Singular Value Decomposition Based Indoor Localization Using Small Scale Crowd Sensing Data

Xiaohao Liu, *Graduate Student Member, IEEE*, Yubin Zhao, *Senior Member, IEEE*, Xiaofan Li, *Senior Member, IEEE*, Huaming Wu, *Senior Member, IEEE*, and Cheng-Zhong Xu, *Fellow, IEEE*

***Abstract*—Traditional crowd sensing based indoor localization methods rely on large scale pre-collected fingerprint data to construct a radio map with cumbersome prior preparation. However, when they lack floor plan information or only have a little of data is willing to share, the tracking accuracy degrades significantly. In this paper, we propose a singular value decomposition (SVD) track matching scheme to obtain an effective radio map based on small scale crowd sensing data, which is a non-learning based system (SVD-CSP). SVD-CSP fuses received signal strength indicator (RSSI), inertial measurement unit (IMU), and magnetic field strength to label surrounding WiFi access points as marker points. The proposed scheme uses SVD method to directly compute the rotation matrix and displacement vector among the crowd sensing trajectories and attain the reliable tracks. The radio map is constructed and users are tracked according to our developed bidirectional Bayesian filter, which contains forward filter and reverse filter. The density-based spatial clustering of applications with noise (DBSCAN) is embedded within the forward filter to improve the radio map quality. Meanwhile, the reverse filter fuses pedestrian dead reckoning (PDR) and radio map-based localization to track users. Experimental results demonstrate that SVD-CSP can achieve robust localization using extremely sparse crowd trajectories (e.g., 4 trajectories in a 648 m^2 scenario, 30 trajectories in a 2856 m^2 scenario) without deep learning training or infrastructure knowledge.**

***Index Terms*—Indoor localization, crowd sensing, track matching, SVD.**

I. Introduction

With the rapid development of mobile Internet and location services, indoor positioning technology has been available for commercial applications. Among many indoor positioning techniques, WiFi fingerprinting is favored for its high accuracy and wide applicability [1]. This method requires collecting WiFi received signal strength indicator (RSSI) data at different locations within the target area via mobile devices (e.g., smartphones). In addition, the collected RSSI data will be associated with geographic locations to form a radio map. Then, the user's current query signal is compared with the pre-stored RSSI values by matching algorithms such as K-nearest neighbor (KNN) to determine the user's location. However, fingerprinting is a laborious and time-consuming process to construct these radio maps in order to perform adequate data collection in different locations in large and complex buildings. The preliminary work of mapping can be reduced to some extent by walking surveys or building sparse fingerprints [2], [3]. However, this still requires site surveys by professionals, which limits the scalability and coverage of location-based services [4], [5].

To overcome the limitations of traditional WiFi fingerprinting localization techniques in constructing radio maps, crowd sensing positioning (CSP) technology has emerged [6]. CSP is a method that utilizes data generated by ordinary users in their daily lives to automatically construct radio maps through crowdsourcing [7]. Thus, the reliance of on-site investigations by professionals has been reduced. The core idea lies in crowdsourcing to collect sensor data from multiple users, such as WiFi signals, inertial sensor data, etc., without users specifying the starting or ending locations [8]. By processing the large amount of data, location-related information and features are extracted. On one hand, the mobile tracks of different users are automatically generated based on inertial sensors. On the other hand, multiple location-related sensor features are fused to match a large amount of track data to record RSS. Then, CSP can automatically generate and update radio maps required for indoor localization. In this way, CSP not only improves the coverage of positioning services, but also dynamically adapts to changes in the environment, providing users with more flexible and real-time positioning services.

A. Motivations

Although CSP system can collect data from many users, the amount of shared data is still limited in practice, which brings several challenges. First, compare with the professional operator who collects the sensing data for a certain time based on a pre-defined trajectory, the user trajectories are rather random and unreliable. In this case, the collected data may be noisy, incomplete, or inaccurate due to the diversity of user devices [9]. In addition, user trajectories are solely generated by PDR

Received 9 February 2025; revised 29 July 2025; accepted 14 September 2025. Date of publication 16 September 2025; date of current version 4 February 2026. This work was supported in part by the National Nature Science Foundation of China under Grant 62271232 and Grant 62171484, and in part by the Guangdong Provincial Key Laboratory of High-end Integrated Circuit Design and Integration Technology (Sun Yat-Sen University) under Grant 2024B1212020007. Recommended for acceptance by J. Zhang. *(Corresponding author: Yubin Zhao.)*

Xiaohao Liu and Yubin Zhao are with the School of Microelectronics Science and Technology, Sun Yat-Sen University, Zhuhai 519082, China (e-mail: liuxh223@mail2.sysu.edu.cn; zhaoyb23@mail.sysu.edu.cn).

Xiaofan Li is with the School of Intelligent System Science and Engineering, Jinan University, Zhuhai 519070, China (e-mail: lixiaofan@jnu.edu.cn).

Huaming Wu is with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China (e-mail: whming@tju.edu.cn).

Cheng-Zhong Xu is with the State Key Lab of IoTSC, Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: czxu@um.edu.mo).

Digital Object Identifier 10.1109/TMC.2025.3610882

estimation. Without the help of floor plans and special landmarks, PDR tracks collected individually by crowdsourcing participants only have the relative movement and orientation of the user. Hence, these trajectories lack the actual information (e.g., references such as latitude and longitude) that can be applied for direct fusion. In some CSP methods, floor plans of the environment and sufficient landmarks (e.g., stairs, doorways, etc.) are used to assist in trajectory matching [10], [11]. Even in the face of crowdsourced data containing noise, these systems work well to construct radio maps with the help of landmarks and floor plans in the environments [12]. However, they cannot be used flexibly in indoor buildings that lack the true location information and floor plan constraints. Second, for privacy reasons, many users may not want to share their tracks to the cloud. Considering a new building which lacks the layout information, only a few users share their data for training. Thus, such data is insufficient for machine learning based algorithm, and specific method should be developed. Third, collecting sufficient data of CSP requires time, which may last for years. When the radio map is initially constructed, the new crowd sensed data is used to update the radio map. In this case, the system should consider the scalability of the trained model. For deep learning based model, the computational complexity is too high for both training and updating. In some CSP research without landmark assistance, many trajectory matching algorithms have high computational complexity [13], [14]. These methods require a large amount of trajectory data for map construction, which occupies more computational resources and cannot locate the target in real-time. Thus, non-machine learning or lightweight learning algorithms are preferred. Finally, the CSP system should consider the dynamic environments, where AP location changes or building structure updates sometimes [15]. In this case, the professional operators cannot sense that case in time, but the system should gradually update the map through continuous sensed data.

B. Contributions

In this paper, we propose the singular value decomposition crowd sensing positioning (SVD-CSP) framework without site survey and landmark assistance. The framework achieves track matching by calculating rotation matrices and translation vectors of the same marker points on different trajectories. The core of the approach lies in matching the walking tracks of different users (with different starting points) and reconstructing radio maps based on the matched tracks. The main contributions are summarised as follows:

- First, we implement track matching based on SVD with location-dependent markers on the tracks. To avoid feature point alterations due to fluctuations in RSSI, we fuse magnetic field strength as the matching threshold to assist track markers determination. Then, SVD is applied to match markers on different tracks to obtain the rotation matrix and translation vector.
- Second, we employ DBSCAN to fuse radio maps of multiple trajectories. Due to the noise in RSSI data collected while moving, the positions of the same APs change after track matching. DBSCAN eliminates the outliers of APs and clusters similar AP positions to improve the quality of radio maps.
- Third, we use bidirectional Bayesian filter to provide localization services for the target. We first calculate initial AP position information based on the forward Bayesian filter and the fused crowd sensing trajectories. Then, the target location is attained by fusing the PDR and the initial localization information through the reverse Bayesian filter.

SVD-CSP improves the coverage and accuracy of radio maps by fusing location dependent data collected by multiple users while walking. It does not rely on the prior knowledge of the environment floor plan or the manual initial calibration of the movement trajectories. Radio maps can be efficiently constructed and users can be localized in small datasets and simple scenarios. Note that the definition of ”small scale data” in this paper does not imply a limitation on the amount of data. Instead, our system aims to efficiently utilize even the smallest initial crowdsourced data (e.g., data from early participants) to guide the construction of radio maps. Meanwhile, incremental updates are supported as more data become available. Thus, the requirement for expert operator is avoided, which is particularly beneficial in dynamic environments where infrastructure or AP deployments may change over time.

We conduct extensive experiments in several indoor environments. Especially for the trajectory matching process that focuses on the path of the leader (service target), radio maps can be constructed using only a few data samples. Compared to existing methods, SVD-CSP is approaching the actual path when using only 4 tracks for matching in an environment of 648 m^2. Moreover, we can achieve an average localization error of 1.38m using only 30 trajectories for matching in the region covering 2856 m^2. In terms of computational efficiency, the computation time of the SVD track matching method increases by only 0.1s when the number of tracks increases by 40.

II. Related Work

A. Pedestrian Dead Reckoning (PDR)

The PDR algorithm consists of three key steps, including walking motion detection [16], step length estimation [17], and heading estimation [18]. However, differences in smartphone hardware configurations and drifts caused by temperature are intrinsic factors that limit the accuracy of PDR. Besides, the unknown position and attitude of the phone can affect the PDR estimation performance, and the magnetometer can be distorted by magnetic field interference from building structures and electronic devices. Although the quaternion-based Kalman filtering algorithm [19] and the hidden Markov model [20] have reduced the heading errors caused by magnetic field interference and equipment heterogeneity, there are still limitations in long time trajectory estimation. On the one hand, the high frequency (above 100 Hz) IMU used to maintain trajectory accuracy imposes a huge energy cost on the smartphone. On the other hand, even the use of low-frequency PDR estimation methods such as setting attitude sensing threshold [21] or training neural network based gravity prediction models [22] will lead to the growth of

cumulative error over time. In this case, SVD-CSP reduces the dependence on high-frequency IMU data streams by integrating sparse WiFi (0.2 Hz), IMU, and magnetic field features (25 Hz). In addition, the PDR traces collected through crowdsourcing are all within 1 minute, which can effectively reduce the impact of cumulative errors.

B. WiFi Fingerprint Localization

The WiFi fingerprint matching localization method is robust to the non-line-of-sight (NLOS) of the environment and does not require additional equipment deployment. The fingerprint database will be constructed by selecting different access points (APs) with large differences in RSS in the offline phase, and K-nearest neighbor (KNN) is used to estimate the target location in the online phase [1]. However, APs are deployed irregularly in indoor environments with a large number of redundant nodes, leading to additional computational time and space overheads. In this case, fingerprint libraries are constructed by calculating the signal loss rate and stability to select APs with relatively continuous signal sources and less fluctuating RSSI [23]. Zhang et al. [24] used an isolated forest outlier detection algorithm to reject abnormal RSS outliers that are disturbed by the environment, and corrected the neighbour temporal RSS weighted filtering to provide high-quality fingerprint data. In addition, Kullback-Leibler divergence is introduced to measure the difference between two RSS probability distributions to enhance the distinction of fingerprint data in complex environments [25]. Meanwhile, the WiFi signals received in the online localization phase also suffer from signal interference and loss, posing a challenge to the performance of KNN with a fixed K value. Luo et al. [26] used the Gaussian mixture model to divide the fingerprint data into multiple sub-regions, then dynamically select the K value according to the distance thresholds designed for different sub-regions, which reduces the amount of computation. SVD-CSP system does not rely on sub-area segmentation and dynamic K value selection, which allows more flexible response to different indoor environments by optimizing radio maps through DBSCAN.

Compared to the KNN method based on distance comparison, the neural network that maps the relationship between WiFi signals and location through a nonlinear function is adapted to complex and dynamic indoor environments. The RSSI data is decomposed into low and high frequency parts with Haar wavelet and these two parts are used as input features to the 1D CNN to obtain the localization results [27]. For dealing with the problem of fingerprint drift and feature disappearance, Deng et al. [28] fuse RSSI, signal strength difference and RSSI kurtosis to construct a radio image fingerprint and use a deep residual network to train the model for location estimation. Moreover, the zero-shot fingerprint enhancement scheme can effectively reduce the number of reference points in the offline phase [29]. Then, the map is reconstructed using a neural network to learn the difference between downsampled interpolated reconstructed maps and sparse maps to construct the offline fingerprint library sample data. However, the above methods still have high complexity and rely on large data set for training. In contrast, SVD-CSP system has better lightweight characteristics, and does not require large scale data training, which enables fast localization on resource-constrained devices.

C. CSI Based Indoor Localization

In recent years, channel state information (CSI) based localization techniques have gained attention for their ability to achieve sub-meter accuracy using fine-grained physical layer channel characteristics such as phase, amplitude, and frequency diversity. The pioneering work of Kotaru et al. [30] introduced SpotFi, which utilized angle-of-arrival (AoA) estimation of commodity WiFi hardware to achieve a median error of 40 cm. Expanding on this work, SPRING+ [31] combined CSI with fine time measurement (FTM) to localize smartphones with a single AP, achieving an accuracy of 1-1.8 meters. Single AP positioning systems additionally include M^3 [32] and SiFi [33]. The former exploits angular diversity of multipath signals and frequency hopping to separate direct and reflected paths via SVD and achieve a median error of 71cm in complex indoor environments. The latter focuses on single channel scenarios and utilizes the frequency-locked multi-antenna array feature to reach a single channel accuracy of 0.93m. Moreover, in recent deep learning approaches such as MFFALoc [34] and OpenPose-inspired models [35], the CSI based localization is further enhanced by fusing multi-featured fingerprints or reducing the computational complexity through innovative neural architectures.

However, if only a single AP is used for localization, the coverage of the system is limited. For the infrastructure based localization technique, prior knowledge of the in-building environment is also required. In addition, these methods rely on specialized hardware (e.g., Intel 5300 NICs or multi-antenna access points) and complex pre-processing (e.g., amplitude calibration) that require significant computational overhead, limiting real-time applicability. The phase and amplitude of CSI will fluctuate dramatically with human movement and layout changes. In contrast, SVD-CSP circumvents these challenges by relying on ubiquitous RSSI and inertial/magnetic sensors, which ensures scalability in cost-sensitive or resource-constrained environments.

D. Crowd Sensing Positioning

CSP primarily crowdsources the sensor data from ordinary users' walks or movements to simplify the process of building radio maps. Zee used the smartphone's inertial sensors to track the user and scanned the WiFi simultaneously [10]. The user's location is inferred based on the user's walking trajectory and floor plan (including paths/corridors and obstacles). HiMLoc used inertial sensors on mobile devices for location tracking and activity recognition, combined with WiFi fingerprints collected at specific locations (stairs, lifts, corners and entrances, etc.) for localization [9]. GROPING built maps using user contributed sensor data and semantic tags (i.e., semantic information and landmarks) and obtained online localization with geomagnetic fingerprints [12]. RCILS utilized crowdsourcing data collected by smartphones to abstract indoor maps into semantic graphs (where edges are possible user paths and vertices are locations

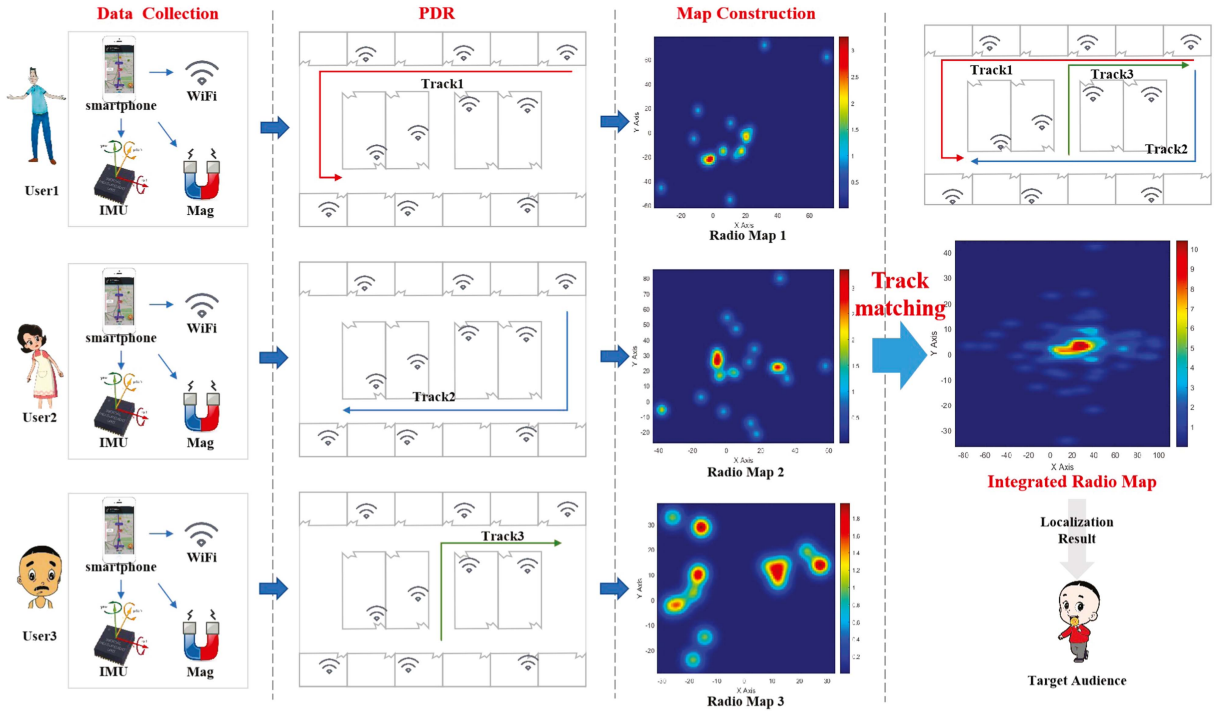

Fig. 1. CSP framework.

where users may perform particular activities) [11]. According to the semantic graph and the sequence of activities, crowdsourcing trajectories can be localized and radio maps can be constructed automatically. However, in the above study, when the floor plan and landmarks are missing, the localization accuracy of the whole system decreases or even fails to operate properly.

Crowdsourcing radio map construction and localization systems without the aid of any a priori information or user specific knowledge have been proposed in several types of research. APFiLoc proposed a distance constraint based clustering method to generate magnetic landmarks (magnetic field anomalies) and directional landmarks in an unsupervised manner [36]. The map information and landmarks are used to eliminate invalid particles (i.e., particles that pass through walls or other obstacles) and to correct the cumulative error of the PDR. PiLoc merged the segments of the walk labeled with user displacement and signal strength information to produce a map of the walking path labeled with radio signal strength [13]. MapGENIE filtered and revised the trajectories using the exterior information of the buildings and encoded the trajectories and the structural information of the buildings in conjunction with the indoor syntax to obtain floor plans [37]. Walkie-Markie determined the correct location of the WiFi access point on each trace based on the maximum RSSI as a marker for the trace, and then used the Arturia spring network to reconstruct the path [14]. WiFi-RITA formulated the trace merging problem as an optimization problem in which each trace is translated and rotated to minimize the distance constraints between the traces defined by the WiFi access points [15].

By comparison, SVD-CSP does not depend on prior information such as floor plans and landmarks. The trajectory data is efficiently processed by SVD and DBSCAN, avoiding complex iterative solving and semantic information. It improves the operation efficiency of the system while ensuring the positioning accuracy.

III. Crowd Sensing Positioning Framework

The general crowd sensing based positioning framework is depicted in Fig. 1. Considering that there are N_A APs in this environment and N_u users involved in data collection. First, each user carries a smartphone equipped with multiple sensors to walk in the same environment and collects data, including the WiFi RSSI, MAC address, IMU data (accelerometer and gyroscope), and magnetometer data. Note that, the sampling period of the WiFi data is different from IMU and magnetometer. Due to Android's hardware limitations (e.g., minimum WiFi scanning interval limitation), we set the WiFi RSSI scanning rate to 5 seconds and the sampling frequency of IMU and magnetometer to 25 Hz in our implementation. Therefore, we use t_w to represent the sampling period of WiFi data and t_s to represent the sampling period of IMU and magnetometer. The track points generated by the PDR are based on t_w. For user n_u, the integrated sensor data packet collected at sample point t_s is as follows:

$$\mathbf{S}_{n_u}(t_s) = (\mathbf{A}_{n_u}(t_s) \quad \mathbf{\Omega}_{n_u}(t_s) \quad \mathbf{B}_{n_u}(t_s)) \tag{1}$$

where $\mathbf{A}_{n_u}(t_s) = (a_x^{n_u}(t_s), a_y^{n_u}(t_s), a_z^{n_u}(t_s))$ is the acceleration, $\mathbf{\Omega}_{n_u}(t_s) = (\omega_x^{n_u}(t_s), \omega_y^{n_u}(t_s), \omega_z^{n_u}(t_s))$ indicates the gyroscope, and $\mathbf{B}_{n_u}(t_s) = (B_x^{n_u}(t_s), B_y^{n_u}(t_s), B_z^{n_u}(t_s))$ represents the magnetometer. Table I presents some of the WiFi packets collected during a typical cycle. And we denote the collected RSSI by $\mathbf{s}_{n_u}(t_w)$.

TABLE I
WIFI DATA PACKET

MAC	RSSI (dbm)
06:70:ab:4c:25:3e	-90
0a:70:ab:4c:3a:2d	-56
...	...
42:b4:bc:3e:3b:57	-63
d4:da:21:70:bf:d0	-45

Second, the system captures the user's movement track in real-time via the smartphone's IMU sensor. Since user p motion state (e.g., position, pace, direction) follows the Markov chain process, the state (position) of each step is directly computed from the previous state through (2).

$$\widehat{\mathbf{p}}_{n_u}(t_w+1)=\widehat{\mathbf{p}}_{n_u}(t_w)+L_{n_u}(t_w)\cdot\angle\psi_{n_u}(t_w) \tag{2}$$

where $L_{n_u}(t_w)$ is the step length at t_w, $\cdot\angle$ indicates the transformation from angle and length into physical positions, and $\widehat{\mathbf{p}}_{n_u}(t_w)=[\widehat{p}_x^{n_u}(t_w),\widehat{p}_y^{n_u}(t_w)]^T$ is the estimated track coordinates at t_w. In this case, Kalman filter smoothes the high frequency noise of the IMU data by combining the motion model predictions with the observations. Finally, the track $\widehat{\mathbf{X}}_{n_u}$ generated by user n_u is as follows:

$$\widehat{\mathbf{X}}_{n_u}=[\widehat{\mathbf{p}}_{n_u}(1),\ldots,\widehat{\mathbf{p}}_{n_u}(t_w),\ldots,\widehat{\mathbf{p}}_{n_u}(T_w^{n_u})] \tag{3}$$

where $T_w^{n_u}$ is the time of the last sample of user n_u.

IV. PROBLEM STATEMENT

With each estimated track, we can construct the radio maps for fingerprinting. However, we obtain multiple independent radio maps based on each independent track, which are presented as 2D heat maps in Fig. 1. In the absence of the ground truth position of each user, the interference in the WiFi signals makes it difficult to obtain an accurate map of the whole area from the direct fusion of these independent radio maps. Thus, our main goal is to extract the features of crowd sensing tracks and develop an effective clustering scheme which can form accurate radio maps. There are two challenges in this work. The first one is the absence of the prior information about the building, which makes the ground truth map unavailable. The second challenge is the small scale data set for crowd sensing tracks. When the building environment is unknown to the system, only a few users can share their data to the system. In this case, we can only use less than 10 tracks to form the radio map. Traditional machine learning or deep learning methods require large data for training, which makes the estimation inaccurate with such a small data set. However, the AP positions are relatively fixed even in the unknown environment. Thus, the feasible solution is to locate the APs and form a virtual radio map without using any prior building information. Then, our work can be divided into three parts. First, we need to calibrate the collected tracks and attain the similar features. Second, we mark the APs to form a virtual radio map based on the calibrated tracks. Finally, we can use the map to track the users.

A. Track Calibration

Consider we have a set of real tracks $\{\mathbf{X}_{n_u}\}_{n_u=1}^{N_u}$ and an estimated tracks $\{\widehat{\mathbf{X}}_{n_u}\}_{n_u=1}^{N_u}$ obtained by PDR localization. Our goal is to optimize the estimated track $\{\widehat{\mathbf{X}}_{n_u}\}_{n_u=1}^{N_u}$ so that the estimated track is as similar as possible to the real track. Here, we use Euclidean distance as a measure of similarity.

$$\begin{aligned}\mathcal{P}_1:\ &\min_{\{\widehat{\mathbf{X}}_{n_u}\}_{n_u=1}^{N_u}}\sum_{n_u=1}^{N_u}\left\|\widehat{\mathbf{X}}_{n_u}-\mathbf{X}_{n_u}\right\|^2\\ &=\min_{\{\widehat{\mathbf{X}}_{n_u}\}_{n_u=1}^{N_u}}\sum_{n_u=1}^{N_u}\sum_{t_w=1}^{T_w^{n_u}}\left\|\widehat{\mathbf{p}}_{n_u}(t_w)-\mathbf{p}_{n_u}(t_w)\right\|^2\end{aligned} \tag{4}$$

where $\widehat{\mathbf{p}}_{n_u}(t_w)$ and $\mathbf{p}_{n_u}(t_w)$ denote the estimated and true positions of user n_u at t_w respectively. PDR relies on inertial sensors in smartphones to estimate the user's movement track. Without the assistance of floor plans and special landmarks, the PDR trajectories collected individually by the crowdsourced participants suffer severe cumulative shifts. Consequently, the absolute position of PDR generated trajectories cannot be directly fused.

In order to correlate the WiFi signals and magnetic field signals with locations, the track movement patterns should be extracted. We introduce rotation angle error function $\mathcal{R}_e(\cdot)$ and track center translation error function $\mathcal{T}_e(\cdot)$ as the main error metrics.

$$\mathcal{R}_e(\widehat{\mathbf{X}}_{n_u},\mathbf{R}_{n_u},\mathbf{X}_{n_u})=\left|\Theta(\mathbf{R}_{n_u}\widehat{\mathbf{X}}_{n_u})-\Theta(\mathbf{X}_{n_u})\right| \tag{5}$$

$$\mathcal{T}_e(\widehat{\mathbf{X}}_{n_u},\mathbf{R}_{n_u},\mathbf{d}_{n_u},\mathbf{X}_{n_u})=\left|\mathcal{F}\left(\mathbf{R}_{n_u}\widehat{\mathbf{X}}_{n_u}+\mathbf{d}_{n_u}\right)-\mathcal{F}\left(\mathbf{X}_{n_u}\right)\right| \tag{6}$$

where $\mathcal{R}_e(\cdot)$ measures the error between the transformed and actual track heading angles; $\mathcal{T}_e(\cdot)$ evaluates the deviation from the centroid of the two tracks; $\Theta(\cdot)$ is a function used to compute the angle of the track; $\mathcal{F}(\cdot)$ is a function used to calculate the center of the track; $\mathbf{R}_{n_u}$ is the rotation matrix of the estimated track and $\mathbf{d}_{n_u}$ is the corresponding displacement vector. In the absence of absolute position, $\widehat{\mathbf{X}}_{n_u}$ is rotated by $\mathbf{R}_{n_u}$ to more closely approximate the heading angle of the true track. Then, $\mathbf{d}_{n_u}$ is used to compensate for the deviation from the center of the track. We define the overall track estimation error E as follows:

$$E=\sum_{n_u=1}^{N_u}\left(\mathcal{R}_e\left(\widehat{\mathbf{X}}_{n_u},\mathbf{R}_{n_u},\mathbf{X}_{n_u}\right)+\mathcal{T}_e\left(\widehat{\mathbf{X}}_{n_u},\mathbf{R}_{n_u},\mathbf{d}_{n_u},\mathbf{X}_{n_u}\right)\right) \tag{7}$$

Thus, $\mathcal{P}_1$ is transformed into $\mathcal{P}_2$:

$$\begin{aligned}\mathcal{P}_2:\ &\min_{\{\mathbf{R}_{n_u},\mathbf{d}_{n_u}\}_{n_u=1}^{N_u}} E\\ \text{s.t.}\ \ &v\le v_m\\ &k_c\le k_{cm}\end{aligned} \tag{8}$$

where v denotes the walking speed, v_m is the maximum walking speed, k_c is the kurtosis of curvature, and k_{cm} is the maximum kurtosis value. Here, we filter some of the invalid or noisy tracks

based on v_m and k_{cm}. Therefore, the main task of this paper is to find the optimal rotation matrix and displacement vector to make the transformed track closer to the real value.

Based on $\mathcal{P}_2$, the CSP system still has some problems. First, we need to solve the problem of IMU data discrepancy caused by heterogeneity such as variations in user devices and behaviors. Second, the trajectories generated by PDR lack ground truth position information and cannot be directly employed for trajectory matching. This requires us to select specific trajectory features for the sampled data. Finally, considering real-time localization, we would like to obtain the solution of $\mathcal{P}_2$ by one-time solving method. Thus, track clustering and SVD based track matching methods are developed to solve $\mathcal{P}_2$.

B. Radio Map Construction

Then, the radio map is constructed from the optimized tracks. For the n_A AP, let $\mathbf{p}_A(n_A)$ denote the true position in the environment and $\widehat{\mathbf{p}}_A(n_A)$ be the estimated position. Our goal is to minimize the total error between the AP positions in the constructed map and the AP positions in the real environment.

$$\mathcal{P}_3: \min_{\{\widehat{\mathbf{p}}_A(n_A)\}_{n_A=1}^{N_A}} \sum_{n_A=1}^{N_A} \left\|\widehat{\mathbf{p}}_A(n_A) - \mathbf{p}_A(n_A)\right\|^2 \tag{9}$$

where N_A is the total number of APs. For $\mathcal{P}_3$, a radio map is generated for each track. However, the estimated positions of the same AP will not overlap due to noise interference. In this work, we optimize the accuracy of the radio maps by excluding outlier AP positions through DBSCAN.

C. Localization

Finally, we focus on minimizing the error between the predicted and true positions of the localization system during the localization phase.

$$\mathcal{P}_4: \min_{\widehat{\mathbf{p}}_U} \left\|\widehat{\mathbf{p}}_U - \mathbf{p}_U\right\|^2 \tag{10}$$

where $\mathbf{p}_U$ denotes the user's true position and $\widehat{\mathbf{p}}_U$ denotes the user's predicted position. In this work, we use a bidirectional Bayesian filter to fuse PDR and RSSI based localization methods to further improve the localization accuracy.

V. Track Cluster

A. System Overview

The main parts of our CSP-SVD system is presented in Fig. 2. In the first part, we develop an Android App to collect sensor data to generate trajectories and cluster them. The sampling frequency of the IMU sensor is 25 Hz. WiFi RSSI and MAC addresses are scanned and stored in every 5 seconds when the user is moving. All sensor data are accompanied by the system time for synchronization and offline processing. When a new track is acquired, it is compared to the database to distinguish if the new track belongs to an area in the database. The second part is to locate the RSSI peak point markers on the trajectory. These markers are mainly used to mark the same information or features on different tracks. In the third part, we use the SVD to iteratively merge user trajectories by rotation and translation. In the fourth part, we use bidirectional Bayesian filter to provide localization services for target users. For a single matched track, the target equation between the AP signal and the anchors is established. The target equation is solved by forward Bayesian filtering to obtain an independent radio map for each track. However, the heterogeneity of user devices and behaviors results in locating the same AP on different positions. Therefore, we use DBSCAN to further aggregate the same anchors for map construction. Finally, we introduce reverse Bayesian filter to fuse PDR and map positioning to obtain optimized positioning results.

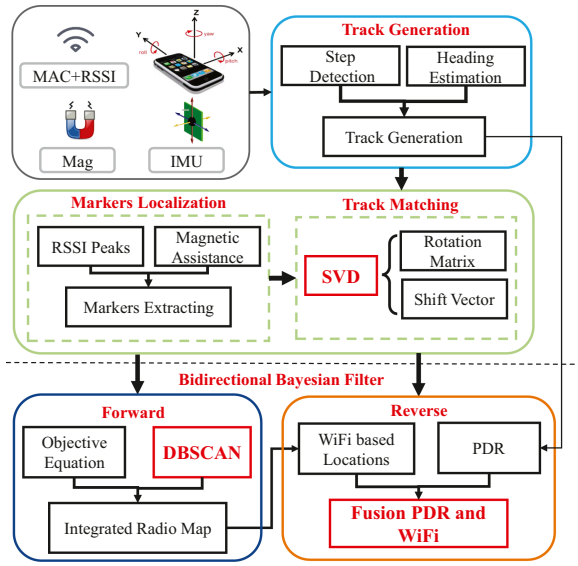

Fig. 2. Crowd sensing track matching and positioning system.

B. Track Generation

To obtain the crowdsourced user's movement trajectory, we design a PDR algorithm using the collected IMU sensor data which includes step detection, step length estimation and heading estimation. To avoid the indoor magnetic interference on heading, the user trajectory is estimated only based on the relative heading direction of the Kalman filtered gyroscope and accelerometer in the sensor coordinate system.

1) Step Detection: As illustrated in Fig. 3, the acceleration in each direction in the same cycle represents the regularity of alternating peaks and valleys during walking. We calculate the joint acceleration of the three-axis acceleration at each sampling point according to the following equation:

$$a_{n_u}(t_s) = \sqrt{a_x^{n_u}(t_s)^2 + a_y^{n_u}(t_s)^2 + a_z^{n_u}(t_s)^2} - g_{n_u}(t_s) \tag{11}$$

where $a_{n_u}(t_s)$ is the integrated acceleration of user n_u at sampling point t_s; $a_x^{n_u}(t_s), a_y^{n_u}(t_s), a_z^{n_u}(t_s)$ are the acceleration at t_s in the X, Y, Z axes of the carrier coordinate system; $g_{n_u}(t_s)$ is the gravitational acceleration. Then, the gait cycle of the pedestrian is determined by detecting the extreme points

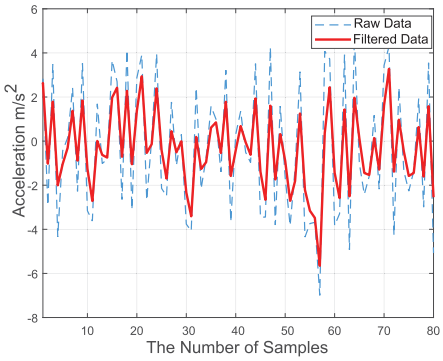

Fig. 3. Acceleration variation during walking.

(i.e., peaks and valleys) of the acceleration signal and setting a reasonable threshold.

2) Step Length Estimation: We use the nonlinear model proposed by Kim et al. [38] to estimate the step length.

$$L_{n_u}(t_w) = b\sqrt[3]{\frac{\sum_{t_s=t_w-1}^{t_w} |a_{n_u}(t_s)|}{n_w}} \tag{12}$$

where n_w is the number of samples in a walking step between $t_w - 1$ and t_w, $a_{n_u}(t_s)$ is the t_s acceleration value in the current step, and b is a constant, which is taken as 0.98in this paper.

3) Heading Estimation: The quadratic method is used to obtain the gyroscope posture angle and hence the heading estimation [39]. The quaternion representing the posture solution is calculated as follows:

$$q_{n_u}(t_w) = \mathrm{Re}\{q_0\} + \mathrm{Im}\{q_1, q_2, q_3\} \tag{13}$$

where q_0 is the angle of rotation; q_1, q_2, and q_3 three-dimensional vectors are used as the axes of rotation to convert the quaternion from the body system (b-system) to the navigation system (n-system). The rotation matrix C_b^n is as follows:

$$C_b^n = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{14}$$

The posture angles calculated from the rotation matrix are illustrated in (15).

$$\begin{cases} \theta_{n_u}(t_w) = arctan\frac{-2(q_1q_3-q_0q_2)}{q_0^2-q_1^2-q_2^2+q_3^2} \\ \varphi_{n_u}(t_w) = arcsin\left(-2\left(q_2q_3+q_0q_1\right)\right) \\ \psi_{n_u}(t_w) = arctan\frac{2(q_1q_2-q_0q_3)}{q_0^2-q_1^2+q_2^2-q_3^2} \end{cases} \tag{15}$$

where $\theta_{n_u}(t_w)$ is the pitch angle, $\varphi_{n_u}(t_w)$ is the roll angle and $\psi_{n_u}(t_w)$ is the yaw angle. Then, we can derive the user trajectory based on the step length and yaw angle.

C. Cohesive Hierarchical Track Clustering

In order to reduce the cumulative error on the PDR trajectories, the crowdsourced trajectories are segmented, where each segment trajectory is within 1 minute. Due to the errors caused by arm sway and other factors, we eliminate some abnormal trajectories from two aspects according to the empirical values from a large number of experiments. First, we prefer to consider tracks formed by normal walking rather than running to reduce the sensor noise. Second, we want stable and regular tracks, i.e., walking without excessive round trips or spins. Therefore, we filter some of the anomalous trajectories with maximum walking speed and maximum curvature kurtosis, where $v \le v_m$ and $k_c \le k_{cm}$.

For data in different indoor environments, we partition the tracks into various regions with a cohesive hierarchical clustering approach of Algorithm 1 to extend the coverage and application scenarios for building fingerprint libraries. Before database construction, we treat each track as a separate cluster $C_{n_u} = \{\widehat{\mathbf{X}}_{n_u}, \mathbf{I}_{n_u}, \mathbf{s}_{n_u}\}$, which contains the coordinates of the track generated by PDR, the MAC identifier of WiFi $\mathbf{I}_{n_u} = \{\mathrm{MAC}_{n_A}\}_{n_A=1}^{N_A^{n_u}}$ and the corresponding RSSI, where $N_A^{n_u}$ is the number of signals detected by the user n_u. The similarity of the two clusters is measured according to the Jaccard similarity in (16). Assume that user data is received from N_r different regions and the mac identifier of a $N_A^{n_r}$ signal in region G_{n_r} is $\mathbf{I}_{G_{n_r}} = \{\mathrm{MAC}_{n_r}\}_{n_r=1}^{N_A^{n_r}}$. As long as one of the signals in the region is detected when the track is moving, $J(\cdot) > 0$. The opposite indicates that the track belongs to another region or a new region. According to this method, different regions of the track are distinguished as $G_{n_r} = \{C_i\}_{i=1}^{N_{G_{n_r}}}$, where $N_{G_{n_r}}$ is the number of tracks belonging to G_{n_r}.

$$J(C_{n_u}, G_{n_r}) = \frac{|\mathbf{I}_{C_{n_u}} \cap \mathbf{I}_{G_{n_r}}|}{|\mathbf{I}_{C_{n_u}} \cup \mathbf{I}_{G_{n_r}}|} \begin{cases} > 0, C_{n_u} \in G_{n_r}, \\ = 0, C_{n_u} \rightarrow \text{new region} \end{cases} \tag{16}$$

VI. Singular Value Decomposition Based Track Matching

A. Magnetic Based AP Marker Localization

In indoor positioning systems, WiFi APs are ideal signal markers due to their ubiquity and high coverage. These markers are achieved by estimating the location of WiFi APs, which are spread throughout the indoor environment, hence the name WiFi markers. These markers are essential for determining the precise location of a device indoors as they provide a stable source of signals that can be detected and tracked multiple times by the device. The RSSI of each AP can be detected by multiple devices and signals from the same AP can be detected by different tracks and multiple locations. Therefore, we first need to determine which of the large number of WiFi APs can be used as WiFi markers and how these markers should be located on each track. In previous research, WiFi markers were identified as the location with the largest RSSI in the track or the center of the track covered by WiFi signals [14], [15]. However,

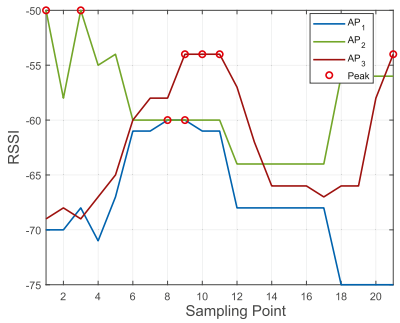

Fig. 4. AP RSSI curves on the track.

Algorithm 1: Cohesive Hierarchical Track Clustering.

Input: Signals detected per track: $\mathbf{I}_{n_u}$;
Total number of tracks: N_u;
Output: Divided region: $\mathbf{G}$
Initializing partition $G_1 = \{C_1\}$, $\mathbf{G} = \{G_1\}$ and $n_r = 1$;
for $n_u = 2, 3, \ldots, N_u$ **do**
  for G_{n_r} in $\mathbf{G}$ **do**
    Calculating Jaccard similarity $J(C_{n_u}, G_{n_r})$;
    if $J(C_{n_u}, G_{n_r}) > 0$ **then**
      $C_{n_u} \in G_{n_r}, \mathbf{I}_{G_{n_r}} = \mathbf{I}_{C_{n_u}} \cup \mathbf{I}_{G_{n_r}}$;
    else if $J(C_{n_u}, G_{n_r}) = 0$ **then**
      $n_r = n_r + 1$, C_{n_r} belongs to G_{n_r};
      Add G_{n_r} in $\mathbf{G}$;
    end if
  end for
end for
Return division of regions $\mathbf{G} = \{G_1, \ldots, G_{n_r}\}$

these WiFi markers are easily affected by multipath propagation. The RSSI profiles of the three APs while the track is moving are illustrated in Fig. 4. Using only the location of the peaks as markers is usually inaccurate and unstable due to multipath effects and signal interference issues that generate multiple peak points. Similarly, we cannot ensure that the coverage of the APs is credible.

Our goal is to define markers that are stable and robust on tracks for use as track matching. Therefore, we propose to use the indoor magnetic field strength to assist RSSI peak points to distinguish the effects of multiple peaks as the track feature points to be selected. In Fig. 5, we choose the two sides at 1.5 m apart from the router as the test points. From the RSSI curves, the amplitudes of the two test points are the same when the RSSI measurements are stable. Therefore, it is difficult to distinguish two points which are close only relying on RSSI as a feature. From the magnetic strength curve, the fluctuation of the same test point will not exceed 0.8 μT, and the difference between the two points is more than 2.5 μT. In this case, even if the two test points are only 3 m apart, we can easily distinguish between the two different test points by the magnetic strength.

Specifically in the implementation process, we need to filter the magnetometer data first to reduce the noise interference.

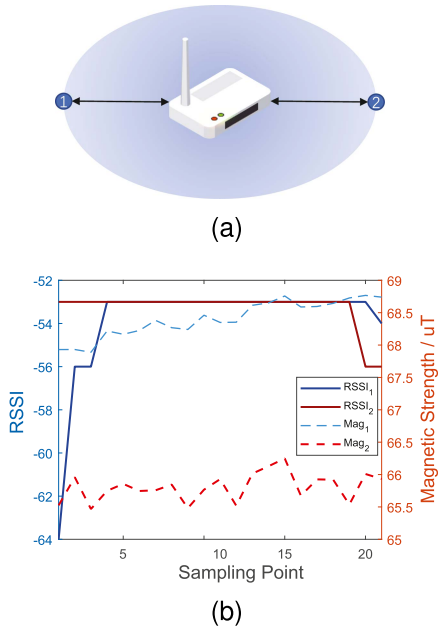

Fig. 5. Measurement of RSSI and magnetic field strength at two locations. (a) Specific measurement position. (b) RSSI and magnetic field strength curves at different positions.

Second, the three-axis components of the magnetic field vary with different postures and are less stable, but the total magnetic field intensity is more stable compared to the three-axis components. Therefore the total magnetic field strength B_{n_u} is used as an aid in this work.

Note that, the WiFi markers estimated on the tracks only represent the spatial distribution of the AP's location at the time of the pedestrian carrying the device, and not its absolute coordinates in the building coordinate system. Consider two trajectories that detect the same WiFi marker (RSSI peak point), i.e., they both detect a strong signal from this AP in the vicinity of this point, thus indicating that the trajectories are close to each other at this location. For the normal RSSI peak points, we can directly treat them as markers on the track. There are two main types of anomalies due to multipath and signal fading, the first is discrete RSSI peaks and the second is continuous RSSI peaks. In the former case, the magnetic field strength of each discrete peak is saved directly, and in the latter case, we tend to mark the middle point of multiple peaks as markers. Due to the step length inconsistency between sampling points, we use linear interpolation to ensure that the markers are on the track. In addition, in order to effectively rotate and translate the tracks, the APs corresponding to our rotated markers should be sensed on at least 3 tracks.

B. SVD Based Track Matching

Since we lack ground truth to directly solve the $\mathcal{P}_2$ problem, we propose the SVD-based track matching method to obtain the optimal rotation matrix $\mathbf{R}_{n_u}$ and displacement vector $\mathbf{d}_{n_u}$. First, we consider the collected N_u tracks as rigid bodies with constant shape and size. Theoretically, the positions of the

corresponding markers of the same AP on different tracks are close. Therefore, the matching of trajectories can be achieved by calculating $\mathbf{R}_{n_u}$ and $\mathbf{d}_{n_u}$ of the same marked points between two tracks. We assume that a total of N_A APs are detected on these N_u trajectories, which are expressed as $\widehat{\mathbf{X}}_{n_u}$. Due to the signal interference, or the moving distance is beyond the AP coverage range, the actual detected signals on on each track will generally be smaller than N_A. Before performing the trajectory transformation, the information matrix $\mathbf{D}$ corresponding to the trajectories needs to be prepared as in (17).

$$\mathbf{D} = \begin{bmatrix} \mathbf{M}_{1,1} & \cdots & \mathbf{M}_{1,N_u} \\ \mathbf{M}_{2,1} & \cdots & \mathbf{M}_{2,N_u} \\ \vdots & \ddots & \vdots \\ \mathbf{M}_{N_A,1} & \cdots & \mathbf{M}_{N_A,N_u} \end{bmatrix} \tag{17}$$

where $\mathbf{M}_{n_A,n_u} = \{(x_{n_A,n_u}, y_{n_A,n_u}), s_{n_A,n_u}, B_{n_A,n_u}\}$ is the marker of the AP n_A on the track n_u, containing the marker's position on the track, RSSI, and magnetic strength. If AP n_A is not detected on the track n_u, $\mathbf{M}_{n_A,n_u}$ is set to 0.

In the absence of a leader (reference) trajectory, we select one of these N_u tracks as a reference. In this case, the track that carries more amount of information is more likely to realize the matching of other tracks. Therefore, we introduce the Hamming distance to measure the AP similarity between the trajectories:

$$h_{i,j} = \sum_{n_A=1}^{N_A} \mathbf{D}_{n_A,i} \oplus \mathbf{D}_{n_A,j} \tag{18}$$

where $\mathbf{h}$ is a $N_u \times N_u$ matrix. Then, we extract the maximum Hamming distance $h_{\max} = \max\{h_{i,j}\}$ and calculate the average similarity $\mathcal{S}_{n_u}$ of the track n_u:

$$\mathcal{S}_{n_u} = \sum_{i=1}^{N_u} \frac{1 - h_{i,j}/h_{\max}}{N_u(N_u - 1)} \tag{19}$$

According to the $\mathcal{S}_{n_u}$ sorting, we match the remaining tracks sequentially by the amount of information. In our track matching method, it is essentially a transformation based on the markers on the tracks. We denote the markers on the reference track r as $\mathcal{M}_r = [(x_{1,r}, y_{1,r})^T, \ldots, (x_{N_A,r}, y_{N_A,r})^T]$, and the markers on the track n_u to be matched as $\mathcal{M}_{n_u} = [(x_{1,n_u}, y_{1,n_u})^T, \ldots, (x_{N_A,n_u}, y_{N_A,n_u})^T]$. Before constructing the covariance matrix, the markers need to be centered to eliminate the offset of the data. Considering that markers with larger RSSI are more reliable, we weight the average according to the RSSI corresponding to the marker points.

$$\mathcal{W}_{n_u} = \sum_{n_A=1}^{N_A} \frac{(x_{n_A,n_u}, y_{n_A,n_u})^T \exp(-0.01\,|s_{n_A,n_u}|)}{\exp\left(-0.01\,|s_{n_A,n_u}|\right)} \tag{20}$$

From the centered markers, we construct the covariance matrix $\mathbf{H}^r_{n_u}$, which contains the rotation matrix between the two marker vectors.

$$\mathbf{H}^r_{n_u} = \left(\mathcal{M}_{n_u} - \mathcal{W}_{n_u}\right)^T \left(\mathcal{M}_r - \mathcal{W}_r\right) \tag{21}$$

Then, SVD is performed on the covariance matrix:

$$\mathbf{H}^r_{n_u} = U^r_{n_u} \Sigma^r_{n_u} {V^r_{n_u}}^T \tag{22}$$

Algorithm 2: SVD-Based Track Matching.

Input: Information matrix: $\mathbf{D}$; N_u tracks: $\widehat{\mathbf{X}}_1, \widehat{\mathbf{X}}_2, \ldots, \widehat{\mathbf{X}}_{N_u}$;
Output: Matched tracks: $\overline{\mathbf{X}}_1, \overline{\mathbf{X}}_2, \ldots, \overline{\mathbf{X}}_{N_u}$

```
for i = 1, 2, ..., N_u − 1 do
  for j = 2, 3, ..., N_u do
    Calculate Hamming Distance h_{i,j};
  end for
end for
Calculate average similarity S = [S_1, S_2, ..., S_{N_u}];
Find reference track X̂_r and [M_{1,r}, ..., M_{N_A,r}]^T with max{S};
Remaining N_u − 1 tracks are sorted by S;
Set magnetic field threshold B_Th = 1.5;
for n_u = 1, 2, ..., r − 1, r + 1, ..., N_u do
  for n_A = 1, 2, ..., N_A do
    if Number of RSSI peak points in M_{n_A,n_u} > 1 then
      Find marker points by |B_{n_A,n_u} − B_{n_A,r}| < B_Th;
    end if
  end for
  Weighting marker points W_r and W_{n_u};
  Construct the covariance matrix H^r_{n_u};
  SVD solves the singular matrices U^r_{n_u} and V^r_{n_u};
  Compute rotation matrix R^r_{n_u} and translation vector d^r_{n_u};
  Transform track X̄_{n_u} = (R^r_{n_u} X̂_{n_u} + d^r_{n_u})^T;
end for
Return matched tracks X̄_1, X̄_2, ..., X̄_{N_u}
```

where $U^r_{n_u}$ and $V^r_{n_u}$ are orthogonal matrices which respectively contains the left and right singular vectors of the covariance matrix $\mathbf{H}^r_{n_u}$; $\Sigma^r_{n_u}$ is a diagonal matrix which contains the singular values. According to SVD, we construct the rotation matrix as follows:

$$\mathbf{R}^r_{n_u} = V^r_{n_u} {U^r_{n_u}}^T \tag{23}$$

where $\mathbf{R}^r_{n_u}$ represents the optimal rotation matrix for rotating the signal markers on track n_u to the reference track r. Here, $\mathbf{R}^r_{n_u}$ represents a reflection matrix if the determinant of $\mathbf{H}^r_{n_u}$ is negative. To solve this problem, the last column of $V^r_{n_u}$ is multiplied by -1 to ensure that $\det(\mathbf{R}^r_{n_u}) = 1$. Thus, we obtain the translation vector:

$$d^r_{n_u} = \left(\mathcal{M}_{n_u} - \mathcal{W}_{n_u}\right)^T - \mathbf{R}^r_{n_u}(\mathcal{M}_r - \mathcal{W}_r) \tag{24}$$

Combining $R^r_{n_u}$ and $\mathbf{d}^r_{n_u}$, we match the remaining tracks by the reference track.

$$\overline{\mathbf{X}}_{n_u} = \left(\mathbf{R}^r_{n_u} \widehat{\mathbf{X}}_{n_u} + d^r_{n_u}\right)^T \tag{25}$$

The specific algorithmic flow for implementing trajectory matching based on SVD is shown in Algorithm 2. First, we rank the tracks in terms of the amount of information they carry. The track with the highest information is used as the reference and the other tracks will be matched sequentially. Second, we set the magnetic threshold based on empirical values

to distinguish multiple peaks and record the effective marker locations of special markers. With the assistance of magnetic field, multiple tracks with the same RSSI peaks detected at different locations are strongly distinguished. When centering the tracks, we consider the problem that the RSSI fluctuations at the peak points are not necessarily the same, and use the natural exponential function weights to compensate. In terms of track matching, we construct a feature matrix containing the rotation matrix $\mathbf{R}_{n_u}^{\tau}$. The rotation matrix and translation vector are obtained by calculating the singular matrix of the feature matrix, so as to realize the track transformation.

VII. Positioning With Bidirectional Bayesian Filter

A. Map Construction With Forward Bayesian Filter

After merging the tracks based on the labeling information, we estimate the relative positions of APs combining the information of multiple sampling points on each track. To avoid the sudden appearance or disappearance of some mobile WiFi APs, we only locate the APs that appear at each sampling point on the track. For these tracks, we can divide the long track into multiple tracks to estimate the AP's position. Here, we take the APs on track n_u and the corresponding RSSI data as an instance.

The RSSI value uses the log-normal attenuation model to indicate the distance $d_U^{n_A}(t_w)$ between the target and the anchor:

$$d_U^{n_A}(t_w) = 10^{\left(\frac{\left|s_{n_u}^{n_A}(t_w)-s_0\right|}{10\beta}\right)} + N_0 \tag{26}$$

where s_0 is the RSSI at 1 m, β is the attenuation coefficient, t_w is the WiFi sampling point, $s_{n_u}^{n_A}(t_w)$ is the RSSI of AP n_A detected by the user n_u at t_w, and N_0 is the noise following a zero-mean Gaussian distribution. We adopt $\mathbf{p}_A^{n_u}(n_A)$ to denote the real position of AP n_A in the environment detected by user n_u. Due to the lack of actual AP locations, $\mathcal{P}_3$ can be obtained by solving the following objective equation:

$$\min_{\mathbf{p}_A^{n_u}(n_A)} \sum_{t_w=1}^{T_w^{n_u}} \left(d_U^{n_A}(t_w) - \|\mathbf{p}_A^{n_u}(n_A) - \overline{\mathbf{X}}_{n_u}(t_w)\|\right)^2 \tag{27}$$

where $\overline{\mathbf{X}}_{n_u}(t_w)$ is the coordinate of the matched track n_u at the sampling point t_w. We combine multiple sampling points to localize the same AP. Due to NLOS and multipath effects, we introduce a set of initial particles $\{\mathbf{p}_{n_f}^{n_u}(n_A)\}_{n_f=1}^{N_f}$ representing the possible locations of AP n_A. Thus, the weight of each AP position is obtained from the conditional probability of the observations.

$$w_{n_f} \propto p\left(d_U^{n_A}(t_w) \mid \mathbf{p}_{n_f}^{n_u}(n_A)\right) \tag{28}$$

where $p(d_U^{n_A}(t_w) \mid \mathbf{p}_{n_f}^{n_u}(n_A))$ is the probability of $\mathbf{p}_{n_f}^{n_u}(n_A)$ occurring for a given observation $d_U^{n_A}(t_w)$. Then, the position of this AP is obtained based on the weights.

$$\widehat{\mathbf{p}}_A^{n_u}(n_A) = \sum_{n_f=1}^{N_f} w_{n_f} \mathbf{p}_{n_f}^{n_u}(n_A)) \tag{29}$$

Thus, we get the position of the AP from multiple tracks estimation.

B. Map Update With DBSCAN

However, some of the APs on the track are caused to deviate from the actual position by factors such as signal interference or track matching errors. Therefore, when fusing the same APs on different tracks, we apply DBSACN [40] to cluster the APs so as to eliminate the map construction errors caused by outliers. DBSCAN is a widely used clustering algorithm that identifies clusters in a dataset based on the concept of density. Compared to methods like K-means, DBSCAN recognizes clusters of arbitrary shapes and does not require the number of clusters to be specified in advance. Especially for constructing maps, it is suitable for handling data with complex and unknown structures and is robust to noise and outliers.

For a given AP n_A coordinate $\widehat{\mathbf{p}}_A^{n_u}(n_A)$ and radius ε, define its set of neighborhood points to be all points whose distance from $\widehat{\mathbf{p}}_A^{n_u}(n_A)$ is less than or equal to ε, denoted $\mathcal{N}(\widehat{\mathbf{p}}_A^{n_u}(n_A), \varepsilon)$.

$$\mathcal{N}\left(\widehat{\mathbf{p}}_A^{n_u}(n_A), \varepsilon\right) = \{o_1 \in D | f(\widehat{\mathbf{p}}_A^{n_u}(n_A), o_1) \leq \varepsilon\} \tag{30}$$

where $f(\widehat{\mathbf{p}}_A^{n_u}(n_A), o_1)$ represents the distance between $\widehat{\mathbf{p}}_A^{n_u}(n_A)$ and point o_1. According to $\mathcal{N}(\widehat{\mathbf{p}}_A^{n_u}(n_A), \varepsilon)$, we can define the core point with at least l other points in its neighborhood.

$$\mathcal{Q}\left(\widehat{\mathbf{p}}_A^{n_u}(n_A), \varepsilon, l\right) = \{\mathcal{N}(\widehat{\mathbf{p}}_A^{n_u}(n_A), \varepsilon) \geq l\} \tag{31}$$

A point o_2 is called directly density reachability from $\widehat{\mathbf{p}}_A^{n_u}(n_A)$, if point o_2 is in the ε neighborhood of $\widehat{\mathbf{p}}_A^{n_u}(n_A)$. Furthermore, if point o_2 does not satisfy $\mathcal{Q}(\widehat{\mathbf{p}}_A^{n_u}(n_A), \varepsilon, l)$, it is a boundary point. These core points are the centers of the clusters and they are connected by density reachability to form clusters. In contrast, those points that are neither core nor boundary points are considered as outliers or noise points, and they are usually regarded as outliers and do not belong to any clusters.

The clustering process of the DBSCAN is iterative, as described in Algorithm 3. Our goal is to find the effective clustering $\mathcal{U}$ of different tracks on the same AP by DBSCAN. Taking the example of multiple tracks all detected AP n_A, each track can estimate a coordinate $\widehat{\mathbf{p}}_A^{n_u}(n_A)$ corresponding to this AP according to (29). Due to the noise introduced by mobile sampling, these coordinates do not overlap to a single location. Therefore, DBSCAN is used to eliminate the outliers among them to improve the reliability of the map. The algorithm starts with an arbitrarily chosen unvisited point in the dataset and then finds all the points within its ε neighborhood. If it is a core point, then all points within its ε neighborhood are considered part of the same cluster. Next, we examine each point in the cluster and add all points with a reachable density to the cluster. This process will continue until all points have been visited.

After filtering outlier AP points using DBSCAN, the AP position $\overline{\mathbf{p}}_A(n_A)$ is determined as the average of the effective AP clusters $\mathcal{U}$. Finally, the generated radio map $\mathbf{R_M}$ is:

$$\mathbf{R_M} = \{\overline{\mathbf{p}}_A(1), \ldots, \overline{\mathbf{p}}_A(n_A), \ldots, \overline{\mathbf{p}}_A(N_A)\} \tag{32}$$

Algorithm 3: DBSCAN.

Input: Coordinates of the AP n_u on different tracks: $\widehat{\mathbf{p}}_A^{n_u}(n_A), n_u \in (1, N_u)$; Neighborhood radius: ε; Minimum Neighborhood Number: l
Output: Effective APs cluster: $\mathcal{U}$; Outliers: $\mathbf{O}$

```
for n_u = 1, 2, ..., N_u do
  if p̂_A^{n_u}(n_A) is not visited then
    Mark p̂_A^{n_u}(n_A) as visited;
    Calculate neighborhood points N(p̂_A^{n_u}(n_A), ε);
    if N(p̂_A^{n_u}(n_A), ε) < l then
      Add p̂_A^{n_u}(n_A) in O;
    else
      Add p̂_A^{n_u}(n_A) to effective cluster U;
      Add N(p̂_A^{n_u}(n_A), ε) to cluster U;
    end if
  end if
end for
Return effective APs cluster U and outliers O
```

C. Fusing Map-Based Positioning and PDR With Reverse Bayesian Filter

On map-based localization, we estimate the location of the target based on forward Bayesian filtering by using WiFi data. First, a random selection of N_s points $\{\mathbf{p}_w(n_s)\}_{n_s=1}^{N_s}$ in the region simulates the location of the user in the online phase. The weight of each simulated position is then proportional to the likelihood of the observation:

$$w_{n_s} = \sum_{n_A=1}^{N_A} p\left(d_U^{n_A} | \mathbf{p}_w(n_s)\right) \tag{33}$$

where $p(d_U^{n_A}|\mathbf{p}_w(n_s))$ indicates the probability with a give observation $d_U^{n_A}$ conditioned on $\mathbf{p}_w(n_s)$. The probability density function for $p(d_U^{n_A}|\mathbf{p}_w(n_s))$ is the zero-mean Gaussian distribution function. Then, we choose 10% of simulated points with the largest weights and form a new set $\{\mathbf{p}_w(n_s)\}_{n_s=1}^{N_s}$. We normalize these weights:

$$w_{n_s} = \frac{w_{n_s}}{\sum_{n_s=1}^{N_s} w_{n_s}} \tag{34}$$

When the iteration condition is satisfied, the estimated user coordinates are weighted average locations:

$$\widehat{\mathbf{p}}_w = \sum_{n_s=1}^{N_s} w_{n_s} \mathbf{p}_w(n_s) \tag{35}$$

The constructed maps are not accurate because they are estimated based on crowd sensing tracks. To further improve the localization accuracy, we use the reverse Bayesian filter fusion method to fuse the PDR with the map localization to obtain the final location of the user. Here, we use recursive dynamic estimation. First, we define the state vector $\widehat{\mathbf{p}}_U(t_w) = [\widehat{p}_U^x(t_w), \widehat{p}_U^y(t_w)]^T$, which denotes the user U position at time t_w.

The step length $L_U(t_w)$ and heading $\psi_U(t_w)$ estimated from the PDR algorithm at the current moment are used as the state equations:

$$\begin{bmatrix} \widehat{p}_U^x(t_w) \\ \widehat{p}_U^y(t_w) \end{bmatrix} = \begin{bmatrix} \widehat{p}_U^x(t_w - 1) \\ \widehat{p}_U^y(t_w - 1) \end{bmatrix} + \begin{bmatrix} L_U(t_w)\cos\psi_U(t_w) \\ L_U(t_w)\sin\psi_U(t_w) \end{bmatrix} + \begin{bmatrix} n_x(t_w) \\ n_y(t_w) \end{bmatrix} \tag{36}$$

where $[n_x(t_w), n_y(t_w)]^T$ is the noise generated by the motion process. The prior probability of the distribution of the position state $\widehat{\mathbf{p}}_U(t_w)$ at moment $t_w - 1$ is:

$$p\left(\widehat{\mathbf{p}}_U(t_w) \mid \widehat{\mathbf{p}}_U(t_w - 1)\right) = \frac{1}{\sqrt{(2\pi)^2 |\mathbf{Q}_n|}} \times \exp\left(-\frac{1}{2}\boldsymbol{\mu}^T \mathbf{Q}_n^{-1} \boldsymbol{\mu}\right) \tag{37}$$

where $\mathbf{Q}_n$ is the process noise covariance matrix, $\boldsymbol{\mu} = \widehat{\mathbf{p}}_U(t_w) - \Delta\widehat{\mathbf{p}}_U(t_w)$ and $\Delta\widehat{\mathbf{p}}_U(t_w) = \widehat{\mathbf{p}}_U(t_w - 1) + [L_U(t_w)\cos(\psi_U(t_w)), L_U(t_w)\sin(\psi_U(t_w))]^T$.

According to Bayes' theorem, Bayesian filtering corrects the predicted state when new observations are acquired. The previous section provides the WiFi observations $\widehat{\mathbf{p}}_w(t_w)$ based on map-based localization. We assume that the error of WiFi localization obeys a Gaussian distribution, and the observation model is:

$$p(\widehat{\mathbf{p}}_w(t_w)|\widehat{\mathbf{p}}_U(t_w)) = \frac{1}{\sqrt{(2\pi)^2 |\mathbf{R}_n|}} \times \exp\left(-\frac{1}{2}\boldsymbol{\nu}^T \mathbf{R}_n^{-1} \boldsymbol{\nu}\right) \tag{38}$$

where $\mathbf{R_n}$ is the observation noise covariance, which represents the error of WiFi localization, $\boldsymbol{\nu} = \widehat{\mathbf{p}}_w(t_w) - h(\widehat{\mathbf{p}}_U(t_w))$ and $h(\widehat{\mathbf{p}}_U(t_w))$ is the observation function obtained from the WiFi system;.

The posterior distribution is obtained by combining the prior distribution with the observation information:

$$p\left(\widehat{\mathbf{p}}_U(t_w)|\widehat{\mathbf{p}}_w(t_w)\right) \propto p\left(\widehat{\mathbf{p}}_w(t_w)|\widehat{\mathbf{p}}_U(t_w)\right) p\left(\widehat{\mathbf{p}}_U(t_w)|\widehat{\mathbf{p}}_U\left(t_w - 1\right)\right) \tag{39}$$

Finally, we use Bayesian filter posterior distribution for location estimation. This posterior distribution can be used to obtain the most probable user location by maximizing the posterior probability.

VIII. Experimental Evaluation

We have extensively evaluated the proposed SVD-CSP in real experiments and simulations. We conduct the experiments in two indoor scenarios respectively. Fig. 6(a) illustrates a section of the test scene in a teaching building, covering an area of 648 m^2. In addition, Fig. 6(b) illustrates the interior of a public laboratory building, covering an area of 2856 m^2. In Scenario 1 we use only a few tracks for small-scale track matching and a fine-grained comparison with the matching method based on ensemble analysis in WiFi-RITA [15]. Scenario 2 is used for

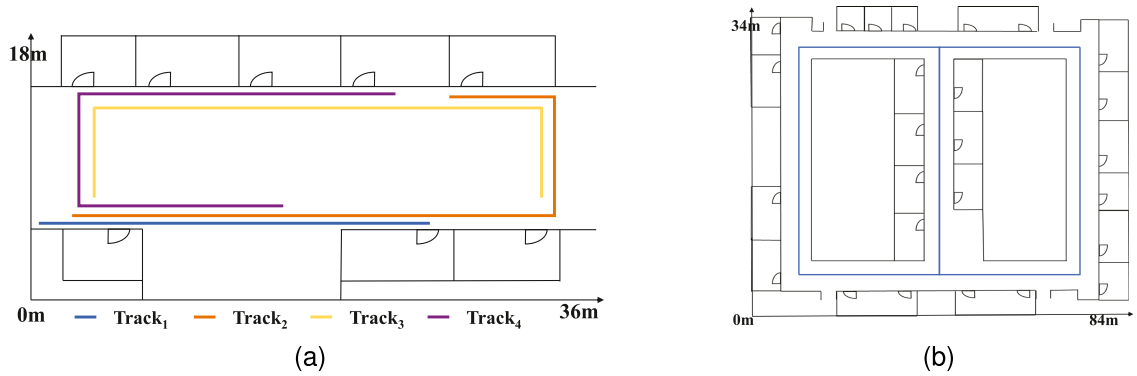

Fig. 6. Experimental Scenarios. (a) Scenario 1 walking paths. (b) Scenario 2 walking paths.

large-scale track matching, radio map construction and target localization testing. In order to reduce the influence of cumulative error on PDR, we control the sampling time of each track within 1 minute. In each scenario, no information such as landmarks is set, and it only relies on the shared IMU and detected WiFi. For limited length trajectories (sampling time less than 30s), shorter trajectories may contain fewer marker points due to the limited walking distance. At the initial stage of building radio maps, this may cause some fluctuation in the accuracy of localization. However, our system does not depend on the absolute length of trajectories. As crowdsourced data continue to accumulate, data from numerous short trajectories can complement each other.

To evaluate the accuracy of track matching and tracking more precisely, we place landmarks in the environment to obtain ground truth positions. Specifically, volunteers walk along these pre-set markers and are sampled so that we know exactly the actual path the volunteer is moving. In the track matching experiment, the volunteer takes the Redmi Note10 to move according to the four different tracks in Scenario 1, in order to test the effectiveness of SVD to find the rotation matrix to realize the track matching. In Scenario 2, the volunteer moves according to the test paths, in which the time of each track is within 1 minute. The initial point of each track is unknown, and the magnetometer and gyroscope are combined to generate the initial heading to distinguish the different tracks.

For user localization, the user walks at a constant speed on the walking path defined in Scenario 2 to obtain multiple tracks. In this case, all the WiFi RSSI data are timestamped and the distribution of APs is obtained based on the merged user tracks. Radio maps are formed based on DBSCAN filtering. Then the users are localized and tracked.

A. Track Matching Evaluation

Fig. 7 illustrates the effect of different methods on fusing the four walking paths in Scenario 1. In this experiment, a total of 160 WiFi APs are detected in the environment. After fusing RSSI peaks and magnetic field strength thresholds, 94 of these APs are identified as valid markers that could be used for trajectory matching. The color of each track is the same as in Fig. 6(a). The blue $\mathbf{Track}_1$ is used as the base track

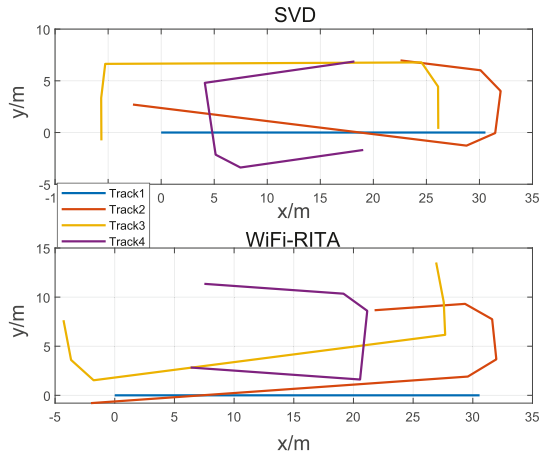

Fig. 7. Comparison of different track matching methods in Scenario 1. Compared to the bottom figure, track 3 and track 4 in the top figure are closer to the real trajectories.

in SVD, the matching of other tracks is realized sequentially, meanwhile the marker points information is updated after each track matching. Compared with the ground truth in Fig. 6(a), the heading directions are correct for all the tracks, and look similar to the real tracks. However, the estimated track 3 and track 4 are not correct based on WiFi-RITA. Although SVD has some rotational errors and positional offsets for small scale track matching, it is still desirable in the absence of actual marker assistance.

B. Positioning Accuracy

After track matching, we use the DBSCAN method to filter the estimated anomalous AP anchors on the tracks. As shown in Fig. 8, we set a threshold of 95% for the confidence interval to filter outliers in terms of Mahalanobis distance for each point. Note that, 7 outlier points are removed. Finally, the average value of all valid APs is taken as the virtual position of the current AP to obtain a radio map of the signal coverage.

In the absence of environmental absolute position landmarks, we evaluate the error between the relative distance of the two targets in Scenario 2 and the radio map localization results.

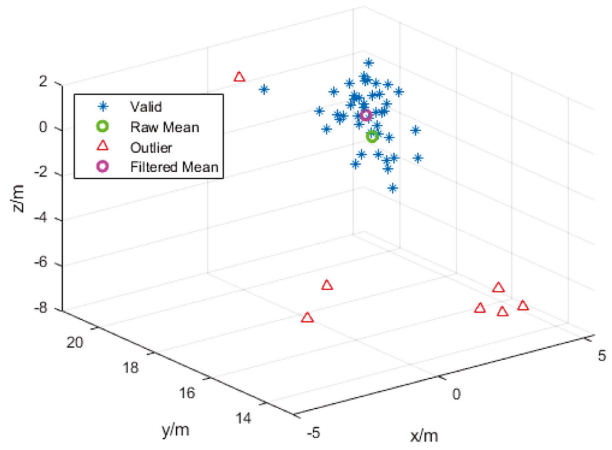

Fig. 8. Same AP filtering on multiple tracks. The red triangles are the rejected outliers, and the mean of the valid values is used as the location of this AP clustering.

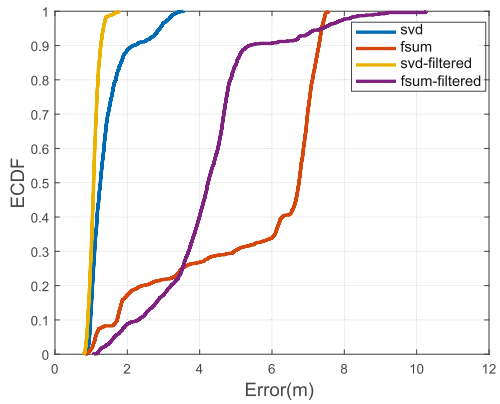

Fig. 9. Localization error ECDF with 30 tracks at 20 m.

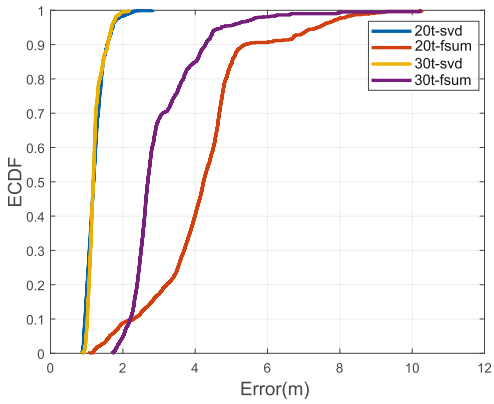

Fig. 10. Localization error ECDF for different number of tracks at 20 m.

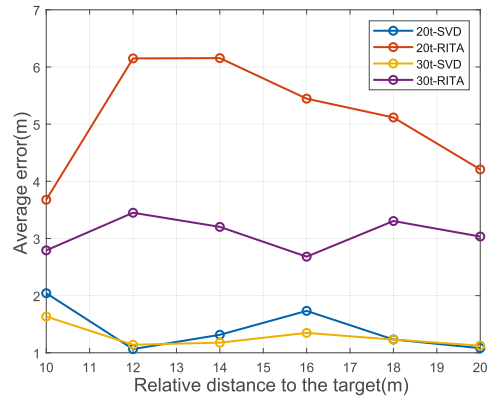

Fig. 11. Mean error at different distance.

From the perspective of the target visible distance, 10 m away is needed to achieve fast localization by position information, so we evaluate the localization performance at a separation distance of 10-20 m. Fig. 9 demonstrates the CDF of the localization error at a distance of 20 m using only 30 trajectories containing more information to construct the radio map. On one hand, the localization error of the SVD method at 20 m is basically within 4 m, which has better localization performance compared to WiRi-RITA. On the other hand, compared with the pre-filtering, the elimination of outlier APs makes the average localization error of SVD decreased by 0.35 m, and the average localization error of WiRi-RITA is decreased by 1.26 m. Therefore, the elimination of calibrated anomalous APs in independent trajectories using DBSCAN can effectively improve the localization accuracy in real scenarios.

Second, the problem of different numbers of tracks carrying different amounts of information as well as having errors of their own can have an impact on map construction as well as target localization. The CDF plots for target localization at a distance of 20 m with different numbers of tracks are presented in Fig. 10. The increase in the number of tracks only slightly improves the localization error of the SVD method when the targets are 20 m apart. On the contrary, for WiFi-RITA, adding only 10 tracks to build the map decreases the average localization error by 1.17 m. This demonstrates that our method achieves more stable localization with small data scales.

Fig. 11 draws the average localization error for different number of tracks under 10 m to 20 m. Each method uses radio maps filtered by DBSCAN for localization. As the number of tracks increases, the localization error of WiFi-RITA at different distances is basically stabilized at about 3 m, with a relative improvement of 66.53%. While the average localization error of the SVD method is 1.38 m, with a relative improvement of 17.45%.

Fig. 12 illustrates the probability distribution of different algorithms over 20 m localization error, including six different algorithms: SVD, WiFi-RITA, WKNN, KNN, Weighted Least Squares (WLS) [41], a combination of Gaussian Process Regression with KNN (GPR+KNN) [42], Residual Network 50 (ResNet50) [28] and generative adversarial network (GAN) [43]. Among them, for the fingerprint matching method, we use the calibrated sampling points in the mobile sampling as the database, so the error of localization is relatively large. Whereas, GPR can simulate the fingerprint expansion from nearby locations, so the localization error is smaller than the traditional fingerprint localization methods. In addition, sparse data leads to poor localization performance of deep learning methods.

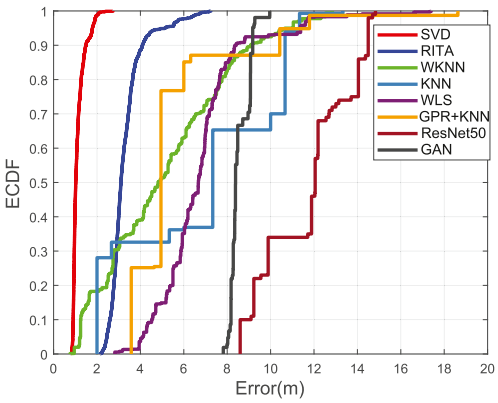

Fig. 12. Comparison of positioning errors of different methods at 20 m.

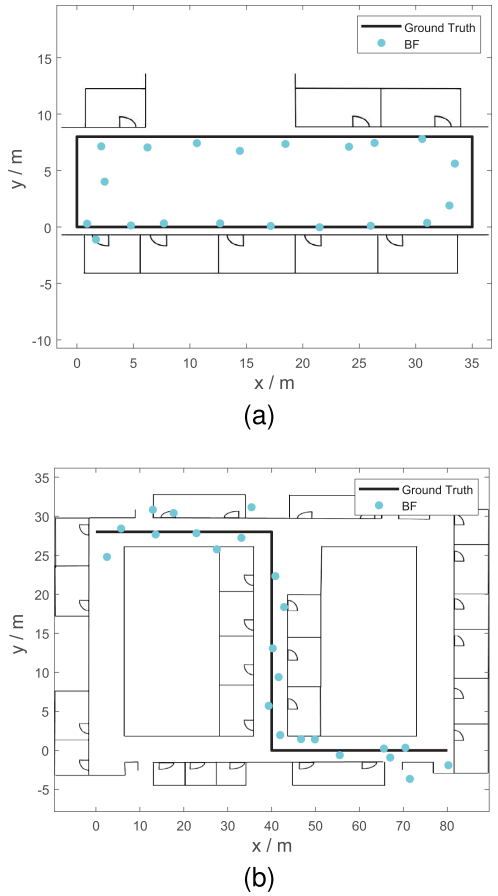

Fig. 13. Estimated points based on reverse Bayesian filter localization. (a) Scenario 1 positioning results. (b) Scenario 2 positioning results.

Compared with these methods, our localization method is more accurate.

Fig. 13(a) demonstrates the performance of the Bayesian filter localization in Scenario 1. The ground truth (the true track) is represented by a blue line and the points estimated by the Bayesian filter are represented by cyan dots. The Bayesian filter is able to track the real track better, although there are some deviations. These deviations may be due to noise in the environment. In most areas of the x-axis and y-axis, the Bayesian filter estimated points are consistent with the true track, which indicates that the algorithm has high accuracy when dealing with simple scenes. Fig. 13(b) displays the localization results of Scenario 2, which is a more complex environment containing more obstacles and possible sources of signal interference. In this environment, the Bayesian filter still performs satisfactorily, although the deviation between the estimated points and the true track is increased compared to Scenario 1. Especially near obstacles, the distribution of Bayesian filter estimated points shows the robustness of the algorithm in dealing with complex environments. However, we also observe an increase in the dispersion of the estimated points in some regions, which is due to the multipath effect of the WiFi signal and the cumulative error of the PDR data.

C. Parameter Analysis

Since the number of APs in the real environment is uncontrollable, in order to explore the impact of the number of APs on the overall performance of the system, we evaluate it by using random selected APs in Scenario 2. For this purpose, we select 30 tracks to explore the impact of the variation of AP numbers on the overall performance of the system. A total of 196 APs are detected on these 30 trajectories, however, only 103 APs can be used for track matching. Therefore we randomly choose a subset of APs among these 103 APs.

Fig. 14(a) illustrates the localization performance with the total used APs covering the whole scenario. When the number of APs increases from 20 to 60, the average error decreases from 14 m to 8 m. Note that track matching presupposes that the WiFi Marker attached to it has at least 3 different locations. When the number of APs is fewer, a part of the tracks fails to participate in the matching to build the radio map, so the localization error will be larger at this time. However, when the number of APs exceeds 60, the decrease in error starts to slow down, especially after the number of APs reaches 80, the decrease in error becomes low. In this case, all the trajectories can participate in the map construction normally. Therefore, to fully express the complete map information (especially the area of 2856 m^2) in the actual environment, at least 80 APs are needed to achieve the accuracy of 6m. However, the distribution of APs in the actual environment is not ideally uniformly dispersed, instead several APs are distributed in an area. Therefore, to achieve an accuracy of about 2m at least 90 APs should be involved.

Considering that there are only 90 APs distributed in the environment, the actual number of APs involved in target localization averages 13 for each position. Thus, we further explore the effect of different number of APs on each location, and the localization accuracy is illustrated in Fig. 14(b). As the number of APs involved in localization increases, i.e., the amount of effective information increases, the mean localization error gradually decreases. Only 12 APs are required to participate in localization to achieve 2 m localization accuracy. Since the AP distribution and density are uncertain in the real environment, if more than 13 effective APs are detected around the target, an accuracy of 1.5 m can be attained.

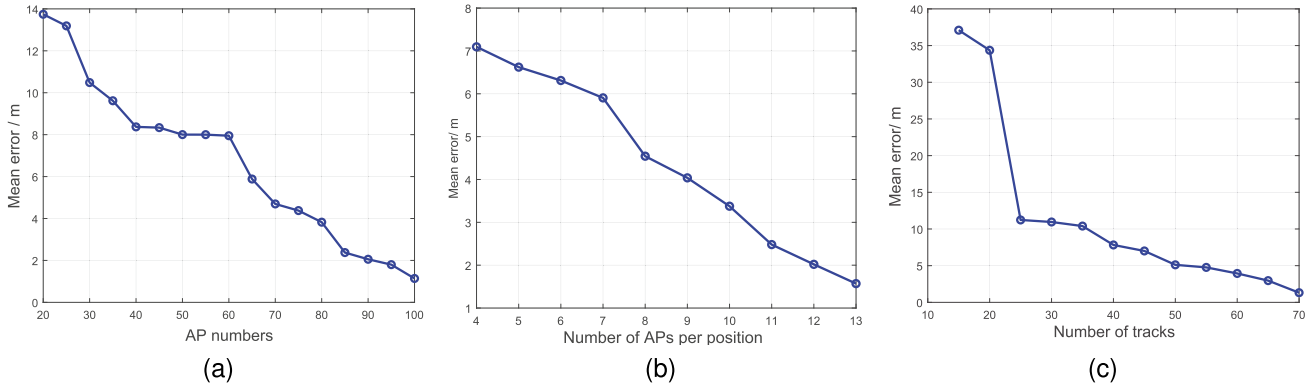

Fig. 14. System performance analysis. (a) Average localization errors for different number of APs in Scenario 2. (b) Mean localization error at different number of APs in the localization phase. (c) Average localization error at different number of tracks.

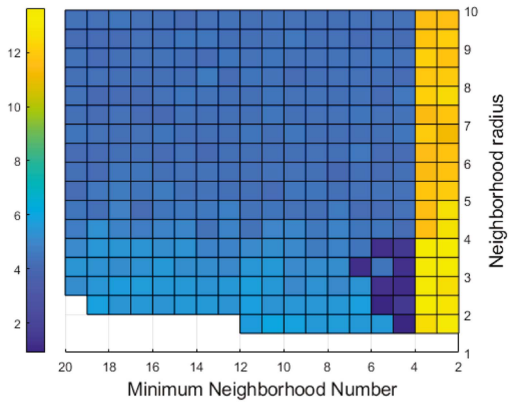

Fig. 15. Heatmap of DBSCAN parameter optimization.

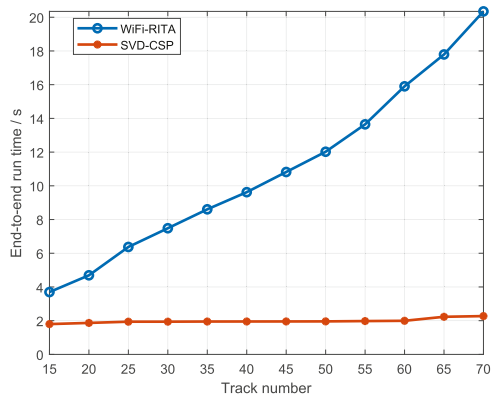

Fig. 16. End-to-end time for map construction.

We randomly resampled 70 tracks to obtain the localization errors in Fig. 14(c). Note that, most of the tracks are short in length, i.e., the sampling time is 50 seconds or less. With only 10 trajectories, the resampled trajectories are mostly scattered all over the place. In this case, it is hard to construct radio maps due to the lack of wifi markers for track matching, and that's why there are no values for trajectories below 10. Furthermore, when the number of trajectories increases from 15 to 30, the radio map can be constructed accurately, so the localization error decreases rapidly. As the number of trajectories further increases to 50 and above, the localization error reaches 5 m and the decreasing trend slows down. It means that after a certain number of tracks, the system's perception of the environment has stabilized. After that, the newly added trajectories are more useful for optimizing the radio map.

Fig. 15 presents the error heatmap of DBSCAN with different parameter values. The horizontal coordinate represents the minimum neighborhood number in DBSCAN, the vertical coordinate is the neighborhood radius, and the color shades indicate the localization error under different parameter combinations. When the minimum neighbor number is less than 4, some normal points are misjudged as noise points, resulting in the loss of some valid information, which makes the localization error larger. When the neighborhood radius is less than 1.5, it fails to form effective clusters, resulting in the creation of blank spaces in the figure. The profile coefficients are higher when the minimum number of neighbors is from 4 to 6 and the neighborhood radius is from 1.5 to 4, indicating better clustering under the combination of these parameters.

D. Complexity Evaluation

Fig. 16 demonstrates the end-to-end time from data preprocessing to radio map construction for SVD and WiFi-RITA. Both systems are written in MATLAB and run on a standard desktop computer with an Intel Core i7-11700H CPU and 32 GB of RAM. When the number of tracks increases from 15 to 70, the running time of SVD increases by only 0.47 s, which provides high computational efficiency and stability. In contrast, when the number of trajectories reaches 40 or more, the runtime of WiFi-RITA is more than 5 times that of SVD. This indicates that SVD is not only able to complete track matching quickly but also has lower time complexity in the case of a small number of tracks, which is especially important for application scenarios that require fast response. The run times for each phase are listed in Table II. First, the data pre-processing phase involves the calibration of the PDR and WiFi markers on the tracks. Although

TABLE II
COMPARISON OF RUNTIMES FOR DIFFERENT NUMBER OF TRACKS

Track number	Pre-processing time (s)		Track matching time (s)		Map construction time (s)		End-to-end time (s)		
	SVD	WiFi-RITA	SVD	WiFi-RITA	SVD	WiFi-RITA	SVD	WiFi-RITA	ResNet50
15	0.044	0.056	0.004	3.093	0.899	0.547	1.798	3.697	25.114
25	0.106	0.198	0.008	5.376	0.970	0.798	1.940	6.373	33.251
35	0.158	0.301	0.011	7.428	0.974	0.877	1.949	8.606	41.315
45	0.259	0.438	0.020	9.499	0.977	0.882	1.954	10.819	49.232
55	0.339	0.536	0.027	12.061	0.988	1.049	1.977	13.647	56.897
65	0.449	0.664	0.032	15.914	1.116	1.214	2.231	17.792	65.178

the pre-processing time increases slightly as the number of tracks increases, it generally stays low. The difference between the two methods is the marker selection. Second, the processing time for SVD track matching is relatively short and does not change much as the number of tracks increases. However, WiFi-RITA in track matching adopts an iterative approach in solving the optimal value, which has a longer processing time, and the increase in the number of tracks leads to an increase in the computational complexity. Finally, the time for map construction depends on the environment complexity, and the difference between the two methods is not significant.

For complexity comparison, one advantage of our SVD-CSP is the less used training tracks than deep learning model. Take Scenario 2 for instance, the sampled data of only 30 tracks is not enough for the deep learning model. In addition, we take ResNet50 [28] as an instance to discuss the computational complexity. The model takes RSSI, signal strength difference and kurtosis value constituting a 3D feature map as input. For 30 tracks with a sampling time of less than 1 minute, the average number of sampling points is considered to be 8. In our scenario, there are a total of 103 APs involved, while about 1/4 can be used as WiFi markers on each track. Therefore, the computational complexity of ResNet50 is $O_{\text{ResNet}}(50 \times 3 \times 103^3 \times 30 \times 8) = O_{\text{ResNet}}(3.95 \times 10^{10})$. The computational complexity of our system is mainly composed of SVD, DBSCAN and Bayesian filtering, i.e., $O_{\text{SVD-CSP}} = O_{\text{SVD}}(30 \times 25^3) + O_{\text{DBSCAN}}(103 \times \log(103)) + O_{\text{Bayesian}}(2^3 \times 100) = O_{\text{SVD-CSP}}(4.7 \times 10^5)$. Thus, $O_{\text{SVD-CSP}} \ll O_{\text{ResNet}}$. We also depict the overall runtime of ResNet50 for different number of trajectories in Table II. Furthermore, we analyze the memory footprints of the two and show that SVD-CSP occupies about 0.02 MB while ResNet50 occupies about 18.25 MB. This significant difference further emphasizes the applicability of SVD-CSP in resource-constrained environments, especially its feature of not requiring GPU support.

Finally, Fig. 17 exhibits the impact of the number of particles in the localization phase on the error and time. When the number of particles increases, the overall localization accuracy decreases. Particularly, when the particle number reaches 300, the average localization error can reach 2 m. It is stabilized with 350 particles. Correspondingly, the computational complexity increases, leading to an increase in the average localization time. When the number of particles exceeds 400, the localization phase will take more than 2s. Therefore, in order to balance the localization accuracy and time, the number of particles can be set between 300 and 400.

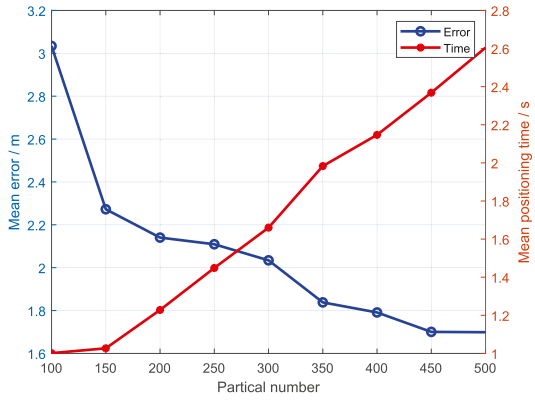

Fig. 17. Effect of particle number on localization performance.

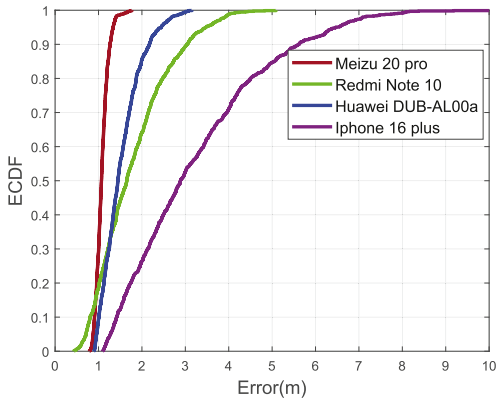

Fig. 18. Comparison of positioning errors of different cell phones.

E. Heterogeneity Analysis

We construct a new experiment for heterogeneous device evaluation. We employs more mobile phones in this experiment, which are Meizu 20 Pro, Redmi Note 10, Huawei DUB-AL00a, and iPhone 16 Plus. The operating systems cover iOS and Android. Considering the RSSI sampling frequency, the maximum allowed WiFi sampling rate of Android devices is 0.2 Hz. Thus, we set the WiFi sampling rate as 0.2 Hz for fairness. In addition, due to the privacy protection scheme of iOS, iPhone does not support access to real-time data of WiFi. Thus, another Android phone is used to collect WiFi data simultaneously with iPhone during the track, which is used as the emulation WiFi data for iPhone. However, this introduces some offset for data fusion.

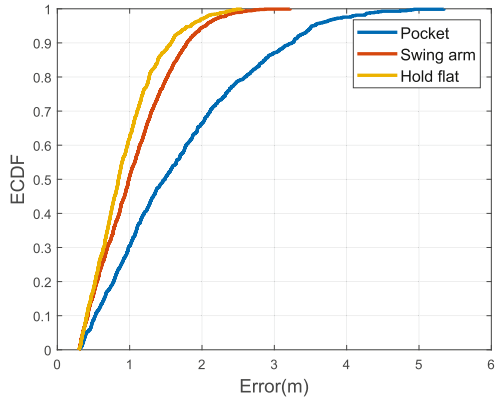

Fig. 19. Comparison of localization error by different holding styles.

The overall comparison results are illustrated in Fig. 18, where we use four phones for comparison. Among them, the error of the Android phones can approach within 3 m with high probability. However, the error of the iPhone 16 Plus is higher than the other phones. This due to the the offset of WiFi data timestamps, making the localization error large. In addition, the Meizu 20 pro offers a sensor module with better performance, resulting in a better overall positioning performance.

Fig. 19 illustrates the effect of three different ways of carrying the cell phone on the positioning error. Among them, pocket means that the phone is placed in a shirt or pants pocket, swing arm indicates that the hand which holds the phone swings naturally with walking, and flat hold is the way of carrying the phone with the screen facing up. The accuracy of swing arm and hold flat is similar, which outperforms the pocket. This dues to the severe drift of the IMU and the RSSI error of WiFi which is introduced by the pocket.

IX. Conclusion

In summary, we propose a crowd sensing radio map construction and localization system. In this system, the proposed SVD algorithm transforms the track matching into a problem of rotating and translating the tracks. Based on track matching, outliers with the same AP on multiple tracks are eliminated by the DBSACN method and the radio map is constructed. Finally, a localization method based on bidirectional Bayesian filter fusion of PDR and WiFi is developed. We evaluate the system in two different indoor scenarios. On the one hand, our proposed method has less error in track matching with smaller data scale. On the other hand, the average localization error of the target relying on only 30 tracks is in the range of 1.38 m. Thus, SVD-CSP has high localization accuracy, efficiency and stable performance relying only on small scale crowdsourced tracks, which outperforms the machine learning or deep learning based fingerprinting methods. In this paper, we only consider using the already crowd source data instead of continuing collecting the sensing data. Thus, the maintenance of the radio map is ignored. In the future work, we will focus on lightweight update scheme of the radio map and the fusion scheme with camera or barometers for 3D localization.

References

[1] C. Zhou, J. Liu, M. Sheng, L. Peng, D. Hou, and J. Li, "Fingerprint similarity indoor localization: An AP selection based approach," in *Proc. IEEE/CIC Int. Conf. Commun. China*, 2019, pp. 893–897.
[2] W. Sun, M. Xue, H. Yu, H. Tang, and A. Lin, "Augmentation of fingerprints for indoor WiFi localization based on Gaussian process regression," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10896–10905, Nov. 2018.
[3] X. Wang, X. Wang, S. Mao, J. Zhang, S. C. G. Periaswamy, and J. Patton, "Indoor radio map construction and localization with deep Gaussian processes," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11238–11249, Nov. 2020.
[4] Z. Li, T. Braun, and D. C. Dimitrova, "A time-based passive source localization system for narrow-band signal," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 4599–4605.
[5] B. Jang and H. Kim, "Indoor positioning technologies without offline fingerprinting map: A survey," *IEEE Commun. Surv. Tuts.*, vol. 21, no. 1, pp. 508–525, First Quarter 2019.
[6] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. ACM SIGMOBILE Int. Conf. Mobile Syst. Appl. Serv.*, 2012, pp. 197–210.
[7] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. ACM/IEEE Int. Conf. Mobile Comput. Netw.*, 2012, pp. 269–280.
[8] B. Zhou et al., "A graph optimization-based indoor map construction method via crowdsourcing," *IEEE Access*, vol. 6, pp. 33692–33701, 2018.
[9] V. Radu and M. K. Marina, "HiMLoc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced WiFi fingerprinting," in *Proc. Int. Conf. Indoor Positioning Indoor Navigation*, 2013, pp. 1–10.
[10] A. Rai, K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. ACM/IEEE Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.
[11] B. Zhou, Q. Li, Q. Mao, and W. Tu, "A robust crowdsourcing-based indoor localization system," *Sensors*, vol. 17, no. 4, pp. 1–16, 2017.
[12] C. Zhang, K. P. Subbu, J. Luo, and J. Wu, "GROPING: Geomagnetism and crowdsensing powered indoor navigation," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 387–400, Feb. 2015.
[13] C. Luo, H. Hong, and M. C. Chan, "PiLoc: A self-calibrating participatory indoor localization system," in *Proc. 13th Int. Symp. Inf. Process. Sensor Netw.*, 2014, pp. 143–153.
[14] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-Markie: Indoor pathway mapping made easy," in *Proc. Symp. Netw. Syst. Des. Implementation*, 2013, pp. 85–98.
[15] Z. Li, X. Zhao, Z. Zhao, and T. Braun, "WiFi-RITA positioning: Enhanced crowdsourcing positioning based on massive noisy user traces," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3785–3799, Jun. 2021.
[16] Q. Tian, Z. Salcic, K. I.-K. Wang, and Y. Pan, "A multi-mode dead reckoning system for pedestrian tracking using smartphones," *IEEE Sensors J.*, vol. 16, no. 7, pp. 2079–2093, Apr. 2016.
[17] Y. Yao, L. Pan, W. Fen, X. Xu, X. Liang, and X. Xu, "A robust step detection and stride length estimation for pedestrian dead reckoning using a smartphone," *IEEE Sensors J.*, vol. 20, no. 17, pp. 9685–9697, Sep. 2020.
[18] J. Kuang, X. Niu, and X. Chen, "Robust pedestrian dead reckoning based on MEMS-IMU for smartphones," *Sensors*, vol. 18, no. 5, pp. 1–18, 2018.
[19] J. Geng, L. Xia, J. Xia, Q. Li, H. Zhu, and Y. Cai, "Smartphone-based pedestrian dead reckoning for 3D indoor positioning," *Sensors*, vol. 21, no. 24, pp. 1–21, 2021.
[20] P. Chen, Z. Li, and S. Zhang, "Indoor PDR trajectory matching by gyroscope and accelerometer signal sequence without initial motion state," *IEEE Sensors J.*, vol. 23, no. 13, pp. 15128–15139, Jul. 2023.
[21] J.-S. Lee and S.-M. Huang, "An experimental heuristic approach to multi-pose pedestrian dead reckoning without using magnetometers for indoor localization," *IEEE Sensors J.*, vol. 19, no. 20, pp. 9532–9542, Oct. 2019.
[22] T. Yoshida, K. Urano, S. Aoki, T. Yonezawa, and N. Kawaguchi, "RGNet: Robust gravity estimation neural network for IMU-based localization using smartphone," in *Proc. 13th Int. Conf. Mobile Comput. Ubiquitous Netw.*, 2021, pp. 1–8.
[23] X. Li, Z. Deng, F. Yang, X. Zheng, L. Zhang, and Z. Zhou, "WiFi indoor location method based on RSSI," in *Proc. 11th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Syst.: Technol. Appl.*, 2021, pp. 1036–1040.
[24] X. Zhang and W. Sun, "Design of indoor localization algorithm based on WiFi fingerprints," in *Proc. China Automat. Congr.*, 2021, pp. 6209–6214.

[25] X. Zhang, Y. Xu, and Z. Yang, "KLDLoc: A WiFi fingerprint-based localization method based on Kullback-Leibler divergence," in *Proc. 5th Int. Conf. Commun. Eng. Technol.*, 2022, pp. 54–58.
[26] M. Luo, J. Zheng, W. Sun, and X. Zhang, "WiFi-based indoor localization using clustering and fusion fingerprint," in *Proc. 40th Chin. Control Conf.*, 2021, pp. 3480–3485.
[27] H. Zhang, K. Shen, and H. Sun, "Indoor localization system based on WiFi RSSI using wavelet-CNN architecture," in *Proc. 2023 China Automat. Congr.*, 2023, pp. 2305–2309.
[28] S. Deng, W. Zhang, L. Xu, and J. Yang, "RRIFLoc: Radio robust image fingerprint indoor localization algorithm based on deep residual networks," *IEEE Sensors J.*, vol. 23, no. 3, pp. 3233–3242, Feb. 2023.
[29] T. Lan, Y. Ye, and S. Zhang, "ZSFA: Zero-shot fingerprint augmentation for WiFi fingerprint based indoor localization," in *Proc. 14th Int. Conf. Wireless Commun. Signal Process.*, 2022, pp. 55–59.
[30] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter level localization using WiFi," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 269–282, Aug. 2015, doi: 10.1145/2829988.2787487.
[31] S. Eleftherakis, G. Santaromita, M. Rea, X. Costa-Prez, and D. Giustiniano, "SPRING: Smartphone positioning from a single WiFi access point," *IEEE Trans. Mobile Comput.*, vol. 23, no. 10, pp. 9549–9566, Oct. 2024.
[32] Z. Chen et al., "M^3M3: Multipath assisted Wi-Fi localization with a single access point," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 588–602, Feb. 2021.
[33] W. Gong and J. Liu, "SiFi: Pushing the limit of time-based WiFi localization using a single commodity access point," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 1, Mar. 2018, Art. no. 10, doi: 10.1145/3191742.
[34] X. Rao, Z. Luo, Y. Luo, Y. Yi, G. Lei, and Y. Cao, "MFFALoc: CSI-based multifeatures fusion adaptive device-free passive indoor fingerprinting localization," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 14100–14114, Apr. 2024.
[35] M. H. Mahmoud, S. N. Shoudha, M. Abdallah, and N. Al-Dhahir, "OpenPose-inspired reduced-complexity CSI-based Wi-Fi indoor localization," *IEEE Commun. Lett.*, vol. 28, no. 9, pp. 2066–2070, Sep. 2024.
[36] J. Shang, F. Gu, X. Hu, and A. Kealy, "APFiLoc: An infrastructure-free indoor localization method fusing smartphone inertial sensors, landmarks and map information," *Sensors*, vol. 15, no. 10, pp. 27251–27272, 2015.
[37] D. Philipp et al., "MapGENIE: Grammar-enhanced indoor map construction from crowd-sourced data," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2014, pp. 139–147.
[38] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *J. Glob. Positioning Syst.*, vol. 3, pp. 273–279, 2004.
[39] J. Solà, "Quaternion kinematics for the error-state Kalman filter," 2015, *arXiv: 1711.02508*.
[40] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 226–231.
[41] Y. Zhou, T. Qiu, F. Xia, and G. Hou, "N-times trilateral centroid weighted localization algorithm of wireless sensor networks," in *Proc. Int. Conf. Internet Things 4th Int. Conf. Cyber Phys. Social Comput.*, 2011, pp. 351–357.
[42] D. D. Nguyen and M. T. Le, "Enhanced indoor localization based BLE using Gaussian process regression and improved weighted kNN," *IEEE Access*, vol. 9, pp. 143795–143806, 2021.
[43] S. A. Junoh and J.-Y. Pyun, "Enhancing indoor localization with semi-crowdsourced fingerprinting and GAN-based data augmentation," *IEEE Internet Things J.*, vol. 11, no. 7, pp. 11945–11959, Apr. 2024.

Xiaohao Liu (Graduate Student Member, IEEE) received the bachelor's degree from Sun Yat-sen University, in 2023. He is currently working toward the master's degree in integrated circuit engineering with the School of Microelectronics Science, Sun Yat-sen University, Zhuhai 519082, China. His main research interests include infrastructure-free indoor localization and its applications.

Yubin Zhao (Senior Member, IEEE) received the BS and MS degrees from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2007 and 2010 respectively, and the PhD degree in computer science from Freie Universität Berlin (FU Berlin), Berlin, Germany, in 2014. He joined Center for Cloud Computing as an associate professor, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, in 2014. He is now the associate professor with the School of Microelectronics Science and Technology, Sun Yat-Sen University, Zhuhai, China. He serves as the guest editor and reviewer for several journals. His current research interest includes wireless power transfer, indoor localization and target tracking. He was a recipient of IEEE distinguished service award in IEEE SmartData 2023 and the Excellent Teacher Award of Collage Computing Science, in China, 2023. He serves as the vice technical chair in IEEE SmartData 2023, IEEE ScalCom 2022, publicity chair of IEEE NFV-SDN 2019 and tutorial chair of IEEE NFV-SDN 2020.

Xiaofan Li (Senior Member, IEEE) received the BS and PhD degrees from the Beijing University of Posts and Telecommunication, in 2007 and 2012, respectively. From 2010 to 2011, she studied with the University of Washington as an exchanged PhD student. She joined the State Radio Monitoring Center and Testing Center (SRTC) from 2012 and was transferred to SRTC Shenzhen Lab from 2013. She is currently an associate professor with the School of Intelligent Systems Science and Engineering, Jinan University, Zhuhai, China. Her research interests include interference analysis among different radio systems, testing and evaluation methods for innovative radio technologies, cooperative communication, cognitive radio, Internet of Things, radio management strategy, etc.

Huaming Wu (Senior Member, IEEE) received the BE and MS degrees from the Harbin Institute of Technology, China, in 2009 and 2011, respectively, both in electrical engineering, and the PhD degree in the highest honor in computer science from Freie Universitt Berlin, Germany, in 2015. He is currently an associate professor with the Center for Applied Mathematics, Tianjin University, China. His research interests include model-based evaluation, wireless and mobile network systems, mobile cloud computing and deep learning.

Cheng-Zhong Xu (Fellow, IEEE) received the PhD degree from the University of Hong Kong, in 1993. He is currently a chair professor in computer science with the University of Macau, China. Prior to that, he was with the Faculty of Wayne State University and Shenzhen Institutes of Advanced Technology of CAS. His recent research interests are in cloud and distributed computing, systems support for AI, smart city and autonomous driving. He published two research monographs and more than 400 journal and conference papers and received more than 13000 citations. He was a Best Paper Awardee or Nominee of conferences, including HPCA'2013, HPDC'2013, Cluster'2015, ICPP'2015, GPC'2018, UIC'2018, and HPBD&IS'2019. He was also a co-inventor of more than 120 patents and a co-founder of Shenzhen Institute of Beidou Applied Technology. He serves or served on a number of journal editorial boards, including *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Parallel and Distributed Systems*, *Journal of Parallel and Distributed Computing*, *Science China*, and *ZTE Communication*. He was the chair of IEEE Technical Committee on Distributed Processing from 2015 to 2020. He was a recipient of the Faculty Research Award, Career Development Chair Award, and the President's Award for Excellence in Teaching of WSU. He was also a recipient of the "Outstanding Oversea Scholar" award of NSFC.