WiFi Localization for Mobile Robots

A project of the 2019 Robotics Course of the School of Information Science and Technology (SIST) of ShanghaiTech University

https://robotics.shanghaitech.edu.cn/teaching/robotics2019

Ningzhi Zhu and Yu Zhou

November 2019

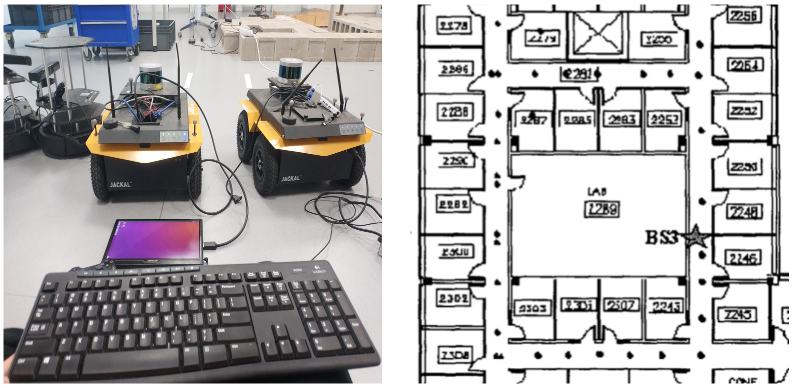
1 Abstract

Localization for mobile robots can be challenging in various environments. But WiFi localization for robots provides a low-cost method or better effect while integrated with other localization technologies, because of high WiFi coverage in environments. For this purpose, we propose to do WiFi localization for mobile robots. Specifically, we want to combine the positioning capability of the SLAM algorithm with the WiFi localization methods, and use fingerprinting to capture the spatial features of the WiFi signal strength. In the end, mobile robots can achieve relatively better accuracy of WiFi localization in a known environment.

2 Introduction

With the introducing of WiFi routers and the widely deployed low-cost WiFi infrastructure, WiFi signal strength provides a great opportunity for the localization of mobile robots. The advantages of ubiquitous infrastructure and no hardware modification requirement make WiFi localization be extensively studied in the past years. We propose to realize WiFi localization for mobile robots based on WiFi fingerprinting or other methods, to demonstrate its potential in positioning. In short words, We plan to use robot in Figure 1(a) to build a WiFi signal strength map like in Figure 1(b). Then, next time the robot comes to the map environment, it can realize WiFi localization.

WiFi fingerprinting based localization consists of two phases: offline and online. Offline is to collect the data to build a RSS map. In the online phase, we a technique called "Nearest Neighbor in Signal Space" to decide the mapping relationship between the real-time date and the RSS map. Finally, we will verify the results. Our ideas are as follows: We use SLAM to get the real map. In the same time during SLAM, we use "iw" command tool on linux to get RSS information to build a RSS map. Then we build the algorithm like "Nearest Neighbor in Signal Space" to localize the robot, when it comes to the environment next time.



(a) Jackal Robot (b) RSS Map

Figure 1: WiFi Localization for Mobile Robots

3 State of the Art

3.1 Paper

[1] is the first paper which uses WiFi to locate the user indoors. [6] uses WiFi Localization for an Autonomous Mobile Robot. [4] focuses on improving Gaussian Processes based mapping in WiFi .

The current research trend is to define the application scenarios more carefully and pursue low cost and high localization accuracy as much as possible. [2] proposed a new approach of WiFi localization and navigation for indoor autonomous mobile robots, and [5] introduced a new idea to improve the accuracy. What's more, [3] expanded WiFi localization from 2D to 3D space and further improved positioning accuracy by virtue of prior information provided by other algorithms.

In [5], authors developed a system for WiFi signal strength-based indoor localization and implement two approaches. The first is improved KNN algorithm-based fingerprint matching method, and the other is the Gaussian Process Regression (GPR) with Bayes Filter approach. Both approaches can be broken up into two stages. The first is the calibration stage, during which the robot records the signal strength at different positions and builds a fingerprint database. In the second stage, which we call the measurement stage, the robot determines its position by pattern recognition and probabilistic methods.

This paper [5] conducted experiments to compare the improved KNN algorithm with the classical KNN algorithm and evaluate the localization performance of the GPR with Bayes Filter approach. The experiment results of [5] show that the improved KNN algorithm can bring enhancement for the fingerprint matching method compared with the classical KNN algorithm.

In addition, this paper observed that the distribution of the WiFi signal strength normally obeys a Gaussian distribution. This intuitively let authors of [5] approximate the RSSI filed as a Gaussian Process. The Gaussian Process Regression (GPR) can predict the RSSI at the unrecorded positions. It is worth emphasizing that the GPR with Bayes Filter approach can provide about $2m$ localization accuracy for the test environment.

3.2 Methodology

As mentioned in introduction above, our methodology is to use robot in Figure 1(a) to build a WiFi signal strength map like in Figure 1(b). Then, next time the robot comes to the map environment, it can realize WiFi localization.

We firstly use SLAM to get the real map. In the same time during SLAM, we use "iw" command tool on linux to get RSS information to build a RSS map. Then we build the algorithm like "Nearest Neighbor in Signal Space" to localize the robot, when it comes to the environment next time.

So, the involved methodologies are about "SLAM", "RSS Map" and "Nearest Neighbor in Signal Space", which will be described in detail through the System Description below.

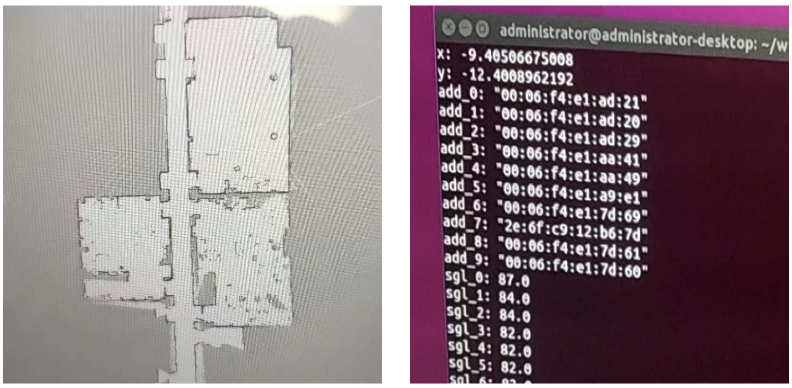
4 System Description

4.1 SLAM and RSS Map

SLAM and RSS Map belong to the offline phase, which is to collect the data to build map for the online phase next time.

As Figure2(a) shows, we use SLAM to get such a real world map, which can be helped by ROS package "gmapping". Only such a real world map is not enough. What we need is a RSS map. So we need to get signal strength information to make the real world map become a RSS map.

Signal strength information is shown in Figure2(b), which means that we get the signal strength information in the position (x,y) in the map. We realize this by using the "iw" command tool in Linux.



(a) SLAM (b) Signal Strength information

Figure 2: SLAM and RSS Map

4.2 Nearest Neighbor in Signal Space

The WiFi signal strength is measured in decibel in milliwatt as dBm with the following formula:

$$RSSI(dBm) = 10log\frac{ReceivedPower(mW)}{1mW}$$

So, theoretically, different places in the map may have different signal strength, which means that if two points are neighbors in signal space, they may be neighbors in real space. So we uses a technique called "Nearest Neighbor in Signal Space" to decide the mapping relationship between the real-time date and the RSS map:

$$dis = \sqrt{(Rss_1 - Rss_1')^2 + (Rss_2 - Rss_2')^2 + (Rss_3 - Rss_3')^2}$$

4.3 Algorithm

The whole process based on "SLAM", "RSS Map" and "Nearest Neighbor in Signal Space" can be described with the following algorithm. We firstly use SLAM to get the real map. In the same time during SLAM, we use "iw" command tool on linux to get RSS information to build a RSS map. Then in online phase, we use "Nearest Neighbor in Signal Space" to localize the robot:

Algorithm 1 WiFi localization for mobile robots

OfflinePhase: Build RSS map
1.Run SLAM by gmapping
2.Get RSS by iw command tool
3.Build the AP mac vector
4.**return** *RSS map*

OnlinePhase: Localize the robot
1.Get RSS by iw command tool
2.Build the RSS vector according to AP mac vector
3.Find nearest neighbor

```
while TRUE do
  for i ⇐ 0 to L do
    if dis[i] less than min then
      min ⇐ dis[i]
      prediction ⇐ position[i]
    end if
  end for
end while
```

4.**return** *prediction*

5 System Evaluation

In order to verify the rationality and specific accuracy of the WiFi positioning algorithm in this report and pave the way for further research in the future, we conducted relevant experiments.

5.1 Software and Hardware Preparation

According to the robotics course requirements, we performed experiments on the J-5 robot in the MARS laboratory. The J-5 robot is a Clearpath robot that uses Linux as its operating system. In addition, this robot is equipped with a dedicated robot operating system ROS, and is equipped with Velodyne16 laser sensor. The above makes all our hardware conditions.

In order to achieve WiFi localization, we need to use ROS's topic/message mechanism. To this end, three ROS nodes were implemented in C ++. The role of the first ROS node was to obtain the SLAM gmapping positioning results, the WiFi signal strengths of the current environment, and the corresponding Mac addresses. The role of the second ROS node was to process the positioning coordinates and WiFi strength, implement WiFi mapping, and preprocess data for subsequent localization. The role of the third ROS node is real-time positioning based on only the WiFi signal strengths of the current robot's location.

What needs to be further explained is that, because there are many Wi-Fi addresses that can be detected in the MARS laboratory, the first ROS node only keeps the 10 strongest Wi-Fi signal strengths and the corresponding Mac addresses per scan.

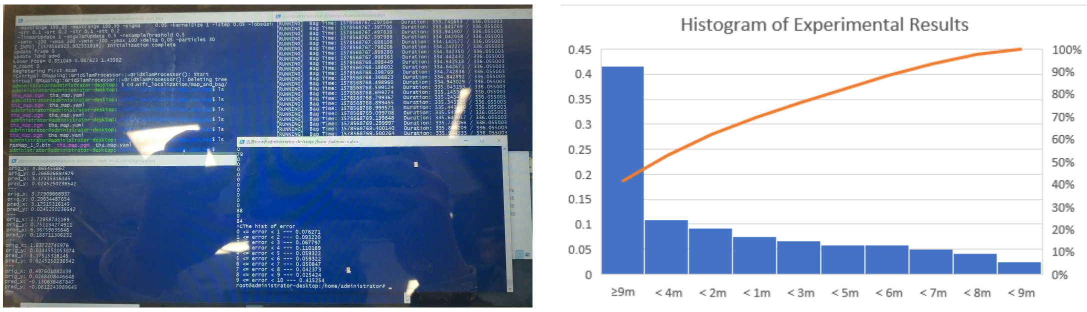
5.2 Design of the Experiment

The whole experimental process is divided into three parts.

In the first part, we drove our J-5 robot in three laboratories and aisles around the MARS center. At the same time, the J-5 robot began to use Velodyne16 and SLAM algorithms for positioning, and continuously published positioning coordinate results on ROS topic / tf. We continuously use the first ROS node mentioned above to collect these positioning results, and WiFi strength information in the relevant environment, and published them with the topic / test. In order to use these data later, we used the ROS rosbag command to monitor and record all messages with the topic / test. For related files, see the /code/wifi_localization/map_and_bag/ folder under the project repository in GitLab.

In the second part, use the prepared second ROS node to process the .bag file obtained above, obtain the correspondence between WiFi signal strength and coordinates in the MARS center environment, and process it into a form that is easy for subsequent processing.

In the third part, the SLAM algorithm, the first and the third ROS nodes are turned on at the same time for real-time WiFi localization. Among them, the first node collected the WiFi strengths with their Mac addresses and the coordinates determined by



(a) Running Programs

(b) The Histogram

Figure 3: The Experimental Results

SLAM, and published these data with topic / test. At the same time, the third ROS node obtained the data processed in the second part, combined it with the WiFi signal strength in / test, and used the algorithm in this report to achieve real-time coordinate localization. In order to facilitate the judgment of the experimental results, the third node sends the predicted coordinates and the coordinates determined by SLAM under the topic /wifi_loc at the same time, and simultaneously calculates the positioning accuracy and the accuracy histogram.

5.3 Results of the Experiment

As shown in Figure 3, in the 60% case, the WiFi localization algorithm of this report achieves relatively normal accuracy. But forty percent of cases fail. We carefully analyzed the reasons for this failure throughout the experiment. Because our WiFi localization is based on the SLAM positioning algorithm, the accuracy of the algorithm in this report must be lower than the SLAM algorithm itself. However, the SLAM algorithm is a vision-based algorithm and is very sensitive to visual interference. Any local visual interference may cause severe fluctuation in local coordinates, which will produce very bad results. During the experiment, the positioning failure occurred when a pedestrian appeared around the robot, or when a narrow doorway entered the laboratory room from the corridor. When the space around the robot is gradually widening and there is less visual interference, the positioning accuracy will return to the normal accuacy level (¡4m). This reason analysis points to the limitations of the experimental procedures in this report and future research directions.

6 How to Reproduce the Project

We can follow the steps below to reproduce the experiments in this report:

- **Step1:** First of all, we need to find a robot that uses the ROS system, is equipped with a laser sensor and installed related packages such as gmapping. Create a new folder anywhere on the robot's computer, and copy the src folder in the

/code/wifi_localization/igmapping folder of the project repository in GitLab to the new folder you just created.

- **Step2:** Then, go to the src folder you just copied, find the three C ++ files under k_tf/src, and modify the absolute path of the rssMap_1_9.bin file to ensure that the code can generate this binary files in the places you can foresee. After completing the above path modification, open a shell in your newly created folder, and use ROS's catkin_make command to create a new workspace and compile the relevant code in the folder src.

- **Step3:** After completing the above compilation, I can start the robot experiment. First, we drive the robot to a fixed position and initialize the coordinates in the ROS system to the origin. Ensure that the coordinate system for subsequent experiments is uniform.

- **Step4:** Open multiple shell windows and source the devel/setup.bash file for the workspace we just created. Then select a window and use the "roslaunch velodyne_points VLP16.roslaunch" command to start the laser (assuming the robot is equipped with a Velodyne16 laser sensor). Use the command "rosrun gmapping slam_gmapping" in another shell to start the SLAM algorithm.

- **Step5:** Then select a new shell, go to the folder where you want to save the .bag file, and start the command "rosbag record -o [filename] /test" to listen and record messages with the topic /test. Finally, in a new shell, enter root mode (this is required), source the devel /setup.bash file in the workspace, and start the command "rosrun k_tf tfListener" to continuously publish the SLAM's map coordinates and WiFi information under the topic /test. We can use "rostopic echo /test" in the new shell to listen to messages (also need to use the source command).

- **Step6:** In the environment where we want to do WiFi localization, drive the robot to start collect data. Note that because we are using the SLAM algorithm, we need to minimize visual interference, avoid the appearance of pedestrians, etc., and try to drive in a more open space. At the same time, in order to improve the accuracy of the algorithm, the robot needs to pass as many locations as possible, and it needs to move slowly (to prevent the WiFi scan command from executing too quickly and causing errors).

- **Step7:** Once the data is collected, we are ready to generate the mapping data file (just the bin file mentioned in step 2). The previous shell can be closed. We need to start a new shell, execut the source command, and start running "rosrun k_tf wifi_mapping". At the same time, in another shell, start to run the "rosbag play [filename]" command. This filename is the bag file we just recorded. When all the messages in the bag file are published. Click "Ctrl + C" in the shell of the "wifi_mapping" node to close the program. Now, the binary mapping file has been successfully generated.

- **Step8:** We can start WiFi localization. First, we need to drive the robot to the fixed position mentioned in step3 and initialize the coordinates to unify the coordinate system. Then,as in steps 4 and 5, execute the following three commands: "roslaunch velodyne_points VLP16.roslaunch", "rosrun gmapping slam_gmapping", and "rosrun k_tf tfListener". Then in a new shell, execute the source command and "rosun k_tf wifi_localize". Execute "rostpic echo / wifi_loc" in another shell. At this point, the robot has started performing WiFi localization.

- **Step9:** Now, when we drive the robot in the environment, we can see the predicted coordinates (pred_x, pred_y) and the SLAM's map coordinates (orig_x, orig_y) by the messages with topic /wifi_loc. The robot's driving requirements are similar to step 6, also needs to drive slowly and avoid visual interference. Finally, we enter "Crtl + C" in the shell of the "wifi_localize" node to end the localization program. The program will eventually return histogram statistics of localization accuracy (based on the pred_x and pred_y, orig_x and orig_y).

- **Note:** This current algorithm implementation can only do WiFi localization in relative small spaces, such as one or two floors. If positioning is to be achieved throughout the campus, some improvements are needed. However, these improvements are pretty simple.

7 Conclusion

Localization for mobile robots can be challenging in various environments. But WiFi localization for robots provides a low-cost method or better effect while integrated with other localization technologies, because of high WiFi coverage in environments. This course project report deeply analyzes the implementation principle behind WiFi localization, studies some of the research results, and presents the classic RSS algorithm in detail. Finally, we implemented the entire experiment process on a real robot based on the RSS algorithm, and verified the rationality and feasibility of the algorithms related to WiFi localization. Our experiments were performed using the positioning function of the SLAM algorithm. Under normal circumstances, this method can achieve better accuracy (¡4m). But because SLAM algorithm is based on vision, it is very susceptible to visual interference or local small environment, which will cause failure in 40% of cases. This also shows the limitations of our experimental process and points the way for future research.

References

[1] Paramvir Bahl and Venkata N. Padmanabhan. *RADAR: An In-building RF-based User Location and Tracking System.* INFOCOM, 2000.

[2] J. Biswas and M. Veloso. Wifi localization and navigation for autonomous indoor mobile robots. In *2010 IEEE International Conference on Robotics and Automation*, pages 4379–4384, May 2010.

[3] M. Jirků, V. Kubelka, and M. Reinstein. Wifi localization in 3d. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4551–4557, Oct 2016.

[4] Renato Miyagusuku, Atsushi Yamashita, and Hajime Asama. *Improving Gaussian Processes based mapping of wireless signals using path loss models.* IROS, 2016.

[5] Y. Sun, M. Liu, and M. Q. . Meng. Wifi signal strength-based robot indoor localization. In *2014 IEEE International Conference on Information and Automation (ICIA)*, pages 250–256, July 2014.

[6] Marshall Sweatt, Adewole Ayoade, Qi Han, John Steele, Khaled Al-Wahedi, and Hamad Karki. *WiFi Based Communication and Localization of an Autonomous Mobile Robot for Refinery Inspection.* ICRA, 2015.