

Article

A Robust Crowdsourcing-Based Indoor Localization System

Baoding Zhou ^{1,2}, Qingquan Li ^{1,2,*}, Qingzhou Mao ³ and Wei Tu ^{1,2,*}

¹ Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen 518060, China; bdzhou@szu.edu.cn

² Key Laboratory for Geo-Environment Monitoring of Coastal Zone of the National Administration of Surveying, Mapping and Geoinformation, Shenzhen University, Shenzhen 518060, China

³ State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; qzhmao@whu.edu.cn

* Correspondence: liqq@szu.edu.cn (Q.L.); tuwei@szu.edu.cn (W.T.); Tel.: +86-755-2697-9741 (Q.L.)

Academic Editor: Antonio R. Jiménez

Received: 9 January 2017; Accepted: 11 April 2017; Published: 14 April 2017

Abstract: WiFi fingerprinting-based indoor localization has been widely used due to its simplicity and can be implemented on the smartphones. The major drawback of WiFi fingerprinting is that the radio map construction is very labor-intensive and time-consuming. Another drawback of WiFi fingerprinting is the Received Signal Strength (RSS) variance problem, caused by environmental changes and device diversity. RSS variance severely degrades the localization accuracy. In this paper, we propose a robust crowdsourcing-based indoor localization system (RCILS). RCILS can automatically construct the radio map using crowdsourcing data collected by smartphones. RCILS abstracts the indoor map as the semantics graph in which the edges are the possible user paths and the vertexes are the location where users may take special activities. RCILS extracts the activity sequence contained in the trajectories by activity detection and pedestrian dead-reckoning. Based on the semantics graph and activity sequence, crowdsourcing trajectories can be located and a radio map is constructed based on the localization results. For the RSS variance problem, RCILS uses the trajectory fingerprint model for indoor localization. During online localization, RCILS obtains an RSS sequence and realizes localization by matching the RSS sequence with the radio map. To evaluate RCILS, we apply RCILS in an office building. Experiment results demonstrate the efficiency and robustness of RCILS.

Keywords: indoor localization; crowdsourcing; radio map; smartphone

1. Introduction

Indoor localization has attracted much interest in recent years due to the diverse location-based services (LBS) that require accurate positioning [1]. There are several technologies available to provide indoor positioning solutions such as WiFi [2], radio-frequency identification (RFID) [3], Bluetooth [4], Ultrawide Band (UWB) [5], inertial sensors-based localization [6,7], etc. In particular, WiFi fingerprinting has been widely used due to its simplicity leveraging on the pre-existing WiFi infrastructures. Moreover, this approach does not require any specialized hardware or additional infrastructure support because most smartphones are WiFi-enabled.

A WiFi fingerprinting-based positioning system consists of two phases: offline training phases and online positioning phases. In the training phase, a set of known locations are selected as the reference points (RPs) and WiFi Received Signal Strengths (RSSs) from all detected access points (APs) are collected at each RP. The RSSs collected at each RP are called fingerprints. To improve the localization performance, this collection takes a few seconds in every point to collect a sufficient

number of measurements in order to overcome the RSS variance problem. In some other cases, the collection takes even more time when it is done in four different orientations to take into account the effect of antenna patterns [2]. After the collection, the radio fingerprint of each location is defined by averaging the RSS measurements or the statistics. The radio fingerprints of all the RPs constitute the radio map. In the online phase, the real-time RSS samples received from the APs are compared against the stored radio map to estimate the user's location.

As can be easily inferred from the above description, building the radio map is a very labor-intensive and time-consuming process, which is the major bottleneck of WiFi fingerprinting in practical applications. To avoid a site survey, researchers have proposed many calibration-free indoor positioning systems [8–14]. Another drawback of WiFi fingerprinting-based localization is the RSS variance problem, which severely degrades the localization accuracy. An RSS variance problem means that the RSS vectors observed in the localization phase are different from the ones collected during the training phase. The RSS variance problem is caused by differences in device type and environmental changes between the two phases.

Crowdsourcing is the most promising solution for solving the site survey problem. This is due to the rise of the smartphone users and every smartphone user may become a potential contributor. By the built-in sensors of the smartphone, the inertial data and WiFi RSS data can be collected. Inertial data can be used to obtain relative trajectory of the user. Based on the relative trajectory, some methods are used to infer the location of each step and label the RSS vectors with the location information. Crowdsourcing is a low-cost and efficient way to extract useful information from data acquired by crowd participants. The crowdsourcing method has been successfully applied to different indoor maps and WiFi radio map construction systems [15,16]. For crowdsourcing-based radio map construction, the RSS variance problem is especially serious. The participant's smartphones are usually different; additionally, the RSS vectors are collected at different times and in different environments.

In this paper, we propose the RCILS, a Robust Crowdsourcing-based Indoor Localization System. RCILS can automatically construct a WiFi radio map using the crowdsourcing data collected by the smartphones. Moreover, RCILS can reduce the influence of RSS variance problem by using a sequence-base radio map. RCILS is based on two key observations. The first observation is that people's activities and trajectories in the indoor environment are restrained by the indoor map. By matching the activities and trajectories to the map, we can get the coordinates of the trajectories and label the RSS collected along the trajectories with location information. The second observation is that, during the localization process, the user is walking and the collected RSS vectors are continuous. From our preliminary experiments, we found that the changing trend of the same path at different times are similar. That is to say, due to the environmental changes, the RSS values may be different at different times, while the changing trend of the RSS during people walking along the same path changes little. Moreover, the changing trends of the RSSs collected by different types of smartphones are also similar, although the RSS values are different due to the device diversity.

The contributions of RCILS include: firstly, RCILS proposes a crowdsourcing-based WiFi radio map construction method; secondly, RCILS propose trajectory fingerprint model for WiFi fingerprint-based localization, which can reduce the RSS variance problem caused by environment changes and device diversity.

In order to realize RCILS, we propose a sequence-based fingerprint model for WiFi fingerprinting indoor localization. The sequence-based fingerprint model can overcome RSS variance problem caused by environment changes and device diversity. We represent indoor map as a semantics graph and model the radio map as the graph model. In the graph-based radio map, the edges represent the RSS sequence on the paths, and the vertexes represent the connection point of the paths. To construct the graph-based radio map, we use activity-based map matching approach to label the RSS collected during the crowdsourcing trajectories. For online localization, RCILS obtains an RSS sequence during the walking process and realizes localization by matching the RSS sequence with the radio map.

The remainder of this paper is organized as follows. Section 2 reviews the related work about crowdsourcing-based indoor localization. Section 3 introduces the methodology of the proposed RCILS. Result and analysis are in Section 4. Section 5 concludes the paper.

2. Related Work

WiFi fingerprinting-based localization is first proposed in a RADAR system [2], which requires a training phase and a localization phase. In the training phase, a radio map is constructed by collecting RSSs from existing APs at all the reference points. In the localization phase, location is determined by the k -nearest neighbor algorithm, which identifies the RSS vector that has the closet Euclidian distance to the currently observed RSS vector. WiFi fingerprinting-based techniques have been widely studied recently, and reviews are given in [17,18].

The major disadvantage of the RADAR system is that the radio map construction is very labor-intensive and time-consuming. Recently, numerous work has been proposed to minimize human effort in fingerprint training [8].

Radio map construction usually involves fingerprint collection and location labeling. For the point model, the fingerprint is collected by point-by-point manual calibration. In the point-by-point manual collection, the target area is partitioned into numerous grid cells, i.e., reference points, and then surveyors collect fingerprint samples at the center of each grids. The coordinate is the location labeling of the reference points. Typically, grids are sized between $2\text{ m} \times 2\text{ m}$ to $5\text{ m} \times 5\text{ m}$, and dozens of samples are collected at each reference points [19]. The point-by-point manual calibration requires considerable time and effort. The walking survey was used instead to reduce the calibration effort of the point-by-point manual collection [20]. In the walking survey, the survey paths are planned in advance and the surveyors walk along the path to collect the fingerprints. The collection points do not have to be specified, and only the specific points, such as the start, corners, and the end point of the paths are marked by the surveyors. The location labeling is obtained by interpolation based on the specific points. Although the walking survey can reduce the collection effort to some extent, it still requires considerable time and effort [21]. Crowdsourcing approaches in which the fingerprint samples are collected from numerous users have been proposed to reduce the cost of radio map construction [8]. The crowdsourced samples can be viewed as unlabeled data since the true locations at which the samples haven been obtained are unknown.

Bolliger et al. proposed a crowdsourcing based radio map construction system named Redpin. In Redpin, the WiFi fingerprints are collected by user uploading [22]. Based on Redpin, Ref. [23] proposed an improved system to increase the number of available samples of the radio measurements by using an accelerometer to detect whether a device is moving or stationary. Similarly, Ref. [24] proposed an organic location system, which constructs radio map by user collaboration. In the system, users manually input their locations. Manual collection limits the application of the crowdsourcing based radio map construction system. Ref. [25] proposed a crowdsourcing based indoor localization system without manual training. In the system, the location of each RSS measurement by imposing constraints on the physics of wireless propagation model. However, it is different to get the accuracy parameters of the wireless propagation model in the complex indoor environment. Woodman and Harle [26] proposed a wearable inertial measurement unit-based WiFi fingerprints automatic construction system. The proposed system realizes pedestrian localization by combining a foot-mounted inertial unit, a detailed building model and a particle filter.

With the development of the smartphones, the built-in sensors can be used for indoor localization. Kim et al. [12] proposed a smartphone-based autonomous war-walking radio map construction system via crowdsourcing. The system used built-in accelerometer and digital compass of the smartphone to realize pedestrian localization. However, the system has the limitation that the initial location and direction need to be given. Zee [13] overcame this limitation by exploiting the constraint of the walls. Zee combined the information extracted from an indoor map and particle filter to realize pedestrian

localization. During the pedestrian localization, the RSS samples of all the locations are collected and the radio map is constructed automatically.

These proposed crowdsourcing methods used a point model-based radio map, which easily suffers from the RSS variance problem caused by environment changes and device diversity. In this paper, RCILS utilizes a trajectory-based radio map model, which can improve the robustness of crowdsourcing-based indoor localization system. WarpMap also used a trajectory-based radio map model for indoor localization [27]. The difference between WarpMap and RCILS is that RCILS proposes a crowdsourcing-based radio map construction system, which uses a trajectory-based model for the data structure of radio map.

3. Methodology

3.1. Trajectory Fingerprint Model

During experiments, we found that the change of the WiFi RSS during a trajectory is smaller than that of the fixed sampling point. We show in Figure 1 how the RSS from an Access Point (AP) changes during the user walking. The RSSs are collected by two different smartphones carried with the user. The user repeated the path four times. From Figure 1, we can see that the RSS values of different smartphones are different. For the same smartphone, the RSS values of different paths are also somewhat different. The RSS difference of two smartphones is caused by the diversity of the WiFi chipsets and antenna. The difference between different paths of the same smartphone is caused by the instability of WiFi strength. However, the changing trend of the RSSs are similar, which can be seen from Figure 1.

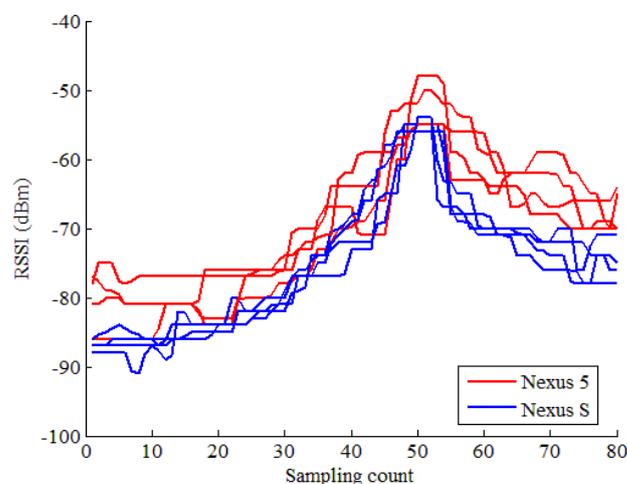


Figure 1. The change of the RSS (Received Signal Strength) from an AP (Access Point) during the user walking.

Based on this observation, RCILS uses a trajectory fingerprint model for indoor localization. In the trajectory fingerprint model, the radio map is stored as a graph $G = (V, E)$. Each node $v \in V$ is a position where a pedestrian would take special activities (special means the activities different from walking straight on level ground, including turning, taking elevator, walking stairs, etc.), and each edge $e = (v_1, v_2) \in E$ corresponds to a trajectory between v_1 and v_2 . Besides the trajectory, the edge also includes the WiFi signatures collected when pedestrians walk along the trajectory.

RCILS includes two phases: radio map construction and trajectory fingerprint-based localization. In the first phase, the radio map is constructed automatically based on crowdsourcing data. In the second phase, RCILS realizes online localization by matching a collected RSS sequence with the fingerprints in the radio map.

3.2. Radio Map Construction

RCILS is a crowdsourcing-based indoor localization system, which utilizes built-in sensors of a smartphone to collect motion data, WiFi fingerprints and air pressure. The motion data includes acceleration, heading and angular velocity. The WiFi fingerprint includes the Medium Access Control (MAC) of the AP and the corresponding Received Signal Strength (RSS) value. The system overview of the proposed radio map construction method is shown in Figure 2.

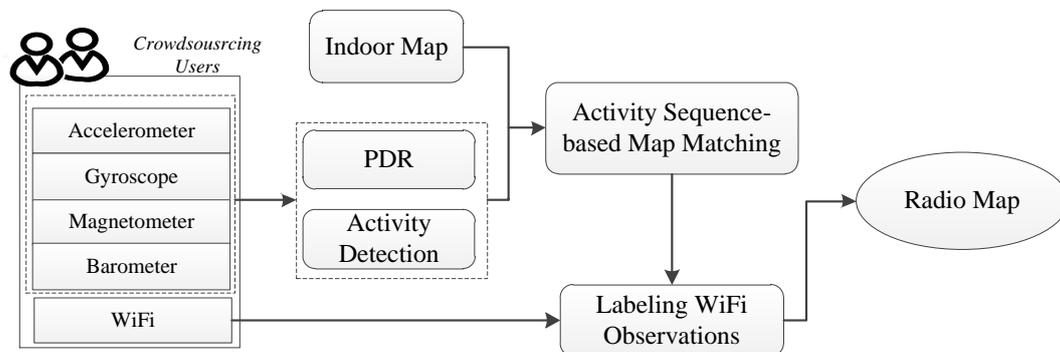


Figure 2. System overview of the proposed radio map construction method.

Based on the collected data, we use an activity detection algorithm to detect the activities and use the pedestrian dead-reckoning (PDR) algorithm to estimate the distance between each two activities. The detected activities and estimated distance between each two activities constitute the activity sequence. In the proposed system, the indoor map is used as a known element. The indoor map contains useful information for indoor localization. On the one hand, it imposes hard constraints on where a pedestrian can walk. On the other hand, based on the user's activities, the indoor map can be used to infer the user's location. For example, if a turn activity is detected, the user may be in a corner. In this paper, the indoor map is used as a semantic graph, in which the edges are the possible user paths and the vertexes are the location where the user may take special activities. Based on the activity sequence and semantic graph of the indoor map, we use activity sequence-based matching to match the trajectory to the indoor map and get the locations of the trajectory. Then, we can label the WiFi observations based on the localization and use the labeled WiFi observations to generate the radio map.

During the online localization phase, the RSS vectors collected during the walking process constitute the RSS sequence. The length of the RSS sequence is determined by the PDR algorithm. Based on the RSS sequence, RCILS realizes pedestrian localization by matching the sequence with the sequence-based radio map.

3.2.1. Semantic Graph Generation

For activity sequence-based map matching, the indoor map should be converted to semantic graph, in which pathways are the edges and the intersections of the pathways are the vertexes, as shown in Figure 3. Based on the semantic graph, the location of the vertexes and displacement between each vertexes can be estimated. Moreover, the vertex also contains semantic information, which is used to match activities to the map. Figure 3 is an example of a semantic graph of the indoor map. The semantic information of the vertexes includes labelling as corner, elevator and stair.

3.2.2. Trajectory Preprocessing

The trajectory of the people in the indoor map has map-related information. On the one hand, the trajectory is restrained by the topology of the map. On the other hand, based on the activity detected during the trajectory, the people's location can be estimated. That is to say, people's location

can be estimated by matching activities to the vertexes of the graph. In order to match the trajectory to the indoor map, we should first detect the activities and estimate the displacement between each two activities.

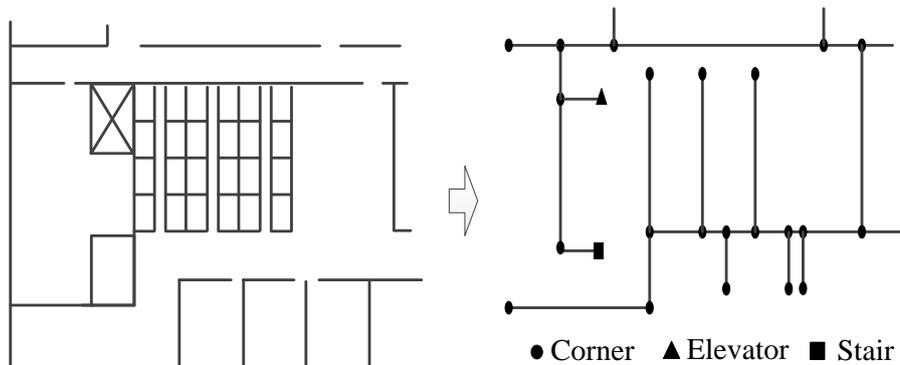


Figure 3. An example of semantic graph of the indoor map.

(1) Activity detection

In an indoor environment, there are usually three types of activities: turning, taking the elevator, and walking stairs. Turning is the most common activity during the walking process. When a pedestrian turns, the angular velocity would generate a peak waveform, as shown in Figure 4 [16]. A turn is detected using the peak detection algorithm, which is used to find the local maximum or minimum during a period of time [28]. To eliminate the influence of the noise, a Butterworth filter of order 4 is used, with a cutoff frequency of 10 Hz.

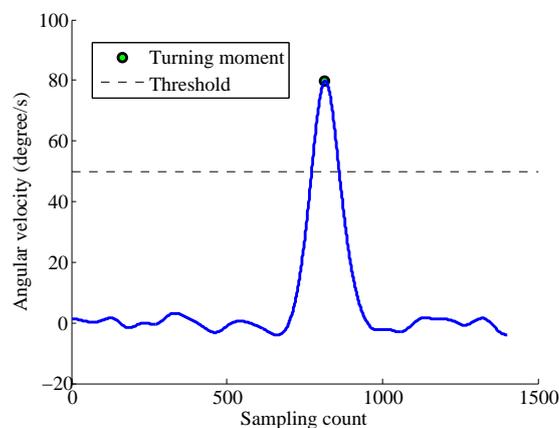


Figure 4. Turn detection by peak detection algorithm.

Generally, when the elevator rises, there will be an overweight state and a subsequent weightless state. On the contrary, when the elevator descends, there will be a weightless state and a subsequent overweight state. Moreover, the air pressure detected by the barometer can also be used for elevator detection, since the air pressure changes with the change of the altitude. The acceleration and pressure of the elevator activity are shown in Figure 5. Another activity with pressure change is walking stairs. Differently from using an elevator, during walking stairs, there is neither an overweight state nor a weightless state. The acceleration and pressure of the walking stairs are shown in Figure 6.

(2) Displacement estimation

The second step of trajectory pre-processing is to estimate the relative displacement between each activity. The distance estimation is implemented by PDR. PDR is a pedestrian localization scheme

that estimates the relative displacement by step detection and heading estimation. Step detection is realized by the peak detection algorithm, as shown in Figure 7. When a step is detected, the location is updated by the following equation:

$$\begin{cases} x_t = x_{t-1} + l \cdot \cos(\theta), \\ y_t = y_{t-1} + l \cdot \sin(\theta). \end{cases} \quad (1)$$

In Equation (1), (x_t, y_t) is the location at time t . l is the step length, calculated using the frequency-based model [29]: $l = a \cdot f + b$, where f is the step frequency, and (a, b) are parameters that can be trained adaptively based on the matching result obtained based on activity sequence-based map matching, which is introduced in the next subsection.

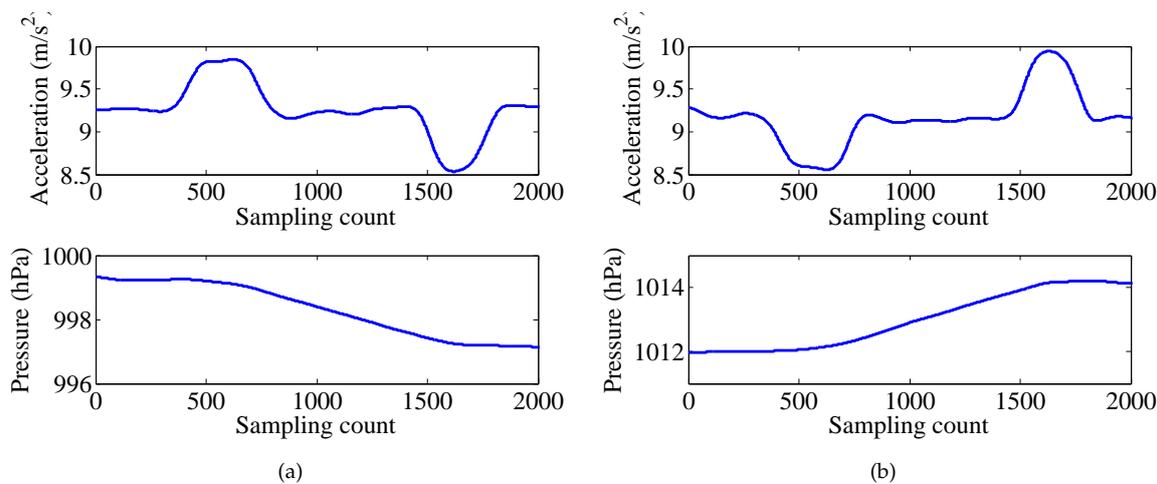


Figure 5. Taking the elevator. (a) up, (b) down.

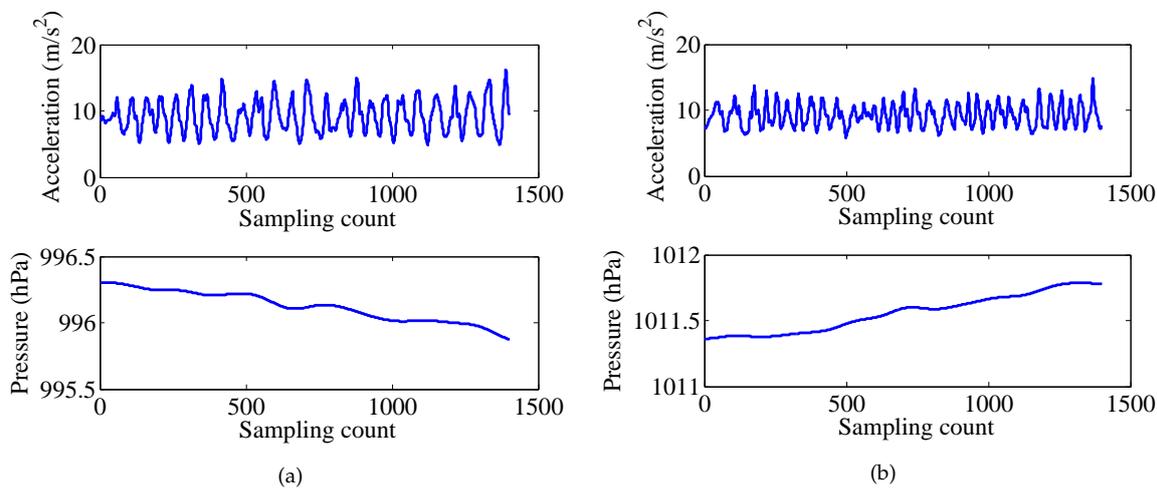


Figure 6. Walking up/down the stairs. (a) up, (b) down.

The step length parameters is trained adaptively based on the matched trajectories. We use Figure 8 as an example to explain the parameters training algorithm. There is a trajectory which has been matched to the indoor map. Based on the known indoor map information, we can get the length of segments AB, BC, CD, DE and EF . Meanwhile, the step numbers that users walked passing these segments can be detected by the step detection algorithms. We assumed that the step length during the same segment is equal. In consequence, the step length for each segment can be calculated. The step frequency is determined based on the step detection result. The step length and

step frequency for these five segments are indicated as: $\langle L, F \rangle = \{(l_1, f_1), (l_2, f_2), (l_3, f_3), (l_4, f_4), (l_5, f_5)\}$. The parameters (a, b) are trained based on vector $\langle L, F \rangle$ using the least squares method.

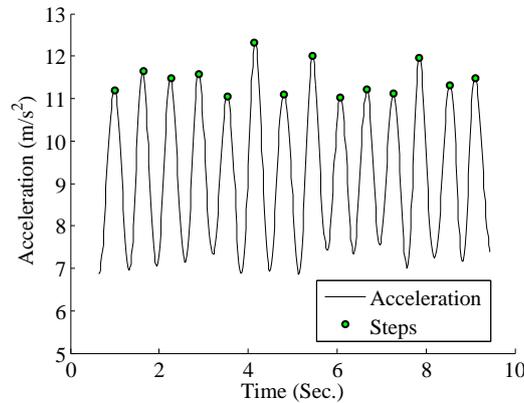


Figure 7. Step detection result.

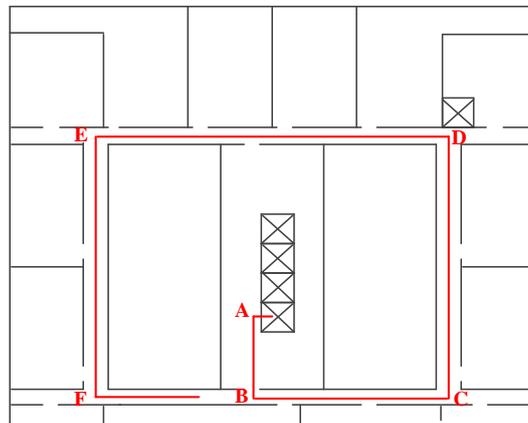


Figure 8. Step length parameters training.

3.2.3. Activity Sequence-Based Map Matching

We use Hidden Markov Model (HMM) to match the activity sequence to the semantic graph of the indoor map. The activity sequence-based map matching method is shown in Figure 9. S_0, S_1, \dots, S_k are the hidden state, namely the nodes of the semantic graph. $P(S_k|S_{k-1})$ is the transition probability from state S_{k-1} to S_k . The transition is assumed to be uniform over all neighbors of a given node. The observations of the HMM are activity detection results and displacement inferred by PDR, represented by Z_k^{act} and Z_k^{PDR} . The subscript k means the observations are obtained at state S_k . $P(Z_k^{act}|S_k)$ and $P(Z_k^{PDR}|S_k)$ are, respectively, the observation probabilities of Z_k^{act} and Z_k^{PDR} . $P(Z_k^{act}|S_k)$ describes the probability of correct activity detection for a given hidden state, namely the confusion matrix. According to the principle of PDR, $P(Z_k^{PDR}|S_k)$ is made up two parts: distance observation probability distribution and heading observation probability distribution. Here, these two probability distributions are assumed to be Gaussian distributions [6,13]. Since distance and heading are independent, the observation probability distributions is defined as

$$P(Z_k^{PDR}|S_k) = \frac{1}{\sqrt{2\pi}\sigma_d} e^{-\frac{1}{2\sigma_d^2}(d_{PDR}-d_{S_k,S_{k-1}})^2} \cdot \frac{1}{\sqrt{2\pi}\sigma_\phi} e^{-\frac{1}{2\sigma_\phi^2}(\phi_{PDR}-\phi_{S_k,S_{k-1}})^2}. \quad (2)$$

σ_d and σ_ϕ are, respectively, the standard deviation of the distance and heading. d_{PDR} is the distance calculated by PDR, and $d_{S_k,S_{k-1}}$ is the distance between S_k and S_{k-1} . ϕ_{PDR} is the heading

estimated by PDR, and $\phi_{S_k, S_{k-1}}$ is the angle between vector $\overrightarrow{S_{k-1}S_k}$ and north direction. d_t is the distance between z_t and the last matched state (indicated by r_{i-1}), d_i is the distance between r_i and r_{i-1} , ϕ_t is the angle between vector $r_{i-1}z_t$ and north direction, and ϕ_i is the angle between vector $r_{i-1}r_i$ and north direction.

Given the detected activity sequence, activity sequence-based map matching aims to find all nodes where the user completes the activities in the activity sequence. The nodes constitute the trajectory. For an activity sequence, there may be many trajectory candidates in the map. We find the best-matching one by the following equation:

$$P(S_k) = P(S_{k-1}) \cdot P(S_k|S_{k-1}) \cdot P(Z_k^{PDR}|S_k) \cdot P(Z_k^{act}|S_k), 1 \leq t \leq T \quad (3)$$

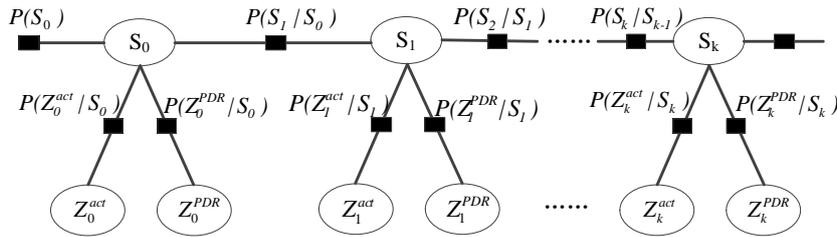


Figure 9. Activity sequence-based map matching method.

By activity sequence-based map matching, we get the tracking results of the crowdsourcing trajectories, namely the locations where the WiFi RSS vectors are collected. Then, we can use these trajectories and RSS vectors to construct the radio map of the indoor environment.

3.2.4. Radio Map Construction

The radio map is stored by the graph structure, $G = (V, E, F)$, where V represents the vertexes, E represents the edges, and F represents the RSS vectors on the edges. By activity sequence-based map matching, the trajectories collected by crowdsourcing can be matched to the semantic graph. The activities contained in the trajectories are matched to the vertexes V of the radio map graph, and the RSS vectors collected on the edges E constitute the RSS vectors F .

3.3. Trajectory Fingerprint-Based Localization

In the online localization phase, the target smartphone collects RSS vectors from the surrounding APs. Moreover, by the inertial sensors of the smartphone, the moving distance can be estimated by PDR. Based on the moving distance, we generate a RSS sequence and realize localization by matching the RSS sequence with the radio map graph.

3.3.1. RSS Sequence Generation

During the moving process, we get an RSS sequence with the length of the moving distance. We use $S_t = (F_t - w + 1, F_t - w + 2, \dots, F_t)$ to denote the RSS sequence collected during the moving distance, where w is the window size and F_t is the latest collected RSS sample. $F_i = \{(mac_1, r_{ss}(1, i)), \dots, (mac_j, r_{ss}(j, i)), \dots, (mac_m, r_{ss}(m, i))\}$, mac_j and $r_{ss}(j, i)$ are, respectively, the MAC address and RSS value of the j th WiFi AP. The RSS sequence can be represented by a $m \times w$ matrix, where m is the number of the APs and w is the length of the moving window. The MAC list is $(mac_1, mac_2, \dots, mac_m)$:

$$S_t = \begin{pmatrix} rss(1,1) & rss(1,2) & \dots & rss(1,w) \\ rss(2,1) & rss(2,2) & \dots & rss(2,w) \\ \dots & \dots & \dots & \dots \\ rss(m,1) & rss(m,2) & \dots & rss(m,w) \end{pmatrix}. \quad (4)$$

3.3.2. Graph-Based Trajectory Search

Based on the RSS sequence generation during the moving window, trajectory fingerprint-based localization is to search the best-match sequence from the radio map graph and determine the location of the best-match sequence as the target's location.

We use Breadth-First-Search to search the best-match sequence in the graph. Searching in the whole graph needs a large computational amount. In this paper, we determine the start vertex based on the similarity between the AP list of S_t and that of the vertex. We use the Jaccard similarity coefficient as the similarity parameter. The Jaccard coefficient is a statistic used for comparing the similarity and diversity of sample sets, which has been used for WiFi-based clustering in [16]. The Jaccard similarity coefficient is calculated using the following equation:

$$J(MAC_t, MAC_i) = \frac{MAC_t \cap MAC_i}{MAC_t \cup MAC_i} \quad (5)$$

where MAC_t is the MAC of the AP list of S_t , and MAC_i is the MAC of the AP list of vertex i . After determining the first vertex, we conduct Breadth-First-Search c steps to find the best-match sequence, where c is the constant, set to 3 herein.

3.3.3. Localization

Trajectory Fingerprint-based localization is to find the best-match sequence based on the collected RSS sequence. During the graph-based trajectory searching, we calculated the similarity metric between RSS sequence and RSS in the radio map graph (called candidate RSS sequence). From Figure 1, we can see that the RSS values of different paths are different, even for the same smartphone. Therefore, using the RSS value as the similarity metric may cause localization error. In this paper, we use the correlation coefficient as the similarity metric. As before, we use S_t to denote the RSS sequence collected during the moving distance, as shown in Equation 4. There are m APs in S_t , for each AP, we calculate the similarity metric, and use the sum of these metrics as the similarity between S_t and the candidate RSS sequence:

$$P = \sum_{i=1}^m \frac{cov(RS_i, RC_i)}{\sigma_{RS_i} \cdot \sigma_{RC_i}}, \quad (6)$$

where RS_i is the RSS set in the collected RSS sequence of the i th AP, and RC_i is the RSS set in the candidate RSS sequence of the i th AP:

$$RS_i = \{rss_s(i,1), rss_s(i,2), \dots, rss_s(i,w)\},$$

$$RC_i = \{rss_c(i,1), rss_c(i,2), \dots, rss_c(i,w)\}.$$

For the locations with null reading from the AP, -100 dB was used as the RSS value.

For each candidate RSS sequence, we get a similarity coefficient by Equation (6). We use the k-nearest neighbour (knn) algorithm to determine the best-match sequence and use the location of the terminal as the localization result. In our experiments, we set the k equal to 1 in the knn algorithm.

4. Evaluation

4.1. Experiment Setup

To evaluate RCILS, we performed experiments in an office building, with a $52.5 \text{ m} \times 52.5 \text{ m}$ floor plan, as shown in Figure 10. We used two different types of Android smartphones, namely Nexus S and Nexus 5, to collect the trajectory data. During the experiment, participants held two smartphones on their two hands in front of themselves and walked normally in the accessible areas of the building. Holding the two smartphones on their two hands causes the WiFi RSSs to be received at the same time. To simulate the crowdsourcing users, participants started at different positions. To evaluate the performance with incremental data, each trace is repeated ten times. In total, 200 user trajectories were collected by three participants using two types of smartphones. In terms of time, these trajectories correspond to 220 min of data collection. The collected data includes acceleration data, compass data, gyroscope data, barometer data, and WiFi.



Figure 10. Experimental environment.

4.2. Performance with Incremental Data

The Cumulative Distribution Function (CDF) of localization error with incremental crowdsourced data is shown in Figure 11. We set different lengths for sliding windows, namely 20 samples, 50 samples, 100 samples, 150 samples, 200 samples, and 250 samples. We can see that, as the crowdsourcing data amount increases, the localization error decreases. When the length of the sliding window is 50 samples, for 15 min data, the 80 percentile of localization error is about 21 m, and when the data amount increases to 45 min, the 80 percentile of localization error decreases to about 15 m. The localization error decreases sharply when the data amount increases from 15 min to 45 min. However, when the data amount is more than 45 min, the decline of the localization error becomes smaller as the data amount increases.

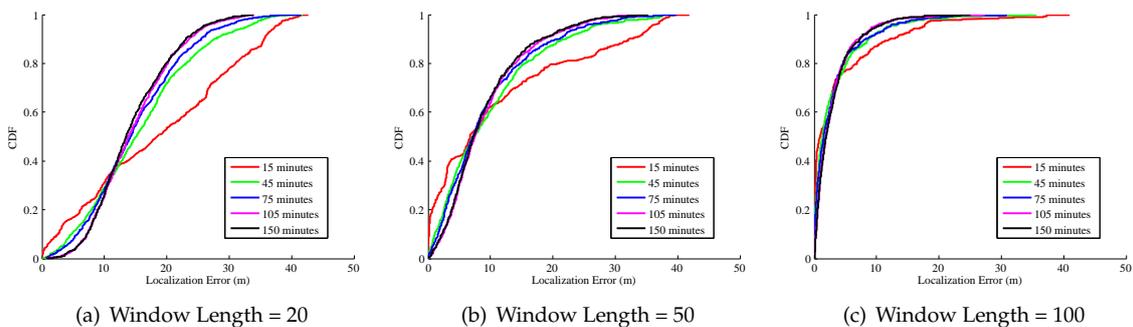


Figure 11. Cont.

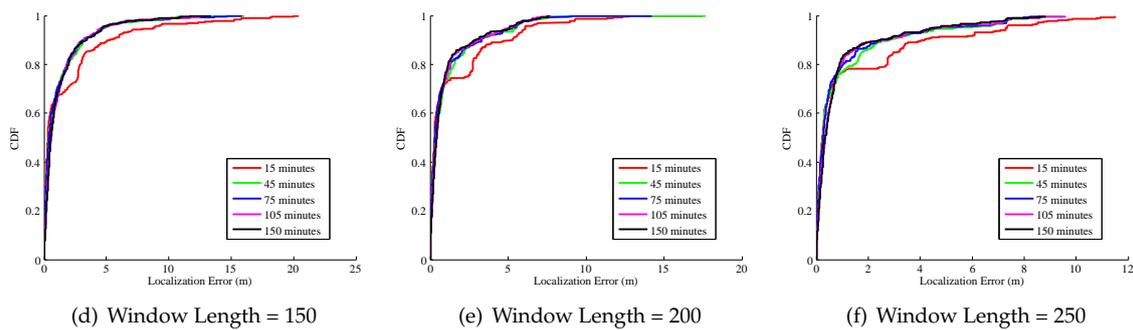


Figure 11. Localization error with incremental data.

4.3. Performance with Length of Sliding Window

Given the fixed data amount (data amount is set to 45 min), Figure 12 shows the CDF of localization error in different lengths of sliding windows. It can be seen that, with the increase of the length of the sliding window, the localization error decreases. When the length increases from 20 to 250, the 80 percentile of localization error decreases from 23 m to 1.3 m.

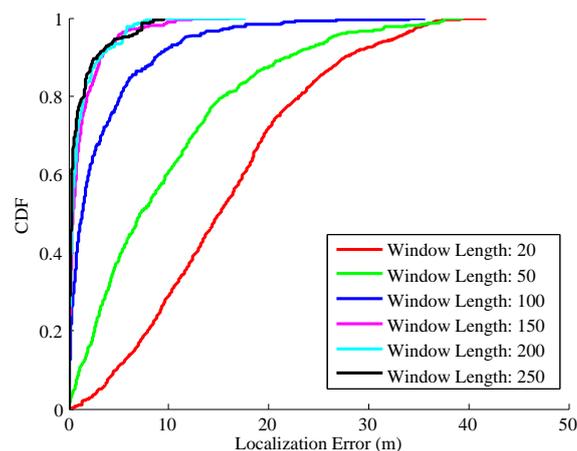


Figure 12. The CDF of localization error with different lengths of sliding windows (the data amount is 45 min).

The mean localization error and time delay with different lengths of sliding windows is shown in Figure 13. From Figure 13, we can see that with the increase of lengths of sliding windows, the localization error decreases. However, the long length of sliding window means a long time delay of the localization system. As it can be seen from Figure 13, when the length is 20 samples, the time delay is 1.9 s, and when the length increases to 250 samples, the time delay is 25 s.

Figure 13 shows the tradeoff between localization error and time delay with the increasing of length of sliding window. From Figure 13, we can see that the time delay increases in linear proportion to the length of sliding window. However, the downtrend of the localization error becomes smaller and smaller as the length of sliding window increases. For an online localization system, we must get a good tradeoff between localization error and time delay. For example, we can set the length to 100, and the time delay at the beginning of the system startup is 9.9 s, and the localization error is 1.6 m. Certainly, after the first localization process (i.e., the first 100 samples), the localization system does not need a startup process, and it can use the scanned samples for localization. That is to say, the proposed system just needs one time delay process at the first startup. For the offline tracking system, the time delay can be ignored, and we can use the longest length of the sliding window.

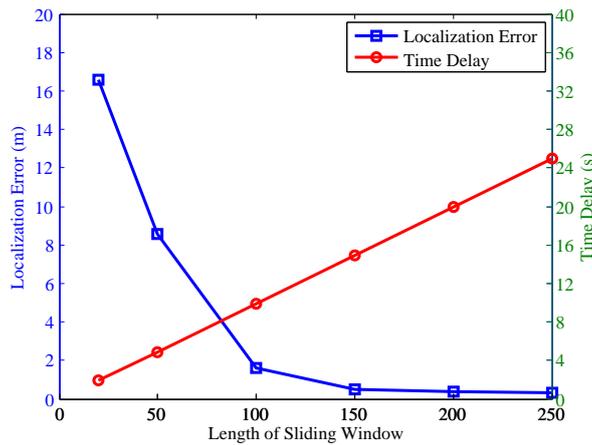


Figure 13. Localization error and time delay with different lengths of sliding windows (the data amount is 45 min).

4.4. Performance of Different Fingerprint Models

We compared the proposed method with the point fingerprint model. In the point model, the radio map consists of fingerprints at each reference point. The localization error of different fingerprint models is shown in Figure 14. We evaluate radio map construction method from the following two aspects: crowdsourcing data amount and device diversity.

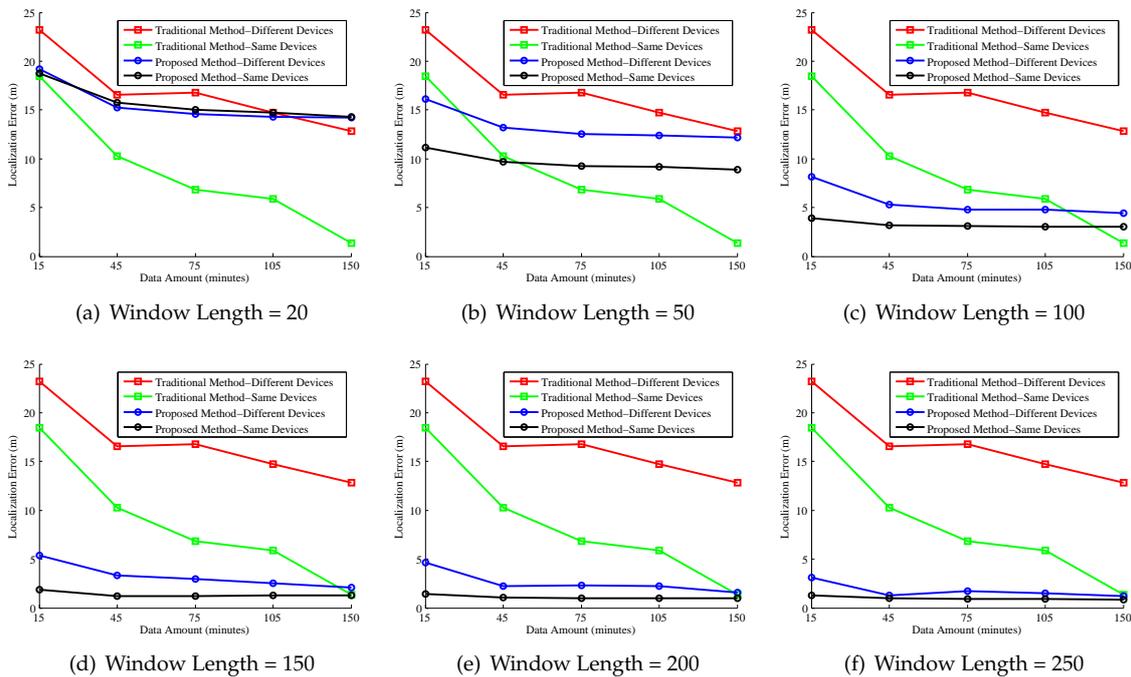


Figure 14. Localization error of different methods.

From Figure 14, we can see that with the increasing of the data size, the localization error of the two methods decreases. However, the data amount needed for the proposed method is much more than that of the traditional method.

For the same device, if the window length is set to 150, in order to achieve 2 m localization error, the data amount needed for RCILS is 15 min, while that for the traditional method is 150 min. RCILS needs smaller amounts of data for crowdsourcing-based localization. This demonstrates that

RCILS is more applicable than the traditional crowdsourcing-based system using a point-based radio map model.

Moreover, we evaluate the impact of device diversity to RCILS. In the evaluation, the data used for radio map construction and that used for online localization are different. This is common in the crowdsourcing systems since the smartphones used by the crowdsourced participants are usually different. From Figure 14, we can see that the localization error of the proposed method is much smaller than that of the traditional point-based radio map. The result demonstrates that the proposed RCILS can reduce the device diversity problem for the crowdsourcing-based indoor localization system.

4.5. Comparison with State-of-the-Art WiFi-Based Indoor Localization Systems

The localization performance of RCILS is compared with that of EZ [25], WiGEM [30], WILL [14], UnLoc [31], Zee [13] and LiFS [10], as shown in Table 1. We compare the localization methods in terms of accuracy, map requirements, anchor point, and device heterogeneity. We can see from the Table 1 that the median error of RCILS is 1.6 m (when the sliding window is set to 100), which is less than that of the other methods. The median error of UnLoc is 1.69 m, which is close to RCILS. However, UnLoc needs sufficient numbers of anchor points (e.g., occasional GPS location), which is not satisfied in many indoor environments. For device heterogeneity problem, EZ [25] and WiGEM [30] used a learning-based approach to train the model parameters, while RCILS used the trajectory-based fingerprint model to reduce the influence caused by the device heterogeneity. The localization error of RCILS is smaller than that of EZ and WiGEM. WILL [14] and LiFS [10] are two room-level localization systems, which is different from RCILS. Zee [13] is a map-assisted localization approach which leverages the topology of the map to restrict pedestrian's trajectory based on a particle filter. However, particle filter is time-consuming and may be not suitable for online localization based on a smartphone.

Table 1. Comparison with state-of-the-art WiFi-based indoor localization systems.

Method	Reported Accuracy	Map requirement	Anchor point	Device heterogeneity
EZ [25]	Median error ~2 m inside small building (486 m ²) and 7 m inside big building (12,600 m ²)	No	Yes	Yes
WiGEM [30]	Median error ~4 m inside small building (600 m ²) and 6 m inside bug building (3250 m ²)	Yes	No	Yes
WILL [14]	86% room level accuracy inside medium sized academic building (1600 m ²)	Yes	No	No
UnLoc [31]	Median error ~1.69 m across three different indoor scenes (largest begin 4000 m ²)	No	Yes	No
Zee [13]	Median error ~3 m inside medium sized building (2275 m ²)	Yes	No	No
LiFS [10]	89% room level accuracy inside medium sized academic building (1600 m ²)	Yes	No	No
RCILS	Median error ~1.6 m inside medium sized academic building (2750 m ²)	Yes	No	Yes

5. Conclusions

In this paper, we propose a robust crowdsourcing-based indoor localization system. RCILS can automatically construct a WiFi radio map based on the crowdsourcing data. In RCILS, an indoor map is first converted to a semantic graph. The trajectory is preprocessed by activity detection and pedestrian dead-reckoning. By trajectory preprocessing, we get the activity sequence contained in the trajectory. Based on the semantic graph and activity sequence, we match the trajectory to the indoor map to get the location of the trajectory. That is to say, the location where the WiFi RSS is collected is determined by the trajectory matching. Then, the radio map is constructed based on the crowdsourcing trajectories. To overcome the RSS variance problem, we use a trajectory fingerprint

model. The experiment results in an office building demonstrate that the proposed RCILS can reduce the variance problem caused by device diversity and environment changes.

In future work, we will include more activities in RCILS, such as opening the door, sitting in the office, and so on. RCILS is an offline system at the moment. We intend to develop an online RCILS system, in which the crowdsourcing data uploading and localization can be realized in real time.

Acknowledgments: This work was supported in part by the National Key Research Development Program of China (2016YFB0502203); by the China Postdoctoral Science Foundation (2015M580732, 2016T90800); by the Open Research Fund Program of State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing (16102); by the Shenzhen Future Industry Development Funding Program (201507211219247860); and by the National Natural Science Foundation of China (41371377, 91546106, 41401444).

Author Contributions: The framework was proposed by Qingquan Li, and further development and implementation were realized by Baoding Zhou. Qingzhou Mao and Wei Tu mainly studied some of the ideas and analyzed the experiment results.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, J.; Chen, R.; Pei, L.; Guinness, R.; Kuusniemi, H. A hybrid smartphone indoor positioning solution for mobile LBS. *Sensors* **2012**, *12*, 17208–17233.
2. Bahl, P.; Padmanabhan, V.N. RADAR: An in-building RF-based user location and tracking system. In Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000), Tel Aviv, Israel, 26–30 March 2000; Volume 2, pp. 775–784.
3. Ni, L.M.; Liu, Y.; Lau, Y.C.; Patil, A.P. LANDMARC: Indoor location sensing using active RFID. *Wirel. Netw.* **2004**, *10*, 701–710.
4. Bargh, M.S.; de Groote, R. Indoor localization based on response rate of bluetooth inquiries. In Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-Less Environments, San Francisco, CA, USA, 19 September 2008; ACM: New York, NY, USA, 2008; pp. 49–54.
5. Stelios, M.A.; Nick, A.D.; Effie, M.T.; Dimitris, K.M.; Thomopoulos, S.C. An indoor localization platform for ambient assisted living using UWB. In Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia, Linz, Austria, 24–26 November 2008; ACM: New York, NY, USA, 2008; pp. 178–182.
6. Zhou, B.; Li, Q.; Mao, Q.; Tu, W.; Zhang, X. Activity sequence-based indoor pedestrian localization using smartphones. *IEEE Trans. Hum. Mach. Syst.* **2015**, *45*, 562–574.
7. Shang, J.; Hu, X.; Cheng, W.; Fan, H. GridiLoc: A Backtracking Grid Filter for Fusing the Grid Model with PDR Using Smartphone Sensors. *Sensors* **2016**, *16*, 2137.
8. Hossain, A.M.; Soh, W.S. A survey of calibration-free indoor positioning systems. *Comput. Commun.* **2015**, *66*, 1–13.
9. Kim, Y.; Shin, H.; Chon, Y.; Cha, H. Crowdsensing-based Wi-Fi radio map management using a lightweight site survey. *Comput. Commun.* **2015**, *60*, 86–96.
10. Yang, Z.; Wu, C.; Liu, Y. Locating in fingerprint space: Wireless indoor localization with little human intervention. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2012; ACM: New York, NY, USA, 2012; pp. 269–280.
11. Zhang, C.; Subbu, K.P.; Luo, J.; Wu, J. GROPING: Geomagnetism and cROwdsensing Powered Indoor Navigation. *IEEE Trans. Mob. Comput.* **2015**, *14*, 387–400.
12. Kim, Y.; Chon, Y.; Cha, H. Smartphone-based collaborative and autonomous radio fingerprinting. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* **2012**, *42*, 112–122.
13. Rai, A.; Chintalapudi, K.K.; Padmanabhan, V.N.; Sen, R. Zee: Zero-effort crowdsourcing for indoor localization. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2012; ACM: New York, NY, USA, 2012; pp. 293–304.
14. Wu, C.; Yang, Z.; Liu, Y.; Xi, W. WILL: Wireless indoor localization without site survey. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *24*, 839–848.
15. Wu, C.; Yang, Z.; Liu, Y. Smartphones based crowdsourcing for indoor localization. *IEEE Trans. Mob. Comput.* **2014**, *14*, 444–457.

16. Zhou, B.; Li, Q.; Mao, Q.; Tu, W.; Zhang, X.; Chen, L. ALIMC: Activity landmark-based indoor mapping via crowdsourcing. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2774–2785.
17. Liu, H.; Darabi, H.; Banerjee, P.; Liu, J. Survey of wireless indoor positioning techniques and systems. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* **2007**, *37*, 1067–1080.
18. He, S.; Chan, S.H.G. Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 466–490.
19. Han, D.; Jung, S.; Lee, M.; Yoon, G. Building a practical Wi-Fi-based indoor navigation system. *IEEE Pervasive Comput.* **2014**, *13*, 72–79.
20. Han, D.; Lee, S.; Kim, S. Kailos: Kaist indoor locating system. In Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, 27–30 October 2014; pp. 615–619.
21. Jung, S.H.; Moon, B.C.; Han, D. Performance Evaluation of Radio Map Construction Methods for Wi-Fi Positioning Systems. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 880–889.
22. Bolliger, P. Redpin-adaptive, zero-configuration indoor localization through user collaboration. In Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-Less Environments, San Francisco, CA, USA, 19 September 2008; ACM: New York, NY, USA, 2008; pp. 55–60.
23. Bolliger, P.; Partridge, K.; Chu, M.; Langheinrich, M. Improving location fingerprinting through motion detection and asynchronous interval labeling. In *Location and Context Awareness*; Springer: Tokyo, Japan, 2009; pp. 37–51.
24. Park, J.G.; Charrow, B.; Curtis, D.; Battat, J.; Minkov, E.; Hicks, J.; Teller, S.; Ledlie, J. Growing an organic indoor location system. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, San Francisco, CA, USA, 15–18 June 2010; ACM: New York, NY, USA, 2010; pp. 271–284.
25. Chintalapudi, K.; Padmanabha Iyer, A.; Padmanabhan, V.N. Indoor localization without the pain. In Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking, Chicago, IL, USA, 20–24 September 2010; ACM: New York, NY, USA, 2010; pp. 173–184.
26. Woodman, O.; Harle, R. RF-based initialisation for inertial pedestrian tracking. In *Pervasive Computing*; Springer: Nara, Japan, 2009; pp. 238–255.
27. Ye, X.; Wang, Y.; Hu, W.; Song, L.; Gu, Z.; Li, D. WarpMap: Accurate and Efficient Indoor Location by Dynamic Warping in Sequence-Type Radio-Map. In Proceedings of the 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), London, UK, 27–30 June 2016; pp. 1–9.
28. Mladenov, M.; Mock, M. A step counter service for Java-enabled devices using a built-in accelerometer. In Proceedings of the 1st International Workshop on Context-Aware Middleware and Services: Affiliated with the 4th International Conference on Communication System Software and Middleware (COMSWARE 2009), Dublin, Ireland, 16 June 2009; ACM: New York, NY, USA, 2009; pp. 1–5.
29. Cho, D.K.; Mun, M.; Lee, U.; Kaiser, W.J.; Gerla, M. Autogait: A mobile platform that accurately estimates the distance walked. In Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom), Mannheim, Germany, 29 March–2 April 2010; pp. 116–124.
30. Goswami, A.; Ortiz, L.E.; Das, S.R. WiGEM: A learning-based approach for indoor localization. In Proceedings of the Seventh Conference on Emerging Networking EXperiments and Technologies, Tokyo, Japan, 6–9 December 2011; ACM: New York, NY, USA, 2011.
31. Wang, H.; Sen, S.; Elgohary, A.; Farid, M.; Youssef, M.; Choudhury, R.R. No need to war-drive: Unsupervised indoor localization. In Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, Low Wood Bay, UK, 25–29 June 2012; ACM: New York, NY, USA, 2012; pp. 197–210.

