



Improving WiFi Fingerprint Positioning Through Smartphone Built-In Sensors Based Trajectory Estimation

Zhenjie Ma, Ke Shi^(✉) , Xiaomei Song, and Aihua Zhang

School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
keshi@hust.edu.cn

Abstract. Wireless indoor localization is fundamental to many smartphone applications, appealing to a great deal of research efforts in the past decades. Among them, RSSI based fingerprinting using existing WiFi infrastructures has become an increasingly popular technique. However, WiFi fingerprinting still suffers from the vulnerable and changeable wireless signal transmissions. To address this issue, we propose a novel smartphone built-in sensors assisted WiFi fingerprinting method. The data generated from the smartphone built-in sensors such as accelerometer and gyroscope is utilized to estimate the user's trajectory. Then an integrated probability model combining the RSSI fingerprint and users' trajectory is established to implement fingerprint-location match. Performance evaluation results show that the proposed method has higher location accuracy and better adaption to changeable environment.

Keywords: RSSI fingerprint · Built-in sensors · Trajectory estimation · Integrated probability model · Indoor Positioning

1 Introduction

The popularity of mobile and pervasive computing stimulates extensive research on wireless indoor localization. Now the solutions based on pervasive WiFi infrastructure has dominated this field. Received Signal Strength Indicator (RSSI) [1] based localization technique is one of the cheapest and easiest methods to implement. However, RSSI is extracted from the radio frequency signal at a per packet level and tends to fluctuate according to changes in the environment or multipath fading.

To address this issue, RSSI-based fingerprinting [2, 3] is proposed, which uses a two-stage mechanism. In offline phase, the signal strengths at the predefined RP (Reference Point) from several access points are recorded and stored in a database along with their coordinates. During the online locating, the current RSSI vector at an unknown location is compared to those stored in the database and the user location is estimated though some close match.

WiFi fingerprinting still suffers from the vulnerable and changeable wireless signal transmissions. Any changes of the environment may change the fingerprint, which make

the fingerprint obtained in online location different with the fingerprint stored in the database during the offline survey.

Nowadays, smartphones are equipped with various functional built-in sensors such as IMU (inertial measuring unit). An IMU usually consists of the sensors like accelerometer, gyroscope, and compass, which respectively reveal the acceleration, rotational velocity, and direction of user motion. From IMU measurements, user's moving trajectory can be estimated.

The main idea of this paper is to improve the localization accuracy of WiFi fingerprinting by leveraging the user's trajectory information obtained from the built-in sensors to determine the best location match during the online location. Besides fingerprint similarity, user's trajectory is used to evaluate the location result to prevent possible deviation caused by unstable RSSI information. An integrated probability model is proposed, which utilizes the results and estimated errors of RSSI matching and trajectory estimation to give the final position.

2 Related Work

Many approaches have been proposed to implement the WiFi fingerprinting matching process such as RADAR [2] and Horus [3]. Machine learning technique such as SVM (Support Vector Machines) is also proposed, which utilize data filtering rules obtained through statistical analysis to improve the quality of training samples and thus improve the quality of positioning model [4].

Multi-sensor infusion based methods are proposed to improve the positioning accuracy, especially utilizing the IMU already embedded in the most smartphones. Kalman filter [5] and its extension [6] are proposed to combine WiFi positioning with PDR (Pedestrian Dead Reckoning) positioning. It is hard for these methods to adapt the complex indoor environment.

Particle filter based methods [7, 8] are proposed to add motion constrains to the positioning model. However, the particle filter is unsuitable for real-time smartphone positioning as it is computationally expensive and time-consuming.

Besides RSSI, many researches use CSI (Channel State Information) [9] fingerprint to improve positioning accuracy. Compared with RSSI, CSI can provide multi-channel subcarrier phase and amplitude information to better describe the propagation path of the signal. However, collecting CSI needs modifying operating system kernel and customizing hardware drivers, which limits its applications significantly.

Motivated by recent advances in deep learning, some researchers utilize deep learning based models such as CNN to overcome limitations of fingerprint-based localization approaches [10, 11]. But these approaches suffer from the limitation of training labels and only work well at the certain scenario.

Therefore, we still focus on utilizing RSSI and IMU measurements that are widely available to improve the positioning accuracy. We build an integrated probability model combining the RSSI-based positioning with user's trajectory in a lightweight way that makes it more suitable for resource-limited smartphones. It can improve the positioning accuracy and adapt the changeable environment.

3 Proposed Method

3.1 Overview

Figure 1 Trajectory estimation assisted WiFi Fingerprinting. shows the framework of trajectory estimation assisted WiFi Fingerprinting, which consists of three modules, RSSI matching, trajectory estimation, and integrated probability model.

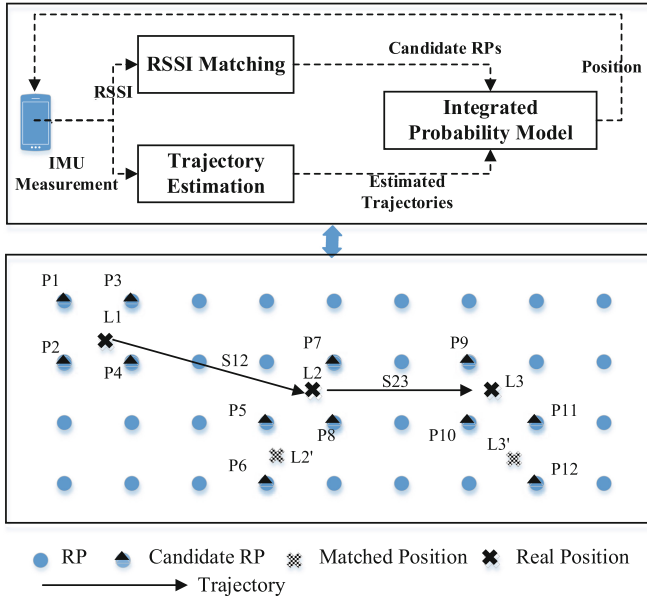


Fig. 1. Trajectory estimation assisted WiFi fingerprinting.

RSSI matching module takes RSSI vector as input and outputs a list of candidate RPs with corresponding probabilities. As shown in Fig. 1, when a user is at location $L1$, $L2$, and $L3$, RSSI matching module outputs candidate RPs list $\{P1, P2, P3, P4\}$, $\{P5, P6, P7, P8\}$, and $\{P9, P10, RP11, RP12\}$ respectively. Due to signal fluctuation, some RPs that have large deviations from the actual positions may be selected as candidate RPs. For example, $P6$ is selected as candidate RP, which cause the estimated position $L2'$ and the actual location $L2$ have a large deviation. There is also a large deviation between $L3'$ and $L3$ since $P12$ is selected.

Trajectory estimation modules utilizes the IMU measurements to get the user's trajectory between the consecutive positions. As shown in Fig. 1, $S12$ is the estimated trajectory between $L1$ and $L2$ and $S23$ is the estimated trajectory between $L2$ and $L3$.

Integrated probability model uses estimated trajectory to amend the selected candidate RPs list and determines the final position. The reachability of selected RP can be evaluated based on estimated trajectory. We can remove the RP with low reachability from candidate list or reduce its contribution to the final positioning result according its reachability. For example, given the trajectory $S12$, a user is unlikely to reach $P6$. Its

probability can be reduced greatly, which makes the final positioning result is closer to the actual position.

3.2 RSSI Matching

Trajectory estimation assisted WiFi Fingerprinting does not require a specialized RSSI matching algorithm. The only demand is the matching outcome should be a likelihood of various RPs given observed RSSI measurements, which allows to estimate the centroid of the all candidate RPs as the solution. The likelihood can be adjusted by trajectory estimation to improve the positioning accuracy.

To determine the extent of likelihood adjustment, we need to estimate the RSSI matching error. It is difficult to get a precise error estimating model in changeable environments. Some positioning accuracy estimating methods based on offline training data analyzing are proposed [12]. In real applications, it is hard to get the training data covering all possible transmission conditions, which causes the error estimation cannot adapt to environmental changes. Therefore we use average inter-candidate distance to infer the positioning error from online measurements.

Average inter-candidate distance is the average distance between the best matching candidate RP and the next $k - 1$ matching candidate RPs. The calculating process contains the following steps:

1. Get the list of the k best matching candidate RPs outputted from RSSI matching algorithm;
2. Compute the distance between the position of the best matching candidate RP and all the other $k - 1$ matching candidate RPs;
3. Return the average distance as the estimated error, ε_{fp} .

3.3 Trajectory Estimation

Nowadays almost all smartphones have built-in IMU. Users' orientation and velocity can be determined with the measurements provided by accelerometers and gyroscopes, and then the relationships between the users' position and a known starting point can be derived. It is known as inertial navigation, which can be used to track user's trajectory.

The built-in sensors are noisy, which brings tiny errors in the measurement of acceleration in inertial navigation. What's more, errors are integrated into progressively large errors in velocity and even larger in position. This process is called integration drift. In order to avoid errors accumulation caused by integration drift, instead of directly integrating acceleration twice over time, we count the steps through analyzing the acceleration changing pattern when user walking. The step length is proportional to acceleration changing in the direction that is perpendicular to the moving direction. This relation can be utilized to determine the length of each step. The distance can be obtained by estimating the number of steps and the length of each step. Owing to high sampling frequency of gyroscope, integrating its measuring value over time to get orientation only causes small errors, so we utilize the gyroscope to estimate the user's orientation.

Obviously, it is inevitable that there will be deviations in the estimation of walking distance and orientation, which cause the error between estimated trajectory and real

trajectory. We use ε_{dis} to represent this error. From previous research [13] and our own experiments, the estimation errors are decided by users' motion smoothness. We can define motion smoothing factor ϕ as follows:

$$\phi = \frac{\mu}{\delta} \quad (1)$$

where μ represents user's average speed and δ represents the mean square variation of user's speed. By using data fitting, ε_{dis} can be derived as follows:

$$\varepsilon_{dis} = \frac{a}{\phi} + b \quad (2)$$

The values of a and b can be calculated by using the result of data fitting as shown in Fig. 2. Relationship between error and smoothing factor.

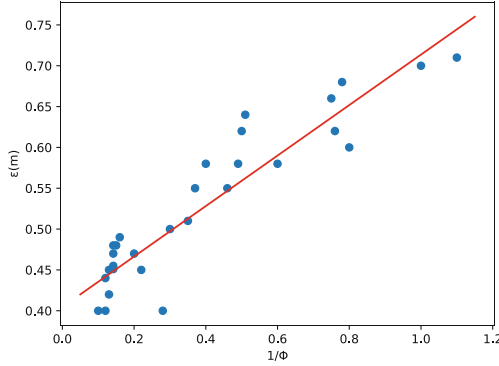


Fig. 2. Relationship between error and smoothing factor.

3.4 Integrated Probability Model

After obtaining the user's trajectory and its error, we propose an integrated probability model fusing the estimated trajectory and RSSI to improve the localization accuracy.

Integrating the Estimated Trajectory from Previous Location to Current Location

As shown in the left part of Fig. 3, there may be deviations between the real trajectory \vec{b} and the estimated trajectory \vec{a} , we need to compute the conditional probability of the real trajectory given the estimated trajectory, $P_{dsp}(\vec{b} / \vec{a})$, which consists of two parts, the distance probability $P_{dst}(\vec{b} / \vec{a})$ and the orientation probability $P_{dir}(\vec{b} / \vec{a})$.

First, there are some constraints that the proposed distance probability model must conform to:

1. The distance difference $\|\vec{a}\| - \|\vec{b}\|$ is positively related with the distance probability.

2. When the value of $\|\vec{a}\| - \|\vec{b}\|$ is fixed, the longer the distance is, the larger the probability is.
3. Since the error of the estimated distance is mainly derived from the estimation of strides, the positive difference has the same probability with the negative difference. Thus, we can get the formula as: $P_{dst}(\vec{b} / \vec{a}) = P_{dst}(\vec{a} / \vec{b})$.

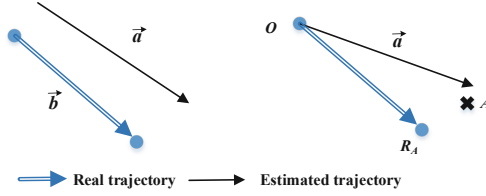


Fig. 3. Estimated trajectory and real trajectory.

We can define the distance probability as:

$$P_{dst}(\vec{b} / \vec{a}) = \frac{\min\{|\vec{a}|, |\vec{b}|\}}{\max\{|\vec{a}|, |\vec{b}|\}} \quad (3)$$

Similarly, the probability of orientation also has the following constraints: the smaller the angle between \vec{a} and \vec{b} is, the larger the probability is. And the probability decreases rapidly while the angle increases. Hence, $\cos(a, b)$ is adopted to indicate the probability of orientation. When angle difference is greater than 90° , it could lead to the negative value. So we set the probability value to 0 to avoid negative value in this situation. It is reasonable since this situation occurs very rarely. Therefore, the orientation probability can be calculated as:

$$P_{dir}(\vec{b} / \vec{a}) = \begin{cases} \cos(a, b), & \langle a, b \rangle \in [0, \frac{\pi}{2}] \\ 0, & \langle a, b \rangle \notin [0, \frac{\pi}{2}] \end{cases} \quad (4)$$

Since distance information is derived from accelerator and orientation information is stemmed from gyroscope, we can assume distance probability and orientation probability are independent. Thus, $P_{dsp}(\vec{b} / \vec{a})$ can be calculated as:

$$P_{dsp}(\vec{b} / \vec{a}) = \begin{cases} \frac{\min\{|\vec{a}|, |\vec{b}|\}}{\max\{|\vec{a}|, |\vec{b}|\}} \cos\langle \vec{a}, \vec{b} \rangle, & \langle a, b \rangle \in [0, \frac{\pi}{2}] \\ 0, & \langle a, b \rangle \notin [0, \frac{\pi}{2}] \end{cases} \quad (5)$$

As depicted in the right part of Fig. 3, when a user walks from a known location O to destination A , we can get an estimated trajectory \vec{a} by using the method proposed in previous section. Through RSSI matching, we also can get a candidate RP R_A and its

matching probability $P_{pf}(\overrightarrow{OR_A}) = P_{pf}(R_A/s_A)$ where s_A is the RSSI vector measured and collected at A. The integrated probability can be defined as:

$$P_{com}(\overrightarrow{OR_A}) = (1 - \alpha)P_{pf}(\overrightarrow{OR_A}) + \alpha P_{dsp}(\overrightarrow{OR_A}/\vec{d}) \quad (6)$$

where α represents the contribution of the estimated trajectory probability to the integrated probability. Its value is determined by the ratio of the accuracy of estimated trajectory versus the accuracy of RSSI matching. Since we use ε_{fp} and ε_{dis} to represent the error of RSSI matching and estimated trajectory respectively, α should satisfy:

$$\frac{\varepsilon_{fp}}{\varepsilon_{dis}} = \frac{\alpha}{1 - \alpha} \quad (7)$$

Integrating the Estimated Trajectory Containing Multiple Locations

The positioning accuracy can be improved by integrating the estimated trajectory from previous location to current location. However, if there is a large error in previous location, the improvement may be limited. We now utilize the historical path containing multiple locations to improve the positioning accuracy further.

In RSSI-based positioning, the k candidate RPs with the highest matching probability can be obtained from each measurement at the online phase, it may lead to multiple candidate paths as shown in Fig. 4. One candidate path can be represented by the points gathered from each measurement at online phase, $[p_1, p_2, \dots, p_n]$, where p_i is the candidate RP for the i^{th} measurement. We can express a path as a vector $path = [\overrightarrow{p_1 p_2}, \overrightarrow{p_2 p_3}, \dots, \overrightarrow{p_{n-1} p_n}]$, where $\overrightarrow{p_{i-1} p_i}$ represents the path segment corresponding to the estimated trajectory p_{i-1} to p_i .

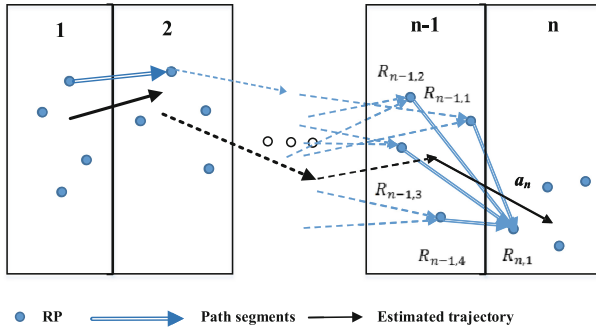


Fig. 4. An example of candidate paths

Considering the measurement is independent of each other, every segment of path can also be regarded as independent. Consequently, the probability of candidate path is

$$P_{com}(path) = P_{pf}(p_1/s_1) \prod_{i=2}^n P_{com}(\overrightarrow{p_{i-1} p_i}) \quad (8)$$

where n represents the total times of measurement, and represents the matching probability of RP p_I . By setting a virtual predecessor p_0 to the first RP p_I , the above equation can be generalized as:

$$P_{com}(path) = \prod_{i=1}^n P_{com}(\overrightarrow{p_{i-1}p_i}) \quad (9)$$

where $P_{com}(\overrightarrow{p_0p_1}) = P_{gf}(p_0/s_1)$.

After defining probability of candidate path, the next step is to compute the probability of candidate points. Intuitively, current candidate point can be reached through many paths from prior RP when user is walking in the indoor environment. Thus, the probability of current candidate point can be deduced as:

$$P(R) = \sum_{path \in PATH(R)} P_{com}(path) \quad (10)$$

where $PATH(R)$ is the set of paths ended with R . As depicted in Fig. 4, we take an assumption that $R_{n-1,j}$ and $R_{n,j}$ are the candidate points for the $(n-1)^{th}$ and n^{th} measurement respectively. The above equation can be simplified as:

$$\begin{aligned} P(R_{n,j}) &= \sum_{i=1}^k \sum_{path \in PATH(R_{n-1,i})} P_{com}(path) P_{com}(\overrightarrow{R_{n-1,i}R_{n,j}}) \\ &= \sum_{i=1}^k \left(P_{com}(\overrightarrow{R_{n-1,i}R_{n,j}}) \sum_{path \in PATH(R_{n-1,i})} P_{com}(path) \right) \\ &= \sum_{i=1}^k P(R_{n-1,i}) P_{com}(\overrightarrow{R_{n-1,i}R_{n,j}}) \end{aligned} \quad (11)$$

Since the RSSI matching probability and the path probability of the k RPs should be stored at each measurement, the space complexity of above algorithm is $O(kn)$. Moreover, the $P(R_{n-1,j})$ has been calculated in $(n-1)^{th}$ measurement, the time complexity of getting $P(R_{n,j})$ is reduced to $O(k)$. Considering that there are k candidate points, the time complexity of getting current location is $O(k^2)$.

4 Performance Evaluation

4.1 Setup

The experimental environment is illustrated in Fig. 5. There are 6 APs (indicated by yellow dots) deployed in the experiment area. The grid of RPs in the operation area includes 78 points with a spacing of 1.5 m. Wherein the blue spots represent the positions of the RPs, and the red spots represent the test positions during online period. During offline phase, we measured 110 RSS training samples at each RP to build the radio map.

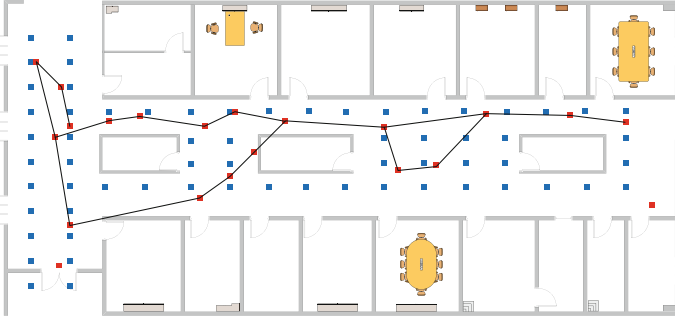


Fig. 5. Experimental environment.

During online phase, we collected 10 real-time RSS vectors at each test position with a sampling period of 250 ms.

Considering the user's trajectory, we generate 200 movement paths through an algorithm simulating the real human movement. We walk along the generated path 10 times to obtain the sensor data.

The positioning accuracy is calculated by the following equation:

$$E = \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (12)$$

where (x, y) represents the positioning coordinate, and (x_0, y_0) represents the real coordinate.

4.2 Results and Analysis

We first compare the positioning accuracy of our method with other methods like RADAR, Horus and PF + IKNN. As seen in Fig. 6, our method has the best performance. The mean localization errors of RADAR, Horus and PF + IKNN are 2.57 m, 1.18 m and 1.05 m respectively, our method reduces the mean localization error to 0.97 m. RADAR uses the KNN algorithm to determine candidate RPs, and simply take the average value of these RPs' coordinates as positioning results. That's why RADAR performs the worst. Horus and our method adopt statistical probability to build a radio map and look for the fittest fingerprint during the online phase. Compared with Horus, our method performs better because it reduces the contribution of RSSI-based results with high matching probability but large deviation to the final positioning results. The performance improvement is more significant when the location error is higher than 1.5 m.

PF + IKNN also has a close performance. Although the improved KNN reduces the positioning time, particle filter is computationally expensive, which is unsuitable for real-time positioning. Instead of using complex filtering mechanism, our method uses a lightweight probability mechanism to adjust the contribution of fingerprinting and IMU to the final results. The time complexity of particle filter is $O(n^2)$, where n indicates the number of particles and is set to 700 in the compared research. And the time complexity

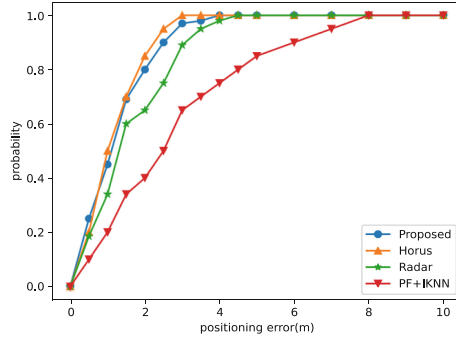


Fig. 6. Cumulative distribution function of positioning error.

of our proposed method is $O(k^2)$, where k indicates the number of candidate RPs, which is usually less than 10.

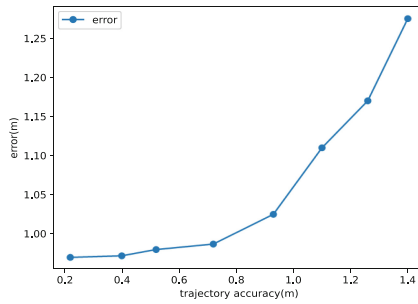


Fig. 7. Positioning error versus the accuracy of estimated trajectory.

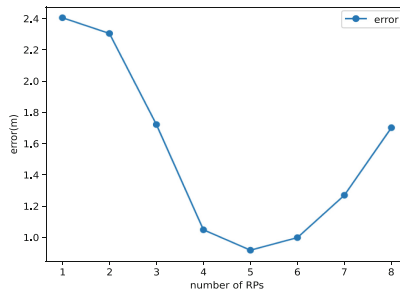


Fig. 8. Positioning error versus the number of RPs.

We also analyse the effects of the accuracy of estimated trajectory on final positioning results. We generate 8 estimated trajectories and then give them different accuracy. Figure 7 shows that as the accuracy of estimated trajectory improve, the final positioning result is more accurate. This can be easily explained according to Eq. 7 that the more accurate the estimated trajectory is, the higher the corresponding weight is.

Figure 8 shows how the number of RPs affects the positioning accuracy. The proposed method performs the worst when only one RP is matched per measurement. WiFi signal is vulnerable to the changeable environment, if only one RP is selected, it is quite possible to select a RP with significant deviation, which has bad influence on the final result. When the number of matched RPs increases, the most selected RPs with high matching probabilities have low deviation, which have a positive impact on accuracy. The accuracy decreases when the number of RPs exceeds the number of 6, which is similar with the case described in the literature [2]. That is because, with the number of RPs increasing, the RPs with the larger deviation may be selected and contribute to the calculation of final probability and furthermore, have a negative impact on final result. Therefore, the number of RPs should be determined empirically according to the actual environment.

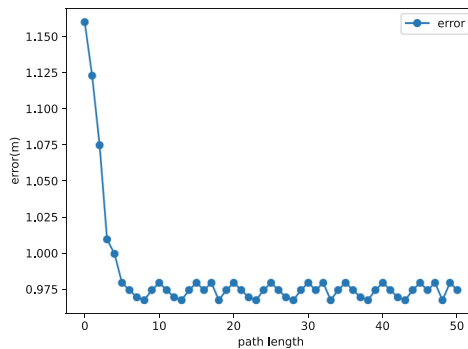


Fig. 9. Positioning error versus the path length

The relationship between the path length and the positioning accuracy is shown in Fig. 9. When we set the path length to 0, which means there is only one RP contained in the candidate path, the sensor data makes no contribution to the final probability, and the positioning result is only determined by the classical RSSI-based algorithm. So it is vulnerable to the labile environment and have the worst performance. With the increase of path length, the estimated trajectory makes contribution to the final probability. We also find the positioning accuracy stays around 0.97 m when the length of path exceeds 8. It means the integrated probability model can ease the adverse effects of changeable environment when the length of candidate path is long enough.

5 Conclusions and Future Directions

RSSI-based location is vulnerable to dynamic environment, which may introduce large deviation to final result. To deal with this issue, an integrated probability model combining the RSSI fingerprint and users' trajectory is proposed to improve online positioning accuracy. Experimental results show that the proposed method can greatly reduce the adverse effects of inconstant environment and meanwhile, it's computationally lightweight enough to be utilized in the smart phones.

The error of users' trajectory estimating and RSSI matching has great impact on the performance of our method because it decides the contribution of a certain candidate RP to the final positioning result. We will continue to develop more precise error model that can adapt to dynamic human motions in different environment. New deep learning methods will be considered. Now our method selects a fix number of candidate RPs. To improve the accuracy, we also will study on adaptive candidate RPs selection.

References

1. Guo, X., et al.: A survey on fusion-based indoor positioning. *IEEE Commun. Surv. Tutorials* **22**(1), 566–594 (2020)
2. Bahl, P., Padmanabhan, V.N.: RADAR: an in-building RF-based user location and tracking system. In: *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, pp. 775–784. IEEE, Tel Aviv (2000)
3. Youssef, M., Agrawala, A.: The horus location determination system. *Wirel. Netw.* **14**(3), 357–374 (2008)
4. Shi, K. et al.: Support vector regression based indoor location in IEEE 802.11 environments. *Mob. Inf. Syst.* **2015**, 1–14 (2015)
5. Chen, Z., et al.: Fusion of WiFi, smartphone sensors and landmarks using the kalman filter for indoor localization. *Sensors.* **15**(1), 715–732 (2015)
6. Sansano-Sansano, E., et al.: Multimodal sensor data integration for indoor positioning in ambient-assisted living environments. *Mob. Inf. Syst.* **2020**, 1–16 (2020)
7. Li, N., et al.: A fast indoor tracking algorithm based on particle filter and improved fingerprinting. In: *2016 35th Chinese Control Conference (CCC)*, pp. 5468–5472. IEEE, Chengdu (2016)
8. Qian, Y., Chen, X.: An improved particle filter based indoor tracking system via joint Wi-Fi/PDR localization. *Meas. Sci. Technol.* **32**(1), 014004 (2021)
9. Liu, W., et al.: Survey on CSI-based indoor positioning systems and recent advances. In: *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8. IEEE, Pisa (2019)
10. Dai, P., et al.: Combination of DNN and improved KNN for indoor location fingerprinting. *Wirel. Commun. Mob. Comput.* **2019**, 1–9 (2019)
11. Wang, X., et al.: Deep convolutional neural networks for indoor localization with CSI images. *IEEE Trans. Netw. Sci. Eng.* **7**(1), 316–327 (2020)
12. Nikitin, A., et al.: Indoor localization accuracy estimation from fingerprint data. In: *2017 18th IEEE International Conference on Mobile Data Management (MDM)*, pp. 196–205. IEEE, Daejeon (2017)
13. Kang, W., Han, Y.: SmartPDR: smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sens. J.* **15**(5), 2906–2916 (2015)