This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Enhanced Fingerprinting and Trajectory Prediction for IoT Localization in Smart Buildings

Kai Lin, Min Chen, Jing Deng, Mohammad Mehedi Hassan, and Giancarlo Fortino

***Abstract*— Location service is one of the primary services in smart automated systems of Internet of Things (IoT). For various location-based services, accurate localization has become a key issue. Recently, research on IoT localization systems for smart buildings has been attracting increasing attention. In this paper, we propose a novel localization approach that utilizes the neighbor relative received signal strength to build the fingerprint database and adopts a Markov-chain prediction model to assist positioning. The approach is called the novel localization method (LNM) in short. In the proposed LNM scheme, the history data of the pedestrian's locations are analyzed to further lower the unpredictable signal fluctuations in a smart building environment, meanwhile enabling calibration-free positioning for various devices. The performance evaluation conducted in a realistic environment shows that the presented method demonstrates superior localization performance compared with well-known existing schemes, especially when the problems of device heterogeneity and WiFi signals fluctuation exist.**

***Note to Practitioners*—This paper was motivated by the problem of developing Internet of Things (IoT) localization systems for smart buildings but it also applies to other IoT applications that have location-based service ability. Existing approaches to design such systems generally utilize the received signal strength (RSS) from WiFi to build fingerprint for obtaining user's position. This paper suggests a novel technique, named novel localization method (LNM), that uses neighbor relative (NR) signal fingerprint and Markov chain for localization in smart buildings. NR-RSS is used as the fingerprint data to build radio map instead of absolute RSS. Meanwhile, Markov-chain model is applied to conduct the mobile device's trajectory analysis. In this paper, we evaluate LNM on different mobile devices with various system parameters. Then we show how the location of mobile device can be accurately computed against device heterogeneity and environmental dynamics. Extensive physical experiments suggest that LNM is feasible and reliable although it has not yet been evaluated on non-Android devices. In future research, we will address the design of IoT localization that has a wide variety of smart objects equipped with heterogeneous communication medium.**

***Index Terms*— Fingerprint, Internet of Things (IoT), Markov chain, mobile positioning, smart building.**

Manuscript received June 19, 2015; revised October 3, 2015 and December 28, 2015; accepted March 13, 2016. This paper was recommended for publication by Associate Editor Q.-S. Jia and Editor M. C. Zhou upon evaluation of the reviewer's comments. This work was supported in part by the Deanship of Scientific Research from King Saud University, Riyadh, Saudi Arabia, through the International Research Group under Grant IRG14-17, in part by the National Natural Science Foundation of China under Grant 61103234, and in part by the China Scholarship Council. *(Corresponding author: Kai Lin.)*

K. Lin is with the School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China (e-mail: link@dlut.edu.cn).

M. Chen is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: minchen@ieee.org).

J. Deng is with the Department of Computer Science, University of North Carolina at Greensboro, Greensboro, NC 27412 USA (e-mail: jing.deng@uncg.edu).

M. M. Hassan is with the College of Computer and Information Sciences, King Saud University, Riyadh 12372, Saudi Arabia (e-mail: mmhassan@ksu.edu.sa).

G. Fortino is with the Department of Informatics, Modeling, Electronics, and Systems, University of Calabria, Rende 87036, Italy (e-mail: g.fortino@unical.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2016.2543242

I. INTRODUCTION

INTERNET of Things (IoT) incorporates concepts from pervasive computing and enables interconnections of everyday objects equipped with ubiquitous intelligence, which becomes an integral part of the Internet [1], [2]. Thanks to rapid advances in underlying technologies, IoT is opening tremendous opportunities for novel applications that promise to improve the quality of our lives [3]. IoT has gained much attention from practitioners and researchers around the world, and spawned a wide variety of smart automated systems, such as smart buildings, smart homes, smart factories, and so on [4].

With the development of IoT, location-based service (LBS) has become increasingly important and extensively used. Designing effective and efficient location mechanisms for LBS is critical to, yet extremely difficult in, IoT scenarios, especially smart buildings. In a smart building, the widely used global positioning system (GPS) [5] becomes impractical because GPS signals cannot be transmitted through obstacles. Moreover, varieties of electronic devices deployed in smart buildings unavoidably produce considerable amounts of signal interference, greatly increasing the difficulty of system design for precise positioning in smart buildings.

Localization using the existing wireless communication infrastructure is regarded as an effective method with great potential. Recently, received signal strength (RSS) fingerprint approaches based on WiFi have gained popularity [6]. However, there are several glaring problems for traditional RSS fingerprint approaches. First, real RSS fingerprints at any locations always change with time. Besides, considering the hardware differences of mobile devices (e.g., smartphones, tablets, mobile robots, mobile smart objects), different mobile devices may get different measurement data, even for the exactly same RSS value. The noisy characteristics cause the measured samples to greatly deviate from those stored in the radio map. Second, in the process of matching, the localization system [7]–[9] need to access the RSS fingerprint database storing a great amount of data, which will take plenty of time. Although some systems [10], [11] adopt clustering of map locations to reduce the computational requirements, clustering algorithm also introduces error and extra complexity. Moreover, localization matching requires WiFi scanning, regarded as an energy-intensive process [12]. Since mobile devices are energy-constrained, it is critical to reduce the WiFi scanning process. Finally, building the fingerprint map requires an extensive and thorough site-survey process. To address the issues of labor-intensive and time-consuming calibration, the signal wave propagation model-based techniques [13] are proposed to estimate the RSS values at given locations. The main focus of these solutions is to build mathematical or theoretic models instead of manually tagging to calculate the RSS values of given locations.

In this paper, we propose a novel localization method (LNM) based on neighbor relative RSS (NR-RSS) and Markov-chain prediction algorithm, which mainly utilizes fingerprint-based technology and Markov-chain model to provide higher accuracy of localization with lower calibration requirement.

By observation of actual RSS value measurement in smart building environments, we find that NR-RSS, the difference of RSS between neighboring locations, compared with the absolute RSS (ARSS) values, is more robust to device heterogeneity and environmental dynamics. Therefore, we adopt NR-RSS instead of ARSS as fingerprint to build the radio map. Although the localization accuracy is significantly improved, it incurs extra computational overhead and the power consumption in each localization process, which is still a considerable burden for mobile devices. To solve these problems, we introduce Markov-prediction model (MPM) to assist positioning. MPM can be utilized for all moving objects equipped with smart devices such as the pedestrian carrying mobile devices, vehicles, and robots. Each position shift produced by movement has a certain probability (degree of purposiveness). Through statistics of the probability of movement, Markov process of object can be constructed for localization. To acquire the probability, mobile devices need accumulate enough history localization data during NR-RSS localization. In this way, the frequency of NR-RSS localization process is reduced, thereby the power consumption of mobile devices and computational requirements are also significantly lowered.

This paper makes the following contributions, addressing the issues mentioned above.

1) We analyze the changes of RSS at a region over time. Based on observations and records, we find that although the ARSS values at a region constantly change, the NR-RSS values of two locations do not vary much.
2) We make use of the Markov-chain model to assist matching the NR-RSS fingerprints at the localization phase.
3) We design a novel localization system for smart buildings that solves the matching problem caused by heterogeneous devices.
4) We have implemented our algorithm and evaluated it in a realistic environment scenario using different types of smartphones.

The rest of this paper is organized as follows. We present related work in Section II and analyze current fingerprint-based localization approaches in Section III. In Section IV, we introduce MPM and its application in our algorithm in detail. Section V describes the system architecture and the workflow of the proposed approach. In Section V, we evaluate the performance of our system through real-world experiments. Section VI draws conclusion from this work.

II. Related Work

Recently, wireless localization has become a focused research topic in the IoT context and a variety of solutions have been proposed. The IoT indoor localization approaches can generally be divided into two categories: passive method and active method [14]. In the passive localization approach, the tracked person (even a smart object) does not carry any electronic device and actively participate in the positioning process. In the active localization case, tracked person (even smart object) carries a physical electronic device, which can collect and process some information and send the results to a localization server for further processing. Relatively mature localization systems may be classified into three categories according to the system requirements and the used techniques:

1) location-sensor-infrastructure-based systems;
2) wave-propagation-modeling-based systems;
3) location-fingerprinting-based systems.

Location-sensor-infrastructure-based techniques typically rely on special-purpose infrastructures installed on walls or ceilings. Early work utilizing ultrasound [15] or short-range infrared [16] promised fine grained localization accuracy. Priyantha [17] proposed a method that uses radio and acoustic transmission and exploits time difference of arrival (TDOA) in the space. Radio frequency identification [18] technique is also extensively used. Topical systems explore multiple-input, multiple-output techniques using commodity access points (APs) and angle of arrival (AOA) to localize accurately [19]. TDOA and AOA are the most common methods used in an ultra-wideband localization system. Although these techniques provide high accuracy, their large-scale deployment is problematic due to the high deployment cost. Among the diverse approaches for indoor localization, the RF-signal-fingerprint-based approach is a significant portion of research work. Recent work proposed some novel forms of fingerprints such as FM Radio [20] and light color, while RSS fingerprint is more practical and widely applied, since the IEEE 802.11 APs are pervasively deployed nowadays. The fingerprint-based localization techniques are considered more attractive because of their advantages of low deployment cost and robustness in environment with interferences. However, building a fingerprint map would incur considerable cost and complexity. Moreover, the static radio map is vulnerable to environmental dynamics and device heterogeneity. Some works have focused on the effective method of constructing the fingerprint databases [21]. Others attempt to improve the localization accuracy of the RSS fingerprint mechanism. To reduce the calibration effort, some researchers focus the signal-wave-propagation-model-based techniques. These systems build mathematical or theoretical models instead of manually tagging to calculate

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

the given location RSS values [22]–[24]. Wang *et al.* [25] proposed a positioning technique based on a wave propagation model, expressing the mathematical relation between the distance from the transmitter to the receiver and RSS. The positioning system in [26] merges a wave propagation model using a polynomial regression and a reference points database. The computed location is shrunk to the knowledge of topology, effectively giving the final location. Model-based techniques do not require the training phase, but their localization accuracy is comparatively low. Today, mobile technology comprises highly sophisticated devices like smartphones with different inertial sensors. Therefore, there are plenty of studies describing the positioning system based on inertial measurements [27]. However, the major flaw of this kind of system is that the estimation error grows with time due to the typical drift of the inertial measurements [28]. For this reason, inertial measurements methods usually combine with other techniques to obtain higher accuracy.

III. Preliminary of Fingerprinting-Based Localization

A. Fingerprinting-Based Localization

In this section, we present the typical fingerprint-based IoT localization algorithms for smart buildings and analyze their shortcomings and limitations. Currently, most localization approaches adopt fingerprint matching scheme as the basic method for location estimation. The fingerprint-based localization mainly consists of two phases.

1) Phase 1 is called offline phase, or training phase. In this phase, the fingerprint maps of interest region are built using empirical measurement operations or a signal propagation model. The information on all positions and their corresponding RSS are collected to build the fingerprinting radio map in a database.
2) Phase 2 is called online phase, or localization phase. The mobile devices measure the RSS at an unknown position. Then, the measured RSS is matched with the fingerprint radio map in the database, and the best matching position information is identified.

These fingerprint-based localization systems usually take ARSS values as the fingerprint. The main challenge is the fact that the techniques are vulnerable to environmental dynamics and heterogeneous devices. To maintain the localization accuracy, the training process needs to periodically update the radio map, implying a huge overhead to be performed.

B. Neighbor Relative RSS

For fingerprint-based localization systems, the construction a robust and precise radio map is crucial. But there are two major issues limiting the accuracy of radio map. The first one is that the RSS value of an AP may vary with the environment and time. The other one is that, due to the heterogeneity of devices, RSS measurement data may obviously vary even when the actual signal strength remains the same. To overcome challenges, NR-RSS, the difference of RSS between neighbor locations, is adopted instead of ARSS to build fingerprint.

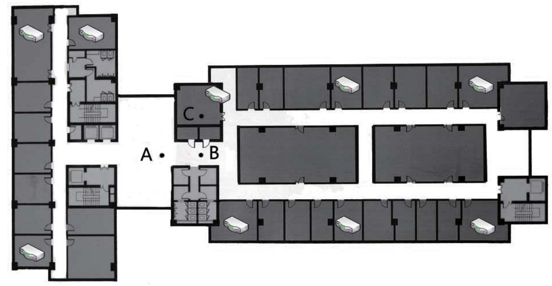


Fig. 1. Floor plan of localization environment.

As the environmental dynamics at close positions are considered almost the same, the influence of environment on RSS values at the positions is nearly identical, these RSS values tend to change synchronously. Besides, for a certain device, deviation of RSS values caused by device is approximately identical. Based on the characteristics, the influence of environmental dynamics and device heterogeneity can be almost eliminated through utilizing NR-RSS. Therefore, compared to ARSS, NR-RSS is more robust to device heterogeneity and environmental dynamics.

We compute the difference value of the two points RSS values at time i, namely, the NR-RSS

$$\text{NR-RSS}_i = \text{RSS}_i^A - \text{RSS}_i^B \tag{1}$$

where RSS_i^A and RSS_i^B stand for the RSS values of points A and B, respectively, at a certain time instant i. RSS_i^A and RSS_i^B can be represented in the following form:

$$\begin{aligned}\text{RSS}_i^A &= \left(\text{MR}_1^A, \text{MR}_2^A \ldots, \text{MR}_n^A\right)\\ \text{RSS}_i^B &= \left(\text{MR}_1^B, \text{MR}_2^B \ldots, \text{MR}_n^B\right).\end{aligned} \tag{2}$$

Here MR_j is the mean RSS value from jth AP, which are measured by surveying users. Moreover, AR_j^A and AR_j^B denote the mean measured RSS without environment and device influence at points A and B, respectively. Besides, Δ_d is the RSS variation caused by measurement device and Δ_e represents the RSS variation that the environment causes. We can derive the following equations:

$$\begin{aligned}\text{MR}_j^A &= \text{AR}_j^A - {\Delta_d}_j^A - {\Delta_e}_j^A\\ \text{MR}_j^B &= \text{AR}_j^B - {\Delta_d}_j^B - {\Delta_e}_j^B\end{aligned} \tag{3}$$

$$\begin{aligned}\text{MR}_j^A - \text{MR}_j^B &= \left(\text{AR}_j^A - \text{AR}_j^B\right)\\ &\quad -\left(\left({\Delta_e}_j^A + {\Delta_d}_j^A\right) + \left({\Delta_d}_j^B + {\Delta_e}_j^B\right)\right).\end{aligned} \tag{4}$$

As points A and B are quite close, the difference between ${\Delta_e}_j^A$ and ${\Delta_e}_j^B$ is negligible. Similarly, the same device is used to measure the RSS values at points A and B, and ${\Delta_d}_j^A$ and ${\Delta_d}_j^B$ are regarded identical. Therefore, at a certain time instant i, $\text{NR-RSS}_i = \text{RSS}_i^A - \text{RSS}_i^B$ is calculated through the following equation:

$$\begin{aligned}&\text{RSS}_i^A - \text{RSS}_i^B =\\ &\left(\left(\text{AR}_1^A - \text{AR}_1^B\right), \left(\text{AR}_2^A - \text{AR}_2^B\right), \ldots, \left(\text{AR}_n^A - \text{AR}_n^B\right)\right)\end{aligned} \tag{5}$$

where $\text{AR}_j^A - \text{AR}_j^B$ is always stable, enabling the stability of NR-RSS_i.

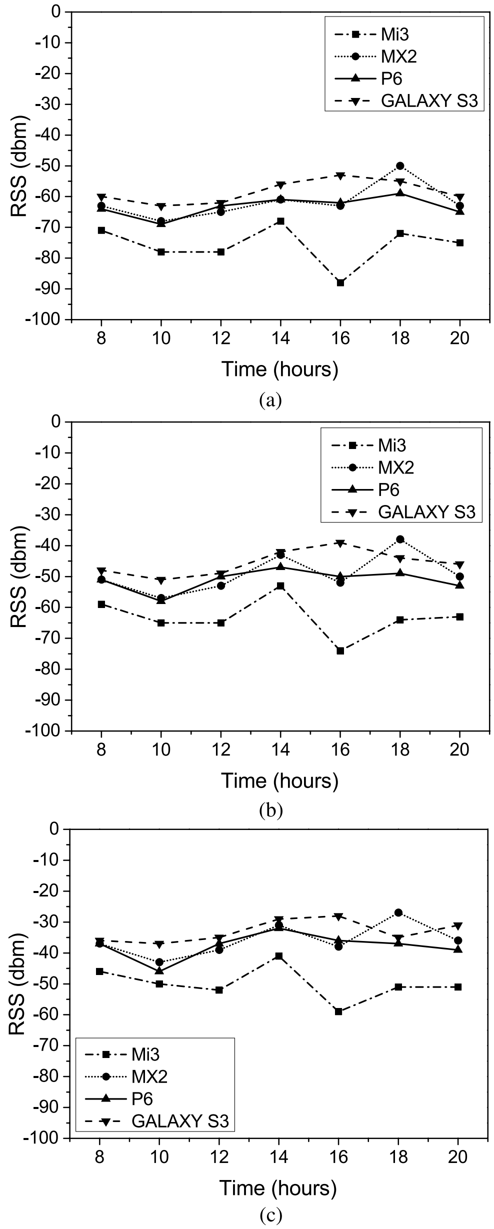


Fig. 2. Fluctuation of RSS value at locations A, B, and C. (a) Fluctuation of RSS value at location A. (b) Fluctuation of RSS value at location B. (c) Fluctuation of RSS value at location C.

Unlike typical fingerprint-based localization systems, we introduce a novel technique adopting NR-RSS to overcome the mentioned weakness. To verify the effectiveness of theoretical analysis, we performed the following experiment by collecting RSS values at points A and B using four different smartphones (Galaxy S3, MX2, Mi3, Ascend P6) throughout the day. The experiment was carried out on the ninth floor of a 17-story building. As shown in Fig. 1, points A and B are both in the corridor, and their distance is about 3 m, while point C is in the room and the distance between B and C is also about 3 m. The measured value of each point was collected ten times and the average taken as the RSS value to remove randomness. Fig. 2 shows the RSS values collected at the three locations throughout from 8:00 to 20:00. The experimental result demonstrates the obvious impact of environment dynamics and device heterogeneity on the RSS value.

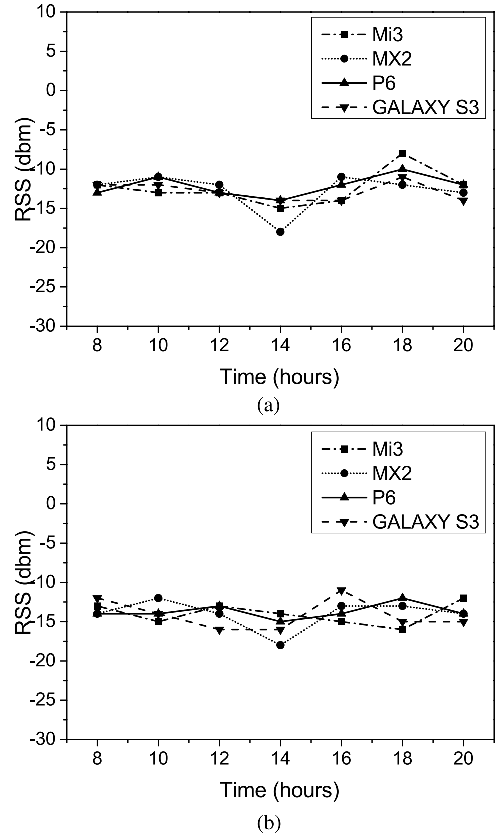


Fig. 3. NR-RSS fluctuation. (a) NR-RSS fluctuation of locations A and B. (b) NR-RSS fluctuation of locations B and C.

As seen from the experiment results, we learn that the RSS value of a particular location can fluctuate throughout the day. However, as Fig. 3 shows, the RSS difference values of points A and B stay relatively stable during the day. Considering that points A and B are both in the corridor and very close, these environmental dynamics produce the almost same effect, so the RSS values of points A and B change almost synchronously. While point B is in the corridor and point C is in room, in their surroundings, there exist some differences that influence RSS difference value of points B and C. But as shown in Fig. 3(b), such an influence is in an acceptable range. First, in the proposed scheme, plenty of APs are uniformly distributed in the environment, consequently weakening the influence of different surroundings. Second, the collected RSS values tend to stabilize through proofreading the average. Besides, neighbor locations adopted to calculate the RSS difference value locate mostly in the same environment.

As shown in Fig. 3, we notice that the NR-RSS$_i$ values of four different smartphones are close, while the four smartphones' RSS values at points A, B, and C are quite different at the same moment, as shown in Fig. 2. For the same smartphone, the collected RSS values may be always higher or lower than the real values. Therefore, the difference values of RSS values at the two points for different smartphones should be close to the same value. The result of NR-RSS experi-

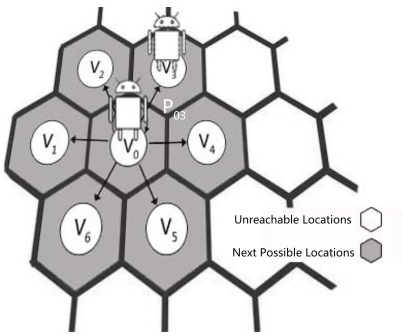


Fig. 4. Object's state transition of motion.

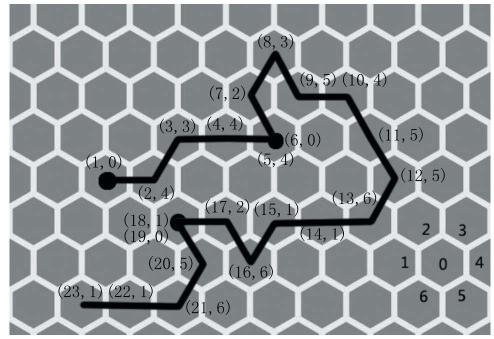


Fig. 5. Random pedestrian's trajectory.

ment supports our theoretical analysis, as shown in Fig. 3. Given these facts, we can easily draw the conclusion that the NR-RSS is more robust and stable against environment dynamics and the heterogeneity of devices. Consequently, we use NR-RSSs as the fingerprint data to build radio map instead of ARSSs.

IV. Markov-Prediction Model

Fingerprint-based localization systems must scan the surrounding RSS on each positioning at online localization phase. It is a high-energy-consuming operation for smart objects such as smartphones. It is more efficient to predict the object's movement by means of mathematical models. Thus, we apply the Markov-chain model to conduct object's trajectory analysis, which can reduce the energy consumption. In the Markov-chain model, localization object is likely to be moving objects equipped with mobile devices (such as robots and vehicles) in IoT. From the point of purposiveness, their movements have a certain probability (degree of purposiveness), complying with the principle of a Markov chain. In addition, the probability of object's movement can be obtained through the process of collecting and training. For example, an object has to go directly to a known location, the probability is close to 1, and the object can be predicted to move along the direction at next moment. In the proposed approach, as the NR-RSS matching localization runs, history data about object's movement are accumulated. Historical data can be used to calculate the probability of movement in Markov-chain model. Based on the probability of movement, we get initial state of Markov process for localization object, where the current location and the probability of movement can be combined to predict the next location.

A. Establishment of the Markov-Chain Model

In the Markov-chain model, each object's movement is modeled as a Markov process, and the probability of each movement only depends on the object's current position. Utilizing the probabilistic model, namely, Markov-chain mode, an object's movement can be predicted. The building map is modeled as a cellular structure and is equally divided into hexagonal cells. The object is located at a cell, represented as v_0 at time 0, as shown in Fig. 4. At time 1, it will either stay where it is or move to one of the six neighbors, v_1, v_2, v_3, v_4, v_5, and v_6, arranged as shown in Fig. 4. At time 2, it will also stand or move to one of the current location's six neighbors. This procedure is then iterated at times $3, 4, \ldots, t$. In the general model, we define n different status of the object's movement.

Due to the difference of the moving ability of object and the size setting of cell, the object may move outside the neighbors. Especially, in the MPM, we expect that the object travels at most the distance of one cell, which is affected by the moving ability of the object and the size of the cell. So, time unit depends on the moving ability of the mobile object

$$\mu^{(0)} = \left(\mu_1^{(0)}, \mu_2^{(0)}, \ldots, \mu_n^{(0)}\right). \tag{6}$$

Here $\mu_s^{(0)} (s = 1, 2, \ldots, n)$ denotes the probability in the state of s at time 0. And after k steps of status transition, the probability in the state of s is $\mu_s^{(k)}$. The $\mu_s^{(k+1)}$ is estimated by the following formula:

$$\mu_s^{(k+1)} = \sum_{i=1}^{n} \mu_s^{(k)} \cdot P_{ij}, \quad (s = 1, 2, \ldots, n) \tag{7}$$

j is also from 1 to n, and its matrix form is

$$\begin{aligned} &\left(\mu_1^{(k+1)} \mu_2^{(k+1)} \cdots \mu_n^{(k+1)}\right) \\ &\quad = \left(\mu_1^{(k)} \mu_2^{(k)} \cdots \mu_n^{(k)}\right) \cdot \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{pmatrix} \end{aligned} \tag{8}$$

namely

$$\mu^{(k+1)} = \mu^{(k)} \cdot P. \tag{9}$$

The elements of the transition matrix P are called transition probabilities. The transition matrix can be obtained by analyzing the object's motion historical data. Furthermore the well-known theorem is obtained [29]: for a Markov chain $(X_0, X_1, \ldots)$ with state space $v_0, \ldots, v_k$, initial distribution $\mu^{(0)}$ and transition matrix P, the distribution $\mu^{(n)}$ at time n satisfies

$$\mu^{(n)} = \mu^{(0)} \cdot P^n. \tag{10}$$

In the model, the state space corresponds to the behavior of the object movement, and it has seven values in total.

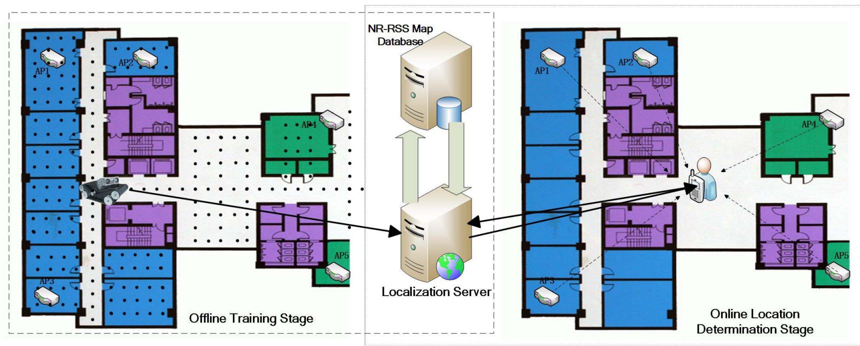


Fig. 6. Illustration of workflow in LNM.

B. Prediction by Markov Model

Let us take an example; a random object's moving trajectory is shown in Fig. 5. The illustration at the bottom-right is the orientation index corresponding to the object's movement state. The black circle represents where the object stayed. The orientation of object movement is represented as a pair of numbers. The first number in parenthesis is the sequence number and the second is the orientation index, namely, the object's movement state. The process continues until enough history data are collected at time i. The MPM is built to predict the object's following movement state. The state transition for heading is shown in Table I.

So we can get state-space $\{0, 1, \ldots, 6\}$ and transition matrix

$$P_i = \begin{pmatrix} 0 & 0 & 0.33 & 0 & 0.33 & 0.33 & 0 \\ 0.25 & 0.50 & 0 & 0 & 0 & 0 & 0.25 \\ 0 & 0.50 & 0 & 0.50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.50 & 0.50 & 0 \\ 0.25 & 0 & 0 & 0.25 & 0.25 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0.25 & 0.25 & 0.50 \\ 0 & 0.67 & 0.33 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The initial distribution is $\mu^{(i)} = \{0, 1, 0, 0, 0, 0, 0\}$. According to (9), the object's next motion state probability vector can be calculated

$$\begin{aligned} \mu^{(i+1)} &= \mu^{(i)} P_i \\ &= (0, 1, 0, 0, 0, 0, 0) \\ &\quad \cdot \begin{pmatrix} 0 & 0 & 0.33 & 0 & 0.33 & 0.33 & 0 \\ 0.25 & 0.50 & 0 & 0 & 0 & 0 & 0.25 \\ 0 & 0.50 & 0 & 0.50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.50 & 0.50 & 0 \\ 0.25 & 0 & 0 & 0.25 & 0.25 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0.25 & 0.25 & 0.50 \\ 0 & 0.67 & 0.33 & 0 & 0 & 0 & 0 \end{pmatrix} \\ &= (0.25, 0.50, 0, 0, 0, 0, 0.25). \end{aligned}$$

Thus, according to the prediction analysis, the object most likely directly moves left next time because the 1st index has the highest probability 50%. With the increasing of history data, the prediction model will be more accurate. In practical application, more history data are collected to improve the accuracy of further prediction. In our evaluation experiment, enough history records are collected to start predicting.

TABLE I
MOVEMENT DIRECTION STATE TRANSITION

Times \ Next State / Current State	0	1	2	3	4	5	6
0	0	0	1	0	1	1	0
1	1	2	0	0	0	0	1
2	0	1	0	1	0	0	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	1	0
5	0	0	0	0	1	1	2
6	0	2	1	0	0	0	0

V. LOCALIZATION ALGORITHM

In this section, we present the architecture and workflow of the proposed localization system leveraging the LNM algorithm in detail. The system mainly operates in two stages: offline training stage and online location determination stage, as shown in Fig. 6. In the offline stage, surveying users use smartphones to collect RSS data at all designated locations and then send them to the remote localization server. The server processes this information to get NR-RSS, building up the NR-RSS fingerprint map database. All interested locations are kept inside this database. In the online localization stage, the remote server that runs localization algorithm will return the location estimation to the device.

A. Fingerprint Collection

Surveying users equipped with smartphones measure RSS from the surrounding APs at the targeted smart building environment. These RSS values are used to calculate NR-RSS, which are stored at the corresponding positions in fingerprint map. The map is divided into equally spaced hexagon cells, and the entire map forms a honeycomb structure. The distance between the centers of adjacent hexagons is equal and close, and consequently the environment dynamics of adjacent cells are similar. Therefore, the cell is used to replace the exact geographic coordinate, where the RSS value of center of each cell is recorded to calculate NR-RSS value. Each cell has its unique Location ID. The cell spacing is crucial to the performance of the system. For the method, it is the ideal situation that the mobile devices receive the location information feedback from the server when they get

TABLE II
EXAMPLE OF NR-RSS FINGERPRINT MAP

Loc	ARSS	NR-RSS					
ID	$\vec{S}$	$\overrightarrow{RS_1}$	$\overrightarrow{RS_2}$	$\overrightarrow{RS_3}$	$\overrightarrow{RS_4}$	$\overrightarrow{RS_5}$	$\overrightarrow{RS_6}$

to the next neighbor cell. In this paper, the best cell spacing is empirically determined according to the normal walking speed of mobile users. The tuple M corresponding to each cell is

$$M = (L, \vec{S}) \tag{11}$$

where L represents the Location ID of the cell, and $\vec{S}$ is the RSS set collected by surveying-users at the real physical locations, which correspond to the current cells. $\vec{S}$ is stored as

$$\vec{S} = \{(\mathrm{ID}_1, \mathrm{MS}_1), (\mathrm{ID}_2, \mathrm{MS}_2), \ldots, (\mathrm{ID}_n, \mathrm{MS}_n)\}. \tag{12}$$

Here, ID represents the MAC address of the AP, MS is the mean value of the RSS values measured by the surveying-users, and n is the number of surrounding APs. Surveying-users should scan the WiFi signals and send such tuple M information of each cell to the fingerprint server.

B. NR-RSS Fingerprint Database Construction

Previous fingerprint-based localization systems build the fingerprint map radio using ARSS values. These systems are vulnerable to environment dynamics and heterogeneous devices. Thus, we devise a novel technique that makes use of NR-RSS values to overcome the above weakness. Let us start with the basic scenario. At some moment, a mobile device is at some place. In the next moment, the device should remain or arrive at one of the six orientations, regardless of the limitation factors of the real environment, such as walls and obstacles, etc. $V_0, V_1, V_2, \ldots, V_6$ stand for the seven states. At the training phase, surveying users scan the WiFi signals at their positions and their six neighbor points. The server receives and processes the information to build the fingerprint data of these positions in the NR-RSS fingerprint database.

Table II is an example of the NR-RSS fingerprint data. Loc stores the unique location ID value for each cell. ARSS stores the ARSS values. To improve the accuracy, we scan the WiFi signals several times and calculate the mean value as ARSS. $\vec{S}$ has the form in (12). The last and most important part is the NR-RSS column, which reflects the main idea of our system. We use the ARSS values of the device position and its six neighbors to calculate the difference values of the position and its neighbors as NR-RSS. $\overrightarrow{\mathrm{RS}_i}$ has the following form:

$$\overrightarrow{\mathrm{RS}_i} = \{(\mathrm{ID}_1, \mathrm{RS}_1), (\mathrm{ID}_2, \mathrm{RS}_2), \ldots, (\mathrm{ID}_n, \mathrm{RS}_n)\} \tag{13}$$

where i is the ith neighbor. We define the west as the first neighbor, increasing in a clockwise direction. One thing to note here is that a location does not always have six neighbors

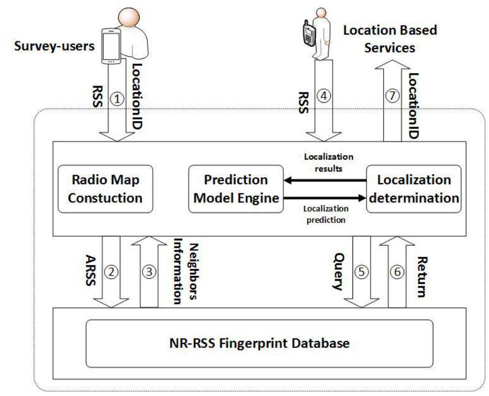


Fig. 7. Data flow in LNM.

because of the limitation of building structure. ID represents the MAC address of the AP, RS is the difference value of the position and its neighbor, and n is the number of surrounding APs.

The fingerprint server handles the raw data received from the clients, builds the NR-RSS fingerprint map, and stores it in the map database. The location information about the interest area is stored in the form described earlier in this paper. Our fingerprint map is robust and stable against environment dynamics and the heterogeneity of devices by using the NR-RSS.

C. Localization

In the online location determination stage, there are two localization methods. At the beginning of localization, as there are not enough movement data for setting up the MPM, NR-RSS matching method mainly works. The localization process runs as follows: the mobile devices can scan the WiFi signals and periodically send information to the localization server. The server combines the received RSS with the history neighbor RSS information to obtain the NR-RSS; then the NR-RSS is compared with all entry locations in the NR-RSS fingerprint database and the most matching one is determined to finish the location estimation. With the mobile device moving, the trajectory of its movement is constantly recorded. When the historical data reach a certain amount, location estimation is mainly performed by the MPM. That is to say, during this phase, location estimation is mainly based on the MPM and supplemented by NR-RSS matching. The data flow in our localization algorithm is shown in Fig. 7.

1) NR-RSS Matching Localization: When initializing the localization application, as there are no history data for the first time localization, the system uses the typical positioning solution RADAR [30] to obtain the initial location of mobile device. We call this process global search localization (GSL). Because GSL has to search all the locations in the fingerprint map, this operation is time-consuming. But this only happens at localizing the initial position. During the following localization, the system will utilize the NR-RSS match method and return location information in real time.

After initializing, system will localize mobile devices by our novel NR-RSS match solution. First, the accelerometer sensor in the smart device is utilized to judge whether the mobile device is in motion or stands. If the mobile device still stands,

Algorithm 1: NR-RSS Matching Localization Algorithm

```
Initialize (GSL)
loop
    if movement then
        Compute CND-RSS
        Match CND-RSS with NR-RSS fingerprints
        if D_min > δ then
            PGSL
        else
            Localization according to D_min
        end if
    else
        current location = last location
    end if
end loop
```

its current location is the same as the last localization outcome. Otherwise, when the mobile device moves to the next place, it sends the raw ambient RSS values to the fingerprint server. When history data are accumulated to a certain amount, the prediction model is built.

Second, for the processing and matching stage, the first step is to calculate the RSS difference value of the current location and its last adjacent location (LAL). This difference value is called current neighbor difference RSS (CND-RSS). LAL has the corresponding NR-RSS stored in the NR-RSS fingerprint map. Therefore, the next thing to do is to determine which neighbor of LAL the mobile device arrives at. A metric and a search methodology are used to compare the neighbors, obtaining the best matching one. Our solution is to compute the Euclidean distance of the CND-RSS and the prestored NR-RSSs of LAL in the fingerprint database, and then pick the neighbor location that minimizes the distance

$$D_i = ||\mathrm{RS}_i - \mathrm{CR}|| \tag{14}$$

where RS_i is the NR-RSS of the ith neighbor of LAL stored in the database and CR is CND-RSS. The neighbor that has smallest $D_{\min}$ is chosen to be the location estimation and gives feedback to the mobile device.

Our approach assumes that the mobile device will exactly move to one of the neighbors. However, due to the difference of moving speed, it is not possible that the mobile device will arrive exactly at the center of the next neighbor every time. Consequently, the error of the estimated position increases over time, and finally the mobile device might move to the other place rather than the neighbors. As shown in Fig. 4, an object moves from its current place, and at next moment, it may arrive at the shadow cells rather than the neighbors. Our algorithm determines such situation by using a threshold value δ

$$D_{\min} > \delta \tag{15}$$

where δ represents the threshold to determine whether the mobile device arrives at the neighbor or other place. The value of δ is set according to the actual fingerprint map state of the indoor environments. In the latter situation, we start a search method resembling the GSL described earlier. Instead of searching all the location items, we set the device place and its neighbors as the center and match outward expansion cells until we find the cell whose ARSS values are close to the observed value. In this way, only a small amount of location records need searching, and the location estimation is returned in real time. This process is termed pseudo GSL (PGSL). Algorithm 1 explains the process of NR-RSS matching localization.

2) Markov-Prediction Localization: At the beginning of the NR-RSS matching localization phase, there is a necessity for scanning the surrounding WiFi signals for each localization estimation process. This is quite a high-energy-consuming and time-consuming operation for mobile devices. Thus, MPM is adopted for localization. As the NR-RSS matching localization runs, movement history data are constantly recorded, and when it accumulates to certain amount, the Markov-prediction localization starts working.

To prevent Markov-prediction localization from causing the accumulated error, the NR-RSS match localization needs to be executed to verify the accuracy of Markov-prediction localization. When utilizing MPM on mobile devices, prediction result and current NR-RSS are sent to the server, where NR-RSS match localization is conducted to confirm whether prediction result is right. If the localization results estimated by NR-RSS matching and prediction model are nearly the same, server only returns confirming information, implying that the prediction result is valid. However, when they are different, there are two conditions: 1) if the result of NR-RSS matching is located at one of the six neighbors of the last location, the result will be sent to mobile devices as localization result and 2) if the result is located outside the six neighbors of the last location, the PGSL will be run to determine the mobile device position and sent to mobile devices. Moreover, as the prediction model has produced erroneous localization estimation, the recent movement history data will be deleted from the prediction model and the model will be rolled back to the last right status. To balance energy consumption and positioning accuracy, mobile devices should control the frequency of transmitting the request of verifying. In the early stage that MPM is built based on history data, once MPM localization is executed, the request of verifying will be sent to the server. With the increase of history data and the accumulation of accurate positioning, mobile devices may reduce the frequency of transmitting the request. However, when checking out that Markov-prediction localization produces localization error, mobile devices will increase the frequency of transmitting the request. The algorithm of building and working process of the MPM is described in Algorithm 2.

In the algorithm, the threshold C is set to decide whether movement history data are enough to support the effective Markov-prediction localization. C varies with motion object and motion situation, because different motion objects and even different motion situation of the same object need to accommodate different amounts of history data to achieve the valid MPM. For instance, when an object moves highly irregularly, more movement history data are needed to calculate the probability to switch to the different movement states.

Algorithm 2: Markov-Prediction Model Algorithm

```
loop
    if History records < C then
        NR-RSS Matching Localization
        History records++
    end if
end loop
if History records >= C then
    Build Markov Prediction Model(MPM)
    Localization by MPM
end if
if MPM localization result == NR-RSS localization result then
    History records++
else
    if NR-RSS localization result is the neighbor of the last location then
        Localization result== NR-RSS localization result
    else
        Run PGSL to get localization result
    end if
    History records−−
end if
```

VI. Performance Evaluation

This section discusses the results of real experiments to evaluate the performance of our proposed LNM. First, experimental testbed and the context of experiment are introduced in detail. Second, we evaluate the performance of the proposed algorithm under heterogeneous devices against other well-known algorithms.

A. Experimental Testbed

In this experiment, mobile devices carried by pedestrians move according to the given trajectory in a about 1000 m^2 area, where nine APs are deployed (shown as Fig. 6). Based on the different cell radius length in the experiment choosing optimal cell radius, the various number of calibration points is dynamically set. When cell radius is 1 m, 380 calibration points are adopted, while 5 m cell radius corresponds to 20 calibration points. However, in our experiment environment, 2 m is selected as the optimal cell radius through the determination of experiment, where the corresponding number of calibration points is set at 100. Thus, cell radius and the number of calibration points are, respectively, defined as 2 m and 100 in the subsequent experiment. Each calibration point is indicated by some stable RSS observations from all orientations. Each observation mainly contains RSS from all active APs. Besides, the thickness of wall between rooms is less than 10 cm, yielding certain interference. In the area, moving people and physical barriers always exist, which also causes fluctuations of RSS. Our experiment system includes smartphone client and server components. To carry out a proper evaluation of LNM in real environments, we implemented the client system on four different smartphones (Galaxy S3, MX2, Mi3, Ascend P6), which are Android smartphones equipped with WiFi (MX2 uses Flyme2.0 based on Android). The configuration information of these smartphones is shown in Table III. The server is developed with JAVA on Windows7 platform. We have collected realistic RSS in a WLAN environment illustrated in Fig. 1 from 6:00 to 22:00 of the day, and over seven days, keeping the executed scenarios as close to realistic as possible.

TABLE III
Smartphones Configuration Information

Smartphone (m)	Galaxy S3	Mi3	Ascend P6	MX2
WiFi	802.11 n/b/g	802.11 n/b/g	802.11 n/b/g	802.11 n/b/g
OS	Android4.0	MIUI V5	Android4.2	Flyme2.0
CPU	1.4GHz	1.8GHz	1.5GHz	1.6GHz
RAM	1GB	2GB	2GB	2GB

B. Experiments

The accuracy of our localization system is significantly influenced by various system parameters. To obtain an ideal location estimation, we should first find out the optimal parameter values. Among them, cell radius has a significant impact on the accuracy of localizing. First, each RSS value corresponding to a cell is used to calculate the NR-RSS value. Consequently, the selection of cell radius seriously influences the accuracy of NR-RSS localization. Second, in MPM, seven different statuses of the object movement are expected to locate at the adjacent cells; therefore, the size of cell radius is an important factor to achieve accurate prediction. We define the relation of location error and cell radius as $L_e = pR_c$. p value has two cases: $p < 1$ denotes that the location error is lesser than the cell radius, namely, real location and localization result are in the same cell. In this case, theoretically, we can shrink R_c to reduce the localization error. However, when R_c of cell is too small, the object may always move outside the neighbors, which can cause more serious inaccuracy of localization. Thus, to set the suitable R_c of cell, the device's computing ability and the time during a step need comprehensively be considered. $p > 1$ means that the localization result is outside the cell where the real position is located. The experiment shows that our approach ensures that location error is lesser than the cell radius. Another important parameter is the threshold δ for determining whether PGSL is implemented.

As we localize the cell instead of specific geographic coordinates, the cell spacing will impact the location error and correct rate significantly. Combined with the aforementioned mathematic analysis, the suitable cell radius should be chosen, enabling location error always less than cell radius ($p < 1$). Next, a series of experiments are conducted to choose the optimal cell radius. In the experiment, the floor plan is divided into many hexagon cells, and a set of localization is performed by altering the cell radius length from 1 to 5 m, while the user is walking at a speed of 1–1.5 m/s. Location error and correct rate are adopted to evaluate the performance corresponding to various cell radius length. Location error refers to the average distance between the position localized by LNM and the target location. And correct rate is the percentage of LNM hitting right cells times during 70 times of localization process. The

TABLE IV
IMPACT OF CELL SPACING

Cell Radius (m)	Location error (m)	Correct rate
1	0.8	41.2%
1.5	1.2	68.9%
2	1.5	80.3%
2.5	2	84.3%
3	2.4	86.1%
3.5	3.2	87.2%
4	3.8	89.1%
4.5	4	89.8%
5	4.6	91.2%

TABLE V
IMPACT OF δ

$\delta(dbm)$	Location Error (m)	Correct Rate
10	3.8	40.5%
15	3.4	43%
20	1.8	60%
25	1.5	80.3%
30	1.7	72.4%
35	2.6	80%

statistical results of the location error and the correct rate are shown in Table IV. With increased cell radius length, the correct rate becomes higher, while the location error also becomes bigger. In our experience, the location error of 1.5 m is an acceptable error with a quite high correct rate of 80.3%. Balancing the two measures, we think 2 m is a relatively ideal value for cell radius, and hence use it for the subsequent experiments.

As mentioned above, δ is used to judge whether the user goes outside the neighbor cells and to determine whether to execute PGSL. To choose suitable δ value, the fingerprint map data for different regions are analyzed, and a sequence of test experiments are performed to determine this threshold value. The result of the experiment for δ is shown in Table V. As the results show, when δ is 25 dbm, both the location error and correct rate have the best performance, and thus the value of δ is set to 25 dbm in the following evaluation experiments.

Moreover, to evaluate the energy consumption and the localization accuracy of using MPM, we respectively implement our system and a system merely based on NR-RSS localization, and four different smartphones are used to build fingerprint map and localize in our testbed. Energy ratio is defined to represent the ratio of energy consuming between the LNM localization system using NR-RSS and MPM and localization system only using NR-RSS. Besides, switch times represent the switch times from MPM prediction to NR-RSS localization in our system. During the 12 hours, energy ratio and switch times on every smartphone are recorded. As is shown in Table VI, on four different smartphones, the localization system using NR-RSS and MPM achieves remarkable energy efficiency, and the limited switch times also manifest the restricted degradation of localization accuracy.

We compare the number of appearance of the deviation of RSS value and NR-RSS value under different tolerable

TABLE VI
IMPACT OF USING MPM

Smartphone	Galaxy S3	MX2	Mi3	Ascend P6
Energy ratio(%)	34.5%	36.1%	38.2%	37.4%
Switch times	9	11	12	11

TABLE VII
NUMBER OF EXCEEDING TOLERABLE DEVIATION

Fingerprint	5%	10%	15%	20%
RSS	43	27	18	11
NR-RSS	11	8	5	1

TABLE VIII
NUMBER OF LOCALIZING AT THE WRONG CELL

Fingerprint	period 1	period 2	period 3	period 4
RSS	9	11	8	10
NR-RSS	2	1	0	1

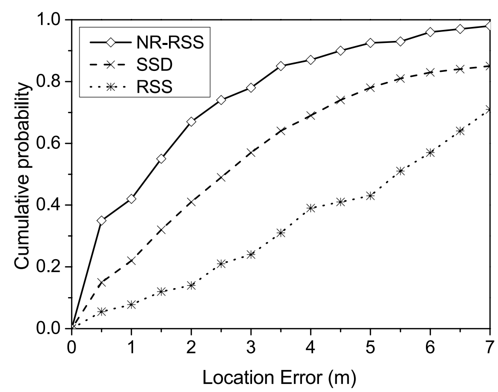


Fig. 8. Comparison of localization using different location fingerprint.

deviation. Tolerable deviation represents acceptable deviation degree of RSS value or NR-RSS. If the measured value exceeds the required tolerable deviation, the measured value cannot be used, and the value needs to be measured again. While, considering the different energy consumption demand of the systems, we set different tolerable deviation for different systems. As is shown in Table VII, under different tolerable deviations (5%, 10%, 15%, 20%), the number of exceeding tolerable deviation of RSS value and NR-RSS value among 100 cells in 5 h are presented.

To show the impact of wrong estimation, we compare the number that the localization result is at the wrong cell in NR-RSS and RSS matching localization. Table VIII shows that the number to localize at the wrong cell in four different periods (1 h).

In addition, in order to verify the performance of NR-RSS in localizing, we, respectively, implement the localization system based on NR-RSS and other location fingerprints, including signal strength difference (SSD) [30] and RSS [31]. As illustrated in Fig. 8, the localization accuracy of the three systems is compared; obviously, the localization system based on NR-RSS outperforms the localization system based on SSD and RSS.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

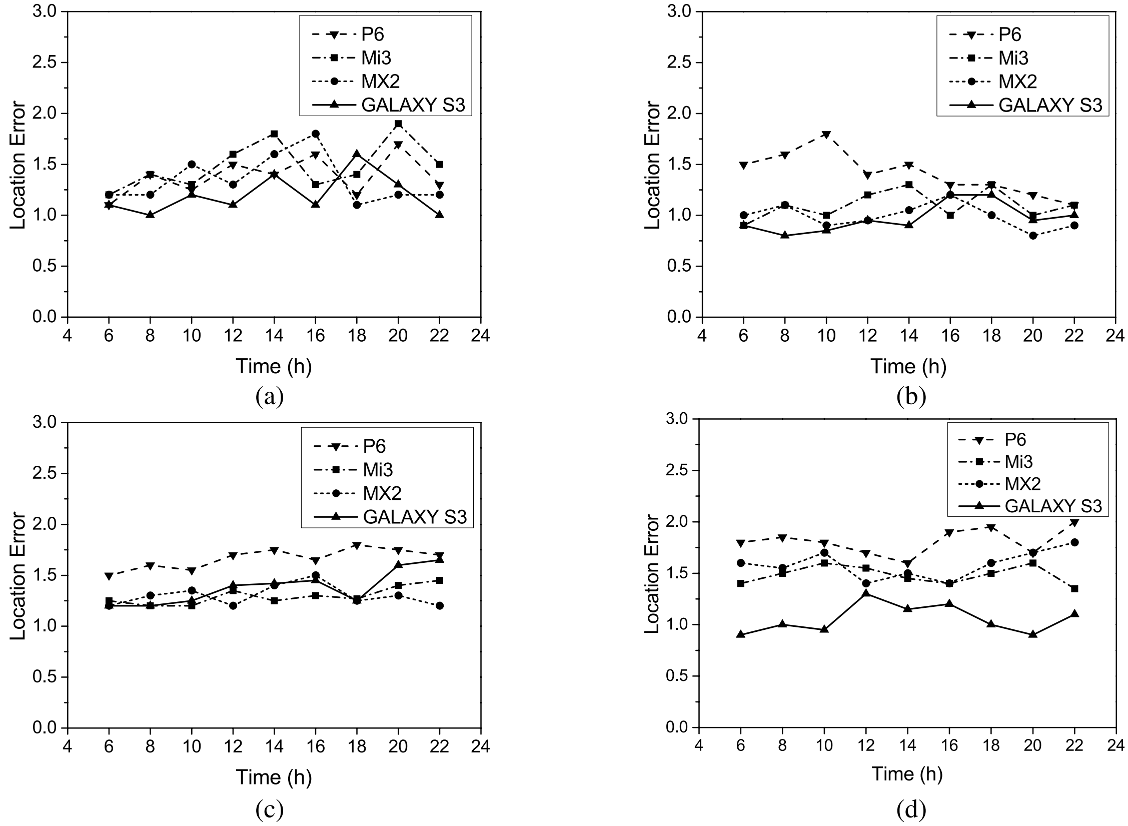


Fig. 9. LNM location error on different smartphones used for map building. (a) Ascend P6 used for map building. (b) Mi3 used for map building. (c) MX2 used for map building. (d) Galaxy S3 used for map building.

C. Impact of Device Heterogeneity

This evaluation mainly analyzes the performance of LNM with different devices to demonstrate that our algorithm works well under heterogeneous devices. In the experiment, four different smartphones were used to build fingerprint map and localize. For each time experiment, only one device was used for building fingerprint map, while four smartphones are used to localize each time. Localization was performed at different times of the day to check the performance of the systems against environmental dynamics. The situation of location error for LNM is shown in Fig. 9. Therefore, our algorithm can achieve stable localization accuracy against device heterogeneity.

D. Comparison Experiment

At last, we compare the performance of LNM and three other well-known systems: RADAR [32], WILL [33], and Zee [34]. These indoor localization systems are quite classical or a relatively new positioning solution. RADAR is an RF-based indoor localization system, which also uses RSS information collected at numerous positions. The approach does not consider the influence of environment dynamics and device heterogeneity on RSS value, causing inaccuracy of localization. The WILL system is based on off-the-shelf WiFi infrastructure, exploiting user motion trajectory to achieve the indoor localization. Utilizing the constructed RSS fingerprint and the floor plan database, the mapping between fingerprints and their measured locations is implemented to localize.

TABLE IX

COMPARISON OF LOCALIZATION ALGORITHMS

Algorithms	Fingerprint	Motion trajectory	Calibration effort	Prediction
RADAR	Yes	No	Yes	No
WILL	Yes	Yes	Yes	No
Zee	No	Yes	No	No
LNM	Yes	Yes	Yes	Yes

TABLE X

POWER CONSUMPTION AND AVERAGE LOCALIZATION TIME

Smartphone	RADAR	WILL	Zee	LNM
Mi3.Power(%)	1%	1.5%	1.2%	1%
Mi3.Time(s)	1.2	0.9	1.8	0.7
MX2.Power(%)	1.1%	1.5%	1.2%	1%
MX2.Time(s)	1.2	1	1.9	0.8
Galaxy S3.Power(%)	1%	1.4 %	1.2%	1%
Galaxy S3.Time(s)	1	0.8	1.1	0.7
Ascend P6.Power(%)	1%	1.4 %	1.2%	1%
Ascend P6.Time(s)	1	0.9	1.2	0.7

Zee system estimates the users' motion trajectory to enable the indoor localization without the calibration effort. The method utilizes various inertial sensors (e.g., accelerometer, compass, gyroscope) embedded in the mobile devices to localize, which simultaneously performs WiFi scanning.

As shown in Table IX, we make the analysis and comparison for the proposed algorithm and these three algorithms in terms of the following key parameters: fingerprint, motion trajectory,

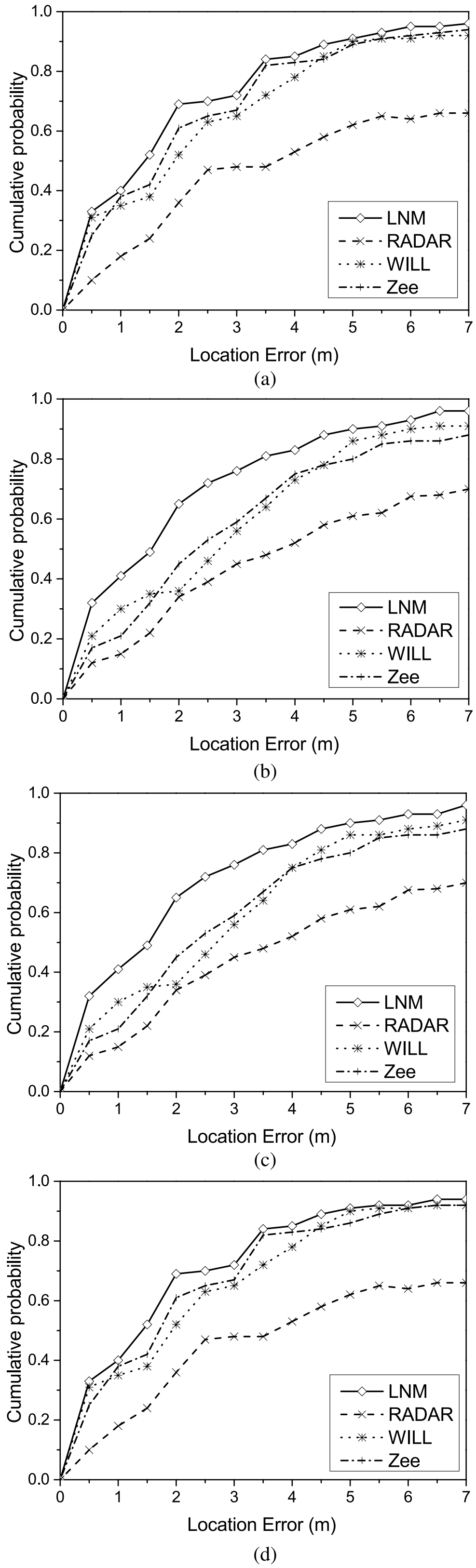


Fig. 10. CDF of location error comparison on different smartphones. (a) Mi3. (b) MX2. (c) Galaxy S3. (d) Ascend P6.

calibration effort, and prediction. Then, the proposed system and these three systems were run at Mi3, MX2, Galaxy S3, and Ascend P6. We compare the four systems on localization accuracy, power consumption, and system running time. Fig. 10 shows the cumulative density function (CDF) of the location error for the four techniques running at different mobile phones. LNM gives nearly 70%, 72%, 71%, and 70% accuracy for localizing the right place within 2.1 m at the platform of Mi3, MX2, Galaxy S3, and Ascend P6, respectively. Compared with the other three systems whose accuracy is relatively low and fluctuates largely, its performance is quite satisfying. The results indicate that the accuracy of LNM does not degrade with device heterogeneity and LNM can get the relatively high accuracy of 1.5 m. This reaffirms our belief that our method will work well in complex real-world scenarios.

In addition, to verify the availability of LNM, we also test the state of power consumption for 1 h and the average system running time for one time localization of LNM. The comparison results of the power consumption and the average system running time of the four systems are shown in Table X. We implement the four systems in the four smartphones (Mi3, MX2, Galaxy S3, Ascend P6), respectively, to ensure the fair and valid comparisons. As the results of comparison experiment show, our algorithm has quite good performance in the aspects of power consumption and system running time under the requirement of the stable high localization accuracy.

VII. Conclusion

In this paper, we have proposed and evaluated a novel method, named LNM, which uses NR signal fingerprint and Markov chain for localizing in smart building environment. The proposed fingerprint radio map building and localization techniques are based on the neighbor relationship. Our techniques provide robust and stable localization accuracy against device heterogeneity and environmental dynamics, which ensures the efficiency of localization. Experiments using heterogeneous smartphones have confirmed that LNM is feasible and reliable. LNM can achieve high localization accuracy with about 1.5 m error on the average. Our LNM outperforms other systems in the literature: RADAR, Zee, and WILL. As LNM can localize in real time with high accuracy, it has reached a level of maturity that allows for the practical realization of IoT localization solutions and services, and has potential for large-scale deployment in the IoT scenarios. For future work, we will evaluate other mobile devices such as aeroterrestrial drones (e.g., WiFiBot and Parrot) [35] in complex buildings, as such smart objects will be used in the future smart buildings for supporting many activities (cleaning, emergency, disabled people support, and so on).

References

[1] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] G. Fortino, A. Guerrieri, and W. Russo, “Agent-oriented smart objects development,” in *Proc. IEEE 16th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2012, pp. 907–912.

[3] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, “Convergence of MANET and WSN in IoT urban scenarios,” *IEEE Sensors J.*, vol. 13, no. 10, pp. 3558–3567, Oct. 2013.

[4] G. Fortino, A. Guerrieri, G. M. P. O’Hare, and A. Ruzzelli, “A flexible building management framework based on wireless sensor and actuator networks,” *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 1934–1952, Nov. 2012.

[5] E. Kaplan, C. Hegarty, Eds., *Understanding GPS: Principles and Applications*. Norwood, MA, USA: Artech House, 2005.

[6] J.-Y. Lee, C.-H. Yoon, H. Park, and J. So, "Analysis of location estimation algorithms for Wifi fingerprint-based indoor localization," in *Proc. 2nd Int. Conf. Softw. Technol.*, vol. 19, 2013, pp. 89–92.
[7] C. Koweerawong, K. Wipusitwarakun, and K. Kaemarungsi, "Indoor localization improvement via adaptive RSS fingerprinting database," in *Proc. Int. Conf. IEEE Inf. Netw. (ICOIN)*, Jan. 2013, pp. 412–416.
[8] J. A. G. Martin, A. V. M. Rodríguez, E. D. Zubiete, O. R. Romero, and S. M. Guillén, "Fingerprint indoor position system based," *J. Netw.*, vol. 8, no. 1, pp. 37–44, Jan. 2013.
[9] Y. Shu, P. Coué, Y. Huang, J. Zhang, P. Cheng, and J. Chen, "G-Loc: Indoor localization leveraging gradient-based fingerprint map," in *Proc. IEEE Conf. IEEE Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr./May 2014, pp. 129–130.
[10] C.-W. Lee, T.-N. Lin, S.-H. Fang, and Y.-C. Chou, "A novel clustering-based approach of indoor location fingerprinting," in *Proc. IEEE 24th Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2013, pp. 3191–3196.
[11] X. Hu, J. Shang, F. Gu, and Q. Han, "Improving Wi-Fi indoor positioning via AP sets similarity and semi-supervised affinity propagation clustering," *Int. J. Distrib. Sensor Netw.*, vol. 2015, Jan. 2015, Art. no. 109642.
[12] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. USENIX Annu. Tech. Conf.*, 2010, p. 21.
[13] A. Bose and C. H. Foh, "A practical path loss model for indoor WiFi positioning enhancement," in *Proc. 6th Int. Conf. IEEE Inf., Commun. Signal Process.*, Dec. 2007, pp. 1–5.
[14] G. Deak, K. Curran, and J. Condell, "A survey of active and passive indoor localisation systems," *Comput. Commun.*, vol. 35, no. 16, pp. 1939–1954, Sep. 2012.
[15] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. ACM 6th Annu. Int. Conf. Mobile Comput. Netw.*, 2000, pp. 32–43.
[16] P. W. Chen, K. S. Ou, and K. S. Chen, "IR indoor localization and wireless transmission for motion control in smart building applications based on Wiimote technology," in *Proc. IEEE SICE Annu. Conf.*, Aug. 2010, pp. 1781–1785.
[17] N. B. Priyantha, "The cricket indoor location system," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2005.
[18] D. A. Tesch, E. L. Berz, and F. P. Hessel, "RFID indoor localization based on Doppler effect," in *Proc. 16th Int. Symp. IEEE Quality Electron. Design (ISQED)*, Mar. 2015, pp. 556–560.
[19] S.-H. Yang, H.-S. Kim, Y.-H. Son, and S.-K. Han, "Three-dimensional visible light indoor localization using AOA and RSS with multiple optical receivers," *J. Lightw. Technol.*, vol. 32, no. 14, pp. 2480–2485, Jul. 15, 2014.
[20] V. Moghtadaiee and A. G. Dempster, "Indoor location fingerprinting using FM radio signals," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 336–346, Jun. 2014.
[21] N. Swangmuang and P. V. Krishnamurthy, "On clustering RSS fingerprints for improving scalability of performance prediction of indoor positioning systems," in *Proc. 1st ACM Int. Workshop Mobile Entity Localization Tracking GPS-Less Environ.*, 2008, pp. 61–66.
[22] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz, "A probabilistic room location service for wireless networked environments," in *Proc. 3rd Int. Conf. Ubiquitous Comput. (UbiComp)*, 2001, pp. 18–34.
[23] L. Kanaris, A. Kokkinis, G. Fortino, A. Liotta, and S. Stavrou, "Sample size determination algorithm for fingerprint-based indoor localization systems," *Comput. Netw.*, pp. 1286–1389, 2016, doi: 10.1016/j.comnet.2015.12.015.
[24] M. Youssef and A. Agrawala, "The Horus location determination system," *Wireless Netw.*, vol. 14, no. 3, pp. 357–374, Jun. 2005.
[25] Y. Wang, X. Jia, H. K. Lee, and G. Y. Li, "An indoors wireless positioning system based on wireless local area network infrastructure," in *Proc. 6th Int. Symp. Satellite Navigat. Technol. Including Mobile Positioning Location Services*, 2003, p. 54.
[26] A. Smailagic and D. Kogan, "Location sensing and privacy in a context-aware computing environment," *IEEE Wireless Commun.*, vol. 9, no. 5, pp. 10–17, Oct. 2002.
[27] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1281–1293, Jul. 2013.
[28] Q. Chang, S. Van de Velde, W. Wang, Q. Li, H. Hou, and S. Heidi, "Wi-Fi fingerprint positioning updated by pedestrian dead reckoning for mobile phone indoor localization," in *Proc. China Satellite Navigat. Conf. (CSNC)*, vol. 342, 2015, pp. 729–739.
[29] W. R. Gilks, *Markov Chain Monte Carlo*. New York, NY, USA: Wiley, 2005.
[30] A. K. M. Mahtab Hossain, Y. Jin, W.-S. Soh, and H. N. Van, "SSD: A robust RF location fingerprint addressing mobile devices' heterogeneity," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, pp. 65–77, Jan. 2013.
[31] K. Kaemarungsi and P. Krishnamurthy, "Properties of indoor received signal strength for WLAN location fingerprinting," in *Proc. 1st Annu. Int. Conf. Mobile Ubiquitous Syst., Netw. Services (MobiQuitous)*, San Diego, CA, USA, Aug. 2004, pp. 14–23.
[32] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 2, Mar. 2000, pp. 775–784.
[33] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: Wireless indoor localization without site survey," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 839–848, Apr. 2013.
[34] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.
[35] P. Pace, G. Aloi, and G. Fortino, "An application-level framework for UAV/rover communication and coordination," in *Proc. IEEE 19th Int. Conf. IEEE Comput. Supported Cooperat. Work Design (CSCWD)*, May 2015, pp. 229–233.

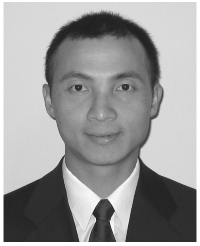
Kai Lin received the M.S. and Ph.D. degrees in communication engineering from Northeastern University, Shenyang, China.

He is currently an Associate Professor with the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. He has authored or co-authored over 50 papers in international journals and conferences. His current research interests include wireless communication, data mining and data fusion, big data analysis, mobile ad hoc network, cyber physical system, and sensor network.

Dr. Lin is an Associate Editor of the *Recent Patents on Telecommunications*, and served as the Editor or Guest Editor for several journals and special issues. He also served as the General Chair, the Technical Program Committee (TPC) Chair, and the Publicity Chair for many international conferences. He also participated in more than 40 international TPCs.

Min Chen was an Assistant Professor with the School of Computer Science and Engineering, Seoul National University, Seoul, South Korea, from 2009 to 2012, where he was a Post-Doctoral Fellow for one and a half years. He was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, for three years. He is currently a Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, where he is the Director of the Embedded and Pervasive Computing Laboratory. He has more than 180 paper publications.

Mr. Chen received the Best Paper Award from the IEEE International Conference on Communications in 2012 and the Best Paper Runner-Up Award from the Fifth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness in 2008.



Jing Deng received the B.E. and M.E. degrees in electronics engineering from Tsinghua University, Beijing, China, in 1994 and 1997, respectively, and the Ph.D. degree from the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA, in 2002.

He served as a Research Assistant Professor with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY, USA, from 2002 to 2004. He was with the Department of Computer Science, University of New Orleans, New Orleans, LA, USA, from 2004 to 2008. He visited the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA, and the Wireless Information Network Laboratory, Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ, USA, in 2005. He is currently an Associate Professor with the Department of Computer Science, University of North Carolina at Greensboro, Greensboro, NC, USA. His current research interests include wireless network and security, information assurance, mobile ad hoc networks, and social networks.

Dr. Deng received the Test-of-Time Award presented by the ACM Special Interest Group on Security, Audit and Control in 2013. He is an Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.

Mohammad Mehedi Hassan received the Ph.D. degree in computer engineering from Kyung Hee University, Seoul, South Korea, in 2011.

He is currently an Assistant Professor with the Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He has authored or co-authored over 70 publications, including refereed IEEE/ACM/Springer journals, conference papers, books, and book chapters. His current research interests include cloud collaboration, media cloud, sensor cloud, mobile cloud, Internet protocol television, and wireless sensor network.

Giancarlo Fortino received the Laurea (B.S. and M.S.) and Ph.D. degrees in computer engineering from the University of Calabria, Rende, Italy, in 1995 and 2000, respectively.

He has been with the Department of Informatics, Modeling, Electronics and Systems, University of Calabria, since 2006, where he is currently an Associate Professor of Computer Engineering. He holds the Scientific National Italian Habilitation for Full Professor, and is also an Adjunct Full Professor of Computer Engineering with the Wuhan University of Technology, Wuhan, China, in the framework of high-end foreign experts in China and an Adjunct Senior Researcher with the Italian National Research Council. He is the Co-Founder and CEO of SenSysCal S.r.l., a spin-off of the University of Calabria, where he is engaged in the advanced applied research and development of IoT systems. He has authored about 250 publications in journals, conferences, and books. His current research interests include distributed computing, wireless sensor networks, software agents, IoT technology, and cloud computing.

Dr. Fortino is currently an Associate Editor of the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, the IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, *Information Fusion*, *Engineering Application of Artificial Intelligence*, the *Journal of Network*, and *Computer Applications*.