Research Article

Robust Indoor Sensor Localization Using Signatures of Received Signal Strength

Yungho Leu, 1 Chi-Chung Lee, 2 and Jyun-Yu Chen 1

1 Department of Information Management, National Taiwan University of Science and Technology, Taipei 10607, Taiwan
2 Department of Information Management, Chung Hua University, Hsinchu 30012, Taiwan

Correspondence should be addressed to Yungho Leu; yhl@cs.ntust.edu.tw

Received 1 July 2013; Revised 20 October 2013; Accepted 22 October 2013

Indoor localization based on the received signal strength (RSS) values of the wireless sensors has recently received a lot of attention. However, due to the interference of other wireless devices and human activities, the RSS value varies significantly over different times. This hinders exact location prediction using RSS values. In this paper, we propose three methods to counter the adverse effect of the RSS value variation on location prediction. First, we propose to use an index location to select the best radio map, among several preconstructed radio maps, for online location prediction. Second, for an observed value of the signal strength of a sensor, we record, respectively, the distances from the sensor to the nearest location and the farthest location where the signal strength value has been observed. The minimal and maximal (min-max) distances for each signal strength value of a sensor are then used to reduce the search space in online location prediction. Third, a location-dependent received signal strength vector, called the RSS signature, is used to predict the location of a user. We have built a system, called the region-point system, based on the proposed three methods. The experimental results show that the region-point system offers less mean position error compared to the existing methods, namely, RADAR, TREE, and CaDet. Furthermore, the index location method correctly selects the best radio map for online location prediction, and the min-max distance method promotes the prediction accuracy of RADAR by restricting the search space of RADAR in location prediction.

1. Introduction

Indoor localization is important for many real-life applications. For example, it gives the location context of a context-aware system that provides proper settings of the system based on the location, activity, and physiology of the user and the environmental context information [1]. Recently, indoor navigation applications, which require an exact indoor location, are becoming a very popular research area [2]. Due to the increasing need for indoor localization, many indoor localization techniques have been proposed. An indoor localization method can be categorized as a range-based or a range-free method [3]. While point-to-point distance information is required for a range-based method, it is not required for a range-free method. The techniques for estimating the distance between two communication nodes include the time of arrival (TOA) [4], time difference of arrival (TDOA) [5], and the angle of arrival (AOA) [6]. The TOA technique uses the radio signal propagation time to estimate the distance. The TDOA technique utilizes two radio signals with different propagation speeds and estimates the distance between the two communication nodes by measuring the difference between the arrival times of the two signals. Unlike TOA and TDOA, AOA technique measures the angle at which a signal arrives. It can be used to complement TDOA or TOA in location calculation [3]. Indoor localization methods that use range information usually achieve high accuracy in location estimation. For example, the Cricket [7] indoor localization system of MIT reported the error of 1 to 3 centimeters in position estimation. Despite being accurate in location prediction, the range-based localization techniques require large scale deployment and costly devices.
The range-free location prediction techniques have received a lot of attention recently. The well-known range-free location prediction methods include RADAR [8] and the probability-based methods [9–12]. RADAR is developed by Microsoft. In RADAR, for a predefined set of training locations, the received signal strength (RSS) values from several IEEE 802.11 access points are recorded in a database, called the radio map. To estimate the position of a user, the RSS values from the access points are collected at the location of the user. Afterwards, RADAR performs pattern matching of the collected RSS values against the RSS values in the radio map to find a fixed number of locations with the most similar RSS values against those of the user. Finally, the positions with the most similar RSS values are averaged to give the estimated position of the user. The probability-based methods also use the RSS values for location prediction. However, instead of a fixed number of locations for prediction, the probability-based methods use the Bayes theorem to predict the location of the user by finding the location where the collected RSS values of the user can be observed with the highest probability. In [13], the authors proposed to learn, at time $t_0$, a set of equations to fit the RSS values of a location using the RSS values of a set of reference points. With this method, the RSS value pattern of a specific location at a later time $t_1$ can be calculated by using the RSS value patterns of the reference locations at time $t_1$. Therefore, the effort to collect the RSS values at the offline training phase can be significantly reduced. However, in an environment where the RSS values observed at a location vary over times, the regression equations learned at time $t_0$ may not properly reflect the relationship between the RSS values of the location and those of the reference points. This may result in poor prediction accuracy. In [14], the authors proposed a method, called CaDet, which uses multiple decision trees for location prediction. They first divide the training dataset into several clusters and build a decision tree for each cluster. To predict the user's location, the RSS values of the user are compared against the means of the RSS values of each cluster center to find the cluster with the least distance from the RSS values of the user for prediction. Finally, the decision tree of the selected cluster is used to predict the location of the user. Besides using the values of the received signal strength, in [15], the authors proposed to use the signal quality indicator (LQI) values for location prediction. They modeled the location prediction problem as a classification problem and used a neural network model to solve the problem. However, their method is more suitable for finding a coarse position for a user, such as in the kitchen or in the living room.

The most difficult problem for the range-free methods in location prediction is that the offline constructed radio map may not be suitable for online location prediction. The variation of the received signal strength values may outdate the radio map when an online location prediction is required. In this paper, we propose three methods to counter the adverse effect of the variation of the received signal strength values on location prediction. First, we propose to construct several radio maps over different nonoverlapping time intervals and use an index location to select the best radio map for online location prediction. Second, for an RSS value of a sensor observed in the location prediction area, we propose to record the minimal and the maximal (min-max) distance from the sensor to the locations where the same RSS value has been observed. The min-max distance information is used to reduce the number of locations required to be searched for in online location prediction. Thirdly, we propose to use a location-dependent received signal strength vector, called the RSS location signature, for pattern matching in online location prediction. A system, called the region-point system, which implemented the three proposed methods, has been implemented. The experimental results show that the region-point system offers less position prediction error compared to the existing methods, including RADAR, TREE, and CaDet. Furthermore, the experiment also shows that the index location method correctly selects the best radio map for location prediction, and the min-max distance method significantly reduces the position prediction error of RADAR. The rest of this paper is organized as follows. In Section 2, we describe the phenomenon of the variation of the received signal strength values. In Section 3, we present the details of the region-point localization system. In Section 4, we present the experimental results. In Section 5, we give a discussion of the experimental result. Finally, in Section 6, we give the conclusion of this paper.

2. Variation of the Received Signal Strength

The most challenging problem for location prediction using RSS values is that the RSS values of a sensor observed at a fixed location change over different times [12–14, 16]. In this paper, we use the MPR2400CA sensor, a ZigBee-based sensor called Mote, to show the phenomenon of RSS value variation over different times. The Mote uses the RF frequency band of 2.4–2.4835 GHz for communication. The 2.4 GHz band frequency is a very noisy band since the wireless local area network (802.11b and 802.11g), the Bluetooth personal area network (802.15.1), and the industrial, scientific, and medical (ISM) devices are all using this unlicensed frequency band. The interference from other networks or devices forces the received signal strength value of a sensor at a fixed location to vary significantly over different times. Furthermore, the unpredictable people moving and door opening or closing cause the changes in the reflection, absorption, diffraction and scattering of the RSS values amplify the variation of the RSS values in an indoor environment [13].

To show the variation of RSS values over different times, we collected 500 RSS values from a fixed location which is 84.85 centimeters away from a ZigBee sensor for a time interval of 4 consecutive hours. Figure 1(a) shows the distribution of the RSS values from 10 a.m. to 2 p.m., while Figure 1(b) shows the distribution of the RSS values from 3 p.m. to 7 p.m. These figures show that not only the shapes of the distributions but also the averages of the signal strength values in different time intervals are different. The variation of the RSS values over different times implies that the RSS values collected at the offline training phase may not be good for online location prediction [13].
3. The Region-Point Location Prediction System

In this section, we present the implementation of a robust sensor prediction system which considers the variation of the RSS values.

3.1. The Components and Layout of the System. The components and the layout of the system are shown in Figure 2. The system is implemented in a classroom measuring 9.3 m \( \times \) 13 m. There are three rows of tables with a desktop on each table. There are two doors and one electronic podium in the room. We placed ten Mote sensors, denoted by \( M \) in Figure 2, as the reference sensors.

A sensor, denoted by \( U \), is mounted on a moving cart for testing the location prediction algorithm. To predict the location of a user, the sensor \( U \) (stands for the user) broadcasts a packet to the reference sensors. Upon receiving the packet from \( U \), a reference sensor records the RSS value of its received packet, stores the RSS value in a new packet, and then sends the new packet to the location prediction computer, denoted by \( C \) in Figure 2, to predict the location of \( U \).

3.2. Architecture of the System. Figure 3 shows the architecture of the region-point location prediction system. It contains the offline training phase and the online location prediction phase. The offline training phase contains the following steps.

1. For different time periods, collect the RSS values of the reference sensors for each training location and store the RSS values in the radio maps.
2. Create a min-max distance table for each radio map.
3. Find the index location for radio map selection.
The online location prediction phase contains the following steps.

(1) Collect a number of RSS values at the index location.
(2) Select the best radio map for online location prediction.
(3) At the location that needs to be localized, collect the RSS values from the reference sensors; find the region for location prediction using the RSS values and the min-max distance table.
(4) Find the position of the predicted location in the selected region using the RSS signature of the collected RSS values.

The details of each step are discussed in the following.

### 3.3. Radio Map Construction.
During the offline phase, we choose a number of different time intervals and construct a radio map for each time interval. A set of training locations denoted by \( L = \{l_1, l_2, \ldots, l_n\} \) is chosen for collecting the RSS values. Each location \( l_i \) is associated with a coordinate \((x_i, y_i)\). Assume that there are \( k \) reference sensors, denoted by \( R = \{m_1, m_2, \ldots, m_k\} \), where \( m_i \) denotes sensor \( i \). Then, each RSS value is stored in a vector \( o_j \) of \( k \) elements, denoted by \( o_j = (ss_{s1}, ss_{s2}, \ldots, ss_{sk}) \), where \( ss_i \) is the RSS value of the packet received from reference sensor \( i \). Table 1 shows an example of the radio map.

### 3.4. The Min-Max Distance Table.
Due to the variation of the RSS values, a reference sensor may observe different RSS
values from the localization sensor $U$ when $U$ is fixed at a specific location. Similarly, the same RSS value observed by a reference sensor may be from different packets transmitted by $U$ at different locations. For example, the RSS value $-29$ dbm of sensor $ss_1$ in Table 1 is observed when $U$ is located at location $(1, 1)$ and location $(1, 7)$. During the offline training phase, for each observed received signal strength value $ss_i$ of reference sensor $m_i$, we keep track of the minimum and the maximum distances from sensor $m_i$ to sensor $U$. Table 2 shows an example of the min-max distance table.

The min-max distance table is used to reduce the search region of locations during the online localization prediction phase.

**3.5. The Index Location.** As noted in [13], the radio map constructed in the training phase may not be suitable for online location prediction. We propose to use several radio maps for location prediction. Assume that the set of time intervals is $T = \{t_1, t_2, ..., t_N\}$. Let $M_t$ denote the radio map constructed at time interval $t$, $t \in T$.

Let $\overline{ss}_{t,j} = (\overline{ss}_{t,j,1}, \overline{ss}_{t,j,2}, ..., \overline{ss}_{t,j,k})$ denote the average RSS vector at location $l_j$ in $M_t$, where $\overline{ss}_{t,j,m} 1 \leq j \leq k$, is the average of the received signal strength values of sensor $j$ at location $l_j$. Then, for each location $l_j$, we calculate $D_j$, the summation of the Manhattan distances between every pair of average RSS vectors at location $l_j$, where each vector belongs to a different radio map. That is,

$$D_j = \sum_{t_j \in T, t_j \neq t_j} \sum_{m=1}^k |\overline{ss}_{t_j,m} - \overline{ss}_{t_j,m} |.$$

The index location $l_j$ is the location which maximizes $D_j$.

The index location $l_j$ is the location which maximizes $D_j$. That is, $D_j \geq D_{j'}, j = 1, \ldots, n$.

During the online localization phase, we collect five received signal strength vectors at the index location. Take the average of the signal strength vectors, and then use the average RSS vector to select the best radio map for online location prediction. Assume that the average RSS vector is $\overline{ss} = (\overline{ss}_{1}, \overline{ss}_{2}, ..., \overline{ss}_{k})$. Then, the radio map $M_{t^*}$ is found by using the following equation:

$$t^* = \arg\min_{t=1,\ldots, T} \sum_{m=1}^k |\overline{ss}_{t,m} - \overline{ss}_{m}|.$$

| Table 1: An example of the radio map of the system. |
|---|---|---|---|---|---|---|---|---|---|---|
| locx | locy | $ss_1$ | $ss_2$ | $ss_3$ | $ss_4$ | $ss_5$ | $ss_6$ | $ss_7$ | $ss_8$ | $ss_{10}$ |
| 1 | 1 | -9 | -33 | -30 | -43 | -40 | -32 | -32 | -30 | -33 | -31 |
| 1 | 1 | -9 | -33 | -30 | -43 | -40 | -32 | -32 | -30 | -33 | -31 |
| 1 | 1 | -9 | -33 | -30 | -43 | -40 | -32 | -32 | -30 | -33 | -31 |
| 1 | 1 | -9 | -33 | -30 | -43 | -40 | -32 | -32 | -30 | -33 | -31 |
| 1 | 1 | -9 | -33 | -30 | -43 | -40 | -32 | -32 | -30 | -33 | -31 |
| 1 | 1 | -9 | -33 | -30 | -43 | -40 | -32 | -32 | -30 | -33 | -31 |
| 1 | 1 | -9 | -33 | -30 | -43 | -40 | -32 | -32 | -30 | -33 | -31 |
| 1 | 1 | -9 | -33 | -30 | -43 | -40 | -32 | -32 | -30 | -33 | -31 |
| 1 | 1 | -9 | -33 | -30 | -43 | -40 | -32 | -32 | -30 | -33 | -31 |

| Table 2: The min-max distance table. |
|---|---|---|---|
| Mote | ss | Min dist. | Max dist. |
| 1 | -35 | 5.099019514 | 23.60084744 |
| 1 | -39 | 5.099019514 | 23.60084744 |
| 1 | -38 | 5.099019514 | 23.60084744 |
| 1 | -40 | 5.099019514 | 23.60084744 |
| 1 | -41 | 5.099019514 | 23.60084744 |
| 1 | -44 | 5.099019514 | 23.60084744 |
| 1 | -43 | 5.099019514 | 23.60084744 |
| 1 | -42 | 5.099019514 | 23.60084744 |
| 1 | -37 | 5.099019514 | 23.60084744 |
| 1 | -36 | 5.099019514 | 23.60084744 |
| 1 | -20 | 5.099019514 | 7.071067812 |

...
That is, we choose the radio map which minimizes the Manhattan distance against the online average RSS vector at location $l_i$ for online location prediction.

### 3.6 The RSS Location Signature

While the probability-based methods use the original radio map, as shown in Table 1, for location prediction, we propose to use a refined variant of the RSS vectors, called RSS signatures, for location prediction. An RSS signature of a location is a distinctive RSS representative for the location. Let $P(ss_{j,r} = k)$ denote the probability that the RSS value $k$ of sensor $r$ is observed at location $l_j$. Probability $P(ss_{j,r} = k)$ is defined in the following equation:

$$P(ss_{j,r} = k) = \frac{fr(ss_{j,r} = k)}{\sum_{z=1}^{n} fr(ss_{z,r} = k)}, \quad (3)$$

where $fr(ss_{j,r} = k)$ denotes the number of observations (frequency) of RSS value $k$ of sensor $r$ at location $l_j$. Note that, since the RSS value $k$ of sensor $r$ may be observed at different locations, $P(ss_{j,r} = k)$ is the location distribution of the RSS value $k$ of sensor $r$ at location $l_j$. We then define the discernability factor $f(ss_{j,r} = k)$ of an RSS value $k$ of sensor $r$ by the following equation:

$$f(ss_{j,r} = k) = 1 - \frac{(-1)}{\log n} \sum_{j=1}^{n} p(ss_{j,r} = k) \log (p(ss_{j,r} = k)). \quad (4)$$

The third and fourth terms of (4) together represent the entropy of location distribution of RSS value $k$ of sensor $r$ over different locations. The second term is used to normalize the entropy value to the interval $(0, 1)$. The maximal value of the entropy function occurs when value $k$ of sensor $r$ is evenly distributed over $n$ locations. In this case, value $k$ of sensor $r$ does not have any discernability to distinguish between different locations. The higher the skewness of the location distribution is, the smaller the normalized entropy is. The normalized entropy value equals zero if the RSS value $k$ of sensor $r$ can only be observed at a single location. Therefore, the discernability factor of an RSS value $k$ of a sensor $r$ is a measure of the ability to distinguish between different locations in the system. Note that $n$ in (4) is the number of locations in the system.

Having defined the discernability factor, we define the weight of an RSS value $k$ of sensor $r$ at location $l_j$ by the following equation:

$$w(ss_{j,r} = k) = \frac{fr(ss_{j,r} = k)}{m_j} \cdot f(ss_{j,r} = k), \quad (5)$$

where $m_j$ is the total number of RSS samples, that is, the number of RSS vectors, collected at location $l_j$. Equation (5) shows that the weight of RSS value $k$ of sensor $r$ at location $l_j$ is the product of the discernability factor of RSS value $k$ and the probability of observing $k$ at location $l_j$.

For location $l_i$, we define its location signature at sensor $r$ to be the RSS value received from sensor $r$ whose weight is greater than that of any other RSS value received by $U$ at location $l_i$ from sensor $r$. To obtain the RSS location signature vector for location $l_i$, we find the RSS location signature value of each sensor $r$, $1 \leq r \leq k$. Table 3 shows an example of the table of RSS location signatures for the radio map in Table 1. Table 4 shows the weights of the corresponding RSS location signatures in Table 3.

### 3.7 The Online Location Prediction Phase

During the online localization phase, we first collect several RSS samples at the index location. Then, we compute the average RSS value vector of the collected samples and use it to select the best radio map for online location prediction.

To find the position of the user, we collect an RSS value vector, denoted by $O^*$, at the designated location of the user. Then, for each component $ss_i$ of vector $O^*$, we refer to the min-max distance table to find the minimum and the maximum distances from sensor $i$ for this signal strength value. Figure 4 shows the minimum and maximum distances from three sensors for an example.

From the circles with radii of minimum and maximum distances from their corresponding sensors, we can find the intersection points, that is, $P1, P2, P3, P4, P5, P6$, and $P7$, as shown in Figure 4. Then, we find the bounding box of the intersection points as the region within which the position (coordinates) of the user is to be found.

Finally, we find the training locations within the bounding box and use these locations to predict the position of the user. The pattern matching on RSS location signatures is used to find the position of the user. For each location $l_i$ in the bounding box, we find the top-$p$ weighted RSS value components of its RSS location signature. Then, we compute the Euclidean distance between the vector of the top-$p$ RSS value components of location $l_i$ and the vector of the corresponding components of $O^*$. Let us denote the top-$p$ weighted RSS value components of the RSS location signature of $l_i$ by $V_{i} = (f_{i1}, f_{i2}, \ldots, f_{ip})$ and the corresponding components of $O^*$ by $O' = (ss_{11}, ss_{21}, \ldots, ss_{p1})$. Then, the Euclidean distance between $V'$ and $O'$ is calculated according to the following equation:

$$\text{distance}(V', O') = \sum_{z=1}^{p} (f'_{z} - ss'_{z})^2. \quad (6)$$

After computing the distances between $O^*$ and the RSS location signatures of the training locations in the bounding box, the position of the user is predicted to be the position of the location with the smallest Euclidean distance of its top-$p$ weighted RSS value components against $O'$.

### 4. Experiments

To show the performance of the region-point system, we perform several experiments on location prediction in the classroom. In this section, we present the experiments and the results.
Table 3: The RSS location signature table.

| locx | locy | s1  | s2  | s3  | s4  | s5  | s6  | s7  | s8  | s9  | s10 |
|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1    | 1    | −9  | −33 | −30 | −43 | −37 | −29 | −29 | −25 | −33 | −27 |
| 1    | 7    | −25 | −21 | −34 | −35 | −33 | −27 | −33 | −37 | −43 | −29 |
| 1    | 13   | −29 | −25 | −19 | −29 | −33 | −27 | −35 | −37 | −47 | −36 |
| 1    | 19   | −35 | −25 | −19 | −13 | −34 | −30 | −24 | −29 | −33 | −36 |
| 5    | 19   | −35 | −19 | −19 | −17 | −23 | −25 | −29 | −25 | −39 | −27 |
| 5    | 13   | −35 | −20 | −9  | −33 | −39 | −32 | −29 | −26 | −44 | −25 |
| 5    | 7    | −47 | −14 | −17 | −35 | −34 | −25 | −24 | −26 | −25 | −29 |
| 5    | 1    | −21 | −17 | −23 | −42 | −39 | −23 | −35 | −26 | −13 | −13 |
| 10   | 1    | −41 | −31 | −30 | −35 | −35 | −46 | −24 | −19 | −31 | −13 |
| 10   | 7    | −31 | −35 | −35 | −46 | −29 | −14 | −15 | −21 | −30 | −32 |
| 10   | 13   | −49 | −27 | −22 | −33 | −25 | −13 | −19 | −21 | −36 | −30 |
| 10   | 19   | −47 | −35 | −22 | −23 | −21 | −17 | −22 | −33 | −35 | −37 |
| 14   | 19   | −48 | −34 | −29 | −29 | −11 | −24 | −38 | −25 | −39 | −38 |
| 14   | 13   | −43 | −27 | −33 | −50 | −22 | −21 | −25 | −24 | −29 | −31 |
| 14   | 7    | −31 | −26 | −43 | −37 | −29 | −29 | −22 | −17 | −28 | −32 |
| 14   | 1    | −25 | −27 | −46 | −35 | −27 | −39 | −19 | −8  | −19 | −38 |

Figure 4: Min-max distance and bounding box.

4.1. The Experimental Environment. As shown in Figure 2, we implement the localization system in a classroom. Figure 5 shows the layout of the reference sensors and the locations where the training samples are taken. The ground of the classroom is decorated with tiles. The tile’s dimension is 60 centimeters on each side. We set the origin of the coordinate system at the top left corner of Figure 5. Ten reference sensors, denoted by large circles in Figure 5, are evenly located in the classroom. The training locations are denoted by small circles. Totally, we have 16 training locations. The coordinates of two examples of training locations are (1, 1) and (1, 7). Note that, since each grid in Figure 5 represents one tile on the floor, the Euclidean distance between any two locations in Figure 5 can be calculated by multiplying their Euclidean distance by 0.6 meters. To build the radio maps, we collect 500 RSS value samples from each of the 16 training locations over a consecutive 4-hour time interval of the day. Three radio maps, denoted by \( M_1 \), \( M_2 \), and \( M_3 \), are constructed for the experiment.

For comparison purpose, we implement the RADAR method and a decision tree method called TREE and the CaDet method. For RADAR, the RSS vectors of different training samples from the same location are averaged. As a result, each location is associated with only one average RSS vector. To predict the coordinates of a test sample, three neighbors whose RSS vectors are among the top 3 shortest distances from the test sample are retrieved from the radio map and their corresponding coordinates are averaged to give the predicted coordinates of the test sample. To examine the effect of the search space reduction on RADAR, we revised
Table 4: The weight table.

| locx | locy | \( s_1 \) | \( s_2 \) | \( s_3 \) | \( s_4 \) | \( s_5 \) | \( s_6 \) | \( s_7 \) | \( s_8 \) | \( s_9 \) |
|------|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 1    | 1    | 0.3640   | 0.3016   | 0.0736   | 0.1075   | 0.0913   | 0.1029   | 0.0906   | 0.1324   | 0.0766   |
| 1    | 7    | 0.1376   | 0.1832   | 0.0733   | 0.1479   | 0.0987   | 0.0535   | 0.0709   | 0.0947   | 0.0822   | 0.1346   |
| 1    | 13   | 0.0862   | 0.0697   | 0.1518   | 0.1566   | 0.0587   | 0.0946   | 0.0864   | 0.0782   | 0.0708   | 0.1242   |
| 1    | 19   | 0.1026   | 0.0782   | 0.1773   | 0.2940   | 0.0703   | 0.1022   | 0.0953   | 0.1014   | 0.1317   | 0.1576   |
| 5    | 19   | 0.1823   | 0.2862   | 0.2723   | 0.4140   | 0.1325   | 0.1147   | 0.0686   | 0.3700   | 0.0813   | 0.2053   |
| 5    | 13   | 0.1411   | 0.1504   | 0.2700   | 0.1944   | 0.0851   | 0.1423   | 0.0668   | 0.0895   | 0.0941   | 0.2057   |
| 5    | 7    | 0.0630   | 0.5700   | 0.0965   | 0.0986   | 0.1421   | 0.1159   | 0.0823   | 0.1278   | 0.4695   | 0.0990   |
| 5    | 1    | 0.7471   | 0.1425   | 0.3014   | 0.0724   | 0.0779   | 0.1030   | 0.0803   | 0.2471   | 0.3600   | 0.3980   |
| 10   | 1    | 0.0646   | 0.0864   | 0.0895   | 0.1881   | 0.0939   | 0.2180   | 0.1038   | 0.0979   | 0.3600   | 0.1663   |
| 10   | 7    | 0.2262   | 0.0787   | 0.0903   | 0.0630   | 0.1668   | 0.1865   | 0.3520   | 0.1376   | 0.1817   | 0.1327   |
| 10   | 13   | 0.0521   | 0.1717   | 0.0967   | 0.0934   | 0.0992   | 0.1764   | 0.2365   | 0.5402   | 0.1206   | 0.1699   |
| 10   | 19   | 0.1322   | 0.1129   | 0.1063   | 0.2117   | 0.3700   | 0.6320   | 0.1667   | 0.1122   | 0.1170   | 0.1287   |
| 14   | 19   | 0.1255   | 0.1264   | 0.0743   | 0.4479   | 0.3220   | 0.2139   | 0.0947   | 0.2848   | 0.0813   | 0.1493   |
| 14   | 13   | 0.0597   | 0.1727   | 0.0806   | 0.1265   | 0.1485   | 0.2720   | 0.1916   | 0.1437   | 0.1105   | 0.1156   |
| 14   | 7    | 0.0929   | 0.0722   | 0.0825   | 0.1094   | 0.3351   | 0.1344   | 0.0863   | 0.2180   | 0.2163   | 0.1156   |
| 14   | 1    | 0.1480   | 0.2418   | 0.0610   | 0.1380   | 0.0970   | 0.0755   | 0.2427   | 0.2360   | 0.2991   | 0.1104   |

The positions of the sensors and the training locations are shown in Figure 5.

The RADAR method by using the min-max distance table to confine the search region of RADAR. We call the revised RADAR method ReRADAR in the experiment.

For the TREE method, a decision tree is constructed for every radio map. The decision tree is then used to predict the coordinates of a test sample. Note that we use the CART decision tree model in R to construct the decision trees.

For CaDet method, we first use the K-means method in R to divide the training samples into three clusters based on their RSS vectors. A CART decision tree is then built for each cluster. To predict the coordinates a test sample, we compare the RSS vector of the test sample against the cluster mean of each cluster and select the decision tree whose corresponding cluster center has the shortest distance against the test sample to predict the location of the test sample.

4.2. The Experimental Results. Figure 6 shows an execution of the radio map selection algorithm. It shows that the location (14, 13) is chosen as the index location since it has the largest variance on the RSS values of different radio maps. Furthermore, based on the index location, radio map \( M_3 \) is selected as the best radio map for the ongoing experiment.

To conduct the experiment, we consecutively collect 20 RSS value samples at each of the 16 testing locations. Totally, there are 320 test samples. Figure 7 shows the four executions of the region point with different lengths of the RSS location signatures.

Figure 7 shows that the longer the RSS location signature, the higher the prediction accuracy. However, the length effect decreases as the length becomes longer. This is evidenced in Figure 7, where the accumulated errors for region point with 9 and 10 components, respectively, are almost the same.
Figure 7: Performance of region point with different lengths of the RSS location signature.

Figure 8: Accumulated errors for different methods based on radio map $M_1$.

Figure 8 shows the accumulated errors for RADAR, ReRADAR, TREE, CaDet, and region point. It shows that the region point method has the smallest accumulated error compared with RADAR, ReRADAR, TREE, and CaDet. It also shows that the accumulated error for ReRADAR is much less than that of RADAR. The mean errors for the 320 test samples are 0.681, 1.29, 2.11, 2.87, and 2.99 for region point, ReRADAR, RADAR, CaDet, and TREE, respectively. Figure 8 shows the fact that the search region restriction using the min-max distance table effectively reduces the prediction error of RADAR.

Figure 9 shows the accumulated errors for different methods based on radio map $M_2$. It again shows that the accumulated error for ReRADAR is much less than that of RADAR. The mean errors are 0.556, 0.822, 1.98, 1.18, and 1.30 meters for region point, ReRADAR, RADAR, CaDet, and TREE, respectively. Note that the errors for different methods based on $M_2$ are all less than their corresponding errors in radio map $M_1$ and $M_2$, respectively. This shows that the index location method correctly selects the best radio map for online prediction. It is also noted, from Figures 8, 9, and 10, that, although clustering before constructing decision trees helps to promote the prediction accuracy of CaDet, the improvement is not significant.

5. Discussion

The fact that the TREE and CaDet methods do not perform well in our experimental environment needs to be carefully studied. To do so we show the decision tree built by CART based on radio map $M_3$ in Figure 11. Note that $M_3$ contains 8000 samples with 500 samples for each of the 16 locations.
The label at the terminal node denotes the class and the number of samples in the training dataset that are classified as this label. For example, the terminal node 1 has label 1,13, which denotes location (1,13), and there are 173 samples being classified as location (1,13). The classification accuracy for the training dataset of this decision tree is 93.3 percent. Table 5 shows the confusion table for predicting the 320 testing samples based on the decision tree of Figure 11. It shows that 234 out of 320 samples are correctly classified, that is, correctly predicted. For comparison, we show the histogram of prediction errors for both TREE and region point in Figure 12. It shows that the region point has more samples correctly classified than the TREE does, that is, 283 versus 234. Furthermore, for the misclassified samples, the region point tends to classify them to their nearby locations. These two observations account for a less mean prediction error in region point than those of the TREE and CaDet.

### 6. Conclusions

In this paper, we present the implementation of a robust indoor localization system using a wireless sensor network. In this system, we propose three methods to counter the adverse effect of variation on the received signal strength values on location prediction. First, we propose to use an index location to select the best radio map for online location prediction. Second, we propose to use the min-max distance table to confine the search region for online location prediction. Finally, we propose to use the RSS location signature for pattern matching in online location prediction. The experimental results showed that the index location method correctly selects the best radio map for online location prediction. It also showed that the min-max distance table method effectively reduces the prediction error of RADAR, and the region point system offers a higher
Figure 11: The CART decision tree for radio map $M_3$.

Figure 12: Histograms of prediction errors for TREE and region point.
prediction accuracy than those of the RADAR, TREE, and CaDet.

Conflict of Interests

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

References

[1] A. Krause, A. Smailagic, and D. P. Siewiorek, “Context-aware mobile computing: learning context-dependent personal preferences from a wearable sensor array,” IEEE Transactions on Mobile Computing, vol. 5, no. 2, pp. 113–127, 2006.

[2] 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, 2012.

[3] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom ’03), pp. 81–95, San Diego, Calif, USA, September 2003.

[4] S. Capkun, M. Hamdi, and J. Hubaux, “GPS-free positioning in mobile ad-hoc networks,” in Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS ’01), pp. 3481–3490, January 2001.

[5] A. Savvides, C.-C. Han, and M. B. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” in Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM ’01), pp. 166–179, Rome, Italy, July 2001.

[6] D. Niculescu and B. Nath, “Ad Hoc Positioning System (APS) using AoA,” in Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’03), vol. 3, pp. 1734–1743, San Francisco, Calif, USA, March-April 2003.

[7] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “Cricket location-support system,” in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM ’00), pp. 32–43, Boston, Mass, USA, August 2000.

[8] 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), pp. 775–784, March 2000.

[9] X. Chai and Q. Yang, “Reducing the calibration effort for probabilistic indoor location estimation,” IEEE Transactions on Mobile Computing, vol. 6, no. 6, pp. 649–662, 2007.

[10] M. A. Youssef, A. Agrawala, and A. U. Shankar, “WLAN location determination via clustering and probability distributions,” in Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom ’03), pp. 143–150, March 2003.

[11] Q. Yao, F.-Y. Wang, H. Gao, K. Wang, and H. Zhao, “Location estimation in Zigbee network based on fingerprinting,” in Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES ’07), December 2007.

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

[13] J. Yin, Q. Yang, and L. M. Ni, “Learning adaptive temporal radio maps for signal-strength-based location estimation,” IEEE Transactions on Mobile Computing, vol. 7, no. 7, pp. 869–883, 2008.

[14] Y. Chen, Q. Yan, J. Yin, and X. Chai, “Power-efficient access-point selection for indoor location estimation,” IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 7, pp. 877–888, 2006.

[15] Y.-G. Ha, A.-C. Eun, and Y.-C. Byun, “Efficient sensor localization for indoor environments using classification of link quality patterns,” International Journal of Distributed Sensor Networks, vol. 2013, Article ID 701259, 6 pages, 2013.

[16] S.-P. Kuo, B.-J. Wu, W.-C. Peng, and Y.-C. Tseng, “Cluster-enhanced techniques for pattern-matching localization systems,” in Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems ( MASS ’07), October 2007.

[17] “The R Project for Statistical Computing,” http://www.r-project.org/.
Submit your manuscripts at
http://www.hindawi.com