

AiFiMatch: Accurate Indoor Map Matching Algorithm Based on Activity Detection and Crowd-sourced Wi-Fi

Wenping Yu*, Jianzhong Zhang†, Jingdong Xu†, Yuwei Xu†

College of Computer and Control Engineering, Nankai University, Tianjin, China

*{yuwenping}@mail.nankai.edu.cn

†{zhangjz, xujd, xuyw}@nankai.edu.cn

Abstract—Map matching has been widely investigated in indoor pedestrian navigation to improve positioning accuracy and robustness. In this paper, we present AiFiMatch: a map matching algorithm that can provide accurate pedestrian walking trajectory tracking based on Hidden Markov model (HMM) for smartphone users. AiFiMatch abstracts the indoor map using a directed graph model in which the location-related activities, such as turn, walking stairs, taking the elevator, are directed edges, indoor road segments between these special locations are nodes. With the help of a novel HMM model, AiFiMatch maps pedestrian’s walking trajectory subset sequence to indoor map based on activity detection (AD). Simultaneously, Wi-Fi fingerprints can be bound to physical locations by timestamp. AiFiMatch can automatically construct and update radio map of indoor road segments via crowdsourcing. With this radio map, AiFiMatch efficiently solves the multiple hypotheses problem. We have evaluated our proposed algorithm using smartphones in the fifth floor of a teaching building on campus. Experimental results show that AiFiMatch can accurately track a pedestrian even without knowing the initial point in the experimental environment. The mean offline positioning error is about 1.24 m. The results also demonstrate that AiFiMatch significantly improves the convergence speed in buildings with Wi-Fi infrastructure.

Index Terms—Map Matching, Hidden Markov Model, Activity Detection, Crowdsourced Wi-Fi

I. Introduction

Location based services (LBS) have spilled over into all aspects of people’s daily life. Outdoors, global positioning system (GPS) is able to provide a reasonably good positioning [1]. However, robust and effective indoor pedestrian positioning is still at its infancy due to the blockage of GPS signals.

It is common for indoor positioning schemes to leverage on pre-installed infrastructure like Wi-Fi, UWB, RFID, Bluetooth, and so on [2], [3]. These systems offer absolute positioning with a limited error level, but rely on expensive hardware or time-consuming pre-training. At present, motion sensors, including accelerometer, gyroscope and magnetometer, are widely used in smartphones. Therefore, dead reckoning (DR) techniques using motion sensors are another way for indoor positioning, which derive the current position by adding the estimated displacement to the previous one. DR techniques are known as pedestrian dead

reckoning (PDR) for pedestrian tracking and positioning [4], [5]. The PDR technique itself does not provide a final solution to the pedestrian positioning problem and another technique is required to provide the initial position estimates.

The positioning error of PDR technique mainly comes from step length and direction estimation errors and even small errors will be magnified by cumulation over time. Indoor map information may be connected with PDR, namely map matching algorithm [6]. Several map matching algorithms have been proposed to constrain the cumulative errors [1], [4], [7], [8]. One algorithm uses activity-related locations as virtual landmarks to find the optimal trajectory of a pedestrian. The activity-related locations are special locations of a building, such as an elevator, a stair, a corner, and so on. When a pedestrian passes these locations, the pedestrian’s activities can be distinguished from normal walking. Human activity detection (AD) algorithms based on smartphones have attracted more and more researchers’ attention in the past decade. Given the symmetry of indoor environment, activity-based map matching algorithms fails since there may be more than one alternatives for the same activity feature or activity sequence [4].

Crowdsourcing is a low-cost and efficient way to extract useful information from data acquired by crowd participants. The crowdsourcing method has been successfully applied to construct Wi-Fi fingerprint database [9]. At present, Wi-Fi access points (AP) have been widely developed in public buildings. It is feasible to introduce Wi-Fi signal feature into map matching algorithms [10], [11]. Therefore this paper proposed AiFiMatch: a novel hidden markov model (HMM) based map matching algorithm, which integrates activity-based map matching and crowd-sourced Wi-Fi. The fundamental idea of this algorithm is that smartphones can simultaneously collect motion sensors data and Wi-Fi received signal strength (RSS) values as pedestrians walking. Wi-Fi RSS can be a new feature to eliminate the symmetry of indoor environments and reduce the number of candidate hidden states in our HMM model. Thus, AiFiMatch can get the pedestrian’s

position estimations within a short time delay.

The remainder of this paper is organized as follows: Section II gives an overview of the AiFiMatch system and its preprocessing components. Then, we detail the proposed HMM-based map matching algorithm in Section III. Section IV shows the experimental results and analysis of the AiFiMatch system. Finally, Section V and Section VI review related work and conclude this paper, respectively.

II. Related Work

A. SLAM algorithm

The SLAM algorithm was originally introduced in robotics for navigation of autonomous agents in unknown environments. It has been applied recently in indoor pedestrian tracking systems. Wi-Fi SLAM builds a landmark map of the Wi-Fi RSS based on a Gaussian Process Latent Variable Model (GP-LVM). Similarly, ActionSLAM combines dead-reckoning based on special foot-mounted inertial sensors with observations of location-related actions. SemanticSLAM generalizes the SLAM concept to work with landmarks that can be sensed by the phone sensors.

B. Map Assisted Localization

For pedestrian localization system, indoor map can bring constraints for the user's trajectory. Woodman and Harle proposed a localization approach based on MM using particle filter, which is entirely self-contained and does not rely on infrastructure. In [43], indoor map is used to filter out infeasible locations over time. In [x], a HMM model is used to match the activity sequence of smartphone user to the corresponding landmarks of indoor map.

C. Map Assisted Localization

For pedestrian localization system, indoor map can bring constraints for the user's trajectory. Woodman and Harle proposed a localization approach based on MM using particle filter, which is entirely self-contained and does not rely on infrastructure. In [43], indoor map is used to filter out infeasible locations over time. In [x], a HMM model is used to match the activity sequence of smartphone user to the corresponding landmarks of indoor map.

III. AiFiMatch Overview

AiFiMatch is an activity-based map matching system to pedestrian dead reckoning problem. In this section, we first provide an overview of the AiFiMatch system and then describe the details of its preprocessing components including PDR implementation based on smartphone, activity detection and indoor floor plan abstraction. We leave the details of the core of AiFiMatch system, the HMM based map matching module, to the next section.

A. Architecture Overview

The overall architecture of the AiFiMatch system is shown in Fig. 1. The input to the system is a indoor floor plan and time-stamped sensor data including motion data and Wi-Fi fingerprint. AiFiMatch utilizes embedded sensors of a smartphone to simultaneously collect motion data and Wi-Fi RSS values while the indoor floor plan can be manually entered or generated by crowdsourcing based indoor map construction algorithm.

Firstly, AiFiMatch starts by indoor floor plan abstraction to create a directed graph to facilitate the map matching model establishment. Secondly, the system estimates the walking distance and heading direction with data of motion sensors. Finally, location-related activities are detected with the pre-trained decision tree, and then the pedestrian's trajectory data including motion data and Wi-Fi RSS values are divided into trajectory subsets sequence.

The directed graph of indoor floor plan and walking trajectory subsets sequence are then passed to the HMM based map matching module whose outputs include the pedestrian position estimations and Wi-Fi fingerprints. These Wi-Fi fingerprints can be used to update indoor radio map. Our proposed HMM based map matching module contains three sub-modules: Candidate Extraction, Incremental Walking Trajectory Matching and an Online Viterbi Decoder. The Candidate Extraction module determines the candidate road segments from the directed graph of indoor floor plan that satisfy the spatial and Wi-Fi signal conditions. The module takes into account the errors of pedestrian heading direction estimation and the previous bound Wi-Fi fingerprint. The Incremental Walking Trajectory Matching module integrates a number of modifications to the standard HMM based map matching algorithm to take walking trajectory subsets sequence into account as well as the detected activities to enhance the accuracy of the estimated indoor road paths. Finally, the Online Viterbi Decoder uses dynamic programming to efficiently determine the most probable indoor road segments sequence.

B. Preprocessing

In the balance of this section, we give the details of three preprocessing components and leave the details of the map matching and Wi-Fi enhancement algorithm to the next section.

1) PDR Implementation Based on Smartphone: PDR is a positioning scheme that derives the current position by adding the estimated displacement to the previous one based on motion sensors carried by smartphones. In PDR scheme, the pedestrian's position is usually updated by the step and the heading direction during one step is supposed to be unchanged. The pedestrian's position after $k - 1$ steps can be denoted as (x_{k-1}, y_{k-1}) . The k -th step length and heading direction are denoted as sl_k and θ_k ,

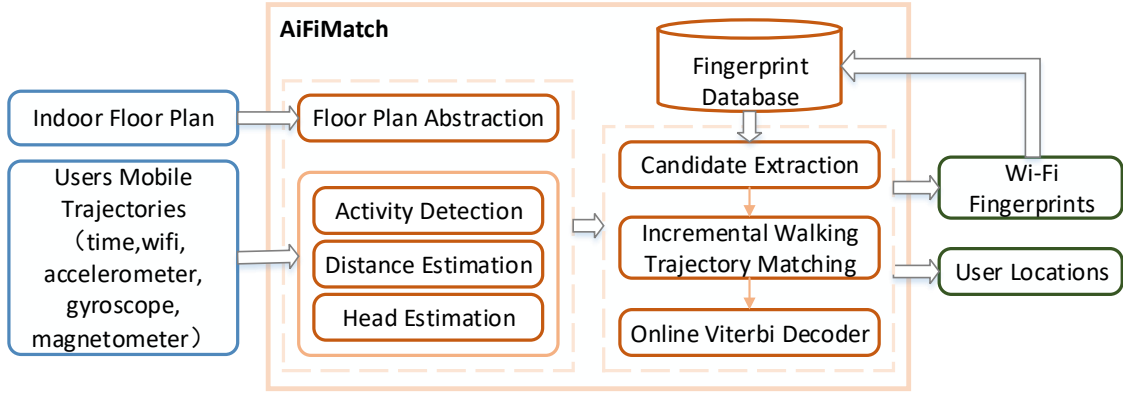
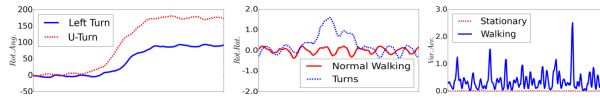
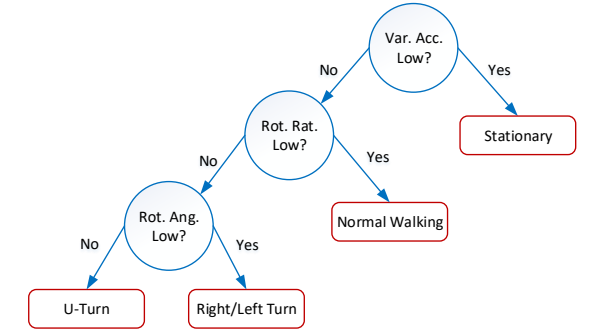


Fig. 1. AiFiMatch System Architecture.

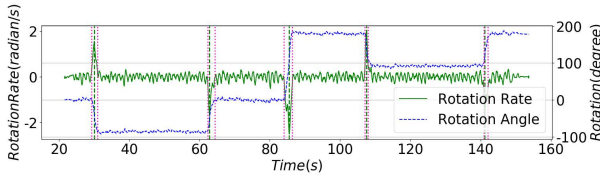
respectively. Then, the pedestrian's position after k steps can be obtained by the following formula:

$$(x_k, y_k) = (x_{k-1} + sl_k \cdot \sin \theta_k, y_{k-1} + sl_k \cdot \cos \theta_k) \quad (1)$$

The heading direction is measured by the fusion of gyroscope and magnetometer. The step length and step count estimate approach is out of scope of this paper and can be found in [12].



(a) Decision Tree for Activity Detection



(b) Activity Detected Example

Fig. 2. Decision Tree for Activity Detection and An Example

2) Activity Detection: In this paper, AiFiMatch considers four types of pedestrian activities: stationary, normal walking, turning at a corner (left or right turn) and turning around (U-turn), which would occur on the flat ground. The decision tree for activity detection and signal

TABLE I
Confusion Matrix of Activity Detection

Activity Type	Left Turn	Right Turn	U-turn	No Type
Left Turn	58	0	0	2
Right Turn	0	59	0	1
U-turn	0	0	60	0

features of each activity is shown in Fig. 2a. The top level separates walking and stationary based on the variance of the accelerometer. With the help of gyroscope, the second level uses the rotation rate to separate the normal walking and turns while the third level separates the U-turn case from the left or right turn case based on the rotation angle during the turning. Three participants with smartphones were asked to complete three activities (U-turn, left and right turn) in our experimental environment. The sample size of each activity was 60 traces. Fig. 2b shows an example for activity detection and the activity detection result is summarized in Table III-B2.

3) Indoor Floor Plan Abstraction: In the indoor environment, there are many activity-related locations such as corridor corners and entrances of rooms where pedestrians may perform different activities other than normal walking. These special locations divide the indoor roads into segments. With road segments as nodes in the form of (coordinate of first endpoint (x_1, y_1), accessible direction of first endpoint (θ_1), coordinate of second endpoint (x_2, y_2), accessible direction of second endpoint (θ_2)), the activity type from one road segment to another as directional edges in the form of (activity type (AT)), the indoor floor plan would be represented as a direction graph. Fig. 3 shows an example of indoor floor abstraction.

IV. HMM based Map Matching Algorithm

Through this section, we detail the HMM based map matching module of AiFiMatch system. We first start by providing the novel HMM model and the details of its components. Then, we describe the Wi-Fi enhancement algorithm for our proposed HMM model.

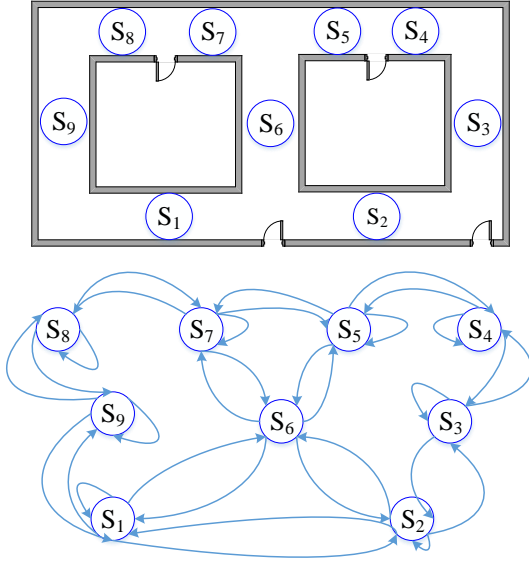


Fig. 3. Floor Plan Abstraction Example.

A. Hidden Markov Model

Taking computing power and energy limit of a smart-phone into account, our proposed map matching algorithm select HMM to match the pedestrian's trajectory subsets sequence to the directed graph of indoor floor plan. A HMM can be represented as $\lambda = (S, V, A, B, \pi)$, where:

1) $S = \{s_1, s_2, s_3, \dots, s_N\}$ is the set of possible states and $N = |S|$. In our case, each state represents an indoor road segment, that is, a node of the directed graph. Note that two or more road segments connected by a straight line can form new states (Fig. 4). Therefore, a state s_i is represented by the ordered tuple in the form of $(id, x_1, y_1, d_1, x_2, y_2, d_2)$, where id is the identification of road segment, x_1, y_1, d_1, x_2, y_2 and d_2 are different attributes of node, respectively.

2) $V = \{v_1, v_2, v_3, \dots, v_M\}$ is the set of observations from the model and $M = |V|$. For each walking trajectory of a pedestrian, PDR technique gives displacement and heading direction estimations. However, due to the interference of many kinds of metal materials to the magnetic field in buildings, the heading direction estimation has a large error. In our case, each observation is a displacement estimation and is represented by $(dist)$.

3) $A = \{a_{ij}\}$ is the state transition probability distribution, where

$a_{ij} = p\{q_{t+1} = s_j | q_t = s_i\}, i, j \leq N$, where q_t denotes the state at time t . In our case, the transition probability a_{ij} is the probability of walking to the next road segment $s_{j,t}$ given the current road segment is $s_{i,t-1}$. Intuitively, for probable transition between two road segments, the pedestrian's activity should match the activity type between the same two segments. Therefore, given the detected activity of a pedestrian $Ped_{AT}(t)$ at time t and the activity type between two segments Seg_{AT}^{ij} , AiFiMatch models this

intuition by a formula 2, where $p(Ped_{AT}(t) | Seg_{AT}^{ij})$ can be found in confusion matrix.

$$p(s_{j,t} | s_{i,t-1}) = p(s_j | s_i, Ped_{AT}(t)) = p(Ped_{AT}(t) | Seg_{AT}^{ij}) \quad (2)$$

4) $B = \{b_i(k)\}$ is the observation probability distribution in state i , where $b_i(k) = p\{z_t = v_k | q_t = s_i\}, 1 \leq i \leq N, 1 \leq k \leq M$ and z_t and q_t are the observation and state at time t , respectively. Observation probabilities also called emission probabilities represent the likelihood that a measurement resulted from a given state. In our case, given a displacement observation $z.dist(t)$, there is an emission probability $p(z.dist(t) | s_i.dist)$ for each candidate road segment s_i . If a pedestrian has passed the complete road segment s_i , AiFiMatch models the emission probability as a Gaussian Distribution:

$$f_1(z.dist(t), s_i.dist) = \frac{1}{\sqrt{2\pi}\sigma_d} e^{-\frac{(z.dist(t) - s_i.dist)^2}{2\sigma_d^2}} \quad (3)$$

where σ_d is the standard deviation of the measured displacement. Based on the distance calculation method of PDR, the displacement is in direction proportion to step length. Therefore, σ_d can be obtained based on the standard deviation of step length. Considering the situation that a pedestrian is walking on the road segment, all long enough candidate road segments should have the same probability, AiFiMatch models the situation using a formula as:

$$f_2(z.dist(t), s_i.dist) = \frac{1}{\sqrt{2\pi}\sigma_d} e^{-4.5} \cdot z.dist(t) + 3\sigma_d \leq s_i.dist \quad (4)$$

Hence, the final emission probability, $p(z | s_i)$, is modeled as:

$$p(z | s_i) = f(z.dist, s_i.dist) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_d} e^{-4.5} \cdot z.dist + 3\sigma_d \leq s_i.dist \\ \frac{1}{\sqrt{2\pi}\sigma_d} e^{-\frac{(z.dist - s_i.dist)^2}{2\sigma_d^2}}, otherwise \end{cases} \quad (5)$$

5) $\pi = \{\pi_i\}$ is the initial state distribution, where $\pi_i = p\{q_1 = S_i\}, 1 \leq i \leq N$. If the starting point or the first road segment is known, the initial state distribution is 1 since the first road segment is the only candidate; otherwise, all candidate road segments are selected by the Candidate Extraction module and the initial state distribution is uniform in all candidates. Let S_c denote the set of all candidates, and the initial state distribution is re-estimated after each step by formula 6 until the first location-related activity is detected.

$$\pi_{i,t} = \pi_{i,t-1} \cdot f(z.dist(t), s_i.dist), s_i \in S_c \quad (6)$$

B. Candidate Extraction

AiFiMatch uses the heading direction of a pedestrian and Wi-Fi fingerprint to select the candidate states. Here, we describe the extraction algorithm by heading direction and leave the Wi-Fi fingerprint extraction algorithm to the Wi-Fi Enhancement Section.

Intuitively, the heading direction of a pedestrian should match the direction of road segment. Therefore, AiFiMatch models the direction difference by the following function to select the candidates:

$$g_1 = g_1(Ped_{dir}, s) = \begin{cases} 1, & \text{if } |Ped_{dir} - s.d_j| < H_{TH}, j = 1, 2 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where Ped_{dir} denotes the heading direction of a pedestrian, s is a state, H_{TH} is the threshold for candidate extraction, which is set to 55° based on the experiments.

C. Optimal State Sequence Estimation

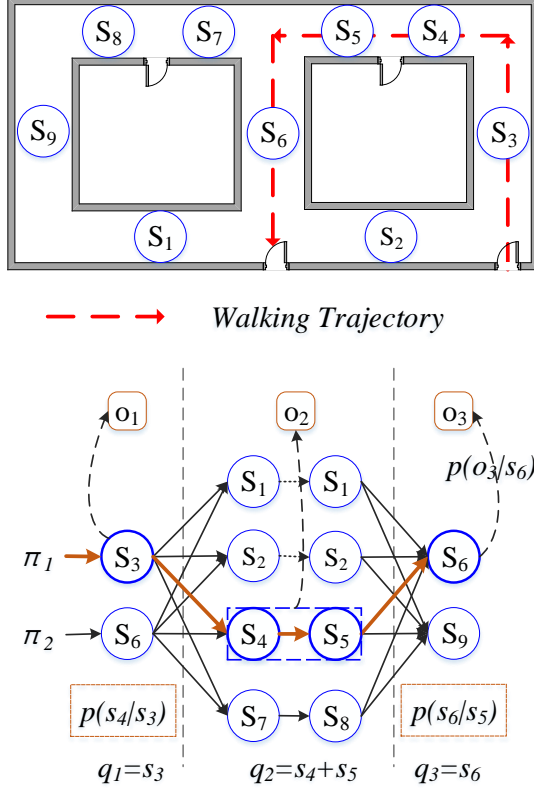


Fig. 4. Illustration of the Viterbi Decoding.

For AiFiMatch to operate in real-time, it cannot wait until the whole sequence is available. Hence, in an incremental manner, AiFiMatch uses a sliding window on the Map Matching module. Every time a new step of a pedestrian is detected, the HMM parameters are calculated for the new introduced sensors data and the associated candidate states. The online Viterbi algorithm [13] is then applied to compute the maximum likelihood sequence of states for the current window using dynamic programming. Fig. 4 illustrates the proposed HMM model Viterbi decoding. The sliding window size is 3 and the decoded sequence is colored in blue. The initial states probability is also updated as the sliding window moves.

Once the HMM parameters are estimated, we can use the Viterbi algorithm to get the most probable hidden

states sequence for a given observation sequence. During the beginning process of the algorithm, if the number of states is too small, the most probable states sequence is not always the correct one, especially when the starting point of pedestrian is unknown. Therefore, we give a criteria by formula $c = p_{fir}/p_{sec}$ to determine the status of this algorithm, where p_{fir} is the highest probability of the states sequence and p_{sec} is the second highest probability of the states sequence. We set a threshold, if c is greater than or equal to the threshold, we choose the Viterbi decoding result. The selected hidden states sequence represents the pedestrian's passed indoor road segments.

D. Wi-Fi Enhancement

Map matching offers an important source of information and an excellent way to improve the position estimations, especially in narrow corridors, but it has some limitations when the indoor environment presents symmetries that generates multiple hypotheses. Supposing that a pedestrian walks along this segments sequence (s_3, s_4, s_5, s_6) , as shown in Fig. 4, for the given observable states, it is directly to be seen that the segments sequence (s_6, s_7, s_8, s_9) may have almost the same matching probability as the actual one. In this situation, traditional HMM based map matching algorithm fails [4].

AiFiMatch introduces Wi-Fi dissimilarity to distinguish multiple hypotheses due to the symmetry of building structure. After AiFiMatch determines a pedestrian's walking trajectory by the Online Viterbi Algorithm, the pedestrian's positions can be derived by PDR using the two endpoints of determined segments as the starting point. Therefore, Wi-Fi fingerprints collected synchronously during the pedestrian walking can be bound to the corresponding positions by time alignment and then update the Wi-Fi fingerprint database of indoor environment where the pedestrian is located. In order to improve the HMM based map matching algorithm described above, the paper not only binds Wi-Fi fingerprints to physical positions but also road segments. All Wi-Fi fingerprints of one road segment are sorted in the road segment's direction. For two fingerprints f_x, f_y , and MAC address set of their Access Points X, Y , define dissimilarity J_δ between f_x, f_y as follows:

$$J_\delta(f_x, f_y) = 1 - J(X, Y) = \frac{|X \cup Y| - |X \cap Y|}{|X \cup Y|} \quad (8)$$

where Equation 8 is known as Jaccard distance. With this dissimilarity of two fingerprints, we define the similarity function $L(S_a, S_b)$ between two fingerprint sequences $S_a = (f_1^a, f_2^a, \dots, f_w^a)$, $S_b = (f_1^b, f_2^b, \dots, f_k^b)$:

$$L(S_a, S_b) = \min \left(\frac{1}{w} \sum_{t=1}^w J_\delta(f_t^a, f_{i+t}^b), 0 \leq i \leq k - w, \right. \\ \left. \frac{1}{w} \sum_{t=1}^w J_\delta(f_t^a, f_{j-t}^b), w + 1 \leq j \leq k + 1 \right) \quad (9)$$

where $w = |S_a|$, $k = |S_b|$ and $2 < w < k$. Given segment candidate set G and the set R of all road segments bound with Wi-Fi fingerprints sequence, Finally, AiFiMatch models the dissimilarity of two Wi-Fi fingerprint sequences and the different stages of fingerprint database by the following function to select the candidates:

$$g_2(S_{ped}, S_i) = \begin{cases} 0, & \text{if } i \neq \underset{j}{\operatorname{argmin}} \{L(S_{ped}, S_j), S_j \in G, G \subseteq R\} \\ 0, & \text{if } \min \{L(S_{ped}, S_i), S_i \in G \cap R\} > d \\ 1, & \text{otherwise} \end{cases} \quad (10)$$

where d is the threshold to distinguish two Wi-Fi fingerprint sequences. A pilot study is conducted to evaluate the dissimilarities between two fingerprint sequences of same and different road segments and the experimental result is shown in Fig. 5. In the paper, d is set to 0.63 based on the experiment.

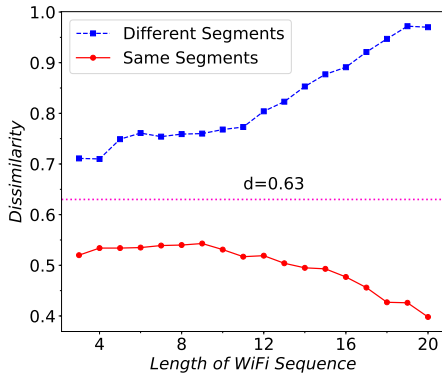


Fig. 5. Dissimilarity of Two Wi-Fi Fingerprint Sequences.

E. Algorithm

Algorithm 1 AiFiMatch online map matching algorithm

Input: Sensor data up to current time t : $data_{0:t}$
Input: Abstract Indoor Floor Plan: fp
Input: Length of state sequence up to last time $t-1$: k
Input: State Sequence up to last time $t-1$: $S_k(t-1)$
Input: Array of Viterbi variable and backward pointer:
 $VtbArr_k(t-1)$
Output: Pedestrian's position estimation at current time
 t : $Ped_{loc}(t)$
Output: Optimal state sequence estimation at current
time t : $Q(t)$

```

1:  $Ped_{AT}(t), at \leftarrow activity\_detect(data_{0:t})$ 
2: if  $Ped_{AT}(t) \neq None$  then
3:    $st \leftarrow at$ 
4:   // Calculate the transition probability
5:    $S(t) \leftarrow []$ 
6:   for  $s_i$  in  $S_k(t-1)$  do
7:     // Select next segments according to  $fp$ 
8:      $S^i(t) \leftarrow next(fp, s_i)$ 
9:     // Extract segments according to Wi-Fi data
10:     $S^{i,wifi}(t) \leftarrow extract(data_{st:t}^{wifi}, S^i(t))$ 
11:    for  $s_j$  in  $S^{i,wifi}(t)$  do
12:       $tr[i][j] \leftarrow p(s_j | s_i, Ped_{AT}(t))$ 
13:    end for
14:     $S(t).extend(S^{i,wifi}(t))$ 
15:  end for
16:  // Calculate the observation probability
17:   $z_t \leftarrow dead\_reckon(data_{st:t})$ 
18:  for  $s_i$  in  $S(t)$  do
19:     $ob[i] \leftarrow p(z_t | s_i)$ 
20:  end for
21:   $VtbArr_{k+1}(t) \leftarrow viterbi(VtbArr_k(t-1), ob[], tr[][])$ 
22:   $k \leftarrow k + 1$ 
23:   $S_k(t) \leftarrow S(t)$ 
24: else
25:    $ob \leftarrow [], S_k(t) \leftarrow S_k(t-1)$ 
26:   // Update the observation probability
27:    $z_t = dead\_reckon(data_{st:t})$ 
28:   for  $s_i$  in  $S_k(t)$  do
29:      $ob[i] \leftarrow p(z_t | s_i)$ 
30:   end for
31:    $VtbArr_k(t) \leftarrow viterbi(VtbArr_k(t-1), ob[])$ 
32: end if
33: // Optimal State Sequence
34:  $Q(t) \leftarrow None$ 
35: if  $check\_status(VtbArr^k(t)) = Convergence$  then
36:    $Q(t) \leftarrow viterbi\_decode(VtbArr^k(t))$ 
37: end if
38: // Update the location estimation according to  $Q(t)$ 
39:  $Ped_{loc}(t) \leftarrow update\_location(data_{st:t}, Q(t))$ 

```

V. Evaluation

A. Environment Setup

In this section, we show our evaluation for the performance of AiFiMatch in real-world environment, we conducted experiments in the fifth floor of a teaching hall on campus. Approximately, we covered a $78.95m \times 56.40m$ floor plan, as shown in Fig. 6. A prototype is implemented on an Android version 6.0.1 Xiaomi III smartphone. The prototype samples the accelerometer, gyroscope and magnetometer sensors at 50Hz and Wi-Fi at 1Hz. Three participants (two males and one female) were asked to complete the experiments. The participants were asked to walking along Trajectory No.1 (T_1), Trajectory No.2 (T_2) and Trajectory No.3 (T_3) in a constant speed. Each trajectory was repeated six times by all participants. In order to record the walking trajectories, all the participants' shoes were painted with colored powder. This offered the ground truth.

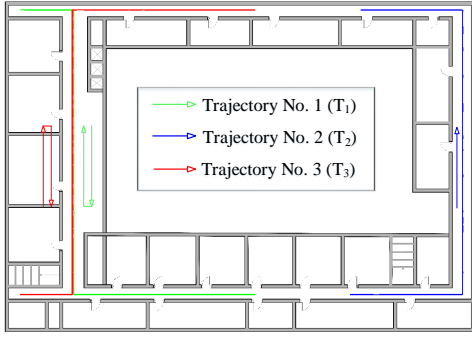


Fig. 6. Experiment Environments and Trajectories.

B. Performance of Map Matching

To evaluate AiFiMatch map matching performance, we start by showing the online positioning performance of AiFiMatch and basic PDR technique. Then, we discussed the convergence performance as compared to an activity sequence-based map matching algorithm proposed by Zhou et al [4]. Finally, we show the offline positioning results when using AiFiMatch.

1) Online Positioning Performance: Euclidean distance between the estimated position and the ground truth is introduced to indicate positioning accuracy. Fig. 7 shows the online positioning results of all trajectories for basic PDR algorithm with known initial point and AiFiMatch without known initial point. Here, the mean position of all segment candidates' starting point is taken as initial point of AiFiMatch algorithm. Generally, for one trajectory, the greater the traveled distance, the larger the positioning error of PDR due to cumulative errors, Fig. 7 shows the trend. However, after passing a number of steps, the traveled segment sequence determined by AiFiMatch even without known the initial point and the cumulative errors are eliminated successfully.

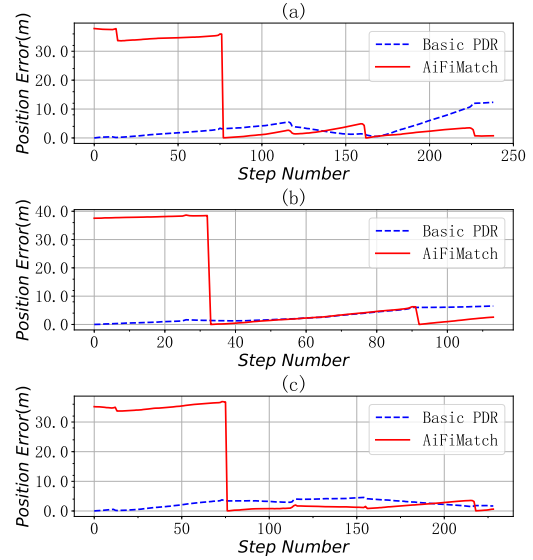


Fig. 7. Online positioning results for Each Trajectory.

2) Convergence Speed: We compare the performance of AiFiMatch algorithm in terms of convergence to map matching algorithm proposed by Zhou et al. Distance traveled before converging to a unique states sequence reflects the convergence speed. The greater the traveled distance, the slower is the convergence speed. Fig. 8 shows the traveled distance before convergence for Zhou et al, AiFiMatch without Wi-Fi enhancement and AiFiMatch with Wi-Fi enhancement. At the initial stage, fingerprint database is empty (T_1), AiFiMatch and Zhou et al. both successfully converge even without known the initial point. AiFiMatch without Wi-Fi enhancement and map matching algorithm proposed by Zhou et al. fail to converge (∞ means T_2 cannot be converged.) due to the symmetry of building structure. However, with the help of Wi-Fi signal, T_2 reaches convergence quickly. Mostly, with Wi-Fi enhancement, the traveled distance is much shorter than without Wi-Fi enhancement(T_2, T_3).

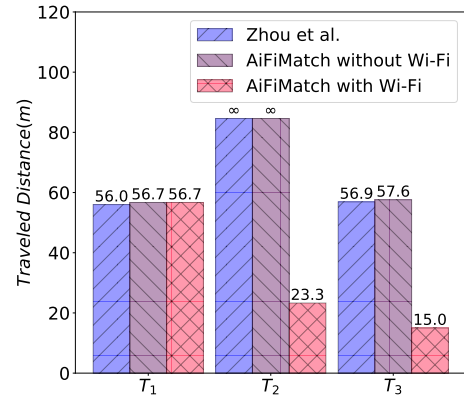


Fig. 8. Distance Traveled before Convergence for Each Trajectory.

3) Offline Positioning Performance: The offline positioning results are derived retrospectively after matching the walking trajectory of a pedestrian to a unique road segments sequence by AiFiMatch. Fig. 9 shows the tracking trajectories and some details are summarized in Table 2. AiFiMatch system tracked pedestrians' trajectories accurately in the experiment environments and the mean error of the offline positioning is about 1.24 m.

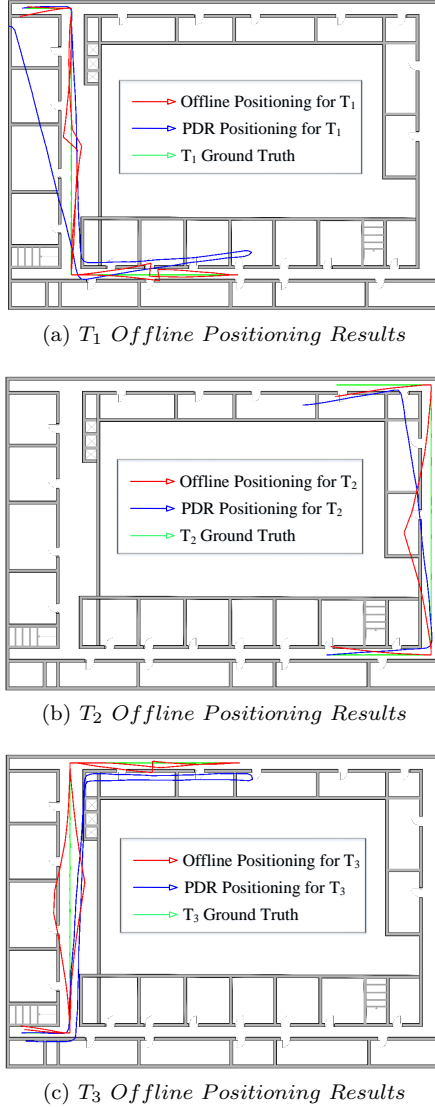


Fig. 9. Environment Setup and Offline Positioning Results for Each Trajectory

VI. Conclusion

This paper proposed AiFiMatch: a novel map matching Algorithm based on activity detection and crowd-sourced Wi-Fi for indoor pedestrian tracking and positioning. AiFiMatch leverages the smartphone's motion sensors to detect different activities using a decision tree and then divides the walking trajectory into trajectory subset sequence. The HMM model is used to match walking

trajectory subset sequence to road segments sequence. AiFiMatch can also construct and update fingerprint database based on road segment by crowd-sourcing. We provided the AiFiMatch's system architecture and presented the details of different modules. The performance of AiFiMatch has been evaluated by experiments in a building on campus. The results show that AiFiMatch can track pedestrian's trajectory accurately even without known initial point. In addition, with the help of Wi-Fi fingerprints, AiFiMatch reaches converge quickly and effectively solves the multiple hypotheses caused by symmetry of building structure.

References

- [1] F. Zampella, A. R. J. Ruiz, and F. S. Granja, "Indoor positioning using efficient map matching, rss measurements, and an improved motion model," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1304–1317, 2015.
- [2] P. Bahl, V. N. Padmanabhan, and A. Balachandran, "Enhancements to the radar user location and tracking system," *Microsoft Research*, vol. 2, no. MSR-TR-2000-12, pp. 775–784, 2000.
- [3] L. M. Ni, Y. Liu, Y. C. Lau et al., "Landmarc: indoor location sensing using active rfid," *Wireless networks*, vol. 10, no. 6, pp. 701–710, 2004.
- [4] B. Zhou, Q. Li, Q. Mao et al., "Activity sequence-based indoor pedestrian localization using smartphones," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 5, pp. 562–574, 2015.
- [5] —, "Alimc: Activity landmark-based indoor mapping via crowdsourcing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2774–2785, 2015.
- [6] H. Aly and M. Youssef, "semmatch: Road semantics-based accurate map matching for challenging positioning data," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2015, p. 5.
- [7] H. Abdelnasser, R. Mohamed, A. Elgohary et al., "Semantic-slam: Using environment landmarks for unsupervised indoor localization," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1770–1782, 2016.
- [8] N. Fu, J. Zhang, W. Yu et al., "Crowdsourcing-based wifi fingerprint update for indoor localization," in *Proceedings of the ACM Turing 50th Celebration Conference-China*. ACM, 2017, p. 34.
- [9] X. Zhang, Z. Yang, C. Wu et al., "Robust trajectory estimation for crowdsourcing-based mobile applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1876–1885, 2014.
- [10] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. ACM, 2009, pp. 336–343.
- [11] J. Seitz, J. Jahn, J. G. Boronat et al., "A hidden markov model for urban navigation based on fingerprinting and pedestrian dead reckoning," in *2010 13th Conference on Information Fusion (FUSION)*. IEEE, 2010, pp. 1–8.
- [12] H. Wang, S. Sen, A. Elgohary et al., "No need to war-drive: Unsupervised indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 197–210.
- [13] J. Bloit and X. Rodet, "Short-time viterbi for online hmm decoding: Evaluation on a real-time phone recognition task," in *ICASSP*. IEEE, 2008, pp. 2121–2124.

TABLE II
Evaluation Results

Trajectory No.	Length (m)	Mean Error (m)	Step Number	Detected Step	Activity Number	Detected Activity
1	176.1	1.09	238	240	5	5
2	84.6	1.65	114	114	3	3
3	175.9	1.19	228	227	5	5