

Autonomous Localization

Jennifer Zheng, Maya Kothare-Arora

I. Abstract

This paper presents an autonomous localization service for the Building-Wide Intelligence segbots at the University of Texas at Austin. The BWI segbots currently localize manually: a human must set a 2D pose estimate on the RViz GUI and then drive the robot so it can accumulate sensor data and determine its location. In order to adopt a more intelligent approach to localizing, we implemented a feature that uses both the global localization service to begin the localization process and the ROS topic “cmd_vel” to drive the robot. We also explored different distances, locations, velocities, and time ranges to determine which amounts for those variables would provide the most accurate results.

II. Introduction

The Building-Wide Intelligence segbots currently localize by either requiring users to manually indicate the robot's location via the 2D Pose Estimate function on RViz or having the users teleop the robot while it attempts to localize via the global localization service. Our project allows the BWI segbots to localize with minimal human guidance as a step towards more autonomous robots.

We used the global localization service in the ROS `amcl` package along with the pre-existing map of the third floor to implement our idea. The global localization service distributes particles across the map, and as the robot moves around the area, gathers data from the sensors and clusters the particles around the predicted locations of the robot. To

drive the robot and accumulate data for the global localization service, we used the `cmd_velocity` topic.

Since the accuracy of localization would be highly dependent on the global localization service, we also experimented with different numbers for variables such as the speed, distance, and location to ensure that our implementation would provide the most accurate results possible.

III. Background and Related Works

Much research has been devoted to addressing the autonomous localization issue. Some approaches involve trying to solve the Kidnapped Robot Problem. The paper “Quick and Dirty Localization for a Lost Robot” by Uwe Gerecke and Noel Sharkey details a way in which a robot can determine its location when placed in a new environment. The robots will create a cluster of locations on a single node and can use reference points to localize. The localization works through three steps: First, the SOM shows several possible locations for the robot, based on sensor input. Each of these locations is incremented by 1 in the evidence vector. Then, the robot must move a small distance, and read in new sensor data. Again, the evidence vector is updated (each possible location is incremented by 1). Lastly, evidence shifting must be performed. This process must be repeated iteratively [1].

A somewhat similar approach was taken in “A Near-tight Approximation Lower Bound and Algorithm For the Kidnapped Robot Problem” by Sven Koenig, Apurva Mudgal, and Craig Tovey. This approach splits the problem into two parts: hypothesis generation and hypothesis elimination. The sensory data helps to create a set of hypotheses, and hypothesis elimination is necessary to limit it down to the exact localization. In this

approach, the hypothesis elimination takes place in stages, with the set of hypotheses being halved in each phase. This happens by classifying each hypothesis h in the initial set as either blocked or traversable [2].

Because we are using the global localization service, our code also works by first dealing with a large set of hypotheses, and then limiting the set based on data from the sensors.

Other works in autonomous localization include “Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans” which introduces two algorithms to evaluate a robot’s relative location. The method for localization involves using sensors and compare sensor data to a map using the algorithms [3]. However, this approach depends on the accuracy of the sensors. To implement this method on the BWI segbots, we would need to account for noise in the sensor data.

“Mobile Robot Localization by Tracking Geometric Beacons” approaches localization by using a geometrical beacon and an algorithm developed by the authors. The algorithm is based on the extended Kalman filter that matches between the beacon and a map, using geometry to pinpoint the location of the robot. There is also validation gate that accounts for any noise when localizing [4]. Unfortunately, we would not be able to implement this exact method as the robots in the paper use sonar rather than lasers.

Finally, "Monocular Vision for Mobile Robot Localization and Autonomous Navigation" proposes a localization method using a camera and outdoor landmarks. This method involves recording a video sequence, building a 3D map from the sequence, and using the map to localize [5]. The approach involves a human initially driving the robot in order to record the video; however, we want to have the robot localize with minimum human involvement. We also cannot implement our localization the same way the authors

because our “landmarks” are more subject to change. Our project relies on the fact that the robot will be localizing in the lab, rather than a big open space outside.

IV. Technical Approach

We first experimented with the accuracy of the global localization service. Using the v3 BWI segbot, we varied the number of min and max particles in amcl.launch file and varied the speed we used to teleop the robot.

Min particles	Max particles	Speed	Successful?
40,000	160,000	0.5	no
25,000	100,000	0.5	no
10,000	40,000	0.5	yes
10,000	40,000	0.44	yes
10,000	40,000	0.39	yes
10,000	40,000	0.25	no
5,000	20,000	0.5	no

We concluded that 10,000 min particles and 40,000 max particles had the most accuracy when attempting to localize using the global localization service. We also found that the speed at which the robot is moving does affect the localization: 0.39 is the slowest the robot could move while still localizing accurately.

To implement our solution, we first attempted to use the `move_base` topic and set goals to move the robot forward and to spin so that the robot would accumulate sensor data. However, because we were setting goals before the robot was localized, the robot was unable to generate a path. To address this problem, we used the `cmd_vel` topic and set the linear speed in the x direction to move the robot forward.

Furthermore, we attempted two different approaches for the path of the robot to determine which path would provide the most accurate results. We first tried having the robot move at a linear velocity of 0.5 m/s in a straight line down the hallway outside the lab for 20 seconds. We also tried moving the bot straight at a linear velocity of 0.5 m/s for 5 seconds, and then rotating the bot for 3 seconds, repeating this entire process for 32 seconds.

V. Evaluation and Example Demonstration

The robot localized better around open areas that had distinct barriers, such as the cubicles near the doors to the elevators. Generally, it would not localize while driving through the hallway where it was surrounded by walls but would localize in just a matter of seconds after reaching open space at the end of the hallway.

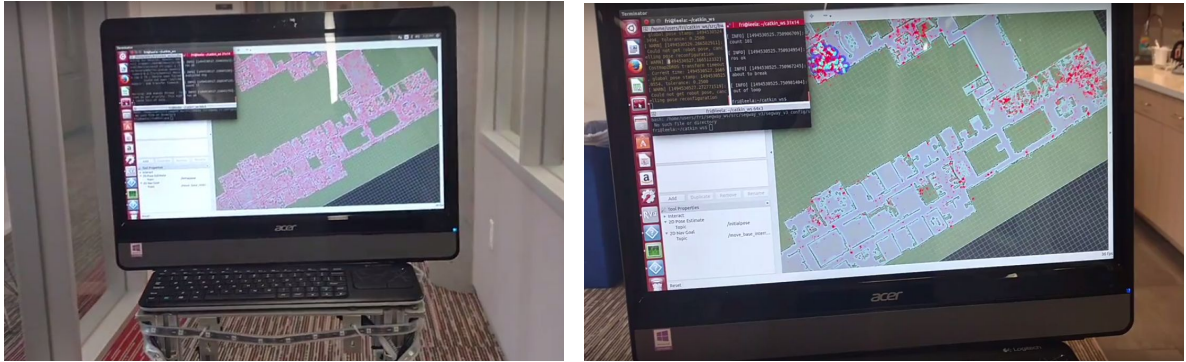


Figure 1: The global localization service is called, and as the robot moves, the particles begin to cluster. Note that the robot did not localize accurately when the path was from the BWI lab to the lounge.

We concluded that the reason that it would not localize until the end of the hallway was because of homogeneity of the sensor data gathered while driving through the hallway. Since the third-floor lab has many hallways, it was hard for the robot to determine which one it was in. Once it reached the cubicles, however, the data gathered was distinct enough for the robot for the global localization particles to cluster to the correct location. The most favorable path was from the lounge area near the lab to the doors leading to the elevators, as shown in Figure 2.

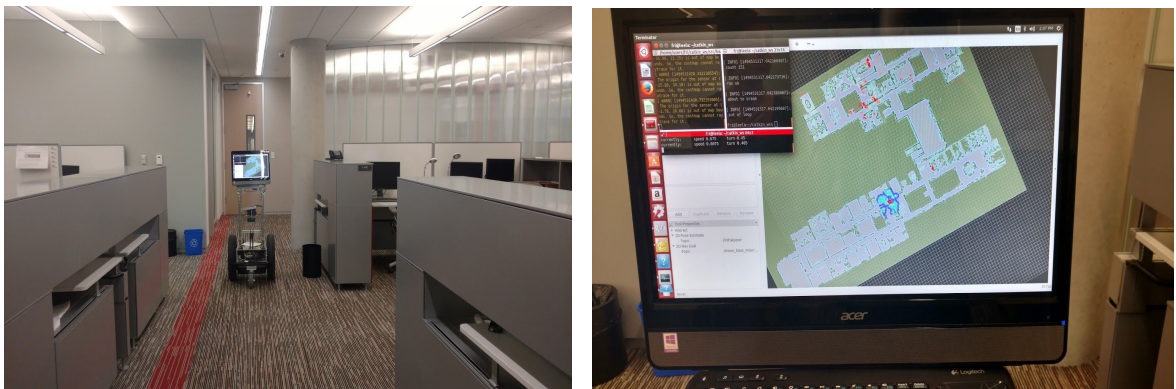


Figure 2: The robot successfully localized once it reached the cubicle area.

We also discovered that having the robot rotate every so often as it advances down the path as opposed to just moving in a straight line down a path did not give better results. We deduced that the reason for this was because the v3 segbot has 360° sensors, therefore already has the accumulated data from the area surrounding it.

Demonstration: <https://www.youtube.com/watch?v=gQGCWbj2h5I&feature=youtu.be>

Code: <https://github.com/Jennifer-Zheng/Autonomous-Localization>

VI. Conclusion and Future Work

Our code was able to call the global localization service and move down the hallway far enough to localize; however, the accuracy of localization was not consistent.

Additionally, because we had to use the `cmd_vel` topic, the robot will not detect obstacles while running our code, and therefore, it is not safe to run without a human supervising the robot yet. Ideally, we want the robot to be able to carry out this process without human supervision. In the future, we could potentially fetch data from the map to estimate where obstacles are and make it safer for the robot to operate on our code even while subscribing to the `cmd_vel` topic.

VII. References

[1] Gerecke, Uwe, and Noel Sharkey. "Quick and Dirty Localization for a Lost Robot." *Computational Intelligence in Robotics and Automation* (1999): 262-67. *IEEEExplore*. Web. 18 Apr. 2017.

[2] Koenig, Sven, Apurva Mudgal, and Craig Tovey. "A Near-tight Approximation Lower Bound and Algorithm for the Kidnapped Robot Problem." *SODA '06 Proceedings*

of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm (2006): 133-42.
ACM Digital Library. Web. 4 May 2017.

[3] Lu, Feng, and Milios, Evangelos E. "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans." 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (1994): 935–938.

[4] Leonard, John J., and Hugh F. Durrant-Whyte. "Mobile robot localization by tracking geometric beacons." *IEEE Transactions on Robotics and Automation* 7.3 (1991): 376-382.

[5] Royer, E., Lhuillier, M., Dhome, M. et al. "Monocular Vision for Mobile Robot Localization and Autonomous Navigation." *Int J Comput Vision* (2007) 74: 237.