IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 4, 2018, accepted May 4, 2018, date of publication May 15, 2018, date of current version July 6, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2836396

A Graph Optimization-Based Indoor Map Construction Method via Crowdsourcing

BAODING ZHOU[1,2,3,4], QINGQUAN LI[1,3,4], GUANXUN ZHAI[3,4], QINGZHOU MAO[5], JUN YANG[3,4], WEI TU[3,4], WEIXING XUE[6], AND LONG CHEN[7]

[1]College of Civil Engineering, Shenzhen University, Shenzhen 518060, China
[2]Institute of Urban Smart Transportation and Safety Maintenance, Shenzhen University, Shenzhen 518060, China
[3]Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen 518060, China
[4]Key Laboratory for Geo-Environment Monitoring of Coastal Zone of the National Administration of Surveying, Mapping and GeoInformation, Shenzhen University, Shenzhen 518060, China
[5]State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, China
[6]School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China
[7]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China

Corresponding author: Qingquan Li (liqq@szu.edu.cn)

This work was supported in part by the National Key R&D Program of China under Grant 2016YFB0502203, in part by the National Natural Science Foundation of China under Grant 41701519, in part by the Natural Science Foundation of Guangdong Province under Grant 2017A030310544, in part by the Open Research Fund Program of State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing under Grant 16I02, and in part by the Shenzhen Future Industry Development Funding Program under Grant 201507211219247860.

ABSTRACT Indoor mapping is an essential element of indoor navigation systems. In this paper, we present a graph optimization-based method for indoor map construction, which can be used in buildings without prior knowledge. By using crowdsourcing data on mobile phone sensors, we can derive activity landmarks where people perform different activities (turning, taking the elevator, taking the escalator, and walking up/down the stairs). Our method uses graph optimization techniques to align crowdsourcing trajectories by their intersecting landmarks. After trajectory alignment, this method applies a pose graph optimization method to construct an indoor map. Finally, our method performs a transform from relative coordinates to an absolute coordinate with reference points and reduces the redundant segments using dynamic time warping. To evaluate the performance, we implement the proposed method in an office building and a shopping mall. Experiment results show that 80th percentile error of the mapping accuracy is about 1.7–3.5 m. Moreover, the proposed method can deal with the curved routes in a building and can also decrease the amount of required data.

INDEX TERMS Indoor mapping, graph optimization, crowdsourcing, smartphone.

I. INTRODUCTION

Indoor map plays an important role in indoor navigation systems [1]. For example, an indoor map is needed to show a user where he/she is, and to show where a point of interest (POI) is. In some indoor localization systems, an indoor map is required to constrain the drift of localization errors [2], [3]. Generally, to obtain the indoor map of a building, the blueprint is needed that is then converted into digital formats. However, blueprints are generally inaccessible because they may belong to different owners, who are often reluctant to share them with the public. Moreover, a building's internal structure usually changes over time, making the original map outdated. Although it is possible to manually construct an indoor map, it is very labor-intensive and time-consuming. Therefore, indoor map construction and updating are normally difficult.

To reduce the difficulty, many solutions for indoor map construction have been proposed. Most approaches are based on the principle of Simultaneous Localization and Mapping (SLAM). SLAM is a well-known technique in the robotic domain which is usually used to solve the problem of locating a robot in an unknown environment while simultaneously building the map [4]. Furthermore, crowdsourcing is often used for indoor map construction [5]–[9]. The crowdsourcing method has been successfully applied to OpenStreetMap (OSM) for outdoor map construction, which uses Global Positioning System (GPS) for localization.

However, GPS is not operational indoors. Therefore, indoor map construction must rely on other localization methods.

In this paper, we propose a graph optimization-based indoor map construction method via crowdsourcing. The proposed method combines the ideas of SLAM and crowdsourcing, and uses SLAM techniques to fuse crowdsourced trajectories collected by different users. The proposed method is based on two observations. The first observation is that the indoor map can be characterized using a link-node model [8], [10] in which pathways are the links and the intersections are the nodes. The link-node model is usually adequate for indoor navigation because it provides a natural framework for locating users and points of interest (POIs). Moreover, the link-node model can also be used for indoor map change detection. The second observation is that the nodes of the link-node model are normally turns, elevators, and stairs. When people pass the nodes, they usually take different activities other than walking. These activities can be detected by the smartphones [3], [11]. In addition, the same node has similar context information, such as the WiFi fingerprint.

In SLAM framework, loop closure is an important part in back-end optimization. When mobile targets receive a signal that a current location has been visited, the system calibrates the measures in the previous locations, trying to minimize the error generated by sensors. A typical method to recognize loop position is using a bag of visual words [12]. However, such a visual method is not applicable when the device is not equipped with a camera. In this paper, we use the graph optimization technique to merge the crowdsourcing trajectories. This is the main innovation of this study.

By activity detection, we can identify some specific activities, and mark their locations along trajectories as loop positions. As following, we implement a two-step optimization based method to accomplish mapping task. The data is preprocessed by activity detection techniques to extract loop position points. Then, to align the trajectories with blind initial position, a graph model is built to optimize the transformations that changes the different relative coordinate system into a uniform global system. Each node in this graph represents a transforming matrix, which is used to transform the trajectory coordinate from its own coordinate system into a global coordinate system. Moreover, the edges of this graph are the errors of the transformed results. The next step is to optimize each of the position measurements to construct a dense point map. By implementing the pose graph optimization (PGO), the measures scattered in between the activity-related positions converge near their corresponding locations. PGO here is the problem of estimating a set of poses from pairwise relative measures [13]. Another further step of optimization can be done by using Dynamic Time Warping algorithm [14], when the constructed map has overlapping segments. This steps introduces some noise but can compress the data efficiently, which is better for further research.

The main contribution of this paper is to propose a graph optimization method to construct an indoor map, using pedestrian activities and context information instead of actual blueprints or other construction information. The proposed indoor mapping method can take advantage of the crowdsourcing data collected by the smartphones. The crowdsourcing mode significantly reduces the cost of indoor map construction and updating.

The rest of this paper is organized as follows. Section II reviews related studies. Section III introduces the indoor map construction methodology. Section IV gives the experiments result, and Section V is the conclusion.

II. RELATED WORK

Many researches have designed indoor mapping approaches by using multi sensor data collected by smartphones. Based on activities detection techniques, inertial sensed activities are used as the landmark for indoor map construction [15]–[17]. These activity landmark-based approaches leverage special activities in indoor environments (such as stairs, elevators, and corners) to reduce the accumulative error of dead reckoning. iFrame uses grid map technique to fuse inertial and radio frequency (RF) signals for indoor map construction [18]. The status of the grid is determined by dead reckoning, Bluetooth and WiFi detections.

Simultaneous Localization and Mapping systems allow long-term mapping in unknown environments without installing infrastructures or using a prior map [19], which is usually used for indoor mapping. Many state-of-art SLAM frameworks implement visual and laser-based odometry. However, such methods require expensive equipment, or violate privacy in some scenarios. Thus, using inexpensive and non-visual sensors is a possible substitute. In [20], foot-mounted Inertial Measurement Unit (IMU) and smartphone are used to construct an ActionSLAM system coherently. The idea is to use user activity such as sitting or standing still, which indicate probable moving patterns, to mark a location. After action landmarks are obtained, a SLAM mapping method is applicable. Despite the advantage of accuracy, foot-mounted IMU require specialized devices and testing process. To apply in more generalized user cases, we consider using only mobile phones and embedded sensors such as gyroscope and magnetometer, to accomplish such a mapping technique.

In visual or laser-based SLAM systems, mapping is done through spatial correlations deduction between surroundings and agents' moving patterns. However, inexpensive sensors normally are incapable of sensing surroundings. To estimate trajectory from sensor data, Pedestrian Dead Reckoning (PDR) algorithm can be applied to derive trajectories.

Because of the error introduced by phone sensors, the trajectory data derived from sensor needs to be optimized to gain better results. In [21], optimization is conducted using both moving patterns of agents (specifically robots in this study) and observable reference points. This method is usually referred to as bundle adjustment (BA), which now has become a state-of-art method in SLAM. As sensor data in the proposed is obtained by the inertial sensors, observation of surroundings would not be possible and BA would not be applicable. Because of this reason, PGO model is required. Pose graph here means a temporal sequence of agents' poses (including position and heading direction in 2D scenes) [13]. By using PDR, we can obtain some pose graphs. In this case, pose graph refers to pedestrian step sequence derived by PDR.

Because pedestrian trajectories are inherently stochastic [22], the optimization of the pose graph should be non-linear. A typical non-linear optimization method is the Extended Kalman Filter (EKF). Because EKF uses Tylor Expansion for non-linear approximation, it would have optimistic estimates failure [23]. Therefore, we can use graph optimization for such a problem [24]. Graph optimization uses a graph model to describe the optimization problem. In such a graph, nodes represent the parameters, and edges represent estimating errors. For our case, graph optimization is specified as a PGO, which has no landmarks for reference.

Trajectories derived from pedestrian data is a group of isolated, single-person behaviors. One single trajectory cannot cover the whole scene, which makes crowdsourcing method applicable. However, implementing crowdsourcing has a problem, that is, trajectories are initialized blindly. Gu *et al.* [25], [26] used Wi-Fi and iBeacons (Bluetooth) for trajectory alignment. A particle filter is used to filter trajectories derived from foot-mounted Inertial Measurement Unit (IMU). In addition, a Gaussian Process regression model is used to adapt Wi-Fi and iBeacons' Received Signal Strength (RSS) acquired along the trajectories. Zhou *et al.* [27] used a sequence of Wi-Fi signal along moving patterns instead. However, these methods do not make full use of sensor data. Moreover, they are not applicable if no such RF devices are installed or if signals are too weak.

Similarly, in Cooperative-SLAM [28], [29], such an initialization problem exists as well. Kim et al. designed a new algorithm for cooperative mapping by integrating multiple relative pose graph of robots [28]. When robots encounter, or observe the same target, constraints derived by the observation are added to the pose graph accordingly. By optimizing the integrated graph, a global map can be constructed. Another insight from this work is the anchor, meaning the offset of the complete trajectory with respect to a global coordinate frame.

In these previous indoor mapping methods, ALIMC [8] is the most similar to this paper. In ALIMC, the map was derived by a link-node mode using Multi-Dimensional Scaling (MDS) techniques. MDS uses the straight-line distance between two nodes for map construction. Therefore, ALIMC can not solve the curves in the map.

III. METHODOLOGY

A. SYSTEM OVERVIEW

The overview of the proposed indoor map construction method is shown in Fig. 1. The proposed indoor map construction method is based on the crowdsourcing data collected by the built-in sensors of a smartphone. The crowdsourcing data includes motion data and WiFi fingerprint. The motion data includes heading, angular velocity, and acceleration. The WiFi fingerprint includes the MACs of the surrounding Access Points (APs) and the corresponding Received Signal Strength (RSS) values.

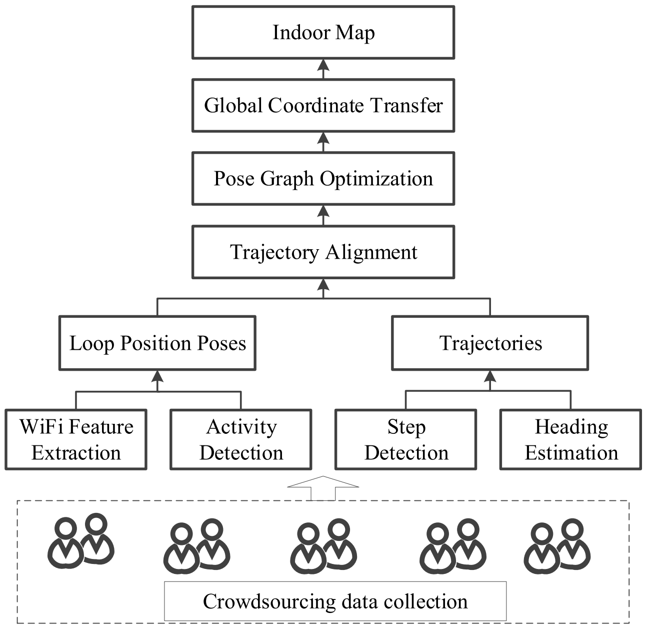


FIGURE 1. System overview.

A methodology description is given as follows: Our method is based on sensor data acquired from mobile devices using crowdsourcing methods. These data are processed by PDR algorithm and Activity Detection algorithms. After processing, we obtain groups of trajectories, and some landmarks scattered in these trajectories. The landmarks are pedestrian position at specific infrastructures or locations, such as elevators, stairs, turns. For simplicity, we denote these landmarks as Loop Position Poses, abbreviated as LPP.

The following are two steps of optimization: trajectory alignment and PGO. Trajectory alignment step is to unify the coordinate systems of different trajectories, because each trajectory has a relative coordinate system. As one single LPP might be detected in different trajectories, we may use such an intersected relation of different trajectories to align all measures derived from sensor data. After alignment, the temporary map is very noisy. Next, we use all pose measurements to apply PGO techniques. Eventually, we can automatically construct a map in a global coordinate system that reflects the ground truth. Intuitively speaking, in this method, we use some back-end techniques of SLAM (simultaneous localization and mapping) systems [19] to construct a dense point map based on mobile sensor data.

Note that in the second optimization step, two constraints are used, the inner constraints, which is the relative position relation of neighbor poses, and the outer constraints, which is the intersected LPP of different trajectories.

B. CROWDSOURCING DATA PREPROCESSING

Crowdsourcing data preprocessing includes Loop Position Pose detection and trajectory estimation. LPP is the landmarks where pedestrians take special activities. For example, in Fig. 2, there is a trajectory, the circle dots indicate the LPP that can be detected. We use position information related to these activities to generate LPP for optimization. In an indoor environment, typical activities include turning, taking the elevator, taking the escalator, walking the stairs, and so on. Turning is a normal activity during the walking process. When a pedestrian turns, the angle velocity changes, and the gyroscope data can be used for turning detection. Generally, an elevator use trace includes an overweight and a weightlessness moment, which can be used for detecting the activity of taking the elevator. Moreover, the feature of air pressure can be also used for elevator detection, which can be measured by the barometer. Similarly, air pressure changes when a pedestrian takes the escalator. However, there is no overweight or weightlessness moment when a pedestrian takes the escalator. All these activities can be detected based on the data collected by the built-in sensors of a smartphone. Here we use the activity detection method proposed in [3] to detect the activities. Trajectory estimation is realized by PDR algorithm [30]. PDR derives a pedestrian's trajectory based on step detection and heading estimation results.

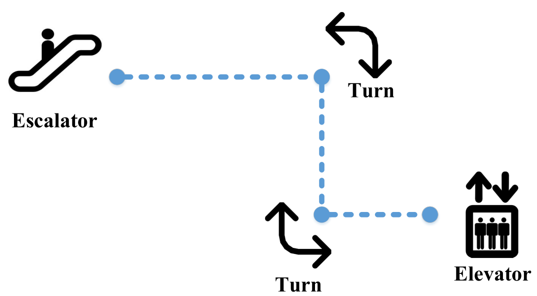


FIGURE 2. **LPP definition.**

By activity detection, we obtain the LPPs in the trajectories and the context information, namely the WiFi fingerprint. Then, we obtain pairs of LPP by LPP clustering. A pair of LPP refers to the index of an LPP in local trajectory with index among clustered locations of the same LPP. The clustering algorithm proposed in ALIMC [8] is used for LPP clustering. The clustering algorithm is a two-step process: first, the LPPs are roughly clustered based on the WiFi fingerprint, and then the spatial information is used to improve the clustering accuracy. For details of the LPP clustering algorithm, please refer to reference [8]. After activity detection and LPPs clustering, we obtain the pairs of LPP in the trajectories.

C. TRAJECTORY ALIGNMENT

In pedestrian trajectories, because of vacancy of actual azimuth of the starting pose, coordinates of all measures are in a non-consistent coordinate system. This leads to an initialization problem as described in Fig. 3. Suppose we have two trajectories, marked green and red, respectively. We can see that the green one passed LPP A, B, and red one passed LPP A, B, C, D in Fig. 3(a). Because crowdsourced data might not have a well-calibrated compass data, the azimuth of two trajectories is non-consistent. Besides, the starting poses are both initialized at origin, which is not applicable in an absolute coordinate system. Because of these two reasons, the derived trajectories by smartphones would be like in Fig. 3(b).

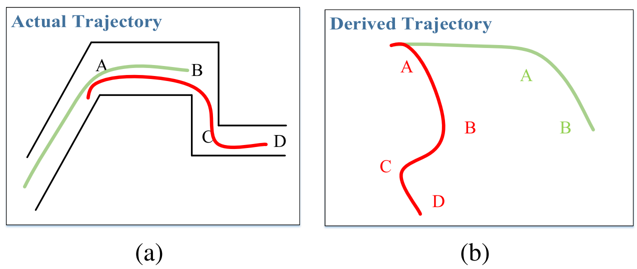


FIGURE 3. **Trajectory alignment. (a) Actual trajectories; (b) Derived trajectories by smartphones.**

To calibrate these inconsistent measures of identical LPP in different trajectories, a trajectory alignment step is required. The transform matrix for alignment is calculated based on the intersected LPP of different trajectories. This matrix contains both translation and rotation variables. Because of the random noise, there is no unique solutions for such a matrix. Therefore, we solved this problem in an optimization manner.

As stated above, trajectory alignment step is used to calculate the optimized transform matrix that can align different trajectories. Consider we have an LPP as Pos_k, which is a 2D coordinate. Relative coordinate of M in the trajectory i is defined as:

$$Pos_k^i = (x_k^i, y_k^i)^T \tag{1}$$

We can further denote Pos_k^i by LPP LPP_l^i and non-LPP $NLPP_n^i$, and suppose we have two trajectories I, J, I has m_i poses, where l_i poses are LPP, J has m_j poses, where l_j poses are LPP.

$$\begin{aligned} I = LPP_1^i,\quad LPP_2^i, \ldots, LPP_{l_i}^i,\ NLPP_1^i,\ NLPP_2^i,\\ \ldots, NLPP_{n_i}^i | l_i + n_i = m_i \end{aligned} \tag{2}$$

$$\begin{aligned} J = LPP_1^j,\quad LPP_2^j, \ldots, LPP_{l_j}^j,\ NLPP_1^j,\ NLPP_2^j,\\ \ldots, NLPP_{n_j}^j | l_j + n_j = m_j \end{aligned} \tag{3}$$

And we define $I \sqcap J$ as:

$$\begin{aligned} I \sqcap J = LPP_{b_1}^i,\quad LPP_{b_2}^i, \ldots, LPP_{b_p}^i,\ LPP_{d_1}^j,\ LPP_{d_2}^j,\\ \ldots, LPP_{d_p}^j | 1 < b_1, \ldots, b_p < l_i, \quad 1 < d_1, \ldots, d_p < l_j \end{aligned} \tag{4}$$

$LPP^i_{b_p}$ and $LPP^j_{d_p}$ are coordinates close to an identical unique location in trajectory I and J. We define them as a pair of intersected LPP, similar to other pairs. b_p and d_p are indexes of these two measures in each corresponding trajectory data. Totally, I and J has p pairs of intersected LPP. Here, the result $I \sqcap J$ indicates relative coordinates of all intersected LPPs in trajectory I and J. We use this special symbol to indicate that these intersected poses are different in the two trajectories' relative coordinate system.

Suppose here we want to align a new trajectory I to an already aligned trajectory J. Then we use two variables:

$$\begin{aligned} X_i &= LPP^i_{b_1}, LPP^i_{b_2}, \ldots, LPP^i_{b_p} \\ X_j &= LPP^j_{d_1}, LPP^j_{d_2}, \ldots, LPP^j_{d_p} \end{aligned} \tag{5}$$

And a transform variable $T_{ij} = [T_x, T_y, \theta]$ is used to transform trajectory I into the coordinate in J. To calculate the transformed coordinates, it is better to use a matrix notation, here we define an arbitrary function $M(T_{ij})$ as:

$$M(T_{ij}) = \begin{bmatrix} \cos\theta & -\sin\theta & T_x \\ \sin\theta & \cos\theta & T_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

To avoid problems of ambiguous initialization, we align trajectories incrementally. In each iteration, a new trajectory data is loaded into the program. The aligned trajectories in previous iterations make up a map, which can also be represented as a much longer trajectory. A transform T_j is calculated to align the new trajectory to the fused map.

Note here if we want to multiply $M(T_{ij})$ by X_i, we need to extend X_i with a Vector of ones as:

$$\overline{X_i} = ext(X_i) = \begin{bmatrix} LPP^i_{b_1} & \ldots & LPP^i_{b_p} \\ 1 & \ldots & 1 \end{bmatrix} \tag{7}$$

As a supplement, we denote $X_i = sub(\overline{X_i})$, which indicates that X_i has one less row of ones than $\overline{X_i}$. These definitions are used to transform the variables to an addable expression.

The error and total error function in one iteration would be:

$$\begin{aligned} e_j(T_j) &= X_j - sub(M(T_j)) \cdot \overline{X_i} \\ F(T) &= e_j^T(T_j)\Omega_j e_j(T_j) \end{aligned} \tag{8}$$

Suppose we have K trajectories in total, then $K-1$ iterations is needed. Ω_j is the information matrix. The target of optimization is to find T that minimize this error function:

$$T_j^* = argminF(T_j) \tag{9}$$

Our target is to use a standard non-linear optimization method such as Gauss-Newton or Levenberg-Marquardt (LM) [31]. Suppose we have an initial guess of $\overline{T_j}$ and an incremental value $\triangle T_j$. We may use the Taylor expansion and express the solution as:

$$\begin{aligned} T_j^* &= \overline{T_j} + \triangle T_j \\ e_j(\overline{T_j} + \triangle T_j) &= e_j(T_j) + J_{ij}\triangle T \end{aligned} \tag{10}$$

And the error function is expanded as:

$$\begin{aligned} F(\overline{T} + \triangle T) &= e_j^T(\overline{T} + \triangle T)\Omega_j e_j(\overline{T} + \triangle T) \\ &= e_j^T(T_j)\Omega_j e_j(T_j) + 2e_j^T(T_j)\Omega_j J_{ij}\triangle T \\ &\quad + \triangle T_j^T J_{ij}^T J_{ij}\triangle T \\ &= c_j + 2b_j\triangle T_j + \triangle T_j^T H_j \triangle T_j \end{aligned} \tag{11}$$

However, an arbitrary function $M(T_{ij})$ is used here. Referring to [32], we can directly use this arbitrary representation. In practice, we use the LM method to minimize the error function. Then the linear equation is expressed as:

$$(H_j + \lambda T)\triangle T_j = -b_j \tag{12}$$

Here, λ is the damping factor that controls the step size of minimizing the error.

To solve this linear problem, we should choose a proper linear solver algorithm. G2o allow different linear solvers, including Cholmod, CSparse, Eigen, PCG [33], and so on.

In this step, the transform matrix is the optimization vertices, that is, the variable to be optimized. The edges, which are constraints conditions, are defined as the differences of intersected LPP' relative coordinates in different trajectories. The index is a list of sequence numbers assigned to LPP of a unique location. The entries are references of each LPP in different trajectories.

Because of the random characteristic of crowdsourcing data, we do the optimization using incremental method. When a new trajectory is added to a current map, a standalone optimization graph is generated as one iteration.

D. POSE GRAPH OPTIMIZATION

After all the measures are aligned to global coordinates, the next step is to perform the PGO for all trajectories, to obtain a better estimate of all the poses. In this step, there are two constraints to optimize the coordinates of poses. Here we denote the two constraints as the inner-constraints and outer-constraints. Inner-constraints are the differences of coordinates of neighboring poses within one single trajectory. Because the difference is a vector, the distance and heading direction can be both optimized. Outer-constraints are sets of landmarks observed in different trajectories. By using graph optimization tools, we can add edges representing both kinds of constraints together into the model. After the data is processed, the output should be a set of fused pose graphs, which reflect the road network in the testing location. This map is not in a consistent coordinate system as the ground truth. Therefore, we need to further align the map after PGO, which is covered in the following section.

The inner-constraints here is normally referred to as the odometry constraints. Odometry is a widely-used method for determining the momentary position of robots [34]. It is an essential module in visual-SLAM systems. To better illustrate the problem, we use "odom" for notation in the following equations.

Using definition in Equ. 2 3, we may define the inner-constraints of trajectory I as:

$$E^i_{odom} = [(Pos^i_2 - Pos^i_1)^T - (Pos^i_{a+1} - Pos^i_a)^T]^T \quad (13)$$

$$e_{odom}(E^i) = E^i_{odom} - \overline{E^i_{odom}} \quad (14)$$

$$F_{odom}(E) = \sum_{i \in A} e_{odom}(E^i)^T \Omega_i e_{odom}(E^i) \quad (15)$$

Here, E^i_{odom} is the difference matrix, $\overline{E^i_{odom}}$ is the initial guess, $e_{odom}(E^i)$ is the error function of inner-constraints of trajectory I, $F_{odom}(E^i)$ is the total error estimation function, A is the total number of trajectories.

For the outer-constraints, we set them to be the differences of the LPP coordinates close to a unique location in different trajectories. This constraint can also be referred to as the Loop constraint, which we use here for notation. Note here the poses are already coarsely aligned after the previous step. Using Equ. 5 and Equ. 4, we define the outer-constraints of the LPP in trajectory I as:

$$e^{ij}_{loop} = X_j - X_i \quad (16)$$

e^{ij}_{loop} is the distance matrix of intersected LPP between trajectory I and J. The constraints are all empirically set as e^{ij}_{loop} to zero matrix. Note that these constraints are not uniform.

The idea of this step is to set each LPP coordinate to a same value, making them ''loop closure poses''. The aim of optimization is to optimize the coordinates to approximate so as to minimize the error function of the inner-constraints based on the outer constraints:

$$\begin{aligned} E^*_{odom} &= argmin F_{odom}(E) \\ s.t.\{e^{ij}_{loop} &= 0 | \forall i, j \in A\} \end{aligned} \quad (17)$$

Then, a similar LM algorithm is used here to solve the optimization problem. In this step, the vertices of optimization graph are all the poses. The edges are determined by two kinds of constraints, inner and outer.

E. TRANSFER TO GLOBAL INDOOR MAP

After PGO, a map with random azimuth is constructed. We still lack true azimuth of the starting position, which in this case is the initial position of the first trajectory. In our method, we need several pairs of reference points with correct azimuth measures to align the map. The reference points can be obtained using GPS data obtained near windows [35], or by some other indoor localization method.

Suppose we have two LPP, noted as A, B. Coordinates of A and B in the constructed map is noted as x_{ia},x_{ib}, and the measured ground truth of A and B are x_{ja},x_{jb}. To compute a direct transform using two points, we use the following equation:

$$\begin{bmatrix} x^2_{ia} + y^2_{ia} & x_{ia} & y_{ia} \\ x^2_{ib} + y^2_{ib} & x_{ib} & y_{ib} \\ x_{ia}x_{ib} + y_{ia}y_{ib} & x_{ia} & y_{ib} \end{bmatrix} \cdot \begin{bmatrix} \cos\theta \\ T_x \\ T_y \end{bmatrix} = \begin{bmatrix} x_{ia}x_{ja} + y_{ia}y_{ja} \\ x_{ib}x_{jb} + y_{ib}y_{jb} \\ x_{ia}x_{jb} + y_{ib}y_{ja} \end{bmatrix} \quad (18)$$

In the PGO step, the differences between landmark measures in different relative coordinates are empirically set to zero. In other words, in the constructed map, each LPP corresponds to exactly one measure. In contrast, for the non-LPP, there are several measures that are close to this location. In this experiment, we use the LPP and their reference point measures to do the ground truth alignment. For each pair of reference, we calculate an aligned result. Furthermore, we calculate the average of these aligned results as the constructed map. Subsequently, we can compute the error of map construction and see in what degree the map can reflect the real scene of this area.

F. REDUNDANCY REMOVAL FOR OPTIMIZING

This is an optional step, which is to optimize the result to reduce redundancy of the constructed map. The results after graph optimization often have a redundancy problem because some part of the area might be a frequently visited location, such as, entrance, central tunnel, and so on. Some trajectories may have overlapped segments. In the optimization steps, these duplicated segments cannot be discarded. Thus, we can merge coordinates of a same segment in different trajectories and reduce the number of points of the whole map, to obtain a cleaner result. Pedestrians' trajectories can be regarded as a time sequence, thus we may use Dynamic Time Warping (DTW) [14], which is a frequently used algorithm in audio processing, to identify which points are near and can be merged.

Dynamic Time Warping calculates the similarity of two sequences and can determine a mapping of one sequence to the other, indicating which points are closer to each other in a time domain. Thus, we can determine which points can be merged. Considering we use a 2D scene here, the distance should be derived based on Euclidean distance. After the measuring value is determined, we may use DTW to calculate the mapping between two overlapped segments, which do not belong to a same trajectory. As each trajectory has several LPPs and different trajectories only have part of segments overlapped, we can split a trajectory into segments by using LPP as cut points and only merge segments regardless of which trajectory it is in. For each two segments, we can directly use DTW to calculate the similarity and mapping of points.

To identify the segments that should be merged, we use the LPP's index at both ends of a segment. If two segments have a similar pair of ends, we may decide that they are a merging candidate pair. Considering that there might be some detour between two ends of a segment, we need to filter out those candidate pairs that have low similarity. To decrease complexity, we perform the merging and filtering iteratively, which means after two candidates are accepted, we merge the two segments and use the merged new result as a candidate for next merging operation. Note that when the constructed map do not have too many overlapped parts, this step can be omitted to avoid introducing errors.

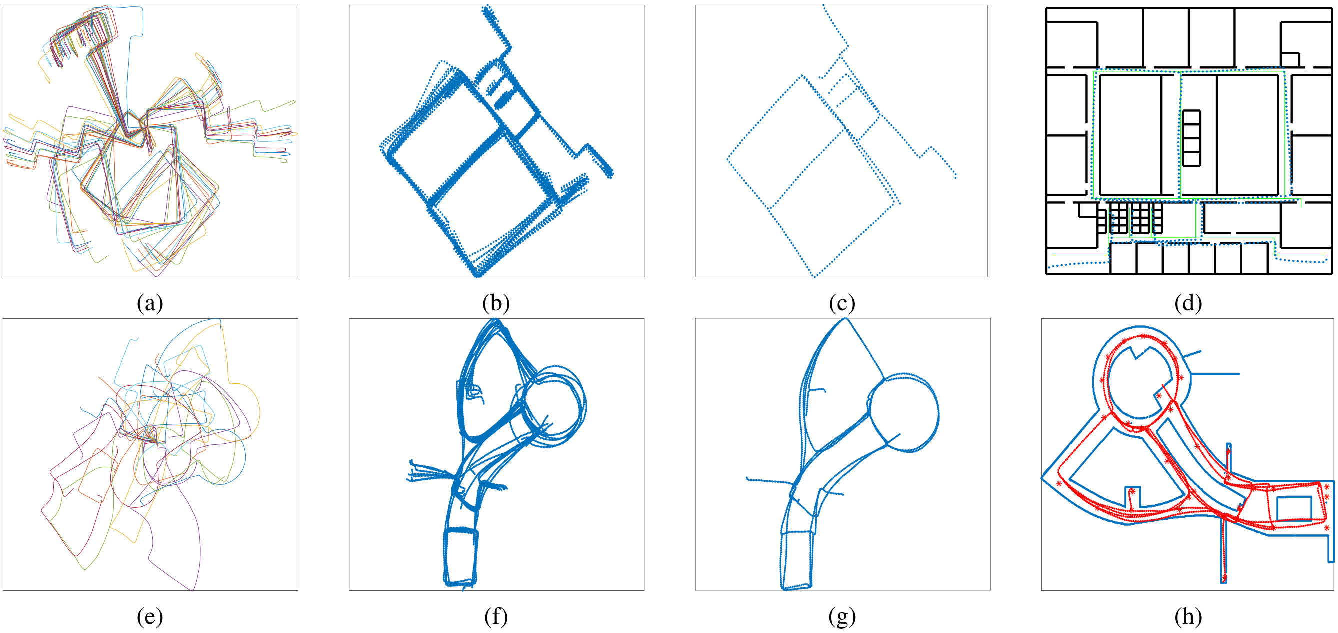


FIGURE 4. **Outcome of the mapping process. (a) Raw trajectories (b) Trajectory alignment; (c) Pose graph optimization; (d) Transfer to global map; (e) Raw trajectories; (g) Trajectory alignment; (g) Pose graph optimization; (h) Transfer to global map**

IV. EXPERIMENTS

A. EXPERIMENT SETUP

To evaluate the performance of the proposed method. We performed experiments in two environments: an academic building at Shenzhen University, with a $52.5\,m \times 52.5\,m$ floor plan as shown in Fig. 5(a), and a shopping mall on 5th floor of Coastal City Shopping Mall in Shenzhen, with a $100\,m \times 70\,m$ floor plan as shown in Fig. 5(b). During the experiments, four participants (one female and three males) carried a smartphone and walked normally in the accessible areas of the buildings. We used three types of Android smartphones, including the Nexus S, Nexus 5, and Galaxy 3. We developed the client application for data collection, and the data is sent to the cloud for processing.

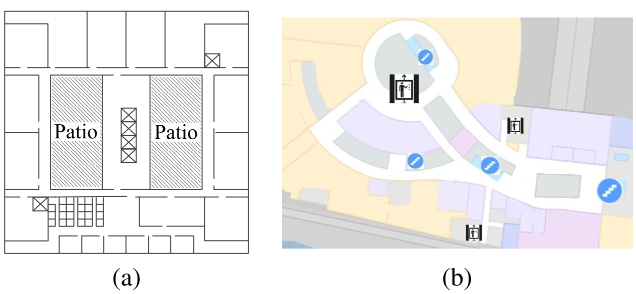


FIGURE 5. **Experiment environments. (a) office building; (b) coastal city.**

To quantify the performance of the proposed indoor map construction method, the following metrics were used [6], [8]:

- Graph Discrepancy Metric (GDM): GDM reflects the differences between the nodes of the inferred map and that of the ground truth. Euclidean distance is used as the difference metric.
- Shape Discrepancy Metric (SDM): SDM reflects the differences between the shapes of the constructed map and the real one. To calculate the SDM, the link segments were uniformly sampled. The distances between corresponding sampling points are used as the metric.

B. VISUAL RESULTS

Before presenting quantitative evaluation results, we provide visual results of this method. Fig. 4 shows the outcome of the mapping process. We introduce the mapping process using the academic building as the example. Firstly, the raw trajectories collected by the crowdsourcing uses are shown in Fig. 4(a). Secondly, the random trajectories are aligned based on LPP marking results. After trajectory alignment, a noisy result of the map is formed as shown in Fig. 4(b). Thirdly, pose graph optimization is implemented to the noisy result, and the optimization result is shown in Fig. 4(c). We can see that the drifting error is reduced and noisy measurements are optimized. After the thirdly step, we can get a relative map. The fourth step is to transfer the relative to global indoor map, as shown in Fig. 4(d). The mapping process of the coastal city is similarly, which can be seen from Fig. 4 (e)-(h).

C. COMPARISON WITH ALIMC

We compare the performance of the proposed indoor mapping method to that of ALIMC [8]. The proposed method utilizes SLAM technique to merge crowdsourcing trajectory data, while ALIMC uses Multi-Dimensional Scaling(MDS) technique for map construction. MDS uses the

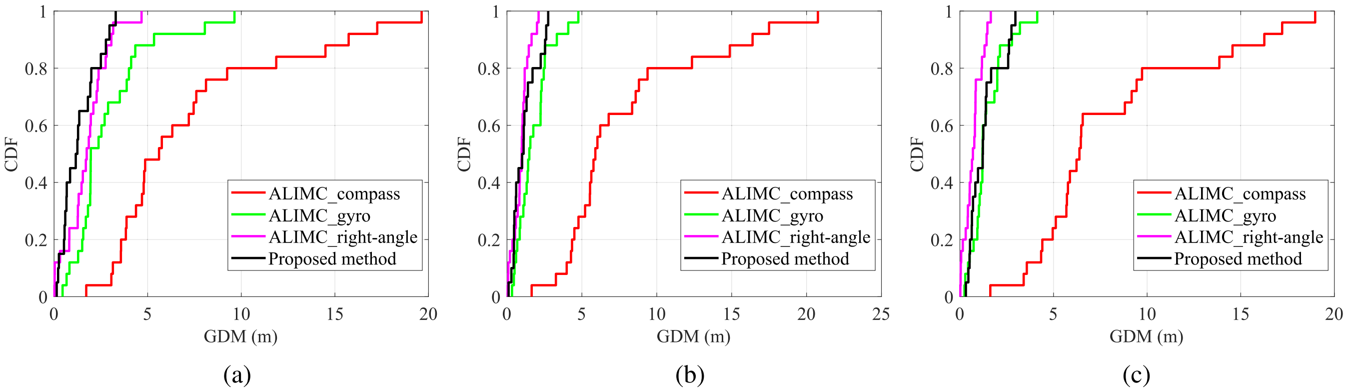


FIGURE 6. CDF of GDM with different methods. (a) 15 minutes; (b) 75 minutes; (c) 150 minutes.

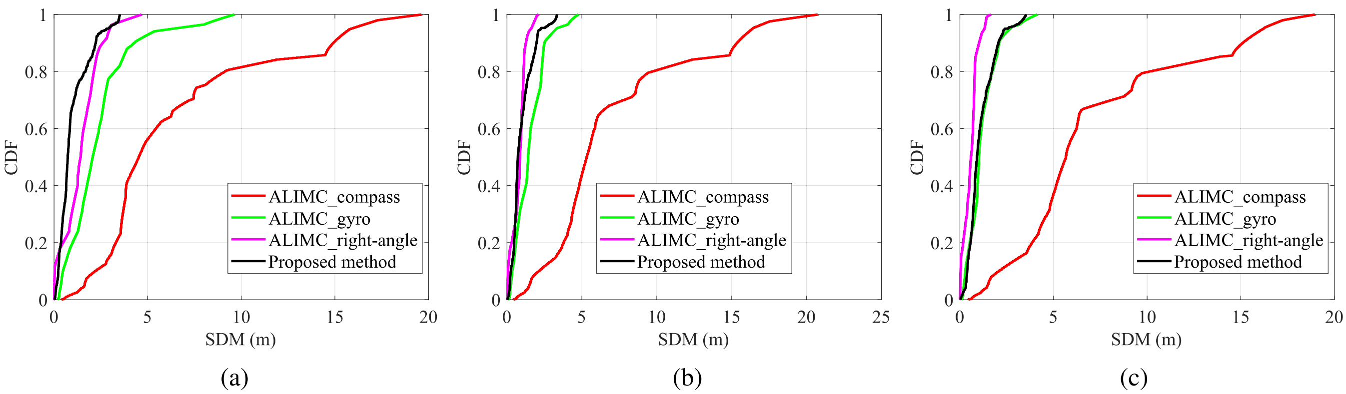


FIGURE 7. CDF of SDM with different methods. (a) 15 minutes; (b) 75 minutes; (c) 150 minutes.

straight-line distance between two nodes to construct the map, which can not operate on the maps with curves. We use the experiments in the academic building for the comparison. ALIMC used different angle estimation methods, namely compass, heuristic method based gyroscope, and right-angle ($ALIMC\_compass$, $ALIMC\_gyro$, and $ALIMC\_right - angle$).

The GDM and SDM of different methods with different crowdsourcing data are shown in Fig. 6 and Fig. 7. The crowdsourcing data is in terms of time. From Fig. 6, we can see that for 15 minutes data, the 80-percentile error of the proposed method is about 2 meters, which is better than that of the other methods. With the increasing of the crowdsourcing data, the error of all the methods decreases. The performance of the proposed method in different data amount are better than that of the $ALIMC\_compass$ and $ALIMC\_gyro$. When the data amount increases to 75 minutes, the performance of $ALIMC\_right - angle$ is better than that of the proposed method. The 80-percentile error of $ALIMC\_right - angle$ is about 1.2 meters, and that of the proposed method is about 1.7 meters. However, $ALIMC\_right - angle$ is based on the assumption that the corners are all right angles. Without this assumption, the performance of ALMC ($ALIMC\_compass$, $ALIMC\_gyro$) is worse than that of the proposed method.

From Fig. 6 and Fig. 7, we can see that with the increasing of the data amount, the GDM and SDM change not too much. We can also see that the data amount needed of the proposed method is much less than that of the ALIMC. For 15 minutes data, the 80-percentile error of the proposed method is about 2 meters, while $ALIMC\_gyro$ need 150 minutes data. This is because the proposed method is based on graph optimization, which uses a Levenberg-Marqartdt method to minimize error. This method is gradient decent optimization method and has enough constraints for optimization. Therefore, a small number of trajectories data can converge to an accurate result.

D. INDOOR MAP PERFORMANCE

To further evaluate the proposed method, we also implement it in a shopping mall. The reconstructed indoor map and their respective ground truths are shown in Fig. 4. We evaluate the quality of the indoor map using GDM and SDM introduced before. Fig. 8 show the Cumulative Distribution Function (CDF) of the GDM and SDM of the constructed map. Fig. 8(a) is the GDM and SDM results of the academic building. 80% of GDM error is within approximately 1.9 m. 80% of SDM error is within approximately 1.7 m. Fig.8(b) is the GDM and SDM results of the coastal city. 80% of GDM

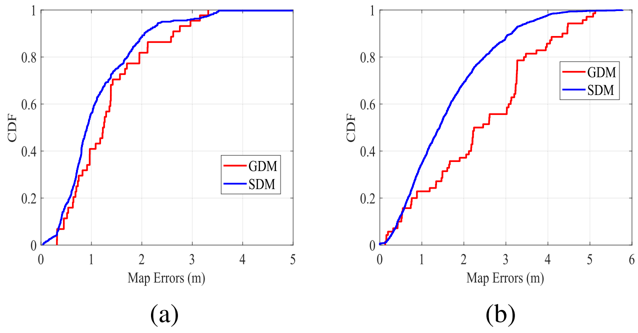


FIGURE 8. CDFs of mapping errors. (a) office building; (b) coastal city.

error is within approximately 3.5 *m*. 80% of SDM error is within approximately 2.5 *m*.

V. CONCLUSION

In this paper, we present a novel method to construct indoor map via crowdsourcing data. Firstly, we detect distinguishable activities such as taking elevators, escalators or turning based on activity detection algorithm, and infer trajectories by PDR. After that, we mark these activities' corresponding positions along the trajectories. Secondly, we use these detected positions as intersection points of different trajectories, and use them to do loop closure to construct an indoor map. Thirdly, we optimize the constructed result by removing overlapping. And finally, we align the map to absolute coordinate as the result. Two experiments done in an academic building and a shopping mall floor demonstrate the effectiveness of this method. The results suggest that our method can achieve an accuracy of 80-percentile mapping error around 1.7-3.5 meters. Additionally, this method can reflect some complex routes such as a curved corridor. And the required data for mapping is relatively small.

REFERENCES

[1] A. Millonig and K. Schechtner, "Developing landmark-based pedestrian-navigation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 43–49, Mar. 2007.

[2] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.

[3] B. Zhou, Q. Li, Q. Mao, W. Tu, and X. Zhang, "Activity sequence-based indoor pedestrian localization using smartphones," *IEEE Trans. Human-Machine Syst.*, vol. 45, no. 5, pp. 562–574, Oct. 2015.

[4] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. J. Cox and G. T. Wilfon, Eds. New York, NY, USA: Springer-Verlag, 1990, pp. 167–193.

[5] M. Alzantot and M. Youssef, "Crowdinside: Automatic construction of indoor floorplans," in *Proc. 20th Int. Conf. Adv. Geograp. Inf. Syst.*, 2012, pp. 99–108.

[6] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie–Markie: Indoor pathway mapping made easy," in *Proc. 10th USENIX Conf. Netw. Syst. Des. Implement.*, 2013, pp. 85–98.

[7] D. Philipp *et al.*, "Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2014, pp. 139–147.

[8] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, and L. Chen, "ALIMC: Activity landmark-based indoor mapping via crowdsourcing," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2774–2785, Oct. 2015.

[9] B. Zhou, Q. Li, Q. Mao, and W. Tu, "A robust crowdsourcing-based indoor localization system," *Sensors*, vol. 17, no. 4, p. 864, 2017.

[10] P.-Y. Gilliéron and B. Merminod, "Personal navigation system for indoor applications," in *Proc. 11th IAIN World Congr.*, 2003, pp. 21–24.

[11] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newslett.*, vol. 12, no. 2, pp. 74–82, Dec. 2010.

[12] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1027–1037, Oct. 2008.

[13] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 545–565, Jun. 2016.

[14] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, 2005.

[15] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services*, 2012, pp. 197–210.

[16] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 444–457, Feb. 2015.

[17] C. Wu, Z. Yang, and C. Xiao, "Automatic radio map adaptation for indoor localization using smartphones," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 517–528, Mar. 2017.

[18] Q. Chen and M. W. Mutka, "iFrame: Dynamic indoor map construction through automatic mobile sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Jul. 2016, pp. 1–9.

[19] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[20] M. Hardegger, S. Mazilu, D. Caraci, F. Hess, D. Roggen, and G. Troster, "ActionSLAM on a smartphone: At-home tracking with a fully wearable system," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, 2013, pp. 1–8.

[21] K. Konolige and M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1066–1077, Oct. 2008.

[22] D. Ellis, E. Sommerlade, and I. Reid, "Modelling pedestrian trajectory patterns with Gaussian processes," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Sep. 2009, pp. 1229–1234.

[23] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 3562–3568.

[24] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, Feb. 2010.

[25] Y. Gu, C. Zhou, A. Wieser, and Z. Zhou. (2017). "WiFi based trajectory alignment, calibration and easy site survey using smart phones and foot-mounted IMUs." [Online]. Available: https://arxiv.org/abs/1706.00636

[26] Y. Gu, Q. Song, M. Ma, Y. Li, and Z. Zhou, "Using ibeacons for trajectory initialization and calibration in foot-mounted inertial pedestrian positioning systems," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, 2016, pp. 1–7.

[27] M. Zhou, A. K.-S. Wong, Z. Tian, X. Luo, K. Xu, and R. Shi, "Personal mobility map construction for crowd-sourced Wi-Fi based indoor mapping," *IEEE Commun. Lett.*, vol. 18, no. 8, pp. 1427–1430, Aug. 2014.

[28] B. Kim *et al.*, "Multiple relative pose graphs for robust cooperative mapping," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2010, pp. 3185–3192.

[29] X. S. Zhou and S. I. Roumeliotis, "Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 1785–1792.

[30] S. Beauregard and H. Haas, "Pedestrian dead reckoning: A basis for personal positioning," in *Proc. 3rd Workshop Positioning, Navigat. Commun. (WPNC)*, 2006, pp. 27–35.

[31] J. More, "The Levenberg–Marquardt algorithm, implementation, and theory," in *Numerical Analysis*, G. A. Watson, Ed. Springer-Verlag, 1977.

[32] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2O: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2011, pp. 3607–3613.

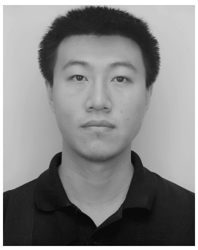
[33] A. V. Knyazev, "Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method," *SIAM J. Sci. Comput.*, vol. 23, no. 2, pp. 517–541, 2001.

[34] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 6, pp. 869–880, Dec. 1996.

[35] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 173–184.

BAODING ZHOU received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2015. He is currently an Assistant Professor with the Shenzhen Key Laboratory of Spatial Smart Sensing and Service, College of Civil Engineering, Shenzhen University, Shenzhen, China. His research interests include indoor localization and mapping, mobile computing, and intelligent transportation.

QINGQUAN LI received the Ph.D. degree in geographic information system (GIS) and photogrammetry from the Wuhan Technical University of Surveying and Mapping, Wuhan, China, in 1998. He is currently a Professor with Shenzhen University, Guangdong, China, and Wuhan University, Wuhan. His research areas include 3-D and dynamic data modeling in GIS, location-based service, surveying engineering, integration of GIS, global positioning system and remote sensing, intelligent transportation system, and road surface checking.



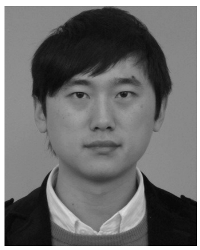
GUANXUN ZHAI received the master's degree in computer science from the City University of Hong Kong, Hong Kong, in 2017. He is currently a Research Assistant with the Shenzhen Key Laboratory of Spatial Smart Sensing and Service, Shenzhen University, Shenzhen, China. His research interests include indoor localization and mapping, and virtual reality.

QINGZHOU MAO received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2008. He is currently a Professor with Wuhan University. His main research fields include satellite navigation system, remote sensing and geographic information system integrates theory and method, GNSS/IMU navigation and position technology, high-precision laser measurement and point cloud data intelligent processing algorithm, pattern recognition and vision measurement technology and its application in mapping, road, railways, and tunnels, and other major projects testing and measurement field.

JUN YANG received the master's degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2016. She is currently a Research Assistant with the Shenzhen Key Laboratory of Spatial Smart Sensing and Service, Shenzhen University, Shenzhen, China. Her research fields include simultaneous localization and mapping and deep learning.

WEI TU received the B.E. and Ph.D. degrees in geographic information science from Wuhan University, Wuhan, China, in 2007 and 2013, respectively. He is currently an Assistant Professor with the Shenzhen Key Laboratory of Spatial Smart Sensing and Service, Shenzhen University, Shenzhen, China. His research interests include spatio-temporal data modeling, spatio-temproal data analysis, and spatio-temproal data mining.

WEIXING XUE received the bachelor's degree from the School of Geodesy and Geomatics, Wuhan University, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include seamless positioning and navigation, multisensor information fusion and data processing theory, and precision engineering measurement.



LONG CHEN received the B.Sc. degree in communication engineering and the Ph.D. degree in signal and information processing from Wuhan University, Wuhan, China, in 2007 and 2013, respectively. He was a Visiting Student with the National University of Singapore, Singapore, from 2011 to 2012.

From 2008 to 2013, he was in charge of environmental perception system for autonomous vehicle SmartV-II with the Intelligent Vehicle Group, Wuhan University. He is currently an Associate Professor with the School of Mobile Information Engineering, Sun Yat-sun University, Guangzhou, China. His areas of interest include computer vision, point cloud processing, and unmanned autonomous vehicles.