Final Project Report

BuffEm'Up

Angel Santana, Miguel Vazquez, Nebiyu Tadesse, Stefano DelPiccolo

Abstract

Our project highlights the differences in time between autonomous mapping and mapping with the help of human interaction of an unknown maze map. The big question we wanted to ask ourselves was, “Does human interaction have a big part in mapping an unknown map faster than if it was a robot exploring the map on it’s own?”

Introduction and Background

Since the goal of our project was to see if human-computer interaction (human-input) was faster at mapping an unexplored map, we wanted to explore a maze. This gave us the opportunity to go around corners, curves, and dead ends. It allowed us to have a more complicated trial rather than have a straight line with just corners for turning. Along with that, we were able to create a robust and complicated map. To test our algorithm for human-imputed exploration, we used a similar technique like in Lab 7 with Teleop. We decided to use the “turtlebot3_teleop_key” to allow us to go through the maze as the turtlebot3. Along with customizing the file, we also went with using the builtin turtlebot3 camera function that allowed us to see what the turtlebot3 was seeing. Once we got that down, we could move on to the turtlebot3 mapping the maze. We then had to find a way to autonomously explore the map with our turtlebot. This technique would help us explore the maze and map it without our interaction. We tested numerous packages but settled on one, the ‘explore_lite’ package. With this package, we were able to fully execute the autonomous mapping of the unexplored maze without any compatibility or version problems. With both of these tests coming together, we finally were able to test and record our results to solve the question if human interaction has a big part of faster mapping than an autonomous robot.

Methods

In our project we use ROS as the main system. From there we did all of the other parts separately for them to work together and do our tests. The first step was to make the simulation environment. We knew we wanted to make a maze, so Nebiyu (our groupmate), did some research on how to create the maze. We did some research on some tutorials and were able to create the world and launch files. We placed those in the prospective folders, ‘worlds’ and ‘launch’. We would then run a *catkin make* so that they could work with the command *roslaunch.* After we created the simulation environment, Stefano and Miguel (our other groupmates) began work on researching different ways we could control a turtlebot autonomously. Stef was exploring possibilities with the package ‘explore_lite’ and its capabilities, while Miguel studied how the system would work with our unexplored maze and mapping software Rviz. Since the maze was done and we were able to launch the maze with Gazebo, we then used some of the mapping techniques we learned in Lab 7 to start Rviz to begin the test. After the research that Stefano did, we were able to complete the tutorial and were able to successfully autonomate the turtlebot. The next step was to use the package and run it with our map. Since we knew that ROS is stack like (run one thing, then the next), we ran our gazebo map, Rviz for mapping, then after, *turtlebot3_navigation move_base.launch.* We were fixing a bug where there were errors with the move_base server. This was preventing the turtlebot from being able to move. Stefano and Miguel had to test and run multiple times to see how we were able to fix it. After that was fixed, we ran our 'explore lite' package, *roslaunch explore_lite explore.launch.* It was finally able to move and explore our map by itself. Unfortunately the explore_lite isn't perfect. When exploring the maze the robot doesn't know how big the map is. There will be points in the video where you can see the Rviz stutter. This is because the robot realized the maze was bigger than it thought and it needs to resize maps. This can cause it to get "lost" in the maze. Which can cause the robot to not recognize some of the paths of the maze. To get around this human input would be needed once or twice. We ran the test to see how long it would take to explore the whole map to get out of the maze and finished in about 30 minutes. Once Stefano and Miguel finished the autonomous research, compatabilies, and testing, Angel, our other teammate, started the human-input piece of the project. Knowing that there was already a similar form of human-input to move around the maze, using "turtulebot3_teleop_key", it was decided that Angel would use a similar code. However, it was designed to be more user friendly, where someone who doesn't have a big familiarity with ROS/Gazebo/Rviz can understand.

For the Human Interaction part, we wanted to test out how fast a human would be able to exit the maze by only seeing the world from the viewpoint of the robot. To simulate that, we launched gazebo with the final maze and started Rviz with only the camera view enabled for the user and allowed them to control the robot using the "turtulebot3_teleop_key" while the gazebo map was hidden. This was done because we didn't want any advantages given to the human exploring the map with the robot. The modified version of the teleop code was deemed not necessary as the user was able to get a good feeling of the motion of the robot due to the incredible notion humans have for such things. In order to measure the time it took for the human to go through the maze, we started measuring as soon as the user started moving the robot and ended it as soon as they passed through the exit.

Results/Discussion

After conducting multiple mapping tests on the unknown maze the robot seemed to take the same path each time and averaged a completion time of 30 minutes. The clear winner however was the turtlebot that had human input. This was caused by a multitude of factors. The biggest factor being that the human was able to get input rather quickly on where to go in the maze. Without the user knowing the map, the user was able to see where there was a possible corner, where to go if some of the map didn't need to be mapped based on intuition, and how to face the turtlebot to get the most effective point of views to understand the local layout of the map. With the autonomous turtlebot, it went wherever there was an empty space for mapping. This caused the autonomous mapping to explore the maze in a longer time. Below you can see our results from the tests we conducted.

With Human Interaction	**Autonomous Mapping**
6 Minutes	30 Minutes

*Rounded to the nearest minute