RESEARCH ARTICLE

Landmark-based device calibration and region-based modeling for RSS-based localization
Hung Nguyen Manh¹, Ching-Chun Huang²* and Lee Hsiao-Yi¹

¹ Department of Electrical Engineering, National Kaohsiung University of Applied Sciences, 415 Chien Kung Rd., Kaohsiung 80778, Taiwan
² Department of Electrical Engineering, National Chung Cheng University, 168 University Rd., Min-Hsiung, Chiay-Yi 62102, Taiwan

ABSTRACT

During the past decades, many fingerprint-based indoor positioning systems have been proposed and have achieved great success. However, uncontrolled effects of device diversity, signal noise, and dynamic obstacles could recognizably degrade the performance of modern fingerprint-based indoor localization systems. In this paper, to amend the variations in radio signal strengths (RSSs) caused by device diversity, we proposed an automatic device calibration process. Because of device diversity, the sensed RSS would deviate from the trained radio map and thus leads to poor positioning. An RSS transform function could be adopted to calibrate the RSS variation between different devices and overcome the device diversity problem. However, to train the transform function, a data collection process is required. Unlike conventional calibration methods requiring manual data collection, we proposed a landmark-based automatic collection process. Based on the detection of Wi-Fi landmarks, our system could automatically collect pair-wise RSS samples between devices and train the RSS transform function without extra human power. In addition, to well represent the effects of signal noise and dynamic obstacles, a region-based RSS modeling method was also proposed. The proposed modeling method allows our system to perform region-based target localization and utilize more robust region information for localization. Experiments in various environments demonstrate that our system could give a better positioning performance by properly handling the RSS variation caused by signal noise, dynamic environment, and device diversity. Copyright © 2015 John Wiley & Sons, Ltd.

KEYWORDS
indoor localization system; auto calibration; region-based model

*Correspondence
Ching-Chun Huang, Department of Electrical Engineering, National Chung Cheng University, 168 University Rd., Min-Hsiung, Chiay-Yi 62102, Taiwan.
E-mail: chingchun.huang6@gmail.com

1. INTRODUCTION

During the past several years, smart phones have become an important personal device for accessing cloud services. Therefore, users could enjoy many mobile services on the go. Among these services, location-based applications are important. According to the location information, the service system can provide time zone, report local weather condition, broadcast local popular activities, suggest optimal navigation paths, and so on. With help from the Global Positioning System (GPS), many outdoor location-based services have gained success recently. Similar to these outdoor services, indoor location-based services are now acquiring increasing attention, which makes indoor localization an emergent and discussible issue.

According to the scenario of service, different indoor localization systems may have different design constraints. For device designer, they could well control the smart device but could not choose the testing environment. Therefore, they would fully utilize the functions and sensors of the device to achieve self-localization. One popular framework for self-localization is dead reckoning [1], which estimates the user’s new location based on his or her previous locations and motion parameters. Dead reckoning is efficient and convenient because all necessary motion parameters can be extracted by the internal motion sensors of a user’s device. However, the localization result is highly subject to cumulative errors. Although device designers could introduce more internal sensors or upgrade the sensor performance to reduce the errors, without the help of
the external environment, the errors would be propagated over time.

In another scenario, the museum guidance, both the smart device and the environment can be well controlled by the localization system. Thus, a good solution for indoor localization should utilize the full potential of the device and the environmental information. For instance, the authors in [2] used the embedded motion sensors to determine the motion model and also combined the radio signal strength (RSS) as the external reference to reduce the cumulative errors.

In contrast, the service provider in a shopping mall may face different system constraints. In that case, the system can control the environment to provide location-based service, but the user’s device is uncontrollable. Basically, the service provider would choose to provide localization service for all possible users. However, users’ devices are diverse. Some devices support fantastic applications; some only provide basic functions. Thus, it is preferable for a localization system to only use the basic functions provided by a user’s device and to look for other useful information from the environment. Some sensor-network-based methods have been introduced for the scenario such as radio frequency identification (RFID) systems [3], infrared radiation (IR) systems, and iBeacon systems [4]. These methods are effective and easy to implement, but the cost for system setup and maintenance is high.

In the paper, we assume that the proposed system is designed for the scenario in a shopping mall. Because of the quick growth of ‘internet of things’, radio signal has become the most economic and ubiquitous environmental information for localization. Thus, the proposed system aims to utilize RSS and limited supports from the user’s device, such as the Wi-Fi sensor and the commercial inertial measurement unit (IMU), for localization.

In the past decades, many wireless indoor localization methods have been proposed. Among them, radio-based solutions are of great interest because of the off-the-shelf Wi-Fi infrastructure and the popularity of Wi-Fi-embedded smart phones. A key step toward radio-based geolocation is to build the mapping between RSSs and locations. In general, the mapping techniques could be divided into model-based and fingerprint-based methods. For model-based methods, a conventional process is to formulate the signal propagation model [9] in order to describe the variation of RSS over distance. However, the signal propagation model could not well model the complex attenuation and fading effects caused by indoor obstacles, such as walls, doors, and moving persons. Meanwhile, to determine the target location, the locations of access points (APs) should be known in advance to serve as reference positions.

To overcome the aforementioned technical challenges, fingerprint-based indoor positioning methods have been proposed [10–13]. A fingerprint-based system is typically composed of a training stage and an online positioning stage. During the training stage, a site survey process [14] is needed. By collecting the RSSs of multiple APs and recording the corresponding locations, a radio map is built and treated as the location fingerprint. During the positioning querying stage, the signature of the observed RSSs is compared with the fingerprint throughout the recorded locations. The site with the highest RSS similarity could then be selected as the estimated target location. By using the data-driven concept rather than modeling the signal propagation, the site survey process provides a more flexible way to describe the mapping between the sensed RSSs and their locations. Also, the prior knowledge of AP locations is unnecessary. Although fingerprint-based methods provide a flexible and efficient manner for indoor positioning, some challenges still exit. We divide the challenges into two aspects: (i) the device diversity or heterogeneity problem and (ii) the signal noise effects and the multi-modal RSS variation caused by dynamic obstacles. These challenges will be discussed individually in the succeeding sections.

1.1. Device diversity

One major issue is the device diversity. The device used to train the radio map and the device used to sense the RSS for position query are different. As we know, mobile devices present different Wi-Fi sensing abilities depending on the embedded hardware and software. Because of device diversity, the sensed RSSs for positioning would deviate from the trained RSS radio map. Figure 1(a) shows the histograms of collected RSSs at the same location from the same AP but by different mobile devices. The RSS distributions of different devices cover quite different intensity ranges. Without calibrating the RSS variation, the localization accuracy would be severely degraded.

To overcome the device diversity problem, the authors in [15–19] proposed to use relative features, such as the RSS differences between pairs of APs, the RSS ratios of APs, or the RSS order of multiple APs to replace the absolute RSS. However, inside a room, the relative RSS features tend to be similar at most locations. Hence, only room-level positioning could be achieved. To have higher positioning accuracy, some studies [20–22] proposed to train an RSS transform function to adjust the variation and make progressive improvement. In [23], the authors have tested more than six devices and found that the sensed RSSs of these devices express strong linear correlation. With a suitable calibration process, the RSS variation could be reduced by using a trained linear transform function.

Although device calibration could help to adjust the RSS deviation, a practice issue that has not been well-addressed is how to automatically collect the RSS pairs to train the transform function. In most conventional calibration methods, the training RSS pairs between devices should be manually collected. However, the collection process is time-consuming and needs a lot of manpower. Even worse, manual collection training samples are impracticable for a real service because we are not able to know the user device in advance. Hence, one of our system’s goals
Figure 1. The radio signal strength (RSS) distribution under the effects of (a) device diversity, (b) signal noise, and (c) dynamic obstacles.

is to automatically collect the training samples for device calibration and save high-priced manpower.

However, the RSS variation cannot be perfectly erased by using only the RSS transform function. When comparing the RSS histograms captured by the two different devices as shown in Figure 1(a), we found that both the mean and variance of the distributions are different. After the device calibration process, the mean bias may be reduced but the difference of distribution variance still remains. To overcome the problem, in this paper, we further adopted a wider intensity kernel for RSS density estimation to relieve the effects of the variance difference. However, as a trade-off, a wider intensity kernel would enlarge the variance of the estimated RSS distribution. If using the kernel-enlarged RSS modeling for target positioning, we may sacrifice the localization accuracy. Thus, another contribution of our system is that we proposed a region-based positioning method whose radio map was modeled by a new method — multi-dimensional kernel density estimation (MD-KDE). Based on the region-wise RSS signature, more discriminative features are considered for positioning. Hence, our system can utilize the kernel-enlarged distribution for indoor positioning and simultaneously improve the localization accuracy.

1.2. Signal noise and dynamic obstacles

Besides the device diversity problem, signal noise and dynamic obstacles are the major factors leading to the RSS variation. To know the effects, we collected many RSSs of one AP at the same position and drew RSS histograms. The RSS distribution affected by noise and obstacles are shown in Figure 1(b) and (c). We found that the RSS variation could not be disregarded.

Many conventional methods used Gaussian distribution [24] to model the noise effects at each location. However, some studies like [25,26] have shown that Gaussian distribution may not be suitable for all real cases. For example, in Figure 1(c), we show a multi-modal RSS distribution caused by obstacles. This situation could be easily observed in many indoor environments. When an obstacle, such as a door, exists between an AP and a receiver, the open and closed statuses of the door would naturally form the multi-modal phenomenon. To handle the multi-modal problem, Park et al. [23] proposed to use kernel density estimation (KDE) to model the RSS variation at a location. In general, KDE is a non-parametric method, which estimates the RSS distribution purely based on the observed data. Without the model constraint, KDE is more flexible in representing a multi-modal distribution and modeling noise effects.

Even so, the aforementioned Gaussian and KDE methods build the radio map in a discrete fashion. Only limited locations with training samples are included in the trained radio map. If we search the optimal target location in the limited location set, quantization errors would decrease the overall accuracy. In order to improve the precision of the fingerprint-based RSS modeling, some state-of-the-art methods, grounded on Gaussian regression process (GRP) [27,28] and its modification [29] have been proposed recently. Those methods introduced a spatial kernel to correlate the information among neighboring regions and represent the radio map in a continuous style. Although GRP brings on some advantages, these GRP-based methods still modeled the RSS variation at each location by using a Gaussian distribution. As mentioned, the RSS Gaussian model could not successfully handle the multi-modal problem.

To address the aforementioned issues, in our system, the proposed region-based modeling utilized a multi-dimensional kernel to cover both the RSS domain and the spatial domain. The intensity kernel allows our system to clearly represent the RSS distribution and model the effects from noise and dynamic obstacles. Furthermore, by introducing the spatial kernel into the RSS modeling, our system could build the RSS radio map at continuous locations, consider region information for positioning, and require fewer training samples. Later on, we will explain the details.

The rest of this paper is organized as follows. In Section 2, we overview the proposed system and highlight the major contributions. In Section 3, we detail the
proposed auto-calibration method for device diversity. In Section 4, we explain the region-based modeling for training a radio map and the region-based target tracking. Experimental results and comparisons are presented in Section 5. Finally, we conclude the paper in Section 6.

2. OVERVIEW OF THE PROPOSED METHOD

As shown in Figure 2, the proposed system is composed of two phases, including the radio map training phase and the operation phase. For the training phase, the data recording the RSS signature and its corresponding location is collected through a site survey process. In most cases, the site survey process automatically attains the RSS samples of multiple APs by a training device but needs to manually record the location information for each training sample. After the site survey process, the proposed region-based RSS modeling method is used to train the continuous radio map, which can model the RSS probability distribution at each continuous location.

For the operation phase, a testing device measures RSS samples of multiple APs. To address the device diversity problem, before localization, we have an RSS calibration step to adapt a measured RSS sample to the training device. These calibrated RSS samples are later input to the proposed region-based RSS modeling to generate the testing RSS probability distribution for location query. By measuring the similarity between the testing RSS probability distribution and the trained radio map, our system can identify the target location continuously.

In order to calibrate RSS, our system needs to determine the RSS transfer function for device calibration. As shown in Figure 2, we introduce an automatic device calibration module to determine the transfer function. In our system, a previous method [20] based on linear regression was used to amend the RSS variation caused by device diversity. To determine the linear RSS transfer function, we proposed a landmark-based approach to automatically collect the RSS pairs between the user’s device and the training device. Later, these collected RSS pairs are used to train the linear regression function. The estimation process of the RSS transfer function is detailed in Figure 3.

In the proposed system, the major contributions could be summarized in three aspects as follows:

1. In the device calibration module: We introduced an auto-calibration procedure to overcome device diversity. We found that the Wi-Fi landmarks [30] provide useful supports for our system to automatically collect the RSS pairs between the user’s device and the training device without manpower. These collected RSS pairs could be used to dynamically train the transfer function for device calibration. Experimental results demonstrate that our method is able to robustly detect the Wi-Fi landmarks, collect RSS pairs, and achieve a reasonable RSS transfer function.

2. In the training phase: We proposed an MD-KDE modeling method, which applies the kernel concept...
on not only the RSS domain but also the spatial domain. Based on the MD-KDE, our system introduced a region-based RSS modeling module to model the noise effect, represent the multi-modal RSS distribution, and reduce the RSS variations caused by device diversity. Moreover, by considering the adjacency correlation, our system can systematically represent the RSS distribution at each continuous location with reduced training samples.

(3) In the operation phase: We proposed a region-based target localization method. In MD-KDE, the spatial kernel spreads the RSS information of a location to its neighborhood. Hence, our method can extend the conventional radio map from single-location-based modeling to region-based modeling. For positioning, the concept of MD-KDE is again used to model the testing RSS distribution. A distribution matching process was introduced for position query instead of resorting only to one observed RSS sample like most previous works. By considering the region-based RSS signature, the distribution matching method can improve the accuracy for target localization.

### 3. AUTO-CALIBRATION METHOD FOR DEVICE DIVERSITY

Because of the device diversity problem, for an AP, two devices at the same location would receive different RSSs. Hence, an inter-device calibration is required before a target can be precisely localized. In our system, we used a linear regression model to compensate the RSS variation. From a device $A$ to a device $B$, the RSS transform function is described in Equation (1).

$$S_B^l = \alpha_A^B S_A^l + \beta_A^B$$

where $S_A^l$ and $S_B^l$ denote the RSS values received by device $A$ and $B$ at the same location $l$; $\alpha_A^B$ and $\beta_A^B$ are coefficients of the linear regression model. To learn the coefficients $\alpha_A^B$ and $\beta_A^B$, we need to collect enough RSS pairs between devices. We denote the set of RSS training pairs as $\{(S_A^l, S_B^l)\}_i, i = 1 \sim SN$ where $SN$ is the sample number. Based on the training set, the optimal solution $(\alpha_A^B, \beta_A^B)^*$ could be determined by minimizing the least square error defined by

$$\mathbf{(\alpha_A^B, \beta_A^B)^* = \arg \min_{\alpha_A^B, \beta_A^B} \sum_{i=1}^{SN} \left( \frac{\alpha_A^B S_A^l - \beta_A^B}{2} \right)^2}$$

(2)

In order to remove some unreliable RSS pairs, RNAdom SAmple Consensus (RANSAC) algorithm was also used in the optimal process to filter out the outlier. Please refer to Fischler and Bolles [31] for the details.

Instead of manually collecting the RSS training pairs, we proposed an automatic collection process as shown in Figure 3. To collect an RSS pair, two devices should sense the RSS at the same location. In our system, we used the ‘landmark’ concept to synchronize the location. In detail, we used a training device to detect the Wi-Fi landmarks and record their positions and RSS features as the landmark dataset. While a user is traveling within an indoor space with his mobile device, Wi-Fi landmarks would be sequentially detected. Once a Wi-Fi landmark is found, the position of a user’s device could be located by matching the landmark feature against the landmark dataset. After matching, some RSS pairs between two devices could be collected and recorded for later use. For matching, we also proposed a novel landmark feature to describe the signature of a Wi-Fi landmark. We illustrate the details in the succeeding sections.

### 3.1. Wi-Fi landmarks

To recognize a Wi-Fi landmark, rather than resort to the absolute signal strength, we looked into the intensity change of the RSS change. In Figure 4(a), we illustrate some examples of Wi-Fi landmarks. If a user walks through the nearby pathway of an AP with a mobile device, the sensed RSS values increase when the user moves closer to the AP and decrease while the user moves away. The RSS tend to increase from the increase to the decrease naturally forms a bell-shape structure whose peak location is defined as the Wi-Fi landmark.

Although the bell-shape tendency of the RSS can help to identify a Wi-Fi landmark, some exceptional cases should be further discussed. As shown in Figure 4(b), if a user takes a U-turn along a pathway, the distribution of the RSS will also appear in a bell shape and a fake Wi-Fi landmark would be detected. To avoid the exceptional case, we used the information from the commercial compass embedded in the smart phone to understand a user’s walking patterns. Once a U-turn is detected, the peak value of the RSS distribution would be treated as a Wi-Fi fake landmark.

To demonstrate the robustness of the Wi-Fi landmark, we showed the RSS variation along a pathway in Figure 5. The blue cycles are RSS values sensed by an HTC smart phone and the red cycles are RSS values sensed by a Samsung smart phone. Because of device diversity, the RSS values received by the two phones at the same location are different. We repeated the measurement process several times for both two devices. As shown in the figure, the RSS value is time-variant because of noise. However, the RSS is quite stable and invariant to both device diversity and noise. This property allows our system to automatically collect stable RSS pairs for training.

### 3.2. Wi-Fi landmark detection

Because the RSS is sensitive to noise, environment obstacles, and communication fading, it is not trivial to detect the Wi-Fi landmark. As the blue curve shown in Figure 6, the RSS frequently vibrates because of environmental effects. To extract the intrinsic RSS and reduce the noise effect, a process for signal smoothing is necessary.
To smooth the signal, many window-based low-pass filters could be used. However, these filters cannot filter out the outliers. In a wireless environment, the device may occasionally report unreasonable RSSs. These RSSs should be treated as outliers and could not be involved in the smoothing process. To reduce the outlier effect, in our system, we used Kalman filter [32] for signal smoothing. Kalman filter provides a systematic framework to dynamically learn the signal trend from previous RSS observations. Based on the signal trend, a Kalman gain would be dynamically derived for a new RSS observation. By using the Kalman gain for signal smoothing, Kalman filter can implicitly filter out an outlier. In addition, Kalman filter has its nature to process streaming data. While a new RSS is observed, our system can directly consider the new RSS for signal smoothing. In particular situations, no RSS is reported for a short period. In this case, Kalman filter can predict the missing RSSs based on the learned signal trend and still keep the system workable. Figure 6 shows an example of the raw RSS distribution and the smoothed RSS distribution. With the help of Kalman filter, the noise effects are significantly reduced; the internal RSS tendency also becomes apparent.

However, in Figure 6, we could still find some local maximums in the smoothed RSS distribution. Those local maximums are caused by dynamic obstacles and communication fading. They are not the landmarks of interest and should be removed. To only detect the true Wi-Fi landmarks, another process is necessary to simultaneously identify the signal trend and local peaks. In our system, we found that the run-length [33] signal representation offers an efficient method to identify both the signal trend and local maximums. Hence, we applied run-length coding to landmark detection. However, it is noteworthy that run-length coding may not be the only one solution.

The run-length coding process records both the duration of incremental tendency and the duration of descending tendency. If the duration is long enough, the tendency would be confirmed; otherwise, we would regard the tendency as unstable and ignorable. Next, our system determined the stable minimums or maximums by looking for the turning points between the incremental and descending tendencies. These extreme points are essential to recognizing the Wi-Fi landmarks.

We used an example shown in Figure 7 to illustrate the detail procedure of our run-length coding. First, a differential operator was applied to indicate the incremental and descending positions. If $RSS_L$ denotes the RSS sensed at location $L$, 
the differential operator is defined as

\[ D_{\text{RSS}_L} = \text{RSS}_L - \text{RSS}_{L-1} \]  

(3)

If \( D_{\text{RSS}_L} \) is positive, it represents an incremental moment; otherwise, we have a descending moment. Next, we accumulated the duration of the incremental and descending tendencies and stored the run length in a table. As shown in Figure 7, we use “+” to denote an incremental tendency and “Δ” to denote a descending tendency. In the run-length table, the first row is the tendency identity (TID), the second row is the status (ST) of the signal tendency, and the third row records the duration (DU).

If we check the run length in Figure 7(a), we may find that some tendencies last only for short duration. These transient tendencies should be further modified. In our system, a run-length threshold \( RL_{th} \) is used. If the duration is longer than \( RL_{th} \), the tendency is stable; otherwise, it is considered unstable.

To correct an unstable tendency, the status of its neighboring stable tendencies could be of some help. We hence checked all the stable tendencies. Between any two stable tendencies, if only unstable tendencies exist, those unstable tendencies would be corrected simultaneously. First, the duration of the successive unstable tendencies is calculated. If the duration is smaller than \( RL_{th} \), all the statuses of these successive unstable tendencies are set to the status of the previous tendency. For instance, in Figure 7(a), between the second tendency (TID2) and the fourth tendency (TID4), there is an unstable tendency, TID3. If we set \( RL_{th} \) equal to 10, the duration of TID3 is smaller than \( RL_{th} \). Thus, the status is corrected as ‘Δ’ equal to TID2. On the other hand, if the duration of the successive unstable tendencies is larger than \( RL_{th} \), a new tendency may occur. In this case, we compared the final RSS value of the previous stable tendency and the first RSS value of the next stable tendency to correct the status. As an example, in Figure 7(a), there are five unstable tendencies between TID4 and TID10. The statuses of TID5 to TID9 are all changed to “+” because the final RSS value of TID4 is smaller than the first RSS value of TID10, which implies an incremental tendency. A corrected run-length result is shown in Figure 7(b).

Finally, to detect the Wi-Fi landmarks, we checked the corrected run-length table and identified the stable peaks and valleys. If the RSS difference between a peak and its neighboring valleys is larger than an experimentally designed threshold \( \delta_c \), the peak would be selected as a
Wi-Fi landmark. The rule is described by

\[
D_i = P_i - \max_{(L,R)} (V_{Li}, V_{Ri}) > \delta, \tag{4}
\]

where \(P_i\) is the RSS of the \(i\)th peak, \(V_{Li}\) and \(V_{Ri}\) are the RSSs of two neighboring valleys, and \(D_i\) is the RSS difference.

\[
FSIM(LM_1LM_2) = \left[ \sum_{p=1}^{N_1} \sum_{k=1}^{N_2} I (AP_{p,L}^{LM_1}, AP_{k,L}^{LM_2}) \right] / \left[ \sum_{p=1}^{N_1} \sum_{k=1}^{N_2} I (AP_{p,L}^{LM_1}, AP_{k,L}^{LM_2}) \right]
\]

\[
3.3. \ \text{Wi-Fi landmark matching}
\]

After a user brought his mobile device and walked around the indoor environment for a while, several Wi-Fi landmarks would be detected. These Wi-Fi landmarks would be matched with the landmarks detected by the training device. To implement the matching process, each Wi-Fi landmark was represented by the proposed relative RSS features, which are insensitive to device diverse. At each landmark, the device would receive RSSs of multiple APs. Among them, the top \(N\) RSSs greater than a threshold were selected. In our system, if an RSS is smaller than \(-90\) dB, the sample is usually unreliable. Thus, the threshold is set to \(-90\) dB. In addition, each RSS attached to its AP’s basic service set identifier (BSSID) is recorded. The \(N\) RSSs and their BSSID would be used to calculate the landmark feature.

We calculated the RSS difference \(D_{RSS}\) of all possible RSS pairs from the \(N\) RSSs. Note that \(D_{RSS}^{(AP_r,AP_j)}\) represents the RSS difference of \(AP_i\) and \(AP_j\). That is \(D_{RSS}^{(AP_r,AP_j)} = RSS_{AP_i} - RSS_{AP_j}\). Next, each \(D_{RSS}\) is transferred by the mapping function defined in Equation (5) to get the final feature \(F_{DRSS}\), ranging from \(-1\) to \(1\).

\[
F_{DRSS}^{(AP_r,AP_j)} = \tanh \left( W_{th} \cdot D_{RSS}^{(AP_r,AP_j)} \right) \tag{5}
\]

In (5), \(W_{th}\) is a tunable parameter to control the contribution of \(D_{RSS}\). When \(\|D_{RSS}\|\) is much larger than \(W_{th}\), \(F_{DRSS}\) would be equal to \(1\) or \(-1\), which indicates that only the RSS order of the RSS pair is concerned. On the other hand, if \(\|D_{RSS}\|\) is smaller than \(W_{th}\), the order information, indicated by the positive or negative sign, is still used; however, the confidence is determined by the magnitude of \(\|D_{RSS}\|\). Reasonably, small \(\|D_{RSS}\|\) implies that the order information is unstable.

Considering all the possible RSS pairs, we have \(M = C^2_N\) features to describe a Wi-Fi landmark \((LM)\). We used \(F_k^{LM}\) to denote the \(k\)th feature of \(LM\) and recorded its AP’s BSSID in the form \((AP_r, AP_j)^{LM}\). In this case, \((AP_r, AP_j)^{LM}\) indicates that the \(k\)th feature is derived from the RSSs sensed from \(AP_r^{LM}\) and \(AP_j^{LM}\). Also, we always keep the BSSID of \(AP_r^{LM}\) smaller than the BSSID of \(AP_j^{LM}\). Based on these notations and Equation (5), \(F_k^{LM}\) is defined by

\[
F_k^{LM} \triangleq F_{DRSS}^{(AP_r, AP_j)^{LM}} = \tanh \left( W_{th} \cdot D_{RSS}^{(AP_r, AP_j)^{LM}} \right)
\]

Next, to match two Wi-Fi landmarks, we measured the feature similarity \(FSIM(LM_1LM_2)\) of two landmarks in Equation (7). High \(FSIM(LM_1LM_2)\) means \(LM_1\) and \(LM_2\) match.

In Equation (7), \(N_1\) and \(N_2\) are the selected RSS numbers of \(LM_1\) and \(LM_2\). \(I(.)\) is an indicator function used to verify if the \(p\)th feature of \(LM_1\) and the \(k\)th feature of \(LM_2\) are derived from the same AP pair. The indicator function \(I(.)\) would output 1 when the equality within \(I(.)\) is satisfied, otherwise \(I(.)\) would output 0. Finally, \(F_p^{LM_1}\) denotes the \(p\)th feature of \(LM_1\) and \(F_k^{LM_2}\) is the \(k\)th feature of \(LM_2\). They could be measured in Equation (6).

However, in the real environments, two APs may be installed in the neighboring regions, which may produce two neighboring landmarks. As previously mentioned, the relative RSS feature tends to be alike within a region. If the landmark matching is simply grounded on the feature similarity, false matching may occur. Please refer to Figure 8 as a real example. To overcome the problem, for each landmark, we denote the BSSID of the AP that produces the landmark as \(AP(LM)\) and define an ‘AP matching’ indicator \(AP_M\) as

\[
AP_M(LM_1LM_2) = \begin{cases} 
1, & \text{if } AP(LM_1) = \neq AP(LM_2) \\
0, & \text{if } AP(LM_1) = \neq AP(LM_2) 
\end{cases} \tag{8}
\]

In principle, if two landmarks are formed by different APs, their BSSIDs should not match. In view of the evidence from the landmark feature and the result of AP matching, the landmark similarity is finally calculated in Equation (9).

\[
SIM(LM_1LM_2) = FSIM(LM_1LM_2) \times AP_M(LM_1LM_2) \tag{9}
\]

Note that we could easily find that the maximum value of the similarity is \(M\). If two landmarks detected by two devices have a similarity larger than 0.5\(M\), we considered them to be a matching pair. In few cases, one landmark may match multiple landmarks. While this happened, we selected the one with highest similarity. After the matching process, we could collect some RSS pairs. These RSS pairs are treated as the training set to determine the linear regression model defined in Equation (2).
where closer neighbors provide highly relevant RSS information by using the spatial Gaussian kernel. For a position, its kernel. It is noteworthy that the proposed method produces the learning process and the function of the spatial simplify the concept, we used one dimension to illustrate the training process. It is noticeable that the location

use Figure 9 to illustrate the training process. It is noticeable that the location

In (10), the conditional distribution could be calculated by

\[ P(S, L) = \frac{1}{N} \sum_{i=1}^{N} K_{RSS} \left( \frac{S - S_i^L}{h_{RSS}} \right) K_L \left( \frac{L - L_i}{h_L} \right) \]  

(10)

In (10), \( K_{RSS}(\cdot) \) and \( K_L(\cdot) \) are two different Gaussian kernels for the RSS domain and the spatial domain; \( h_{RSS} \) and \( h_L \) are their kernel widths correspondingly. Based on the joint probability distribution, we extracted the conditional distribution of RSS given a known location \( L_o \). The conditional distribution could be calculated by

\[ P(S | L = L_o) = \frac{P(S, L_o)}{P(L_o)} = \frac{\int P(S | L_o) dS}{P(L_o)} \]

(11)

where \( \int P(S, L_o) dS \) is a normalization term to make sure the equation, \( \int P(S | L_o) dS = 1 \), is satisfied. We can also use Figure 9 to illustrate the training process. It is noticeable that the location \( L \) represents the space location. To simplify the concept, we used one dimension to introduce the learning process and the function of the spatial kernel. It is noteworthy that the proposed method propagates the RSS information to its neighboring locations by using the spatial Gaussian kernel. For a position, its closer neighbors provide highly relevant RSS information so that our method could use fewer samples for learning. On the other hand, the RSSs of these farther and less relevant neighbors are also propagated to the position with smaller weights. By combining the neighboring information into the RSS model, our system can integrate the region-based RSS signature into the trained radio map. Hence, the discrimination ability for target positioning is enhanced.

In an indoor environment, there are multiple APs. By fusing the RSS signature of these APs, more features could be utilized for target tracking. Intuitively, we can assume that the RSSs of these APs are independent. Hence, we could individually train the RSS distribution of different APs and use the product rule to integrate information from multiple APs. Suppose we have \( M \) APs within the indoor environment and \( P_{train}(S_{AP_i} | L) \) stands for the RSS distribution of the \( i \)-th AP, the trained joint RSS distribution at a given location \( L \) could be formulated as Equation (12).

\[ P_{train}(\vec{S} | L) = P_{train}(S_{AP_1}, \ldots, S_{AP_M} | L) = \prod_{i=1}^{M} P_{train}(S_{AP_i} | L) \]  

(12)

Here, \( P_{train}(S_{AP_i} | L) \) is determined by Equation (10); the suffix “train” indicates the distribution of our training process. \( \vec{S} = (S_{AP_1}, \ldots, S_{AP_M}) \) defines the joint RSS configuration.

Moreover, the introduced spatial kernel enables our system to train the RSS model at any continuous location. For any location \( L_o \), the joint distribution \( P(S, L_o) \) could always be estimated by Equation (10) even if no training sample is collected at \( L_o \). To learn the multi-modal RSS distribution caused by the dynamic obstacles, MD-KDE also provides a systematic way. As shown in Figure 9, the distribution of the learned RSS could be approximated by the mixture of many Gaussians and is purely dependent on the training samples.
4.2. Operation phase: target tracking

In the operation phase, we also used MD-KDE method to model the testing RSS distribution based on the sequentially sensed samples. As shown in Figure 10, a user’s device periodically senses the RSSs of multiple APs. These samples are recorded as the system observation. For a location of interest $L_{K_{LOI}}$, the user’s device would sense $2N$ testing samples before and after the location $L_{K_{LOI}}$. In practice, $N$ is set to 5. The observed RSS at $L_{K_{LOI}}$ is denoted as $TS_{K_{LOI}}^{K_{LOI}}$. The sample index $K_{LOI}$ means that $TS_{K_{LOI}}^{K_{LOI}}$ is the $K_{LOI}$th sample in the observation sequence. To form the RSS distribution of $L_{K_{LOI}}$ for the $i$th AP, $2N+1$ RSS observations with their sample indices centered at $K_{LOI}$, $TS_{K_{LOI}+q}^{K_{LOI}+q}AP_{j}$, are collected. By applying the proposed MD-KDE method to these samples, we can determine a region-based RSS conditional distribution of the $i$th AP and at the location $L_{K_{LOI}}$ by

$$P_{test}(S_{AP,i}|L = L_{K_{LOI}}) = \frac{P_{test}(S_{AP,i},L_{K_{LOI}})}{\int P_{test}(S_{AP,i},L_{K_{LOI}}) dS_{AP,i}}$$

(13)

where

$$P_{test}(S_{AP,i},L_{K_{LOI}}) = \frac{1}{2N+1} \sum_{q=-N}^{N} K_{RSS} \left( \frac{|S_{AP,i} - TS_{K_{LOI}+q}^{K_{LOI}+q}|}{h_{RSS}} \right) K_L \left( \frac{|L_{K_{LOI}} - L_{K_{LOI}+q}|}{h_L} \right)$$

(14)
In Equation (13), we use the suffix “test” to indicate that the distribution is used for testing.

However, we still need to determine the location distance \(|L_{K_{\text{LOI}} - L_{K_{\text{lo}}}}|\) in Equation (14). In a practical system, by synchronizing the RSS sensing time and the user’s step tempo, the distance term \(|L_{K_{\text{LOI}} - L_{K_{\text{lo}}}}|\) could be approximated by the summarization of the user’s step lengths. Nowadays, many off-the-shelf methods [34,35] could be used to roughly estimate the step tempo and step lengths. In practice, the acceleration meter may output regular patterns while the target is moving. Each pattern represents one step. By detecting the pattern, the step number and step tempo could be determined. For step length estimation, according to a biomechanical model, previous works [34,35] have shown that the step length is closely relative to the user’s height (\(H\)) and step tempo (\(S_t\)). In our system, we adopted the step length (\(S_L\)) estimator proposed in [34], which is defined as

\[
S_L = (a \times S_t + b) \times H + c
\]  

(15)

In this equation, \(a\), \(b\), and \(c\) are three constant parameters. By manually giving a set of \(H, S_t\), and \(S_L\) for training, we can determine the three parameters by finding the least square error solution.

The step length estimated by the method may introduce some inevitable errors. However, in Equation (14), the term we need is \(K_L(\frac{|L_{K_{\text{LOI}} - L_{K_{\text{lo}}}}|}{h_L})\) rather than \(|L_{K_{\text{LOI}} - L_{K_{\text{lo}}}}|\). We may find that \(K_L(\frac{|L_{K_{\text{LOI}} - L_{K_{\text{lo}}}}|}{h_L})\) is also depended on the spatial kernel size \(h_L\) and the kernel function \(K_L(\cdot)\). Because \(K_L(\cdot)\) is a Gaussian kernel, by giving a larger kernel size \(h_L\), the term \(K_L(\frac{|L_{K_{\text{LOI}} - L_{K_{\text{lo}}}}|}{h_L})\) would become less sensitive to the estimation errors of \(|L_{K_{\text{LOI}} - L_{K_{\text{lo}}}}|\).

Next, if we consider the RSS observations from \(M\) APs, the joint RSS distribution for testing could be formulated by

\[
P_{\text{test}}(\tilde{S}|L) = \prod_{i=1}^{M} P_{\text{test}}(S_{\text{AP}_i}|L_{K_{\text{lo}}})
\]  

(16)

To localize the target, a conventional way is to measure the similarity between the trained RSS distribution \(P_{\text{train}}(\tilde{S}|L)\) in the radio map and the observed RSS distribution \(P_{\text{test}}(\tilde{S}|L_{K_{\text{lo}}})\). As we assumed that the RSSs from different APs are independent, the similarity could be calculated by

\[
\text{Sim}(L) = F_m(P_{\text{train}}(\tilde{S}|L), P_{\text{test}}(\tilde{S}|L_{K_{\text{lo}}})) = \prod_{i=1}^{M} F_m(P_{\text{train}}(S_{\text{AP}_i}|L), P_{\text{test}}(S_{\text{AP}_i}|L_{K_{\text{lo}}}))
\]  

(17)

In Equation (17), \(M\) is the number of AP. \(F_m(\cdot)\) could be any reasonable metric for similarity measurement of two distributions. In our system, we used Earth Mover Distance [36] to design \(F_m(\cdot)\).

However, it may not be efficient to thoroughly check all possible locations especially for the proposed continuous radio map. To successively track the target of concern, we relied on the temporal constraint and the RSS observations. Given the target location at time \(T-1\), the temporal constraint limits the system to search only the neighboring region. On the other hand, the RSS observation provides the evidence to infer the target location by matching the RSS singularity with the trained radio map. To realize target tracking, we applied the particle filter algorithm [37]. Because particle filter has been thoroughly studied in the
1. At time \( T - 1 \), we have \( N \) location particles \( \{ L_{i}^{T-1} \}_{i=1 \sim N} \). Each particle has an equal weight \( 1/N \).

2. At time \( T \), each particle moves to its next destination by following a random walk model in order to uniformly discover the neighboring region. Thus, we achieve a new location sample set \( \{ L_{i}^{T} \}_{i=1 \sim N} \), which represents the candidate locations for the target of concern at the current moment.

3. For each candidate location \( L_{i}^{T} \), we updated its weight \( \tilde{w}_{i}^{T} \) to reveal the likelihood of \( L_{i}^{T} \) to be the target location. In principle, the weight \( \tilde{w}_{i}^{T} \) is proportional to the RSS similarity of \( L_{i}^{T} \), which is defined as \( \text{Sim} \left( L_{i}^{T} \right) F_{m} \left( \tilde{S}_{\text{train}} \left( L_{i}^{T} \right) \right) F_{\text{test}} \left( \tilde{S}_{\text{test}} \left( L_{i}^{T} \right) \right) \) in Equation (17). That is, if the similarity is high between the trained RSS distribution at \( L_{i}^{T} \) and the tested RSS distribution, we have high confidence to treat \( L_{i}^{T} \) as the target location. Hence, in our system, \( \tilde{w}_{i}^{T} \) is determined by \( \text{Sim} \left( L_{i}^{T} \right) \). Next, we normalized the set of weights \( \{ \tilde{w}_{i}^{T} \} \) to ensure that the sum of the weight set is 1. The set of the normalized weights is notated as \( \{ \tilde{w}_{i}^{T} \} \).

4. The system estimated the target location \( L_{LOI}^{T} \) at time \( T \) by the weighted combination of the candidate locations. That is \( L_{LOI}^{T} = \sum_{i=1}^{N} \tilde{w}_{i}^{T} \times L_{i}^{T} \).

5. The resample process [37] is used to generate a new particle set \( \{ L_{i}^{T} \}_{i=1 \sim N} \). Although each new particle \( L_{i}^{T} \) has equal weight \( 1/N \), in statistics, the resample process would make sure the original location particle set \( \{ L_{i}^{T} \}_{i=1 \sim N} \) with the normalized weights \( \{ \tilde{w}_{i}^{T} \} \) would be well-approximated by the equally weighted particle set \( \{ L_{i}^{T} \}_{i=1 \sim N} \).

6. Return to step (1) and consecutively track the target for the next time moment.

**Figure 11.** The setting of the testing environment for landmark detection and matching.

**5. EXPERIMENTS AND COMPARISON**

In this section, we discuss the performance of the proposed system from three aspects. First, we evaluate our auto-calibration method and show its performance for device diversity. In the second aspect, we would analyze the performance of our region-based localization method under the environmental effects including noise and dynamic obstacles. In the third aspect, we would test our system in a complicated environment and compare the performance with some state-of-the-art localization systems. At the end, a complete discussion of these localization systems would also be provided.

**5.1. Evaluation of the auto-calibration method for device diversity**

To evaluate the effectiveness of the proposed auto-calibration method, we collected RSS data by using five mobile devices including an HTC Butterfly smart phone, a Samsung Note 1 smart phone, an Infocus M810 smart phone, an HTC EVO 3D smart phone, and a Samsung Note 2 smart phone. Nine APs are used in the test field. We collected RSS samples of the nine APs at 200 locations and repeated 10 times for each location at different time moments. The 200 locations are uniformly distributed within the testing environment. The distance between any two neighboring locations is 0.3 m. Also, the test field has 47 m in length and 15 m in width. Please refer to Figure 11 for the setting of the testing environment. We evaluated the proposed auto-calibration method in three aspects: (a) performance of landmark detection, (b) performance of landmark matching, and (c) accuracy of the estimated transfer function. We introduce the details in the succeeding paragraphs.

(a) Performance of landmark detection

To measure the performance of landmark detection, true positive rate \( (TPR) \) and false positive rate \( (FPR) \) are used.
TPR and FPR are defined as

\[
TPR = \frac{\# \text{ of True Positive Detection}}{\# \text{ of Positive Sample}}
\]  

(18)

\[
FPR = \frac{\# \text{ of False Positive Detection}}{\# \text{ of Negative Sample}}
\]  

(19)

to acquire both the positive and negative samples for landmark detection, the testing space is uniformly divided into 38 regions as shown in Figure 11. The distance between two neighboring regions is 1.5 m. A region with landmarks inside is treated as a positive sample, while that without any landmark is treated as a negative one. In the test field, there are nine Wi-Fi landmarks whose true locations are manually determined as the ground truth. Because we repeated 10 times of the testing procedure for landmark detection, there are totally 90 positive samples and 290 negative ones. If the location of a detected landmark happens to be located within the region of a positive sample, we define it as true detection; otherwise, the detected landmark is false detection.

By adjusting the combination of the contrast threshold \( \delta_c \) and the run-length parameter \( RL_{th} \) used in our landmark detection process, we could draw the receiver operating characteristic (ROC) curve of landmark detection to understand the performance of detection. In Figure 12, we show the ROC curves of landmark detection by using “Note2” and “EVO3D” for reference. From the curves, we may find that the optimal TPR is 0.9048 and FPR is 0.0215 for “Note2”; as for “EVO3D,” the optimal TPR and FPR are 0.7889 and 0.0483. Both devices show acceptable detection performance, even though “Note2” has better landmark detection than “EVO3D” because of the difference of Wi-Fi sensing abilities.

Furthermore, based on the ROC analysis, our system could find a balanced setting of the contrast threshold \( \delta_c \) and the run-length parameter \( RL_{th} \) to compromise the TPR and FPR. As the trade-off between TPR and FPR, we slightly preferred low FPR to high TPR. In fact, with less true landmark detection, our system could still estimate the parameters of the RSS transform model accurately, whereas false landmark detection may produce invalid RSS pairs and severely degrade the estimated transform model. In the experiments, we set \( \delta_c \) and \( RL_{th} \) as 7 and 9 correspondingly for all devices.

Next, we tested the performance of landmark matching between two devices. To demonstrate the result, we chose “Note2” as the training device and “EVO3D” as the user device. Before landmark matching, “Note2” was used to robustly detect the nine reference landmarks in the testing environment. For each landmark, the RSS signature is calculated and recorded in the database for later use.

Next, for performance evaluation, we used “EVO3D” to detect the landmarks and repeated the detection process 10 times. The number of correct detection and the number of false detection are listed in Table I. In the experiment, 71 Wi-Fi landmarks out of the 90 positive samples are correctly detected by “EVO3D”, but 14 false landmarks out of the 290 negative samples are also reported. For landmark matching, the RSS signature of each detected landmark is also calculated. Ideally, based on the measurement of landmark similarity, the true Wi-Fi landmarks would be kept, whereas false ones would be removed.

For landmark matching, we perform the similarity measurement between the tested landmark detected by “EVO3D” and the reference landmark detected by “Note2”. If the similarity is higher than the experimentally defined threshold, a matching pair is found. The result is then recorded in Table I. Sometimes, we may find multiple matching pairs for a tested landmark. At the time, we chose the pair producing the highest similarity as our result. In another case, a tested landmark may not find a suitable match because of the low similarity. This “no match” situation was also recorded in the “Nmatch” slots in Table I.

The match rate for each EVO3D-detected landmark is shown in Table I. In this experiment, our system does not
Landmark-based device calibration and region-based RSS modeling

### Table I. Matching table of Wi-Fi landmarks.

<table>
<thead>
<tr>
<th>WM_1</th>
<th>WM_2</th>
<th>WM_3</th>
<th>WM_4</th>
<th>WM_5</th>
<th>WM_6</th>
<th>WM_7</th>
<th>WM_8</th>
<th>WM_9</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>True_Det_EVO3D</td>
<td>9 7 8 10 6 9 8 7 7</td>
<td>71</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False_Det_EVO3D</td>
<td>1 2 1 3</td>
<td>2 1 1 0 3</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WM_1</td>
<td>9/10</td>
<td>0 0 0 0 0 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WM_2</td>
<td>0</td>
<td>7/9</td>
<td>0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WM_3</td>
<td>0 0 8/9</td>
<td>0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WM_4</td>
<td>0 0 0 10/13</td>
<td>0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WM_5</td>
<td>0 0 0 0 6/8</td>
<td>0 0 0 0 0 71/85</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WM_6</td>
<td>0 0 0 0 0 9/10</td>
<td>0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WM_7</td>
<td>0 0 0 0 0 0 8/9</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WM_8</td>
<td>0 0 0 0 0 0 7/7</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WM_9</td>
<td>0 0 0 0 0 0 0 0 7/10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

“WM. i/j” is the Wi-Fi landmark identity. “Sum” means summation. The third row “True_Det_EVO3D” and fourth row “False_Det_EVO3D” show the true detection number and false detection number for different landmarks. “a/b” in the table shows the matching rate. In this respect, “a” is the matching number and “b” is the number of detected landmark to be tested. Finally, “Nmatch” indicates no suitable match is established.

### Table II. The minimum (min), maximum (max), and mean location distance between landmark pairs. The unit is meter (m).

<table>
<thead>
<tr>
<th>Landmark ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
<td>0</td>
<td>0.3</td>
<td>0</td>
<td>0.3</td>
<td>0</td>
</tr>
<tr>
<td>Max</td>
<td>2.7</td>
<td>2.4</td>
<td>1.2</td>
<td>3.0</td>
<td>2.7</td>
<td>2.4</td>
<td>2.1</td>
<td>3</td>
<td>3.0</td>
</tr>
<tr>
<td>Mean</td>
<td>1.38</td>
<td>0.93</td>
<td>0.66</td>
<td>1.35</td>
<td>1.45</td>
<td>0.73</td>
<td>1.25</td>
<td>1.15</td>
<td>1.52</td>
</tr>
</tbody>
</table>

produce error matching, and hence, the matching rate of all non-diagonal elements in Table I is 0. On the other hand, for these false landmarks detected by EVO3D, our system may classify them as “no match” because of low landmark similarity. That is, our system has the ability to remove the false landmarks and reduce the false match rate. In practice, we prefer to set critical rules for landmark detection. While matching, we also select a high threshold for landmark matching. Both settings enable our system to achieve good matches and find the robust RSS pairs for transfer function estimation.

Furthermore, we hope that the distance of two matched landmarks is small. To know the performance, we measure the minimum, maximum, and mean distance between a tested landmark and its matched reference landmark. The results of the nine Wi-Fi reference landmarks are presented in Table II. The small distance shows that our method could find the reliable RSS pairs.

(c) Accuracy of the estimated transfer function

Finally, we assess the accuracy of the estimated transfer function. The five devices are tested in the experiment. Two different ways are adopted to evaluate the performance. First, we checked all the possible device pairs from the five devices. In total, there are 10 device pairs. For each pair, we estimated the RSS transfer function based on the proposed automatic device calibration method. We also calculated the RSS transfer function based on the manually selected RSS pairs and treated it as the ground truth. For comparison, we drew the estimated transfer function and the ground truth together. As shown in Figure 13, for the 10 possible device pairs, the estimated transfer functions could well approximate the ground truth.

On the other hand, we treated “Note 2” as the training device and the other four devices as the user devices. By using the proposed method, we estimated the RSS transfer function from a user device to the training device. For performance evaluation, we manually collected some RSS pairs. These RSS pairs are not involved in the estimation of the transfer function and majorly used as the ground truth. If an RSS pair is denoted as \((\text{RSS}_{\text{user}}, \text{RSS}_{\text{target}})\) and the estimated transfer function is denoted as \(T_{\text{user} \rightarrow \text{target}}(\text{RSS})\), we could measure the estimation error \(|\text{RSS}_{\text{target}} - T_{\text{user} \rightarrow \text{target}}(\text{RSS}_{\text{user}})|\) to evaluate the accuracy of the transfer function. In Table III, we present the mean and standard deviation of the estimation error to show the estimation results of the four user devices.

In Table III, we also list the results reported by two other RSS calibration methods for comparison. The first one determines the transfer function based on manually selected RSS pairs; the other one is a histogram-based device calibration method, which is a state-of-the-art algorithm proposed by Laoudias et al. [22]. The experimental results show that the transfer function estimated by the proposed method can produce less estimation errors than the histogram-based method.
Landmark-based device calibration and region-based RSS modeling

H. N. Manh, C.-C. Huang and L. Hsiao-Yi

Table III. The comparison of three radio signal strength calibration methods.

<table>
<thead>
<tr>
<th>User device</th>
<th>HTC Butterfly</th>
<th>Infocus M810</th>
<th>SS Note 1</th>
<th>HTC 3D EVO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration method</td>
<td>MEAN</td>
<td>STD</td>
<td>MEAN</td>
<td>STD</td>
</tr>
<tr>
<td>Manually selected</td>
<td>5.96</td>
<td>4.6</td>
<td>6.35</td>
<td>4.87</td>
</tr>
<tr>
<td>Landmark-based</td>
<td>7.41</td>
<td>5.33</td>
<td>6.65</td>
<td>5.08</td>
</tr>
</tbody>
</table>

“MEAN” and “STD” are the mean value and standard deviation of the estimation error. Unit is dB.

In principle, Laoudias et al. proposed an implicit device calibration method. Instead of collecting RSS pairs, they assumed the RSS histograms collected in the same indoor space by two devices are similar if a proper transfer function is applied. By finding the optimal RSS transfer function to meet the criterion, the method can address device diversity. However, the histogram assumption may be true only if the RSS samples are uniformly collected over the indoor space by two devices. In contrast, the proposed method explicitly extracts the RSS pairs to estimate the transfer function. By using Wi-Fi landmarks, our calibration process is more robust and independent to the RSS collection process.

5.2. Evaluation of region-based target localization

To evaluate the proposed region-based target localization, we set up the testing environment as Figure 14. In the environment, we collected the RSS samples at 215 training sites, which cover the test field uniformly. The distance between two neighboring sites is 1.5 m. Within the environment, we also used the same nine APs that produce the stable landmarks in the previous experiments. Moreover, we designed a testing route to test our system. As shown in Figure 14, 56 locations in the route are selected. Their true locations are manually recorded as the ground truth. Note that these testing locations are not the same as the training sites in the radio map in order to mimic the real application. We also indexed the tested locations to indicate the moving trajectory. In addition, in this experiment, we used “Note2” for both training and testing so that we can ignore the effect from device diversity.

Two external effects are considered in the localization experiments including noise effects and dynamic obstacles. We used several measurement metrics to evaluate the performance of the target localization. The first one is “mean error distance (MED)”, measuring the distance between the ground truth and the estimated location. A smaller MED means that the system has higher accuracy. The second is “variance of error distance (VED)”, indicating the spread of the error distance. A small VED means the predicted trajectory is smooth. We also used “maximum error distance (MaxED)” to evaluate the performance of our system in the worst case and used “cumulative distribution function (CDF)” of the error distance to judge the overall system performance. Finally, based on the localization accuracy, we compared our region-based RSS modeling method with Gaussian model (GM) [24], Gaussian process model (GPs) [28], and KDE [23]. We detail the localization experiments under different external effects in the succeeding sections.
5.2.1. Noise effects.

To test the system performance under different noise effects, we designed two experimental environments. The first one, the slow-moving case, has low-noise effects. In this setting, we opened all the doors for both the training and testing steps. Without obstacles, the signal-to-noise ratio is higher. In the testing phase, we moved slowly; at each location, five RSS samples were collected. The mean value of the five samples was treated as the observation for localization. With the mean operation, the noise effect could be distinctly reduced. The second environment, the fast-moving case, has higher noise effects. We closed all the doors for the training and testing steps in this environment. While testing the system, we moved along the testing route quickly and observed one RSS sample at each location. Without multiple observations at a location, the noise effects are higher. The testing trajectory is presented in Figure 14. Also, 30 RSS samples were collected at each location and used for training. We repeated the testing process four times. The results are shown in Table IV and Figure 15.

The experimental results show that all the systems perform better under the low-noise setting than the high noise one. Compared with the other three systems, the proposed region-based method is more robust to noise effects in terms of MED, VED, MaxED, and CDF metrics.

5.2.2. Dynamic obstacles and multi-modal radio signal strength variation.

We created a complicated wireless environment with dynamic obstacles and multi-modal RSS variations by opening or closing the office doors. In this environment, half doors remain open and the other half doors remain closed randomly. We also tested our system in the slow-moving case and the fast-moving case as mentioned previously. For each case, we repeated the testing procedure four times to evaluate our system. While training the radio map, at each location, we mixed 30 RSS samples collected under the open-door setting and other 30 RSS samples collected under the closed-door setting. The system performance and comparison are shown in Table V and Figure 16. The experiments are accomplished by using “Note2” for training and testing.

If we compare Table IV with Table V, the localization accuracy of all systems is reduced when the noise effects and the dynamic obstacles coexist in the environment. Moreover, the proposed method produces better results in localization accuracy. Here, we spotted some differences among those systems. For example, the GPs-based method has better performance than the KDE-based and GM-based methods when the noise effect is considered as the only factor. This is because the KDE-based and GM-based methods are discrete positioning systems. They limit the target to be located at discrete positions. This introduces quantization errors. However, if we take both noise and obstacles into consideration, the KDE-based method conversely shows better result than the GPs-based methods.
Landmark-based device calibration and region-based RSS modeling

H. N. Manh, C.-C. Huang and L. Hsiao-Yi

Figure 15. The comparison of 90% positioning error distance under (a) low-noise effects and (b) high-noise effects. The 90% positioning error distances of the proposed method are 1.4 (m) in both cases. CDF, cumulative distribution function; KDE, kernel density estimation; GPs, Gaussian process; MD-KDE, multi-dimensional kernel density estimation.

Table V. Performance evaluation under multi-modal radio signal strength variation and noise.

<table>
<thead>
<tr>
<th>Multi-modal variation:</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MED</td>
<td>3.763779</td>
<td>2.665414</td>
<td>3.040989</td>
<td>1.621278</td>
</tr>
<tr>
<td>VED</td>
<td>7.210243</td>
<td>7.407716</td>
<td>3.886224</td>
<td>1.080723</td>
</tr>
<tr>
<td>MaxED</td>
<td>13.5</td>
<td>10.81665</td>
<td>8.081536</td>
<td>4.080355</td>
</tr>
<tr>
<td>Multi-modal variation:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MED</td>
<td>3.753672</td>
<td>2.973628</td>
<td>3.212960</td>
<td>1.786694</td>
</tr>
<tr>
<td>VED</td>
<td>6.717785</td>
<td>8.12823</td>
<td>4.107537</td>
<td>0.880117</td>
</tr>
<tr>
<td>MaxED</td>
<td>13.5</td>
<td>13.5</td>
<td>8.178050</td>
<td>4.078949</td>
</tr>
</tbody>
</table>

The unit is meter (m)

GM, Gaussian model; KDE, kernel density estimation; GPs, Gaussian process; MED, mean error distance; VED, variance of error distance; MaxED, maximum error distance.

model. The reason is that the KDE-based method could well model the multi-modal RSS variation. As for our system, the MD-KDE method provides a systematic way to construct a continuous radio map and well model the RSS variation caused by noise effects and dynamic obstacles. Accordingly, the proposed system achieves better results.

5.3. Comparison of localization systems

Lastly, we tested our system in a complicated environment with challenges from device diversity, noise, and obstacles all the time. To test the accuracy of inter-device localization, we used “Samsung Note 2” as the training device and used “HTC butterfly”, “Infocus 810”, “Samsung Note 1”, and “HTC EVO3D” as user devices. In the experiment, we tested the system in the fast-moving (high noise) case with multi-modal RSS variation.

Moreover, we compared our system with some state-of-the-art methods including the methods of Dong et al. [15], Kjærgaard et al. [16], Huang et al. [19], Laoudias et al. [22], and Au et al. [13]. Among them, the methods of Dong et al., Kjærgaard et al., and Huang et al. proposed different RSS relative features for localization. Because RSS relative features are less sensitive to device diversity, these methods can localize targets without device calibration. For simplification, these relative features are abbreviated as “DIFF” [15], “Hyperbolic” [16], and “RE3” [19] separately. In contrast, the methods of Laoudias et al., Au et al., and our method need device calibration before positioning. For them, we applied the estimated transfer function to adjusting the RSS difference to overcome the variation from device diversity. In [22], the author proposed a histogram-based calibration method, abbreviated as “HistCalib”, to estimate the RSS transfer function. As for the method of Au et al. [13] and our method, we applied the proposed landmark-based method to estimating the RSS transfer function.

The system performance on target localization is shown in Table VI. For comparison, the localization results without device calibration are also listed for reference. Recognizably, the device diversity problem has critical effects on target localization. Without the RSS calibration process, the performance of the method of Au et al. our method is degraded apparently. After applying the estimated transfer function to adjusting the RSS difference, the system performance could be significantly improved. Furthermore, we could tell that the proposed method gives better accuracy than the method of Au et al., where the authors proposed a novel localization system based on compress sensing. However, the method of Au et al. applied a Gaussian model to representing the RSS distribution at a reference site. Instead, our system proposed a region-based RSS model, which is more robust to noise and obstacles as shown in Table V.
Figure 16. Comparison of 90% positioning error distance under the multi-modal radio signal strength variation. CDF, cumulative distribution function; KDE, kernel density estimation; GPs, Gaussian process; MD-KDE, multi-dimensional kernel density estimation.

(a) The slow motion case and (b) the fast motion case.

Table VI. Performance comparison of indoor localization systems.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HTC Med</td>
<td>4.7</td>
<td>4.68</td>
<td>4.68</td>
<td>3.29</td>
<td>9.2</td>
<td>5.37</td>
<td>2.06</td>
<td>1.95</td>
</tr>
<tr>
<td>VED</td>
<td>43.5</td>
<td>45</td>
<td>36</td>
<td>12</td>
<td>38.03</td>
<td>39.51</td>
<td>4.08</td>
<td>4.08</td>
</tr>
<tr>
<td>Butterfly</td>
<td>20.07</td>
<td>19.54</td>
<td>18.84</td>
<td>5.15</td>
<td>1</td>
<td>5.24</td>
<td>0.89</td>
<td>0.86</td>
</tr>
<tr>
<td>InFocus M810</td>
<td>4.26</td>
<td>4.49</td>
<td>4.47</td>
<td>3.57</td>
<td>7.84</td>
<td>5.52</td>
<td>2.12</td>
<td>1.93</td>
</tr>
<tr>
<td>VED</td>
<td>15.6</td>
<td>18.06</td>
<td>18.49</td>
<td>7.16</td>
<td>89.59</td>
<td>61.69</td>
<td>0.93</td>
<td>0.94</td>
</tr>
<tr>
<td>MaxED</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>16.5</td>
<td>35.27</td>
<td>39.53</td>
<td>4.08</td>
<td>4.08</td>
</tr>
<tr>
<td>Samsung Note 1</td>
<td>4.62</td>
<td>4.74</td>
<td>4.65</td>
<td>3.02</td>
<td>8.85</td>
<td>5.18</td>
<td>2.16</td>
<td>1.96</td>
</tr>
<tr>
<td>VED</td>
<td>16.81</td>
<td>20.88</td>
<td>16.4</td>
<td>4.85</td>
<td>126.94</td>
<td>61.06</td>
<td>0.95</td>
<td>0.86</td>
</tr>
<tr>
<td>MaxED</td>
<td>36</td>
<td>40.65</td>
<td>30.65</td>
<td>10.5</td>
<td>39.58</td>
<td>39.76</td>
<td>4.08</td>
<td>4.08</td>
</tr>
<tr>
<td>HTC EVO 3D</td>
<td>3.98</td>
<td>3.96</td>
<td>3.97</td>
<td>3.82</td>
<td>4.43</td>
<td>3.9</td>
<td>2.53</td>
<td>1.89</td>
</tr>
<tr>
<td>VED</td>
<td>21.1</td>
<td>24.05</td>
<td>21.1</td>
<td>15.07</td>
<td>39.51</td>
<td>38.03</td>
<td>7.36</td>
<td>4.08</td>
</tr>
</tbody>
</table>

*NoCalib* denotes no device calibration and “LanM” means the landmark-based calibration method. The unit is meter (m).

MED, mean error distance; VED, variance of error distance; MaxED, maximum error distance.

In addition, in Table VI, the proposed method outperforms “HistCalib” method [22]. According to the previous discussion and comparison in Table III, our landmark-based method could achieve better transfer function estimation than the histogram-based method. This may be the reason why our system shows better performance on target localization than the “HistCalib” method.

Results given by “DIFF” [15], “Hyperbolic” [16], and “RE3” [19] are also provided in Table VI for further comparison. These methods use relative RSS features for localization. However, inside a room, our experiment finds that the relative RSS features at most locations are similar. This property may reduce the discrimination of a system. Hence, the relative RSS features are more suitable for room-level positioning. Instead of using relative features, our system learns the RSS signature directly in a region-based model. By recording the possible RSS variation, our model keeps more RSS information for location discrimination and therefore provides better localization as shown in Table VI.

6. CONCLUSIONS

In this paper, we proposed an auto-calibration process and a region-based RSS modeling method to overcome the effects from device diversity, signal noise, and dynamic obstacles. To solve the device diversity problem, current methods manually collect RSS pairs between devices to train the transform function in order to amend the RSS variation. However, the supervised training process is not workable in terms of the system design. Instead of making use of the supervised training, our method collects the training RSS pairs based on the detection of Wi-Fi landmarks. By using Wi-Fi landmarks as reference points, our landmark-based method could automatically estimate the RSS transform function to effectively amend the RSS.
variation. Also, the proposed MD-KDE method enables our system to adopt the region information for positioning, which makes our system resistant to noise effects and dynamic obstacles. Experimental results have shown that our method can automatically calibrate the inter-device variation and precisely model the RSS variation caused by noise and obstacles. Compared with previous indoor localization methods, our system has demonstrated its superiority for target localization in a complicated testing environment.

In the paper, we have designed an RSS-based localization system for the scenario in a shopping mall. However, for different business scenarios, system designers may face different constraints for indoor localization. Also, for each scenario, many technical issues about indoor localization still exist. For example, in the museum guidance case, a better framework should well utilize the full potential of user devices and even combine more environmental information to achieve better localization accuracy. These issues are worth for further discussion and will be included in our future works.

ACKNOWLEDGEMENT

This work was supported by Ministry of Science and Technology of Taiwan under Grant No. 103-2622-E-194-009-CC3 and No. 104-2221-E-194-044-MY2

REFERENCES


27. Ferris B, Fox D, Lawrence N. WiFi-SLAM using Gaussian process latent variable models. In International Joint Conference on Artificial intelligence, Hyderabad-India, 2007; 2480–2485.


36. Levina E, Bickel P. The EarthMover’s distance is the mallows distance: some insights from statistics. In International Conference on Computer Vision, Vancouver-Canada, 2001; 251–256.
