al., Cremers, A. B., Fox, D., Hahnel, D., Lakemayer, G., Schulz, D.,...Thrun, S. (1998). Experiences with an interactive museum tour-guide robot. Artificial Intelligence, p.2. Retrieved from:

http://robots.stanford.edu/papers/thrun.tourguide.pdf

[3] ROS.org | Powering the world's robots. (2017). Ros.org. Retrieved 22 October 2017, Retrieved from: http://www.ros.org/

[4] Gazebo. (2017). Gazebosim.org. Retrieved 22 October 2017, Retrieved from: http://gazebosim.org/

[5] rviz - ROS Wiki. (2017). Wiki.ros.org. Retrieved 22 October 2017, Retrieved from: http://wiki.ros.org/rviz

[6] turtlebot_capabilities - ROS Wiki. (2017). Wiki.ros.org. Retrieved 21 October 2017, from http://wiki.ros.org/Robots/TurtleBot

[7] Converting between ROS images and OpenCV images (C++) - OpenCv Answers: Open Source Q&A Forum. (2015). Answers.opencv.org. Retrieved 13 July 2017, from

http://wiki.ros.org/cv_bridge/Tutorials/UsingCvBridgeToConvertBetweenROSImagesAndOpenCVImages