

Article

# An Indoor Pedestrian Positioning Method Using HMM with a Fuzzy Pattern Recognition Algorithm in a WLAN Fingerprint System

Yepeng Ni <sup>1,\*</sup>, Jianbo Liu <sup>2</sup>, Shan Liu <sup>2</sup> and Yaxin Bai <sup>3</sup>

<sup>1</sup> Computer and Network Center, Communication University of China, No. 1 Dingfuzhuang East Street, Chaoyang District, Beijing 100024, China

<sup>2</sup> Information Engineering Institute, Science and Technology Department, Communication University of China, No. 1 Dingfuzhuang East Street, Chaoyang District, Beijing 100024, China; lj@uc.edu.cn (J.L.); liushan@cuc.edu.cn (S.L.)

<sup>3</sup> School of Journalism, Faculty of Journalism and Communication, Communication University of China, No. 1 Dingfuzhuang East Street, Chaoyang District, Beijing 100024, China; rhbyx@cuc.edu.cn

\* Correspondence: nyp\_2010@cuc.edu.cn; Tel.: +86-131-4649-7114

Academic Editor: Fan Ye

Received: 4 July 2016; Accepted: 5 September 2016; Published: 8 September 2016

**Abstract:** With the rapid development of smartphones and wireless networks, indoor location-based services have become more and more prevalent. Due to the sophisticated propagation of radio signals, the Received Signal Strength Indicator (RSSI) shows a significant variation during pedestrian walking, which introduces critical errors in deterministic indoor positioning. To solve this problem, we present a novel method to improve the indoor pedestrian positioning accuracy by embedding a fuzzy pattern recognition algorithm into a Hidden Markov Model. The fuzzy pattern recognition algorithm follows the rule that the RSSI fading has a positive correlation to the distance between the measuring point and the AP location even during a dynamic positioning measurement. Through this algorithm, we use the RSSI variation trend to replace the specific RSSI value to achieve a fuzzy positioning. The transition probability of the Hidden Markov Model is trained by the fuzzy pattern recognition algorithm with pedestrian trajectories. Using the Viterbi algorithm with the trained model, we can obtain a set of hidden location states. In our experiments, we demonstrate that, compared with the deterministic pattern matching algorithm, our method can greatly improve the positioning accuracy and shows robust environmental adaptability.

**Keywords:** pedestrian positioning; fuzzy pattern recognition algorithm; RSSI variation trend; hidden markov model; smartphone; fingerprint system

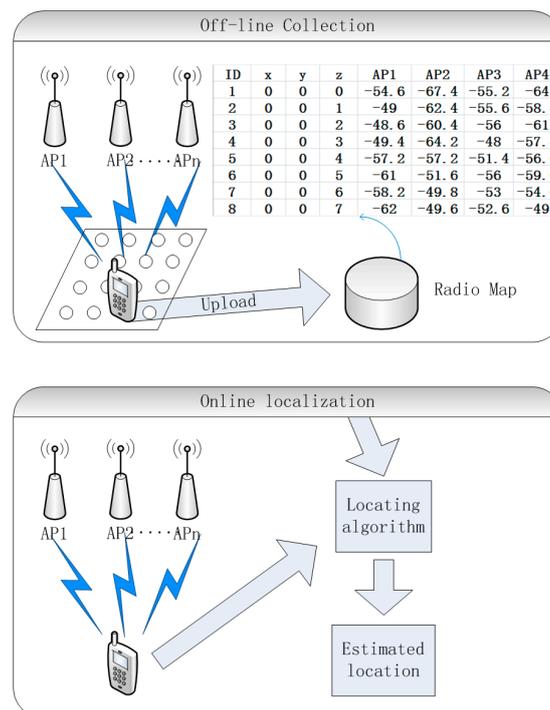
## 1. Introduction

Nowadays, with the rapid development of smartphones and wireless networks, people can conveniently access location-based services (LBSs) with mobile applications. Location information has become the cornerstone of several new technologies and applications including big data, cloud computing, Internet of things, and online-to-offline. The accurate positioning of the users is the key to providing LBS guaranteed by quality of experience (QoE).

The localization technology is mainly divided into outdoors and indoors. The outdoors localization technology is relatively mature, and mainly includes the Global Positioning System (GPS) and cellular positioning systems [1]. However, for indoor localization, GPS signals cannot reach the receivers and the positioning accuracy is too low to support indoor services. For this purpose, people have done a lot of research with different technologies, and developed RFID positioning systems, infrared ray positioning systems, Bluetooth positioning systems, ZigBee positioning systems,

ultrasonic positioning systems, vision positioning systems, voice recognition positioning systems, and WLAN (Wireless Local Area Network) fingerprint positioning systems, etc. [2]. Among these, the WLAN fingerprint positioning technology has become a research hotspot in the indoor localization field on account of the popularity of WLAN signals in indoor environments.

The WLAN fingerprint system is an economical and practical positioning system. Bahl et al. [3,4] have introduced the basic principle of WLAN fingerprinting. The system workflow is shown in Figure 1. In the off-line phase, users sequentially record Received Signal Strength Indicator (RSSI) values with current position coordinates into a database; this process is often called calibration and the database is called a radio map. In the on-line localization phase, a user enters into a WLAN environment with terminals and samples the RSSI. The user position coordinates are estimated by comparing the RSSI sample against the radio map to determine the closest signal distance.



**Figure 1.** The workflow of a WLAN (Wireless Local Area Network) fingerprint system.

Although there are some classical WLAN fingerprint systems, the research objective is always focused on a static target. There are few papers discussing pedestrian positioning, which is more relevant in a real environment. Due to distance variation and human body interference, the RSSI of moving targets will fluctuate over a large range, especially at a relatively slow speed [5]. Moreover, the sampling time is limited, which means we cannot use the averaging method to reduce the influence of noise because of the too small sample size. The above factors lead to severe distortions in the WLAN fingerprint systems that sample RSSI, thus seriously compromising the positioning accuracy.

In order to offer users LBS services with better positioning accuracy in a real environment, in this paper, a pedestrian positioning method using a Hidden Markov Model (HMM) [6] with a Fuzzy Pattern Recognition (FPR) algorithm in a WLAN fingerprint system is proposed. Firstly, we use the FPR algorithm to rebuild the pre-existing radio map; then, we utilize a forward-backward algorithm to train the HMM using the planned path and the rebuilt radio map; finally, we can utilize a Viterbi algorithm to estimate the pedestrian position using the constructed HMM. This method take full advantage of the prior information of the WLAN fingerprint system and user information (RSSI, user trajectories), and our experiments indicate that this method can effectively improve the positioning accuracy, consequently improving the user experience.

This paper is organized as follows: Section 2 briefly introduces the related work. Section 3 proposes a fuzzy pattern recognition algorithm to improve pedestrian positioning accuracy. Section 4 elaborates on how to train the HMM with the FPR algorithm in the fingerprint system, and use the Viterbi algorithm to estimate pedestrian position. Section 5 describes the experimental testbed and conducts the experiment before the experimental results are analyzed and compared. The last section concludes the research contents in this paper and gives some suggestions of future research.

## 2. Related Work

In recent years, research on indoor fingerprint positioning has been conducted actively. We will discuss some representative works here, especially the ones that employ HMM in their systems. To reduce the calibration work, Wallbaum et al. [7,8] employed a radio propagation model and HMM to quickly implement an indoor positioning system. Given some independent assumptions, it uses a discrete probability distribution to describe all hypothetical positions, and this probability distribution is only updated when new RSSI data is collected or the user has moved. The position is estimated by weighing the different hypotheses. This method needs high computation overhead on user terminals, and the accuracy is not good due to the fact the radio propagation model can hardly model realistic environments.

Seitz et al. [9] improve upon the updating method mentioned by Wallbaum et al. [8], and consider the angle factor in the movement updating procedure. Using this method, the proposed HMM directly includes the movement measurements. Although this method could achieve a relatively better positioning accuracy, the experiments were carried out in a clean environment such as a market with no visitors and it was assumed that the fluctuating RSSI followed a Gaussian distribution. It did not consider that there is a mismatch between RSSI measurements in movement and the fingerprint database.

Liu et al. [10] presented a lightweight and robust indoor positioning system. To reduce the calibration work, this system employs a Weibull distribution to represent the distribution of the RSSI over time. Then, it uses an accelerometer sensor to provide the information on the movement distance to calculate the transition probability of the HMM, which will reduce the computational complexity of the positioning process. However, the Weibull distribution cannot accommodate the fluctuating RSSI samples, and the transition probability hardly reflects the real state transition process.

A novel solution named HIPE (Hybrid Indoor Positioning Engine) [11] has applied data fusion to indoor positioning. The HIPE is an engine which fuses the motion dynamics information of a mobile user with the RSSI. Two algorithms are employed in the HIPE, and which one is selected depends on the need for real-time service. The problem with this solution is that the HIPE must install some software on the user's smartphones and this relies on the computational resources of the smartphone. Besides that, the authors do not outline an explicit method to determine the transition probability.

Coleri et al. [12] employed a HMM-based positioning algorithm to position vehicles in constrained routes by exploiting GSM RSSI data. Compared with the previously proposed HMM techniques, their algorithm trains HMM using the driver behavior statistics for the road segments instead of employing the driver behavior as an equal-probability distribution for each of the adjacent reference points (RPs) without any training. They do not take into account that the speed of each movement is different, but rather assume the vehicle is moving with uniform speed on the road segments.

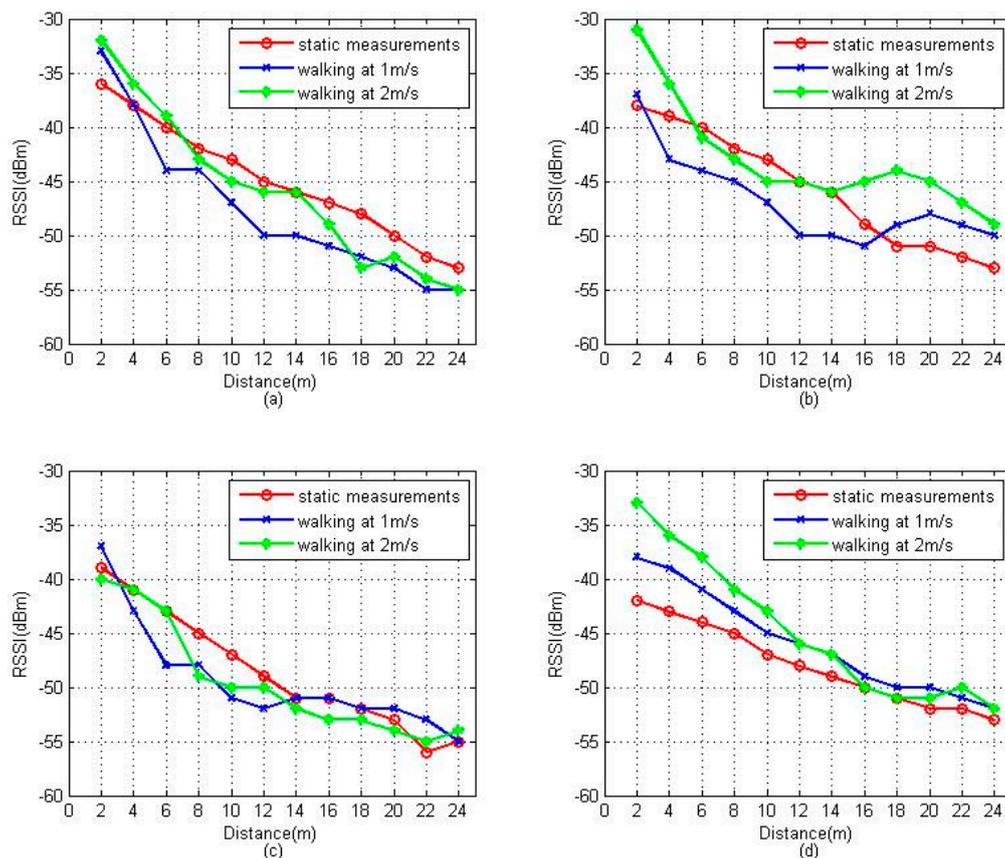
## 3. Fuzzy Pattern Recognition Algorithm

In previous studies, there is a simple scheme of pattern recognition in indoor positioning. It is assumed that each position is associated with a unique set of RSSI values, so the distance between a user's measured location and the reference location can be calculated by Equation (1):

$$d(s, r) = \sqrt{\sum_{i=1}^n (s_i - r_i)^2} \quad (1)$$

where  $s = [s_1 \cdots s_i]$  and  $r = [r_1 \cdots r_i]$  represent the RSSI vector, respectively,  $n$  represents the number of Access Points (APs),  $d(s, r)$  represents the Euclidean distance between two locations. The user coordinates are computed by the Euclidean distance from the measured location to all reference locations stored in the radio map, and the RSSI vector with the smallest distance is the nearest location to the true user location. This method is called the Nearest Neighbor (NN) algorithm, and an enhanced method employing the average  $K$  nearest RP is called KNN. Both are widely known deterministic pattern matching algorithms.

Although this deterministic pattern matching algorithm with better positioning accuracy can be easily implemented in a WLAN fingerprint system, it is based on an accurate radio map and high precision sampling of RSSI, the latter being almost impossible while people are walking due to the vulnerabilities of the RSSI. Figure 2 shows the fluctuation of RSSI sampling by a user carrying a smartphone walking at different speeds within radio range. Here, we assume that the maximum walking speed is no more than 2 m per second (2 m/s), as mentioned by Alexander et al. [13]. To show that the RSSI variation trend is independent of the device used, we repeated the experiment several times with two APs and two smartphones, both from different vendors. For comparison, the static positioning measurements of RSSI, which are usually stored in the radio map, are also given in the figure. Pu et al. [14] have presented similar results, but collected at a much faster speed.



**Figure 2.** RSSI variation trend at different walking speed with different devices: (a) Samsung S5 with TP-LINK TL-WR840N; (b) XIAOMI, (XIAOMI, Beijing, China), HM2 with TP-LINK TL-WR840N; (c) Samsung S5 with MERCURY MW325R; (d) XIAOMI HM2 with MERCURY MW325R.

Compared with the static positioning measurements, we can see that the RSSI values show significant variations while people are walking due to the sophisticated propagation of the radio signal. The RSSI deviations caused by the different measurement conditions (static or dynamic) will introduce critical positioning errors. Meanwhile, we find that most of the dynamic positioning

measurements also follow the propagation and attenuation law, whereby the RSSI fades with the increasing distance between the measuring point and the AP. From this rule, the RSSI variation trend can be used to indicate the relative direction and distance from the initial location to the positioning point. For example, if the RSSI is stronger than at the initial location, it means the positioning point is closer to the AP than the initial location. If there are enough RPs in the area, it can be easily compared to determine the relative locomotion of the pedestrian.

Based on above analysis, we propose a fuzzy pattern recognition algorithm to estimate the pedestrian position, thus reducing the accuracy loss caused by people walking. It uses the RSSI variation trend to replace the actual RSSI value, and the HMM observation state is also indicated by the RSSI variation trend. Relative to the last observation state, RSSI has three main variation trends in HMM, namely, stronger, weaker and flat. To simplify our algorithm, we use the numbers 1, -1, and 0 to respectively represent these three trends. Except for the initial observation state, all the subsequent observation states are indicated by three numbers. For example  $O_{(t,t-1)} = (-1, 0, \dots, 0, 0, 1, 1, -1)$ , presents the variation of RSSI value at time  $t$  compared to time  $(t-1)$  in a localization area that contains  $n$  APs. The fuzzy pattern recognition arithmetic steps are as follows:

- (1) Given an initial RP  $P_0$  in the grid area established via static positioning or user feedback, the RSSI vectors and the position coordinates of  $P_0$  are respectively  $s_{t-1} = (rss_{1(t-1)}, rss_{2(t-1)}, \dots, rss_{n(t-1)})$  and  $l_{t-1} = (x_{p0}, y_{p0})$ .
- (2) Measure RSSI vectors  $s_t = (rss_{1t}, rss_{2t}, \dots, rss_{nt})$  during pedestrian walking per second, and compare each AP's RSSI of  $s_t$  with  $s_{t-1}$ . Considering the signal random disturbance, we give a  $\pm 1$  dB m threshold to compensate for the influence of RSSI fluctuation.  $O_{(t,t-1)}$  is the comparison result. The pseudo-code of this procedure is shown as follows:

there are:

```

st = (rss1t, rss2t, ... rssnt);
st-1 = (rss1(t-1), rss1(t-1), ... rss1(t-1))
O(t,t-1) = (o1, o2, ... on);
for(j = 1; j ≤ n; j++)
{
if (rssjt > (rssj(t-1) + 1))
oj = 1
else if (rssjt < (rssj(t-1) - 1))
oj = -1
Else
oj = 0
}

```

- (3) Compare each AP's RSSI of  $P_0$  with the adjacent RPs in four directions, namely  $P_e, P_w, P_s$  and  $P_n$ . The position coordinates of the four RPs are  $l_{p_e}, l_{p_w}, l_{p_s}, l_{p_n}$ . By employing the above method, the comparison results are  $O_e, O_w, O_s$  and  $O_n$ .
- (4) For estimating the position of the pedestrian, we need to separately compare the RSSI variation trend by performing a bitwise logical XOR operation between  $O_{(t,t-1)}$  and  $O_e, O_w, O_s, O_n$ . The results can determine the relative distance from the pedestrian to the four RPs, which can be used to deduce the approximate position of the pedestrian. The pseudo-code of this procedure is shown as follows:

there are:

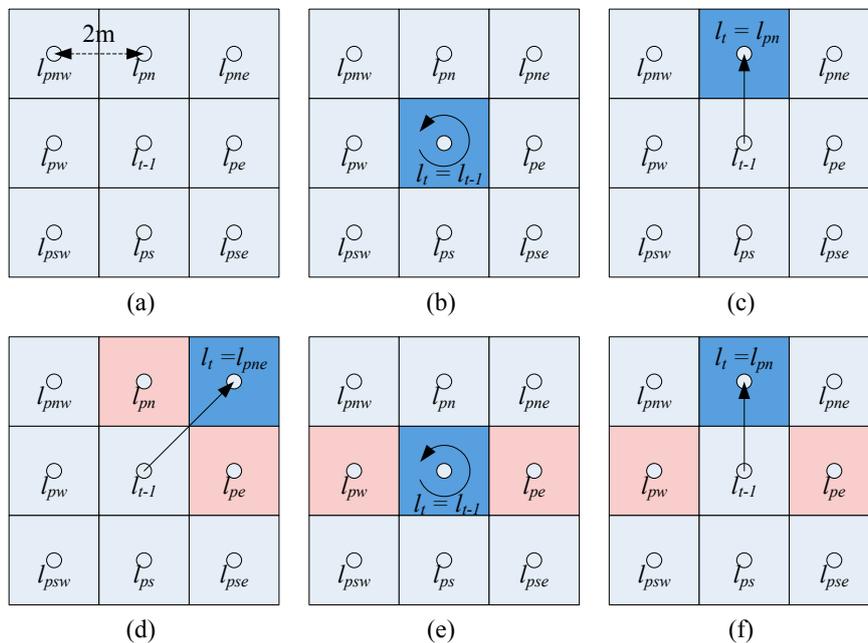
```

lt-1, lp_e, lp_w, lp_s, lp_n, lp_ne, lp_se, lp_sw, lp_nw, lt
O(t,t-1), Oe, Ow, Os, On;
Oe ⊕ O(t,t-1) = (x1e, x2e, ... xne);
Ow ⊕ O(t,t-1) = (x1w, x2w, ... xnw);
Os ⊕ O(t,t-1) = (x1s, x2s, ... xns);
On ⊕ O(t,t-1) = (x1n, x2n, ... xnn);

```

$|O_{(t,t-1)}| = |(o_1, o_2, \dots, o_i)| = (o'_1, o'_2, \dots, o'_i)$ ;  
 $O'_e = (x_{1e} + x_{2e} + \dots + x_{ne})$ ;  
 $O'_w = (x_{1w} + x_{2w} + \dots + x_{nw})$ ;  
 $O'_s = (x_{1s} + x_{2s} + \dots + x_{ns})$ ;  
 $O'_n = (x_{1n} + x_{2n} + \dots + x_{nn})$ ;  
 $O'_{(t,t-1)} = (o'_1 + o'_2 + \dots + o'_n)$ ;  
 if  $(O'_{(t,t-1)} = 0$  or  $O'_{(t,t-1)}$  equal or approximately equal to all of  $(O'_e, O'_w, O'_s, O'_n)$   
 $l_t = l_{t-1}$ ;  
 else if  $(O'_{(t,t-1)}$  only equal or approximately equal to one of  $(O'_e, O'_w, O'_s, O'_n)$   
 $l_t = \text{one of } (l_{pe}, l_{pw}, l_{ps}, l_{pn})$ ;  
 else if  $(O'_{(t,t-1)}$  equal or approximately equal to two of  $(O'_e, O'_w, O'_s, O'_n)$   
 {  
 if (two of  $(O'_e, O'_w, O'_s, O'_n)$  are adjacent )  
 $l_t = \text{one of } (l_{pne}, l_{pes}, l_{psw}, l_{pwn})$ ;  
 else  
 $l_t = l_{t-1}$ ;  
 }  
 else if  $(O'_{(t,t-1)}$  equal or approximately equal to three of  $(O'_e, O'_w, O'_s, O'_n)$   
 $l_t = \text{one of } (l_{pe}, l_{pw}, l_{ps}, l_{pn})$ ;

The “equal” means all elements in the two vectors are totally equal, and the “approximately equal” means the two vectors have the largest number of identical elements in order. The  $l_{pne}, l_{pes}, l_{psw}, l_{pwn}$  respectively indicate the northeast, southeast, southwest and northwest RP coordinates ( $P_{ne}, P_{es}, P_{sw}, P_{wn}$ ). A more visual description of this algorithm is shown as Figure 3.



**Figure 3.** Schematic of the fuzzy pattern recognition algorithm: (a) nine RPs in a grid area and the interval of the RPs is 2 m; (b)  $l_t = l_{t-1}$  when the RSSI has no variation or a similar variation in four directions; (c)  $l_t = l_{pn}$  when the RSSI variation trend follows the trajectory from  $P_0$  to  $P_n$ ; (d)  $l_t = l_{pne}$  when the RSSI has a similar variation trend in the north and east, we assume the RSSI variations follow the trajectory from  $P_0$  to  $P_{ne}$ ; (e)  $l_t = l_{t-1}$  when the RSSI has a similar variation in the east and west; (f)  $l_t = l_{pn}$  when the RSSI has a similar variation in the east, west and north, we assume the RSSI variations follow the trajectory from  $P_0$  to  $P_n$ .

Figure 3 contains all the possible cases of pedestrian movement if we consider the maximum walking speed is no more than 2 m/s, as mentioned above. To estimate the pedestrian position with this algorithm iteratively, we assume that the positioning result is RP  $P_{ne}$  in the case of Figure 3d, although it is impossible for people to walk from  $P_0$  to  $P_{ne}$  at 2 m/s walking speed. From Figure 3 it is easy to see that the positioning accuracy of the fuzzy pattern recognition algorithm is limited by the interval between RPs. The comparison of positioning results under different measuring conditions (static or dynamic) will be discussed in the later experiments.

#### 4. Pedestrian Positioning Method Based on HMM

As discussed in Section 2, some researchers employ HMM to implement moving target positioning, but all such approaches more or less have some defects and problems. To make up for the defects and solve the problem, we propose a novel scheme that combines the WLAN fingerprints system with the HMM. To decrease the accuracy loss caused by the vulnerability of WLAN signals, it uses a fuzzy pattern recognition method instead of the deterministic pattern matching algorithm in HMM. It uses the FPR algorithms to train HMM with user trajectories and estimate hidden location states through the Viterbi algorithm.

##### 4.1. Mathematical Model of Pedestrian Positioning

The Hidden Markov Model (HMM) is a well-known statistical model which has been successfully applied to acoustic treatment, biological information and natural language processing. Until now, it is still widely used in speech recognition [15]. Different from the traditional Markov model, HMM is a simple dynamic Bayesian network, where there are two kinds of states: hidden state and observation state. The hidden state transition process is a Markov process, and the observation states are generated by hidden Markov processes according to a certain probability distribution [16]. Figure 4 shows the basic architecture of a first-order HMM, where the term  $l_t$  is the hidden state at time  $t$ ,  $o_t$  is the observation at time  $t$ ,  $P(l_t|l_{t-1})$  is the transition probability from  $l_{t-1}$  to  $l_t$ ,  $P(o_t|l_t)$  is the conditional probability when the observation is  $o_t$  in the hidden state  $l_t$ .

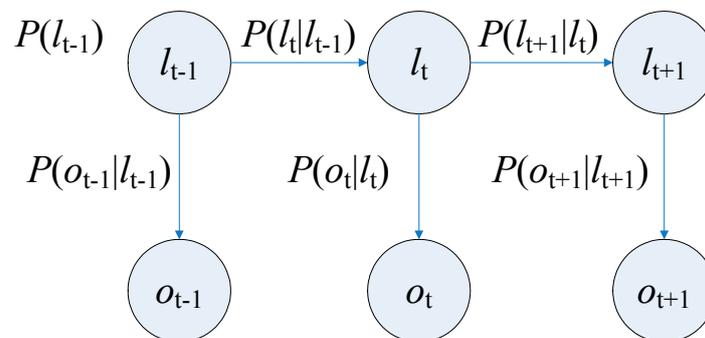


Figure 4. The architecture of the first-order HMM.

Employing HMM can effectively improve pedestrian positioning accuracy in a WLAN fingerprint system. The HMM can be mathematically defined as a triple  $\lambda = \{\pi, A, B\}$  which includes three probability matrices, and two sets of states (L, O) in our pedestrian positioning method.

L is the location space,  $L = \{l_1, l_2, \dots, l_N\}$ ,  $l_i$ , ( $1 \leq i \leq N$ ), each state represents a reference grid point with coordinates in the localization area [17], N is the total number of points.

V is the observation signal space,  $V = \{v_1, v_2, \dots, v_M\}$ ,  $v_j$ , ( $1 \leq j \leq M$ ), each state represents a RSSI variation trend vector in a location state, M is the total number of possible vectors. The vector has three kinds of symbol, namely 1, 0, -1, which are respectively presented as  $z_{-1}$ ,  $z_0$ ,  $z_1$ .

$\pi$  is an initial location state probability distribution,  $\pi = P(l_i)$  ( $l_i \in L$ ) is the probability of state  $l_i$ . Here, we assume that  $\pi$  follows a uniform probability distribution.

A is an  $N \times N$  state transition matrix,  $A = \{a_{ij} = P(L_{t+1} = l_j | L_t = l_i)\}$  ( $1 \leq i, j \leq N$ ),  $a_{ij}$  is the probability of the transition from location  $l_i$  to location  $l_j$ , thus there is  $\sum_{j=1}^N a_{ij} = 1$  ( $0 \leq a_{ij} \leq 1$ ). Because of the limitation of pedestrians' walking speed (i.e., 2 m/s), if we divide the location area into  $2 \text{ m} \times 2 \text{ m}$  grids, we can consider that the transition probability is only valid between the adjacent grids, and the transition probability is zero in the case of cross grids. The transition probability of HMM is time invariant, so  $P(L_t = l_t)$  solely depends on the last location state  $l_{t-1}$ .

B is a confusion matrix,  $B = \{b_i(k) = P(v_k | l_i)\}$  ( $v_k \in O, l_i \in L$ ),  $b_i(k)$  is the emission probability when the RSSI variation trend is  $v_k$  in location  $l_i$  at time  $t$ , thus there is  $\sum_{k=1}^N b_i(k) = 1$  ( $0 \leq b_i(k) \leq 1$ ). Here, we assume that the emission probability solely depends on the location state at time  $t$ .

#### 4.2. Pedestrian Positioning Using Viterbi Algorithm

Given a set of RSSI variation trends,  $V = \{v_1, v_2, \dots, v_M\}$  and a settled HMM  $\lambda = \{\pi, A, B\}$ , the hidden location sequence  $L = \{l_1, l_2, \dots, l_N\}$  can be estimated by employing the Viterbi algorithm. The Viterbi algorithm is proposed on the basis of dynamic programming for finding the optimal hidden state sequence [18]. The observed RSSI variation trends and the possible location transition are shown as a trellis in Figure 5.

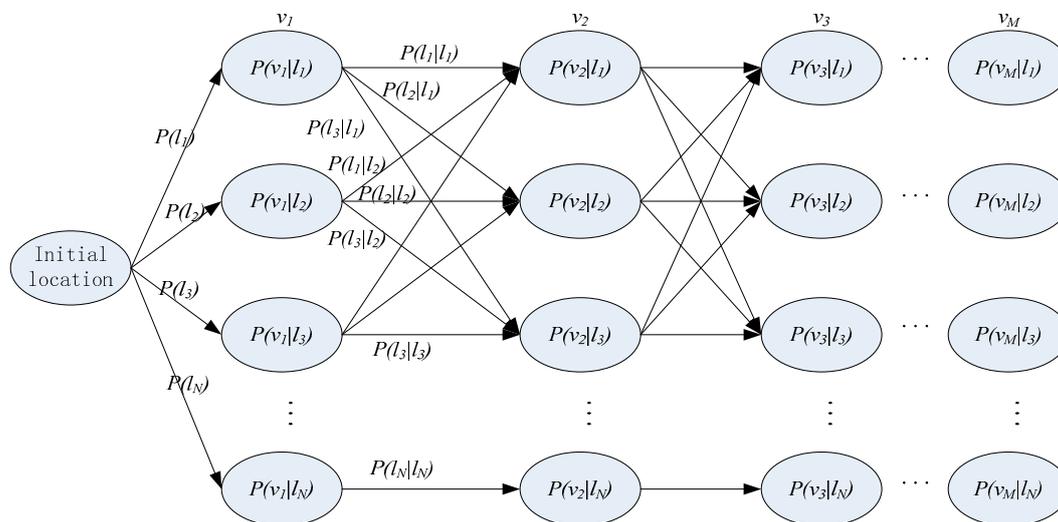


Figure 5. The trellis of the RSSI variation trends and the location transition.

To find the most likely sequence of the location, i.e., marking the path of the actual location transition in Figure 5, the Viterbi algorithm will be manipulated through four procedures as follows: Initialization, Recursion, Termination, and Path Backtracking [10]. Since there are a lot of resources describing the algorithm, it will not be covered here.

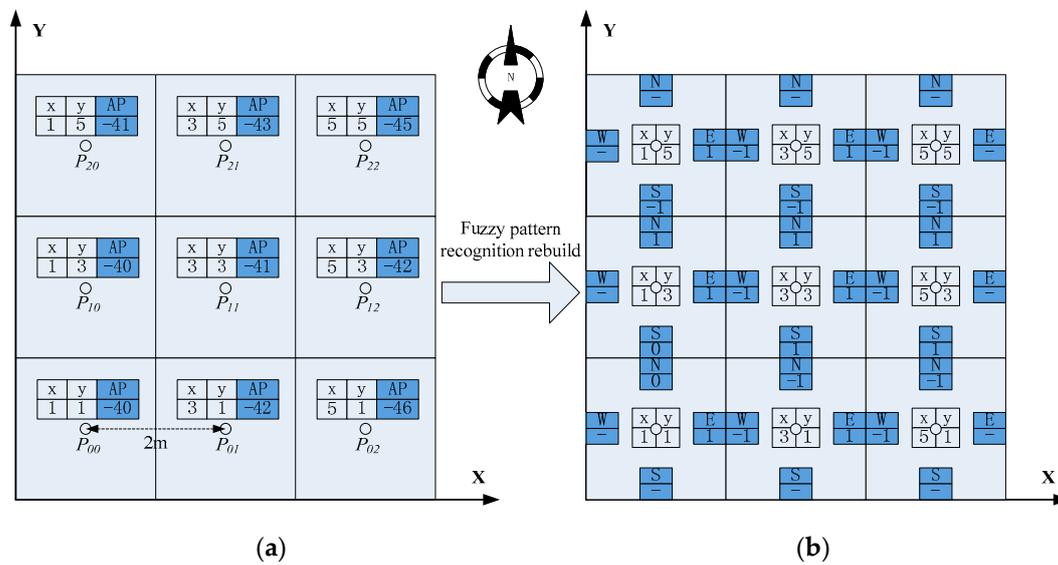
Through the procedures, we can learn that the transition probability has an important role in the Viterbi algorithm. Not only is the computation complexity of the Viterbi relative to the transition probability, but also the positioning accuracy is dependent on the transition probability. So far, the problem has been centered on how to calculate the transition probability.

#### 4.3. Employing the FPR Algorithm to Estimate the Transition Probability

Transition probability is the most critical parameter in the pedestrian positioning model. The positioning accuracy of the Viterbi algorithm depends on the precision of the transition probability. To obtain the transition probability which can reflect the pedestrian walking pattern, we utilize the FPR algorithm to estimate the transition probability using a rebuilt radio map and planned path.

The pre-existing radio map stored the RSSI and the corresponding position coordinates. To reduce the RSSI deviation caused by different measurement conditions, we use the fuzzy pattern recognition

algorithm to rebuild the pre-existing radio map. A comparison of the two kinds of radio maps is shown in Figure 6.



**Figure 6.** The comparison of two kinds of radio maps: (a) the pre-existing radio map; (b) the reconstructive radio map.

The left part of Figure 6 is a traditional radio map, where to keep things simple, we only give RSSI values of one AP in the radio map. The right part is the new radio map which is rebuilt by comparing the RSSI variation trend between RP and its adjacent RPs in four directions. Correspondingly, the HMM observation signal space also changes from the RSSI value to the RSSI variation trend.

In indoor areas, people will follow certain rules in moving, such as a limited walking speed, that there is no way through the walls, and that they walk in a straight line in a corridor. Using these rules, we can reasonably plan the user’s path, and use these trajectories as the HMM training samples so as to effectively improve the positioning accuracy. For training the HMM with the FPR algorithm, a set of consecutive RSSIs are recorded by volunteer sampling along the planned path. The planned path should contain all the RPs [17], so it will be able to obtain higher positioning accuracy. Because the FPR algorithm is sensitive to the direction of pedestrian travel, multiple training trajectories are provided in all four directions, as well.

As mentioned before, the pedestrian locomotion is limited to a certain walking speed (2 m/s in this study) in an indoor environment. Figure 7a demonstrates that, after a sampling period (one second in this study), the next location can only be within a circle with a radius of two meters. If we collect the training sample at approximately the maximum speed and we set the RPs’ interval as equal to the maximum speed, the transition probability can be estimated through the FPR algorithm.

The main goal of the FPR algorithm is to consider the adjacent RPs as the possible pedestrian locations in the next state, as shown in Figure 7b.  $V = \{v_1, v_2, \dots, v_M\}$  is the set of RSSI values along the planned path, and  $M$  is the number of the RPs along the path. It is obvious that  $v_i$  is a RSSI sampling value at the position of  $RP_i$ . The transition probability from  $RP_i$  to the adjacent RPs can be estimated as follows:

- (1) Compare the  $v_i$  and  $v_{i+1}$  using the FPR arithmetic step two. The  $O_{(i,i+1)}$  is the comparison result.
- (2) The  $O_e, O_w, O_s, O_n$  are comparison results from  $RP_i$  to adjacent RPs in the four directions. Because the radio map has been rebuilt, the  $O_e, O_w, O_s, O_n$  are already known. We can directly perform comparison operations between  $O_{(i,i+1)}$  and  $O_e, O_w, O_s, O_n$  just as the FPR arithmetic steps 3 and 4.

- (3) According to the comparison results between  $O_{(i,i+1)}$  and  $O_e, O_w, O_s, O_n$ , we can estimate the transition probability using a proportion method. To simplify the calculation we consider that the transition probability is in direct proportion to the logical XNOR operation result between  $O_{(i,i+1)}$  and  $O_e, O_w, O_s, O_n$ . No matter whether the RSSI is stronger or weaker, the more identical they are, the higher the transition probability is. The pseudo-code of this procedure is shown as follows:

there are:

$O_{(i,i+1)}, O_e, O_w, O_s, O_n, P_c, P_e, P_w, P_s, P_n, N_c, N_e, N_w, N_s, N_n, N_{sum}$ ;

$RP_e, RP_w, RP_s, RP_n$ , are adjacent RPs to  $RP_i$  on four direction;

$O_e \odot O_{(i,i+1)} = (y_{1e}, y_{2e}, \dots, y_{ne})$ ;

$O_w \odot O_{(i,i+1)} = (y_{1w}, y_{2w}, \dots, y_{nw})$ ;

$O_s \odot O_{(i,i+1)} = (y_{1s}, y_{2s}, \dots, y_{ns})$ ;

$O_n \odot O_{(i,i+1)} = (x_{1n}, x_{2n}, \dots, x_{nn})$ ;

$N_c$  = The number of zeros of  $O_{(i,i+1)}$ ;

$N_e = (y_{1e} + y_{2e} + \dots + y_{ne})$ ;

$N_w = (y_{1w} + y_{2w} + \dots + y_{nw})$ ;

$N_s = (y_{1s} + y_{2s} + \dots + y_{ns})$ ;

$N_n = (x_{1n} + x_{2n} + \dots + x_{nn})$ ;

$N_{sum} = N_c + N_e + N_w + N_s + N_n$ ;

$P_c = P(L_{t+1} = RP_i | L_t = RP_i) = \frac{N_c}{N_{sum}}$

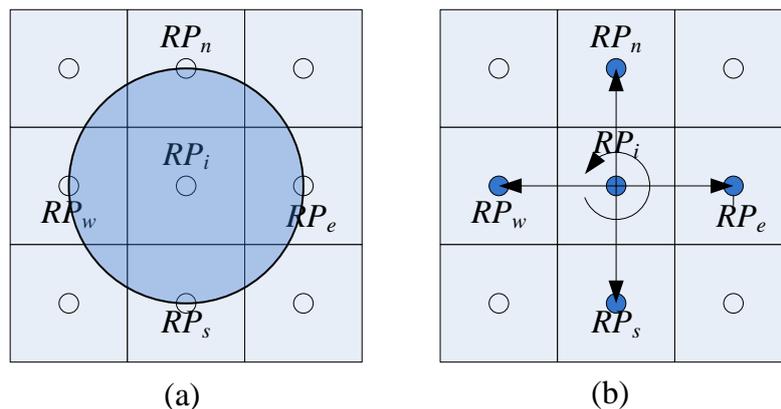
$P_e = P(L_{t+1} = RP_e | L_t = RP_i) = \frac{N_e}{N_{sum}}$

$P_w = P(L_{t+1} = RP_w | L_t = RP_i) = \frac{N_w}{N_{sum}}$

$P_s = P(L_{t+1} = RP_s | L_t = RP_i) = \frac{N_s}{N_{sum}}$

$P_n = P(L_{t+1} = RP_n | L_t = RP_i) = \frac{N_n}{N_{sum}}$

- (4) If there are multiple training trajectories from other directions, we use the above steps to estimate the transition probability from other directions, and the final results are the arithmetic means of the transition probability from different directions. In this way, we use the FPR algorithm to estimate the transition probability, which provides strong support for the pedestrian positioning with the Viterbi algorithm.



**Figure 7.** The pedestrian location transition to the next state: (a) the possible location after a sampling period; (b) consider the adjacent RPs as the possible pedestrian locations.

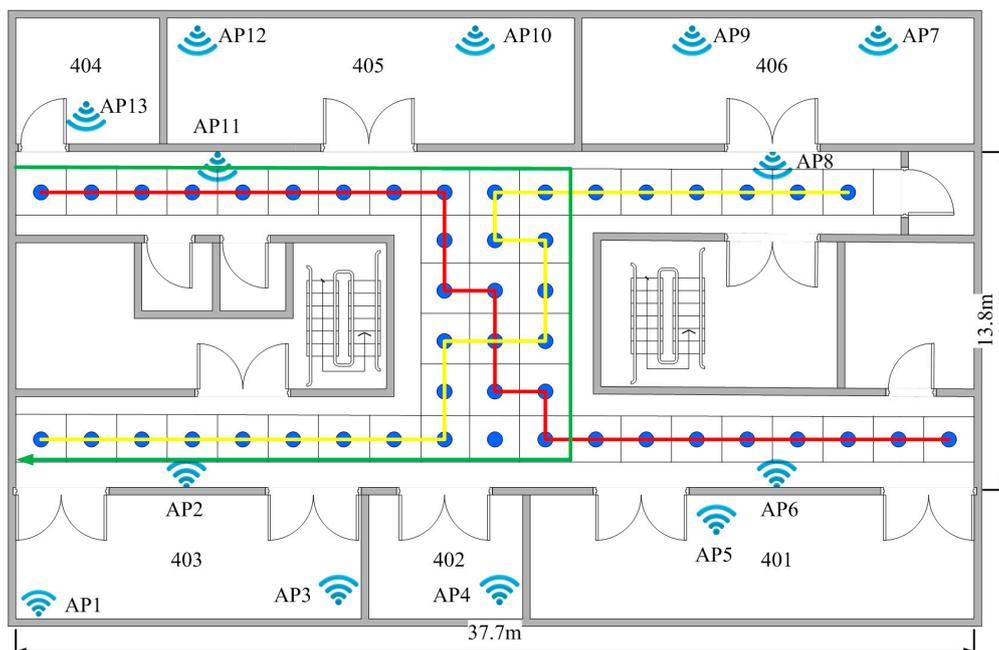
## 5. Positioning Experiments and Results

To validate our algorithm and method, we perform some experiments in two real indoor environments. The first scenario is an office area located on the fourth floor of the YiFu building, and the second scenario is an open area located in the library.

To meet the needs of our experiment, the RSSI collection was performed using a Samsung S5 smartphone, in which the API supports scanning the RSSI with an interval of one second. If not otherwise specified, all measurements are collected with this smartphone. During the offline phase, we have taken a series of measures to improve the RSSI measurement accuracy. To mitigate the impact of the distribution difference of RSSI in different directions, we collected the measurements in four directions. Then, we collected 40 measurements at each reference point to decrease the random noise effects (thus 10 samples for each direction). Finally, we conducted the collection at midnight to avoid the interference caused by walking people. In the experiments, we mainly use the cumulative distribution function of error (CDF) and root mean square error (RSME) to calculate the positioning error.

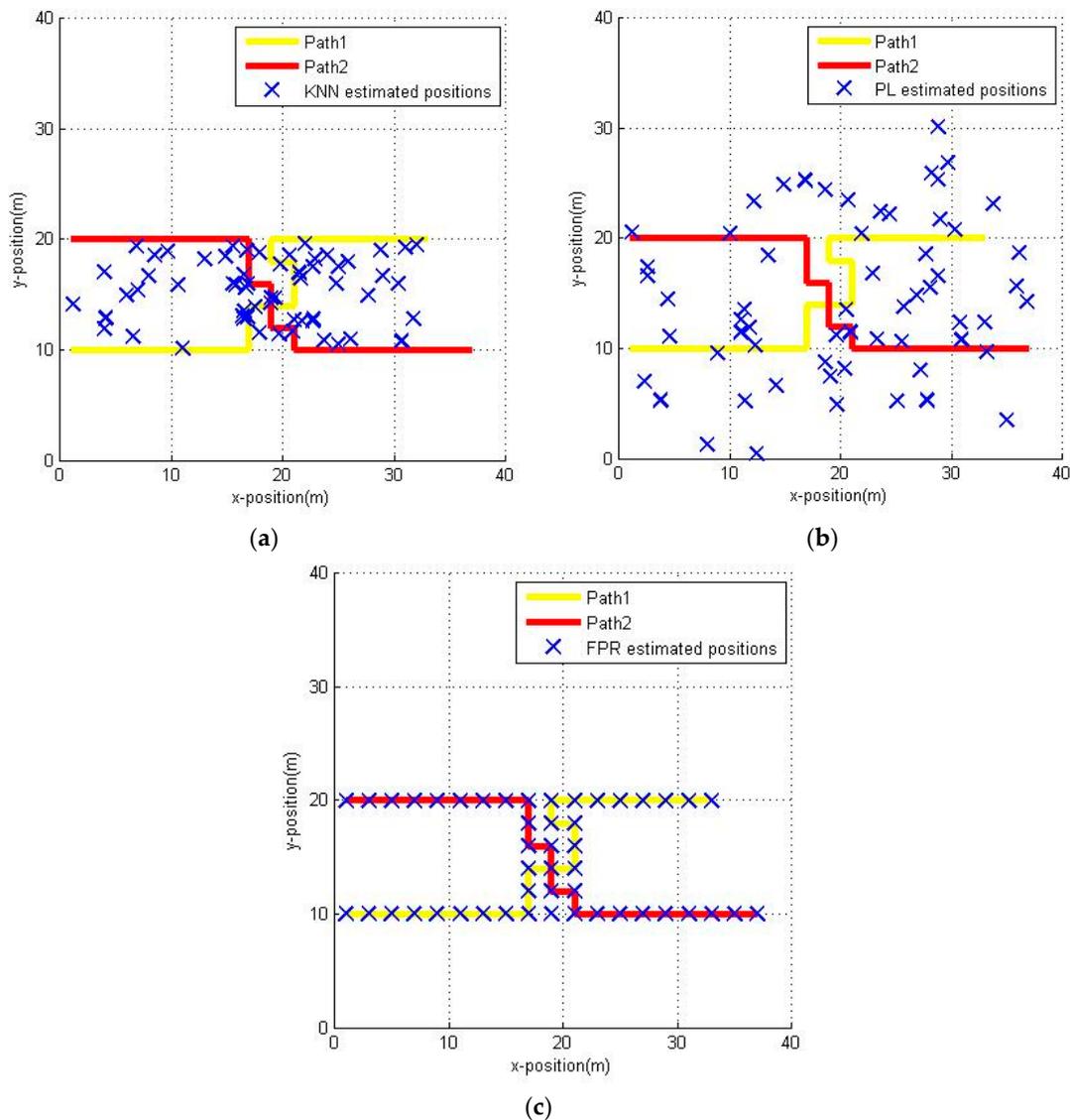
### 5.1. Performance Evaluation in an Office Building

This office area includes a horizontally H-shaped corridor and some rooms, as shown in Figure 8. For the purposes of simplifying the experimental procedure, we only chose the corridor area, which has a dimension of  $37.7\text{ m} \times 13.8\text{ m}$  for localization. There are a total of 13 APs on this floor, four of them are RG-AP220-E APs which are deployed in the ceiling; the rest of the nine household APs are produced by different manufacturers, which are deployed in the different offices. The rough position of the APs is shown in Figure 8 as a wireless symbol. Due to the high-density deployment of the APs on the floor, almost all the localization area can receive 13 signals from the different APs. If there is an area where the signal cannot be received at that moment, it will be recorded as  $-99\text{ dBm}$ . The whole corridor area has been divided into 48 uniform  $2\text{ m} \times 2\text{ m}$  grids, and the reference point is the center of each grid.



**Figure 8.** The layout of the fourth floor of the YiFu building.

After the radio map of this office area has been built, we can perform the experiments. The measurements are collected by a user walking along the planned paths at about  $1.5\text{ m/s}$ . The planned paths are shown in Figure 8 as a yellow line and red line, respectively. Both of them are  $46\text{ m}$  long; that means we can collect 60 measurements while the user is walking at a speed of about  $1.5\text{ m/s}$ . We estimated the user position by three methods, namely the KNN algorithm ( $k = 8$ ), the probabilistic location (PL) [19,20] algorithm and the fuzzy pattern recognition algorithm, respectively. The estimated positions of the user are shown in Figure 9.

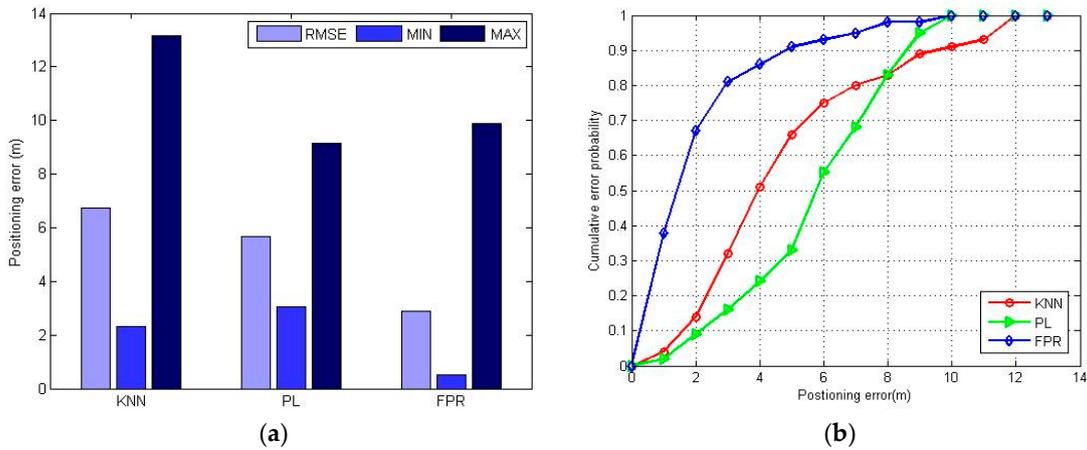


**Figure 9.** Estimated positions by different positioning algorithms: (a) KNN algorithm; (b) PL algorithm; (c) FPR algorithm.

From Figure 9, we can clearly see that because of the RSSI deviations caused by the different measuring conditions (static or dynamic), the estimated positions present a cloud-like pattern in Figure 9a,b. That means there are lots of ambiguities in the results when we use the KNN algorithm and PL algorithm. Unlike the two left figures, Figure 9c shows that the estimated positions are punctuated clearly and accurately at the reference points along the planned path. That is because the path just passes through the RPs, and the FPR algorithm selects the adjacent RPs as the candidate pedestrian position. Actually, the performance of the FPR algorithm is better than that of the other algorithms but it is not as strong as Figure 9 shows.

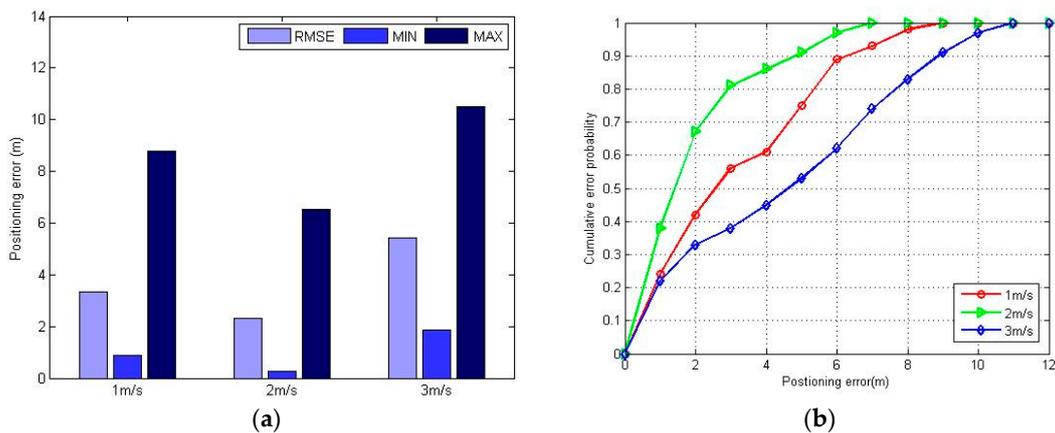
To fairly compare the positioning error, Figure 10 shows the experimental results of RMSE, minimum error, maximum error and cumulative error probability of the different positioning algorithms. Obviously, the FPR algorithm performs better than the KNN and PL algorithm. The RMSE and the minimum error of the FPR algorithm are both minimal, but the maximum error of the FPR algorithm is greater than that of the PL algorithm. This is because the FPR algorithm introduces an accumulating error with each positioning process when direction-hopping happens.

The direction-hopping means the ambiguities of the positioning result are too numerous to distinguish the direction of the next position.



**Figure 10.** The positioning error of KNN, PL and FPR algorithm: (a) maximum positioning error, minimum positioning error and RMSE of different positioning methods; (b) CDF of different positioning methods.

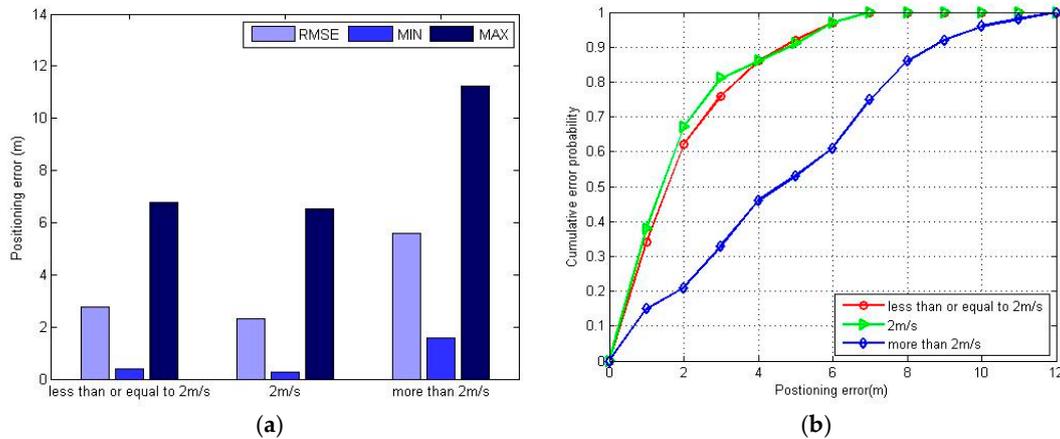
Figure 11 shows the experimental results of the positioning error using the FPR algorithm when the pedestrian walks at different speeds. In the case where the pedestrian speed is no more than 2 m/s, the dynamic positioning error only fluctuates slightly at different speeds, and the positioning error is smaller when the speed is approaching 2 m/s. When the pedestrian speed is 3 m/s, the positioning error becomes larger. The main factor is that when the distance a pedestrian walks in a second is more than the RPs’ interval—and the distance a pedestrian moves may span one or two grid areas—the FPR algorithm has difficulties to distinguish the real position because it only considers the RSSI variation trend, and this ambiguity will also accumulate in each positioning process and finally bring about a large positioning error.



**Figure 11.** The positioning error at different walking speeds using FPR algorithm: (a) maximum positioning error, minimum positioning error and RMSE at different speed; (b) CDF at different speed.

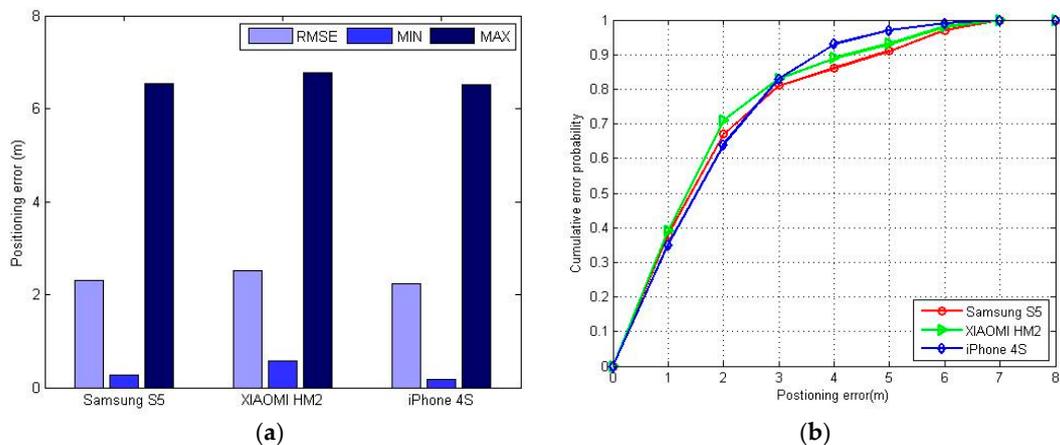
Figure 12 shows the positioning error of the proposed FPR algorithm at uniform and variable speeds. When the variable speed range is less than or equal to 2 m/s, the positioning error is slightly worse than at the uniform speed of 2 m/s. Otherwise, the positioning accuracy will decrease rapidly. From Figures 11 and 12, we know that FPR algorithm offers better accuracy when the pedestrian walking speed is no more than the RPs’ interval per second (here is 2 m/s), either at uniform or

variable walking speed. The initial position of each estimation is replaced iteratively by adjacent RPs in the FPR algorithm, which can easily compensate for errors of less than the RP interval, but it cannot decrease the error accumulation caused by walking too fast.



**Figure 12.** The positioning error under uniform and variable speed using FPR algorithm: (a) maximum positioning error, minimum positioning error and RMSE at uniform and variable speeds; (b) CDF at uniform and variable speeds.

To test the adaptability of the FPR algorithm to different devices, we used a XIAOMI HM2 and an iPhone 4S (IOS 6.1.3, Apple, Cupertino, CA, USA) to repeat the experiments. There was little change in the positioning error, which means the FPR algorithm has good adaptability to different smartphones, as can be seen in Figure 13.



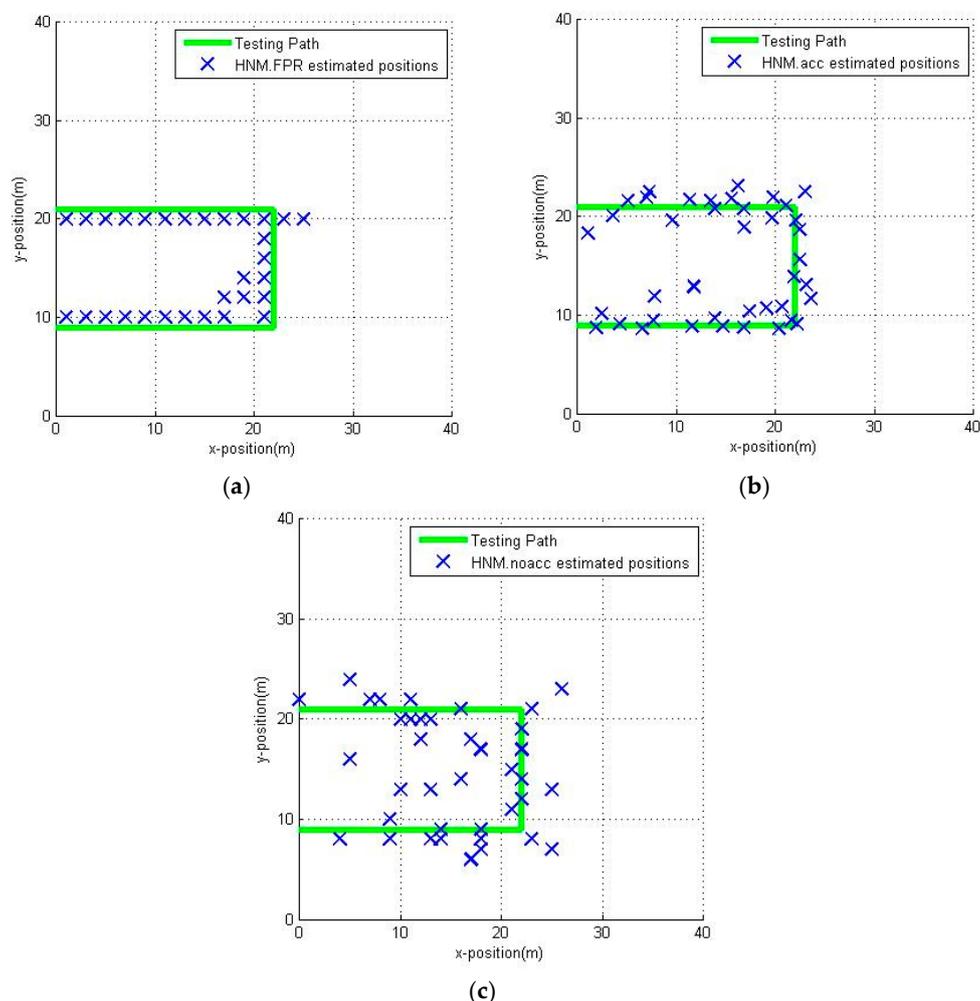
**Figure 13.** The positioning error with different smartphones using the FPR algorithm: (a) maximum positioning error, minimum positioning error and RMSE with different devices; (b) CDF with different devices.

From the above experimental results, we can see that the FPR algorithm can offer better positioning accuracy over a certain range of speeds and has good adaptability to different devices, but it also has two drawbacks: one is that its positioning accuracy is limited by the RP interval, the other is the accumulating positioning error when direction-hopping is happening.

In the HMM method testing, the red line and the yellow line are treated as the model training trajectories, and the walking path is the green line as shown in Figure 8. We use the FPR algorithm to estimate the transition probability with the measurements collected at the speed of 2 m/s from two directions, one time from east to west, and the other time from west to east. To collect the

measurements for HMM method testing, a user walks at a variable speed which is no more than 2 m/s with the Samsung S5 (Samsung, Tianjin, China) following the green line from the door of the room marked as 404 to the west door of room 403. For comparison, we also use the method mentioned by Liu et al. [10] to estimate the pedestrian position. The method HMM.acc combines the HMM with the accelerometer sensor to compute the pedestrian position. The HMM.noacc is an alternative solution if the accelerometer sensor is unavailable, which needs to assume the pedestrian walks at a uniform speed.

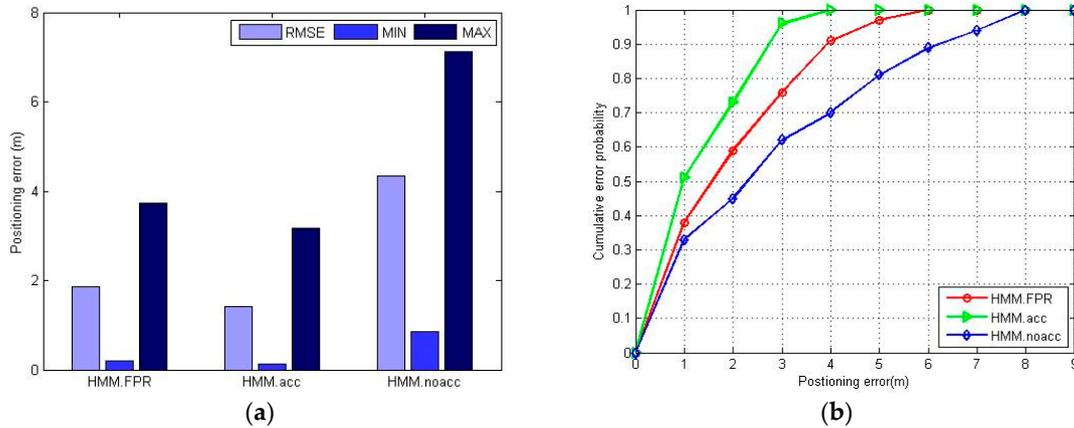
The estimation of the user position along the green path is shown in Figure 14. Because of the characteristics of the FPR algorithm, all the estimated positions are on the RPs, thus reducing the positioning accuracy to a certain extent but within the range of acceptability. Also, from Figure 14 we can learn that, because of the incomplete coverage of the planned path, the estimated position has been hopped in the lower-middle area of the corridor. The HMM.acc has the best positioning accuracy, as the estimated positions with HMM.acc are closer to the walked path in Figure 14b. It seems like the HMM.noacc has the worst positioning accuracy since some estimated positions cluster together in Figure 14c. The results are further compared in Figure 15.



**Figure 14.** Estimated positions by different positioning methods: (a) HMM.FPR; (b) HMM.acc; (c) HMM.noacc.

It can be seen in Figure 15 that the HMM.FPR (Fuzzy Pattern Recognition) indeed has better positioning accuracy than HMM.noacc, but slightly worse than HMM.acc. The HMM.acc utilizes an accelerometer sensor to provide the pedestrian dynamics information, which is useful to improve

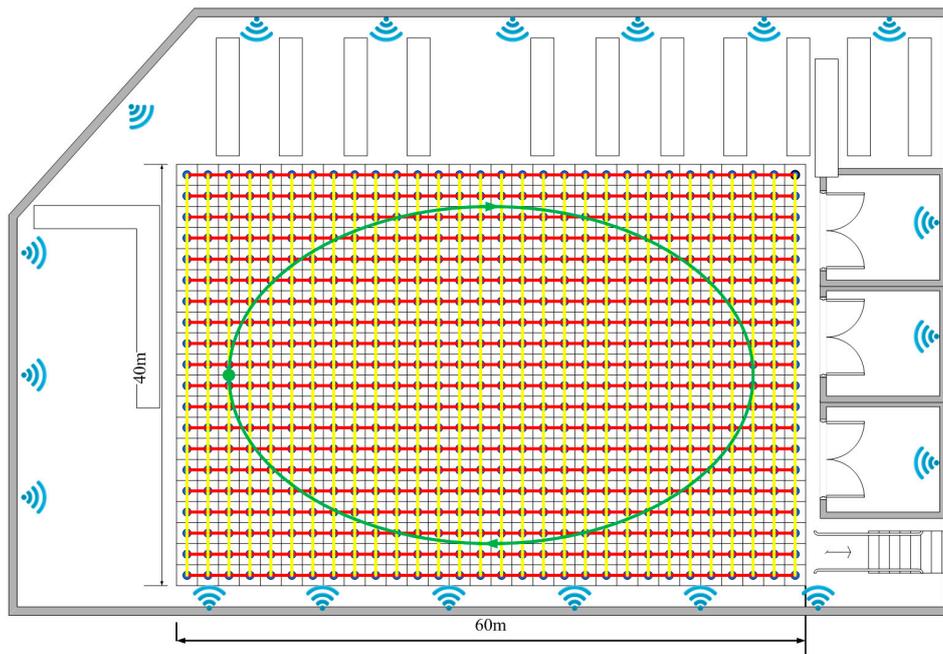
the positioning accuracy. The HMM.noacc offers the worst positioning accuracy because the pedestrian walks at variable speeds, and the assumption of uniform speed will compromise the positioning accuracy. HMM.FPR uses the estimated transition probability to mitigate the effect of direction-hopping, which will effectively reduce the maximum positioning error.



**Figure 15.** The positioning errors of HMM.FPR, HMM.acc, HMM.noacc: (a) maximum positioning error, minimum positioning error and RMSE of different positioning methods; (b) CDF of different positioning methods.

### 5.2. Performance Evaluation in a Library Building

After testing the method in the office area, we chose a relatively open indoor area to further evaluate our method. The open area is on the first floor of our university library (Beijing, China), which has dimensions of 60 m × 40 m. There are a total of 19 APs covering this area, all of which are RG-AP220-E APs deployed in the ceiling. Because there are not so many barriers, some APs can reach this area even at a great distance. The rough position of the APs is given in Figure 16. The whole area has been divided into 600 uniform 2 m × 2 m grids, and the RP interval is 2 m.



**Figure 16.** The layout of the first floor of the university library.

As in Figure 8, we use three colors to present the planned path and the testing path. The red line and the yellow line are the planned path for training the transition probability. In this testing, we collect training samples from four directions to maximize precision. The green line is the only testing path that means all methods use the same measurements to estimate pedestrian position. The starting point is marked as a green dot and arrows are used to mark the walking direction. The 100 test measurements are collected by a user walking along the green path at a variable speed of less than 2 m/s when the library is open.

The positions in the open area estimated by different methods are shown in Figure 17. The KNN and the PL algorithm have the worst positioning accuracy because of the numerous ambiguous results shown in Figure 17a,b. The FPR algorithm is much better than the KNN and PL algorithm but also has some estimated positions with a large distance to the test path. The errors caused by direction-hopping are greater in an open area. The HMM.FPR performs better than HMM.noacc but worse than HMM.acc, and there are few large distance errors in Figure 17d, indicating that the HMM plays an important role in mitigating the effect of direction-hopping. The HMM.acc also has the best positioning accuracy, and the open area seems to have no influence on HMM.acc and HMM.noacc. The RMSE and the CDF of the different methods are shown in Figure 18.

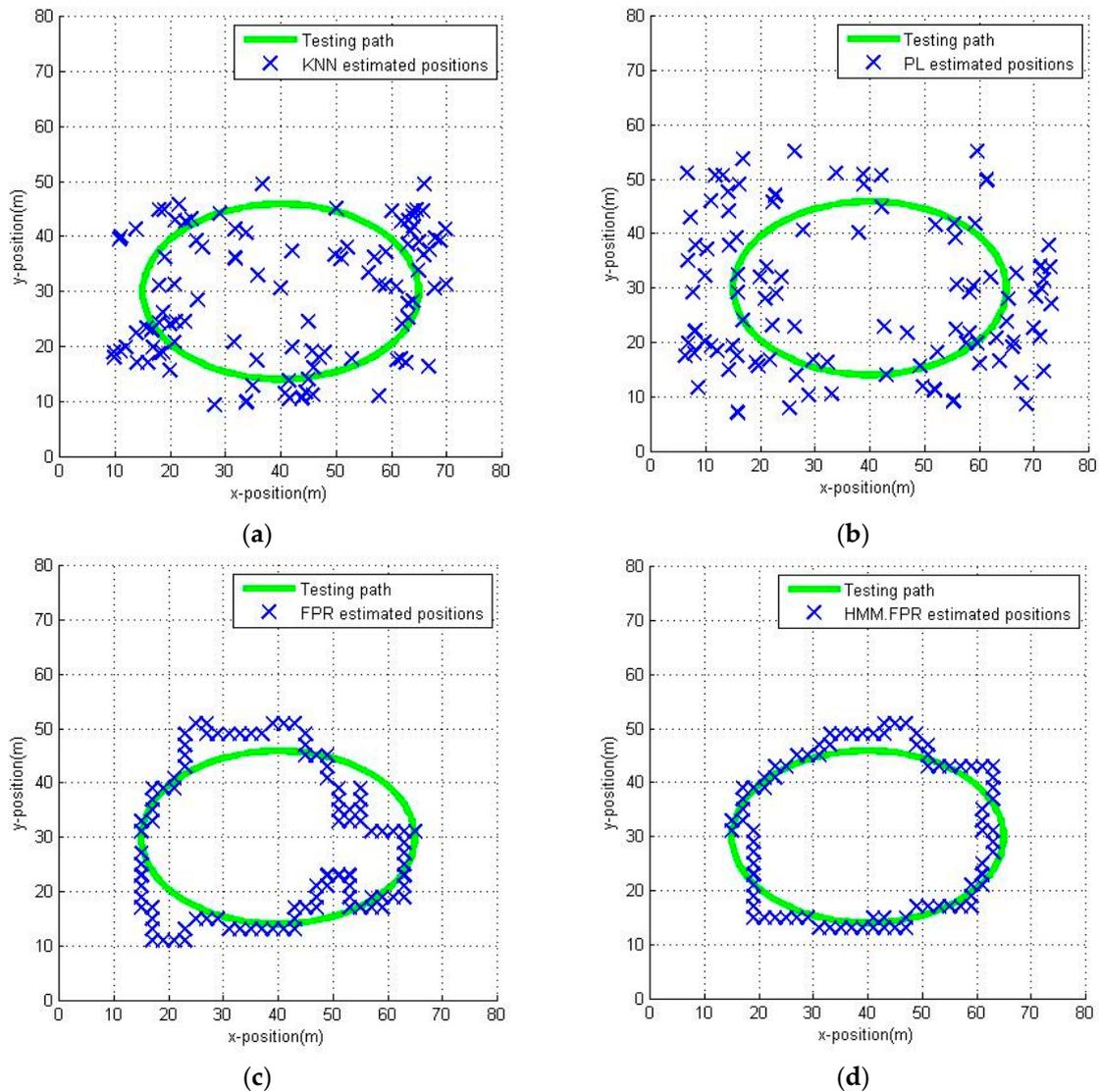
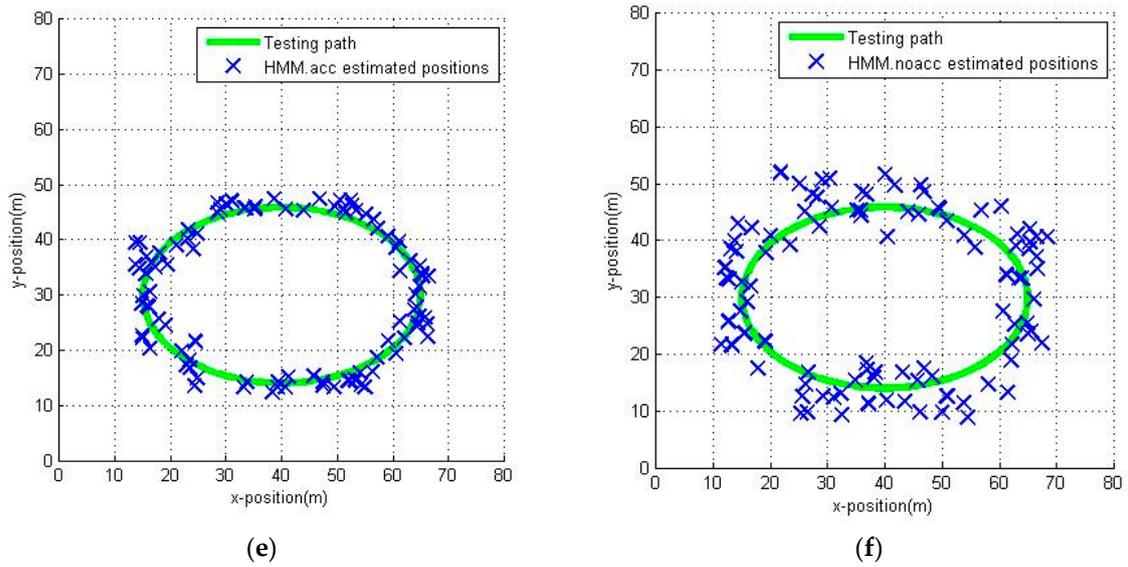
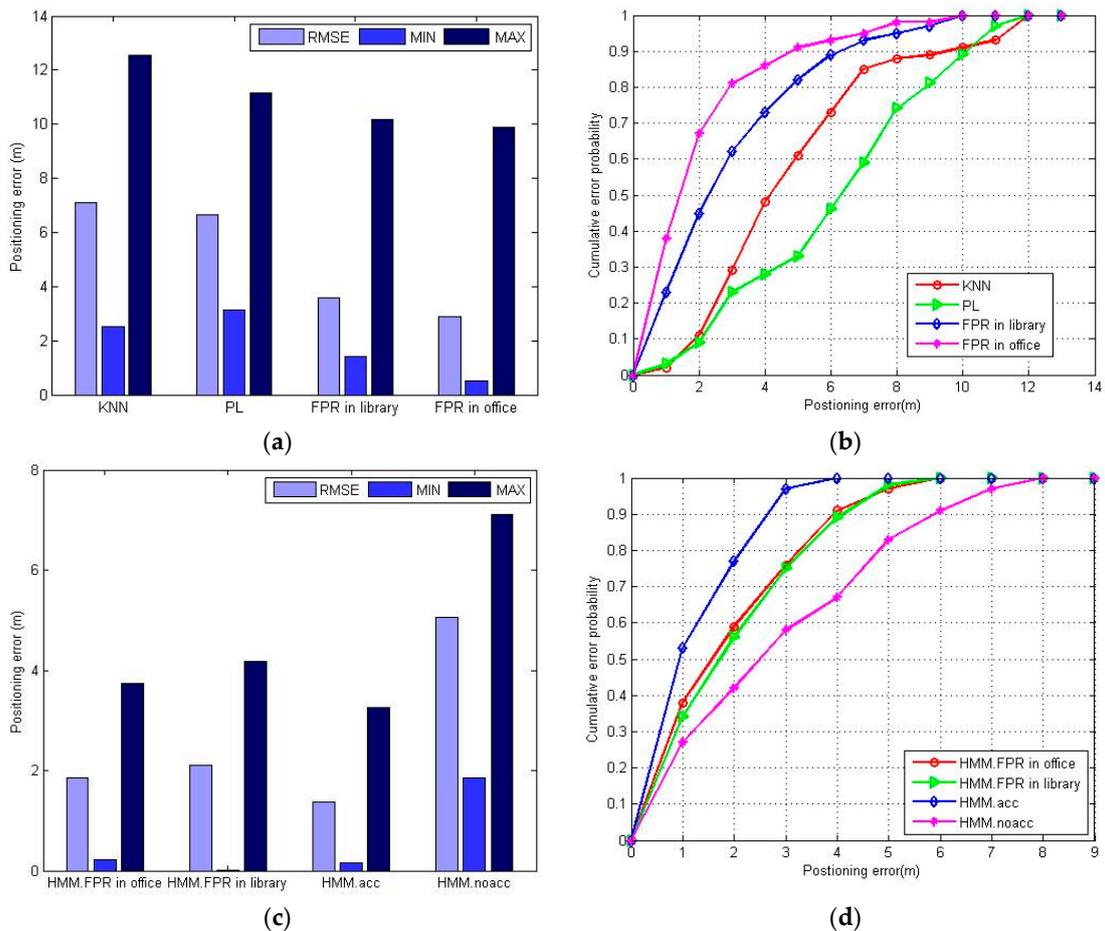


Figure 17. Cont.



**Figure 17.** Estimated positions by different positioning methods in the open area: (a) KNN; (b) PL; (c) FPR; (d) HMM.FPR; (e) HMM.acc; (f) HMM.noacc.



**Figure 18.** The positioning error of the different algorithms and methods: (a) maximum positioning error, minimum positioning error and RMSE of different positioning algorithms; (b) CDF of different positioning algorithms; (c) maximum positioning error, minimum positioning error and RMSE of different positioning methods; (d) CDF of different positioning methods.

It is more obvious in Figure 18a that the FPR algorithm performs better than the KNN and PL algorithms. To compare the influence of the different scenarios, we give the RMSE (Root Mean Square Error) and the CDF (Cumulative Distribution Function) of the FPR algorithm in the office scenario in Figure 18a,b. Although the FPR algorithm shows a slight performance degradation in the open area, the RMSE of this algorithm is still less than 4 m, and more than 70% of the positioning errors are no more than 4 m as shown in Figure 18b. The influence of the different scenarios indicate that the FPR algorithm has better positioning accuracy in the space-constrained area.

In Figure 18c,d. we also give the error results of the HMM.FPR method in the office. Unlike the FPR algorithm, the HMM.FPR works well in both scenarios. The RMSE of the HMM.FPR is about 2 m and the accuracy of the HMM.FPR within 3 m is about 75%, which is acceptable for pedestrians in an indoor area. Also, this method mitigates the effect of the direction-hopping of the KNN algorithm in the open area. As expected, the HMM.FPR has better positioning accuracy than HMM.noacc, but one that is worse than HMM.acc. The experimental results show that the proposed HMM.FPR method can obtain satisfactory accuracy in pedestrian dynamic positioning without using other sensors.

## 6. Conclusions

In this paper. A method for precise pedestrian positioning in an indoor environment using a trained HMM with a fuzzy pattern recognition algorithm was proposed. The presented fuzzy pattern recognition algorithm can offer better positioning accuracy during pedestrian walking because this algorithm utilizes the RSSI variation trend of the different locations to replace the specific RSSI values, which display significant deviations during dynamic measurement. To improve the positioning accuracy and mitigate the effect of direction-hopping, we introduce the trained HMM with the FPR algorithm in the WLAN fingerprint system. We use a rebuilt radio map and training trajectories to train the HMM with the FPR algorithm; thus, the most likely position of the pedestrian can be estimated by employing the Viterbi algorithm in the HMM. In our experiments, we demonstrate that the fuzzy pattern recognition algorithm has acceptable positioning accuracy and good environmental adaptability, and the accuracy is better after we embedded the fuzzy pattern recognition algorithm into the trained HMM. Future studies will focus on optimizing features of the proposed method, such as the AP selection, planned path selection and multi-sensor fusion.

**Acknowledgments:** This work is supported by the National Science-Technology Support Plan Projects of China (No. 2015BAK22B02, No. 2014BAH10F00).

**Author Contributions:** Yepeng Ni and Jianbo Liu conceived and designed the experiments; Yepeng Ni and Yaxin Bai performed the experiments; Yepeng Ni and Shan Liu analyzed the data; Jianbo Liu contributed analysis tools; Yepeng Ni wrote the paper. All authors read and approved the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest. The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

## References

1. Cheng, J.; Cai, Y.; Zhang, Q.; Yan, C. A new three-dimensional indoor positioning mechanism based on wireless lan. *Math. Probl. Eng.* **2014**, *2014*, 67–75. [[CrossRef](#)]
2. Caron, C.; Chamberland-Tremblay, D.; Lapierre, C.; Hadaya, P.; Roche, S.; Saada, M. *Indoor Positioning*; Springer US: New York, NY, USA, 2008; pp. 553–559.
3. Bahl, B.P.; Padmanabhan, V.N. Radar: An in-building RF-based user location and tracking system. In Proceedings of the IEEE Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies ( INFOCOM 2000), Tel Aviv, Israel, 26–30 March 2000.
4. Bahl, P.; Padmanabhan, V.N.; Balachandran, A. *Enhancements to the Radar User Location and Tracking System*; Technical Report MSR-TR-2000-12; Microsoft Corporation: Redmond, WA, USA, 2000.
5. Ahmed, S.H.; Bouk, S.H.; Mehmood, A.; Javaid, N.; Iwao, S. Effect of fast moving object on RSSI in WSN: An experimental approach. *Comput. Sci.* **2012**, *281*, 43–51.

6. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286. [[CrossRef](#)]
7. Wallbaum, M.; Wasch, T. *Markov Localization of Wireless Local Area Network Clients*; Springer: Berlin, Germany; Heidelberg, Germany, 2004; pp. 1–15.
8. Wallbaum, M.; Spaniol, O. Indoor Positioning Using wireless local Area Networks. In Proceedings of the IEEE John Vincent Atanasoff International Symposium on Modern Computing, Sofia, Bulgaria, 3–6 October 2006; pp. 17–26.
9. Seitz, J.; Vaupel, T.; Meyer, S.; Gutiérrez Boronat, J.; Thielecke, J. A Hidden Markov Model for Pedestrian Navigation. In Proceedings of the Positioning Navigation and Communication, Dresden, Germany, 11–12 March 2010; pp. 120–127.
10. Liu, J.; Chen, R.; Pei, L.; Chen, W.; Tenhunen, T.; Kuusniemi, H.; Kroger, T.; Chen, Y. Accelerometer Assisted Robust Wireless Signal Positioning Based on a Hidden Markov Model. In Proceedings of the Position Location & Navigation Symposium, Indian Wells, CA, USA, 4–6 May 2010; pp. 488–497.
11. Liu, J.; Chen, R.; Pei, L.; Guinness, R.; Kuusniemi, H. A hybrid smartphone indoor positioning solution for mobile LBS. *Sensors* **2012**, *12*, 17208–17233. [[CrossRef](#)] [[PubMed](#)]
12. Coleri Ergen, S.; Tetikol, H.S.; Kontik, M.; Sevlian, R.; Rajagopal, R.; Varaiya, P. Rssi-fingerprinting-based mobile phone localization with route constraints. *IEEE Trans. Veh. Technol.* **2014**, *63*, 423–428. [[CrossRef](#)]
13. Alexander, M.N. Energetics and optimization of human walking and running: The 2000 Raymond Pearl memorial lecture. *Am. J. Hum. Biol.* **2002**, *14*, 641–648. [[CrossRef](#)] [[PubMed](#)]
14. Pu, C.C.; Chung, W.Y.; Pu, C.C.; Chung, W.Y. Indoor RSSI characterization using statistical in wireless sensor network. *J. Korea Inst. Inf. Commun. Eng.* **2007**, *11*, 2172–2178.
15. Rabiner, L.; Juang, B.H. *Fundamentals of Speech Recognition*; Pearson Education, First Indian Reprint; Prentice-Hall: Englewood Cliffs, NJ, USA, 1993.
16. Ayodele, T.O. *Introduction to Machine Learning*; MIT Press: Cambridge, MA, USA, 2004; pp. 104–138.
17. Xia, Y.; Zhang, Z.; Ma, L. Radio map updated method based on subscriber locations in indoor wlan localization. *Syst. Eng. Electron. J.* **2015**, *26*, 1202–1209. [[CrossRef](#)]
18. Forney, G.D. The Viterbi algorithm. *Proc. IEEE* **1973**, *61*, 268–278. [[CrossRef](#)]
19. Castro, P.; Chiu, P.; Kremenek, T.; Muntz, R. A Probabilistic Room Location Service for Wireless Networked Environments. In Proceedings of the Ubicomp 2001: Ubiquitous Computing, Third International Conference Atlanta, Georgia, USA, 30 September–2 October 2001; pp. 18–34.
20. Madigan, D.; Einahrawy, E.; Martin, R.P.; Ju, W.H.; Krishnan, P.; Krishnakumar, A.S. Bayesian indoor positioning systems. In Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), Miami, FL, USA, 13–17 March 2005.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).