Unsupervised Localization by Learning Transition Model[*]

XUEHAN YE, DEKE MOE, Information School, Renmin University of China
SHUO HUANG, DEKE MOE, Information School, Renmin University of China
YONGCAI WANG[†], DEKE MOE, Information School, Renmin University of China
WENPING CHEN, DEKE MOE, Information School, Renmin University of China
DEYING LI, DEKE MOE, Information School, Renmin University of China

Nowadays, it becomes very convenient to collect synchronized WiFi *received signal strength and inertial measurement (RSS+IMU)* sequences by mobile devices, which enables the promising solution to conduct unsupervised indoor localization without the pain of radio-map calibration. To relax the needs of floor-map information or trajectory knowledge, this paper proposes to learn a *transitional model (TM)*, which segments the massive unlabeled sequences to train a model that captures the expected relationship between $\{z_{t-1}, z_t\}$ and u_{t-1}, where z_{t-1}, z_t are two consecutive signal states at t and $t-1$, and u_{t-1} is the one step motion calculated from inertial data. We present both a *transitional model in signal space (TMS)* and a *transitional model to predict motion from signal change (TMM)* to represent the relationship in different ways. In particular, from the massive sequences, both the signal states and the one step motion are smoothed from the nearest neighbours, so that the transition model learns the expected relative signal state change triggered by the smoothed one step motion. Its distinctive features are that (1) no external floor-map or trajectory knowledge is needed; (2) it can be continuously on-line refined as unlabeled sequences are incrementally collected. KALMAN filter based on-line mobile user location tracking methods are given for both models.Experiments show that the transition model based localization method provides comparable accuracy with the manually fingerprint calibration methods.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Computer Human Interaction*; Indoor Localization; • **Networks** → Wireless Networks.

Additional Key Words and Phrases: Unsupervised Learning, WiFi, Localization, Inertial Navigation, IMU

ACM Reference Format:
Xuehan Ye, Shuo Huang, Yongcai Wang, Wenping Chen, and Deying Li. 2019. Unsupervised Localization by Learning Transition Model. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 2, Article 65 (June 2019), 23 pages. https://doi.org/10.1145/3328936

[*] This work was supported in part by the National Natural Science Foundation of China Grant No. 61672524, 11671400.
[†] Corresponding author: Yongcai Wang, ycw@ruc.edu.cn

Authors' addresses: Xuehan Ye, DEKE MOE, Information School, Renmin University of China, Renmin University of China, Beijing, P.R. China, 100872, xuehanye@ruc.edu.cn; Shuo Huang, DEKE MOE, Information School, Renmin University of China, Renmin University of China, Beijing, P.R. China, 100872, huangshuo@ruc.edu.cn; Yongcai Wang, DEKE MOE, Information School, Renmin University of China, Renmin University of China, Beijing, P.R. China, 100872, ycw@ruc.edu.cn; Wenping Chen, DEKE MOE, Information School, Renmin University of China, Renmin University of China, Beijing, P.R. China, 100872, chenwenping@ruc.edu.cn; Deying Li, DEKE MOE, Information School, Renmin University of China, Renmin University of China, Beijing, P.R. China, 100872, deyingli@ruc.edu.cn.

2474-9567/2019/6-ART65 $15.00
https://doi.org/10.1145/3328936

1 INTRODUCTION

Indoor localization using WiFi-signal signature has attracted great attention in the last decade[3][26][47][43][15][35][14][36]. Its key technology is to build a model that relates the signal signatures in the signal space to the locations in the physical space, which is generally called *fingerprints (FPs)*[3][26][43]. One of the critical challenges of the FP-based method is that it is highly laborious and time consuming to collect the signal fingerprints and to calibrate the location labels for the signal fingerprints in large scale applications[38][14][5][45]. Another challenge is that the FP radio-map may be impacted by environment changes, which makes its on-line updating a critical requirement[46][2].

Tremendous efforts have been continuously drawn to solve these two challenges. The traditional supervised radio-map learning methods[3][26] have been improved to semi-supervised learning[27][31] and unsupervised learning[43][25] to reduce the radio-map calibration. As WiFi module and *inertial measurement unit (IMU)* are widely equipped in today's mobile devices, such as mobile phones and smart watches, it makes the synchronized WiFi *received signal strength (RSS)* sequences and inertial data sequences collection very convenient. Therefore, unsupervised indoor radio-map construction based on massive unlabeled RSS+IMU sequences is intensively studied.

There are three major ways for unsupervised indoor location based on RSS+IMU sequences:(1) Exploiting the floor-map information[43][12][41], which estimates distances among fingerprints by user mobility. It transforms the floor-map to a high dimension space in which the distance between two locations reflects the walking distances. Then the alignment of the stress-free floor plan and fingerprint space is exploited to label fingerprints by real locations. (2) The second method exploits that certain locations in an indoor environment which have identifiable signatures on one or more sensing dimensions, so that mobile devices can detect these landmarks as the user moving to calibrate the signal signature at these locations. The trajectory between landmarks are inferred by dead-reckoning schemes[47][33]. (3) The third method exploits WiFi-SLAM, which formulates a network of Gaussian constraints between robot poses and landmarks[10]. The WiFi measurement is modeled as a Guassian process of underlying locations, and the locations are solved by non-linear graph optimization by maximizing the likelihood of observations. The solved locations then give labels to the WiFi measurements. We leave the detailed introduction of related works to Section. 2.

However, these methods have limitations on that they need external information or knowledge. The first method needs floor-map, the second method needs additional sensing methods to extract organic landmarks, and the third method needs long and smooth trajectories for loop closure detection and for keeping solution accuracy. The WiFi-SLAM method is only proved effective in small scale scenarios with repetitive paths[10][18]. They are not suitable to process the massive, crowd-sourced, unlabeled RSS+IMU sequences which generally do not have additional knowledge. To address these challenges, this paper proposes *transitional model (TM)* based methods to address the problem of *unsupervised localization with unlabeled training data and without additional knowledge.*

We consider there are massive fragmented and unlabelled RSS+IMU sequences which are collected by crowd sourcing. The RSS and IMU sequences are denoted by $\{\mathbf{Z}^f\}$ and $\{\mathbf{U}^f\}$. $\mathbf{Z}^f = \{\mathbf{z}_1^f, \mathbf{z}_2^f, \ldots\}$ represent a RSS sequence. Each RSS vector $\mathbf{z}_t^f$ is composed by RSS values measured from surrounding access points (APs). It is the RSS scan taken at time t on the trajectory. The IMU sensors generally collect acceleration and angular velocities in much higher frequency than the WiFi scans. We assume the IMU data between two WiFi scans are pre-processed by pedestrian dead reckoning (PDR) algorithms [28] to generate a motion vector, i.e., $\mathbf{u}_t^f$, which indicates the user motion between RSS samples $\mathbf{z}_{t-1}^f$ and $\mathbf{z}_t^f$. So let $\mathbf{U}^f = \{\mathbf{u}_1^f, \mathbf{u}_2^f, \ldots\}$ be the 2D movement vectors corresponding to the RSS sequence $\mathbf{Z}^f$. There are three sub-problems we want to address:

(1) How to off-line learn a radio-map model that models the relationship between the physical positions of the area of interest and the signal signatures in the signal space based on these unlabeled sequences.

(2) How to realize an on-line localization algorithm which can accurately predict $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$ given the online measured data sequences $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{N-1}\}$ and $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}$ of a user, where $\mathbf{p}_t$ is the predicted location of the user at time t.
(3) How to realize a radio-map feedback mechanism, which uses on-line collected sequences $\mathbf{U}$ and $\mathbf{Z}$ to refine the off-line learned radio-map model.

The key challenges to address these problems are that the RSS measurements $\{\mathbf{Z}^f\}$ and inertial measurements $\{\mathbf{U}^f\}$ are noisy and unlabeled. Without the external knowledge, in this paper, we introduce a novel idea to learn a *transitional model (TM)* to solve the above problems. The TM off-line captures the expected relationship between $\{\mathbf{z}_t, \mathbf{z}_{t-1}\}$ and $\mathbf{u}_{t-1}$, i.e., the relationship between the signal state change and the location change from the crowd-sourced unlabeled $\{\mathbf{Z}^f\}$ and $\{\mathbf{U}^f\}$. The difference from TM to traditional radio-map methods is that: due to the missing of absolute position labels from $\{\mathbf{Z}^f\}$ and $\{\mathbf{U}^f\}$, the fingerprint relationship between $\mathbf{z}_i$ and $\mathbf{p}_i$ or the kernel relationship between $\{\mathbf{z}_i, \mathbf{z}_j\}$ and $\{\mathbf{p}_i, \mathbf{p}_j\}$ cannot be known. But we can learn the relatively weak kernel relationship between $\{\mathbf{z}_{t-1}, \mathbf{z}_t\}$ and $\mathbf{u}_{t-1} = \mathbf{p}_t - \mathbf{p}_{t-1}$ where $\mathbf{u}_{t-1}$, $\mathbf{z}_{t-1}$ and $\mathbf{z}_t$ can be extracted by aggregating the nearest neighbours in the crowd-sourced sequences.

In particular, we propose a *transition model in signal space (TMS)*, which learns $\overline{\mathbf{z}}_t = TMS(\mathbf{z}_{t-1}, \mathbf{u}_{t-1})$ and a *transition model to predict motion by signal change (TMM)*, which learns $\hat{\mathbf{u}}_{t-1} = TMM(\mathbf{z}_{t-1}, \mathbf{z}_t)$ based on the off-line RSS+IMU sequences. Correspondingly, Kalman filer based on-line locating methods are proposed for corresponding TMs, i.e., *extended Kalman filter for TMS (TMS-EKF)* and *Kalman filter for TMM (TMM-KF)*. We also show the TMS and TMM models can be on-line refined as RSS+IMU sequences are incrementally collected. Experiments are conducted which prove the effectiveness of TM-based methods in locating accuracy and self-learning ability. The contributions of the paper are summarized as following:

(1) At first, the unlabeled RSS and IMU sequences provide massive observations of RSS transitions between $\mathbf{z}_{t-1}$ and $\mathbf{z}_t$ triggered by motions $\mathbf{u}_{t-1}$ recorded by IMUs. We propose the idea of using the TM to effectively capture some specific expected relationships in extracted triples, i.e., $\{\mathbf{u}_{t-1}, \mathbf{z}_{t-1}, \mathbf{z}_t\}$.
(2) Secondly, this paper gives two specific forms $\overline{\mathbf{z}}_t = TMS(\mathbf{z}_{t-1}, \mathbf{u}_{t-1})$ and $\hat{\mathbf{u}}_{t-1} = TMM(\mathbf{z}_{t-1}, \mathbf{z}_t)$. As the learned TM aims to return the expected values from the noisy input values, we propose one basic smoothing method for TMS and TMM model training, which is conducted by *K-nearest neighbors (KNN)* algorithm. Other smoothing algorithms can also be exploited to train the TM model.
(3) Thirdly, two forms of on-line locating methods are proposed, i.e., *TMS-EKF* and *TMM-KF*, which use the TM for on-line tracking the mobile targets.
(4) At last, simulations and experiments on real datasets show the two TMs can be learned rather accurately from the unlabeled crowded sourced data without any floor-map or trace information. TM-based methods provide on-line location accuracy comparable to the traditional radio-map methods, which proves they are promising methods for large scale WiFi locating via crowd sourcing.

Table. 1 gives the list of notations and explanations. The remaining sections of this paper are organized as following. The related works are introduced in Section. 2. The problem model and the TM-based unsupervised locating methods are introduced in Section. 3. Simulation results are reported in Section. 4. Results of real experiments are given in Section. 5. The paper is concluded with discussions in Section. 6.

2 RELATED WORKS

WiFi fingerprint based indoor locating has been widely investigated. Fast supervised fingerprint calibration methods using motion trajectories are investigated in [45]. We review the related work of semi-supervised radio-map construction and unsupervised radio-map construction methods in this section. Detailed formulas of WiFi-SLAM are given, which are used as the strong motivation for our works.

Table 1. List of notations and explanations

Notations	Explanations	Notations	Explanations
$\{\mathbf{Z}^f\}$	off-line RSS data	$\{\mathbf{U}^f\}$	off-line IMU data
$\mathbf{Z}^f$	one specific off-line RSS sequence	$\mathbf{U}^f$	one specific off-line IMU sequence
$\mathbf{z}_t^f$	one specific off-line RSS signal at time t	$\mathbf{u}_t^f$	one specific off-line IMU motion at time t
$\mathbf{Z}$	one on-line RSS sequence	$\mathbf{U}$	one on-line IMU sequence
$\mathbf{P}$	one ground-truth position sequence	$\mathbf{z}_t$	one specific on-line RSS signal at time t
$\mathbf{u}_t$	one specific on-line IMU motion at time t	L	the dimension of signal
N	the time length of $\mathbf{P}$	$\bar{\mathbf{p}}_t$	the estimate of $\mathbf{p}_t$ based on IMU data
$\bar{\mathbf{z}}_t$	the estimate of $\mathbf{z}_t$ based on IMU data	$\hat{\mathbf{p}}_t$	the estimate of $\mathbf{p}_t$ based on RSS data
$\hat{\mathbf{u}}_t$	the estimate of $\mathbf{u}_t$ based on RSS data	$\mathbf{T}_h$	the off-line TMS database
$\mathbf{T}_h^c$	the off-line candidate set for TMS	$\mathbf{T}_h^u$	the update set for TMS
$\mathbf{T}_l$	the off-line TMM database	$\mathbf{T}_l^c$	the off-line candidate set for TMM
$\mathbf{T}_l^u$	the update set for TMM	M	the size of all off-line triples
M_1	the size of $\mathbf{T}_h^c$ or $\mathbf{T}_l^c$	M_2	the size of $\mathbf{T}_h$ or $\mathbf{T}_l$
M_3	the size of $\mathbf{T}_h^u$ or $\mathbf{T}_l^u$		

2.1 Semi-supervised Methods Using a Portion of Labelled Data

Different semi-supervised radio-map calibration methods were proposed in the literature. *Manifold method* is one of the representative methods to infer location labels based on partially labeled input data. Pan et al. [27] adopt a SVD based method to conduct dimension reduction and use manifold assumption to infer target location. Pulkkinen et al. [31] using ISOMap method to conduct the dimension reduction to map the measured signal strength information to low dimension to assign the location labels. Majeed et al.[24] propose to use very low rate of labelled data to perturbing the local geometries of the plan coordinates. It reduces the required portion of labelled data. Zhou et al.[48] propose graph-based semi-supervised manifold alignment approach to construct a cost-efficient radio map by a small number of labeled fingerprints. Gu et al. [13] proposed Semi-supervised deep extreme learning machine for Wi-Fi based localization. This method improves the feature extraction procedure in model learning and on-line classification.

2.2 Unsupervised Methods Using Floor-Map Information

LIFS-based method [43] and its variations [12][41] exploit floor map information to carry out unsupervised learning. Yang et al. [43] consider the RSS fingerprints collected on a user moving path are geographically connected. Therefore, the high dimension fingerprint sequences are associated onto the paths recorded by IMUs. The spatial similarity of stress-free floor plan and fingerprint space is then utilized to automatically label the fingerprint sequences. Hilsenbeck et al.[16] propose a graph-based data fusion method, which propose a graph representation to the indoor environment and utilize pedometer and WiFi information for indoor localization. Xiao et al. [42] propose Markov Random Field based method to model the floor map and a MRF-based localization algorithm is developed. But the requirement of digital floor map prevents the wide applicability of these methods in crowd sourcing scenarios, for the floor maps are hardly available in many buildings. Jung et al. [21] eta al. propose hybrid global-local optimization scheme. The method determines the optimal placement of fingerprint sequences on an indoor map. In a recent work, Gao et al. [12] estimate the surveyor's trajectory post-hoc using Simultaneous Localization and Mapping and particle filtering to incorporate a building floorplan.

2.3 Unsupervised Methods Using WiFi-SLAM

WiFi-SLAM method combines the WiFi model learning and inertial navigation[10][25][7] [22]. The physical transitions between steps are recorded by inertial navigation, which builds spatial links among the RSS measurements. The *Gaussian process latent variable model (GP-LVM)* [10][19] is used to estimate the posterior distributions over functions $g(\cdot)$ from the RSS training sequences. In particular, the location labels of the RSS are treated as latent variables, which are calculated by likelihood maximization to enable unsupervised RSS-location model training. But WiFi-SLAM based methods generally requires the training path is long for the purpose of closure detection. Huang et al. [18] relaxed the assumption of RSS fingerprint uniqueness in the radio-map and proposed a more general and more efficient WiFi-SLAM method. A related work proposed by Fink et al. [11] is to online mapping the received radio signal strength by mobile robots and localizing the source of the radio signal. In later works, Hilsenbeck et al. [16] propose a graph-based, low-complexity sensor fusion approach for ubiquitous pedestrian indoor positioning using mobile devices. They employ fusion technique to combine relative motion information based on step detection with WiFi signal strength measurements. Mirowski et al. [25] propose SignalSLAM for simultaneous localization and mapping with mixed WiFi, Bluetooth, LTE and magnetic signals. It is not suitable when there are massive, unlabeled and fragmented RSS sequences reported by the mobile users.

2.4 Unsupervised Methods by Detecting Organic Landmarks

Another method of unsupervised locating is to use additional sensing to detect organic landmarks in the environment. Park et al.[29] proposed the idea of detecting organic landmarks from the environment using additional sensors while conducting the WiFi fingerprint surveying. The locations of organic landmarks are estimated by dead reckoning distances so as to estimate the location labels of the Wifi fingerprints. [40] propose to localize the organic landmarks using extended kalman filter based SLAM method. [9] exploits electrical compass to discover organic landmarks. Shen et al. propose Walkie-Markie[32] which detect organic landmarks by using the radio signal attending signature of indoor pathways. More related works for unsupervised location model training can be referred to [17].

3 TRANSITION MODEL

3.1 Problem Model

We consider the problem when massive RSS+IMU sequences are collected by crowd-sourced data from mobile users who move in an area of interest. Let's denote collected sequences as $\{\mathbf{Z}^f\}$ and $\{\mathbf{U}^f\}$ where $\mathbf{Z}^f = \{\mathbf{z}_1^f, \mathbf{z}_2^f, \ldots\}$ and $\mathbf{U}^f = \{\mathbf{u}_1^f, \mathbf{u}_2^f, \ldots\}$.These sequences are assumed redundantly collected in the area of interest. Some sequences maybe overlapped. No location labels are available in these sequences and the floor-map and motion patterns of users are unknown.

The problem we consider is to learn a model that maps the relative signal state change triggered by the user motion. It contains an off-line phase and an on-line phase. In the off-line phase, massive RSS+IMU sequences are processed to learn the *transition model (TM)*. In the on-line phase, on one hand the on-line collected RSS+IMU sequence is used to update the TM, on the other hand, the real-time location of the user is estimated by Kalman filter using the TM. The overview of TM framework is shown in Fig. 1. It illustrates the main steps of TM based unsupervised learning, online model updating and locating.

3.2 Motivation and Basic Idea

The key motivation is from locating algorithms that use *extended Kalman filter (EKF)* [34] based on WiFi *fingerprints (FPs)*[37]. The key step for state prediction is as follow.

$$\overline{\mathbf{p}}_t = \mathbf{p}_{t-1} + \mathbf{u}_t \tag{1}$$

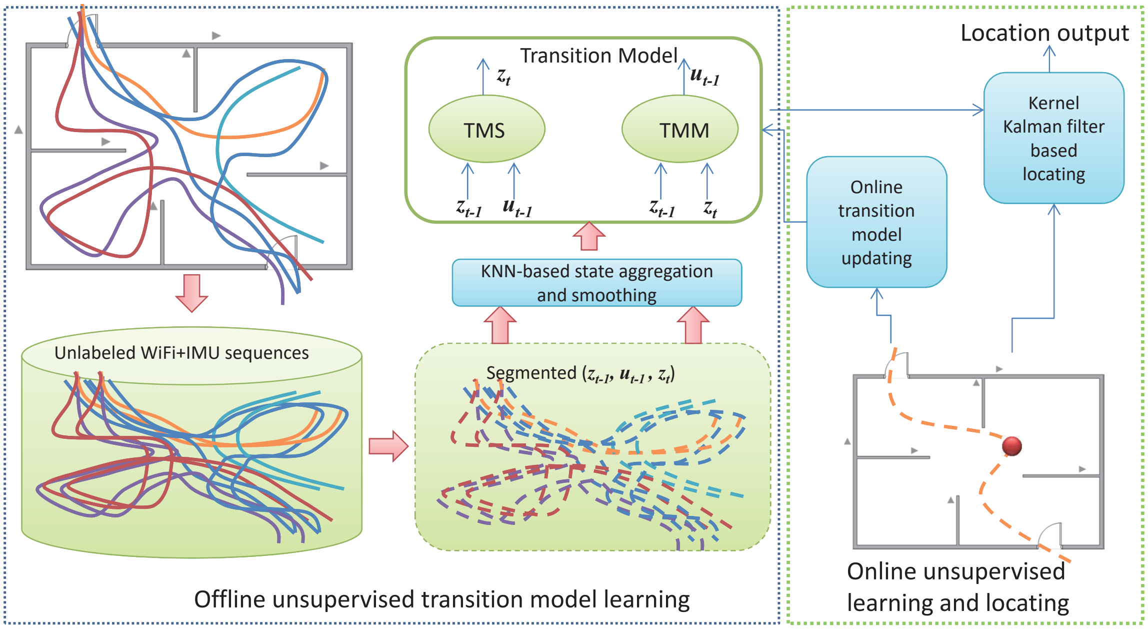

Location output
Transition Model
z_t
u_{t-1}
TMS
TMM
z_{t-1} u_{t-1}
z_{t-1} z_t
Kernel Kalman filter based locating
Online transition model updating
KNN-based state aggregation and smoothing
Unlabeled WiFi+IMU sequences
Segmented (z_{t-1}, u_{t-1}, z_t)
Offline unsupervised transition model learning
Online unsupervised learning and locating

Fig. 1. The overview of transition model based unsupervised localization framework

$\mathbf{u}_t$ is calculated based on the IMU information and $\overline{\mathbf{p}}_t$ is the predicted user location at t based on $\mathbf{u}_t$. Then by observing signal vector $\mathbf{z}_t$, user's location is updated to:

$$\mathbf{p}_t = \overline{\mathbf{p}}_t + \mathbf{K}_t(FPLow(\mathbf{z}_t) - \overline{\mathbf{p}}_t) \tag{2}$$

where $\mathbf{K}_t$ is the Kalman gain and $FPLow(\mathbf{z}_t)$ is the off-line trained fingerprint radio-map. Observation of $\hat{\mathbf{p}}_t$ is inferred from the radio-map model by $\hat{\mathbf{p}}_t = FPLow(\mathbf{z}_t)$. Because $FPLow(\mathbf{z}_t)$ projects high dimensional signal $\mathbf{z}_t$ to low dimensional location $\hat{\mathbf{p}}_t$, we call this model the *low dimensional fingerprint model (FPLow)*. Then the user location is corrected by Equ. 2 via correcting the predicted location $\overline{\mathbf{p}}_t$ by the signal observation. We can also use the fingerprint radio-map reversely. Assume we can generate the expected RSS vector based on the predicted user location by a *high dimensional fingerprint model (FPHigh)*, then the location correction step in Equ. 2 can be revised as:

$$\mathbf{p}_t = \overline{\mathbf{p}}_t + \mathbf{K}_t(\mathbf{z}_t - FPHigh(\overline{\mathbf{p}}_t)) \tag{3}$$

$\overline{\mathbf{z}}_t = FPHigh(\overline{\mathbf{p}}_t)$ is the fingerprint model to approximate the unknown actual function $\mathbf{z}_t = g(\mathbf{p}_t)$. Equ. 3 uses the difference between $\mathbf{z}_t$ and $\overline{\mathbf{z}}_t$ to correct $\overline{\mathbf{p}}_t$. Since $\overline{\mathbf{p}}_t = \mathbf{p}_{t-1} + \mathbf{u}_t$, we rewrite:

$$\overline{\mathbf{z}}_t = FPHigh(\mathbf{p}_{t-1} + \mathbf{u}_t) \tag{4}$$

This model projects the low dimension location to high dimension signal fingerprint.

In the unsupervised case, $\mathbf{p}_{t-1}$ cannot be known exactly when location labels are not available, but it must be a function of $\mathbf{z}_{t-1}$. Therefore, we assume $\mathbf{p}_{t-1} = g^{-1}(\mathbf{z}_{t-1})$. Then Equ. 4 is rewritten as:

$$\overline{\mathbf{z}}_t = FPHigh(g^{-1}(\mathbf{z}_{t-1}) + \mathbf{u}_t) \tag{5}$$

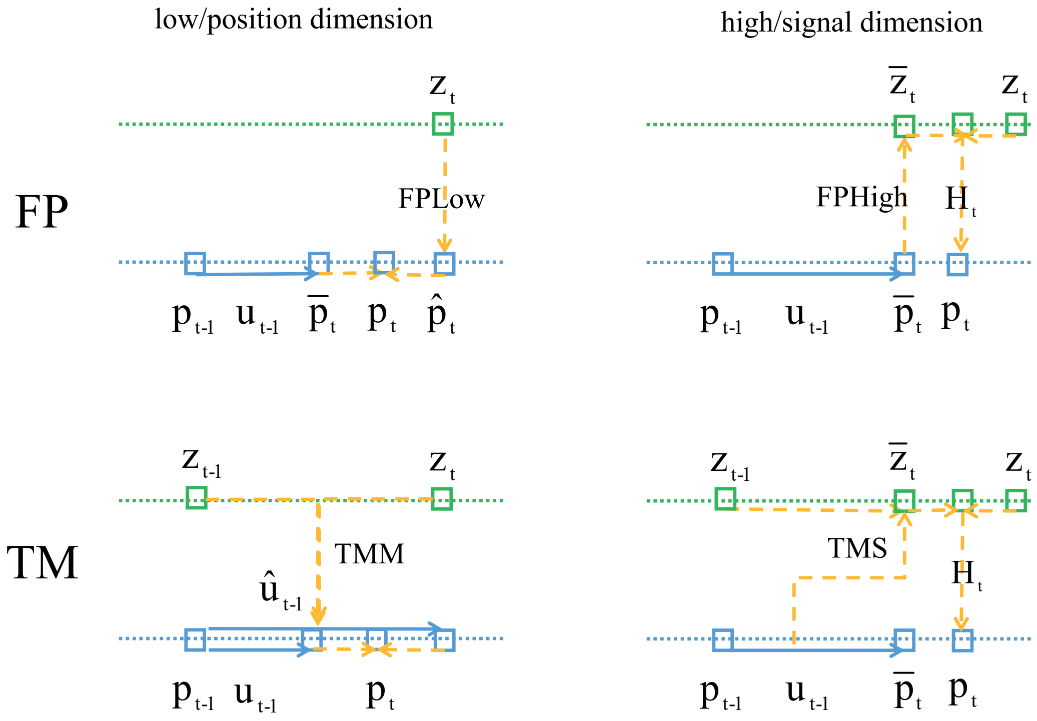

low/position dimension
high/signal dimension
FP
TM
z_t
FPLow
p_{t-1} u_{t-1} $\bar{p}_t$ p_t $\hat{p}_t$
$\bar{z}_t$ z_t
FPHigh
H_t
p_{t-1} u_{t-1} $\bar{p}_t$ p_t
z_{t-1} z_t
TMM
$\hat{u}_{t-1}$
p_{t-1} u_{t-1} p_t
z_{t-1} $\bar{z}_t$ z_t
TMS
H_t
p_{t-1} u_{t-1} $\bar{p}_t$ p_t

Fig. 2. The on-line localization phase

We therefore propose a *transition model in signal space (TMS)* to model the signal map purely by signal space observations and the IMU measurements.

$$\bar{\mathbf{z}}_t = TMS(\mathbf{z}_{t-1}, \mathbf{u}_t) \tag{6}$$

The TMS can be trained off-line by RSS+IMU sequences without the needs of location labels. Since $\bar{\mathbf{z}}_\mathbf{t}$ stays in signal space, we call this model *transition model in signal space*, i.e., *TMS*. Correspondingly, we also propose the *transition model to predict the motion from signal change*, i.e., *TMM*.

$$\hat{\mathbf{u}}_{t-1} = TMM(\mathbf{z}_{t-1}, \mathbf{z}_t) \tag{7}$$

where $\hat{\mathbf{u}}_{t-1}$ is the predicted location change by giving $\mathbf{z}_{t-1}$ and $\mathbf{z}_t$. We provide the detailed design of TMS and TMM in the following subsections.

3.3 TMS: Transitional Model in Signal Space

The TMS method consists of three phases: (1) the expected TMS relationship, i.e., the Equ. 6, is off-line learned from $\{\mathbf{Z}^f\}$ and $\{\mathbf{U}^f\}$; (2) the *extended Kalman filter for TMS (TMS-EKF)* method needs to be designed to on-line track the user's position; (3) one feedback mechanism is designed to refine off-line learned TMS when on-line $\mathbf{Z}$ and $\mathbf{U}$ are incrementally collected.

3.3.1 Off-line Learning of TMS. The off-line phase is divided into two steps: (1) dividing training data $\{\mathbf{Z}^f\}$ and $\{\mathbf{U}^f\}$ into needed triples $\{\mathbf{u}^f_{t-1}, \mathbf{z}^f_{t-1}, \mathbf{z}^f_t\}$; (2) filtering triples' noises and constructing the expected TMS model, i.e., Equ. 6.

The segmentation phase is very intuitive, which is benefited by the convenience of synchronized data collected of RSS+IMU sequences by mobile devices. What needs illustration is that $\mathbf{u}^f_{t-1}$ spanning one certain time period should be estimated by dead reckoning [6], such as zero-velocity aided inertial navigation algorithm [33] in advance of the processing the triples. We segment each $\{\mathbf{Z}^f\}$ and $\{\mathbf{U}^f\}$ into step length segments and suppose there are totally M segmented triples are obtained after sequence segmentation.

The segmented triples are highly noisy. One method to filter the noises of triples, which also means geting the expectation of Equ. 6, is designed as follow. We divide the M triples into two sets: a *candidate set* $\mathbf{T}^c_h$ whose size is M_1 and a *target set* $\mathbf{T}_h$ whose size is $M_2 = M - M_1$. M_2 should be kept sufficiently large to cover the whole environment. The key idea of this smoothing method is that: for each target triple we find top-K similar triples in $\mathbf{T}^c_h \cup \mathbf{T}_h$ and use the average of these similar triples to smooth the target triple. After processing all the triples in the target set, the smoothed $\mathbf{T}_h$ can be regarded as Equ. 6, which is also called as the *TMS database*.

Algorithm. 1 gives the detailed off-line learning phase of for the TMS model. There are many alternative methods to search for similar triples (line 12-13), e.g., the above mentioned *K-nearest neighbors (KNN)* [30] or the matching algorithms based on different distance functions [23][8]. We use KNN to select the top-K similar triples in our implementation. In the future, other smoothing methods can be investigated.

Algorithm 1 $[\mathbf{T}_h]$ = TMS Database Construction($\{\mathbf{Z}^f\}$, $\{\mathbf{U}^f\}$)

```
 1: %Segmentation phase
 2: for each $\mathbf{U}^f \in \{\mathbf{U}^f\}$ do
 3:     for each $\mathbf{u}^f_t \in \mathbf{U}^f$ do
 4:         Extract corresponding $\mathbf{z}^f_{t-1}$ and $\mathbf{z}^f_t$
 5:         Add $\{\mathbf{u}^f_{t-1}, \mathbf{z}^f_{t-1}, \mathbf{z}^f_t\}$ in the triple database
 6:     end for
 7: end for
 8: %Filter phase
 9: Divide the triple database into the candidate set $\mathbf{T}^c_h$ and the target set $\mathbf{T}_h$
10: for each triple $\in \mathbf{T}_h$ do
11:     Use its $\mathbf{u}^f_{t-1}$ as the matching vector to get top-$K$ similar triples from $\mathbf{T}^c_h \cup \mathbf{T}_h$
12:     Use its $\mathbf{z}^f_{t-1}$ as the matching vector to get top-$K$ similar triples from $\mathbf{T}^c_h \cup \mathbf{T}_h$
13:     Get the intersection of the two obtained similar triple sets
14:     Use the average of the intersection to replace the triple in $\mathbf{T}_h$
15: end for
```

Theorem 3.1. *If there are totally M triples, which are divided into a candidate set of M_1 triples and a target set of M_2 triples, then the computation complexity of Algorithm. 1 for off-line TMS learning is $O(M + 2M_2 * M * log(M))$.*

Proof. The computation complexity of the segmentation phase is $O(M)$. For each triple in the target set, it searches the top-K similar triples in $\mathbf{T}^c_h \cup \mathbf{T}_h$. The sorting cost of the M triples is $O(Mlog(M))$ for both line 12 and line 13, which needs $O(2Mlog(M))$ cost. So the total cost of off-line TMS learning is $O(M + 2M_2Mlog(M))$. □

We can control the segment length to control the size of M, M_1 and M_2. The computation cost to obtain the off-line TMS model can be decreased with larger step length.

3.3.2 On-line Locating by TMS. In the on-line locating phase, a triple $\{\mathbf{u}_{t-1}, \mathbf{z}_{t-1}, \mathbf{z}_t\}$ is on-line constructed in each time instance to search the TMS database to estimate the location of the target. We propose an EKF-based on-line locating algorithm for the TMS model. It contains three steps: (1) obtaining the predicted signal state using the TMS model; (2) calculating the observation matrix; (3) updating location by the EKF iteration.

(1) By substituting $\{\mathbf{u}_{t-1}, \mathbf{z}_{t-1}\}$ into the TMS model, we want to obtain $\overline{\mathbf{z}}_t = TMS(\mathbf{z}_{t-1}, \mathbf{u}_{t-1})$, i.e., the expected signal state from the TMS. Since there may be no exact match of $\{\mathbf{u}_{t-1}, \mathbf{z}_{t-1}\}$ in the TMS database, we propose a KNN-based smoothing method to predict $\overline{\mathbf{z}}_t$. In particular, we use $\mathbf{u}_{t-1}$ and $\mathbf{z}_{t-1}$ to search the top-K similar triples in $\mathbf{T}_h$ and use the average to predict $\overline{\mathbf{z}}_t$, and also estimate $\overline{\mathbf{z}}_{t-1}$ for calculating the observation matrix accurately. The algorithm is given in Algorithm. 2.

Algorithm 2 $[\overline{\mathbf{z}_t}, \overline{\mathbf{z}}_{t-1}]$ = TMS($\mathbf{u}_{t-1}$, $\mathbf{z}_{t-1}$)

```
Use $\mathbf{u}_{t-1}$ as the matching vector to get top-$K$ similar triples from $\mathbf{T}_h$, denoted by set $\mathrm{S}_u$
Use $\mathbf{z}_{t-1}$ as the matching vector to get top-$K$ similar triples from $\mathbf{T}_h$, denoted by set $\mathrm{S}_z$
Get the intersection set $\mathrm{S}_{in}$ of the two result triple sets $\mathrm{S}_u$ and $\mathrm{S}_z$
$\overline{\mathbf{z}}_{t-1}$ is estimated by averaging $\mathbf{z}_{t-1}$ in set $\mathrm{S}_z$
$\overline{\mathbf{z}}_t$ is estimated by averaging $\mathbf{z}_t$ in the intersection set $\mathrm{S}_{in}$
```

(2) Then we construct $\mathbf{H}_t$, which represents the observation matrix relating the signal state observation to the target location. It is the derivative of RSS observations over the derivative of location at time t. By the predicted $\overline{\mathbf{z}}_t$ and the smoothed $\overline{\mathbf{z}}_{t-1}$, the observation matrix $\mathbf{H}_t$ can be estimated by:

$$\mathbf{H}_t = \left[\frac{\overline{\mathbf{z}}_t - \overline{\mathbf{z}}_{t-1}}{\mathbf{u}_{t-1_1}}, \frac{\overline{\mathbf{z}}_t - \overline{\mathbf{z}}_{t-1}}{\mathbf{u}_{t-1_2}}\right] \tag{8}$$

where $\overline{\mathbf{z}}_t$ and $\overline{\mathbf{z}}_{t-1}$ are outputs of Algorithm. 2. $\mathbf{u}_{t-1_1}$ is the x-value of $\mathbf{u}_{t-1}$ while $\mathbf{u}_{t-1_2}$ is the y-value of $\mathbf{u}_{t-1}$. It approximates the derivative of location change by the one step motion $\mathbf{u}_{t-1}$.

(3) Based on the predicted signal state and the derived observation matrix, the mean and variance matrix of the target location is calculated using the EKF iteration, which is given in Algorithm 3.

Algorithm 3 $[\mathbf{p}_t, \Sigma_t]$ = TMS-EKF($\mathbf{p}_{t-1}, \Sigma_{t-1}, \mathbf{u}_{t-1}, \mathbf{z}_t, \mathbf{z}_{t-1}$)

```
1: %Prediction phase
2: $\mathbf{R}_t = \sigma_r^2\mathbf{I}$
3: $\overline{\mathbf{p}_t} = \mathbf{p}_{t-1} + \mathbf{u}_{t-1}$
4: $\overline{\Sigma_t} = \Sigma_{t-1} + \mathbf{R}_t$
5: %Correction phase
6: $[\overline{\mathbf{z}_t}, \overline{\mathbf{z}}_{t-1}]$ = TMS($\mathbf{u}_{t-1}$, $\mathbf{z}_{t-1}$)
7: $\mathbf{H}_t = \left[\frac{\overline{\mathbf{z}_t} - \overline{\mathbf{z}}_{t-1}}{\mathbf{u}_{t-1_1}}, \frac{\overline{\mathbf{z}_t} - \overline{\mathbf{z}}_{t-1}}{\mathbf{u}_{t-1_2}}\right]$
8: $\mathbf{Q}_t = \sigma_q^2\mathbf{I}$
9: $\mathbf{S}_t = \mathbf{H}_t\overline{\Sigma_t}\mathbf{H}_t^T + \mathbf{Q}_t$
10: $\mathbf{K}_t = \overline{\Sigma_t}\mathbf{H}_t^T\mathbf{S}_t^{-1}$
11: $\mathbf{p}_t = \overline{\mathbf{p}_t} + \mathbf{K}_t(\mathbf{z}_t - \overline{\mathbf{z}_t})$
12: $\Sigma_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\overline{\Sigma_t}$
```

Theorem 3.2. *The on-line computation complexity of TMS-EKF is $O(M_2 logM_2 + L^3)$.*

Proof. The smoothing in line 6 has complexity $O(M_2 logM_2)$ because of sorting $\mathbf{T}_h$. Then the computation complexity of TMS-EKF is basically identical to the traditional EKF algorithm. The main cost is inversion computation of $\mathbf{S}_t$ whose size is $L * L$ (L is the dimension of signal), which is $O(L^3)$. When L is small, the computation complexity of TMS-EKF can be regarded as $O(M_2 log M_2)$. □

3.3.3 Online Update TMS Model by User Sequence. An distinctive feature of the TMS model is that the newly collected $\mathbf{Z}$ and $\mathbf{U}$ can be used to refine the off-line learned TMS model. The method is to refine the learned $\mathbf{T}_h$ by the on-line collected sequence. To avoid updating too frequently, the on-line newly collected triples are accumulated into a set $\mathbf{T}_h^u$ until its size reaching M_3. Then each triple $\{\mathbf{u}_{t-1}, \mathbf{z}_{t-1}, \mathbf{z}_t\}$ is updated by merging $\mathbf{T}_h^u$ into $\mathbf{T}_h$. Algorithm. 4 gives the algorithm details.

```
Algorithm 4 [$\mathbf{T}_h$] = TMS Database Update($\mathbf{Z}$, $\mathbf{U}$)
%Segmentation phase
Divide $\mathbf{Z}$ and $\mathbf{U}$ into triples using segmentation method
Accumulate the online segmented triples into a set $\mathbf{T}_h^u$
%Filter phase
if the size of $\mathbf{T}_h^u > M_3$ then
    for each triple $\in \mathbf{T}_h$ do
        Use its $\mathbf{u}_{t-1}$ as the matching vector to get top-$K$ similar triples from $\mathbf{T}_h^u \cup \mathbf{T}_h$
        Use its $\mathbf{z}_{t-1}$ as the matching vector to get top-$K$ similar triples from $\mathbf{T}_h^u \cup \mathbf{T}_h$
        Get the intersection of the two resulting sets
        Use the average of the intersection to replace the triple in $\mathbf{T}_h$
    end for
    Clear the update set
end if
```

Theorem 3.3. *The computation complexity of Algorithm. 4 is* $O(M_3 + 2M_2 * (M_3 + M2) * log(M_3 + M_2))$.

Proof. The computation complexity of the segmentation phase can be regarded as $O(M_3)$. The filter phase of Algorithm. 4 is $O(2M_2 * (M_3 + M2) * log(M_3 + M_2))$ computation complexity. So the total cost is $O(M_3 + 2M_2 * (M_3 + M2) * log(M_3 + M_2))$. □

3.4 TMM: Transition Model to Predict Motion from Signal Change

The TMM model is a different representation of TMS. It estimates the expected motion $\hat{\mathbf{u}}_{t-1}$ by $\mathbf{z}_t$ and $\mathbf{z}_{t-1}$, i.e., $\mathbf{u}_{t-1} = TMM(\mathbf{z}_t, \mathbf{z}_{t-1})$. The TMM-based localization method also consists of three phases: (1) the *TMM database* is off-line built; (2) one KF-based on-line locating algorithm is designed for TMM; (3) one online feedback mechanism is proposed to refine TMM.

3.4.1 Off-line Learning. The off-line phase of TMM learning also consists of the segmentation phase and the filter phase. The main difference between two models is the expected input values and the output results are different referring to Equ. 6 and Equ. 7. We directly give the off-line learning algorithm for the TMM model.

Algorithm. 5 has computation complexity $O(M + 2M_2 * M * log(M))$. The reason is that the dimension of signal L can be regarded as $O(1)$. The other computation cost is the same as Algorithm. 1.

3.4.2 On-line Locating Using TMM. Different from that in TMS, the predicted motion $\overline{\mathbf{u}}_{t-1}$ can be directly calculated by inertial data. The observation of $\hat{\mathbf{u}}_{t-1}$ is calculated by TMM model via two signal states $\mathbf{z}_{t-1}$ and $\mathbf{z}_t$. The online KF structure is modified in Algorithm. 6 by directly using $\mathbf{u}_{t-1}$ as the estimated state. This avoids to calculate the observation matrix for $\mathbf{p}_t$. The on-line locating complexity of TMM-KF is also $O(M_2 log M_2 + L^3)$. Then the location is updated by adding the estimated $\mathbf{u}_{t-1}$ to $\mathbf{p}_{t-1}$.

Algorithm 5 $[\mathbf{T}_l]$ = TMM Database Construction($\{\mathbf{Z}^f\}$, $\{\mathbf{U}^f\}$)

```
%Segmentation phase
Use the same segmentation strategy to construct the triple database
%Filter phase
Divide the triple database into the candidate set $\mathbf{T}_l^c$ and the target set $\mathbf{T}_l$
for each triple $\in \mathbf{T}_l$ do
    Use its $\mathbf{z}_{t-1}^f$ as the matching vector to get top-$K$ similar triples from $\mathbf{T}_l^c \cup \mathbf{T}_l$
    Use its $\mathbf{z}_t^f$ as the matching vector to get top-$K$ similar triples from $\mathbf{T}_l^c \cup \mathbf{T}_l$
    Get the intersection of two similar sets
    Use the average of the intersection to replace the triple in $\mathbf{T}_l$
end for
```

It is easy to find TMS and TMM related methods are both greedy algorithms, which means they aim to make the optimal estimate for the current step and omit the error accumulation of previous steps. This is one disadvantage of proposed TM-based methods and can be detailed discussed in our future works.

3.4.3 Online Update TMM Model by User Sequence. The TMM model can also be on-line updated using the on-line collected user sequences. The method is analogous to the method in Algorithm. 4, which is omitted for the space limitation. Its computation complexity of is $O(M_3 + 2M_2 * (M_3 + M2) * log(M_3 + M_2))$.

Algorithm 6 $[\mathbf{u}_{t-1}, \mathbf{p}_t]$ = TMM-KF($\mathbf{u}_{t-1}, \mathbf{z}_t, \mathbf{z}_{t-1}$)

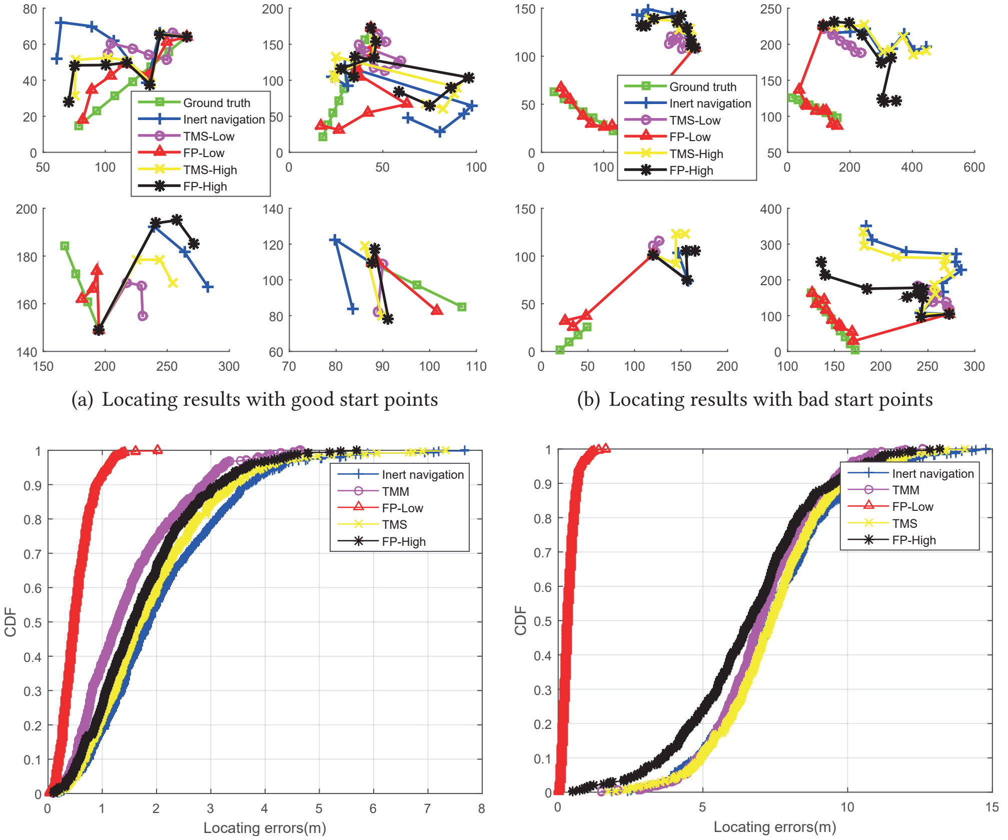
```
%Prediction phase
$\mathbf{R}_t = \sigma_r^2 \mathbf{I}$
$\overline{\mathbf{u}}_{t-1} = \mathbf{u}_{t-1}$
$\overline{\Sigma_t} = \mathbf{R}_t$
%Correction phase
$\hat{\mathbf{u}}_{t-1} = TMM(\mathbf{z}_{t-1}, \mathbf{z}_t)$
$\mathbf{Q}_t = \sigma_q^2 \mathbf{I}$
$\mathbf{S}_t = \overline{\Sigma_t} + \mathbf{Q}_t$
$\mathbf{K}_t = \overline{\Sigma_t} \mathbf{S}_t^{-1}$
$\mathbf{u}_{t-1} = \overline{\mathbf{u}}_{t-1} + \mathbf{K}_t(\hat{\mathbf{u}}_{t-1} - \overline{\mathbf{u}}_{t-1})$
$\Sigma_t = (\mathbf{I} - \mathbf{K}_t)\overline{\Sigma_t}$
$\mathbf{p}_t = \mathbf{p}_{t-1} + \mathbf{u}_{t-1}$
```

4 SIMULATION EVALUATION

In the section, we analyze the localization accuracy and self-learning ability of two TM models and evaluate effect of different factors on them.

4.1 Simulation Settings

4.1.1 Environment Settings. In simulations, $N = 50$ access points (APs) are randomly generated in one $10 * 10m^2 (200 * 200pix^2)$ map. Then, we randomly generate $M_f = 100$ training traces and $M_o = 100$ testing traces in the map. Each trace is divided into steps whose size is $1m(20pix)$. Assume two endpoint positions of current step to be $\mathbf{p}_{t-1}$ and $\mathbf{p}_t$. $\mathbf{u}_{t-1}$ can be calculated by $\mathbf{p}_t - \mathbf{p}_{t-1}$ adding Gaussian noise $N(0, V_u = 1)$. For each $\mathbf{p}_t$, we



(a) Locating results with good start points

(b) Locating results with bad start points

(c) Cumulative distribution function (CDF) graph with good start points

(d) Cumulative distribution function (CDF) graph with bad start points

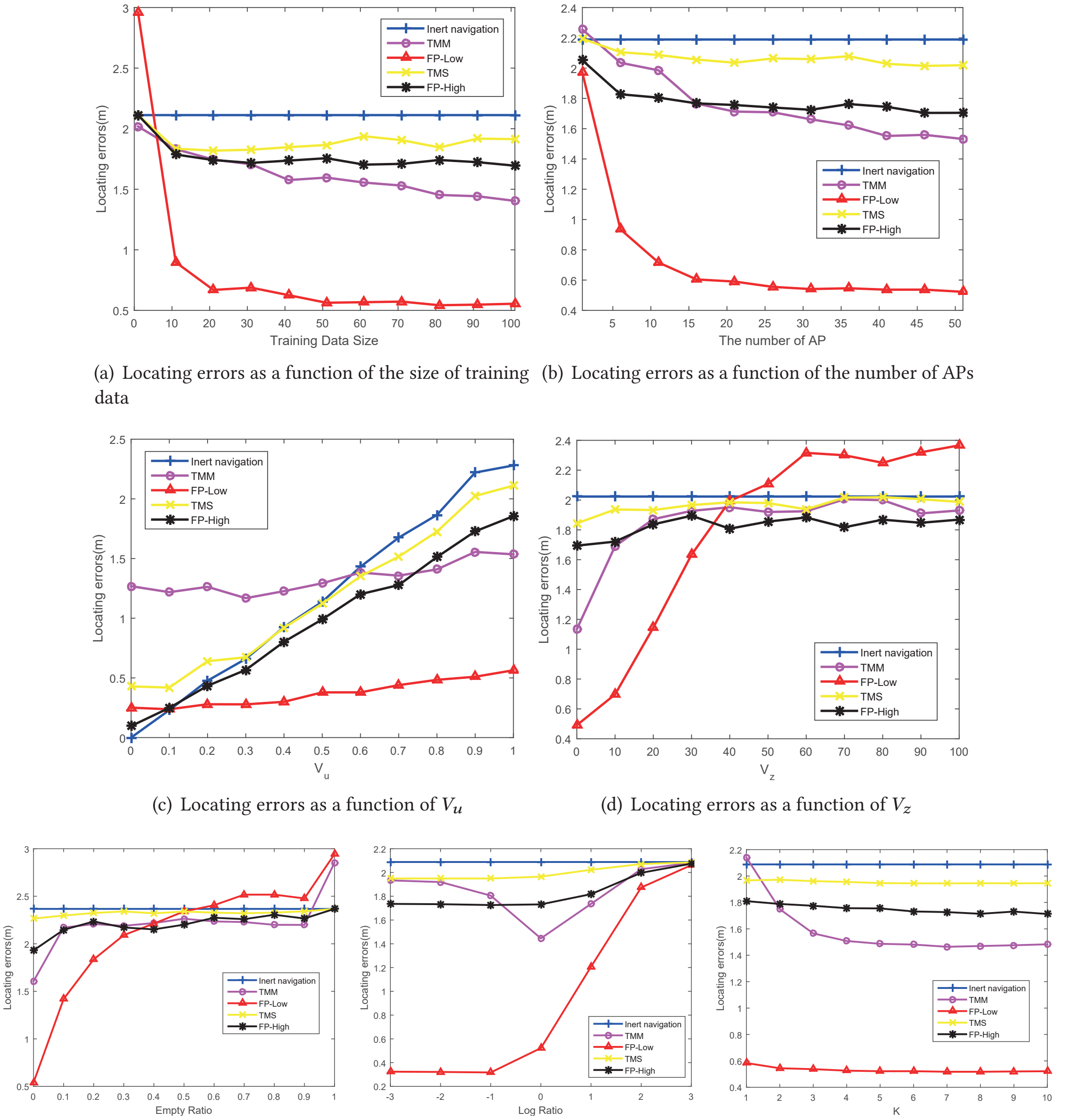
Fig. 3. Simulation evaluation of accuracy

calculate its WiFi signal strength by $\mathbf{z}_t = S_t - S_0 - 10 * 2 * log(d/1)$ adding Gaussian noise $N(0, V_z = 5)$, where d is the distance from AP positions to $\mathbf{p}_t$. We define e to represent the *empty ratio* of $\{\mathbf{Z}^f\}$ and $\mathbf{Z}$ and use the minimal signal value to fill the gaps.

4.1.2 Algorithm Settings. $M_f = 100$ off-line training traces are used to construct the TMS database and the TMM database. In the on-line phase, TMS-EKF and TMM-KF are used to calculate the on-line positions. K, which is the parameter of the off-line smoothing methods, is chosen as 10. σ_q^2 is set as 1 while σ_q^2 is set as 1 and we define $r = log(\sigma_q^2/\sigma_r^2)$, which is equal to 0 here.

4.1.3 Comparison Algorithms. Inertial navigation [6] and FP-based locating methods [37] are chosen as comparison algorithms. The specific steps of two FP algorithms, i.e., FP-High and FP-Low, are introduced firstly.

(1) *FP-Low* Method. The fingerprint method in low dimensional space is just the traditional fingerprint method which calibrate the fingerprint using the location labels in 2D space. Its aim is to estimate $\mathbf{p}_t$ based on RSS



(a) Locating errors as a function of the size of training data

(b) Locating errors as a function of the number of APs

(c) Locating errors as a function of V_u

(d) Locating errors as a function of V_z

(e) Locating errors as a function of e

(f) Locating errors as a function of $r = log(\sigma_q^2/\sigma_r^2)$

(g) Locating errors as a function of K

Fig. 4. Simulation evaluation of different factors

signature. So

$$\hat{\mathbf{p}_t} = FPLow(\mathbf{z}_t) \tag{9}$$

where *FPLow* represent the offline learned fingerprint model. The off-line radio-map of FP-low is trained by associating the smoothed fingerprint vectors to the location labels. The location labels are collected in our data set by marking all turning points on the trajectory during user movement. Then the location labels of the intermediate scans are interpolated by the turning points. We omit its detailed process. In the off-line phase, KNN is chosen to smooth the RSS vectors to construct the *FP-Low database*.

(2) *FP-High* Method. The on-line locating algorithm for FP-High is similar with Algorithm3. The unique difference is that we use

$$\overline{\mathbf{z}}_t = FPHigh(\overline{\mathbf{p}_t}) \tag{10}$$

to calculate $\overline{\mathbf{z}}_t$ and $\overline{\mathbf{z}}_{t-1}$. Similarly, we use KNN to construct the off-line *FP-High database*.

The limitations of other unsupervised methods have been given in Section. 1 and Section. 2, which are summarized as follow: methods exploiting floor map information [43][12][41] needs external floor-maps; methods exploiting organic landmarks[47][33] needs external sensing methods; WiFi-SLAM related methods [10][25][7][22] are proved to only work well for long repetitive circular traces. Based on above reasons, we compare TM-based methods with universal, independent of specific environment, real-time online locating methods, i.e., FP-based methods and inertial navigation.

4.2 Accuracy

In order to further clearly show the localization features of FP-based methods and TM-based methods, we calculate locating accuracy in 2 schemes, i.e., the start points of testing traces are assigned to correct absolute positions in the map (the *scheme goodstart*) and wrong absolute positions (the *scheme badstart*). We give the conclusion in advance, FP-based methods aim to correct the absolute positions while TM-based methods aim to correct the relative positions (or can be considered as the contours of the testing traces).

Fig. 3(a) and Fig. 3(c) give the locating results and accuracy in the *scheme goodstart*. The x and y axis of Fig. 3(a) represent 2-D coordinate of the simulation map ($200 * 200pix^2$). Generally speaking, all 4 methods guarantee better locating accuracy than inertial navigation. Under the same dimension, i.e., TMS is compared with FPHigh while TMM is compared with FPLow, FP-based methods work better than TM-based methods, as there is no noise in training $\{\mathbf{P}^f\}$ for FP-based methods while $N(0, V_u = 1)$ Gaussian noise in $\{\mathbf{U}^f\}$ for TM-based methods.

Fig. 3(b) and Fig. 3(d) give the locating results and accuracy in the *scheme badstart*. Two FP-based methods aim to correct the absolute positions of testing traces while two TM-based methods correct the relative positions of testing traces as the information of absolute positions are not provided in the off-line phase for TM models.

4.3 Self-learning Ability

For ideal localization systems, we expect its locating accuracy can improve when the size of training data increase. So we calculate the locating errors of 4 methods, i.e., 2 FP-based methods and 2 TM-based methods, under different sizes of training traces, i.e., M_f varying from 1 to 101. The numbers and distributions of APs and testing traces remain the same for different M_f, which cause the locating errors of inertial navigation stay stable. Fig. 4(a) shows corresponding prediction errors. We can find two FP-based methods and two TM-based methods all own good self-learning abilities.

4.4 Effect by Size of Access Points (APs)

Fig. 4(b) shows the average locating errors with the size of APs varying from 1 to 101. When the size of APs becomes larger, the locating errors of 4 methods become smaller and then approach stability. Under the same dimension, FP-based methods own better locating performance than that of TM-based methods, as the training data of TM-based methods own more noise, i.e., $N(0, V_u = 0.5)$ in $\{\mathbf{U}^f\}$ and $N(0, V_z = 5)$ in $\{\mathbf{Z}^f\}$, than that of the FP-based methods, i.e., $N(0, V_z = 5)$ in $\{\mathbf{Z}^f\}$ and no noise in $\{\mathbf{P}^f\}$. The result also reveals to some extent, the localization accuracy of WiFi-based methods cannot go beyond one threshold accuracy due to the noise exitance of signal measurements.

4.5 Effect by Movement Noise

Fig. 4(c) gives the average locating errors with V_u varying from 0 to 1. When $V_u < 0.1$, both low dimensional methods, i.e., FPLow and TMM, works worse than inertial navigation and both high dimensional methods, i.e., FPHigh and TMS. The reason is compared with the low dimensional methods, both high dimensional methods rely more on the locating results of inertial navigation as they use linear relationship $\mathbf{H}_t$ to approximate the non-linear relationship between $\mathbf{z}_t$ and $\mathbf{p}_t$, in which much effective information in signal observation is lost. When $V_u > 0.1$, with V_u increases, the locating errors of 4 methods increase. Generally speaking, when V_u is not small, which is the actual situation of localization tasks, FPLow works better than TMM than FPHigh than TMS, which proves TM-based methods own comparable locating accuracy with FP-based methods.

4.6 Effect by Signal Noise

For V_z varying from 0 to 100, the average locating errors of 4 methods are calculated in Fig. 4(d). We can see when V_z becomes bigger, the locating accuracy of 4 methods becomes worse. When $N(0, V_z = 100)$ Gaussian noise is added in signal measurements, the locating accuracy of TMM, FPHigh and TMS degrade to that of inertial navigation while the locating accuracy of FPLow works worse than that of inertial navigation. We try to give some explanation. Both high dimensional methods use linear relationship $\mathbf{H}_t$ to approximate the non-linear relationship, which causes the effective information loss in signal measurements, so their locating results are more similar to that of inertial navigation. So even when signal measurements are totally unreliable, the locating accuracy of FPHigh and TMS just degrades to that of inertial navigation. However, for two low dimensional methods, i.e., FPLow and TMM, as they extract more effective information from signal measurements, when V_z increases, their locating accuracy deteriorates sharply. When signal measurements are generated by white noise, which means they totally lose the sensitivity to physical positions, as the value range of $\hat{\mathbf{p}_t}$ for FPLow is bigger than that of $\hat{\mathbf{u}_t}$ for TMM, the locating performance of FPLow is worse than that of TMM.

4.7 Effect by e

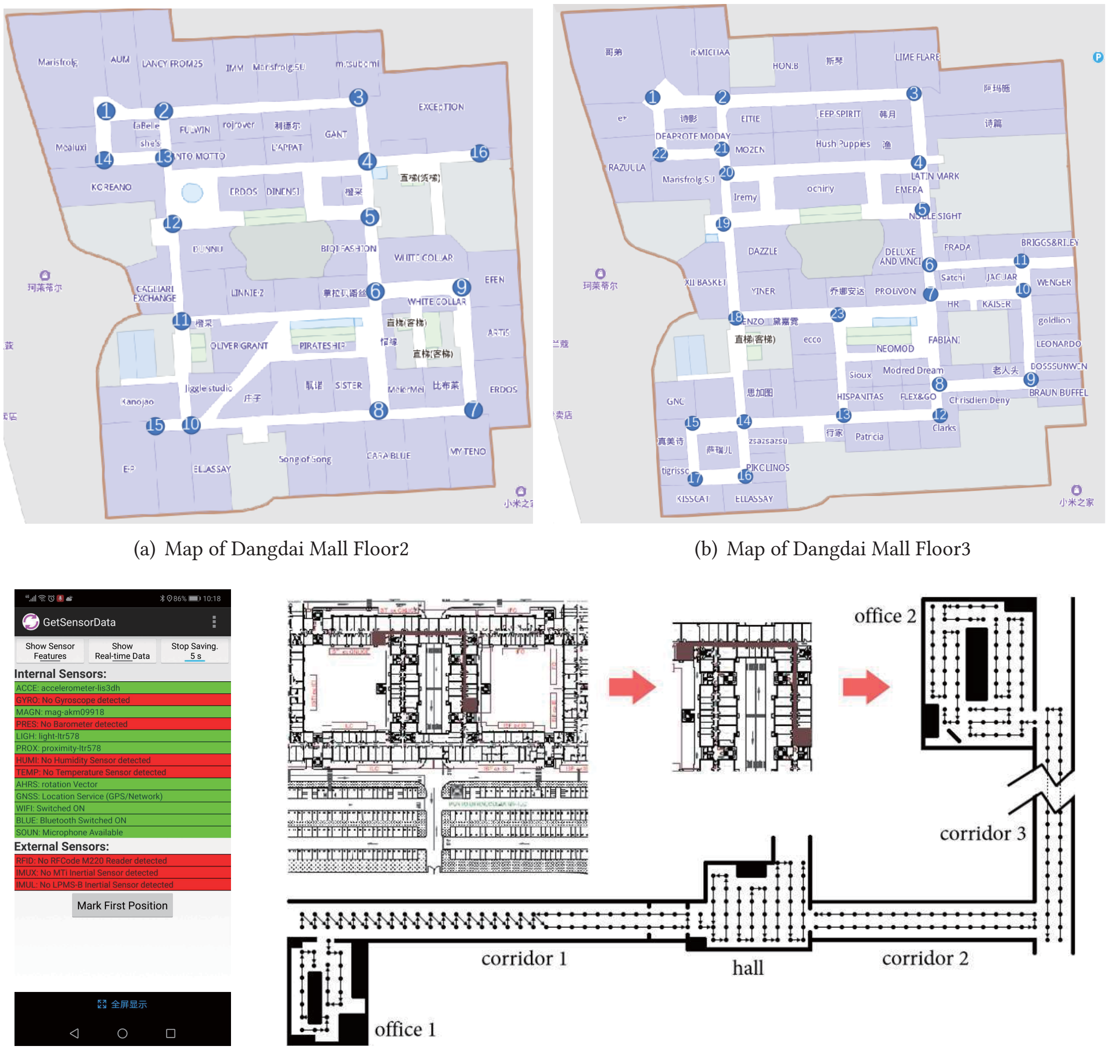
We calculate the average locating errors with e varying from 0 to 1. For each e and $\mathbf{Z}^f$ or $\mathbf{Z}$, we randomly choose $e * L*$(column length of $\mathbf{Z}^f$ or $\mathbf{Z}$) positions and use the minimal signal value to fill the positions. When $e = 1$, 100% signal data cannot be observed. Fig. 4(e) shows the result. We can find Fig. 4(e) is greatly similar to Fig. 4(d), which means data missing can be regarded as one type of special noise.

4.8 Effect by r

For all 4 methods, the ratio $r = log(\sigma_q^2/\sigma_r^2)$ has great effect on their locating accuracy. The locating results for different r are shown in Fig. 4(f). $r < 0$ means we believe signal measurements more than inertial navigation while $r > 0$ means we believe inertial navigation more than signal measurements. We can find under current parameters, $\hat{\mathbf{p}_t}$ estimated by FPLow and $\overline{\mathbf{z}_t}$ and $\mathbf{H}_t$ approximated by FPHigh and TMS are more reliable than inertial navigation information. However, $\hat{\mathbf{u}}_{t-1}$ estimated by TMM is not so reliable that $r = 0$ is the best choice for TMM.

4.9 Effect by K

We use K to reflect the smoothness of returned value from 4 off-line databases. For K varying from 1 to 25, the average locating errors of 4 methods are calculated in Fig. 4(g). It can be found for two FP-based methods and two TM-based methods, when K increases, the locating accuracy increases.



(a) Map of Dangdai Mall Floor2

(b) Map of Dangdai Mall Floor3

GetSensorData

Show Sensor Features | Show Real-time Data | Stop Saving. 5 s

Internal Sensors:

ACCE: accelerometer-lis3dh
GYRO: No Gyroscope detected
MAGN: mag-akm09918
PRES: No Barometer detected
LIGH: light-ltr578
PROX: proximity-ltr578
HUMI: No Humidity Sensor detected
TEMP: No Temperature Sensor detected
AHRS: rotation Vector
GNSS: Location Service (GPS/Network)
WIFI: Switched ON
BLUE: Bluetooth Switched ON
SOUN: Microphone Available

External Sensors:

RFID: No RFCode M220 Reader detected
IMUX: No MTi Inertial Sensor detected
IMUL: No LPMS-B Inertial Sensor detected

Mark First Position

全屏显示

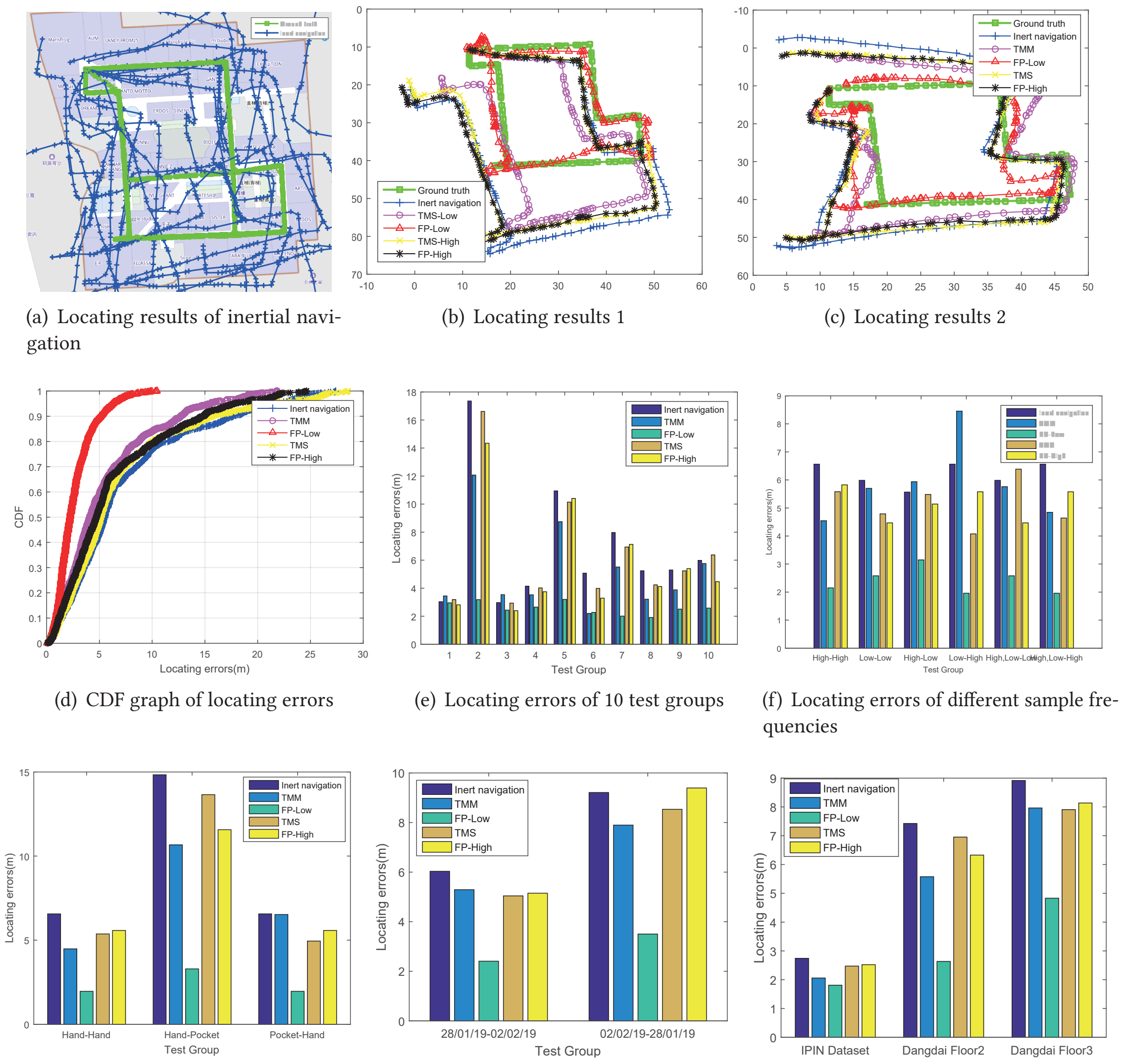
(c) Snapshot of collection application

(d) Map of Ipin dateset

Fig. 5. Maps of experiment environments

5 EXPERIMENT EVALUATION

Simulations have proved the effectiveness of TM-based methods in open environments. In the section, we evaluate the locating performance of TM-based methods in two shopping mall environments and one corridor-like environment.



(a) Locating results of inertial navigation

(b) Locating results 1

(c) Locating results 2

(d) CDF graph of locating errors

(e) Locating errors of 10 test groups

(f) Locating errors of different sample frequencies

(g) Locating errors of different postures

(h) Locating errors of different dates

(i) Locating errors of different environments

Fig. 6. Experiment evaluation

5.1 Experiment Settings

5.1.1 Environment Settings. We use three datasets to test the locating performances of the proposed TM-based methods.

(1) *Dangdai Floor2 dataset*: 10 IMU+RSS sequences were collected by the open source application [1][39][20], whose snapshot is shown in Fig. 5(c), in the 2nd floor of Dangdai Mall (floor space is 22400 m^2) in Haidian district, Beijing, China [44]. Its map can be referred to Fig. 5(a). It is a half-open environment. In the

Table 2. Information of collected sequences in the 2nd floor of Dangdai Mall

Traces (marked by key positions in Fig. 5(a))	WiFi Sample Frequency(Hz)	Walking postures	Date
[1, 2, 3, 4, 6, 9, 7, 8, 6, 11, 10, 15, 10, 8, 6, 11, 12, 13, 14, 1]	69.5	in hand	28/01/2019
[1, 2, 3, 4, 6, 9, 7, 8, 6, 11, 10, 15, 8, 6, 11, 12, 13, 14, 1]	69.5	in pocket	28/01/2019
[1, 3, 6, 9, 7, 15, 10, 13, 14, 1]	69.5	in hand	02/02/2019
[1, 14, 13, 10, 15, 7, 9, 6, 3, 1]	69.5	in hand	02/02/2019
[1, 13, 10, 15, 7, 9, 6, 3, 1]	69.5	in hand	02/02/2019
[1, 3, 6, 9, 7, 15, 10, 13, 14, 1]	69.5	in hand	02/02/2019
[1, 3, 6, 9, 7, 15, 10, 13, 14, 1]	69.5	in hand	02/02/2019
[1, 14, 13, 10, 15, 7, 9, 6, 3, 1]	69.5	in hand	02/02/2019
[1, 2, 3, 4, 5, 6, 9, 7, 8, 10, 15, 10, 11, 12, 13, 14, 1]	11.5	in hand	28/01/2019
[1, 2, 3, 4, 5, 6, 9, 7, 8, 10, 15, 10, 11, 12, 13, 14, 1]	11.5	in pocket	28/01/2019

Table 3. Information of collected sequences in the 3rd floor of Dangdai Mall

Traces (marked by key positions in Fig. 5(b))	WiFi Sample Frequency(Hz)	Walking postures	Date
[1, 2, 3, 4, 12, 13, 14, 15, 17, 14, 18, 23, 13, 12, 4, 20, 22, 1]	69.5	in hand	28/01/2019
[1, 2, 3, 4, 12, 13, 14, 15, 17, 14, 18, 23, 13, 12, 5, 4, 20, 22, 1	69.5	in pocket	28/01/2019
[1, 22, 21, 14, 15, 17, 14, 12, 3, 1]	69.5	in hand	13/02/2019
[1, 3, 12, 14, 17, 15, 14, 21, 22, 1]	69.5	in hand	13/02/2019
[1, 3, 12, 14, 15, 17, 14, 21, 22, 1]	69.5	in hand	13/02/2019
[1, 22, 21, 14, 17, 14, 12, 3, 1]	69.5	in hand	13/02/2019
[1, 22, 21, 19, 5, 12, 14, 17, 15, 14, 21, 22]	69.5	in hand	13/02/2019
[1, 2, 3, 12, 13, 14, 17, 14, 18, 23, 18, 19, 20, 21, 22, 1]	11.5	in hand	28/01/2019

collection process, we click the bottom button of the application, which is shown in Fig. 5(c), to mark the key positions and calculate the ground truth traces by aligning the marked positions of collected data with the key positions in the map, which are marked in Fig. 5(a) and Fig. 5(b). The number of effective APs of the environment is 482. Each sequence is long enough to cover most areas of the environment and each sequence is collected in diversed ways, e.g., by different collection postures and sample frequencies. More information of the collected sequences can be referred to Table 2.

(2) *Dangdai Floor3 dataset*: 8 IMU+RSS sequences were collected by the open source application [1][39][20] in the 3rd floor of Dangdai Mall (floor space is 22400 m^2) in Haidian district, Beijing, China [44]. Its map can be referred to Fig. 5(b), which is also a half-open environment. The number of effective APs is 668. More information of the collected sequences can be referred to Table 3.

(3) *IPIN dataset*: The dataset *Geo-Magnetic field and WLAN dataset for indoor localisation from wristband and smartphone Data Set* [4] from *UCI* is used to calculate the locating errors. Its map can be referred to Fig. 5(d) [4], which is a corridor-like environment. The dataset consists of two groups of data of 16 paths in the same environment. Each consists of 16 RSS sequences from 127 APs together with 16 inertial sensor sequences.

5.1.2 *Algorithm Settings and Comparison Algorithms.*

We calculate $\{\mathbf{U}^f\}$ and $\{\mathbf{U}\}$ based on the *step counting based dead reckoning algorithm* [28]. The step size is set as a constant 0.8m. Other related parameters of TM-based algorithms are set as following for the three datasets.

(1) *Dangdai Floor2 dataset*: We randomly choose nine traces in Table 2 as training data and the remaining one trace as the testing data. We name the *ith testing group* whose testing trace is the *i*th trace in Table.

2. We calculate the statistical locating errors of ten test groups. We set $K = 3$, $\sigma_q^2 = 5$, $\sigma_r^2 = 1$ and $r = log(\sigma_q^2/\sigma_r^2) = 1.7$ in the experiments.

(2) *Dangdai Floor3 dataset*: We randomly choose three traces in Table 3 as the training data and the remaining one trace as the testing data to calculate the statistical locating errors. We set $K = 3$, $\sigma_q^2 = 5$, $\sigma_r^2 = 1$ and $r = log(\sigma_q^2/\sigma_r^2) = 1.7$.

(3) *IPIN Dataset*: We randomly choose one group (the number of traces is 16) as the training data and the remaining one group as the testing data to calculate the statistical locating errors. As the covered number of IPIN dataset in Fig. 5(d) is the smallest among the three datasets, we set $K = 1$, $\sigma_q^2 = 10$, $\sigma_r^2 = 1$ and $r = log(\sigma_q^2/\sigma_r^2) = 0$, which means we believe inertial navigation information more than WiFi signal measurements.

We choose inertial navigation and two FP-based methods, which has been introduced in Section. 4.1.3, as comparison algorithms.

5.2 Accuracy

Fig. 6(a) shows the ground truth and inertial navigation results of the ten traces of the Dangdai Floor2 dataset. It can be found that the noise of inertial navigation including orientation noise and step size noise greatly impact the navigation accuracy. We extract the expected transitional knowledge, i.e., TM, from these inertial navigation and their corresponding RSS sequences.

Fig. 6(b) and Fig. 6(c) show two typical locating results based on the Dangdai Floor2 dataset. Fig. 6(b) corresponds to the result of 7th testing group and Fig. 6(c) is the 8th testing group. It can be found that both the TM-based methods and FP-based methods can correct errors of inertial navigation. Fig. 6(d) is the statistical CDF result of the 10 test groups. It is very similar to Fig. 3(c) and proves the effectiveness of the TM-based methods in shopping mall environments. Fig. 6(e) shows the locating errors for the ten test groups. The locating errors of TMM are smaller than those of inertial navigation in most of time, except in the 1st and 3rd testing group.

For the 1st and 3rd testing group, the help from RSS measurements is very little, which can be reflected by the limited accuracy improvement of FP-Low method compared with that of inertial navigation. So it is acceptable that the locating accuracy of the TMM is slightly worse than that of inertial navigation. The locating errors of TMS for 10 testing groups are smaller than inertial navigation, which show that TMS works more stable, which is consistent with the simulation results.

5.3 Effect by Sample Frequency

The subsection tests the robustness of the TM-base methods in using different collection devices. Usually different mobile phones have different WiFi signal sampling frequency [45]. Fig. 6(f) shows the locating errors of TM-based methods under different sets of WiFi sample frequencies based on the Dangdai Floor2 dataset. The 1st group of columns reflect the locating errors when the sample frequencies of the training data and the testing data are both high, i.e., 69.5Hz, and the 2nd group of columns reflect the results when the sample frequencies are both low, i.e., 11.5Hz. We can find the two TM-based methods work well when the training and testing data have the same sampling frequency.

For the columns of 3rd and 4th group in Fig. 6(f), in which the training sample frequency is different from that of the testing one, the locating accuracy of TMM is worse than inertial navigation. The results can be well explained. TMM aims to use extracted expected $\hat{\mathbf{u}}_{t-1}$ indexed by $\mathbf{z}_{t-1}$ and $\mathbf{z}_t$ to correct the IMU prediction $\overline{\mathbf{u}}_{t-1}$, so when the testing frequency is different than the training one, which means the index values $\mathbf{z}_{t-1}$ and $\mathbf{z}_t$ cannot find very suitable matched values in TMM database, the locating performance of TMM deteriorates greatly. Compared with TMM, the locating performance of TMS is more robust.

One very effective method to overcome the above limits is to add little training data having the same frequency with the testing data to the TMM database. The 5th group of columns reflect the locating errors when the training data consists of most high-frequency data and little low-frequency data and the testing data is in low-frequency. The 6th group of column reflects the results when the training data consists of little high-frequency data and most low-frequency data and the testing data is in high-frequency. It can be found TMM works well in above two situations.

5.4 Effect by Collection Posture

The subsection tests the robustness of TM-base methods for different collection postures, i.e., holding the device in hand and placing the device in pocket. By analyses, the IMU noise in pocket is greater than that in hand. Fig. 6(g) shows locating errors under different sets of collection postures based on the Dangdai Floor2 dataset. The 1st group of columns in Fig. 6(g) reflect the locating errors when training data and testing data are both collected in hand. The 2nd group reflects the results when the training data is collected in hand and testing data is collected in pocket. And the 3rd group reflects the results when the training data is collected in pocket and testing data is collected in hand. For the first two groups, as the IMU noise of training data is not larger than the testing one, TMM and TMS work better than inertial navigation. For the last group, as the IMU noise of training data is larger than the testing one, the locating performance of TMM deteriorates slightly. TMS is rather robust to different postures.

5.5 Effect by Time Changing

This subsection summarizes the robustness of TM-based methods for signal changing overtime. Fig. 6(h) shows locating errors under different sets of dates based on the Dangdai Floor2 dataset. For the 1st group of columns, training data is collected in 28/01/2019 and testing date is collected in 02/02/2019. For the 2nd group, training data is collected in 02/02/2019 and testing data is collected in 28/01/2019. It can be found that the two TM-based methods are robust for data collected on different dates.

5.6 Effect by Environments

The subsection tests the robustness of TM-base methods for different environments. Fig. 6(i) shows locating errors based on the IPIN dataset, the Dangdai Floor2 dataset and the Dangdai Floor3 dataset. It can be found both TM-based and FP-based methods are robust for different environments. And the locating performance of FP-Low is better than that of TMM than that of FP-High than that of TMS than inertial navigation, which is consistent with the simulation results.

6 CONCLUSION

The paper proposed one light, on-line locating model *transitional model (TM)*, which captures the map relationship between two consecutive signal states $\{z_{t-1}, z_t\}$ and their intermediate one step motion u_{t-1}, based on massive fragmented and unlabeled synchronized WiFi RSS+IMU sequences. Two specific TM methods are introduced: (1) *Transitional model to predict motion from signal change (TMM)* off-line learns expectation map relationship $\hat{\mathbf{u}}_{t-1} = TMM(\mathbf{z}_{t-1}, \mathbf{z}_t)$ and estimates on-line target positions based on the TMM-KF algorithm. (2) *Transitional model in signal space (TMS)* off-line learns expectation map relationship $\overline{\mathbf{z}}_t = TMS(\mathbf{z}_{t-1}, \mathbf{u}_{t-1})$ and estimates on-line target positions based on the TMS-EKF algorithm. In particular, K-nearest neighbors (KNN) smoothing method is proposed to capture above two expectation map relationship. TM is the first unsupervised localization model that does not require any additional information, such as floor maps, additional sensing methods or long and circular trajectory requirements. Simulations and experiments show that the TM models show better locating accuracy than inertial navigation and comparable accuracy with the FP-based method. TM is even proved to own good self-learning ability which means its can work better and better when more and more on-line user sequences are acquired and strong robustness for different noise distribution, size of APs, sample frequencies of devices, collection postures of users and other related parameters.

In future work, there are several problems need to be addressed related to TM: (1) The TM model can be easily generalized to different types of signal measurements, e.g., WiFi, Bluetooth, light and pressure sensors and even camera pictures. And multi-sensor TM model is expected to be developed in the future. (2) TM only captures one step transitional relationship between time $t-1$ and t. Long-time transitional relationships, e.g., the map between $\{z_{t-N},\ldots,z_t\}$ and $\{u_{t-N},\ldots,u_{t-1}\}$, are expected to be constructed to improve locating accuracy. (3) TM can motivate related data mining works in the field of persuasive computing. Even though the noise of raw signal is great, some expectation map relationships can be extracted by aggregate a large number of information. One direct follow-up work is to construct the indoor map and fingerprint map based on synchronized noisy multi-sensor data.

As TM and its follow-up works are independent of specific environment and expert knowledge, large-scale indoor localization prototype systems are also expected in the future.

REFERENCES

[1] Getsensordata. https://lopsi.weebly.com/downloads.html, 2013.

[2] M. Atia, A. Noureldin, and M. Korenberg. Dynamic Online-Calibrated Radio Maps for Indoor Positioning in Wireless Local Area Networks. *IEEE Transactions on Mobile Computing*, 12(9):1774–1787, Sept. 2013.

[3] P. Bahl and V. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *IEEE INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, volume 2, pages 775–784 vol.2, 2000.

[4] P. Barsocchi, A. Crivello, D. L. Rosa, and F. Palumbo. A multisource and multivariate dataset for indoor localization methods based on wlan and geo-magnetic field fingerprinting. In *International Conference on Indoor Positioning and Indoor Navigation*, 2016.

[5] A. Bernardos, J. Casar, and P. Tarrio. Real time calibration for RSS indoor positioning systems. In *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7, 2010.

[6] K. R. Britting. *Inertial navigation systems analysis /*. Wiley-Interscience„ 1971.

[7] L. Bruno and P. Robertson. WiSLAM: Improving FootSLAM with WiFi. In *2011 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–10, Sept. 2011.

[8] Y. H. Chung, R. FÃďre, and S. Grosskopf. Productivity and undesirable outputs: A directional distance function approach. *Microeconomics*, 51(3):229–240, 1997.

[9] I. Constandache, R. R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In *2010 Proceedings IEEE INFOCOM*, pages 1–9, March 2010.

[10] B. Ferris, D. Fox, and N. Lawrence. Wifi-slam using gaussian process latent variable models. In *International Joint Conference on Artifical Intelligence*, pages 2480–2485, 2007.

[11] J. Fink and V. Kumar. Online methods for radio signal mapping with mobile robots. In *2010 IEEE International Conference on Robotics and Automation*, pages 1940–1945, May 2010.

[12] C. Gao and R. Harle. Semi-Automated Signal Surveying Using Smartphones and Floorplans. *IEEE Transactions on Mobile Computing*, 17(8):1952–1965, Aug. 2018.

[13] Y. Gu, Y. Chen, J. Liu, and X. Jiang. Semi-supervised deep extreme learning machine for wi-fi based localization. *Neurocomputing*, 166:282 – 293, 2015.

[14] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara. Localization algorithms of Wireless Sensor Networks: a survey. *Telecommunication Systems*, 52(4):2419–2436, Apr. 2013.

[15] I. Haque and C. Assi. Profiling-Based Indoor Localization Schemes. *IEEE Systems Journal*, Early Access Online, 2013.

[16] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and 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*, UbiComp '14, pages 147–158, New York, NY, USA, 2014. ACM.

[17] A. K. M. M. Hossain and W.-S. Soh. A survey of calibration-free indoor positioning systems. *Computer Communications*, 66:1–13, July 2015.

[18] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal. Efficient, generalized indoor WiFi GraphSLAM. In *2011 IEEE International Conference on Robotics and Automation*, pages 1038–1043, May 2011.

[19] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal. Efficient, generalized indoor wifi graphslam. In *IEEE International Conference on Robotics and Automation*, pages 1038–1043, 2011.

[20] A. R. Jimenez, F. Zampella, and F. Seco. Light-matching: A new signal of opportunity for pedestrian indoor navigation. In *International Conference on Indoor Positioning & Indoor Navigation*, 2014.

[21] S. Jung, B. Moon, and D. Han. Unsupervised Learning for Crowdsourced Indoor Localization in Wireless Networks. *IEEE Transactions on Mobile Computing*, 15(11):2892–2906, Nov. 2016.

[22] S. Kumar, S. Gil, D. Katabi, and D. Rus. Accurate Indoor Localization with Zero Start-up Cost. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 483–494, New York, NY, USA, 2014. ACM.

[23] S. T. Liu. Permutation methods: A distance function approach. *Technometrics*, 44(3):289–290, 2002.

[24] K. Majeed, S. Sorour, T. Al-Naffouri, and S. Valaee. Indoor localization and radio map estimation using unsupervised manifold alignment with geometry perturbation. 15:1–1, 01 2015.

[25] P. Mirowski, T. K. Ho, S. Yi, and M. MacDonald. SignalSLAM: Simultaneous localization and mapping with mixed WiFi, Bluetooth, LTE and magnetic signals. In *International Conference on Indoor Positioning and Indoor Navigation*, pages 1–10, Oct. 2013.

[26] L. Ni, Y. Liu, Y. C. Lau, and A. Patil. LANDMARC: indoor location sensing using active RFID. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003)*, pages 407–415, 2003.

[27] J. J. Pan, S. J. Pan, J. Yin, L. M. Ni, and Q. Yang. Tracking Mobile Users in Wireless Networks via Semi-Supervised Colocalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):587–600, Mar. 2012.

[28] M. S. Pan and H. W. Lin. A step counting algorithm for smartphone users: Design and implementation. *IEEE Sensors Journal*, 15(4):2296–2305, 2015.

[29] J.-g. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an organic indoor location system. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 271–284, New York, NY, USA, 2010. ACM.

[30] L. Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[31] T. Pulkkinen, T. Roos, and P. Myllymäki. Semi-supervised learning for wlan positioning. In T. Honkela, W. Duch, M. Girolami, and S. Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 355–362, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[32] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang. Walkie-markie: Indoor pathway mapping made easy. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, nsdi'13, pages 85–98, Berkeley, CA, USA, 2013. USENIX Association.

[33] I. Skog, J. Nilsson, and P. HÃďndel. Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. In *2010 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–6, Sept 2010.

[34] T. L. Song and J. L. Speyer. A stochastic analysis of a modified gain extended kalman filter with applications to estimation with bearings only measurements. In *Decision and Control, 1983. the IEEE Conference on*, pages 1291–1296, 1983.

[35] B. Sujak, D. Kumar Ghodgaonkar, B. Ali, and S. Khatun. Indoor propagation channel models for WLAN 802.11b at 2.4 GHz ISM band. In *Asia-Pacific Conference on Applied Electromagnetics, 2005. APACE 2005*, pages 5 pp.–, 2005.

[36] T. Sun, Y. Wang, D. Li, Z. Gu, and J. Xu. WCS: weighted component stitching for sparse network localization. *IEEE/ACM Trans. Netw.*, 26(5):2242–2253, 2018.

[37] N. Swangmuang and P. Krishnamurthy. Location fingerprint analyses toward efficient indoor positioning. In *IEEE International Conference on Pervasive Computing & Communications*, pages 100–109, 2008.

[38] J. Torres-Sospedra, R. Montoliu, A. MartÃŋnez-UsÃş, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta. UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 261–270, Oct. 2014.

[39] J. Torressospedra, A. R. JimÃĺnez, S. Knauth, A. Moreira, Y. Beer, T. Fetzer, V. C. Ta, R. Montoliu, F. Seco, and G. M. Mendozasilva. The smartphone-based offline indoor location competition at ipin 2016: Analysis and future work. *Sensors*, 17(3):557, 2017.

[40] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 197–210, New York, NY, USA, 2012. ACM.

[41] C. Wu, Z. Yang, and Y. Liu. Smartphones Based Crowdsourcing for Indoor Localization. *IEEE Transactions on Mobile Computing*, 14(2):444–457, Feb. 2015.

[42] Z. Xiao, H. Wen, A. Markham, and N. Trigoni. Lightweight Map Matching for Indoor Localisation Using Conditional Random Fields. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, IPSN '14, pages 131–142, Piscataway, NJ, USA, 2014. IEEE Press.

[43] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space:wireless indoor localization with little human intervention. In *International Conference on Mobile Computing and NETWORKING*, pages 269–280, 2012.

[44] X. Ye, S. Huang, Y. Wang, and D. Li. Simulation code of tm and dangdai dataset. https://github.com/coldsnowleaf/TM, 2019.

[45] X. Ye, Y. Wang, W. Hu, L. Song, Z. Gu, and D. Li. Warpmap: Accurate and efficient indoor location by dynamic warping in sequence-type radio-map. In *13th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON 2016, London, United Kingdom, June 27-30, 2016*, pages 1–9, 2016.

[46] J. Yin, Q. Yang, and L. Ni. Adaptive Temporal Radio Maps for Indoor Location Estimation. In *Third IEEE International Conference on Pervasive Computing and Communications, 2005. PerCom 2005*, pages 85–94, 2005.

[47] F. Zampella, A. Jimenez R, and F. Seco. Robust indoor positioning fusing PDR and RF technologies: The RFID and UWB case. In *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10, Oct. 2013.

[48] M. Zhou, Y. Tang, W. Nie, L. Xie, and X. Yang. Grassma: Graph-based semi-supervised manifold alignment for indoor wlan localization. *IEEE Sensors Journal*, 17(21):7086–7095, Nov 2017.

Received November 2018; revised February 2019; accepted April 2019