WiFi Localization and Autonomous Map Creation

Eric Maras                          Chester Omenukor                          Moises Holguin                          James Sweetman

esmaras@gmail.com                          ochidimon@gmail.com                          moises_holguin@yahoo.com                          jamestsweetman@gmail.com

Github Code: https://github.com/esmaras/bwi.git

The University of Texas at Austin

ABSTRACT

This project aims to implement WiFi localization using the Monte Carlo Localization method with Bayesian filtering processes. The code will follow Carnegie Mellon University's WiFi localization methods as implemented on their CoBot system. Along with this, the project will also be built upon existing similar programs. The long term goal is to enable the robot to create its own map using the WiFi localization. The project is not currently completed, but the aim is to use a gradient map of WiFi strengths to add in WiFi ports onto the map, which would potentially be used in WiFi triangulation. Ultimately, the robot should be able to use ROS to merge another algorithm for SLAM, such as AMCL, with the created WiFi localization code to create a map autonomously with continuous updates. This ability would allow for the possibility that the robot could create a map and localize in any building with WiFi access.

I. INTRODUCTION

Originally, the project aim was to implement WiFi localization on the Segbots using a predetermined map that Piyush created located at http://farnsworth.csres.utexas.edu/maps. We successfully implemented the WiFi strength signal node that Robert has created in our package to find the WiFi hotspots. First, we created a node that will use an algorithm for finding the strongest WiFi
signal strength. This allowed us to create a program that would make the robot define where the signal strengths are highest on the map using a color gradient in gray scale. Originally, the final goal was to implement triangulation in real time according to a paper on real-time indoor WiFi localization using smart antennas [1]. The paper notes that avoiding an off-line training phase removes the requirement for a database that would give values based on one instance alone. However, we were unable to complete this task. Additionally, this paper also includes useful information on how to use the hokoyu laser, advanced Monte Carlo Localization method, and AMCL to localize. To localize using WiFi, the plan was to implement Monte Carlo Localization with the Bayesian filter. Carnegie Mellon's WiFi localization as implemented in CoBot could be used in the future as a model for this project's code [2]. Instead, we focused on creating a map editing tool that would change the pixels in our image of the map of the floor to reflect varying WiFi signal strengths as a smaller step to a larger goal. This program would eventually aid in finding a relation between distances from the WiFi ports to possibly create an algorithm for placing the locations of the ports onto the map. This would make localization with triangulation much easier and more autonomous. We will also need to use odometry to check the real-time WiFi localization as the “Experiments in monte-carlo localization using wifi signal strength” paper discusses [3]. Additionally, another goal to expand upon our project will be to enable the robot to create its own map using the WiFi localization that it will create coupled with sensory data, including the camera, Kinect, and odometry. This process, known as SLAM, could probably be achieved, but the accuracy of this method can be undesirable at times, especially if the odometry of the robot is slightly off.

II. INITIAL AGENDA

Our first step was to be able to retrieve WiFi signal strengths from the wireless hot spots around the building. This was achieved using Robert's node, which displays this kind of information.
After this task was finalized, we began gathering information about the signal strengths around the building and attempted to start to develop code to determine location from these values. We planned on finding and adapting existing code for the simpler aspects of our project. This would require reading the CMU research papers as well as the Smart Antennae paper, and modeling the basis of algorithms off of their given equations. Creating a calculation for triangulation between the WiFi locations may also have lead to some degree of uncertainty. To ensure that the robot is localized within a certain desirable range of uncertainty, odometry would be used to check against the location calculations given from the WiFi signals. Furthermore, Robert's generous contribution, a node that will publish different readings regarding WiFi signal strength and their respective MAC address, has served useful in identifying the Segbot's location with respect to the hot-spot locations. Ultimately, the robot should be able to use the ROS framework to merge another algorithm for SLAM with the created WiFi localization code to create a map autonomously with continuous updates. Additionally, further tweaking would be necessary upon gaining results from the testing process.

III. WIFI LOCALIZATION SETUP

In order to set up the segbot to localize through the use of WiFi access points, different hardwares and aspects of ROS were used. ROS, the robot operating system, was used to move the robot via teleop, access the strengths of WiFi hotspots, as well as initialize localization through a hardware piece known as Hokuyo. Using a ROS package developed by fellow researcher, Robert Lynch, the WiFi access points within the building became accessible. The package records and stores the decibel strengths of each hot-spot as well as the MAC-address. Along with this package, the AMCL pose feature of ROS was also taken advantage of to estimate the initial location. AMCL accomplishes this by utilizing the information gathered by the Hokuyo laser. The code ran accesses a local map and uses a gray-scale feature in order to track the robot's location and path on the map. Its design draws a 3x3
pixel block for every meter.

IV. METHOD

These are the steps of ROS codes that we implemented explained in detail. Each command specifies a node and its package (roscore must be run on the desktop, but everything else should be executed on the robot, especially sensory nodes):

```plaintext
$ roscore
$ rosrun wifi_lookup wifi.py
$ rosrun wifi_lookup wifi_test
```

The first of these three commands initiate the master node. The next two commands run the python package that accesses the WiFi hot spots and run the code to track the position of the robot.

```plaintext
$ roslaunch segbot_navigation amcl.launch --screen
$ rostopic echo amcl_pose
```

This launches the visualization module that provides a visual for the robot on the computer.

```plaintext
$ roslaunch segbot_navigation rviz.launch
```

The teleop command initiates the means to control the robot via a set of keys.

```plaintext
$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

These three lines should be used if the wifi_test code does not run properly. Remove the CmakeCache file from the src folder and remake and run the wifi lookup.

```plaintext
$ rm CMakeCache.txt
$ rosmake wifi_lookup
```
$ rosrun wifi_lookup wifi_test

V. EXPERIMENT

The experiment consisted of teleoperating the robot with AMCL to localize with the map created with SLAM. The node we created should update our current location by updating the pixels on the map with the current position of the robot. Results are compared to the actual physical position of the robot by looking at the room it is in, proximity to a wall, and distance from certain objects.

VI. RESULTS

The graphing of the robot's location was unsuccessful. As of now the conversion from AMCL coordinates to pixels for the map has been inaccurate. The map pixels are edited but in the wrong position because the scaling inconsistencies.
Despite this drawback, the robot is able to sense the different WiFi signal strengths as it is teleoperated across the room. Figure 1 is an image of the map with edited pixels in the wrong location due to the incorrect scaling factor in our code.

VII. CONCLUSION

The overall goals of this project, to use the Segbots to create a map of its surroundings using WiFi localization and use this to move around in the environment, still lies in the future. Thus far, the project, with corrected scale on the map, allows visualization of the WiFi signal strengths of the strongest recognized WiFi ports in the building. Without being able to map the path and location of the robot onto the map, progression towards the ultimate goal of the project was halted. Additionally, the graphing code to color the map with WiFi signal strengths depends on AMCL, which means that the robot must learn to localize using a laser prior to learning WiFi localization. This restriction could be seen as an undesirable complication. Ideally, WiFi localization could be built without relying on separate localization methods.

VIII. FUTURE

The project could be improved in certain aspects in the future. As of right now, the map and the ability to mark on it with the robots location has been achieved. Later steps would include accurate positioning inside the building and then the creation of a gradient based on WiFi hot spot strengths. Once accomplished this project could be used to assist a multitude of other projects. For example, this would take care of localization in autonomous programs within buildings. This could be applied to projects that require the robot to bring robots to certain areas of a building. In further development, the project could be used to track the starting position and the path and add a destination point on a map that a viewer could see.
ACKNOWLEDGEMENTS

We would like to thank Dr. Todd Hester, Piyush Khandelwal, and Robert Lynch for their help and support throughout the research process. Also, thank you to Dr. Peter Stone for overseeing the entire research stream.
References

