Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 109642, 11 pages
http://dx.doi.org/10.1155/2015/109642

Research Article

Improving Wi-Fi Indoor Positioning via AP Sets Similarity and Semi-Supervised Affinity Propagation Clustering

Xuke Hu,[1,2] Jianga Shang,[1,2] Fuqiang Gu,[1,2] and Qi Han[3]

[1]*Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China*
[2]*National Engineering Research Center for Geographic Information System, Wuhan 430074, China*
[3]*Department of Electrical Engineering and Computer Science, Colorado School of Mines, Golden, CO 80401, USA*

Correspondence should be addressed to Jianga Shang; jgshang@cug.edu.cn

Received 18 September 2014; Revised 16 December 2014; Accepted 18 December 2014

Academic Editor: Javier Bajo

Indoor localization techniques using Wi-Fi fingerprints have become prevalent in recent years because of their cost-effectiveness and high accuracy. The most common algorithm adopted for Wi-Fi fingerprinting is weighted K-nearest neighbors (WKNN), which calculates K-nearest neighboring points to a mobile user. However, existing WKNN cannot effectively address the problems that there is a difference in observed AP sets during offline and online stages and also not all the K neighbors are physically close to the user. In this paper, similarity coefficient is used to measure the similarity of AP sets, which is then combined with radio signal strength values to calculate the fingerprint distance. In addition, isolated points are identified and removed before clustering based on semi-supervised affinity propagation. Real-world experiments are conducted on a university campus and results show the proposed approach does outperform existing approaches.

1. Introduction

Localization techniques are essential for increasingly popular location-based services in pervasive computing and Internet of Things. In indoor environments, GPS signals cannot penetrate, so Wi-Fi based localization methods become the dominant positioning technique. Fingerprinting based on received signal strength (RSS) is the most popular method of indoor positioning that was first proposed in the radar system [1]. This technique includes an offline training phase and an online location estimation phase. The offline phase detects Wi-Fi signal strength from surrounding APs and collects location fingerprints to create a radio map. In the online phase, a Wi-Fi enabled mobile device obtains a vector of signal strengths in real time. These signal measurements are then compared to the fingerprints in the radio map. The location of the best matching fingerprint is used as the estimated location.

The Wi-Fi fingerprinting based method has two problems.

(i) *APs mismatch:* we have found that the visible Wi-Fi access points (APs) set varies over time and space, which implies that there might not exist a perfect match for current location when trying to find a match in the fingerprint database. Existing work typically uses one of the two methods to work around this issue. One is to select the APs shared by online radio measurements and reference points (RPs) in radio map, and then these APs with the strongest RSS will be used for final matching [2–4]. The other method is to assign a small RSS value to a nonobserved AP [5–7]. However, both methods fail to completely represent signal characteristics at a certain location due to the addition or deletion of some information, which ultimately affects the accuracy of fingerprint distance estimation.

(ii) *Clustering inefficiency:* to compare multiple locations and select the one that best matches the observed signal strength, weighted K-nearest neighbors (WKNN) method is often used due to its simplicity. However, due to the inherent time-varying nature of wireless signal, K-nearest neighboring points are not always close to a user's real position. Consequently, directly

using K-nearest neighbors may lead to bad estimations. To address this problem, clustered WKNN algorithms have been developed [8–13], and most of them either use c-means [10] or K-means [12] as the clustering algorithm. However, they need to prespecify the clustering number, which is unsuitable for classifying RPs whose distribution is initially unknown. In fact, ideally the cluster should have low similarity to other clusters.

In this work, we aim to tackle these two problems in Wi-Fi fingerprinting based localization method: we address the APs mismatch problem by combining AP sets similarity and RSS distance when calculating fingerprint distance and we address the clustering inefficiency issue by enhancing semi-supervised affinity propagation clustering algorithm in combination with detection of isolated points. Hence, we call the improved algorithm semi-supervised affinity propagation based WKNN (WKNN-SAP). WKNN-SAP has been implemented on a server and evaluated through real-world experiments performed in a teaching building.

The rest of this paper is organized as follows. We discuss related work in Section 2. Section 3 provides an overview of WKNN-SAP, followed by algorithm details in Section 4. Results of performance evaluation are analyzed in Section 5 and finally Section 6 concludes the paper.

2. Related Work

Research in Wi-Fi fingerprinting localization recently has been mainly focusing on how to improve the collection of signal fingerprints and how to improve localization accuracy. For fingerprint collection, various alternatives have been proposed and they include using a signal propagation model, ray-tracing [14], interpolation, and even unsupervised crowdsourcing using the inertial sensors and indoor floor plans [15]. To improve localization accuracy, researchers have adapted to heterogeneous Wi-Fi clients [16, 17], used historical tracks, and fused other positioning techniques such as PDR [18, 19] or acoustic ranging [20, 21]. In our work, we focus on two specific issues in current method: APs mismatch and clustering inefficiency.

The visible APs are often different during online and offline stages. In existing work, researchers simply select these common APs with the largest signal strength for matching [2–4] or set the corresponding RSS entity to a small value (e.g., −110 dBm) if no RSS reading is found for an AP [5–7]. In fact, these approaches either introduce some false information or omitted some useful information. Thus, they cannot accurately describe the signal characteristic of surrounding Wi-Fi environment.

Due to the time-varying nature of indoor radio propagation, received signal measurements in online stage are different from the fingerprints collected in offline stage. To eliminate this adverse impact, researchers used clustering technique to partition the neighbors into multiple clusters and the one with the most members and/or the lowest average RSS distance is chosen as delegate for calculating the position of target. Ma et al. first proposed the cluster filtered KNN (CFK) method [9], using hierarchical clustering to partition the nearest neighbors of RPs. The results show that KNN is improved when clustering technology is used. Altintas and Serif [12] improve this method by replacing hierarchical clustering with K-means to gain a higher positioning accuracy. Likewise, Sun et al. [10] developed a KNN-FCM hybrid algorithm. They use fuzzy c-means (FCM) clustering method to divide K-nearest neighbors into several clusters and one cluster is chosen to calculate user position. As a consequence, this method has better results than KNN when the distance error is less than 2 meters. However, it is difficult to specify K value for K-means and fuzzy c-means, while hierarchical clustering cannot make sure that obtained clusters achieve the greatest sum of distance between all clusters. From a different perspective, Tian and Au et al. [13, 22] both used affinity propagation to cluster fingerprints in the offline stage. First, a coarse-grained position which is normally represented by one or more clusters is estimated through clustering matching in online stage, followed by a fine-grained positioning. The disadvantage is that clustering all the RPs in offline stage takes more time, and in fact only a small part of RPs contributes to accurate positioning.

3. Algorithm Overview

Figure 1 depicts the overall flow of WKNN-SAP. We first create an offline radio map and then collect RSS vectors in real time for any point that we need its location. For each point, we calculate the AP sets similarity and the average RSS distance between this vector and all the reference points in the radio map. We then estimate fingerprint distances to all the reference points with our proposed fingerprint distance model. To get more suitable neighbors for comparison [23, 24], K-nearest neighbors are dynamically selected according to a signal distance threshold before eliminating isolated points. The neighbors are then partitioned into several clusters with semi-supervised affinity propagation (SAP) and one cluster is chosen as the delegate. The user's position is then calculated by WKNN.

4. Details of WKNN-SAP

4.1. Calculation of Fingerprint Distance by Combining AP Sets Similarity and RSS Distance. In the online stage of the Wi-Fi fingerprinting based localization method, trying to find an exact match of visible AP sets in the radio map constructed offline is a challenge. Several factors contribute to the differences in AP sets.

(i) *Limited coverage of an AP:* scattered deployment of Wi-Fi APs at different physical locations and limited signal coverage of each AP will inevitably lead to only partial APs being visible at certain locations of indoor buildings.

(ii) *Signal interference:* wireless signals are vulnerable to multipath and shadow effects as well as people's movement. Even performing multiple scans at the same position, visible AP sets are not always the same.

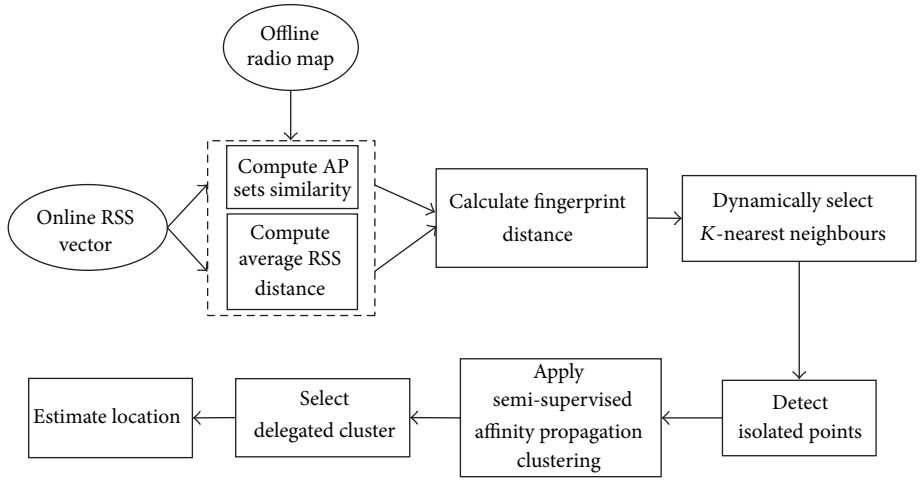

Figure 1: Flow of WKNN-SAP.

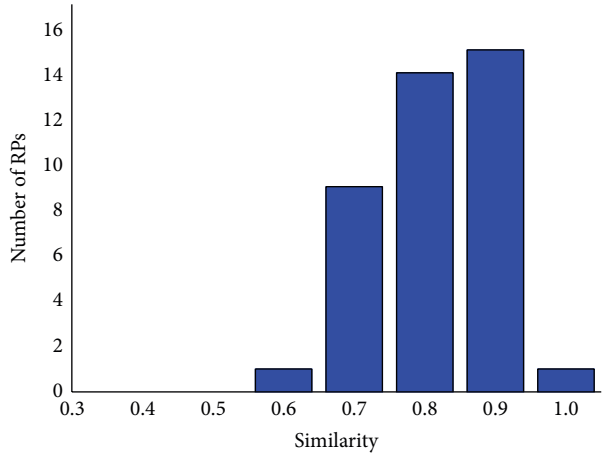

Figure 2: Histogram of AP sets similarity at different positions.

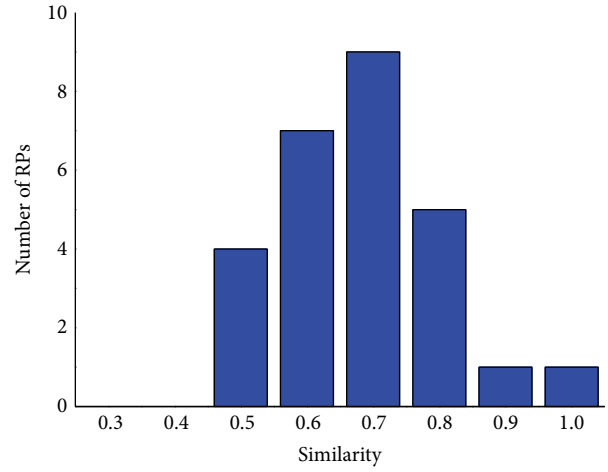

Figure 3: Histogram of AP sets similarity at different times.

(iii) *Insertion or removal of APs:* some damaged APs may be removed manually, or new APs may be added into the environment, which will cause the difference of AP sets between RPs and observed signal vector.

To represent the similarity of two AP sets, we use Jaccard similarity coefficient [25]. Jaccard coefficient measures similarity between two sets and is defined as the size of the intersection divided by the size of the union of the sample sets: $\text{sim}(A, B) = |A \cap B|/|A \cup B|$.

Our experiments have validated the causes of APs mismatch described above. In an area of about 30 m^2, we choose an RP as the base point and calculate the Jaccard similarity of AP sets between base point and other 39 RPs. Figure 2 shows the histogram of the similarity: the number of RPs whose AP set is the same as the base point (i.e., similarity equals 1.0) is small. In other words, visible APs at different positions differ considerably, even if these positions are physically close. In addition, we conducted continuous scan at a certain position to capture multiple signal vectors and then choose one signal vector as the base point. We calculate the AP sets Jaccard similarity between base point and other signal vectors. The results in Figure 3 show that it is very rare to have the same visible APs at different times even at the same position.

Using AP sets similarity or RSS distance alone to represent fingerprint distance results in undesirable outcomes: AP sets similarity only provides coarse-grained information about two reference points, while RSS distance may delete some useful signal characteristics [2–4] or add artificial information [5–7]. Therefore, we propose to combine AP sets similarity and RSS distance to calculate the distance between two fingerprints. Fingerprint is defined as $\text{fp} = \{\text{coor}, \text{APs}, \text{rssis}\}$, where coor is the coordinate and it is an unknown quantity in online measurements; APs are the AP set of fingerprint points and $\text{APs} = \{\text{ap}_1, \text{ap}_2, \ldots, \text{ap}_n\}$; rssis is the set of signal strength of corresponding APs. Therefore, to estimate the AP sets similarity of two fingerprints or measurements, we should calculate the set similarity between set $\text{APs}_1 = \{\text{ap}_{11}, \text{ap}_{12}, \ldots, \text{ap}_{1n}\}$ and set $\text{APs}_2 = \{\text{ap}_{21}, \text{ap}_{22}, \ldots, \text{ap}_{2m}\}$, where m and n are the size of two sets and probably are not equal.

We denote RSS average distance of common APs between observation signal vector and RPs using L_q. For instance, when $q = 1$, L_1 is Manhattan distance; when $q = 2$, L_2 is Euclidean distance:

$$L_q = \frac{\left(\sum_{i=1}^{sl} \left|\text{rssi}_{oi} - \text{rssi}_{ti}\right|^q\right)^{1/q}}{sl}, \quad (1)$$

where sl is the number of elements in intersection set and rssi_{oi} and rssi_{ti} are the signal strength received at two different RPs from the same AP.

Typically, a small similarity of AP sets means a larger distance between two fingerprints, while a small RSS distance implies small distance between them, and vice versa. To find a reasonable model describing the relationship between the two independent variables (i.e., AP sets similarity and RSS distance) and a dependent variable (i.e., fingerprint distance), we use 1stOpt [26] to fit their linear or nonlinear relationships. 1stOpt is a set of mathematical optimization analysis software packages specializing nonlinear regression, curve fitting, and parameter estimation of misaligned complex models. This modeling problem can be described as follows: based on the AP sets similarity sim and RSS average distance of the AP intersection set, derive the real physical distance between two RPs. Therefore, we seek a function that minimizes the difference between the estimated distance and the physical distance. If the AP sets similarity and RSS distance between two reference points are 1 and 0, respectively, the estimated distance should be 0.

Based on the fitting results, we use the following model to compute fingerprint distance FD_q:

$$\text{FD}_q = p_1 * \frac{\left(\sum_{i=1}^{sl} \left|\text{rssi}_{oi} - \text{rssi}_{ti}\right|^q\right)^{1/q}}{(sl) + p_2 * \ln(\text{sim})}, \quad (2)$$

where p_1 and p_2 are parameters and sim is Jaccard similarity coefficient calculated. Because only proximity degree instead of the absolute distance is needed when performing comparisons, we merge the two parameters p_1 and p_2 into one parameter p by dividing p_1 on both sides of the equation and we get the relative fingerprint distance RFD_q as follows:

$$\text{RFD}_q = \frac{\left(\sum_{i=1}^{sl} \left|\text{rssi}_{oi} - \text{rssi}_{ti}\right|^q\right)^{1/q}}{(sl) + p * \ln(\text{sim})}. \quad (3)$$

4.2. Isolated Points Identification Based on Nearest Neighbors. With our proposed fingerprint distance model and a fingerprint distance threshold, we can obtain K-nearest RPs to user position. The next step is to detect and delete isolated points among these nearest RPs. An isolated point is defined as one that has very few neighboring points in physical space. The accuracy of clustering is always affected by the presence of isolated points in the data set. We design an isolated points identification algorithm that considers nearest neighbors. Assuming the training data set is $A = \{a_i\}, i = 1, \ldots, n$, nearest neighbors of a point are defined as follows: if $d(a_i, a_j) < \theta$, $1 \le i, j \le n$, then $a_i \in P_j$, where P_j is the set of nearest neighbors of point a_j and $d(a_i, a_j)$ is the distance between a_i and a_j and θ is a distance threshold.

Isolated points detection works in two steps. (1) Traverse all the points in the data set A, and record the number of nearest neighbors of every point. If a particular point holds very few neighbors or is even without any neighbor, this point is considered as an isolated point. (2) When one isolated point is identified, other points are analyzed again. If one point is the neighbor of this isolated point, the number of nearest neighbors of this point is decreased by 1. However, if the first isolated point is not identified or not all the isolated points are detected, go back to (1) for another iteration until no isolated point exists in the data set.

4.3. Cluster Neighbors Based on Semi-Supervised Affinity Propagation. After calculating the fingerprint distances between the online signal measurements and all the reference points in the fingerprint database, traditional WKNN uses the K-nearest reference points to estimate user position, where a commonly used weight c is the inverse of the RSS distance: $c = \sum_{i=1}^{K}((1/(L_{q_i} + \varepsilon)) \times c_i)$. However, due to the radio interference, K-nearest neighbor points are not always physically close to user's position, so using the K-nearest neighbors directly may lead to poor estimation. If we choose these K neighbors more carefully before calculation, a more accurate estimation may be obtained. As shown in Figure 4, the numbered points are nearest neighbors. The smaller the number is, the closer the fingerprint distance is from this RP to signal measurement collected at user's position. The actual user position is marked as solid red circle. If WKNN ($K = 3$) is directly used, the estimated position represented by the solid blue diamond is quite far away from the actual position. However, we can cluster all the neighboring points according to their physical locations and then select one cluster as the delegate based on the average fingerprint distance and the size of clusters. In this way, some "fake neighbors" such as point 2 are removed; thus, the final estimation can be improved (the solid green square in this case). This example demonstrates that how to cluster neighboring points is important. In the following, we propose a new scheme to filter out useless neighbors.

Different from K-means clustering algorithm that randomly selects K initial points, the main idea of affinity propagation clustering [27] is using preference to label RPs. The preference of RP_i, called preference(i) or $s(i, i)$, is the a priori suitability of RP_i to serve as an exemplar. The RPs with larger preference values are more likely to be selected as cluster centers. The number of clusters is influenced by the values of the input preferences. High values of the preferences will cause affinity propagation to find many clusters, while low values will lead to a small number of clusters. Preferences can be set to a shared value or customized for particular data points. Normally, the shared value could be the median of the input similarities or their minimum. In our work, the shared preference is used and represented as

$$\text{preference} = p\text{\_coefficient} * \text{median}(\text{similarities}), \quad (4)$$

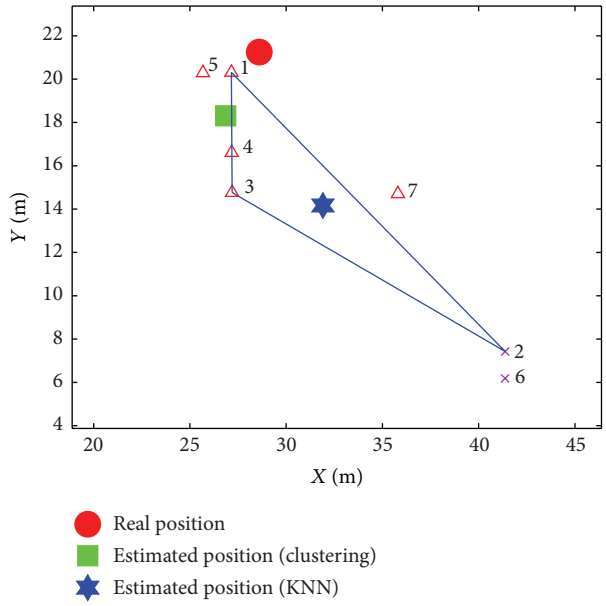

FIGURE 4: An example to demonstrate the effectiveness of clustering.

where p_coefficient is the preference parameter, which influences the number of clusters, and median(similarities) is the median of the input similarities.

The pairwise similarity $s(i, j)$ shows how well RP_j is suited to be the exemplar for RP_i. Here, we define pairwise similarity between RP_i and RP_j as $s(i, j) = -\mathrm{RFD}(i, j)$, where $\mathrm{RFD}(i, j)$ is computed using (3). The similarity matrix s is used as input for affinity propagation clustering and the output is the clustering results of reference points and the corresponding centers.

The most important factor that affects the performance of affinity propagation is the similarity matrix. In other words, if the similarity matrix can accurately capture relationship among data, affinity propagation can achieve excellent clustering. We, therefore, try to modify similarity matrix in our proposed semi-supervised affinity propagation algorithm.

If two RPs are very close to each other, they should be assigned to the same cluster. We use M to denote the set of such RPs with a *must-link* relationship: $M = \{(a_i, a_j)\}$. The basic idea of adjusting similarity matrix is that when a pair of points satisfy the constraint that $(a_i, a_j) \in M$, the two points a_i and a_j have high similarity ($s(i, j) = 0$). According to the transitivity of the must-link relationship, new pairs that satisfy this relation can be formed:

$$\begin{aligned} &(a_i, a_j) \in M \Longrightarrow s(i, j) = 0, \qquad s(j, i) = 0, \\ &(a_i, a_k) \notin M, \qquad (a_i, a_j) \in M, \\ &(a_j, a_k) \in M \Longrightarrow s(i, k) = 0, \qquad s(k, i) = 0, \\ &M = (a_i, a_k) \cup M. \end{aligned} \tag{5}$$

Affinity propagation is then used to cluster these points based on the new similarity matrix. Since the must-link relationships only change some of the similarity values, the original affinity propagation may put a pair of points into different clusters while they should belong to the same cluster: $\{(a_i, a_j)\} \in M$, but $a_i \in C_m$ and $a_j \in C_n$, where C_m and C_n are distinct clusters. To address this issue, we make further adjustments.

(i) If the sizes of the two clusters are equal, compute d_{in} (i.e., distance between a_i and cluster C_n) and d_{jm} (i.e., distance between a_j and cluster C_m). If $d_{in} < d_{jm}$, then assign a_i to C_n; otherwise, point a_j is added to C_m.

(ii) Otherwise, the cluster having more members will include the other point that did not belong to this cluster before.

4.4. Delegate Cluster Selection. After partitioning nearest RPs into several clusters, one cluster should be chosen as the delegate region. Previous work selects candidate cluster mainly based on the number of members in clusters [9] or the average fingerprint distance [12], and only when one criterion cannot pick out the candidate cluster, the other will be used. Our empirical findings show that depending too much on either criterion would result in large errors, so we propose a unified selection rule considering both the criteria. Given the maximum number of members of all the clusters is N, when the size of a cluster exceeds $N/2$ (an adjustable threshold), the cluster with smallest average fingerprint distance will be chosen. Otherwise, choose the cluster having the most members.

5. Performance Evaluation

We implement WKNN-SAP and test it in real-world environments. We compare its performance against existing approaches from several different angles, including impact of isolated points removal, semi-supervised clustering, delegate cluster selection, clustering, and fingerprint distance model. We next describe our experimental setup and results.

5.1. Experimental Setup. To evaluate the performance of WKNN-SAP, we collected Wi-Fi RSS data using a Huawei C8650 smartphone on the first floor of the third teaching building of China University of Geosciences in Wuhan. The dimension of the building is 55 × 35 meters, as depicted in Figure 5. This building has seven floors with a total of 19 Aruba APs: 5 on the first floor, 4 on the second floor, and 2 on each of the other floors. Figure 6 shows the Aruba APs deployed in classrooms. Besides, there are several different types of APs outside the building. The number of visible APs also varies dramatically at different locations (from 4 to 22 in our test).

We collected data twice, one for building the radio map and the other for testing it. The training data was collected from 160 distinct locations with 2 m spacing between each reference point. There are 30 temporal RPs and we recorded RSS values at a rate of 1 Hz for each location in the radio map. During the testing, RSS samples were collected at 77 test points with a 3 m separation. The test points were placed on and off the training points and 10 samples per location

Figure 5: The layout of the experimental testbed.

Figure 6: Aruba APs deployed in classrooms.

are recorded for testing. Figures 7 and 8 show the graphical user interface used for offline data collection and online positioning, respectively.

5.2. Experimental Results

5.2.1. Impact of Isolated Points Selection. In this experiment, we set the distance threshold value as 4 meters, which means if the distance between two points is less than 4 meters, then they are each other's nearest neighbors. Figures 9 and 10 show the positioning experiment at one test point, in which 6 initial nearest reference points are selected by using dynamical selection algorithm based on RSS threshold. As shown in Figure 9, if we do not detect the isolated points, 6 RPs are grouped into one cluster. WKNN is then used to estimate the position with the 3 nearest RPs. Figure 10 shows

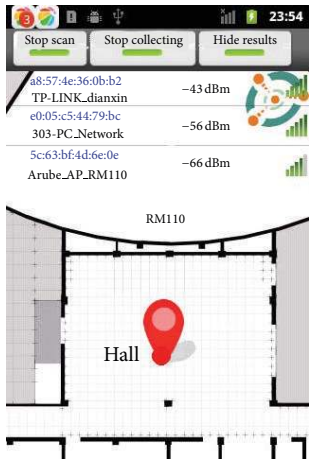

Figure 7: GUI for offline Wi-Fi fingerprint database construction.

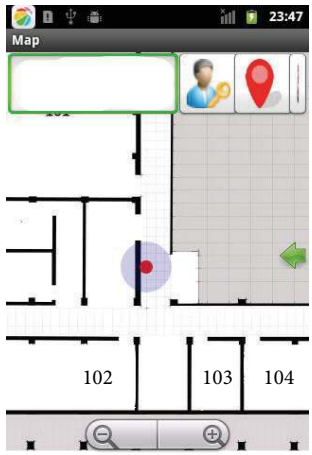

Figure 8: GUI for online position estimation.

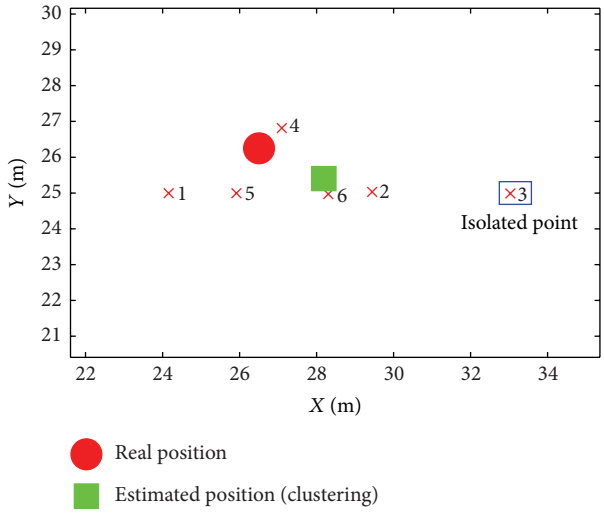

Figure 9: Result of clustering without removing isolated points.

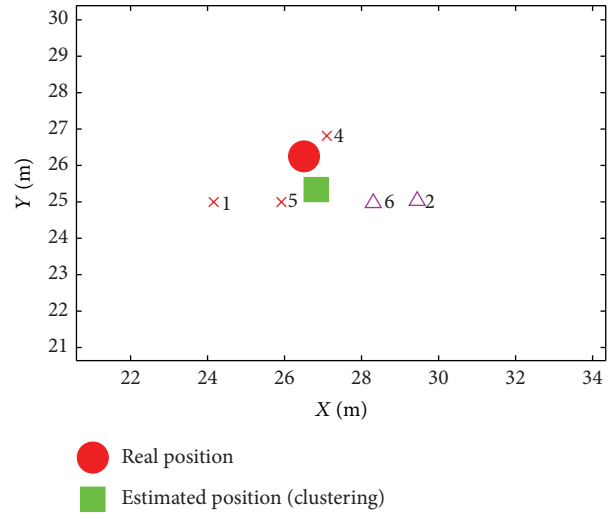

Figure 10: Result of clustering after removing isolated points.

that after isolated points (point 3) are identified and deleted, the remaining RPs are divided into two clusters. The cluster containing RP 1, 4, and 5 has the largest number of members as well as the smallest average fingerprint distance compared to the other cluster. Therefore, this cluster is selected as the delegate, and the final estimation has been improved compared to the previous result without removing isolated points.

5.2.2. Impact of Semi-Supervised Clustering Based Affinity Propagation. In this experiment, we select one test point as an example and compare the performance of affinity propagation and semi-supervised affinity propagation. We define that two points satisfy the must-link relationship, only if their distance is below 2 meters. We use color and shapes to distinguish different clusters. Figure 11 shows that when using the original affinity propagation idea, points 1, 6, and 9 belong to the same cluster which is selected as the delegate due to its smaller average fingerprint distance compared to the other clusters. However, using our proposed semi-supervised affinity propagation approach, point 1 is adjusted to another cluster as shown in Figure 12. The estimated position is much closer to the actual position due to this adjustment.

5.2.3. Impact of Delegate Cluster Selection. To demonstrate how delegate cluster selection affects the position estimation, we show two results. When there is a cluster whose size exceeds $N/2$ (where N is the maximum size of all clusters), the selection criteria should be RSS distance. Otherwise, the number of elements in clusters should be the metric. As shown in Figure 13, the RPs are partitioned into 3 clusters with a size of 1, 3, and 14, respectively. The size of cluster denoted using pink lower triangles is markedly larger than the other two clusters, so it serves as the delegate. In Figure 14, 11 RPs are partitioned into 2 clusters with a size of 4 and 7, respectively. Due to the small difference in the size of the two clusters, average fingerprint distance is then used as the

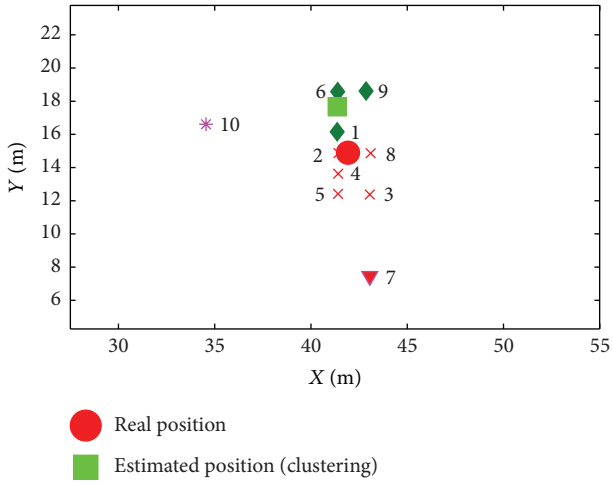

Figure 11: Result of clustering using affinity propagation.

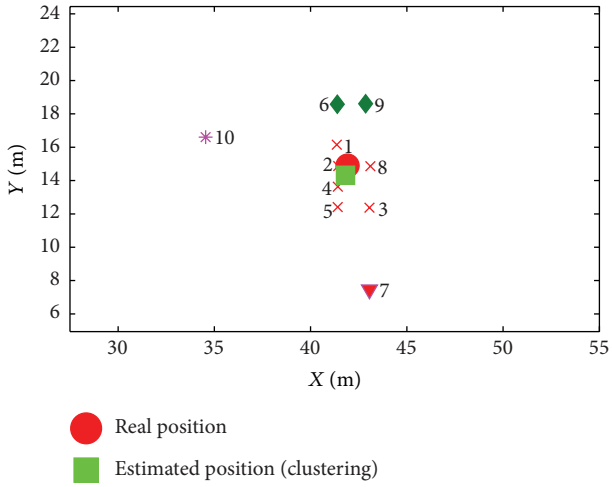

Figure 12: Result of clustering using semi-supervised affinity propagation.

selection metric. Hence, the cluster consisting of points 1, 2, and 9 is selected.

5.2.4. Impact of Clustering Techniques. In this experiment, we compare WKNN-SAP with classic K-means clustering and WKNN without clustering. Four metrics are compared, including standard deviation of error, mean error, median error, and 67% CEP (circular error probability). The CEP is defined as the radius of the circle that has its center at the true location and contains the location estimates with a probability. The location error is defined as the Euclidean distance between the estimated position and the real position of the client device. For fairness, the three methods use the same fingerprint distance estimation scheme that combines AP sets similarity and signal distance, and the parameter p is set to be 1. For WKNN-SAP, we set the values of several parameters as follows: the fingerprint distance threshold of selecting the nearest neighbors is set to 0.4, which means we will choose those RPs whose fingerprint distance to the

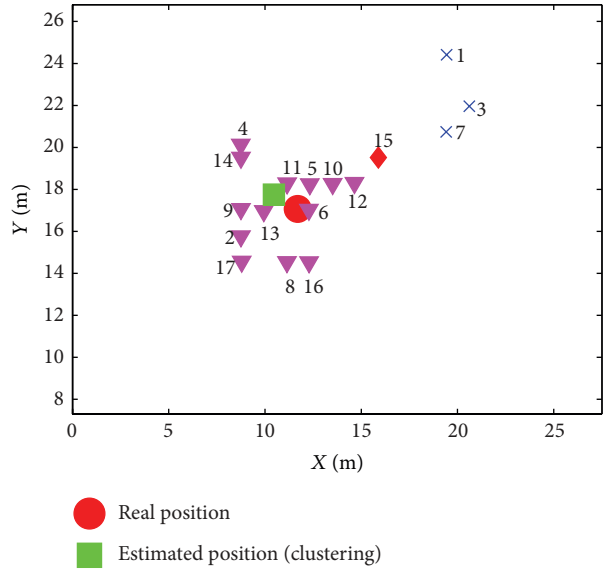

Figure 13: Delegate cluster selection based on cluster size.

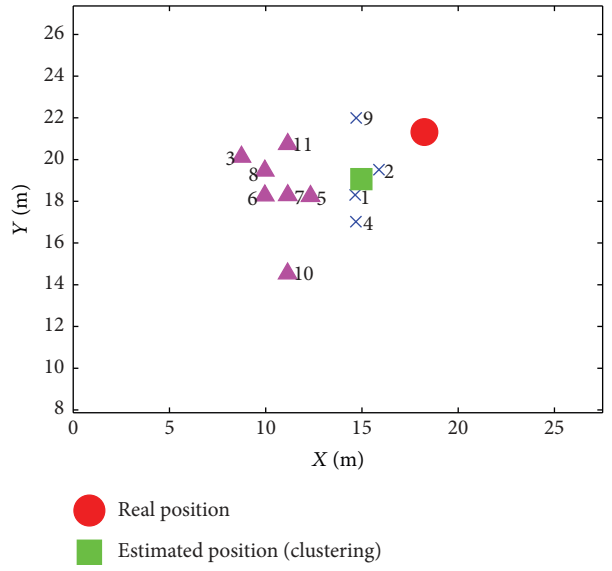

Figure 14: Delegate cluster selection based on the average fingerprint distance.

measurement of target location is at most 0.4 more than the smallest fingerprint distance. When detecting isolated points and if the physical distance between two RPs is below 4 meters, then we consider one as the nearest neighbor of the other. A pair of RPs have must-link relationship, only if the physical distance between them is less than 2 meters. As for K-means clustering, we select 13 nearest neighbors, which yields the best location accuracy in our experimental environment. WKNN-SAP and K-means use the final remaining 3 nearest RPs in the delegate cluster to calculate the position of the target. For WKKN without clustering, the 3 nearest RPs are directly used.

The number of clusters is the key factor that influences the positioning results in clustering based WKNN approach,

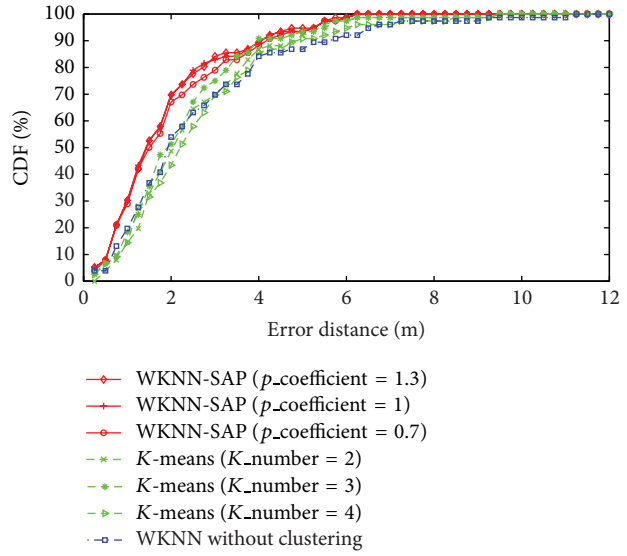

Figure 15: Comparison of positioning accuracy based on different clustering strategy.

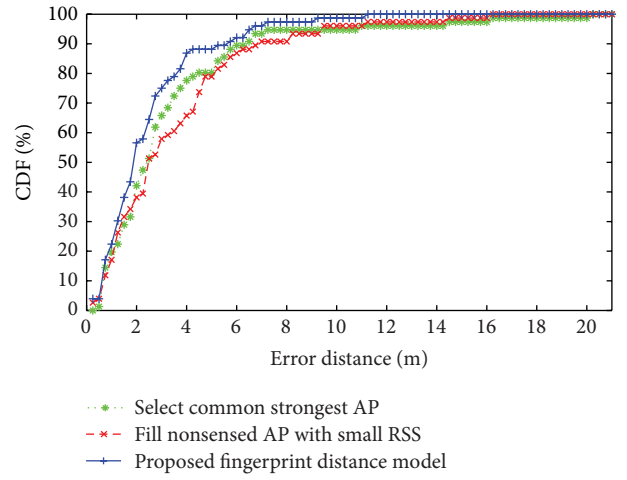

Figure 16: Comparison of positioning accuracy using different fingerprint distance models (entire environment).

so we vary the number of clusters for both K-means and semi-supervised affinity propagation. For K-means, the clustering number K_number should be assigned manually. In this case, we set the clustering number as 2, 3, and 4, respectively. As for SAP, the clustering number is determined by the value of p_coefficient, which is shown in (4). Here, p_coefficient is set to be 1.3, 1, and 0.7, respectively, which can dramatically change the clustering results, and accordingly the clustering number increases. Figure 15 and Table 1 show results of WKNN-SAP and K-means with different clustering number as well as the traditional WKNN approach. As we can see, WKNN-SAP achieves a better positioning accuracy compared to the other two methods, while K-means based approach can get the best result when the clustering number is 3. Moreover, WKNN-SAP is less affected by the clustering number compared with K-means. This is because the clustering number of semi-supervised affinity propagation is eventually adjusted based on the defined must-link relationship.

5.2.5. Impact of Fingerprint Distance Estimation Techniques. To evaluate the performance of the proposed fingerprint distance model combining AP sets similarity and average RSS distance, we compare it with traditional methods including selecting the shared APs with strongest RSS (beyond −85) and filling nonobserved AP with a small RSS value (−97). As for the proposed distance model, the parameter p is set to be 1. For fairness, three methods use the same WKNN method. K is set to be 2, and positioning result is shown in Figure 16. To consider the effect of different K values for WKNN algorithm, we test the positioning results when K is equal to 2, 4, 6, and 8, respectively. The results in Table 2 demonstrate that proposed fingerprint distance model used in WKNN-SAP provides better positioning performance compared with traditional approaches. This can be explained by the capacity of capturing the absence or appearance of APs in WKNN-SAP.

In order to further evaluate the performance of the proposed fingerprint distance model in an environment with different Wi-Fi conditions, positioning results of two subspaces of the experimental environment are compared. As shown in Figure 5, the corridors outside the classrooms are divided into three parts, which have considerable difference in sensed AP sets because of the interference of walls. The other subspace is an open area located in the center of the experimental environment, and received AP sets at different locations are basically the same. We can regard this subspace as the normal Wi-Fi environment for traditional WKNN approach. The results are shown in Figures 17 and 18, respectively. We can conclude that proposed fingerprint distance model is superior to the traditional methods when the difference of AP sets is large. However, for normal Wi-Fi environment with small AP sets difference, the proposed approach achieves similar performance to the one selecting common strongest APs, which can also be derived from (3). When sim equals 1, (3) approximately evolves to the one selecting common strongest APs. This experiment shows that the proposed method can better adapt to the indoor environment with distinct AP sets, and the greater the difference is, the more superior the proposed method is to traditional methods.

6. Conclusions

In this paper, we propose an improved Wi-Fi fingerprinting positioning algorithm called WKNN-SAP. WKNN-SAP centers around two major contributions. First, we propose a new fingerprint distance estimation model using AP sets similarity and RSS distance to deal with the observation that visible AP sets are often different in offline and online stages. Second, we design a semi-supervised affinity propagation clustering algorithm coupled with isolated points removal to gain a more reasonable clustering result and eliminate some outliers. Our evaluation results indicate that both SAP and

Table 1: Four error measures for different algorithms.

Methods	Mean	Std.	Median	67% CEP
WKNN-SAP (p_coefficient = 1.3)	1.85 m	1.39 m	1.46 m	1.93 m
WKNN-SAP (p_coefficient = 1)	1.85 m	1.43 m	1.43 m	1.93 m
WKNN-SAP (p_coefficient = 0.7)	1.95 m	1.46 m	1.5 m	1.94 m
K-means (K_number = 2)	2.48 m	1.58 m	2.07 m	2.86 m
K-means (K_number = 3)	2.28 m	1.57 m	1.94 m	2.47 m
K-means (K_number = 4)	2.58 m	1.81 m	2.22 m	2.88 m
WKNN without clustering	2.59 m	2.08 m	1.92 m	2.67 m

Table 2: Average distance errors and standard deviation of test results (entire environment).

	Select APs with largest RSS		Fill nonsensed AP with small RSS		Proposed fingerprint distance model	
	Mean	Std.	Mean	Std.	Mean	Std.
$K = 2$	3.28 m	3.40 m	3.47 m	3.06 m	2.50 m	2.11 m
$K = 4$	3.14 m	2.36 m	3.45 m	2.56 m	2.69 m	1.62 m
$K = 6$	3.20 m	2.08 m	3.75 m	2.84 m	3.02 m	1.72 m
$K = 8$	3.33 m	2.03 m	3.78 m	3.08 m	3.13 m	1.79 m

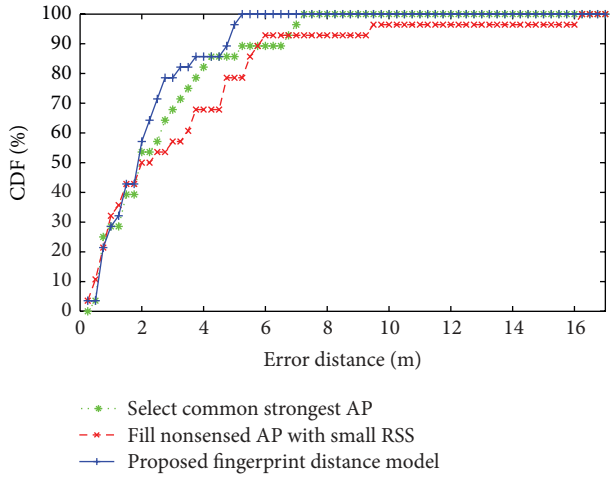

Figure 17: Comparison of positioning accuracy using different fingerprint distance models (subspace: corridor).

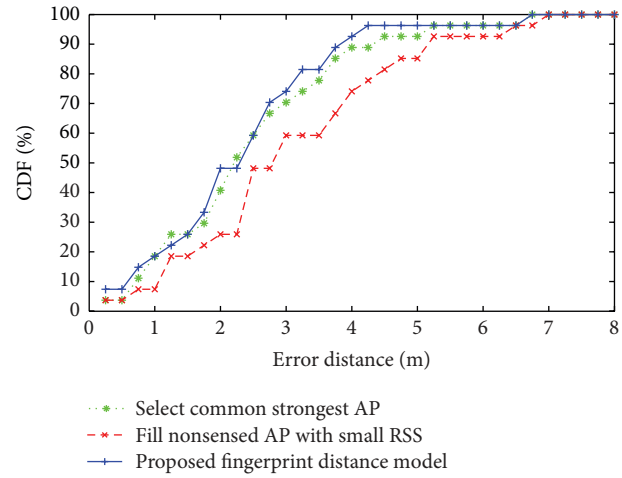

Figure 18: Comparison of positioning accuracy using different fingerprint distance models (subspace: open area).

K-means clustering can be used to improve the localization accuracy of traditional WKNN, while SAP outperforms K-means in both the accuracy improvement and stability because the former one is less affected by the clustering number. Moreover, proposed fingerprint distance model can better adapt to the indoor environment with distinct AP sets, and the greater the difference is, the more superior the proposed method is to traditional approaches.

As for future work, the difficulties of radio map building in the offline phase can be reduced by using calibration free radio map generation technique, such as crowdsourcing. Also, PDR (pedestrian dead reckoning) can be integrated to improve timeliness and accuracy. Last, location model based positioning technique can utilize spatial contexts (e.g., walls and obstacles) and connectivity between indoor entities (e.g., rooms and corridors) to constrain users' movement in indoor environment.

Conflict of Interests

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

Acknowledgment

This work is supported by the National Natural Science Foundation of China (no. 41271440).

References

[1] P. Bahl and V. N. Padmanabhan, "Radar: an in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '00)*, vol. 2, pp. 775–784, IEEE, 2000.

[2] V. Radu, L. Kriara, and M. K. Marina, "Pazl: a mobile crowdsensing based indoor wifi monitoring system," in *Proceedings of the 9th International Conference on Network and Service Management (CNSM '13)*, pp. 75–83, 2013.

[3] E. Mok and B. K. S. Cheung, "An improved neural network training algorithm for Wi-Fi fingerprinting positioning," *ISPRS International Journal of Geo-Information*, vol. 2, no. 3, pp. 854–868, 2013.

[4] Z.-A. Deng, Y.-B. Xu, and L. Ma, "Indoor positioning via nonlinear discriminative feature extraction in wireless local area network," *Computer Communications*, vol. 35, no. 6, pp. 738–747, 2012.

[5] J.-Y. Lee, C.-H. Yoon, P. Hyunjae, and J. So, "Analysis of location estimation algorithms for wifi fingerprint-based indoor localization," in *Proceedings of the 2nd International Conference on Software Technology*, vol. 19, pp. 89–92, 2013.

[6] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Received-signal-strength-based indoor positioning using compressive sensing," *IEEE Transactions on Mobile Computing*, vol. 11, no. 12, pp. 1983–1993, 2012.

[7] A. W. S. Au, *RSS-based WLAN indoor positioning and tracking system using compressive sensing and its implementation on mobile devices [M.S. thesis]*, 2010.

[8] W. Jing, H. Zhao, C. Song, X. Lin, and X. Shen, "Community detection based reference points clustering for indoor localization in WLAN," in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP '13)*, pp. 1–6, Hangzhou, China, October 2013.

[9] J. Ma, X. Li, X. Tao, and J. Lu, "Cluster filtered KNN: a WLAN-based indoor positioning scheme," in *Proceedings of the IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM '08)*, pp. 1–8, 2008.

[10] Y. Sun, Y. Xu, L. Ma, and Z. Deng, "KNN-FCM hybrid algorithm for indoor location in WLAN," in *Proceedings of the 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS '09)*, vol. 2, pp. 251–254, IEEE, Shenzhen, China, December 2009.

[11] C.-W. Lee, T.-N. Lin, S.-H. Fang, and Y.-C. Chou, "A novel clustering-based approach of indoor location fingerprinting," in *Proceedings of the 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC '13)*, pp. 3191–3196, IEEE, September 2013.

[12] B. Altintas and T. Serif, "Improving RSS-based indoor positioning algorithm via K-Means clustering," in *Proceedings of the 11th European Wireless Conference 2011—Sustainable Wireless Technologies (European Wireless)*, pp. 1–5, VDE, 2011.

[13] Z. Tian, X. Tang, M. Zhou, and Z. Tan, "Fingerprint indoor positioning algorithm based on affinity propagation clustering," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, article 272, 2013.

[14] Y. Noh, H. Yamaguchi, U. Lee, P. Vij, J. Joy, and M. Gerla, "CLIPS: infrastructure-free collaborative indoor positioning scheme for time-critical team operations," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom '13)*, pp. 172–178, IEEE, 2013.

[15] V. Radu and M. K. Marina, "HiMLoc: indoor smartphone localization via activity aware Pedestrian Dead Reckoning with selective crowdsourced WiFi fingerprinting," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN '13)*, pp. 1–10, Montbeliard, France, October 2013.

[16] L.-H. Chen, E. H.-K. Wu, M.-H. Jin, and G.-H. Chen, "Homogeneous features utilization to address the device heterogeneity problem in fingerprint localization," *IEEE Sensors Journal*, vol. 14, no. 4, pp. 998–1005, 2014.

[17] Y. Kim, H. Shin, and H. Cha, "Smartphone-based Wi-Fi pedestrian-tracking system tolerating the RSS variance problem," in *Proceedings of the 10th IEEE International Conference on Pervasive Computing and Communications (PerCom '12)*, pp. 11–19, Lugano, Switzerland, March 2012.

[18] W. B. Yu, P. Li, Z. Chen, and C. Li, "PDR-aided algorithm with WiFi fingerprint matching for indoor localization," in *Applied Mechanics and Materials*, vol. 701, pp. 989–993, Trans Tech Publications, 2015.

[19] M. I. Khan and J. Syrjarinne, "Investigating effective methods for integration of building's map with low cost inertial sensors and wifi-based positioning," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN '13)*, pp. 1–8, IEEE, Montbeliard, France, October 2013.

[20] H. Liu, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Accurate WiFi based localization for smartphones using peer assistance," *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2199–2214, 2014.

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

[22] A. W. S. Au, C. Feng, S. Valaee et al., "Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device," *IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 2050–2062, 2013.

[23] B. Shin, J. H. Lee, T. Lee, and H. K. Seok, "Enhanced weighted k-nearest neighbor algorithm for indoor wi-fi positioning systems," in *Proceedings of the 8th International Conference on Computing Technology and Information Management (ICCM '12)*, vol. 2, pp. 574–577, 2012.

[24] P. Marcus, M. Kessel, and M. Werner, "Dynamic nearest neighbors and online error estimation for SMARTPOS," *International Journal on Advances in Internet Technology*, vol. 6, no. 1-2, pp. 1–11, 2013.

[25] A. da Silva Meyer, A. A. F. Garcia, A. Pereira de Souza, and C. Lopes de Souza Jr., " Comparison of similarity coefficients used for cluster analysis with dominant markers in maize (*Zea mays* l)," *Genetics and Molecular Biology*, vol. 27, no. 1, pp. 83–91, 2004.

[26] S. Y. Hu, X. Y. Cheng, F. X. Chai, J. Gao, and G. N. Yang, "Optimal regulation model of reservoir discharge based on 1 stopt," *Journal of Hohai University: Natural Sciences*, vol. 39, no. 4, pp. 377–383, 2011.

[27] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.