

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/338883273>

Path Distance Based Map Matching for Wi-Fi Fingerprinting Positioning

Article in *Future Generation Computer Systems* · January 2020

DOI: 10.1016/j.future.2020.01.053

CITATIONS

0

READS

48

4 authors, including:



Pan Chen

7 PUBLICATIONS 80 CITATIONS

[SEE PROFILE](#)



Fuqiang Gu

National University of Singapore

23 PUBLICATIONS 260 CITATIONS

[SEE PROFILE](#)



Jianga Shang

China University of Geosciences

41 PUBLICATIONS 305 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Context sensing and UI design [View project](#)



image understanding [View project](#)



Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Path distance-based map matching for Wi-Fi fingerprinting positioning

Pan Chen^{a,c}, Xiaoping Zheng^{a,c}, Fuqiang Gu^{b,*}, Jianga Shang^{a,c}

^a Faculty of Information Engineering, China University of Geosciences, Wuhan, China

^b Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Canada

^c National Engineering Research Center for Geographic Information System, Wuhan, China

ARTICLE INFO

Article history:

Received 11 April 2019

Received in revised form 12 December 2019

Accepted 28 January 2020

Available online xxxx

Keywords:

Indoor positioning

Navigation

Map matching

Fingerprinting

Location-based services

ABSTRACT

Map matching is a commonly-used technique that employs spatial constraints to improve positioning results. While map matching can improve the positioning performance to a large extent, existing map matching methods consider only adjacent transitions between reference points (RPs). This makes these map matching methods depend highly on the sampling size of RPs. To reduce the influence of the RPs' sampling size, a novel map matching method called PDMatching is proposed in this paper, which considers both adjacent and non-adjacent transitions. These transitions are described based on the path distance of the RP sequence obtained by the shortest path algorithm. Compared to the commonly-used Euclidean distance, the path distance is more suitable for map matching as it takes into account spatial constraints. It allows to estimate the transition distance more accurately, which can further improve the positioning accuracy. To infer the location of a user, the student's t-distribution is used to transform the path distance into a transition probability, from which the location can be obtained via the Viterbi algorithm. Extensive experiments have been conducted to evaluate the proposed PDMatching in a large museum environment. Experimental results show that the proposed PDMatching can achieve a mean localization error of 3.4m and 4.6m for uniform and varying speed modes, respectively, which outperforms the state-of-the-art methods (e.g., MapCraft, VTrack, XINS). Moreover, the PDMatching is more robust to the sampling size of RPs than other methods.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Indoor positioning method plays an important role in various location-based services (LBS) [1,2]. Wi-Fi fingerprinting is one of the most popular indoor positioning methods because of the ubiquitous availability of Wi-Fi infrastructure. It includes an offline training phase and an online localization phase. In the training phase, fingerprints are collected at a grid of known locations (referred as reference points) and stored in a fingerprint database (also known as radio map). A fingerprint can be a vector of received signal strength indicator (RSSI) [3,4] or channel station information (CSI) [5] from visible access points (APs). CSI-based methods can usually achieve better accuracy than RSSI-based methods, but they have poorer coverage [6]. In the localization phase, the newly-collected fingerprint is compared with those stored in the fingerprint database, and then the corresponding location can be computed via certain methods such as K-nearest neighbors (KNN). The accuracy of Wi-Fi positioning is affected by the layout of indoor environments, human movement, and other

obstacles that may cause signal multipath, shading, and reflection [7,8]. To improve the localization accuracy, one can combine different localization signals or technologies (e.g., inertial sensors, Wi-Fi, Bluetooth) [9]. However, these signals or technologies may not be always available in the environment. On the other hand, spatial context [10], such as landmarks and floor plans, can be used to improve the accuracy of Wi-Fi fingerprinting without requiring additional hardware.

Landmarks are defined as specific location points with certain unique patterns identifiable from the smartphone sensor readings. The positioning accuracy can be improved by integrating landmarks [11–14]. For instance, an improvement of more than 35% localization accuracy over Wi-Fi fingerprinting is witnessed in [11] when combining landmarks. However, landmark-based methods rely highly on the accuracy of landmark detection. An omission or commission error of landmark detection will lead to a large positioning error [10]. Apart from landmarks, spatial constraints imposed by walls and other obstacles can also be used to improve the positioning accuracy, which is usually done through a particle filter [15–19]. The particle filter-based methods estimate locations with the statistical results of all the particles, of which the coordinates are inferred from the last

* Corresponding author.

E-mail address: fuqiang.gu@mail.utoronto.ca (F. Gu).

location. The particles propagate forward based on a system/state model and their associated weights are corrected by using the spatial constraints. However, particle filtering requires a large number of particles to maintain a satisfactory accuracy, which is computationally expensive for resource-limited mobile platforms such as smartphones. Moreover, the inference of particle filtering methods uses only the last location and current sensor readings to estimate the current location rather than the whole previous information. This may result in a locally optimal result, leading to a large positioning error.

To obtain a globally optimal estimation, map matching-based positioning methods are often used, which can make use of a sequence of observations and the previous location estimates to infer the current location. Commonly-used map matching methods are usually based on a hidden Markov model (HMM) [20–22] or conditional random field (CRF) [23,24], which can improve the positioning accuracy. For example, the accuracy (measured by mean error) achieved by the method in [22] that uses a heading-based HMM doubles the accuracy achieved by the common weighted KNN method. However, these matching methods consider only adjacent transitions and set their transition probability manually. The problem of allowing adjacent transitions only is that it may lead to a large localization error when the RPs are too dense. For example, if the distance between two neighboring RPs is 0.3 m and a user's step length is 0.6 m, then the matching methods considering only adjacent transitions will lead to a large matching error. The problem of manually setting the transition probability is that it ignores the location distance between two RPs in indoor spaces. As the transition probability describes the movement between the RPs, the probability of a large location distance should be smaller than that of small location distance, since people may not be possible to move to a location that is far away from the previous location during a short period of time (e.g., 1 s).

In this paper, a novel HMM-based map matching method called PDMatching is proposed, which is robust to the sampling size of RPs. In the PDMatching method, the transition probability matrix is constructed by considering both adjacent and non-adjacent RPs, rather than only adjacent RPs that are commonly used in conventional HMM-based matching methods. As the proposed method allows non-adjacent transitions, the user's location could be mapped to a non-adjacent RP that is closer to the true location than adjacent RPs. This will result in a more accurate matching than conventional methods that use adjacent transitions only. The transition probability is calculated based on the shortest path distance obtained by the Dijkstra, rather than on the Euclidean distance. Compared to the Euclidean distance, which simply computes the distance between two points without considering spatial constraints (e.g., walls), the path distance can consider spatial constraints, which allows to express the transitions between RPs more accurately. Then, the student's t -distribution is taken to transform the path distance into corresponding probability as the student's t -distribution does not require to configure parameters. After that, the locations are inferred by finding the most likely sequence of RPs with the Viterbi algorithm [25] and inserting the possible RPs into the sequence to avoid the inferred path crossing walls. Although there are certain algorithms extended from the original Viterbi can be used to accelerate the matching process, the focus of this study is to investigate the usefulness of path distance and matching with considering both adjacent and non-adjacent transitions. As the size of RPs in this study is not very large, the classical Viterbi algorithm is adopted for simplicity and the experimental results show its efficiency on the task.

Overall, our contributions are listed as follows:

1. A novel HMM-based map matching method called PDMatching is proposed, which is insensitive to the sampling size of RPs. The transition probability matrix of PDMatching is calculated based on both adjacent and non-adjacent RPs. This makes the proposed method robust to the different density of the RPs as it allows transitions in both short and long ranges.
2. The path distance, which is computed by the Dijkstra algorithm, is proposed to estimate the transitions between the RPs. Different from the commonly-used Euclidean distance, the path distance considers the spatial information of the indoor environment and leads to a more accurate estimation of the transition distance to improve the results of map matching.
3. The proposed PDMatching method is evaluated through extensive experiments. Experimental results show that the PDMatching outperforms the state-of-the-art methods and is insensitive to RP sampling sizes, walking speeds, and Wi-Fi scanning frequencies.

The rest of the paper is organized as follows. Section 2 introduces related works. Section 3 illustrates the proposed PDMatching step by step. In Section 4, the proposed PDMatching is evaluated by extensive experiments, and this paper is concluded in Section 6.

2. Related work

In recent years, many indoor positioning methods that use spatial context to improving positioning accuracy have been proposed, which are mainly categorized into landmark-based and probabilistic model-based methods.

2.1. Landmark-based methods

Landmarks are useful in improving indoor positioning accuracy without requiring additional hardware. In [26], a seamless indoor and outdoor positioning method is proposed by matching locations to the landmarks where certain activities are detected. UnLoc [27] achieves a median location error of about 1.7m in an indoor environment by using seed landmarks (e.g., staircases, elevators, turns) and organic landmarks (locations with particular ambient signatures). SemanticSLAM [11] extends the UnLoc system by implementing it under a simultaneous localization and mapping (SLAM) framework and obtains a more accurate result. APFiLoc [28] fuses landmarks with inertial sensor readings and map information through a particle filter to enhance the localization accuracy of the PDR in indoor environments. ALIMC [29] uses a tree model to detect activity landmarks to improve the accuracy of indoor mapping. To make proper placement of landmarks, Magnago et al. [13] proposed a model to measure the positioning uncertainty to minimize the number of landmarks. Besides the inertial sensors and Wi-Fi sensors, visual features from images can be also used to describe landmarks [14]. Visual landmarks require no extra infrastructure and have attracted much attention recently.

Landmarks have been used to improve indoor positioning accuracy in many works [12,11,29], since they are naturally or easy-to-deploy in indoor environments and can be identified at a low cost. Unlike sensor fusion methods [9], which require multiple types of sensors, landmark-based positioning methods do not require additional hardware and can also achieve competitive positioning accuracies. However, the performance of landmark-based methods depends highly on the accuracy of landmark detection. A wrong detection of landmarks may lead to a large error of the location estimation [10]. Besides, there are not standard rules to detect landmarks, which makes landmark-based positioning methods less generic in practice.

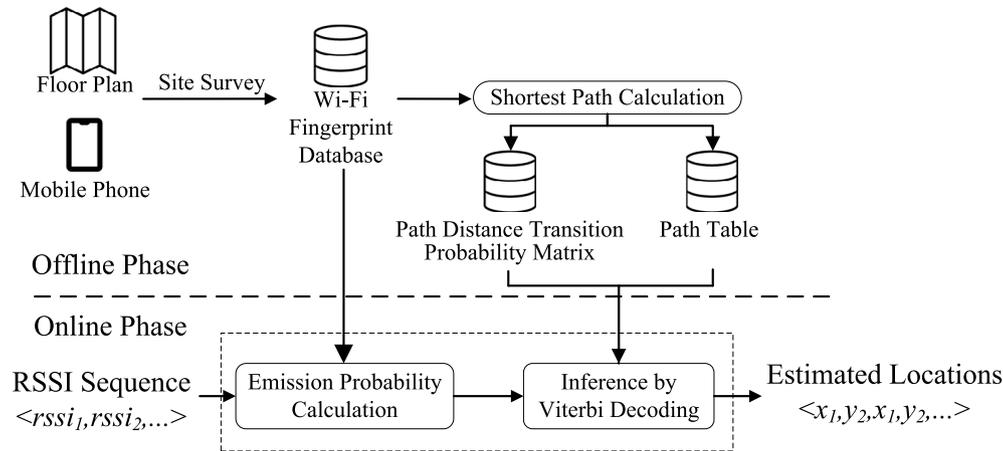


Fig. 1. System architecture of the PDMatching.

2.2. Probabilistic model-based methods

Probabilistic model-based methods usually use previous information along with spatial information to restrict locations. Compared to landmark-based methods, probabilistic model-based methods are usually more robust since they make use of a sequence of history information [30]. Common probabilistic model-based methods include particle filter, HMM, and CRF. Particle filters are usually used to integrate sensor data with spatial information (e.g., a floor plan). Zee [15] utilizes a particle filter to combine inertial sensor readings, which are used to count steps and estimate heading, with Wi-Fi fingerprints and map information to achieve accurate indoor localization. Similarly, XINS [31] also uses a particle filter to fuse inertial sensor readings and map information with Wi-Fi measurements, GPS (global positioning system) or location information transmitted from another device. In [32], the particle filter is used to combine inertial sensor data and polarized light, and a median positioning error of 4.3m is achieved. PFSurvey [33] estimates the location of fingerprints by using a particle filter to fuse inertial sensor data with a floor plan to reduce the effort of the site survey. In [34], the particle filter is used to integrate inertial sensor readings, iBeacon RSSIs, and map constraints to improve positioning accuracy as well as to enhance user experience in navigation. However, particle filtering-based methods are usually computationally expensive and can consume the smartphone's battery quickly.

To reduce the computational cost, Liao et al. [35] employed the particle filter to estimate the locations of a user on a one-dimensional Voronoi graph that can significantly reduce the number of the particles. Hilsenbeck et al. [36] proposed a graph-based and low-complexity indoor positioning method. By simplifying the indoor map structure to a linked graph, it significantly reduces the number of particles required for obtaining accurate location estimation from the pedometer and Wi-Fi measurements. Nurminen et al. [37] developed a graph-based motion model for indoor positioning. Chen et al. [38] presented a graph-based method for indoor subarea positioning without requiring configuration. Yu et al. [39] developed a low-cost indoor navigation method by combining a Kalman filter and a particle filter in a cascade structure to reduce the computational burden. However, these methods consider only limited previous information rather than the complete estimations in the past, which may not be able to obtain the optimal estimation in some cases.

HMM is another popular probabilistic model that can make use of all previous information for the current estimation. VTrack [20] uses a HMM to estimate locations and travel time from a sequence of GPS and Wi-Fi measurements. Viel et al. [40]

presented a HMM-based map matching method for reconstructing device trajectories from cellular fingerprints. It considers spatial constraints imposed by roads when computing transition probability and then uses the Viterbi algorithm to generate the trajectories of interest. Ye et al. [41] developed a HMM to combine Wi-Fi RSSIs and accelerometer readings for indoor positioning, in which the transition probability is computed based on the movement distance estimated from the accelerometer readings. Algizawy et al. [42] proposed an adaptive HMM-based method that employs trip antenna coverage zones to learn trajectory patterns and infer corresponding roads for a given sequence of the trip antenna. Apart from the HMM, CRF is also often used for map matching. MapCraft [23] uses the CRF to integrate inertial and radio frequency (RF) sensor readings and constraints from a floor plan, landmarks, RF AP locations, and RF fingerprints to achieve robust and accurate indoor tracking. Mapel [43] utilizes the CRF to fuse geomagnetism with the pedometer to estimate the user location.

In this paper, the HMM-based map matching method is also used. However, different from existing HMM-based methods [20, 40, 42], which are used for reconstructing trajectories or outdoor positioning from GPS locations or cellular fingerprints, the HMM in this study is used for indoor positioning based on Wi-Fi fingerprints. While the method in [41] is developed for indoor positioning, it uses accelerometer readings to compute the transition probability. By contrast, the proposed method does not require additional hardware to compute the transition probability. Instead, the proposed method utilizes the path distance between two RPs to compute the transition probability. To increase the robustness of matching, both adjacent transitions and non-adjacent transitions between RPs are considered in the proposed method.

Table 1 summarizes typical related works, including spatial context involved, sensor used, key contributions, and limitations.

3. Method

In this section, the proposed PDMatching is described in detail. Firstly, the system architecture is introduced, which is followed by the problem formulation. Then, the processes of computing transition matrix, emission probability, and inference with a path table are described.

3.1. System architecture

The architecture of the proposed PDMatching is shown in Fig. 1, which includes an offline phase and an online phase. In the offline stage, the floor plan is divided into grids, of which

Table 1
Recent methods based on landmark and probability model.

Reference	Spatial context	Sensor	Key contribution	Limitation
[26]	Landmarks	GPS + inertial sensors	Seamless indoor and outdoor localization by combining GPS and PDR	Requirement for the occasional GPS signals
[27,11]	Landmarks	Wi-Fi + inertial sensors	Unsupervised indoor localization by combining landmarks with PDR	Dependence on the landmark density
[28]	Landmarks + Floor plan	Inertial sensors	Infrastructure-free indoor localization by using the particle filter to fuse PDR, landmarks, and maps	Requirement for accurate heading estimation
[29]	Landmarks	Wi-Fi + inertial sensors	Accurate indoor mapping via crowdsourcing	Dependence on landmark detection accuracy
[14]	Landmarks + maps	Camera + inertial sensors	Resource-efficient localization by using visual features and maps	Dependent on image retrieval accuracy
[15]	Floor plan	Inertial sensors + Wi-Fi	Zero-effort fingerprint collection via crowdsourcing by using a particle filter to integrate inertial sensor data with a floor plan	High computational cost
[31]	Floor plan	Inertial sensors + Wi-Fi/GPS	Fusing multiple signals with a floor plan to improve positioning accuracy	High computational cost
[33]	Floor plan	Inertial sensors + geomagnetism	Fast signal map creation by using SLAM and particle filtering	High computational cost and complex preprocessing steps
[39]	Floor plan	Inertial sensors + Wi-Fi	Accurate indoor positioning by fusing Wi-Fi, inertial sensor data, and Map via a cascaded Kalman/particle filter framework	High computational cost
[40]	Road network	Cellular	Trajectory generation from cellular fingerprints	Not applicable for indoor environments
[20]	Road network	GPS + Wi-Fi	Accurate traffic delay estimation by a HMM	Low accuracy for positioning
[42]	Road network	GPS + cellular	Learning road-level trajectory patterns from antenna coverages zones	Not applicable for indoor positioning
[36]	Voronoi graph	Inertial sensors + Wi-Fi	Efficient indoor positioning with graph-based sensor fusion	Low performance in open areas
[37]	Voronoi graph	Wi-Fi	Proposing a graph-based motion model	Assuming a uniform distribution in the graph's detail
[38]	Voronoi graph	Wi-Fi	Subarea indoor positioning without requiring configuration	Low positioning accuracy
[41]	N/A	Inertial sensors + Wi-Fi	Improving the Wi-Fi positioning by integrating fingerprinting with displacement ranging	Requiring accurate displacement estimation
[24]	Grid graph	N/A	Trajectory correction with a CRF	The result is dependent on the grid size
[23]	Floor plan	Inertial sensors + Wi-Fi	Efficient indoor tracking and trajectory reconstruction by using the CRF to fuse inertial sensor data, Wi-Fi measurements, and spatial constraints	Requirement for accurate estimation of step length and heading
[43]	Map	Inertial sensors + geomagnetism	A graphical model that fuses geomagnetic field and pedometer for calibration-free indoor positioning	Support adjacent transitions only

the centers are treated as the RPs. Then, a fingerprint database is constructed by the site survey, namely by standing at each RP to collect the RSSIs from visible APs. Based on the fingerprint database, the shortest distances between any two RPs are calculated using the Dijkstra algorithm. These RPs and their shortest distances to each other are then used to construct the path

table and the transition probability matrix. Each row in the path table represents the starting RP, ending RP, and the sequence of connected RPs from the starting RP to the ending RP. In the online stage, the RSSI query from the mobile device is used as the input of the proposed system. The similarities between the RSSI query and these RSSIs stored in the fingerprint database are

measured by the emission probabilities, which are transformed by the student's t-distribution from RSSI distances. Then, the transition probability matrix together with the emission probability matrix is used to infer the trajectory that best matches with observations (Wi-Fi measurements) by the Viterbi algorithm. The possible RPs are inserted in the adjacent RPs of the matched trajectory by comparing with those stored in the path table. Finally, the locations of the RP sequence are used as the output of the system.

3.2. Problem formulation

Let $S = \{s_1, s_2, \dots, s_N\}$ be a set of hidden states (namely RPs), and O be the set of observations (namely Wi-Fi RSSIs). The transition matrix is denoted by A , and its element is represented by $P(s_j|s_i)$, which represents the probability of state s_i transitioning to state s_j . The emission probability $P(o|s)$ is the probability of observing o at state s , which is described based on the distances of the online RSSI and the fingerprints. To obtain the location by map matching, the HMM model selects the most likely sequence of the hidden states $X = \{x_1, x_2, \dots, x_T\}$, where $x_i \in S$. The HMM model ensures to maximize the probability of the observations $O = \{o_1, o_2, \dots, o_T\}$ as follows:

$$P(O) = \sum_X P(O|X)P(X) \quad (1)$$

$$= \sum_X \pi_{s_i} \prod_{t=1}^{T-1} P(x_{t+1}|x_t)P(o_t|x_t).$$

where the initial state distribution π_{s_i} is defined as $1/N$, which means the initial location could be at any RP. After obtaining the sequence of hidden states X that correspond to given observations, the current location of the user can be inferred by looking up the path table.

3.3. Path distance-based transition probabilities

Due to the existence of obstacles, the path distance is utilized to replace the Euclidean distance to measure the transitions between RPs. Specifically, the Dijkstra algorithm is used to compute the shortest path distances between two RPs with considering spatial constraints of the indoor environment. Based on the shortest path distances, the transition matrix A is calculated using the student's t-distribution. There are several reasons why the student's t-distribution is used rather than the commonly-used normal distribution. First, as shown in Fig. 2, the probability of normal distribution drops quickly with the increase of the distance. As a result, the probability with a large distance is close to 0 and hence limits the transition to a local region. Second, the probability of normal distribution depends on the variance, which is difficult to estimate in a real scene. If the variance is set to a very small value, the probabilities between the RPs are nearly the same and could not well describe the transition. On the contrary, a large value of the variance will cause most transition probabilities being close to 0. Therefore, the long-tailed student's t-distribution with one-degree of freedom is used to compute the transition probability. The student's t-distribution relies only on the distance. Fig. 2 visualizes the comparison of the normal and student's t-distributions, the probability of the student's t-distribution is smaller than that of the normal distribution but is bigger when the distance is large. The student's t-distribution allows the transition between the RPs with larger distance.

In practice, since the user may move to any location within a small range, the same value is assigned to the probability whose path distance is below a distance threshold. Thus, the transition

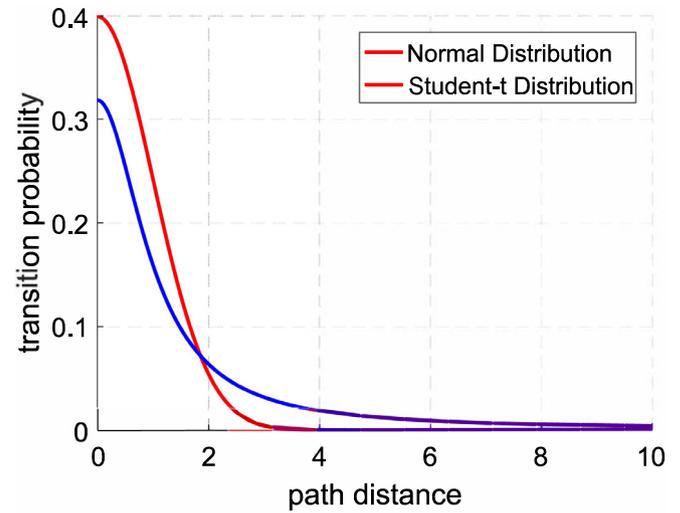


Fig. 2. Comparison of normal and student's t-distributions.

probability between RPs s_i and s_j is defined as the following piecewise function:

$$a_{s_i, s_j} = \begin{cases} (1 + \delta^2)^{-1}, & d_{path}(s_i, s_j) \leq \delta \\ (1 + d_{path}(s_i, s_j)^2)^{-1}, & d_{path}(s_i, s_j) > \delta \end{cases} \quad (2)$$

where δ is a threshold of path distance. $d_{path}(s_i, s_j)$ is the path distance between s_i and s_j . To ensure that the sum of the transition probabilities from the same RP equals to 1, the probabilities is transformed as follows:

$$p_{s_i, s_j} = \frac{a_{s_i, s_j}}{\sum_{s_j \in S} a_{s_i, s_j}} \quad (3)$$

where S is the set of the RPs. Then, by taking each RP as the starting point, the complete transition probability matrix A is constructed as follows:

$$A = \begin{bmatrix} p_{s_1, s_1} & p_{s_1, s_2} & \cdots & p_{s_1, s_N} \\ p_{s_2, s_1} & p_{s_2, s_2} & \cdots & p_{s_2, s_N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{s_N, s_1} & p_{s_N, s_2} & \cdots & p_{s_N, s_N} \end{bmatrix}. \quad (4)$$

3.4. Emission probabilities

The emission probability describes the probability of an observation o_t being generated from a state s_i . The student's t-distribution is also used to transform the Euclidean distance between the online RSSI and fingerprints as the emission probability, which is expressed as follows:

$$p(o_t | s_i) = (1 + \|rssi_o(t) - rssi_{s_i}\|^2)^{-1} \quad (5)$$

where $rssi_o(t)$ is the online RSSI treated as the observation at time t . $rssi_{s_i}$ is the RSSI fingerprint at the RP s_i . As the $rssi_{s_i}$ is collected offline, the value is fixed in the map matching process. For simplicity, $b_t(s_i)$ is used to replace $p(o_t | s_i)$.

3.5. Map matching

After calculating the transition and emission probabilities, the location result is inferred via the Viterbi algorithm [25]. The Viterbi algorithm is a typical dynamic programming method aiming to solve the inference of HMM. The algorithm is adopted to select a sequence of RPs by the maximizing the joint probability. Let $\delta_t(s_i)$ be the probability of the path ending at the RP s_i at time

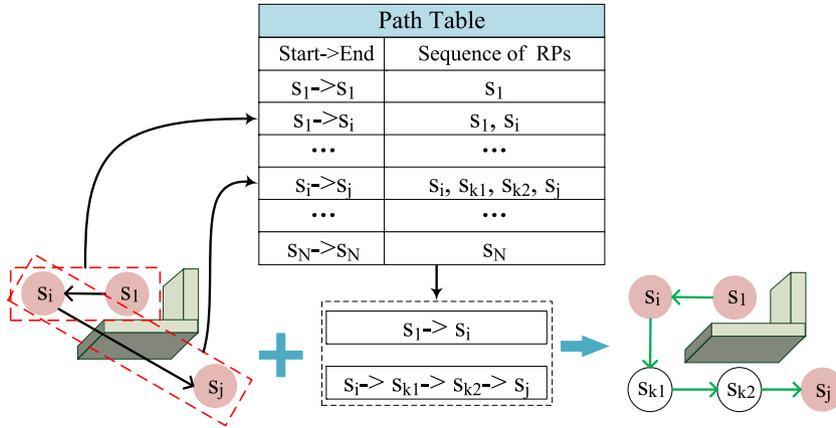


Fig. 3. Filling the gap by using path table. The white nodes are queried from the path table.

t . The initial $\delta_1(s_i)$ is set to $b_1(s_i)$. Thus, the $\delta_t(s_i)$ can be updated as follows:

$$\delta_t(s_i) = \max_{s \in S} [\delta_{t-1}(s) \cdot p_{s,s_i} \cdot b_t(s_i)]$$

$$\mathcal{T}(s_i, t) = \arg \max_{s \in S} [\delta_{t-1}(s) \cdot p_{s,s_i} \cdot b_t(s_i)] \quad (6)$$

where $b_t(s_i)$ is the emission probability at time t , p_{s,s_i} is the transition probability from state s to state s_i . $\mathcal{T}(s_i, t)$ is a table recording the selected state s of time $t - 1$ when the path ends at the RP s_i at time t . After $T - 1$ updates, δ_T and the table $\mathcal{T}(s_i, T)$ are obtained. Then the last hidden state is determined with the maximal value δ_T according to the following equation.

$$x_T = \arg \max_{s \in S} [\delta_T(s_i)]. \quad (7)$$

Then the hidden state x_{T-1} in the most likely sequence is selected by looking up table as follows:

$$x_{T-1} = \mathcal{T}(x_T, T). \quad (8)$$

All the hidden states in the most likely sequence $X = \{x_1, x_2, \dots, x_T\}$ are obtained by recursive computation in Eq. (8).

With the most likely sequence X , the matched RPs are selected. However, using the RPs in X as the result of map matching is not reasonable, since the transition between the time-adjacent RPs x_t and x_{t+1} does not consider the other possible RPs on the path and may lead to the inferred path crossing walls (as illustrated in Fig. 3). Let (s_1, s_i, s_j) be the inference result, they are divided into two pairs of time-adjacent nodes, denoted by (s_1, s_i) and (s_i, s_j) . These pairs are treated as the clues to look up the sequence of RPs from the path table. After looking up the table, two nodes s_{k1} and s_{k2} are inserted into the pair (s_i, s_j) to obtain the path $(s_i, s_{k1}, s_{k2}, s_j)$ that describes the transition from s_i to s_j . Then, the complete RP sequence $(s_1, s_i, s_{k1}, s_{k2}, s_j)$ is taken as the map matching result which considers the spatial information of the

indoor environment. The complete procedures of the proposed PDMatching method are given in Algorithm 1.

Algorithm 1: Map Matching with Path Table

Input: RSSI sequence $\{rss_1, rss_2, \dots, rss_T\}$, RPs in the floor plan $S = \{s_1, s_2, \dots, s_N\}$, fingerprint dataset F .

Output: RP Sequence X^*

- 1 Construct the transition probability matrix A according to the methods described in Section 3.3;
- 2 Compute the emission probabilities $\{b_1(s_1), b_1(s_2), \dots, b_1(s_N)\}$ according to the methods described in Section 3.4;
- 3 Initialize $\delta_1(s_i) = b_1(s_i)$, $X^* = \emptyset$;
- 4 **for** $t = 2, 3, \dots, T$ **do**
- 5 Compute the emission probabilities $\{b_t(s_1), b_t(s_2), \dots, b_t(s_N)\}$;
- 6 Compute $\delta_t(s_i)$ and update \mathcal{T} by Eq. (6);
- 7 Find the last matching RP x_T by Eq. (7);
- 8 **for** $t = T, T - 1, \dots, 2$ **do**
- 9 Find the matching RP x_{t-1} with \mathcal{T} by using Eq. (8);
- 10 **for** $t = 1, 2, \dots, T - 1$ **do**
- 11 Search the sequence $\{x_t, \dots, x_{t+1}\}$ in the path table using $\{x_t, x_{t+1}\}$;
- 12 $X^* = X^* \cup \{x_t, \dots, x_{t+1}\}$;
- 13 **return** X^* .

4. Evaluation

Extensive experiments are conducted to evaluate PDMatching in a real scene. Experimental data was collected on the second floor of a typical museum environment (as shown in Fig. 4) that has an area of 87×57 m². The experimental environment contains regions of open space and irregular shapes, which raise the complexity of the radio propagation. To build the fingerprint database, the floor plan is divided into grids with a size of 1 m and the centers of grids are considered as the RPs. In total, 543 RPs are obtained in the experimental environment. Fingerprints were collected by standing at each RP for a while. Two smartphones, namely one *Samsung note 4* and one *Google nexus 6*, were used in the experiments and their Wi-Fi sampling frequency are 2.5 Hz and 1 Hz, respectively. In total, 8145 fingerprint records were obtained from 384 heard APs and stored in the fingerprint database. The value of receiving no signal is set to -100 dBm.

Six participants were recruited for conducting the experiments. They were asked to walk along 18 planned paths and

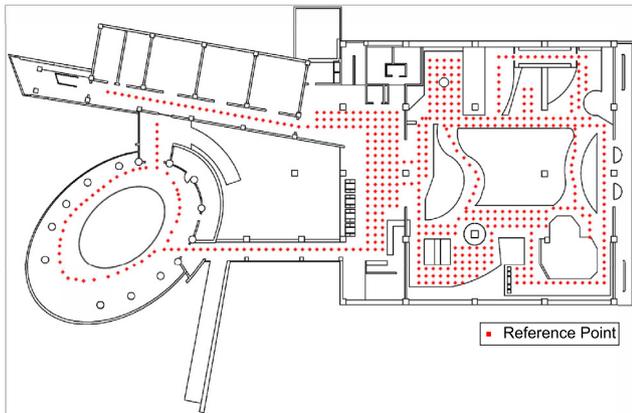


Fig. 4. The floor plan.

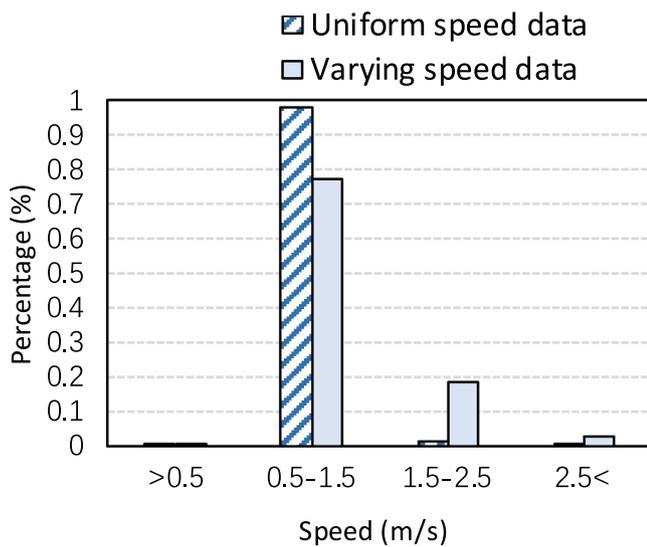


Fig. 5. Speed distribution of uniform speed data and varying speed data.

Table 2
Dataset Description.

Dataset	Description
Number of RPs	543
Number of fingerprints	8145
Number of trajectories	180
Wi-Fi observations	23760
Total length of trajectory	~ 41 km
Collection time of trajectory	~ 10 h

repeat each path for 10 times for collecting Wi-Fi RSSI observations and inertial sensor readings. In total, 180 trajectories are obtained with a total length of about 41 km. During the 10 times of walking on each path, the participants walked at a uniform speed for 6 times and at a varying speed for 4 times. Fig. 5 illustrates the statistical information of the walking speed in the two speed modes. The details of the data are shown in Table 2.

To evaluate its superiority, the PDMatching is compared with several state-of-the-art methods, including VTrack [20], MapCraft [23], XINS [36], and SPBI-PF [34]. As the map matching methods aim to improve the positioning method by spatial information, the positioning results of a representative Wi-Fi fingerprinting positioning system called RADAR [44] are taken as the baseline to compare the performance of different map matching methods. VTrack and MapCraft are typical Bayesian network-based methods, while XINS and SPBI-PF are common particle filter-based

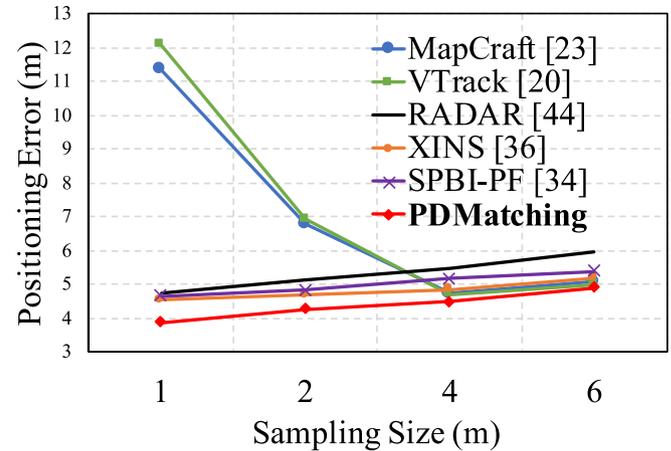


Fig. 6. Results under different sampling sizes.

methods. VTrack uses a HMM model to describe the positioning trajectory. Although this method is designed for the outdoor map matching, it can be easily utilized in the indoor environment by treating the RPs as hidden states. The transition in the VTrack is obtained by using adjacent RPs and the transition probability is determined manually. Different from VTrack, MapCraft uses a CRF to fuse multiple cues, such as landmarks, Wi-Fi RSSI, and inertial sensor data, to infer the result of the map matching. In order to make a fair comparison, PDMatching, VTrack, and MapCraft use only the Wi-Fi RSSI observations as the cue to obtain the result of map matching. XINS and SPBI-PF feed inertial sensor data and Wi-Fi RSSI to the state model for propagating particles and use spatial constraints to refine the positioning results. The number of particles for the XINS and SPBI-PF is set to 5000.

4.1. Impact of RPs sampling size

To evaluate the impact of the interval size of RPs on the accuracy of map matching methods, their performance is analyzed under different sampling sizes, namely 1, 2, 4, 6 m. Fig. 6 shows the positioning error of different methods under different sampling sizes of RPs. It can be seen that the proposed PDMatching is robust to the sampling size and performs the best among the methods compared since the PDMatching considers both adjacent and non-adjacent transitions between RPs. VTrack and MapCraft are very sensitive to the sampling size of RPs. The positioning error achieved by the VTrack and MapCraft is large when the sampling size is small, which decreases as the sampling size increases up to 4 m. This is because the VTrack and MapCraft allow only the adjacent transition, and hence the length of the matching sequence is greatly constrained by the sampling size. Compared to VTrack and MapCraft, XINS and SPBI-PF are not significantly affected by the sampling size as they use both inertial sensor data and Wi-Fi RSSI for location estimation. However, XINS and SPBI-PF require a large number of particles to achieve satisfactory accuracy, which may lead to a huge computational cost, which will be discussed in Section 4.6.

Fig. 7 shows the cumulative distribution function (CDF) of the positioning errors of different methods under varying sampling sizes. Most methods witness the best positioning accuracy when the sampling size is 1 m except the MapCraft and VTrack that achieve the best accuracy when the sampling size is 4 m. Specifically, the PDMatching performs the best with an accuracy of 80% for the positioning error below 5 m (for the sampling size of 1 m), which is followed by the SPBI-PF and the XINS (with the accuracy of around 72% and 71%, respectively). The RADAR stands in the

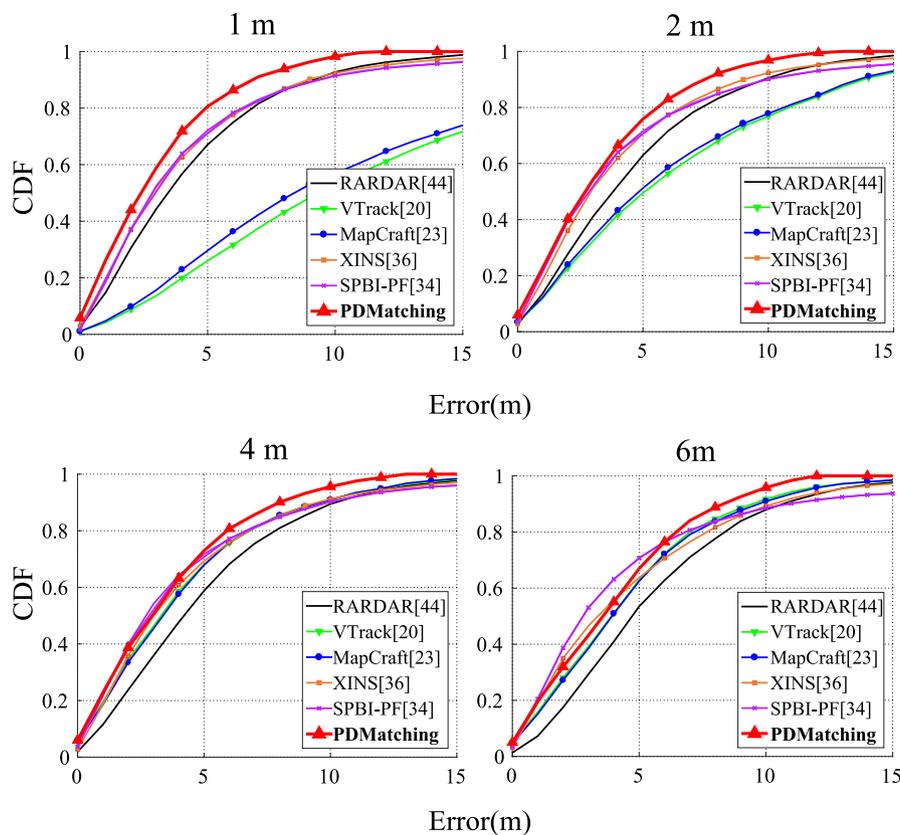


Fig. 7. CDF of different methods under different sampling sizes.

middle with an accuracy of around 68%, which is much higher than the VTrack (around 27%) and MapCraft (around 30%). However, the positioning accuracy of the VTrack and MapCraft increases dramatically to around 68% (for the positioning error below 5 m) when the sampling size is 4 m. Overall, the performance curves of these methods move closer to each other when increasing the sampling size from 1 m to 4 m. When the sampling size increases to 6 m, the positioning accuracy below the error 5 m for the SPBI-PF (around 72%) is higher than that of the PDMatching (around 68%). This might be explained that the SPBI-PF uses the extra information (inertial sensor data) to assist positioning, and is not affected by the sampling size of RPs. However, the PDMatching is still superior to other methods at the sampling size of 6 m.

4.2. Impact of walking speed

The walking speed of users affects the performance of map matching as the transition distance for one step will change at different walking speeds. A set of experiments were conducted to evaluate the robustness of the proposed PDMatching under different walking speeds. A sampling size of 1 m is used in the experimental analysis, which is a commonly-used sampling size as can be seen in [23,24,43]. From Fig. 8, it can be concluded that the PDMatching is more robust to varying walking speeds than other methods. In both speed modes, PDMatching outperforms the other methods in terms of positioning performance. The positioning errors for all the methods at the uniform speed mode are lower than at the varying speed mode. Among these methods compared, VTrack and MapCraft have a large positioning error as they allow only adjacent transitions, which result in the matched results to be constrained by the sampling sizes. The RADAR, XINS, and SPBI-PF perform better than the MapCraft and VTrack and are less affected by the varying walking speeds since the XINS and SPBI-PF utilize both inertial sensor data and Wi-Fi measurements while the RADAR does not use previous information for matching.

4.3. Impact of Wi-Fi scanning frequency

The Wi-Fi scanning frequency of smartphones also affects the transition distance for one step because it leads to a change in transition probabilities and sequence inference. The data collected at the uniform speed mode is used for map matching under the sampling size 4 m where the VTrack and MapCraft can achieve their best positioning accuracy. The Wi-Fi scanning frequencies considered range from 0.4 Hz to 0.1 Hz.

As shown in Fig. 9, the results indicate that the PDMatching is robust to the change of Wi-Fi scanning frequency with a slight increase in the positioning error from 3.95 m to 4.59 m as the scanning frequencies decrease from 0.4 Hz to 0.1 Hz, while the performance of the MapCraft and VTrack degrades quickly. This is because the transition matrices of the MapCraft and VTrack are constructed using only the adjacent RPs whose sampling size limits the transition distance without considering the Wi-Fi scanning frequency. Due to the change of the scanning frequencies, the real transition distance might be beyond the distance limited by the sampling size and therefore lead to a large positioning error for these methods that do not allow the non-adjacent transition. Since the PDMatching supports the non-adjacent transition between RPs, it is possible to correctly compute the transition probability and achieve a similar performance under various scanning frequencies. It can also be seen that the XINS and SPBI-PF are insensitive to the Wi-Fi scanning frequency since they use inertial sensor readings to estimate the transition distance that are independent of Wi-Fi measurements. However, the extra resource for updating a great number of particles by the sensor readings results in a larger time delay and energy consumption.

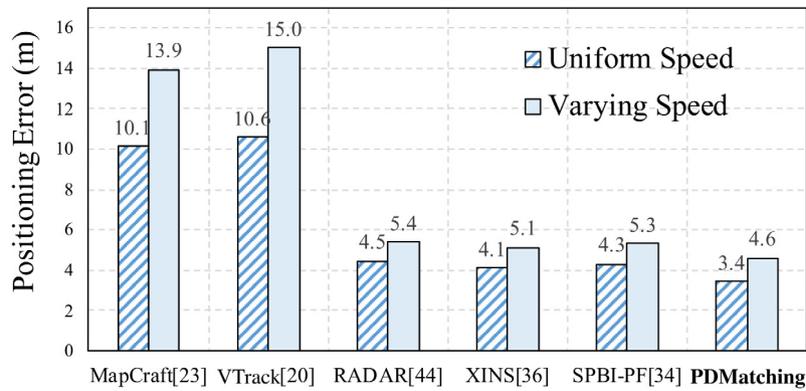


Fig. 8. Performance under different speeds.

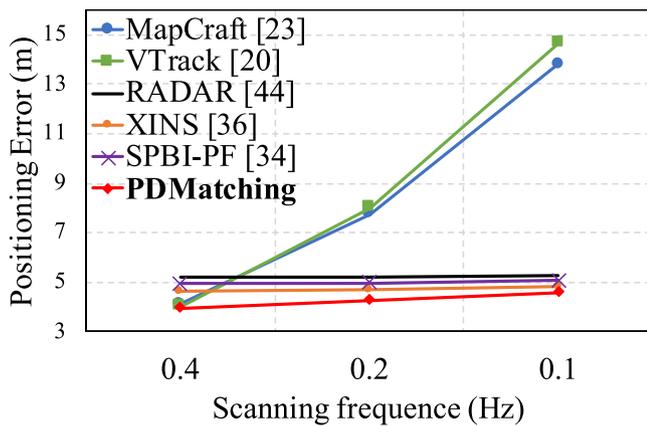


Fig. 9. The impact of Wi-Fi scanning frequency.

4.4. Comparison of different transition distance

The distance between RPs is used to describe the transition probability. A proper distance is required to correctly calculate the transition probability that will determine the performance of map matching. To show the advantage of using path distance in

map matching, the path distance-based map matching is compared with those matching methods using the commonly-used Euclidean distance, Manhattan distance, and constant distance (a fixed value).

Fig. 10 gives the CDF of the positioning errors based on the four distance metrics. The positioning results of the RADAR are used as the baseline to compare the results of the proposed PDMatching using four distance metrics. From Fig. 10, one can see that the path distance metric is superior to other distance metrics. This is because the path distance uses not only the coordinates, which are also used in the Euclidean and Manhattan distances, but also the spatial constraints imposed by obstacles such as walls, which is not considered in other distance metrics.

Fig. 11 shows the matching results of the proposed PDMatching method using four transition distances. The path distance leads to a better matching result than other distance metrics as the path distance considers spatial constraints in the environments. The constant transition distance witnesses the worst matching results because it includes many adjacent matchings of crossing walls. As the transition distance is fixed, the elements of the transition probability matrix are the same, and therefore its matching results are similar to that of the pointwise positioning method such as the Nearest Neighbor (NN) matching. The matching result of using the Euclidean distance is better than that using the constant distance, but there are still some jump matchings as the Euclidean distance does not consider the obstacles and some

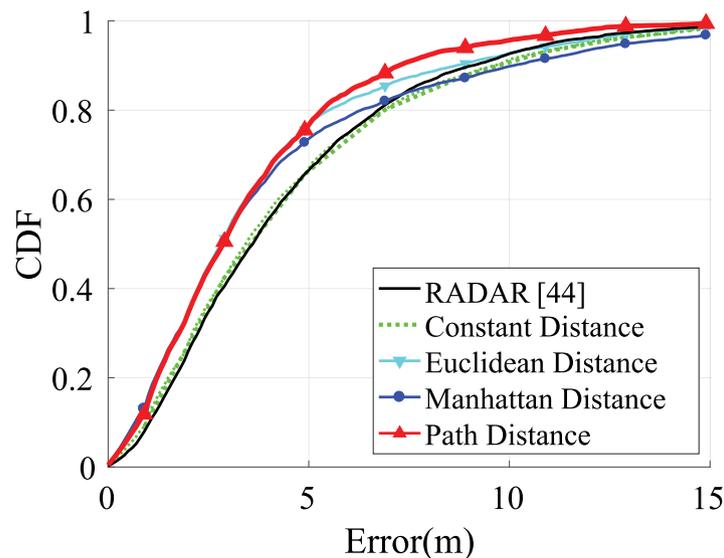


Fig. 10. The CDF of the positioning error of using different distance metrics.



Fig. 11. The trajectory estimated by the PDMatching using different distances. (a) Ground-truth. (b) Constant distance and NN trajectory. (c) Euclidean distance trajectory. (d) Manhattan distance trajectory. (e) Path distance trajectory.

matchings may cross the walls. The Manhattan distance performs worse than the Euclidean distance. The Manhattan distance is popular for map matching in outdoor environments where the walking distance can be measured accurately with the blocks of regular shapes (e.g., rectangles), but it does not work well since indoor environments contain various regions of irregular shapes [10].

4.5. Impact of the distance threshold δ

The distance threshold δ is used to determine the radius of the region, in which the transition probabilities are considered as the same. For the transitions that are out of δ , the assigned probability value declines as the distance increases. Therefore, δ has an impact on the transition probability matrix and on the performance of map matching.

Fig. 12 shows the impact of δ on the positioning accuracy. The threshold distance considered ranges from 1 m to 9 m and different sampling sizes (1, 2, 4, 6 m) are considered. It is obvious that a small value of δ results in a large error of map matching for all the sampling sizes. As a result, the performance with a small δ is not good, since the small δ favors transitions in a small region. As the increase of the δ , the positioning error of map matching decreases and achieves the best when δ is in the range (4, 6). It could be explained that the most possible transition distance for

one step is about 5 m in the experiment. For the sampling size of 6 m, the transition distance needs to be larger than 6 m to ensure the transition to other RPs rather than itself. When δ is larger than 6 m, the student's t-distribution becomes flatter, causing that the transition probabilities are similar. In that case, the transition probability matrix contains little spatial information and is not beneficial to the map matching. Therefore, the results with a large value of δ are similar to the pointwise positioning methods.

4.6. Computational cost

The complexity of the PDMatching mainly comes from the Dijkstra algorithm and the Viterbi inference. While the Dijkstra algorithm is relatively computationally expensive, it can be conducted on a PC or server in the offline phase, and hence it does not affect the efficiency of the online location inference. Thus, the main computational cost is from the Viterbi algorithm. Its computational complexity is $O(N^2T)$, where N is the number of RPs and T is the length of the trajectory. The complexity of the proposed method is similar to the traditional HMM-based matching method since the transition matrices used in both methods have the same size and both methods use the Viterbi for inference.

Fig. 13 indicates the computational cost of different methods under different sampling sizes. The results show that the computational costs of the PDMatching, VTrack, and MapCraft are

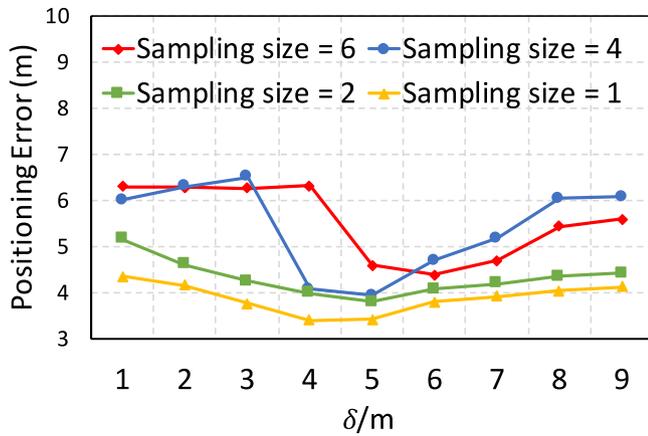


Fig. 12. Impact of the distance threshold δ .

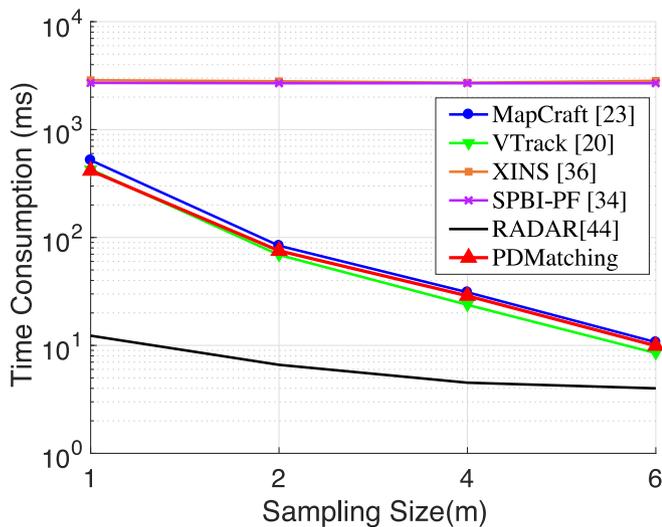


Fig. 13. Computational cost of different methods under varying sampling sizes.

heavily impacted by the RP sampling size and decrease quickly as the sampling size increases. The computational cost of the RADAR is less influenced by sampling sizes, and is much lower than the costs of other methods. This is because the RADAR is used to provide the initial location estimates that are taken as the input of other methods. Therefore, the computational costs of other methods contain the time consumed by the RADAR. The particle filtering-based methods (the XINS and SPBI-PF) have fixed computational costs under varying sampling sizes as they do not require information about RPs. Overall, the HMM-based methods have lower computational costs than particle filtering-based methods. The three HMM-based methods (PDMatching, VTrack, and MapCraft) have similar computational cost and could report the location result within 1 s, which is usually sufficient for most indoor positioning applications.

5. Discussion

The proposed PDMatching is based on a commonly-used HMM model, but it outperforms the state-of-the-art methods in terms of positioning accuracy (measured by mean positioning error) at different sampling sizes of RPs, walking speed modes, and Wi-Fi scanning frequencies. Specifically, the PDMatching performs the best among different methods with a mean positioning error of about 3.9 m at the sampling size 1 m and a mean positioning

error of 4.9 m at the sampling size of 6 m. By contrast, MapCraft and VTrack have a mean positioning error of 11.3 m and 12 m at the sampling size of 1 m, and a mean positioning error of 5 m for both at the sampling size of 6 m. The mean positioning errors for RADAR, XINS, and SPBI-PF stay in the middle, ranging from 4.5 m to 6 m. While the sampling size of 1 m results in the best positioning accuracy, a small sampling size means more RPs are required to be collected, which needs more effort on the site survey from the human and will lead to a higher computational cost (e.g., hundreds of milliseconds) in the location inference. Therefore, there is a trade-off between the positioning accuracy and the computational cost. Besides, the PDMatching outperforms other methods at varying speed modes, achieving a mean positioning error of 3.4 m at the uniform speed mode and 4.6 m at the varying speed mode. The mean errors of the PDMatching at both modes are lower than XINS (4.1 m and 5.1 m, respectively), SPBI-PF (4.3 m and 5.3 m, respectively), RADAR (4.5 m and 5.4 m, respectively), MapCraft (10.1 m and 13.9 m, respectively), and VTrack (10.6 m and 15 m respectively). In addition, the PDMatching is robust to the scanning frequency of Wi-Fi. It achieves a mean positioning error of 4 m and 4.6 m at the scanning frequencies of 0.4 Hz and 0.1 Hz, respectively. By contrast, the mean positioning errors for MapCraft and VTrack increase from 4.1 m and 4 m to 13.8 m and 14.7 m when the scanning frequency changes from 0.4 Hz to 0.1 Hz. RADAR, XINS, and SPBI-PF are less affected by the different scanning frequencies, and have a mean positioning error between 4.6 m and 5.2 m. The reason why the PDMatching performs better than other methods in different cases can be partially attributed to the use of both adjacent and non-adjacent transitions.

Path distance can also be beneficial for the PDMatching to obtain better positioning accuracy. In Fig. 10, we compare the performance of PDMatching using different distance metrics. It shows that the path distance can result in the best positioning accuracy at about 97% (for positioning error below 10 m), which is higher than the Euclidean distance (about 93%), constant distance (about 92%), and Manhattan distance (about 90%). This is attributed to that the path distance can take into account spatial constraints and eliminate invalid estimates, so as to improve the positioning accuracy.

In the following, the threats to the validity [45] of the proposed PDMatching are discussed. As the proposed method is evaluated by comparing with the state-of-the-art methods over the same dataset and using the same preprocessing techniques, the effectiveness of the proposed method can be justified with the best performance at varying cases as described above. However, the limited number of the participants and the RSSI variation in indoor environments may impose some threats on internal validity if one evaluates the method without comparison with other methods. As for the external validity, the proposed method is evaluated on the data collected in the experimental environment containing regular and irregular regions (e.g., such as open space, office room) and other common structures. This makes it practical to be applied in other environments such as office buildings and shopping malls. In other words, the proposed method can be generalized to other environments and its external validity is guaranteed. As for the construct validity, it measures the degree to which a test supports the claim. The main claim of this study is that the localization accuracy can be improved by using the path distance and considering both adjacent and non-adjacent transitions between RPs. This has been justified by experiments and analysis at different cases. As for the statistical validity, the collected trajectories have a total length of 41 km, which is much longer than the experimental trajectories in existing relevant studies. While the six participants involved may not be diverse enough in quantitative research, it is sufficient to evaluate the effectiveness of indoor positioning methods.

6. Conclusion

In this paper, a novel HMM-based map matching method (called PDMatching), which considers both adjacent and non-adjacent transitions, is proposed to address the problem of being sensitive to the sampling size of RPs in conventional map matching methods. Experimental results show that the proposed method achieves best positioning accuracy with a mean positioning error of about 4 m, which outperforms the state-of-the-art methods (MapCraft, VTrack, XINS, SPBI-PF, and RADAR). In terms of the accuracy of the positioning error below 5 m, the proposed PDMatching achieves an accuracy of about 80%, which is more than 8% higher than the other methods. This can be partially attributed to the use of both adjacent and non-adjacent transitions. Also, using the path distance to compute the transition matrix can result in a better positioning accuracy than using the other distance metrics (including Euclidean distance and Manhattan distance) with an improvement of more than 4% (the ratio of positioning error below 10 m).

In future work, we will investigate more efficient inference methods to further reduce the computational cost of the proposed method. Also, we will apply the proposed PDMatching method in a large shopping mall to provide practical services for customers.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (No. 2016YFB0502200) and the Fundamental Research Funds for National University, China University of Geosciences (Wuhan) (No. G1323511822).

References

- [1] S. Li, Z. Qin, H. Song, C. Si, B. Sun, X. Yang, R. Zhang, A lightweight and aggregated system for indoor/outdoor detection using smart devices, *Future Gener. Comput. Syst.* (2017).
- [2] E. Hu, Z. Deng, M. Hu, L. Yin, W. Liu, Cooperative indoor positioning with factor graph based on fm for wireless sensor network, *Future Gener. Comput. Syst.* 89 (2018) 126–136.
- [3] Y. Zhuang, Z. Syed, Y. Li, N. El-Sheimy, Evaluation of two WiFi positioning systems based on autonomous crowdsourcing of handheld devices for indoor navigation, *IEEE Trans. Mob. Comput.* 15 (8) (2015) 1982–1995.
- [4] X. Hu, J. Shang, F. Gu, Q. Han, Improving Wi-Fi indoor positioning via AP sets similarity and semi-supervised affinity propagation clustering, *Int. J. Distrib. Sens. Netw.* 11 (1) (2015) 109642.
- [5] N. Tadayon, M.T. Rahman, S. Han, S. Valaee, W. Yu, Decimeter ranging with channel state information, *IEEE Trans. Wirel. Commun.* 18 (7) (2019) 3453–3468.
- [6] Y. Ma, G. Zhou, S. Wang, WiFi sensing with channel state information: A survey, *ACM Comput. Surv.* 52 (3) (2019) 46.
- [7] S. He, S.-H. Chan, Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons, *IEEE Commun. Surv. Tutor.* 18 (1) (2015) 466–490.
- [8] F. Orujov, R. Maskeliūnas, R. Damaševičius, W. Wei, Y. Li, Smartphone based intelligent indoor positioning using fuzzy logic, *Future Gener. Comput. Syst.* 89 (2018) 335–348.
- [9] Y. Li, Z. He, Z. Gao, Y. Zhuang, C. Shi, N. El-Sheimy, Toward robust crowdsourcing-based localization: A fingerprinting accuracy indicator enhanced wireless/magnetic/inertial integration approach, *IEEE Internet Things J.* 6 (2) (2018) 3585–3600.
- [10] F. Gu, X. Hu, M. Ramezani, D. Acharya, K. Khoshelham, S. Valaee, J. Shang, Indoor localization improved by spatial context - A survey, *ACM Comput. Surv.* 52 (2019) 64:1–35.
- [11] H. Abdelnasser, R. Mohamed, A. Elgohary, M.F. Alzantot, H. Wang, S. Sen, R.R. Choudhury, M. Youssef, SemanticSLAM: Using environment landmarks for unsupervised indoor localization, *IEEE Trans. Mob. Comput.* 15 (7) (2016) 1770–1782.

- [12] F. Gu, K. Khoshelham, J. Shang, F. Yu, Sensory landmarks for indoor localization, in: *Ubiquitous Positioning, Indoor Navigation and Location Based Services, UPINLBS, 2016 Fourth International Conference on, IEEE, 2016*, pp. 201–206.
- [13] V. Magnago, L. Palopoli, R. Passerone, D. Fontanelli, D. Macii, Effective landmark placement for robot indoor localization with position uncertainty constraints, *IEEE Trans. Instrum. Meas.* (2019) 1–13.
- [14] Q. Niu, M. Li, S. He, C. Gao, S.-H. Gary Chan, X. Luo, Resource-efficient and automated image-based indoor localization, *ACM Trans. Sensor Netw.* 15 (2) (2019) 19.
- [15] A. Rai, K.K. Chintalapudi, V.N. Padmanabhan, R. Sen, Zee: Zero-effort crowdsourcing for indoor localization, in: *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, ACM, 2012*, pp. 293–304.
- [16] O. Woodman, R. Harle, Pedestrian localisation for indoor environments, in: *Proceedings of the 10th International Conference on Ubiquitous Computing, ACM, 2008*, pp. 114–123.
- [17] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, F. Zhao, A reliable and accurate indoor localization method using phone inertial sensors, in: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing, ACM, 2012*, pp. 421–430.
- [18] T.E. Abrudan, Z. Xiao, A. Markham, N. Trigoni, Distortion rejecting magneto-inductive 3-D localization (MagLoc), *IEEE J. Sel. Areas Commun.* 33 (2015) 2404–2417.
- [19] W. Li, B. Wang, L.T. Yang, M. Zhou, RMapTAFa: Radio map construction based on trajectory adjustment and fingerprint amendment, *IEEE Access* 7 (2019) 14488–14500.
- [20] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, J. Eriksson, Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones, in: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, ACM, 2009*, pp. 85–98.
- [21] J. Seitz, J. Jahn, J.G. Boronat, T. Vaupel, S. Meyer, J. Thielecke, A hidden markov model for urban navigation based on fingerprinting and pedestrian dead reckoning, in: *Information Fusion, FUSION, 2010 13th Conference on, IEEE, 2010*, pp. 1–8.
- [22] Y. Wu, P. Chen, F. Gu, X. Zheng, J. Shang, *HTrack*: An efficient heading-aided map matching for indoor localization and tracking, *IEEE Sens. J.* 19 (8) (2019) 3100–3110.
- [23] Z. Xiao, H. Wen, A. Markham, N. Trigoni, Indoor tracking using undirected graphical models, *IEEE Trans. Mob. Comput.* 14 (11) (2015) 2286–2301.
- [24] S. Bataineh, A. Bahillo, L.E. Diez, E. Onieva, I. Bataineh, Conditional random field-based offline map matching for indoor environments, *Sensors* 16 (8) (2016) 1302.
- [25] A.J. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, *IEEE Trans. Inform. Theory* 13 (2) (1967) 260–269.
- [26] D. Gusenbauer, C. Isert, J. Krösche, Self-contained indoor positioning on off-the-shelf mobile devices, in: *Indoor Positioning and Indoor Navigation, IPIN, 2010 International Conference on, IEEE, 2010*, pp. 1–9.
- [27] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, R.R. Choudhury, 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.
- [28] J. Shang, F. Gu, X. Hu, A. Kealy, Apfiloc: An infrastructure-free indoor localization method fusing smartphone inertial sensors, landmarks and map information, *Sensors* 15 (10) (2015) 27251–27272.
- [29] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, L. Chen, ALIMC: Activity landmark-based indoor mapping via crowdsourcing, *IEEE Trans. Intell. Transp. Syst.* 16 (5) (2015) 2774–2785.
- [30] S. He, S.G. Chan, Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons, *IEEE Commun. Surv. Tutor.* 18 (1) (2016) 466–490.
- [31] Y. Gao, Q. Yang, G. Li, E.Y. Chang, D. Wang, C. Wang, H. Qu, P. Dong, F. Zhang, Xins: The anatomy of an indoor positioning and navigation architecture, in: *Proceedings of the 1st International Workshop on Mobile Location-Based Service, ACM, 2011*, pp. 41–50.
- [32] Z. Tian, Y.-L. Wei, W.-N. Chang, X. Xiong, C. Zheng, H.-M. Tsai, K.C.-J. Lin, X. Zhou, Augmenting indoor inertial tracking with polarized light, in: *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, ACM, 2018*, pp. 362–375.
- [33] C. Gao, R. Harle, Semi-automated signal surveying using smartphones and floorplans, *IEEE Trans. Mob. Comput.* 17 (8) (2018) 1952–1965.
- [34] H. Xia, J. Zuo, S. Liu, Y. Qiao, Indoor localization on smartphones using built-in sensors and map constraints, *IEEE Trans. Instrum. Meas.* (99) (2018) 1–10.
- [35] L. Liao, D. Fox, J. Hightower, H. Kautz, D. Schulz, Voronoi tracking location estimation using sparse and noisy sensor data, in: *Intelligent Robots and Systems, 2003. IROS 2003. Proceedings. 2003 IEEE/RSJ International Conference on, volume 1, IEEE, 2003*, pp. 723–728.

- [36] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, E. Steinbach, Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2014, pp. 147–158.
- [37] H. Nurminen, M. Koivisto, S. Ali-Löytty, R. Piché, Motion model for positioning with graph-based indoor map, in: 2014 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2014, Busan, South Korea, October 27–30, 2014, 2014, pp. 646–655.
- [38] Y. Chen, M. Guo, J. Shen, J. Cao, Graphloc: a graph-based method for indoor subarea localization with zero-configuration, *Pers. Ubiquitous Comput.* 21 (3) (2017) 489–505.
- [39] C. Yu, H. Lan, F. Gu, F. Yu, N. El-Sheimy, A Map/INS/Wi-Fi integrated system for indoor location-based service applications, *Sensors* 17 (6) (2017) 1272.
- [40] A. Viel, D. Gubiani, P. Gallo, A. Montanari, A.D. Torre, F. Pittino, C. Marshall, Map matching with sparse cellular fingerprint observations, in: 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services, UPINLBS, 2018, pp. 1–10.
- [41] A. Ye, J. Shao, L. Xu, J. Chen, J. Xiong, Local HMM for indoor positioning based on fingerprinting and displacement ranging, *IET Commun.* 12 (10) (2018) 1163–1170.
- [42] E. Algizawy, T. Ogawa, A. El-Mahdy, Real-time large-scale map matching using mobile phone data, *ACM Trans. Knowl. Discov. Data* 11 (4) (2017) 52:1–52:38.
- [43] H. Wu, S. He, S.H.G. Chan, A graphical model approach for efficient geomagnetism-pedometer indoor localization, in: IEEE International Conference on Mobile Ad Hoc and Sensor Systems, 2017, pp. 371–379.
- [44] P. Bahl, V.N. Padmanabhan, RADAR: An in-building RF-based user location and tracking system, in: INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, volume 2, IEEE, 2000, pp. 775–784.
- [45] D.T. Campbell, J.C. Stanley, *Experimental and Quasi-Experimental Designs for Research*, Ravenio Books, 2015.



Pan Chen is an assistant professor in the Faculty of Information Engineering at the China University of Geosciences at Wuhan. He received the Ph.D. degree in Huazhong University of Science and Technology (HUST), Wuhan, China, in 2016. His current research areas include indoor localization and pattern recognition. He is a member of China Computer Federation. E-mail: chenpan@cug.edu.cn.



Xiaoping Zheng received the B.S. degree in 2016 from Information Engineering at China University of Geosciences, Wuhan, China, where he is currently pursuing his M.S. degree in Software Engineering. He also works for the Engineering Research Center for Geographic Information System (China). His research interests include indoor localization, mobile sensing, and machine learning. E-mail: zheng.xiaoping@cug.edu.cn.



Fuqiang Gu is currently a postdoctoral researcher in the Department of Mechanical and Industrial Engineering at the University of Toronto. He obtained his Ph.D degree from the University of Melbourne in 2018. His research interests include indoor localization and navigation, location-based services, activity recognition, context sensing, and machine learning. E-mail: fuqiang.gu@mail.utoronto.ca.



Jianga Shang is currently a professor with the School of Geography and Information Engineering, China University of Geosciences, Wuhan. He received the Ph.D. degree in computer system architecture from the Huazhong University of Science and Technology. He is also the director of Ubiquitous Location-aware Computing Lab in the National Engineering Research Center for Geographic Information System. Dr Shang is a committee member of ISPRS WG IV/5 on indoor/outdoor seamless modeling, LBS, and mobility. He is also a member of the IEEE, ACM, China Computer Federation (CCF), and GNSS & LBS Association of China (GLAC). His research interests include indoor location-awareness, mobile and context-aware computing, geospatial information systems, and software engineering. E-mail: jgshang@cug.edu.cn.