Transition Model–driven Unsupervised Localization Framework Based on Crowd-sensed Trajectory Data

XINGFA SHEN, CHUANG LI, and WEIJIE CHEN, Hangzhou Dianzi University, China
YONGCAI WANG, Renmin University of China, China
QUANBO GE, Nanjing University of Information Science and Technology, China

The rapid popularization of mobile devices makes it more convenient and cost-efficient to collect synchronized WiFi received signal strength (RSS) and inertial measurement unit sequences by crowdsensing. The *transition model* has proven to be a promising unsupervised localization approach that captures the transition relationship between the change of RSS signal space and the change of physical space, alleviating the need of extra knowledge for creating radio map. However, it faces two essential challenges in real-world deployments. First, model coverage affects its locating performance, because a specific transition model only represents its local space. Second, the instability of RSS leads to a conflicting relationship between changes of two spaces because of the complex environment and the heterogeneous type of devices. To address these challenges, we propose Lightgbm-CTMM, a novel unsupervised localization framework. First, a clustering method is adopted to capture the expected relationship to ensure robust coverage. Second, direction filter is employed to guarantee that the change in signal space corresponds to the change in physical space. The feasibility and effectiveness of Lightgbm-CTMM are evaluated by extensive experiments, and the locating performance of Lightgbm-CTMM is better than that of conventional approaches. Moreover, Lightgbm-CTMM reduces the work on quality assessment of trajectories.

CCS Concepts: • **Computer systems organization → Embedded systems**; • **Networks → Location based services**; • **Human-centered computing → Mobile phones;**

Additional Key Words and Phrases: RSS, IMU, unsupervised learning, transformation relationship, signal space, physical space

ACM Reference format:
Xingfa Shen, Chuang Li, Weijie Chen, Yongcai Wang, and Quanbo Ge. 2022. Transition Model–driven Unsupervised Localization Framework Based on Crowd-sensed Trajectory Data. *ACM Trans. Sen. Netw.* 18, 2, Article 26 (January 2022), 21 pages.
https://doi.org/10.1145/3499425

This work was supported by the Natural Science Foundation under Grant No. 61672198 and No. 62072121.
Authors' addresses: X. Shen, C. Li, and W. Chen, Hangzhou Dianzi University, Xiasha Higher Education Zone, Hangzhou, Zhejiang, 310018, P.R. China; emails: {shenxf, li_chuang}@hdu.edu.cn, 540496406@qq.com; Y. Wang, Renmin University of China, No. 59 Zhongguancun Street, Haidian District, Beijing, 100872, P.R. China; email: ycw@ruc.edu.cn; Q. Ge (corresponding author), Nanjing University of Information Science and Technology, No. 219, Ningliu Road, Nanjing, Jiangsu, 210044, P.R. China; email: quanboge@163.com.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Association for Computing Machinery.
1550-4859/2022/01-ART26 $15.00
https://doi.org/10.1145/3499425

1 INTRODUCTION

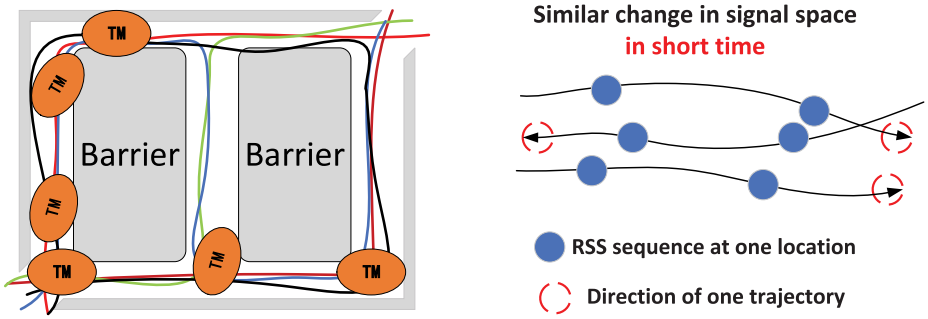
Indoor location information is critical in numerous industrial and commercial applications, involving decades of research. The **Pedestrian Dead-Reckoning (PDR)** [11], which relies on **inertial measurement unit (IMU)** to track a user by continuously estimating displacement from a known location, is effective for locating indoors without the need for infrastructure assistance. But the noise of inertial sensor data causes track drift [17, 21]. To solve the problem of PDR, many studies combine other locating information with PDR algorithm to achieve accurate localization. One of the most effective locating methods is WiFi fingerprinting. The fingerprint localization based on WiFi [2, 9, 14, 35] has attracted wide attention due to its characteristics such as widespread infrastructure, acceptable locating accuracy, and so on. The key step is to build a model that relates the signal signatures in the signal space to the locations in the physical space. However, most existing radio-based solutions require a process of site survey to build a radio map, which incurs great labour and time expenditures.

The rapid popularization of mobile and wearable devices makes the collection of massive trajectory data more convenient and cost-efficient. Therefore, unsupervised learning for radio map is intensively studied based on massive and unlabeled trajectories that include **received signal strength (RSS)** and IMU. Unsupervised learning for indoor localization based on crowdsensing trajectories can be done in three ways: (1) building the radio map by using the floor-map information to constraint user mobility [12, 30, 33], (2) extracting the landmark information by additional sensors or floor-map to infer the location of the trajectory between the landmarks [4, 22], and (3) exploiting WiFi-SLAM [6], which uses Gaussian process latent variable models to connect RSS fingerprints and human movements. The above indoor location techniques, however, have limitations in that they require extra information. The first way needs floor-map information, but the digital floor-map is not available for many buildings or too expensive to obtain. The second way always requires additional sensors or floor-map information, but the cost of additional sensors is so high that it may be not available for diverse user devices in crowdsensing scenarios. The third way entails long and smooth trajectories for loop closure detection to keep solution accuracy.

The above methods are not suitable to process the RSS+IMU sequences that are massive, crowdsensed, and unlabeled, because the trajectories normally do not have additional knowledge. Therefore, the **Transition Model (TM)** [34] is proposed to address the problems of unlabeled training data and lacking additional knowledge. The TM is the transition model of local space, which establishes the relationship between the spatial changes of RSS signal and physical location, shown as the relationship between $\{RSS_{t-1}, RSS_t\}$ and $\{IMU_{t-1}\}$. In local space, a transition relationship is one that occurs over a short period of time, where the IMU_{t-1} represents the motion with a short distance. The TM realizes location based on traditional locating methods and corrects the relative position of traditional locating results. The **Transition Model to Predict Motion by Signal Change (TMM)** is a transition model that predicts motion by signal change so as to capture relationship $\hat{IMU}_{t-1} = TMM(RSS_{t-1}, RSS_t)$.

There are two problems that affect the locating performance in traditional transition model:

(1) The insufficient robustness of the spatial coverage of the model. Great locating performance requires a thorough coverage of the space by TMMs, because each TMM represents only one local space. Crowdsensing trajectories are unlabeled, and the construction of TMM is random. Therefore, the coverage of TMMs in space is not robust. The problem is shown in Figure 1(a), and we call this problem the poor robustness of the spatial coverage of the model.
(2) The discordance between signal space change and physical space change. The RSS is unstable in the actual environment because of the complex indoor environment and the heterogeneous types of intelligent portable equipments [26, 32]. The instability of RSS makes



(a) Poor robustness of the spatial coverage of the model

(b) Discordance between signal space change and physical space change

Fig. 1. Problems.

the relationship between the change of signal space and physical space inaccurate. Similar signal space changes may correspond to changes in physical space in different directions. The problem is shown in Figure 1(b), and we call this problem the discordance between signal space change and physical space change. A similar problem has also been found in Reference [16].

To solve the aforementioned problems, we propose the unsupervised localization framework Lightgbm-CTMM from massive and unlabeled trajectories in this article. The framework is mainly divided into two stages: offline training and online locating. In the offline stage, the similar relationship triples $\{RSS_{t-1}, IMU_{t-1}, RSS_t\}$ in local space are searched and divided into the same cluster by clustering the change in signal space $\{RSS_{t-1}, RSS_t\}$ to ensure robust distribution. Then the results of each cluster are grouped according to $\{IMU_{t-1}\}$, and the more rational CTMMs are established to solve the problem of the discordance between signal space change and physical space change. To build Lightgbm-CTMM, the Lightgbm algorithm [15] is used to train the model based on all triples that have their matching CTMMs. In the online stage, Lightgbm-CTMM is used to track mobile targets online based on the Kalman Filter, where the direction filter is used to solve the problem of the discordance between signal space change and physical space change.

The Lightgbm-CTMM is a model for reducing the influence of inertial sensor noise on PDR algorithm. To some extent, this model replaces other high-cost locating methods, like fingerprint map and floor map, as auxiliary locating information. The aim of traditional unsupervised localization approaches is to build fingerprint map or location model with low cost to achieve absolute position locating, but these ways often require additional prior knowledge or have some constraints. The Lightgbm-CTMM model is used as the auxiliary locating information in the PDR algorithm to decrease the influence of inertial sensor noise. Besides, Lightgbm-CTMM can only be built with a large number of unlabeled crowdsourced trajectories. As a result, the cost is lower compared with traditional unsupervised localization approaches.

The main contributions are summarized as follows:

(1) To solve the problem of the poor robustness of the spatial coverage of the model, clustering for the transition relation of signal space is proposed instead of establishing transition relation randomly. The more rational transition model CTMM is established with robust distribution and robust fraction of coverage.
(2) To solve the problem of discordance between signal space change and physical space change, a direction filter is proposed to guarantee the consistency between signal space transition

and physical space transition, which replaces the previous work of evaluating the quality of trajectories.
(3) Lightgbm's classification model improves the accuracy of finding the current matching CTMM in the online locating stage.
(4) Lightgbm-CTMM reduces the workload of evaluating the quality of training trajectories and mobile targets' trajectories.

The remaining sections are organized as follows. Section 2 introduces related work. Section 3 is the description of Lightgbm-CTMM framework. The construction of Lightgbm-CTMM is introduced in Section 4. Section 5 presents the online locating of Lightgbm-CTMM. Simulation and experiment evaluation are conducted in Sections 6 and 7. Section 8 is the conclusion.

2 RELATED WORK

We focus on the unsupervised learning for indoor location based on massive and unlabeled trajectories, which can be roughly classified into two categories.

2.1 Unsupervised Localization for Relative Position

Unsupervised localization for relative position means locating by scheme to correct the relative positions of traditional locating results, and the specific location is given by traditional methods. The TM [34] is the first unsupervised localization model that does not require any additional information, such as floor maps, additional sensing methods, or trajectories with special requirements. The Transition Model captures the relationship between two consecutive signal states $\{RSS_{t-1}, RSS_t\}$ and their intermediate motion $\hat{IMU}_{t-1}$ to improve the locating accuracy of traditional locating method.

2.2 Unsupervised Localization for Absolute Position

Unsupervised localization for absolute position means locating by scheme to obtain specific locations in the environment. Its primary task is the construction of the radio map.

2.2.1 Unsupervised Learning Using Floor-map Information. The WILL [31] proposes that to build the radio map, the "virtual rooms" obtained by different room-based features and connectivity of rooms calculated by pedestrian trajectory can be used to map to the physical rooms. The HMM is modeled for trajectories, and a hybrid global-local optimization scheme is proposed to determine the optimal position of the fingerprint sequence on the indoor map [13]. A method based on Huffman transform and Harris corner detection is proposed, in which the unlabeled trajectory data is matched with all the possible trajectories on the floor map to determine its possible location [19]. LIFS [33] proposes that the multi-dimensional scaling algorithm is used to transform the floor map and trajectory into the Stress-free map and to match two Stress-free maps to build the radio map. Zee [24] fuses user traces with a floor plan using a particle filter and WiFi to help locate traces. PiLoc [29] merges walkinging segments and signal strength information to derive a map of walking paths.

2.2.2 Unsupervised Learning by Organic Landmarks. UnLoc [28] locates traces to a floor plan via landmarks found on the indoor map and some specific locations. Additional sensors are used to detect the organic landmark at the same time as the WiFi fingerprint measurement, and the position between the landmarks is inferred by the track estimation scheme, resulting in the WiFi fingerprint point position [22]. A preliminary radio map is constructed by the location information of the mobile payment, and a considerable amount of trajectory data is accumulated to refine the radio map [1].

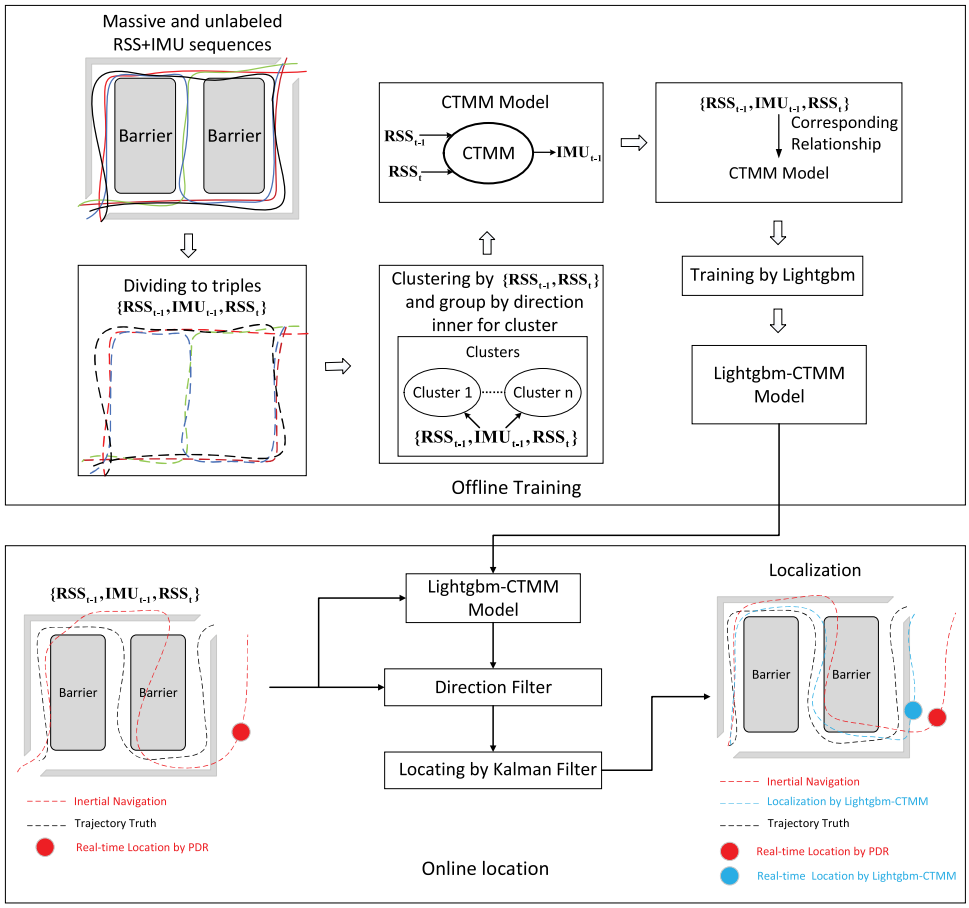


Fig. 2. Framework of Lightgbm-CTMM.

2.2.3 Unsupervised Learning by WiFi-SLAM. WiFi-SLAM [6] models the signal strength distribution and simultaneously recovers the trajectories. It needs some labeled data to train the initial values for some model parameters. By using a Gaussian process, an online model of the signal strength map can be constructed, which in turn can be used to provide a current maximum likelihood estimate of the source location [7]. SignalSLAM [20] is proposed on the basis of hybrid WiFi, Bluetooth, LTE, and geomagnetic signals. WiFi-GraphSLAM [10] also recovers and corrects user traces using WiFi signals by incorporating inertial sensors.

3 SCHEME DESCRIPTION

3.1 Framework of Lightgbm-CTMM Scheme

In this section, we introduce the framework of the Lightgbm-CTMM, as shown in Figure 2. The Lightgbm-CTMM framework is mainly divided into two stages: offline training stage and online locating stage.

The work in the offline training stage is as follows: (1) preprocessing for massive and unlabeled trajectories to get the data in triple format $\{RSS_{t-1}, IMU_{t-1}, RSS_t\}$, (2) clustering for triples based on the change in signal space $\{RSS_{t-1}, RSS_t\}$ and grouping based on direction information of $\{IMU_{t-1}\}$

Table 1. List of Notations and Explanations

Notations	Explanations	Notations	Explanations
S	unlabeled sequences of RSS+IMU	s^i	ith trajectory
Z^i	RSS data of ith trajectory	U^i	IMU data of ith trajectory
z_t^i	RSS of ith trajectory at record location of time t	$\overrightarrow{u}_t^i$	inertial vector of ith trajectory between time t and $t+1$
g	local transition relation triple $\{z_{t-1}, \vec{u}_{t-1}, z_t\}$	T	all triples set
rq	confidence coefficient in Kalman Filter	K_t	Kalman gain
$\bar{u}_{t-1}$	estimated inertial vector	$\hat{u}_{t-1}$	inertial vector from Lightgbm-CTMM
Q_t	process noise uncertainty	R_t	measurement uncertainty

in every cluster, (3) creating a CTMM based on averaging triples from the same group, and (4) using Lightgbm to obtain the relationship between all triples and CTMMs so as to build Lightgbm-CTMM.

The work in the online locating stage is as follows: (1) getting the observing motion $\hat{IMU}$ with the current triple of mobile target by Lightgbm-CTMM and direction filter and (2) locating by Kalman Filter using the observing motion $\hat{IMU}$ and the motion of inertial navigation $\overrightarrow{IMU}$.

3.2 Description of Notations

The trajectories are defined as $S = \{s^1, \ldots, s^N\}$, where N is the number of trajectories; $s^i = \{Z^i, U^i\}$ represents the ith trajectory, and $Z^i = \{z_1^i, \ldots, z_t^i, \ldots, z_{M^i}^i\}$ is the RSS sequences of the ith trajectory composed of M^i samples, where z_t^i represents the RSS sequence recorded on the ith trajectory at time t. $U^i = \{\overrightarrow{u}_1^i, \ldots, \overrightarrow{u}_t^i, \ldots, \overrightarrow{u}_{M^i-1}^i\}$ represents inertial vector sequences between each two continuous RSS sequences in Z^i. The inertial vector between two RSS sequences z_{t-1}^i and z_t^i is defined as $\overrightarrow{u}_{t-1}^i$. The local transition relation triple $\{RSS_{t-1}, IMU_{t-1}, RSS_t\}$ is recorded as $g = \{z_{t-1}, \vec{u}_{t-1}, z_t\}$. As we do not consider which trajectory the triples belong to, all triples are recorded as T. The main notations are described in Table 1.

4 CONSTRUCTION OF LIGHTGBM-CTMM

The basic idea of Lightgbm-CTMM construction is to effectively capture specific expected relationships between transitions of signal space and corresponding motion by smoothing the motions $\vec{u}$ in similar transitions of signal space. The expected motion relationship is shown in Figure 3. The RSS_{t-1} is RSS feature at location p_{t-1} at time $t-1$, and the RSS_t is RSS feature at location p_t at next time t. There is a change in physical location between p_{t-1} and p_t that is recorded as $\overrightarrow{IMU}_{t-1}$. So this relationship is between RSS_{t-1} and RSS_t. Because the time interval between time $t-1$ and time t is so short due to the WiFi module's recording frequency, the location changes are slightly recorded by IMU. The main work includes the construction and training of local space model CTMM to build classification model Lightgbm-CTMM.

4.1 Construction of CTMM

The basic ideas of CTMM construction: The RSS sequence z represents one position in the actual environment, although the specific location is unknown. A clustering method is used to find similar signal messages in References [3, 5, 18, 25], and the employment of clustering algorithm improves the locating performance of their methods. Likewise, two continuous RSS sequences $\{z_{t-1}, z_t\}$ represent one local space in the actual environment, although the exact location of the local space

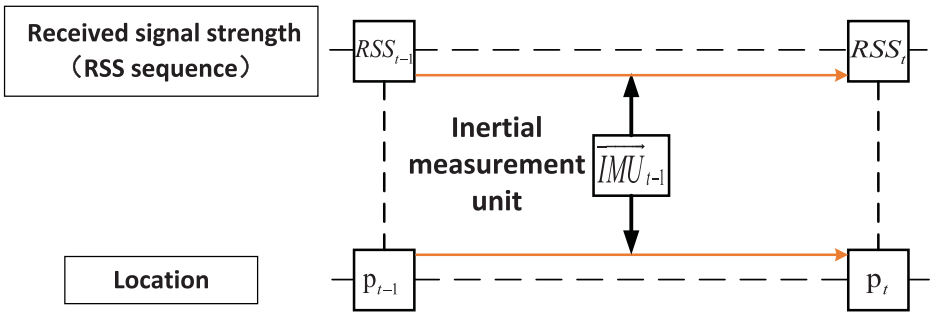


Fig. 3. Explanation of motion relationship.

is unknown. But we can cluster for $\{z_{t-1}, z_t\}$ to find similar changes in signal space and group them into the same cluster. In this way, the clustering method is used to find similar transitions of signal space.

We propose to use the clustering method to find similar transitions of the signal space to ensure the great robustness of model coverage. In our framework, we adopt k-means for clustering. We give a reference calculation of the preset values; the value of K can be set according to the size of the environment and the length of motion, which records slight changes in physical position because of the recording frequency of WiFi module. In most cases, the recording frequency of a WiFi module is constant in a single device. Although it varies from device to device, the difference is not noticeable. So, we take the average length of motion recorded by IMU between continuous WiFi records times as the estimated length of motion to calculate the expected number of models. The length of the path in environment is recorded as l, and the estimated average length of motion is recorded as $step$. The expected number of models, K, can be calculated by $K = l/step$. The purpose of estimating the average length of motion is to determine the number of models.

Because of the discordance between signal space change and physical space change, as well as the transitory relationship in transition model, the similarity of signal space transitions is high, whereas physical space transitions have different directions in the same cluster. Therefore, triples in same cluster need to be regrouped by motion $\vec{u}$. Grouping by direction aims to reduce the impact of different motion directions. Most importantly, it aims to avoid the circumstance where the expected motion is offset by motions with opposite direction when smoothing the motion. Figure 4 shows the basic construction idea of CTMM.

The trajectories' original format is not available, so we need to preprocess the data. The goal is to capture the relationship in local space, so the massive and unlabeled trajectories need to be converted into triple format $g = \{z_{t-1}, \vec{u}_{t-1}, z_t\}$ that represents one local space. We get the triples set $T = \{g_1, g_2, \ldots, g_m\}$, where m represents the number of all triples. After data preprocessing, CTMMs can be constructed by the triples set T. First, k-means clustering is conducted to generate cluster indices for triples set T, where the signal transition $\{z_{t-1}, z_t\}$ is taken as feature vectors. We get the results of clustering $C = \{c_1, c_2, \ldots, c_M\}$, where $c_i \in \{1, K\}$. Even though we have local spaces by clustering based on signal transition $\{z_{t-1}, z_t\}$, there are different directions of motion in triples that belong to the same cluster because of the discordance between signal space change and physical space change. To address this problem, we group the results in each cluster by direction of motion $\vec{u}$ as our second step. We decompose the motion vector into two-dimensional coordinate axes, while the direction with the longest distance on the coordinate axis is denoted as the direction of the triple. Figure 4 also shows the definition of direction of triple. In addition, you can also use other methods to smooth the results in different directions. Now, triples that belong to

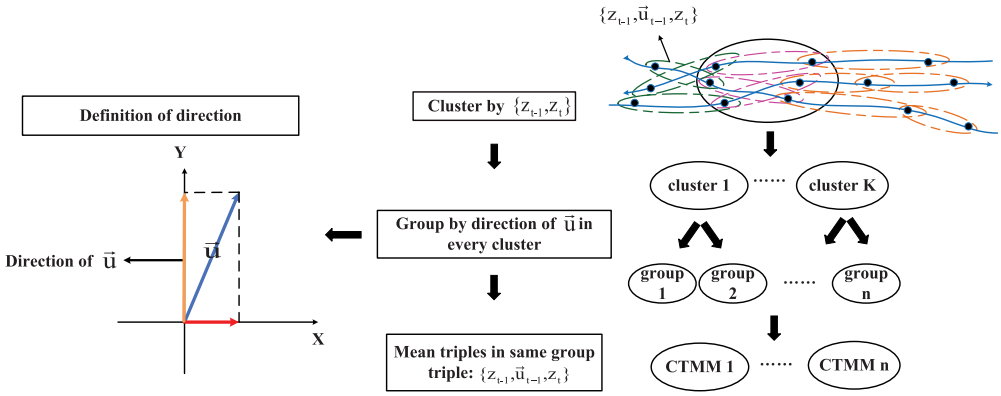


Fig. 4. Basic construction idea of CTMM.

the same group have similar signal transition and motion. Finally, to minimize the impact of noise in IMU, the CTMM is built by averaging the triples that belong to one group, where one triple is $g = \{z_{t-1}, \vec{u}_{t-1}, z_t\}$. We average the $\{z_{t-1}, z_t\}$ as the representative signal transition of CTMM that is built by one group and average the $\vec{u}_{t-1}$ as the expected motion $\hat{u}_{t-1}$ of this CTMM. The CTMM is shown as $\hat{u}_{t-1} = CTMM(\bar{z}_{t-1}, \bar{z}_t)$, where $\bar{z}_{t-1}, \bar{z}_t$ represent the average RSS features. The specific process is described in Algorithm 1.

ALGORITHM 1: Construction of CTMM

Input: K, T
Output: $CTMM(\bar{z}_{t-1}, \bar{z}_t)s$

#Data Preprocessing Part
for each $s \in S$ **do**
  change s to triple format $g = \{z_{t-1}, \vec{u}_{t-1}, z_t\}$
  add g to T
end for
#Construction Part
$C = k\text{-}means(T, K)$ by $\{z_{t-1}, z_t\}$
#$\{z_{t-1}, z_t\}$ as eigenvector
for $i = 1$ to K **do**
  $G^{'} = T(find(C == i))$
  group by direction of $\vec{u}$ of g in $G^{'}$ recorded as G_as
  averaging triples $\{z_{t-1}, \vec{u}_{t-1}, z_t\}$ in G_a to build $CTMM(\bar{z}_{t-1}, \bar{z}_t)$
end for

4.2 Construction of Lightgbm-CTMM

In the online stage, the optimal matching CTMMs need to be searched by a real-time triple of mobile target. This is a classification task, in which Lightgbm algorithm is well applied [8, 36]. Therefore, we suggest using Lightgbm algorithm to capture the corresponding relationship between CTMMs and the transition of signal space, so as to construct the Lightgbm-CTMM.

The construction of CTMM makes each triple correspond to its CTMM. The transitions of signal space are eigenvectors and CTMMs are the corresponding categories. For example, one triple $\{z_{t-1}, \vec{u}_{t-1}, z_t\}$ belongs to group i that is used to construct CTMMi. It means CTMMi is the category

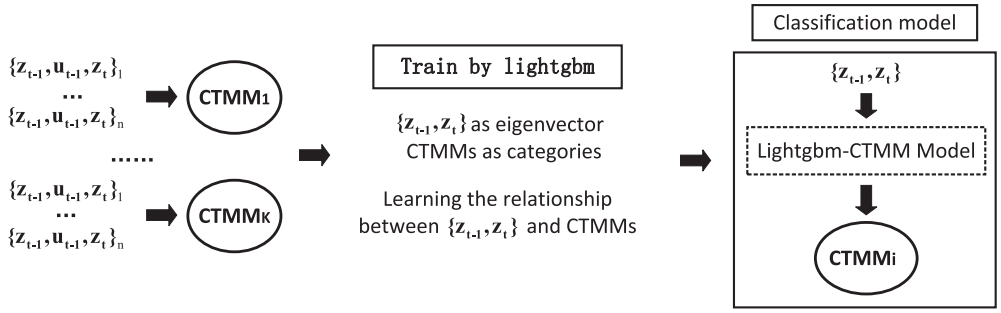


Fig. 5. Basic idea of Training Lightgbm-CTMM.

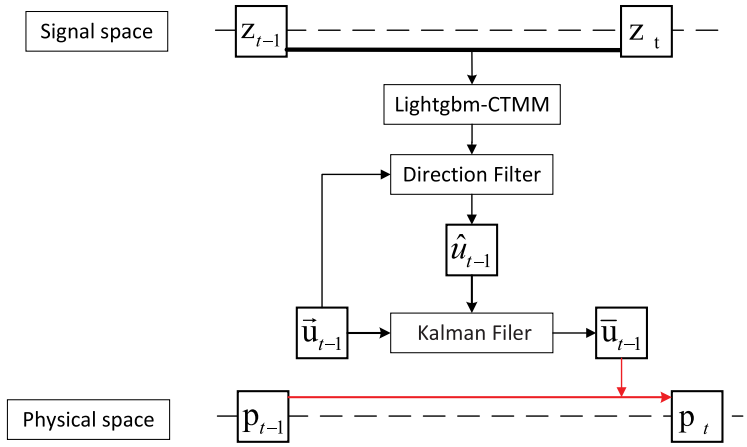


Fig. 6. Basic idea of online locating.

of transitions of signal space $\{z_{t-1}, z_t\}$. Lightgbm algorithm is utilized to capture the relationship between signal transition $\{z_{t-1}, z_t\}$ and CTMMs to build classification model Lightgbm-CTMM. Figure 5 shows the basic idea of constructing Lightgbm-CTMM. The aim of Lightgbm-CTMM is to obtain more accurate corresponding CTMM based on real-time signal transition in the trajectory. Lightgbm algorithm can learn a more accurate corresponding relation between signal transition $\{z_{t-1}, z_t\}$ and CTMM.

5 ONLINE LOCATING OF LIGHTGBM-CTMM

In the online locating stage, the observing motion $\hat{u}$ is obtained by Lightgbm-CTMM so that $\{z_{t-1}, z_t\}$ of mobile targets can be located. We combine the observing motion $\hat{u}$ with predicted motion $\vec{u}$ via Kalman Filter for online locating. The basic idea of the online stage is the utilization of direction filter and Kalman Filter, which is depicted in Figure 6.

The basic idea of direction filter: The inertial navigation signal is so noisy that trajectory drift occurs. But the motion is brief in transition model. We believe that the motion has certain credibility, although there is noise in the distance and direction of motion. Therefore, the best-matched K_{online} CTMMs are filtered based on the motion direction of mobile target to ensure the consistency of the motion direction. Most importantly, it aims to avoid the condition that best-matched CTMMs have opposite direction. However, the distance of motion is not taken into consideration, because we need to reduce the addition of noise data.

The basic idea of locating by Kalman Filter: The Kalman Filter is successful in the application of trajectory tracking [37]. The online locating method based on Kalman Filter is proposed by TMM [34]. In this method, the key step for state prediction is as follows:

$$\bar{p}_t = p_{t-1} + \vec{u}_{t-1}, \tag{1}$$

where $\bar{p}_t$ is the predicted position based on the position of the previous moment p_{t-1} and the motion $\vec{u}_{t-1}$ between t and $t-1$.

Currently, we can get the observing motion by Lightgbm-CTMM to correct the result:

$$\bar{u}_{t-1} = \vec{u}_{t-1} + K_t(\hat{u}_{t-1} - \vec{u}_{t-1}), \tag{2}$$

where K_t is the Kalman gain, $\hat{u}_{t-1}$ is observing motion, and $\vec{u}_{t-1}$ is predicted motion. Therefore, the estimated position can be written as

$$p_t = p_{t-1} + \bar{u}_{t-1}. \tag{3}$$

The specific process is described in Algorithm 2. There are two parameters σ_r and σ_q being processed, and the calculation result $rq = \log(\sigma_r^2/\sigma_q^2)$ is an important parameter that affects the result. We referred the calculation result rq as the confidence coefficient. When $rq < 0$, it means that the result of transition model is more plausible; when $rq > 0$, it means that the results of inertial navigation are more plausible. In the subsequent experiments, we will carry out specific evaluations of rq.

ALGORITHM 2: Online locating based on Kalman Filter

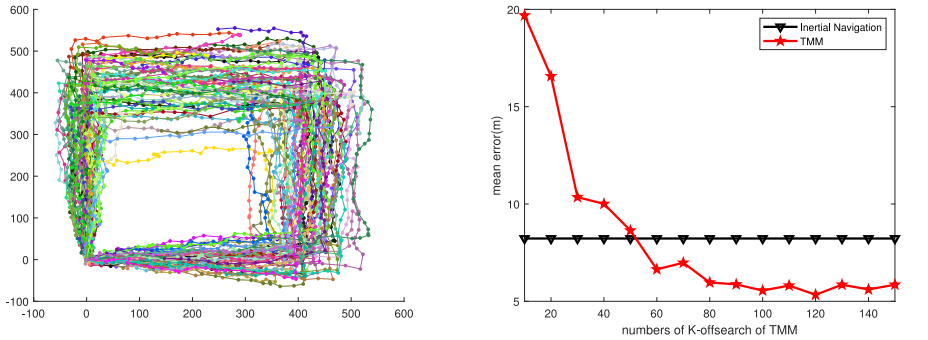
Input: p_{t-1}, $Lightgbm - CTMM$, $\{z_{t-1}, \vec{u}_{t-1}, z_t\}$, K_{online}
$\{z_{t-1}, \vec{u}_{t-1}, z_t\}$ is the current state of mobile target
Output: p_t
1: $Q_t = \sigma_q^2 I$
2: $\bar{u}_{t-1} = \vec{u}_{t-1}$
3: $\bar{\Sigma}_t = \Sigma_{t-1} + Q_t$
4: $Aim = Lightgbm - CTMM(z_{t-1}, z_t, K_{online})$
#K_{online} CTMMs for highest probability
5: $Aim^f = Direction\ Filter\ by\ \vec{u}_{t-1}\ for\ Aim$
6: $\hat{u}_{t-1} = mean(\vec{u}\ in\ Aim^f)$
7: $R_t = \sigma_r^2 I$
8: $S_t = \bar{\Sigma}_t + R_t$
9: $K_t = \bar{\Sigma}_t S_t^{-1}$
10: $u_{t-1} = \bar{u}_{t-1} + K_t(\hat{u}_{t-1} - \bar{u}_{t-1})$
11: $\Sigma_t = (I - K_t)\bar{\Sigma}_t$
12: $p_t = p_{t-1} + u_{t-1}$

6 SIMULATION EVALUATION

In this section, simulation experiments are used to investigate the performance of the Lightgbm-CTMM scheme under various conditions.

6.1 Simulation Setting

In the simulation experiments, we set 100 **access points (APs)** generated randomly in a 42 × 42 m^2 (420 × 420 pix^2) environment. The starting point of the trajectory is near the coordinates (0,0), and the starting direction is randomly up or to the right. The trajectories are moving along the



(a) View of trajectories in simulation evaluation (b) Parameter evaluation of TMM in offline training

Fig. 7. Evaluation environment and parameter evaluation of TMM in offline training.

Table 2. Interpretation of Contrast Experiment

Methods	Interpretation
Inertial Navigation (PDR)	pedestrians dead reckoning without additional optimization algorithms
TMM	traditional TMM method
TMM-filter	using direction filter in online stage of TMM additionally
CTMM-filter	the TMM model is replaced by CTMM model in TMM-filter method
Lightgbm-CTMM	classification model Lightgbm-CTMM using direction filter in online stage

edge of the environment. The number of trajectories is 100. We set the estimated average length of motion as 2 m (20 pix) and add the Gaussian noise $N(0, V_u = 10)$ to IMU. For each RSS sequence, it is calculated by

$$\mathrm{rss}(p_i) = \mathrm{rss}(p_0) - 10 * \alpha * \log\left(\frac{d_i}{d_0}\right) + \sigma, \tag{4}$$

where the σ represents the Gaussian noise $N(0, V_z = 10)$ in RSS value, $\mathrm{rss}(p_0)$ is the transmit power when the reference point is $d_0 = 1$m, $\mathrm{rss}(p_i)$ is the received signal strength at position p_i, α represents the path loss coefficient, and d_i is the distance between the position p_i and the AP. The view of trajectories is shown in Figure 7(a).

In simulation evaluation, TMM and PDR approaches are taken as the main contrast experiments, because the PDR is the foundation of TMM and Lightgbm-CTMM. Moreover, the TMM is the first to be proposed. The interpretation of all contrast experiments is shown in Table 2.

The followings are some influential parameters that need to be set:

(1) the confidence coefficient rq: Since the training data and test data are generated under the same conditions, the confidence coefficient rq is 0.
(2) the number of models K_{number}: Based on trajectory length and estimated average length of motion, we set K_{number} as 120. Because there are two main directions of motion in the

Table 3. Locating Results of Simulation Evaluation

Methods	Average Error(m)
Inertial Navigation (PDR)	7.26
TMM	5.95
TMM-filter	5.35
CTMM-filter	4.71
Lightgbm-CTMM	3.68

environment, the K of CTMM construction is set to half of TMM's K_{number} to ensure the consistency in the number of TMM and CTMM.

(3) the number of matching models in the online stage K_{online}: K_{online} of TMM is 5, K_{online} of Lightgbm-CTMM is 2. We choose top-K_{online} best matching models to get our offline motion. We believe that selecting the average of the K_{online} best values as the final result can improve the locating performance of each model to the expected level.

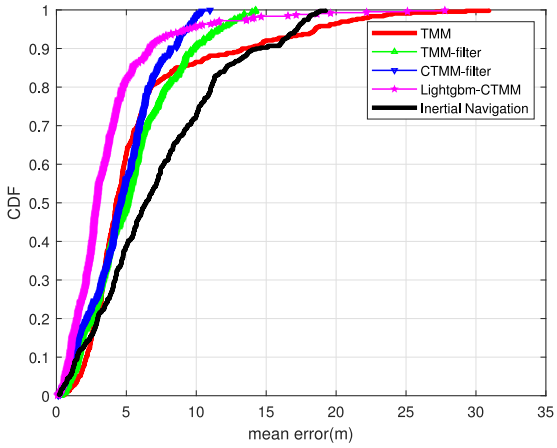
In addition to these parameters, the parameter of offline training $K_{offline}$, which affects TMM's locating performance, needs to be considered. In the offline stage, the segmented triples obtained by TMM have strong noise, so TMM uses a smoothing method based on the KNN algorithm to smoothly average $\vec{u}_{t-1}$ to minimize the drift caused by the PDR algorithm. So $K_{offline}$ is the parameter in the KNN algorithm. This value is evaluated when other parameters are consistent, and the result of this parameter is shown in Figure 7(b). When the value of $K_{offline}$ is set as 100, the locating results of all approaches are basically stable. The results of TMM are fluctuant, because the construction of TM is random to some extent. So the $K_{offline}$ is set as 100 in this simulation evaluation. Compared with TMM, Lightgbm-CTMM does not require the consideration of this parameter. The Lightgbm-CTMM reduces the influential parameter in the offline stage.

6.2 Locating Accuracy

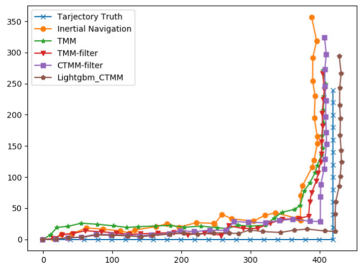
In this experiment, we evaluate the performance of the Lightgbm-CTMM in simulation evaluation. We conducted 10 cross-validation experiments, and we choose 10 trajectories as test data, and the rest of the tracks are used as training data. Figure 8(a) shows the **cumulative distribution function (CDF)** of locating error of different approaches. Table 3 shows the specific average error of different methods. The locating accuracy of Lightgbm-CTMM is better than that of TMM. Figures 8(b) and 8(c) show the locating views of trajectories. It is obvious that the trajectories of Lightgbm-CTMM are closer to the true tracks. Because the RSS sequences are created by an ideal signal propagation model, the locating result of TMM-filter is better than that of TMM on average error while the locating result of CTMM-filter is worse than that of TMM of small locating error in the CDF graph. Although we add the Gaussian noise, the discordance between signal space change and physical space change is not obvious enough.

6.3 Effect by Number of Model

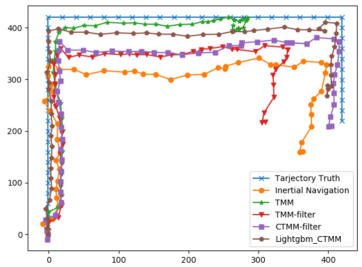
In this experiment, we evaluate the effect of the number of models. Figure 9(a) shows the performance under different numbers of models. When the number of models K_{number} reaches 70, the locating results of all approaches are basically stable. But the locating result of Lightgbm-CTMM is better than that of TMM when the results are mainly steady. Most importantly, Lightgbm-CTMM can achieve greater performance with less models than TMM. The results of all approaches are



(a) CDF of locating error

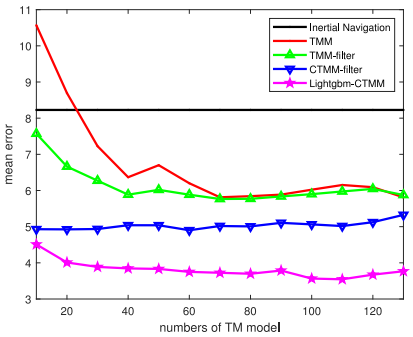


(b) View of locating results 1

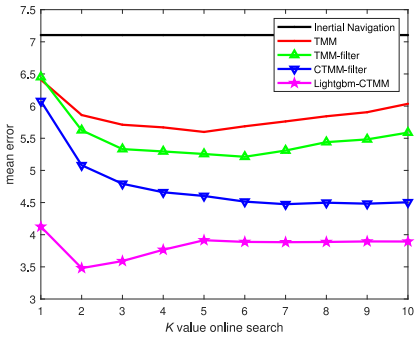


(c) View of locating results 2

Fig. 8. The locating results of simulation evaluation and view of locating trajectories.



(a) Effect of the number of model



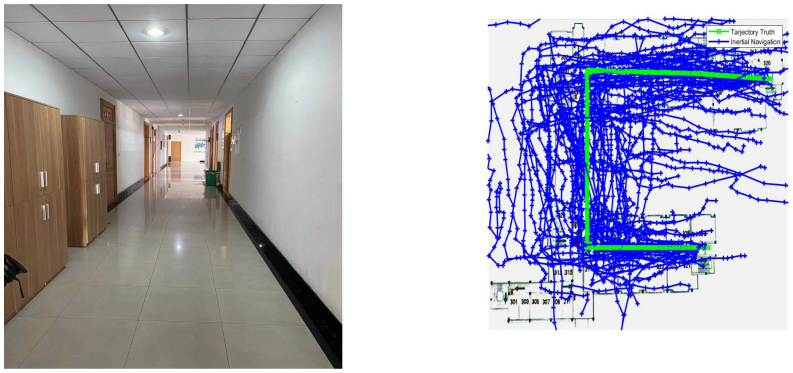
(b) Effect of the value of K in online stage

Fig. 9. Simulation evaluation of different factors.

fluctuant, because the construction of all approaches is random to some extent. However, the stability of Lightgmb-CTMM is preferable to that of others.

6.4 Effect by Value of K in Online Stage

In this experiment, we evaluate the effect of K_{online} in the online locating stage. Figure 9(b) shows the performance under different K_{online} in the online locating stage. The setting of different K_{online}



(a) Floor map (b) Trajectories view

Fig. 10. Experimental environment.

influences the locating results. The best K_{online} of Lightgbm-CTMM is 2. When the K_{online} is set as 4 or 5, the best locating results are obtained. But when K_{online} exceeds the optimum, the locating result of TMM becomes worse than the best result of TMM. The increase of K_{online} indicates that the search area has been expanded, and some unreasonable TMMs have been added, which makes the locating performance worse. Lightgbm-CTMM has a better locating result than TMM with varied K_{online}. Most importantly, the increase of K_{online} does not lead to a continuous decline in locating performance. The experimental results also prove that when $K_{online} > 1$, each model will get a better expected locating performance.

7 EXPERIMENTAL EVALUATION

7.1 Experiment Settings

The trajectories are collected on the 3rd floor of Teaching Building 1 at Hangzhou Dianzi University in Hangzhou, China. The floor space is about 150×15 m^2 and the effective length of trajectory coverage is about 150 m. The environment is shown in Figure 10(a). The number of effective APs in the environment is measured to be 205. Fifty-four IMU+RSS trajectories are collected by the open source application [23, 27], with 40 trajectories collected by holding the device in hand and 14 trajectories collected by putting the device into a pocket. Trajectories are collected by three different phones: Vivo, Xiaomi, and Meizu. Different devices produce varied WiFi frequencies, resulting in distinct length of motion $\vec{u}$. The estimated average length of motion $\vec{u}$ is about 2 m. Figure 10(b) shows the view of trajectories, where the green part represents the true trajectory coverage and the blue part represents the trajectories located by PDR algorithm. The goal of Lightgbm-CTMM is to see if we can achieve effective locating correction based on the traditional PDR algorithm, so the basic PDR algorithm is used in this experiment.

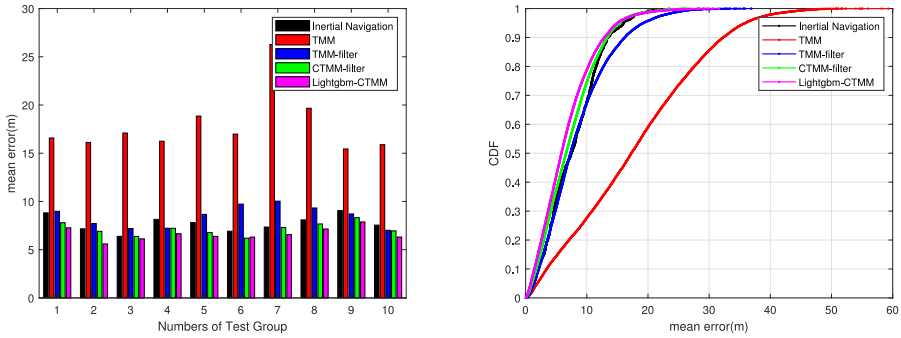
In this evaluation experiment, the PDR algorithm and TMM are taken as comparative experiments. Ten groups of experiments are performed where 5 trajectories are selected randomly as test data. The main influence parameters are shown in Table 4.

7.2 Locating Accuracy

In this experiment, we evaluate the locating performance of the Lightgbm-CTMM. First, we evaluate the locating performance when the PDR results of test trajectories have small locating errors. Figure 11(a) shows the experiment results. Figure 11(b) shows the CDF result of experiments. Because the traditional TMM model does not consider the inconsistency between the signal space

Table 4. Setting of Parameters

Parameters	TMM	Lightgbm-CTMM
Number of models K_{number}	150	150 (K value of k-means = 75)
Number of matching models in online stage K_{online}	3	2
Confidence coefficient rq:	0	0
Additional parameters of TMM in training stage $K_{offline}$	60	—



(a) Locating results of cross-validation experiments (b) CDF of cross-validation experiments

Fig. 11. Results of cross-validation experiments.

change and the physical space change without trajectory quality evaluation, and the confidence parameter $rq = 0$, the locating result of TMM in this experiment is worse than that of PDR. When we set the confidence parameter $rq > 0$, TMMs are accurately located by relying on inertial navigation rather than on motion from TMMs. In this experiment, compared with the locating results of Lightgbm-CTMM and TMM, it is unreasonable when we set $rq = 0$. The effectiveness of the direction filter is proved by the fact that the locating result of TMM-filter are better than that of TMM. The result of CTMM-filter is better than that of TMM-filter in experiments, which means that the CTMMs established through clustering have better coverage than traditional TMMs. Meanwhile, from Figures 11(a) and 11(b), we can find that the locating performance of TMM-filter is worse than that of PDR approach. The reason is that, while the direction filter filters out the offline motion in the opposite direction, the motion with large angle deviation is still not filtered out. However, we have grouped $\vec{u}_{t-1}$ in the direction in CTMM, so the locating error of the CTMM-based method is better than that of PDR and TMM. The result of Lightgbm-CTMM is better than that of CTMM-filter in most of the experimental groups. It means that a more robust relationship between signal transition $\{z_{t-1}, z_t\}$ and CTMMs has been established by lightgbm algorithm. The locating results of Lightgbm-CTMM and PDR are almost the same in the third group. The reason is that the trajectories of this group may match wrong CTMM and get the wrong inertial motion relationship so the locating results fail to get promoted or even go down compared with PDR. Table 5 shows the specific average error of all methods. In terms of statistical results, Lightgbm-CTMM is most effective among all methods. We use the result of PDR as the base to calculate the specific amount of improvement. The result of Lightgbm-CTMM increases by 14.43% approximately. Figure 12 shows the locating view when the test trajectories have normal locating result of PDR algorithm.

Table 5. Locating Results of Real Environment

Experimental Data	Methods	Average Error (m)	Improved Percentage (based on PDR)	Improved Percentage (based on TMM-filter)
Normal PDR result with small locating error	Inertial Navigation(PDR)	7.76	–	–
	TMM	17.77	–	–
	TMM-filter	8.42	−8.50%	–
	CTMM-filter	7.18	7.47%	14.47%
	Lightgbm-CTMM	6.64	14.43%	21.12%
Poor PDR result with big locating error	Inertial Navigation (PDR)	13.34	–	–
	TMM	21.06	–	–
	TMM-filter	12.30	7.79%	–
	CTMM-filter	11.77	11.76%	4.30%
	Lightgbm-CTMM	10.58	20.68%	13.98%

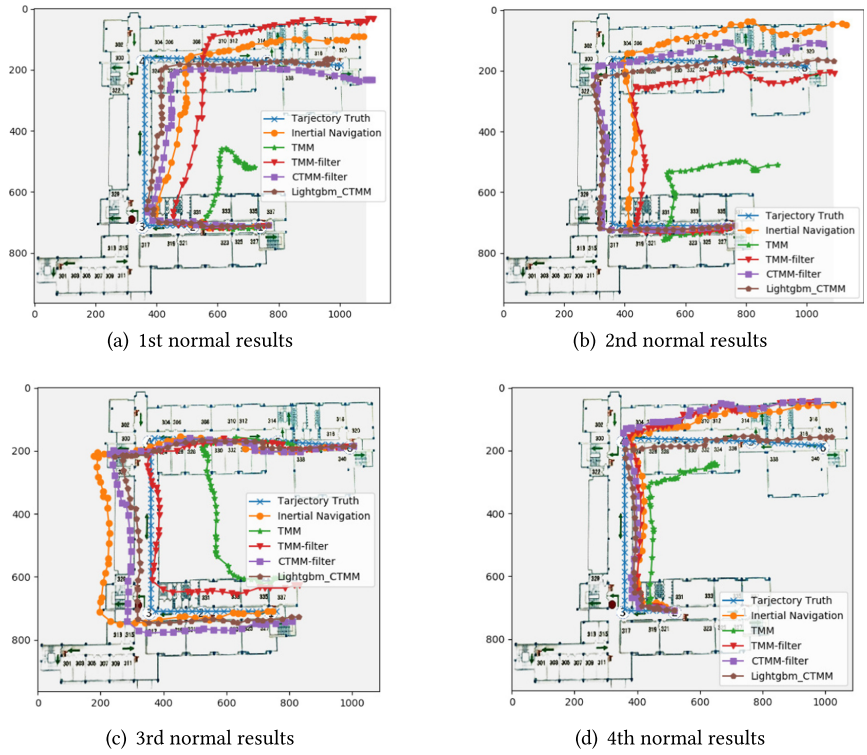
Compared with other approaches, Lightgbm-CTMM gains a better locating view and is closer to the actual trajectory. The result of TMM is quite different from the actual trajectory. We can also find that the TMM result is so much worse than the PDR result. Moreover, the result of the model revision can only guarantee the current optimal rather than the global optimal, so some locating results after partial revision are poor. This is also reflected in the third group experiment. However, the locating results after model revision are usually better from the perspective of statistical results.

In addition, we evaluate the locating performance when the PDR result of test trajectories has big locating errors. Figure 13(a) shows the CDF result of experiments. Table 5 also demonstrates the specific average error in this experiment. TMM-filter achieves a better result than PDR. In this experiment, the locating result of Lightgbm-CTMM is best when compared with other methods. Figures 13(b) and 13(c) show the locating views when the test trajectories have poor locating results of PDR algorithm. The result accuracy of Lightgbm-CTMM increases by 20.68% and 13.98% approximately compared with PDR and TMM-filter respectively. Since the PDR algorithm is the basis of Lightgbm-CTMM, the magnitude of the locating correction is valid.

In real-world experiments, TMM performs poorer than PDR in terms of location, but it outperforms PDR in simulation experiments. The reason is that the RSS sequences at close locations are more unstable in real environment, in which the discordance between signal space change and physical space change is more obvious. Although Gaussian noise is added in the simulation experiment, there is still a big difference between simulation and the real environment.

7.3 Effect by Confidence Coefficient

In this experiment, we evaluate the effect of the confidence parameter rq. Figure 14(a) shows the results with different confidence coefficients. $rq < 0$ means that we believe the results of transition model more than inertial navigation, that is, the correct result of trajectory is closer to the result got by framework. $rq > 0$ means that we believe inertial navigation more than the result of transition model, implying that the correct result of trajectory is closer to the result got by PDR. The setting of confidence coefficient means to evaluate the quality of trajectories data that contain training



(a) 1st normal results (b) 2nd normal results

(c) 3rd normal results (d) 4th normal results

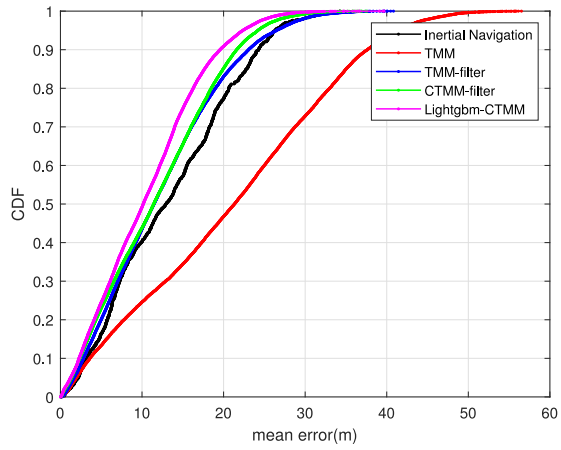
Fig. 12. Locating views with normal PDR result.

data and locating data. The problem of the discordance between signal space change and physical space change is not considered in TMM. Accurate locating of TMM is achieved by trusting inertial navigation more than the motion from TMMs, where the inertial navigation includes distance and direction of motion.

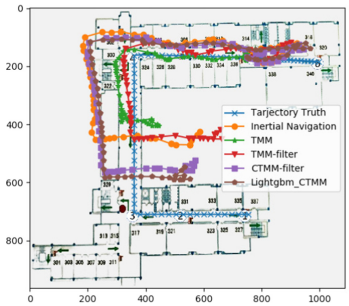
When $rq < 0$, the locating performance of TMM and TMM-filter improves obviously and the locating performance of CTMM-filter and Lightgbm-CTMM improves slightly as the rq increases. When $rq > 0$, the locating performance of TMM and TMM-filter improves obviously and the locating performance of CTMM-filter and Lightgbm-CTMM declines slightly with the increase of rq. When the rq is set as 1.5, the locating performance of all methods are basically equivalent to PDR algorithm, because the result of PDR algorithm plays a leading role and the correction effect of observing inertial motion relation has been mainly ignored. Lightgbm-CTMM can get great locating performance with different confidence coefficients, which means the Lightgbm-CTMM framework does not need to consider the confidence coefficient rq. Lightgbm-CTMM also believes the inertial navigation signal, but just considers the direction information of the inertial navigation signal. Lightgbm-CTMM reduces the evaluation of data quality, especially of the distance of motion.

7.4 Effect by Types of Device

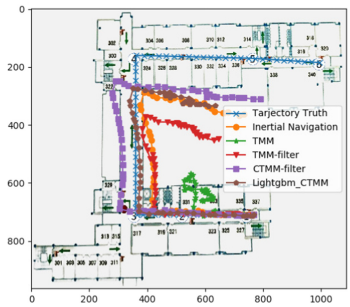
In this experiment, we evaluate the effect of device types. Different types of devices mean different frequencies of WiFi acquisition and different received signal strength in the same cases. The results can be seen in Figure 14(b). The problem of discordance between signal space change and physical



(a) CDF result of wrong PDR result

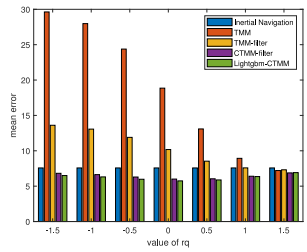


(b) 1st wrong results

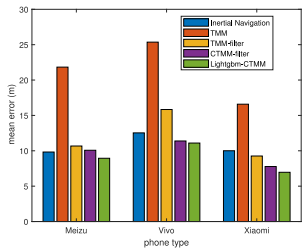


(c) 2nd wrong results

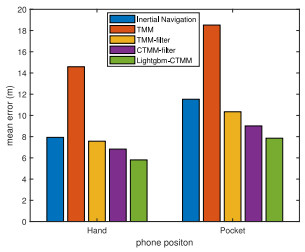
Fig. 13. Locating views with poor PDR result.



(a) Effect of confidence coefficient



(b) Effect of device types



(c) Effect of device postures

Fig. 14. Evaluation of different factors.

space change is more serious. Lightgbm achieves better locating result than TMM-filter and PDR with different devices. The robustness of types of the Lightgbm-CTMM framework is proved.

7.5 Effect by Device Positions

In this experiment, we evaluate the effect of the device positions. In our experiment, the device positions mainly include holding the device in hand and putting the device in pocket. Figure 14(c) shows the results of different device postures. The Lightgbm-CTMM has a better locating performance than both PDR and TMM-filter with different positions. The robustness of positions of the

Lightgbm-CTMM framework is proved. The PDR locating result of holding the device in hand is better than PDR locating result of putting the device in pocket. The reason is that the amplitude of swing of device in hand is greater than that in pocket, implying that the inertial navigation signal is better. The same is true for Lightgbm-CTMM.

8 CONCLUSION

In this article, we propose the unsupervised localization framework Lightgbm-CTMM to improve the locating accuracy of dead reckoning approach, which is built by massive, unlabeled, and crowd-sensing trajectories. The Lightgbm-CTMM is a model for reducing the influence of inertial sensor noise on PDR algorithm. To some extent, this model replaces the introduction of other high-cost locating methods as auxiliary locating information. Compared with unsupervised learning for radio map, this model is effective in reducing the workload and additional requirements. Furthermore, the Lightgbm-CTMM framework captures specific expected relationships and gives the motion from signal change in a more accurate way.

The traditional transition model's locating performance is hampered by the model's inadequate spatial coverage and the discordance between signal space change and physical space change. The CTMM, that is, transition model of Lightgbm-CTMM framework, is established through clustering the change relation of signal spatial to ensure the robustness of model coverage. Although the inertial navigation signal is noisy and includes the direction and distance, short-time motion is believable. Based on the direction filter of target motion change, the consistency of signal space change and physical space change is guaranteed. In short, our model outperforms TMM in terms of coverage robustness and direction recognition ability.

The simulation experiment evaluates the performance of the Lightgbm-CTMM framework under different parameters and proves the effectiveness of the Lightgbm-CTMM framework. The actual-environment experiment proves the feasibility and effectiveness of Lightgbm-CTMM. Compared with TMM, lightgbm-CTMM has higher locating accuracy. Besides, Lightgbm reduces the requirement for trajectory quality evaluation on quality assessment of trajectories. The basis for locating by Lightgbm-CTMM framework is dead reckoning approach. Therefore, the Lightgbm-CTMM framework is intended to correct the relative positions. In future work, the main research direction is to find a way to build scheme for absolute locating.

REFERENCES

[1] Jeonghee Ahn and Dongsoo Han. 2017. Crowd-assisted radio map construction for Wi-Fi positioning systems. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'17)*. 1–8.

[2] Paramvir Bahl and Venkata N. Padmanabhan. 2000. RADAR: An In-building RF-based user location and tracking system. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'00)*. IEEE Computer Society, 775–784.

[3] Yiqiang Chen, Qiang Yang, Jie Yin, and Xiaoyong Chai. 2006. Power-efficient access-point selection for indoor location estimation. *IEEE Trans. Knowl. Data Eng.* 18, 7 (2006), 877–888. https://doi.org/10.1109/TKDE.2006.112

[4] Ionut Constandache, Romit Roy Choudhury, and Injong Rhee. 2010. Towards mobile phone localization without war-driving. In *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM'10)*. 2321–2329.

[5] Chen Feng, Wain Sy Anthea Au, Shahrokh Valaee, and Zhenhui Tan. 2012. Received-signal-strength-based indoor positioning using compressive sensing. *IEEE Trans. Mob. Comput.* 11, 12 (2012), 1983–1993. https://doi.org/10.1109/TMC.2011.216

[6] Brian Ferris, Dieter Fox, and Neil D. Lawrence. 2007. WiFi-SLAM using gaussian process latent variable models. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*. 2480–2485.

[7] Jonathan Fink and Vijay Kumar. 2010. Online methods for radio signal mapping with mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'10)*. 1940–1945.

[8] Xile Gao, Haiyong Luo, Qu Wang, Fang Zhao, Langlang Ye, and Yuexia Zhang. 2019. A human activity recognition algorithm based on stacking denoising autoencoder and lightGBM. *Sensors* 19, 4 (2019), 947.

[9] Suining He and S.-H. Gary Chan. 2016. Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Commun. Surv. Tutor.* 18, 1 (2016), 466–490.

[10] Joseph Huang, David Millman, Morgan Quigley, David Stavens, Sebastian Thrun, and Alok Aggarwal. 2011. Efficient, generalized indoor WiFi GraphSLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'11)*. 1038–1043.

[11] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara. 2009. A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. In *Proceedings of the IEEE International Symposium on Intelligent Signal Processing*.

[12] Suk Hoon Jung and Dongsoo Han. 2018. Automated construction and maintenance of Wi-Fi radio maps for crowdsourcing-based indoor positioning systems. *IEEE Access* 6 (2018), 1764–1777.

[13] Suk Hoon Jung, Byung-chul Moon, and Dongsoo Han. 2016. Unsupervised learning for crowdsourced indoor localization in wireless networks. *IEEE Trans. Mob. Comput.* 15, 11 (2016), 2892–2906.

[14] Suk Hoon Jung, Byeongcheol Moon, and Dongsoo Han. 2017. Performance evaluation of radio map construction methods for Wi-Fi positioning systems. *IEEE Trans. Intell. Transport. Syst.* 18, 4 (2017), 880–889.

[15] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 3146–3154.

[16] Chuang Li, Xingfa Shen, Quanbo Ge, and Weijie Chen. 2021. An enhanced transition model for unsupervised localization. *IEEE Trans. Instrum. Meas.* 70 (2021), 1–11. https://doi.org/10.1109/TIM.2021.3107010

[17] Fan Li, Chunshui Zhao, Guanzhong Ding, Jian Gong, Chenxing Liu, and Feng Zhao. 2012. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the ACM Conference on Ubiquitous Computing (Ubicomp'12)*, Anind K. Dey, Hao-Hua Chu, and Gillian R. Hayes (Eds.). ACM, 421–430.

[18] Chengwen Luo, Hande Hong, and Mun Choon Chan. 2014. PiLoc: A self-calibrating participatory indoor localization system. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (IPSN'14)*, Adam Wolisz, Jie Liu, and Lin Zhong (Eds.). IEEE/ACM, 143–154.

[19] Lin Ma, Yanyun Fan, Yubin Xu, and Yang Cui. 2017. Pedestrian dead reckoning trajectory matching method for radio map crowdsourcing building in WiFi indoor positioning system. In *Proceedings of the IEEE International Conference on Communications (ICC'17)*. 1–6.

[20] Piotr W. Mirowski, Tin Kam Ho, Saehoon Yi, and Michael MacDonald. 2013. SignalSLAM: Simultaneous localization and mapping with mixed WiFi, Bluetooth, LTE and magnetic signals. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'13)*. 1–10.

[21] Deepak Pai, Sasi Inguva, Phani Shekhar Mantripragada, Mudit Malpani, and Nitin Aggarwal. 2012. Padati: A robust pedestrian dead reckoning system on smartphones. In *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'12)*, Geyong Min, Yulei Wu, Lei (Chris) Liu, Xiaolong Jin, Stephen A. Jarvis, and Ahmed Yassin Al-Dubai (Eds.). IEEE Computer Society, 2000–2007.

[22] Jun-geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth J. Teller, and Jonathan Ledlie. 2010. Growing an organic indoor location system. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*. 271–284.

[23] Francesco Potortì, Valérie Renaudin, Kyle O'Keefe, and Filippo Palumbo (Eds.). 2019. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'19)*. IEEE.

[24] Anshul Rai, Krishna Kant Chintalapudi, Venkata N. Padmanabhan, and Rijurekha Sen. 2012. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (Mobicom'12)*. 293–304.

[25] Ricardo Santos, Marília Barandas, Ricardo Leonardo, and Hugo Gamboa. 2019. Fingerprints and floor plans construction for indoor localisation based on crowdsourcing. *Sensors* 19, 4 (2019), 919.

[26] Chunjing Song and Jian Wang. 2017. WLAN fingerprint indoor positioning strategy based on implicit crowdsourcing and semi-supervised learning. *ISPRS Int. J. Geo-Inf.* 6, 11 (2017), 356.

[27] Joaquín Torres-Sospedra, Antonio Ramón Jiménez, Stefan Knauth, Adriano J. C. Moreira, Yair Beer, Toni Fetzer, Viet-Cuong Ta, Raúl Montoliu, Fernando Seco, Germán M. Mendoza-Silva, Oscar Belmonte, Athanasios Koukofikis, Maria João Nicolau, António Costa, Filipe Meneses, Frank Ebner, Frank Deinzer, Dominique Vaufreydaz, Trung-Kien Dao, and Eric Castelli. 2017. The smartphone-based offline indoor location competition at IPIN 2016: Analysis and future work. *Sensors* 17, 3 (2017), 557.

[28] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys'12)*. 197–210.

[29] Adam Wolisz, Jie Liu, and Lin Zhong (Eds.). 2014. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (IPSN'14)*. IEEE/ACM.

[30] Chenshu Wu, Zheng Yang, and Yunhao Liu. 2015. Smartphones based crowdsourcing for indoor localization. *IEEE Trans. Mob. Comput.* 14, 2 (2015), 444–457.

[31] Chenshu Wu, Zheng Yang, Yunhao Liu, and Wei Xi. 2013. WILL: Wireless indoor localization without site survey. *IEEE Trans. Parallel Distrib. Syst.* 24, 4 (2013), 839–848.

[32] Sungwon Yang, Pralav Dessai, Mansi Verma, and Mario Gerla. 2013. FreeLoc: Calibration-free crowdsourced indoor localization. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'13)*. IEEE, 2481–2489.

[33] Zheng Yang, Chenshu Wu, and Yunhao Liu. 2012. Locating in fingerprint space: Wireless indoor localization with little human intervention. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (Mobicom'12)*. 269–280.

[34] Xuehan Ye, Shuo Huang, Yongcai Wang, Wenping Chen, and Deying Li. 2019. Unsupervised localization by learning transition model. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 3, 2 (2019), 65:1–65:23.

[35] Jaegeol Yim, Seunghwan Jeong, Kiyoung Gwon, and Jaehun Joo. 2010. Improvement of Kalman filters for WLAN based indoor tracking. *Expert Syst. Appl.* 37, 1 (2010), 426–433.

[36] Hong Zeng, Chen Yang, Hua Zhang, Zhenhua Wu, Jiaming Zhang, Guojun Dai, Fabio Babiloni, and Wanzeng Kong. 2019. A LightGBM-Based EEG analysis method for driver mental states classification. *Comput. Intell. Neurosci.* 2019 (2019), 3761203:1–3761203:11.

[37] Jian Zhang, Bo Zhou, Shimin Wei, and Yuan Song. 2016. Study on sliding mode trajectory tracking control of mobile robot based on the Kalman filter. In *Proceedings of the IEEE International Conference on Information and Automation (ICIA'16)*. IEEE, 1195–1199.

Received October 2020; revised October 2021; accepted November 2021