RSSI-Based Wi-Fi Indoor Localization With Accelerometer

Chun-Yen Lee, Chen-Wei Huang
chunyenlee@cs.ucla.edu jimmyhuang29@g.ucla.edu

ABSTRACT

Wi-Fi is ubiquitous equipment that is essential to modern life. With the development of Wi-Fi protocol and techniques, we can effortlessly gain more information while using Wi-Fi, and the Received Signal Strength Indicator (RSSI) is one of them. Using RSSI to achieve localization has been conducted by many researchers. However, it can be inaccurate due to obstacles, blocking, or sudden movement of the user. We argue that simply using RSSI is insufficient to detect users' locations efficiently. To minimize the deviation, we introduce RSSI with the accelerometer. We can adjust users' movement with accelerometer data and achieve a more accurate localization technique. We show that the proposed mechanism outperforms the naive RSSI localization by 66 percent.

1 Introduction

Localization services have gained massive attention recently, as several IoT applications rely on accurate localization services, such as smart cities and smart homes. Localization services can be further categorized into outdoor and indoor localization. GPS is a promising technique for outdoor localization services that can provide precise localization services with error in centimeter scale [4]. As for indoor localization services, techniques such as Wi-Fi and RFID [14] can provide localization services.

Because of the widespread use of Wi-Fi devices, using Wi-Fi is one of the best choices for providing localization services. Several methods are proposed to use Wi-Fi signals to provide indoor localization services. These can be categorized into RSSI-based, Arrival of Angle (AoA) Based, and Time of Flight (ToF) Based [9]. RSSI-Based localization, which uses received signal strength from the Wi-Fi access point (AP) to estimate the localization, doesn't require additional hardware or hardware modification. This advantage allows us to deploy localization services using existing Wi-Fi devices easily. In this project, we focus on RSSI-Based Wi-Fi localization.

However, RSSI-Based Wi-Fi localization services are inaccurate in reality, as the RSSI value fluctuates even if the location of the device is fixed. The main reason is that wireless channels are volatile. The quality of wireless channels can be drastically reduced with fading. Multipath, an effect that reflected are added in the original signal, can also cause fluctuations in RSSI values. A more robust method is needed in order to provide accurate localization services.

Inertial Measurement Unit (IMU) is an electronic device comprising accelerometers, gyroscopes, and magnetometers. IMUs can be easily obtained and installed in devices such as smartphones and smartwatches. With data from IMUs, we can calibrate RSSI-based indoor localization services [6]. Besides IMUs, several sensors, such as barometers, can also improve the accuracy of indoor localization services [2].

In this project, we focus on mitigating the error of RSSI-Based localization using accelerometers. We integrate RSSI-based localization with accelerometers with different methods. We proposed ways of using acceleration values to reduce the error of RSSI-Based Wi-Fi localization up to 66%.

2 Background and Related Works

RSSI-based Wi-Fi localization is a localization service that uses RSSI value from Wi-Fi to estimate the distance of the device. The higher received power in RSSI-based localization means the distance between the device and the Wi-Fi AP is closer, and vice versa. It can be demonstrated from the formula assuming the free space model.

$$P_{Rx} = \frac{G_{Tx}G_{Rx}P_{Tx}\lambda^2}{4\pi^2 d^2} \Rightarrow P_{Rx} \propto \frac{1}{d^2} \tag{1}$$

This method is simple and does not require hardware modifications. However, it is highly inaccurate. When the distance is far away, i.e., low RSSI, minor errors due to noise could lead to tremendous distance errors. Furthermore, it does not work well with multipath.

There are existing works focusing on RSSI-Based Wi-Fi localization using RSSI Wi-Fi fingerprinting for indoor positioning. Hoang *et al.* proposed a recurrent neural networks (RNN) model for Wi-Fi RSSI fingerprints [5]. Li *et al.* proposed a kNN algorithm to calculate the signal difference from reference fingerprints to improve the accuracy of RSS-based localization [8]. Jun *et al.* designed a middleware Social-Loc, which combines Wi-Fi fingerprinting and social interactions to improve indoor localization accuracy [7].

Several papers utilize IMUs to calibrate RSSI-Based localization. Huang *et al.* proposed VariFi, which applies variational inference with gradient descent algorithm to process IMU data and RSS fingerprint to estimate the location [6]. Chen *et al.* applied pedestrian dead reckoning (PDR) algorithms, which use IMU data to assess and calculate the position of the device [3]. To calibrate the drifting problem caused by PDR algorithms, they use the RSS of iBeacon, a BLE service, to mitigate the error.

3 Experiments Setup

3.1 Experiment Devices

The system is shown in Figure 1. In this work, we use Arduino Nano 33 IoT device to get the Wi-Fi RSSI value and acceleration. We utilize the Wi-Fi module and IMU module equipped on the board. The u-blox NINA-W102 wireless module supports Wi-Fi 802.11b/g/n and can connect to open or encrypted networks (WEP, WPA) [13]. The accelerometer value produced by the IMU module is set at ±4 g with a resolution of 0.122 mg, where g is the gravitational acceleration ($9.8m/s^2$). The output data rate is fixed at 104 Hz [1].

We set up the system by placing the Wi-Fi AP and modem at one end and the laptop on the other end, on the same straight line. The Arduino device is fixed at the corner of the laptop, as displayed in the image, with its -x direction (of the IMU module) pointing toward Wi-Fi devices. Additionally, a box is used to maintain the same height across all devices, allowing us to move the device smoothly.

Figure 1: System setup

3.2 RSSI Measurement

In this work, we first consider the free space model, where the path loss exponent is 2. The received power P_{Rx} can be expressed as [11]

$$P_{Rx} = \frac{G_{Tx} G_{Rx} P_{Tx} \lambda^2}{4\pi^2 d^2} \tag{2}$$

where G_{Tx} is the transmitter gain, G_{Rx} is the receiver gain, P_{Tx} is the transmit power, λ is the wavelength of the Wi-Fi signal, and d is the distance between the transmitter and the receiver. We can further simplify Equation (4) as

$$RSSI = \frac{k}{d^2} \tag{3}$$

where k is the fixed constant of the system.

The RSSI can also be expressed as the Log-normal Distance Path Loss form [16],

$$RSSI(dBm) = -10nlog(d) + C \tag{4}$$

where n is the path loss exponent, and C is a fixed constant of the system. With Equation (4), we can use Ordinary Least Squares (OLS) regression to find the parameter C and path loss exponent n. We use statsmodels package from Python [10] to run OLS regression with data.

Because of the indoor environment, the Wi-Fi signals are strongly affected by fading, multipath interference, and shadowing. Therefore, the RSSI values fluctuate even if the device's position is fixed. The Wi-Fi signals with larger RSSI are usually affected by fading, and the weak signals are generally influenced by multiple factors [15]. Moreover, we can mitigate the fading effect by averaging the measurements since most fading is modeled as circularly symmetric complex Gaussian random variables [12]. To measure the RSSI value, we use two ways to calculate the RSSI using the measurements. The first method calculates the average of all RSSI values, which is the **Base** method. The second method calculates the average of the top 50% RSSI values. We use the name **50 %** to indicate the method.

3.3 Acceleration Measurement

We use the IMU unit in Arduino Nano 33 IoT to measure the acceleration of the devices. The overall procedure is described in Algorithm 1. We assume that we know the initial position and initial velocity. Also, we place the device in the positive x direction, assuming that the Wi-Fi AP is in origin. With the algorithm, we can calculate the final position and the maximum velocity estimated using the accelerometer.

Algorithm 1 Acceleration Measurement and Estimation

Input X_0, V_0, δT, T_e
Output X, V_{max}
  $X \leftarrow X_0$
  $V \leftarrow V_0$
  $T \leftarrow 0$
  $V_{max} \leftarrow 0$
  while $T \leq T_e$ **do**
    Measure the acceleration a_i in time T
    $V \leftarrow V + a_i \times \delta T$
    $X \leftarrow X + V \times \delta T$
    $V_{max} \leftarrow max(V_{max}, V)$
    $T \leftarrow T + \delta T$
  end while

3.4 Experiment Procedure

We perform the experiment in the following steps to utilize the accelerometer data while performing Wi-Fi localization.

First, the device is placed at a point. We measured the distance to the origin (Wi-Fi AP) and ensured the device's -x position was pointed directly to the AP. Next, move the device smoothly toward the +x position. The ax value was measured along the way. Finally, the device stopped at random destinations. We then measured the RSSI value (80 to 100 value) at the destination. The above process was conducted at six different starting points.

We can calculate the distance between the destination and the origin with the initial distance and the ax values gained while the device traverses. Moreover, another distance estimation was obtained by leveraging the RSSI values (measured on the destination) with the k value and OLS regression coefficients estimated in the previous experiment.

4 Experiments Results

4.1 Measure the coefficient

4.1.1 Free Space Model

To get the value k, we measure the RSSI value in eight different distances. When collecting these data, we calculate the average RSSI values and the top 50 % RSSI values, respectively.

With the RSSI values and distance data, we can calculate the value k for two different RSSI calculation methods. First, we calculate the value k with each RSSI and distance pair. We average the value k of eight different locations to get the estimated result. The k value for the **Base** method is 1.486^{-6}, and the k value for **50%** is 1.878^{-6}. We plot the curve of both methods in Figure 2, and use a scatter plot to mark the measurements for every location.

4.1.2 OLS Model

For OLS, we find the relationship between distance and RSSI of **Base** can be expressed as

$$RSSI_{Base}(dBm) = -22.6658 \times log_{10}(d) - 28.7088 \quad (5)$$

with $R^2 = 0.778$

For the **50 %**, the relationship between distance and RSSI is

$$RSSI_{50\%}(dBm) = -22.7634 \times log_{10}(d) - 27.6760 \quad (6)$$

with $R^2 = 0.805$

4.1.3 Comparison between OLS and Free space model

In Figure 3, we compare the error distribution between OLS and the free space model. We can see that OLS method performs better with the **Base** method with the most significant error bounded in 1.5 meters. OLS with **50 %** also performs well in half of the cases. Therefore, we consider only the OLS model in the following experiments.

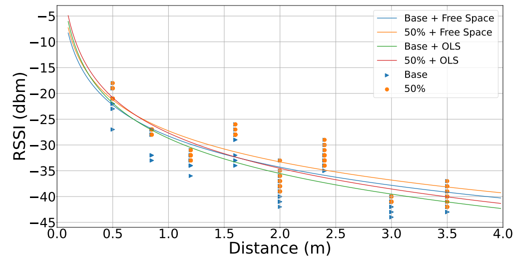

Figure 2: The relationship between RSSI and distance

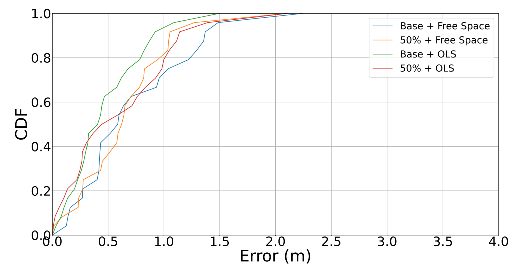

Figure 3: Comparison between OLS and free space model

4.2 Adding accelerometer in Wi-Fi localization

By getting the relationship between RSSI and distance, we can use the RSSI value of Wi-Fi signals to estimate the device's position. Also, by using the acceleration collected by the accelerometer and processed with Algorithm 1, we can also estimate the device's position. The easiest way is to add two estimated values and take the average using the accelerometer to mitigate the measurement error. We plot the CDF distribution of the error in Figure 4.

In Figure 4, the estimation using only an accelerometer is quite accurate. However, the accuracy of estimation using Wi-Fi RSSI values is not precise. When using RSSI-based localization, more than 50% of measurement has errors higher than 0.5 meters. With the estimation of the accelerometer, only 20% of the measurements have distance error larger than 0.5 meters.

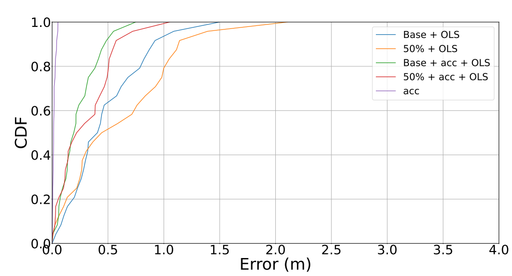

Figure 4: The CDF of error distribution with acceleration values

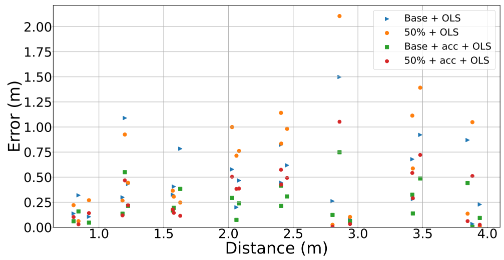

Figure 5: The relationship between error and distance with acceleration values

In Figure 5, we plot the scatter plot to show the relationship between error values and the distance. We observe that when the distance increases, the errors will also increase. The main reason is that RSSI-Based localization needs more accurate RSSI values for more precise estimation if the distance between AP and the device is far. From the results, we successfully reduced the estimation error with the help of accelerometers. However, a high sample rate is needed for accelerometers in order to get a precise estimation. Moreover, from Algorithm 1, the location is estimated using double integrals. If the measurements of the acceleration values aren't precise, it may cause estimation errors. Moreover, doing integrals may amplify these errors. Therefore, a more robust method is needed to calibrate RSSI-Based localization using accelerometers without double integration.

4.3 Filter RSSI values

We can also use the maximum velocity to calibrate the accuracy of Wi-Fi localization. In our experiment, we only move the device in one direction. Therefore, the possible location of the device X_p can be described as

$$X_0 \leq X_p \leq X_0 + V_{max} * T \tag{7}$$

where X_0 is the initial position, T is the total time of movement, and V_{max} is the maximum velocity. With these inequalities, we can filter the RSSI value r that doesn't satisfy the inequality

$$T(X_0) \geq r \geq T(X_0 + V_{max} * T) \tag{8}$$

where T is the function converting distance to RSSI values.

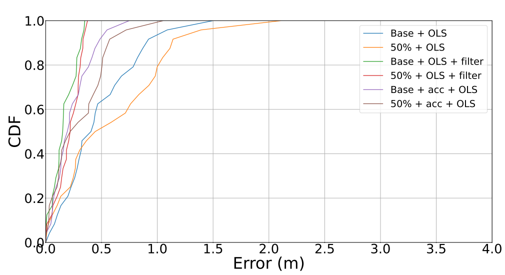

Figure 6: The CDF of error distribution with filtering

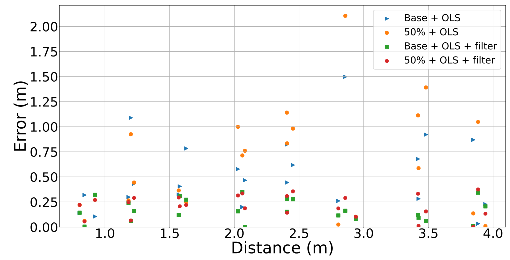

Figure 7: The relationship between error and distance with filtering

We plot the CDF of error distribution and compare it with other previous methods. The results are shown in Figure 6. With the filter algorithm, we can limit the error to 0.5 meters for every measurement. Also, in Figure 7, the filter algorithm can reduce the error in the further distance (3-4 meters). With this algorithm, we can avoid doing double integrals, which can mitigate the effect caused by measurement errors.

We organize the average error for the aforementioned methods in Table 1.

Methods	Average Error (m)
OLS RSSI with Base	0.495
OLS RSSI with 50%	0.627
OLS RSSI with Averaging with Base	0.244 (Reduce 50.6%)
OLS RSSI with Averaging with 50%	0.313 (Reduce 50.0%)
OLS RSSI with Filtering with Base	0.168 (Reduce 66.0%)
OLS RSSI with Filtering with 50%	0.212 (Reduce 66.1%)

Table 1: Average Error of different methods

5 Conclusion

In this report, we present different methods to reduce error by integrating Wi-Fi RSSI-based localization with accelerometers. We set up a system to collect accelerometer data and utilize these data to calibrate the estimation from RSSI-Based Wi-Fi localization. The results show that our mechanism outperforms the baseline RSSI localization by 66 percent.

6 Statement of Work

1. Chun-Yen Lee Report, data processing & plotting
2. Cheng-Wei Huang Report, Ardunio setup & experiment

7 References

[1] Arduino. Arduino wifi spec https://www.arduino.cc/reference/en/libraries/wifinina/.

[2] D. Banerjee, S. K. Agarwal, and P. Sharma. Improving floor localization accuracy in 3d spaces using barometer. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*,

ISWC '15, page 171–178, New York, NY, USA, 2015. Association for Computing Machinery.

[3] Z. Chen, Q. Zhu, and Y. C. Soh. Smartphone inertial sensor-based indoor localization and tracking with ibeacon corrections. *IEEE Transactions on Industrial Informatics*, 12(4):1540–1549, 2016.

[4] W. Hedgecock, M. Maroti, J. Sallai, P. Volgyesi, and A. Ledeczi. High-accuracy differential tracking of low-cost gps receivers. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, page 221–234, New York, NY, USA, 2013. Association for Computing Machinery.

[5] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy. Recurrent neural networks for accurate rssi indoor localization. *IEEE Internet of Things Journal*, 6(6):10639–10651, 2019.

[6] H. Huang, J. Yang, X. Fang, H. Jiang, and L. Xie. Varifi: Variational inference for indoor pedestrian localization and tracking using imu and wifi rss. *IEEE Internet of Things Journal*, pages 1–1, 2022.

[7] J. Jun, Y. Gu, L. Cheng, B. Lu, J. Sun, T. Zhu, and J. Niu. Social-loc: Improving indoor localization with social sensing. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, New York, NY, USA, 2013. Association for Computing Machinery.

[8] D. Li, B. Zhang, and C. Li. A feature-scaling-based k-nearest neighbor algorithm for indoor positioning systems. *IEEE Internet of Things Journal*, 3(4):590–597, 2016.

[9] J. Liu, J. Gao, S. Jha, and W. Hu. Seirios: Leveraging multiple channels for lorawan indoor and outdoor localization. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, MobiCom '21, page 656–669, New York, NY, USA, 2021. Association for Computing Machinery.

[10] S. Seabold and J. Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[11] B. Sklar. Digital communications: Fundamentals and applications, 1988.

[12] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, USA, 2005.

[13] U-Box. U-box spec https://www.u-blox.com/en/product/nina-w10-series-open-cpu.

[14] P. Wang, B. Liang, R. Zhao, P. Zhang, X. Zhang, and C. Xu. An rfid localization system for smart logistics. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, SenSys '22, page 849–850, New York, NY, USA, 2023. Association for Computing Machinery.

[15] W. Xue, W. Qiu, X. Hua, and K. Yu. Improved wi-fi rssi measurement for indoor localization. *IEEE Sensors Journal*, 17(7):2224–2230, 2017.

[16] Z. Yang, Z. Zhou, and Y. Liu. From rssi to csi: Indoor localization via channel response. *ACM Comput. Surv.*, 46(2), dec 2013.