Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 372425, 8 pages
http://dx.doi.org/10.1155/2015/372425

Research Article

TraIL: Pinpoint Trajectory for Indoor Localization

Shengnan Li, Zheng Qin, Chenshu Wu, and Zheng Yang

School of Software and TNList, Tsinghua University, Beijing 100084, China

Correspondence should be addressed to Shengnan Li; xiaoling4329@gmail.com

Received 19 January 2015; Accepted 29 March 2015

Academic Editor: Chen Qian

Indoor localization on smartphones is an enabler for a number of ubiquitous and mobile computing applications attracting worldwide attentions. Many location-based services rely on WiFi fingerprinting approaches to achieve a reasonable accuracy. However, there is still room for improvement due to the prevalent-existing errors (e.g., 8~12 m). In this study, we devise and implement a high-accuracy indoor localization solution leveraging the WiFi-based method and pedestrian mobility provided by smartphones. Our basic idea is that WiFi-only localization can generate rough but absolute positions, while user motion is able to bring accurate but relative locations. Taking both sides into account simultaneously, we design techniques to refine the raw WiFi positions in the process of laying the precise local trajectory appropriately down to the absolute coordinate using a novel least median of squares (LMS) fit algorithm. We develop a prototype system, named TraIL, and conduct comprehensive experiments in a building along different shaped routes. The evaluation results show that TraIL can achieve 80% improvement on average error with respect to WiFi-only indoor localization.

1. Introduction

In many mobile computing applications, location-based service (LBS) has become an indispensable element. Indoor localization on a smartphone has prospered in the past two decades in both academia and industry. However, current smartphone-based indoor positioning system is not satisfactory enough to locate or navigate a user [1–3], requiring significant accuracy improvement.

Different from GPS as a dominating solution for outdoor localization, there is a variety of techniques when the issue comes to the indoor environment. WiFi-based RSSI fingerprinting schemes can achieve a reasonable accuracy (e.g., 3~4 m), but large errors (e.g. 8~12 m) still exist [4]. A plethora of academic works seek for assistance from other means, such as precise ranging [5, 6] and peer devices [7] to improve accuracy. However, these extra requirements may result in the loss of ubiquity held by original WiFi localization. Emerging research on leveraging user motion attracts growing attention [7–12]. These works all leverage user motion and WiFi-based technology; however, one means is usually treated as an auxiliary to another. It still remains a research challenge to take advantages of both sides' strengths.

In this paper, we propose TraIL, a trajectory-oriented indoor localization system implemented on off-the-shelf smartphones. We reveal the rationale behind TraIL from two basic observations: WiFi-based methods can provide coarse-grained but absolute positions, while pedestrian mobility is able to offer accurate but relative locations. TraIL adopts an RSSI fingerprinting-based localization method for the absolute coordinate generation; in relative coordinate the user's trajectory can be treated as a rigid graph holding the shape feature of the route. Having both relative and absolute coordinates, we devise a coordinate transformation scheme converting the former to the latter. Speaking of the coordinate transformation, we draw lessons from LMS fit [13] and exert its strong merits of maintaining the rigid core and immunizing against outliers to the utmost. To sum up, the raw WiFi positions are refined, in the process of laying the local trajectory faithfully down to the absolute coordinate, as shown in Figure 1.

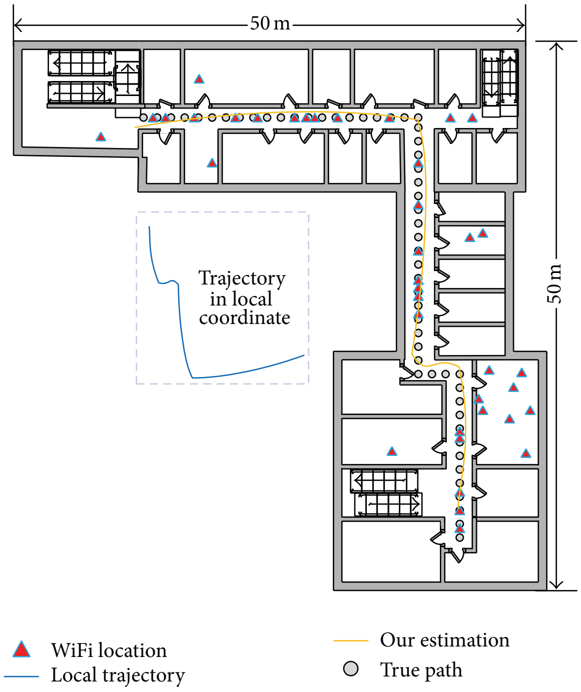

Figure 1: An illustration with both absolute and relative trajectories.

To evaluate our design, we implement a prototype system on Android OS using HTC One phones and execute a series of experiments. The results demonstrate that the maximum and mean errors are reduced to 4 m and 2 m, respectively. Around 80% enhancement in indoor localization proves the feasibility of TraIL in real world deployment [14, 15].

We summarize our contribution as follows.

(i) We identify both WiFi-based scheme and pedestrian mobility of equal importance for indoor localization. TraIL can work on a single smartphone without the assistance of other peers, or additional constraints, thus forming a self-contained system.

(ii) We develop a novel approach to generate the local trajectory. TraIL delivers a motion state and an acceleration measurement to the Kalman filtering process, thus achieving a precise velocity estimation.

(iii) We introduce the LMS fit algorithm to coordinate transformation. It can cope with the unexpected outliers appearing in WiFi indoor localization due to the merit in preserving more core regions.

The rest of this paper is organized as follows. Section 2 reviews the related literature. We introduce the system design of TraIL in Section 3. In Section 4, we propose a novel user motion estimation scheme for local trajectory generation and employ a WiFi-based localization technique for absolute coordinate generation. Coordinate transformation algorithm is detailed in Section 5. We discuss the system performance in Section 6 and conclude this work in Section 7.

2. Related Work

Indoor localization is studied extensively in pervasive and mobile computing literature. We review the state-of-the-art works most relevant to ours.

2.1. WiFi-Based Localization. The main stream of indoor localization approaches leverages RSSI fingerprinting, such as [16, 17], HORUS [4] and Landmarc [18]. However, this technique takes effect on condition that fingerprints differ from each other among those reference locations, which cannot always be held ideally.

2.2. User Motion-Assisted Localization. Off-the-shelf smartphones, mounted on the stage as a popular mean in localization approaches, provide a plethora of user motion information. There are considerable literature in this area lately [8–12, 19].

Under this context, the most relevant works to ours are [8] and GloCal [19]. In [8], the approach is to determine the absolute location of a pedestrian given a path of step events describing their relative movement and a map of environment. However, foot-mounted IMUs are less pervasively available than smartphones, leaving system deployment difficult.

2.3. Coordinate Transformation. Coordinate transformation using least squares (LS) methods is well known in computer graphics. Particularly, Horn's method is selected into the previous work GloCal using a set of translation, scaling, and rotation operations. For further improvement, least median squares (LMS) fit is taken as an attractive perspective to solve various superposition problems [13], as well as coordinate transformation. We absorb the merits of these two means to solve the transformation problem to our satisfaction. To deal with more outliers in WiFi-based methods, our approach leverages the LMS fit-based coordinate transformation explained later, to make the system insensitive to outliers.

3. System Overview

In this section, we present the system architecture of TraIL, with the problem statement of indoor localization assisted by user motion.

As shown in Figure 2, the working process of TraIL consists of two main parts: coordinate generation and coordinate transformation. The WiFi-based method collects absolute locations and sensor readings forge a relative trajectory representing the user motion information.

After raw inertial sensor data are transferred into relative coordinate generation operator, forward speed and rotational angular are first estimated by accelerometer and gyroscope, respectively, following the algorithms explained in Section 4. Accordingly, displacement and direction become attainable naturally, delineating our local trajectory together. On the other side, when treating rough locations from WiFi as absolute positions, the TraIL fixes the user traces on the floor plan determinately.

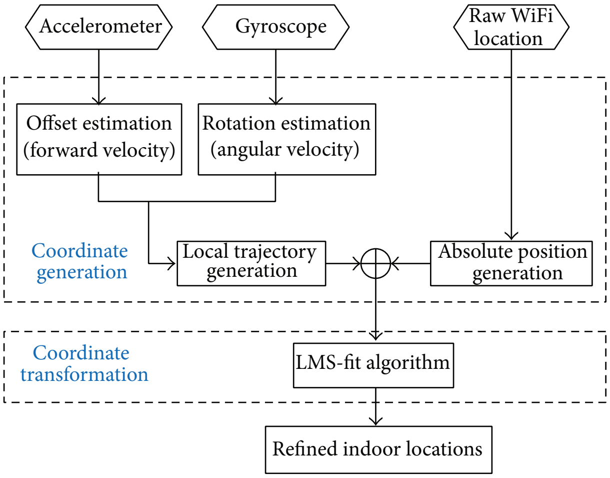
Accelerometer
Gyroscope
Raw WiFi location
Offset estimation (forward velocity)
Rotation estimation (angular velocity)
Coordinate generation
Local trajectory generation
Absolute position generation
Coordinate transformation
LMS-fit algorithm
Refined indoor locations

Figure 2: System architecture of TraIL.

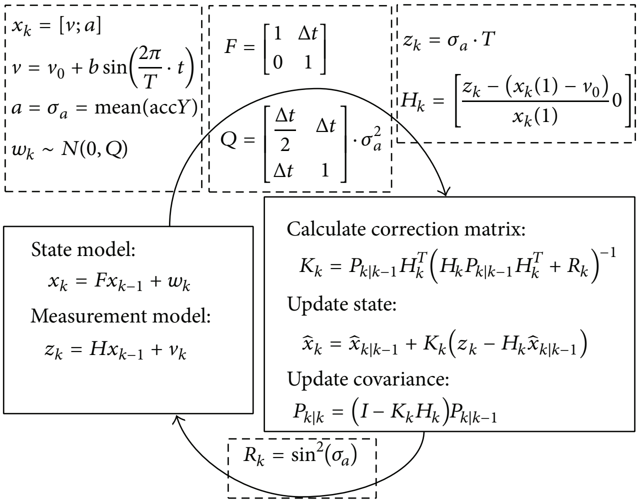
$x_k = [v; a]$
$v = v_0 + b\sin\left(\frac{2\pi}{T}\cdot t\right)$
$a = \sigma_a = \mathrm{mean}(\mathrm{acc}Y)$
$w_k \sim N(0, Q)$
$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$
$Q = \begin{bmatrix} \frac{\Delta t}{2} & \Delta t \\ \Delta t & 1 \end{bmatrix}\cdot\sigma_a^2$
$z_k = \sigma_a \cdot T$
$H_k = \left[\frac{z_k - (x_k(1) - v_0)}{x_k(1)} 0\right]$
State model:
$x_k = Fx_{k-1} + w_k$
Measurement model:
$z_k = Hx_{k-1} + v_k$
Calculate correction matrix:
$K_k = P_{k|k-1}H_k^T\left(H_k P_{k|k-1} H_k^T + R_k\right)^{-1}$
Update state:
$\hat{x}_k = \hat{x}_{k|k-1} + K_k\left(z_k - H_k\hat{x}_{k|k-1}\right)$
Update covariance:
$P_{k|k} = \left(I - K_k H_k\right)P_{k|k-1}$
$R_k = \sin^2(\sigma_a)$

Figure 3: Kalman filter process and our parameters.

Learned from Figure 1, the relative locations from dead-reckoning preserve a better structure of the actual trajectory than WiFi-based trace. Thus, our aim is to assign the absolute positions to the local trajectory; in other words, place the relative curve properly on the absolute coordinate. We adopt a superposition algorithm to achieve a more precise fitting by improving conventional LS means with robust LMS. Finally, all absolute locations along the user trace are refined with the transformed relative locations.

4. Trajectory Generation

We introduce two components of trajectory generation which are local and absolute ones in this section. The local trajectory is curved by the smartphone-driven dead-reckoning method and the absolute one is depicted by the WiFi-based indoor localization technique.

4.1. User Motion Estimation. Offset and turning are two essential parts to local trajectory generation. They are estimated using only smartphone built-in accelerometer and gyroscope sensors. Offset and turning are characterized by displacement and rotational angle, respectively. Our algorithm regards local user motion and absolute WiFi localization of equal importance.

4.1.1. Displacement Measurement. Ideally, displacement can be obtained by integrating forward acceleration twice directly. However, the accumulative error makes this method unusable for researchers. Heuristically, we treat this as a velocity estimation problem. Since we can easily get the noisy acceleration of y-axis (forward direction, denoted by acc$Y$), we can then leverage it to estimate walking velocity using Kalman filter. The acceleration of z-axis (acc$Z$) is also adopted for displacement measurement, not merely for step counting as usual.

If we admit $v = at$ as the description of the user's walking pattern, Kalman filter can work well as the system is linear [20]. After reviewing the Kalman filter and the user motion, we can formulate the velocity estimation problem in Figure 3.

To make it simple, we set $v_0 = 1.2$, and $b = 0.3$ at first according to the mean values in [21]. Thus far, the estimated velocity is calculated through the above operations, which borrow from Kalman filter and are adjusted to our specific scenario.

Notice that, in the aforementioned formula, v_0 and Δt are set fixed, which cannot adapt to various situations. As a result, we improve our algorithm mainly in two ways: Δt estimation using step counting technique and v_0 revision by exploring walking patterns of different users. Besides, we recalculate the σ_a according to the new Δt to improve precision.

To obtain an accurate Δt, we may leverage acc$Z$ to count steps first. We can learn from Figure 4 that acc$Z$ data after wavelet filtering present quite stable amplitude of vibration but may suffer from multiple lowest points when calculating steps. Thus, two thresholds would be adopted to control the counting process, for example, one corresponding to the amplitude *thresAMP* and the other dealing with multiple lowest points *thresLP*. These two thresholds can be estimated from a learning based method. As can be found in Figure 4, acceleration amplitude of different groups clustered by various user walking patterns ranges from ±2 to ±4, according to which we set the *thresAMP* to be −2 as the guideline of the trough selection. Although wavelet filter offers us a relatively stable amplitude convenient for determining the *thresAMP*, the multiple lowest points create unnecessary obstacles when picking appropriate minimal value. We take advantage of sampling frequency to determine how many points, say, *thresLP*, should be skipped. As the average accelerometer frequency of off-the-shelf smartphones is 50 Hz, we come to 20 as *thresLP* after learning. From this point on, the whole algorithm can yield accurate counting of steps, which enables a more precise estimate of Δt by calculating the time elapsed

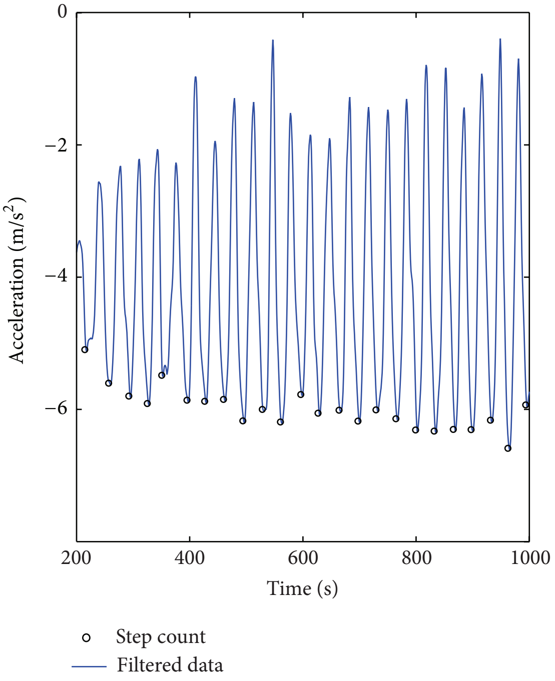

Figure 4: Step counting after wavelet filtering.

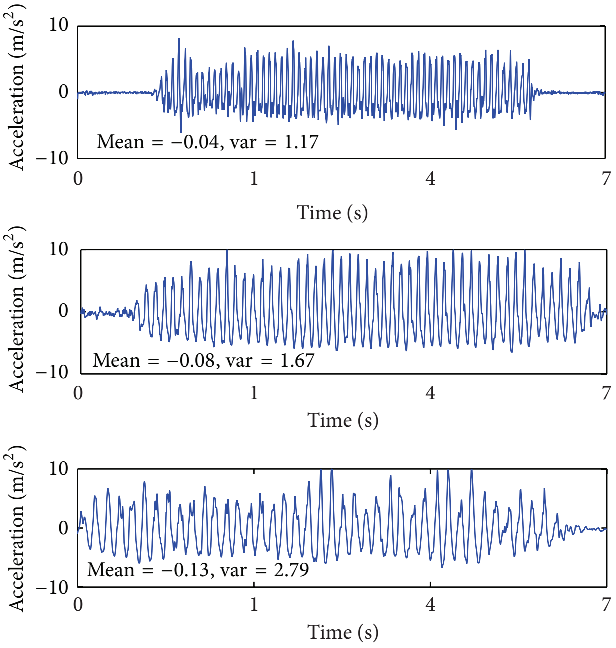

Figure 5: Walking patterns of different users exhibit distinct characteristics.

between troughs. Now, we have the exact Δt between steps, which can be used to correct σ_a in previous formula. To make it simple, σ_a is equal to the mean value of the accelerations among steps, instead of one of merely acc$Y$.

As mentioned above, different users exhibit different walking patterns, for example, as shown in Figure 5, mean and variance of acc$Z$ change with users' velocity (characterized by acc$Y$). Therefore, we offer a solution to update v_0 and b mapping to the z-axis variance and y-axis acceleration. Let $v_0 = \sigma_a \cdot \Delta t$ first; then make $b = v_0/\text{variance}$. Note that v_0 and b are put into further iteration; there is no need to calculate the exact value of them, since only a reasonable trend would be enough.

After the foregoing process, the previous algorithm with new Δt, σ_a, v_0, and b can offer a more accurate estimation of walking velocity. On this basis, the displacement simply equals the product of velocity and time. The main part of the displacement measurement is done here. In retrospect, however, a bunch of previous works [9, 10, 12, 19, 22] estimate the displacement using the product of stride length and steps. Our method, directly calculating the walking velocity, may further compute users' stride length as a bonus.

The cumulative errors in dead reckoning using the accelerometer are used to be avoided in the process of velocity estimation, but we heuristically leverage them to play a role in updating the covariance matrix as residual in Kalman filter process.

4.1.2. Rotational Angle Measurement. As direction estimation is not our main contribution, we generate the rotational angle integrating the angular velocity $\Delta\gamma$ multiplied by elapsed time between steps Δt, as $\theta = \Delta\gamma \cdot \Delta t$. Thanks to our accurate estimation of Δt, there is a doddle to obtain the interval of a step. Also, we benefit from the high sampling frequency of gyroscope, from which we can easily correspond the time of proper angular velocity to that of relevant acceleration.

4.2. Relative Coordinate Generation. Both offset and direction suffice for sketching moving trajectory. Let $T = \{t_i, i = 1, \ldots, S-1\}$ be the set of trajectory points, where S is the number of steps. Assuming that the coordinates of t_i is (x_i, y_i) and t_1 $(0,0)$, we would like to regard each step as a point which is represented by t_i. With the offset o_i and the rotational angle θ_i of each step, we can calculate the coordinate of every point on the trajectory shown in Figure 6, following the formula:

$$\begin{aligned} x_{i+1} &= x_i + o_i \cos\left(\alpha + \theta_{i+1}\right) \\ y_{i+1} &= y_i + o_i \sin\left(\alpha + \theta_{i+1}\right) \end{aligned} \qquad (1)$$

in which $\alpha = \sum_{j=1}^{i-1} \theta_j$ and θ_i is positive when turning counterclockwise and vice versa. Then, the local trajectory is portrayed in a Cartesian coordinate.

4.3. Absolute Coordinate Generation. Absolute coordinate consists of the spots we locate using the WiFi fingerprinting technique during the interval. Since RSSI fingerprints acquisition and matching are not our emphasis, we refer to the approach in [11] and adjust it to ours by finding the least distance between the experimental vector and a vector from the WiFi fingerprint database.

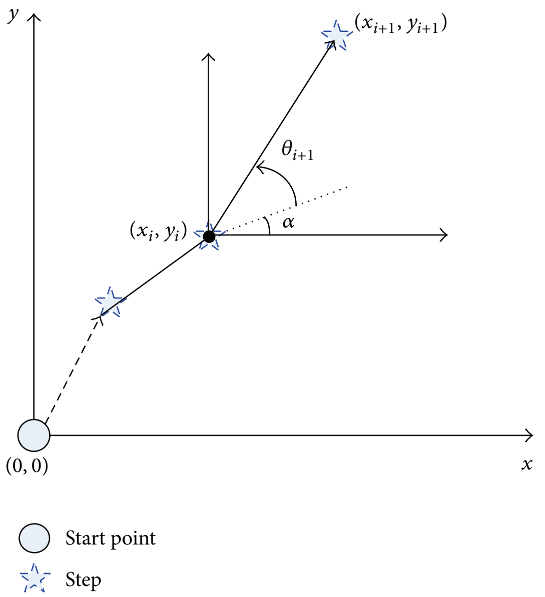

Figure 6: Relative coordinate generation using offset and turning.

We are proceeding on the supposition that there is already a fingerprint database, by whatever means, crowdsourcing, or site survey. During the time when user moves, we record the sensor data and take RSS samples of each obtainable AP using smartphones. Let the offline and online fingerprint sets be $F = (f_1, f_2, \dots, f_n)$ and $F' = (f_1', f_2', \dots, f_n')$, respectively. As a user moves, the location rapidly changes accordingly. This brings about a problem that the smartphone a user holds may not obtain the whole RSS set of all APs it should have scanned. So much so that we can have only a few f_i' values assigned in F' compared with that in F during one or two scanning process at a spot. As generally admitted, the RSS value under −90 dB is too fluctuant to be reliable. Besides, the higher the RSS is, the more featured information it takes. We assign another constraint to the f_i'; that is to say, in our formula, high RSS value weighs much. We leverage the modified Euclidean distance to calculate the dissimilarity between F and F'

$$D = \frac{\sum w_i \left(f_i - f_i'\right)^2}{\sum w_i} I \quad \left(f_i' > -90\right), \tag{2}$$

where $w_i = (RSS_i' + 100)/100$.

5. Coordinate Transformation

In this section, our purpose is to rectify the absolute location generated by WiFi-based technique by means of coordinate transformation with the help of local trajectory curved by dead reckoning. We describe the problem as a process that places the trajectory in local coordinate into the absolute one perfectly fitting the origin route shown by WiFi located spots. We adopt a robust fitting method for this problem, called least median of squares (LMS) fit algorithm, which is helpful dealing with outliers.

5.1. Problem Analysis. We can see in our case a curve fitting problem instead of coordinate transformation. Suppose that $\mathbb{L} = (m_1, m_2, \dots, m_N)$ and $\mathbb{A} = (u_1, u_2, \dots, u_N)$ are two sets of local and absolute spots, respectively, where N is the number of points. Our aim is to find an optimal transformation $\mathbf{V}$ composed of translation, scaling, and rotation, which minimizes the median distance if the residuals defined as follows:

$$\min_{\mathbf{V}} \operatorname*{median}_{i} \left\| \mathbf{V} m_i - u_i \right\|, \quad 1 \le i \le N. \tag{3}$$

5.2. Method Details. In this subsection, we explain the solution in detail which consists of LS fit, random sampling algorithm (RANSAC), and forward search.

5.2.1. Least Squares Fit Algorithm. LS fit is an ordinary solution to our problem at the beginning, and it is able to deliver a reasonable report to our problem. LS fit is classical for its robustness to noise, but unfortunately, not immune to outliers which are quite often in this specific scenario where indoor absolute coordinate is generated by untrustworthy WiFi-based technique.

We adopt Horn's method [19] to solve LS fit and get the rotation matrix $\mathbf{V}$ for each step. Our problem here is to find the transformation

$$\mathbf{m}^a = sR\left(\mathbf{m}^l\right) + t, \tag{4}$$

where $\mathbf{m}^l \in \mathbb{L}$, $\mathbf{m}^a$ is the transformed point in absolute coordinate system, s is a scale factor, t is the translational offset, $R(\mathbf{m}^l)$ is the rotated vector, and $\mathbf{m}_i = m_{x,i} + i m_{y,i}$, $\mathbf{u}_i = u_{x,i} + i u_{y,i}$ are the complex number form of each point. Our aim in this paragraph is to minimize the sum of squares of the residual errors

$$\sum_{i=1}^{N} \left\| e_i \right\|^2 = \sum_{i=1}^{N} \left\| u_i^a - m_i^a \right\|^2, \tag{5}$$

where $u_i^a \in \mathbb{A}$ and e_i is the difference between u_i^a and m_i^a.

We first deal with the translation, then scaling, and rotation at last, with respect to the sum of residual errors for each process following the method in [19].

5.2.2. Random Sampling Algorithm. RANSAC is an effective way to solve (3). The first step is to select k point pairs randomly in both point sets, computing the initial transformation $\mathbf{V}$ using LS algorithm to the k pairs. Next, we can calculate the median of the residuals of the remaining $N - k$ point pairs. Then the above steps are repeated M times to generate M transformation candidates, in which we specify the transformation $\mathbf{V}$ with minimal median value as the final transformation $\mathbf{V}$.

5.2.3. Forward Search Algorithm. Noting that the RANSAC is not that satisfactory due to its usage of fixed k and M, we adopt the forward search algorithm to avoid fixed variables. The key point of this algorithm is to search a small subset using LMS first and iteratively update it by adding an eligible point at a time. It is in contrast to the backward algorithms, dealing with the whole set and then deleting ill points. Forward search algorithm is more efficient than the above for our real-time demand.

5.3. LMS Fit Algorithm. Now, putting all these means together, we explain our LMS fit algorithm step by step. For the point sets $\mathbb{L}$ and $\mathbb{A}$, we have the following.

(1) Generate the small subset S_l and S_a, using RANSAC.

(2) Compute the transformation $\mathbf{V}$ for S_l and S_a, using Horn's method.

(3) Add the pair of points with the minimal residual in the remaining sets into S_l and S_a, respectively.

(4) Repeat (2) and (3) until the minimal residual is larger than a threshold and the iterative times are larger than another threshold. At last, we have the optimal transformation $\mathbf{V}$, which correspond to the desired minimal residual.

With the help of the robustness of LMS fit in preserving the rigid core [13], we can have the transformation process less sensitive to outliers, thus achieving a better localization result.

6. Evaluation

We evaluate the proposed approach above, by implementing a prototype system of TraIL on the Android platform. We first explain the experiment settings and methodology, and then we present the performance of the integrated system after the detailed evaluation of each parts.

6.1. Experiment Environment and Methodology. We implemented the TraIL on Android OS on HTC One phones, supporting acceleration, angular velocity, and RSS data collection. The accelerometer and gyroscope are of 50 Hz and 800 Hz frequency, respectively, and the WiFi signals are scanned twice per second. In our TraIL prototype, the sampling rate is set to follow the WiFi signal receiving module for the consideration of energy cost and computation complexity.

We evaluated the performance of TraIL in the East Main Building at Tsinghua University as shown in Figure 1. The experiment area is 50 m × 50 m in size, including hallways, classrooms, stairs, and laboratories. All areas are covered by deployed APs. When users are walking and using their smartphones for localization, the data are meanwhile collected automatically, including RSS and sensor readings. Accordingly, we form absolute and local trajectories for further use.

We collected data from 3 users with diverse heights, different walking paces, and stride lengths. In the mean time, our experiment is tested on 21 traces. In realistic scenarios, users can walk naturally and casually instead of addicting to the predicted routes.

6.2. Performance Evaluation. We first validate the walking speed estimation, along with step counting, velocity, and stride length accuracy. Then we come to the trajectory generation analysis, including the local one and the absolute one. Finally, we can evaluate the localizing accuracy by examining the coordinate transforming method.

6.2.1. Walking Speed Estimation Accuracy. To validate the estimation algorithm, we analyze 21 traces with various velocities. One user's actual walking speed is measured as the quotients of the path length to the time duration. The accuracy of velocity estimation is enhanced by 38% in average under different circumstances. There follows the velocity error of 21 traces in Figure 7(a), where we can get a mean error of 0.21 m/s.

6.2.2. Step Counting and Stride Length Estimation Accuracy. In the process, we calculate the user's stride length using the equation Stride Length = Velocity × Duration Time/Step Count. The mean stride length error is less than 7 cm and the maximum error is around 10 cm shown in Figure 7(b). It should be pointed out that our system has the ability to achieve a relative high accuracy of stride length by means of estimating the walking speed. In return, the accurate stride length proves the speed estimation effective.

6.2.3. Localization Accuracy. Now we turn to the localization accuracy of the raw WiFi method and the coordinate transformation algorithm. The positioning errors are analyzed in Figure 7(c). As shown in the graph, Horn's method reduces the average and maximum errors by around 6 m and 10 m, respectively, while LMS reduces the average and maximum errors by around by 8 m and 12 m. That is to say, by appropriately placing local trajectories on absolute coordinate, we can achieve an impressive improvement of 80.8%.

TraIL can achieve a submeter accuracy with smartphones. In Figure 8, we can see that the TraIL method works better for indoor localization compared to other four systems, Horus [4], Zee [23], UnLoc [7, 22]. Besides, [8, 12] have competitive reported accuracy of 0.73 m and 0.67 m. In [8], Woodmans work leveraged an extra device, foot-mounted inertial unit, to enhance the performance, while ours do not need additional equipment. As for [12], the authors focus on the fingerprint ambiguity, while we emphasize maintaining the shape of the trace to achieve a better accuracy. Also, [12] and ours both mention the stride length estimation. At this point, their offset measurements have a median and maximum error of 0.13 m and 0.46 m, respectively, while ours are 0.068 m and 0.17 m.

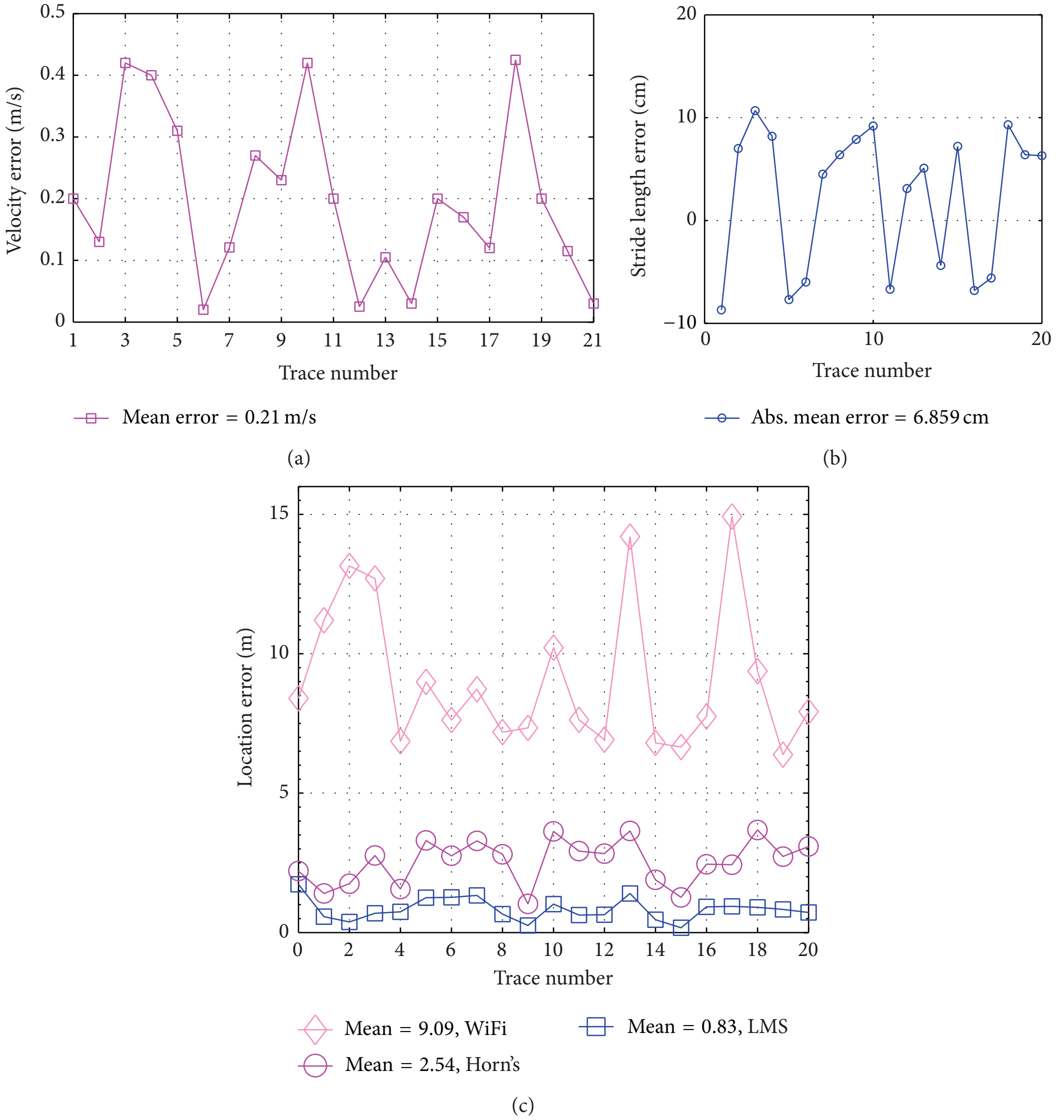

Figure 7: Errors in different traces: (a) the velocity error; (b) the stride length error; (c) the localization error.

Methods	Accuracy
Horus	2.7 m
Zee	1.8 m
Peer-assisted	1.3 m
UnLoc	2.1 m

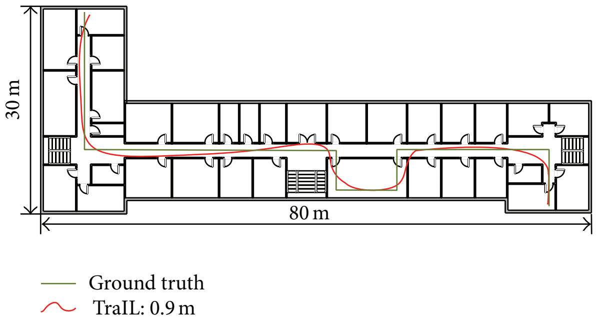

Figure 8: Experiment areas in the 10th district of the East Main Building.

7. Conclusion

In this paper, we have developed a novel architecture to provide a seamless indoor localization leveraging trajectories from RSSI fingerprinting and pedestrian mobility for coordinate transformation. On this basis, the WiFi-based positions are refined by the well-shaped local trajectory through the LMS fit. To explore the feasibility of our approach, we implemented a prototype of TraIL in a building. The experiment results suggest that TraIL can achieve an 80% accuracy improvement on average, manifesting its promising utilization in reality.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported in part by NSFC under Grants 61171067, 61272429, 61303209, and 20120002130007 and Beijing Nova Program.

References

[1] Y. Liu, Z. Yang, X. Wang, and L. Jian, "Location, localization, and localizability," *Journal of Computer Science and Technology*, vol. 25, no. 2, pp. 274–297, 2010.

[2] J. Han, C. Qian, P. Yang et al., "Geneprint: generic and accurate physical-layer identification for UHF RFID tags," *IEEE/ACM Transactions on Networking*. In press.

[3] K. Sheng, Z. Gu, X. Mao et al., "The collocation of measurement points in large open indoor environment," in *Proceedings of the IEEE INFOCOM*, Hong Kong, April 2015.

[4] M. Youssef and A. Agrawala, "The horus WLAN location determination system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, pp. 205–218, ACM, June 2005.

[5] K. Liu, X. Liu, and X. Li, "Guoguo: enabling fine-grained indoor localization via smartphone," in *Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*, pp. 235–248, June 2013.

[6] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: indoor localization via channel response," *ACM Computing Surveys*, vol. 46, no. 2, article 25, 2013.

[7] H. Liu, Y. Gan, J. Yang et al., "Push the limit of WiFi based localization for smartphones," in *Proceedings of the 18th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '12)*, pp. 305–316, August 2012.

[8] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp '08)*, pp. 114–123, 2008.

[9] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: wireless indoor localization without site survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2013.

[10] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom '12)*, pp. 269–280, August 2012.

[11] Y. Jiang, Y. Xiang, X. Pan et al., "Hallway based automatic indoor floorplan construction using room fingerprints," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*, pp. 315–324, 2013.

[12] W. Sun, J. Liu, C. Wu, Z. Yang, X. Zhang, and Y. Liu, "MoLoc: on distinguishing fingerprint twins," in *Proceedings of the IEEE 33rd International Conference on Distributed Computing Systems (ICDCS '13)*, pp. 226–235, July 2013.

[13] Y.-S. Liu, Y. Fang, and K. Ramani, "Using least median of squares for structural superposition of flexible proteins," *BMC Bioinformatics*, vol. 10, article 29, 2009.

[14] Y. Jiang, H. Zhang, X. Song et al., "Bayesian-network-based reliability analysis of PLC systems," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, pp. 5325–5336, 2013.

[15] Y. Jiang, Z. Li, H. Zhang et al., "Design and optimization of multi-clocked embedded systems using formal technique," in *Proceedings of the 9th Joint Meeting on Foundations of Software Engineering (ESEC/FSE '13)*, pp. 703–706, ACM, St. Petersburg, Russia, August 2013.

[16] Z. Zhou, C. Wu, Z. Yang, and Y. Liu, "Sensorless sensing with wifi," *Tsinghua Science and Technology*, vol. 20, no. 1, pp. 1–6, 2015.

[17] Y. Wen, X. Tian, X. Wang, and S. Lu, "Fundamental limits of RSS fingerprinting based indoor localization," in *Proceedings of the of IEEE Conference on Computer Communications (INFOCOM '15)*, Hong Kong, China, April 2015, http://iwct.sjtu.edu.cn/Personal/xwang8/paper/INFOCOM2015_indoorlocalization.pdf.

[18] M. N. Lionel, Y. Liu, Y. C. Lau, and A. P. Patil, "Landmarc: indoor location sensing using active rfid," *Wireless Networks*, vol. 10, no. 6, pp. 701–710, 2004.

[19] C. Wu, Z. Yang, Y. Xu, Y. Zhao, and Y. Liu, "Human mobility enhances global positioning accuracy for mobile phone localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 131–141, 2015.

[20] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[21] B. W. Stansfield, S. J. Hillman, M. E. Hazlewood, and J. E. Robb, "Regression analysis of gait parameters with speed in normal children walking at self-selected speeds," *Gait & Posture*, vol. 23, no. 3, pp. 288–294, 2006.

[22] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: unsupervised indoor localization," in *Proceedings of the 10th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*, pp. 197–210, June 2012.

[23] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom '12)*, pp. 293–304, ACM, August 2012.