ALIMC: Activity Landmark-Based Indoor Mapping via Crowdsourcing

Baoding Zhou, *Student Member, IEEE*, Qingquan Li, Qingzhou Mao, *Member, IEEE*, Wei Tu, *Member, IEEE*, Xing Zhang, and Long Chen

***Abstract*—Indoor maps are integral to pedestrian navigation systems, an essential element of intelligent transportation systems (ITS). In this paper, we propose ALIMC, i.e., Activity Landmark-based Indoor Mapping system via Crowdsourcing. ALIMC can automatically construct indoor maps for anonymous buildings without any prior knowledge using crowdsourcing data collected by smartphones. ALIMC abstracts the indoor map using a link–node model in which the pathways are the links and the intersections of the pathways are the nodes, such as corners, elevators, and stairs. When passing through the nodes, pedestrians do the corresponding activities, which are detected by smartphones. After activity detection, ALIMC extracts the activity landmarks from the crowdsourcing data and clusters the activity landmarks into different clusters, each of which is treated as a node of the indoor map. ALIMC then estimates the relative distances between all the nodes and obtains a distance matrix. Based on the distance matrix, ALIMC generates a relative indoor map using the multidimensional scaling technique. Finally, ALIMC converts the relative indoor map into an absolute one based on several reference points. To evaluate ALIMC, we implement ALIMC in an office building. Experiment results show that the 80th percentile error of the mapping accuracy is about 0.8–1.5 m.**

***Index Terms*—Indoor mapping, activity landmark, crowdsourcing, smartphone.**

I. Introduction

PEOPLE spend the majority of their time indoors, such as in offices, schools, and shopping malls. Therefore, indoor navigation is an important part of intelligent transportation systems (ITS) [1]. In an indoor navigation system, an indoor map is a key factor, which is often needed by the indoor localization system to show a user where she/he is. An indoor map is also required in some indoor localization methods that use accurate indoor maps to constrain the drift of inertial sensors [2], [3]. Moreover, several recent approaches to automatic radio map construction require knowledge of real maps [4], [5].

Indoor maps 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 structures and its corresponding functionalities often change over time, making the original maps outdated [6]. Although it is possible to manually construct an indoor map or update an existing one, this process is very labor-intensive and time-consuming.

Crowdsourcing is a low-cost and efficient way to extract useful information from data acquired by crowd participants. The crowdsourcing method has been successfully applied to OpenStreetMap (OSM) for outdoor map construction, which utilizes Global Positioning System (GPS) for localization. However, since GPS cannot work indoors, indoor map construction must rely on other devices to obtain location information.

In this paper, we propose ALIMC: an Activity Landmark-based Indoor Mapping approach using Crowdsourcing. ALIMC can automatically construct indoor maps of unknown buildings using the crowdsourcing data collected by the smartphones. ALIMC is based on two key observations. The first observation is that the indoor map can be characterized using a link-node model [7] in which pathways are the links and the intersections of the pathways are the nodes, as shown in Fig. 1. For indoor navigation, the link-node mode is adequate because it provides a natural framework for locating users and points of interest (POIs). Moreover, people usually walk along pathways, and indoor POIs are connected by pathways. The second observation is that the nodes of the link-node model are normally corners, elevators, and stairs. When people pass the nodes, they usually complete special activities[1] other than walking. These activities can be detected by the built-in sensors of a smartphone [8]–[10]. Furthermore, the relative distance between these nodes can be calculated by Pedestrian Dead Reckoning (PDR) using the inertial sensors of a smartphone [11], [12].

The key concept of ALIMC is using the activity-related location as the landmark to merge the crowdsourcing trajectories. For simplicity, the activity-related location landmark is called "activity landmark" in this paper. An activity landmark has

Manuscript received September 30, 2014; revised March 9, 2015; accepted April 9, 2015. This work was supported in part by the Shenzhen Dedicated Funding of Strategic Emerging Industry Development Program under Grant JCYJ20121019111128765; by the Shenzhen Scientific Research and Development Funding Program under Grants ZDSY20121019111146499, JSGG20121026111056204, JCYJ20120817163755063, JCYJ20140418095735587, and JCYJ20140828163633980; by the National Natural Science Foundation of China under Grants 41371377, 41201483, 41301511, 41401444, and 41401525; and by the China Postdoctoral Science Foundation under Grants 2013M542199 and 2014M560671. The Associate Editor for this paper was A. Hegyi. *(Corresponding author: Qingquan Li.)*

B. Zhou, Q. Li, W. Tu, and X. Zhang are with the Shenzhen Key Laboratory of Spatial Smart Sensing and Services and the Key Laboratory for Geo-Environment Monitoring of Coastal Zone of the National Administration of Surveying, Mapping and GeoInformation, Shenzhen University, Shenzhen 518060, China (e-mail: bdzhou@whu.edu.cn; liqq@szu.edu.cn; tuwei@szu.edu.cn; xzhang@szu.edu.cn).

Q. Mao is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: qzhmao@whu.edu.cn).

L. Chen is with the Department of School of Mobile Information Engineering, Sun Yat-sen University, Zhuhai 519082, China (e-mail: chenl46@mail.sysu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2015.2423326

[1]In this paper, activities mean transient activities, such as turning, taking the elevator, and walking up/down the stairs.

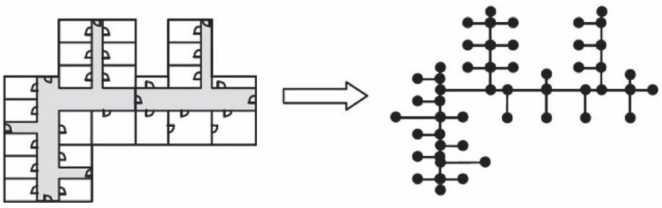

Fig. 1. Link-node model of an indoor map [7].

two features: activity type and WiFi fingerprints collected at the activity. For indoor map construction, ALIMC first extracts activity landmarks using activity detection algorithms. Then, ALIMC clusters all the activity landmarks into different clusters, each of which is treated as a node of the map. After clustering, ALIMC estimates the distances between all the nodes and obtains a distance matrix. Based on the distance matrix, the indoor map can be constructed using MultiDimensional Scaling (MDS) technique. For indoor map construction, ALIMC faces the following challenges:

- Activity landmark clustering: The crowdsourcing data includes many activity landmarks, some of which are collected at the same node. Generally, there are more than one of the activity landmarks with the same activity feature in an indoor environment. For example, there are normally a lot of corners in a building. For this reason, it is impossible to cluster activity landmarks only based on the activity type. Therefore, WiFi fingerprint is used for differentiating these landmarks. In certain situations, there may be a higher spatial density of activity landmarks than the spatial resolution provided by WiFi fingerprint. The first challenge is how to cluster the activity landmarks collected at the same node into one cluster without human intervention.
- Relative distance calculation: The relative distances between all the nodes are one of the basic elements of ALIMC. The distance between the adjacent nodes can be calculated by PDR. However, if two nodes are non-adjacent, the angles between the links are required for distance calculation. The errors of the angle measured by compass and gyroscope are very large, which makes the distance measurement is inaccurate. The second challenge is how to compensate for the angle estimation error for distance calculation.
- Crowdsourcing data: Crowdsourcing offers a potentially highly scalable solution for automatic map construction. The main challenge is how to fuse the crowdsourcing data collected by different users.

To tackle these challenges, ALIMC consists of the following components:

- Activity landmark clustering algorithm: This algorithm is used to cluster the activity landmarks collected at the same node into one cluster. WiFi fingerprint is first used to classify these activity landmarks with the similar WiFi features into one cluster. However, some nodes are so close that their WiFi features are indistinguishable. Therefore, a spatial-information-based clustering algorithm is adopted to eliminate these particular activity landmarks.
- Relative-distance estimation method: For relative distance estimation, the intersection angle between two links is needed. In order to estimate the accuracy angle information, a heuristic method is used for angle calculation based on the gyroscope. Moreover, the interior angles of a closed polygon is used as the priori knowledge to infer the bias drift of the gyroscope.
- Indoor map construction approach: For indoor map construction, the relative distances between all the activity landmarks are first calculated based on the distance and angle estimation results. Then, MDS technique is used to create a relative map. Lastly, the relative map is transformed into an absolute map based on several reference points.

The main contribution of this paper is using activity-based context information as the landmark for indoor map construction. Moreover, the indoor map constructed by ALIMC includes not only the topology information, but also node attributes, such as the location of elevators and stairs. These attributes are important for pedestrian navigation, since elevators and stairs are frequently used.

The rest of this paper is organized as follows. Section II reviews related work. Section III presents the ALIMC system overview. Sections IV, V, and VI detail the methodology of ALIMC, followed by our results and analysis in Section VII. Section VIII discusses the proposed method. Section IX concludes the paper.

II. Related Work

Most approaches for indoor mapping are based on the principle of Simultaneous Localization and Mapping (SLAM). SLAM [13] is a well-known technique in the robotic domain that is concerned with solving the problem of locating a robot in an unknown environment while simultaneously building a map of the surrounding area. Typically, SLAM methods rely on visual landmarks or obstacles detected by cameras, sonar, or laser-range sensors. The types of sensors used by traditional SLAM prevents it from being used with smartphones. SmartSLAM [14] presents a modified SLAM algorithm for smartphones that employs inertial sensors as a motion model along with WiFi signals as an observation model to construct the map. FootSLAM [15] proposes a SLAM-based approach for pedestrians that generates a probabilistic map that can be used for localization by means of a particle filter to learn the step direction probabilities at each location the pedestrian visits. Puyol *et al.* extend FootSLAM to multistory building using an autoregressive integrated moving average model for the odometry error along the vertical component [16]. These SLAM-based approaches do not use crowdsourcing-based methods for indoor map construction, which is different from ours.

For indoor map construction, Alzantot *et al.* propose CrowdInside, which constructs indoor map using different user trajectories [17]. In CrowdInside, each trace requires GPS coordinates as the initial point. However, this requirement may not be met in practice because users may start their traces at

different places in the building. Walkie-Markie [18] proposed an indoor pathway mapping system that can automatically reconstruct internal pathway maps of buildings without any priori knowledge. The key concept of Walkie-Markie is using WiFi-defined landmarks (WiFi-Mark) to anchor the various partial trajectories collected by different users. A WiFi-Mark is defined as a pathway location at which the trend of received signal strength changes from increasing to decreasing when moving along the pathway. In the work of Walkie-Markie, the attributes of the nodes cannot be detected automatically.

Jiang *et al.* propose an automatic indoor map construction system using WiFi fingerprints and user motion information [6]. The system proposed in [6] first uses WiFi fingerprints to determine which rooms are adjacent, then utilizes hallway layout learning to estimate room sizes and to order rooms along each hallway, and lastly uses force directed dilation to adjust room sizes and optimize overall map accuracy. However, the system proposed in [6] is based on the assumption that each room has been associated with a unique room ID and the corresponding WiFi fingerprint. This assumption is usually not supported since there is not such prior information before a map has been constructed.

PiLoc [19] proposed an indoor localization system that can automatically construct a map and a radio map through user generated data. The key technique of PiLoc is the trajectory matching algorithm, which utilizes movement displacement as well as the associated signal to match disjoint trajectory segments. PiLoc is based on the assumption that the motion traces and changing trends of the received signal strength along the same path are highly correlated. MapGENIE [20] proposes an indoor mapping approach that automatically derives indoor maps from traces collected by pedestrians moving around in a building. MapGENIE realizes indoor mapping by leveraging exterior information about the building and uses grammar to encode structural information about the building. MapGENIE is based on the assumption that the grammar used for modeling is provided by parsing a known indoor map of another building that has a similar architectural style using a grammar generator proposed in [21]. Moreover, MapGENIE requires a foot-mounted inertial measurement unit (IMU) to collect traces, which is not internally installed on smartphones.

III. System Overview

The overview of the ALIMC system is shown in Fig. 2. ALIMC utilizes built-in sensors of a smartphone to collect motion data and WiFi fingerprints. The motion data includes heading, angular velocity, pressure, and acceleration. The WiFi fingerprint includes the MAC of the Access Point (AP) and the corresponding Received Signal Strength (RSS) value.

The key challenge in ALIMC is how to merge these crowdsourcing data to construct an indoor map. Since the indoor map can be characterized by the link-node model, the aim of ALIMC is to extract the links and nodes from the collected data. ALIMC consists of three modules. The first module is activity landmark clustering, which is used to cluster the activity landmarks collected at the same node into one cluster. The input of this module is the activity detection result, the associated WiFi fingerprint, and the motion data. The output is the clustering result and each cluster is treated as one node of the map. The second module is relative distance estimation, which calculates the distance between all the nodes and generates a distance matrix. The input of this module is the activity landmark clustering result and the motion data. The output is the distance matrix. The third module is indoor mapping, which is used to construct the indoor map based on the distance matrix. The input of this module is the distance matrix and several reference points. The output is the indoor map.

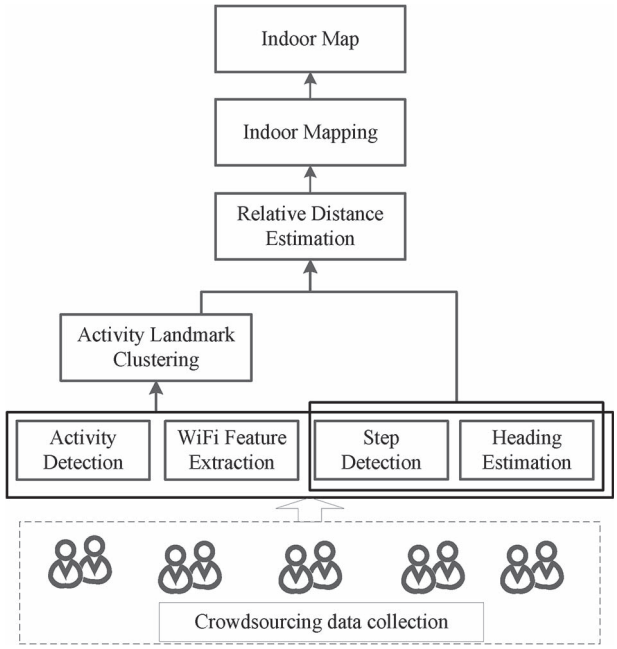

Fig. 2. System overview.

IV. Activity Landmark Clustering

The crowdsourcing data include many activity landmarks, some of which are collected at the same node. Therefore, the activity landmarks collected at the same node should be clustered into the same cluster, and each cluster is treated as a node of the map. This section details how to cluster the activity landmarks.

A. Activity Landmark Definition

The indoor map can be characterized as a link-node model [7] in which pathways are the links and the intersections of the pathways are the nodes. In this paper, the term activity landmark is used to express the nodes in context form. We identify an activity landmark by the following six-tuple.

$$\mathrm{AL} \triangleq \{\mathrm{ID}, \mathrm{type}, F, \Delta H, (D_p, D_f), (\mathrm{ID}_p, \mathrm{ID}_f)\} \qquad (1)$$

where ID is the identification number of the activity landmark; type is the activity type, including turn, taking the elevator, and walking up/down the stairs, F is the WiFi fingerprint collected when the activity is detected; ΔH is the heading change value when the activity takes place, namely the angle between two

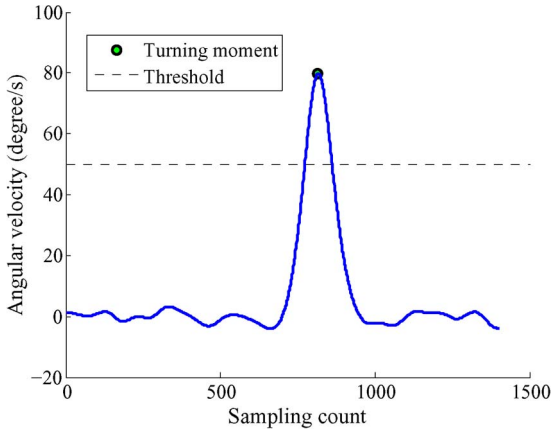

Fig. 3. Turning detection by peak detection algorithm.

connected links; D_p is the distance between the current activity landmark and the previous one, D_f is the distance between the current activity landmark and the following one; and ID_p and ID_f are the ID of the previous and following activity landmark, respectively.

B. Activity Detection

In order to get the activity landmarks, the first step is to detect activity. In an indoor environment, there are usually three types of activities: turning, taking the elevator, and walking up/down the stairs. This subsection will introduce how to detect these activities.

1) Turning: During the walking process, turns are common. When pedestrian turns, the angle velocity would generate a peak waveform, as shown in Fig. 3. Turning is detected using peak detection algorithm [22], and the threshold is set to 50 in this paper. The peak detection algorithm is used to find the local maximum or minimum during a period of time [22], the turning detection result by peak detection is shown in Fig. 3. Using the turn detection algorithm, the turning time is obtained. Since it is associated with the WiFi fingerprint collected at the turning time, the activity landmark is extracted from the collected data. Moreover, the turning time is also be used for calculating ΔH, which will be detailed in the following subsection.

2) Other Types of Activity Landmarks: Besides corners, there are other points where pedestrians do different activities, such as taking elevators and waling up/down stairs. These activities can also be used as activity landmarks for indoor map construction. Moreover, by detecting these activities, the attributes of the nodes can be labeled automatically.

Generally, an elevator use trace includes the following period: walking to the elevator, waiting for the elevator, walking into the elevator, standing inside for a short time, an overweight/weightlessness moment, a stationary period, another weightlessness/overweight moment, and walking out period [23]. The elevator use trace will generate particular inertial feature. In addition to the inertial feature mentioned above, the other feature about air pressure can be also used for elevator detection, which can be measured by the barometer embedded in the smartphone. Theoretically, the air pressure changes with the change of the altitude. The relationship between altitude and pressure is that when the altitude increases, the pressure drops, and vice versa. The acceleration and pressure of an elevator rising process are shown in Fig. 4.

Another activity with pressure change in office building is walking up/down the stairs. Different from elevator, during walking up/down stairs, there is neither an overweight moment nor a weightlessness moment. The acceleration of walking up/down stairs is similar to that of walking on flat ground, which can be seen from Fig. 5. The different feature is the pressure, which is varied during walking up/down stairs, and constant during walking on flat ground. Thus, the pressure change is used to distinguish between walking in stairs and walking in flat ground.

TABLE I
CORRELATION COEFFICIENT OF DIFFERENT SMARTPHONES

	Nexus S	Nexus 5	Galaxy 3
Nexus S	1	0.84	0.93
Nexus 5	0.84	1	0.92
Galaxy 3	0.93	0.92	1

C. Activity Landmark Clustering

After activity detection, several activity landmarks are generated, some of which are collected at the same node. Activity landmark clustering is used to cluster the activity landmarks taken at the same node into the same cluster and to give them a unique identification number, i.e., each cluster is treated as a node of the map.

1) WiFi Fingerprint Based Activity Landmark Clustering: To cluster the activity landmarks, WiFi features are used, since WiFi networks are widely available. WiFi fingerprints have been used as the feature for location identification in many previous works [4], [5]. These works use the RSS difference to distinguish different locations, which works well if the types of smartphones users use are the same. However, if the data is collected via crowdsourcing, participants' smartphones are usually very different. The different WiFi chipsets and antenna of different devices cause different RSSs for the same location. Some papers proposed several approaches to solve this heterogeneous device problem [24], [25]. In this paper, RSS order is used as the feature to distinguish different activity landmark clusters. The order of RSSs from a set of APs measured by different smartphones at the same location are theoretically the same. To verify the effectiveness of the RSS order feature, we collected the RSSs of surrounding APs at the same node by three different smartphones; namely, the Nexus S, Nexus 5, and Galaxy 3. Fig. 6 shows the results.

From Fig. 6, we can see that the RSS values measured by different smartphones are different. However, the RSS order of them is nearly the same. Generally, the correlation coefficient can be used to evaluate the similarity of the RSS order. The Pearson correlation coefficient of these three different smartphones is shown in Table I. From Table I, it can be seen that the correlation coefficient is very high. In this paper, we use the correlation coefficient of RSS as the WiFi feature to distinguish different activity landmarks.

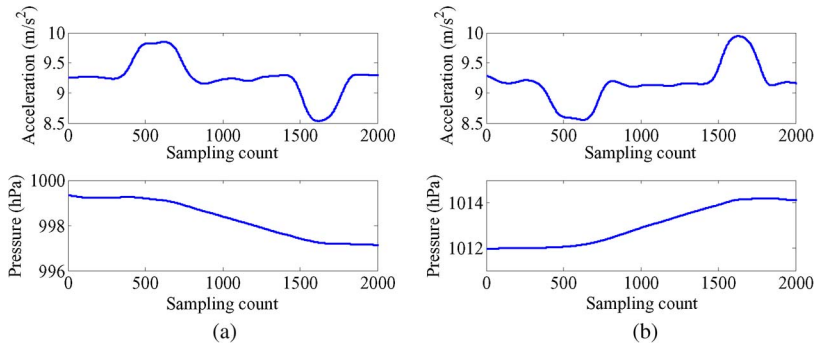

Fig. 4. Taking the elevator. (a) Up,(b) Down.

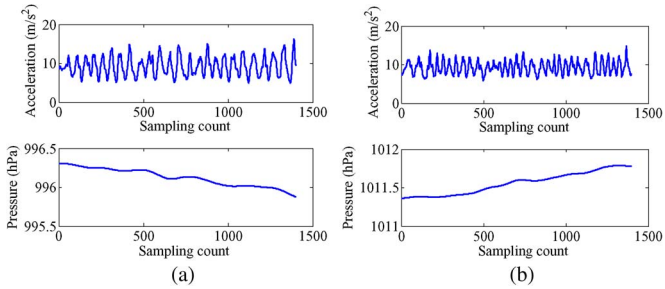

Fig. 5. Walking up/down the stairs. (a) Up, (b) Down.

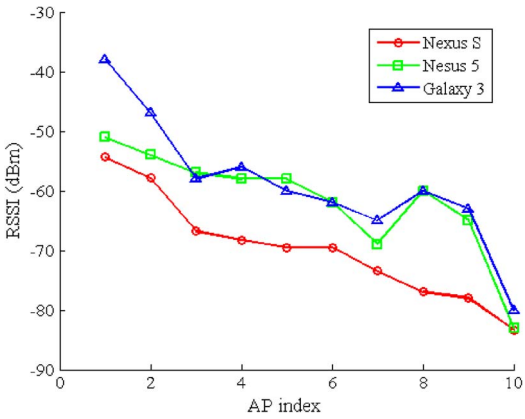

Fig. 6. RSSs measured at the same location by different smartphones.

We can infer that if the correlation coefficient is high enough, the two fingerprints are collected at the same location. However, if the number of the APs is too small, the correlation coefficient may be high even if the two fingerprints are collected at different locations. To solve this problem, we use the Jaccard similarity coefficient as another metric for activity landmark clustering. The Jaccard coefficient is a statistic used for comparing the similarity and diversity of sample sets [26]. In the following, we introduce how to calculate the correlation coefficient and Jaccard coefficient.

Without loss of generality, we assume f_i and f_j are the WiFi fingerprints of AL (we use AL to express Activity Landmark) i and AL j, respectively.

$$f_i = \{(\text{mac}_{i1}, \text{rss}_{i1}), (\text{mac}_{i2}, \text{rss}_{i2}), \ldots, (\text{mac}_{im}, \text{rss}_{im})\}$$
$$f_j = \{(\text{mac}_{j1}, \text{rss}_{j1}), (\text{mac}_{j2}, \text{rss}_{j2}), \ldots, (\text{mac}_{jn}, \text{rss}_{jn})\}$$

where mac is the MAC of AP, which is unique and rss is the RSS value of the AP.

First, we obtain the intersection and union of the MACs of f_i and f_j.

$$\text{MAC}_{int} = \text{MAC}_i \cap \text{MAC}_j$$
$$\text{MAC}_{uni} = \text{MAC}_i \cup \text{MAC}_j$$

where $\text{MAC}_i = \{\text{mac}_{i1}, \text{mac}_{i2}, \ldots, \text{mac}_{im}\}$, $\text{MAC}_j = \{\text{mac}_{j1}, \text{mac}_{j2}, \ldots, \text{mac}_{jn}\}$.

The Jaccard coefficient of f_i and f_j is calculated using the following equation:

$$\text{Jac}_{ij} = \text{num}(\text{MAC}_{int})/\text{num}(\text{MAC}_{uni}) \tag{2}$$

where num() is the number of the MAC in the set.

To calculate the correlation coefficient of f_i and f_j, the RSSs of the same APs must be extracted first, namely the RSS of each AP in the MAC_{int} set of f_i and f_j ($\text{MAC}_{int} = \{\text{mac}_1, \text{mac}_2, \ldots, \text{mac}_k\}$).

$$f_i' = \{(\text{mac}_1, \text{rss}_{i1}), (\text{mac}_2, \text{rss}_{i2}), \ldots, (\text{mac}_k, \text{rss}_{ik})\}$$
$$f_j' = \{(\text{mac}_1, \text{rss}_{j1}), (\text{mac}_2, \text{rss}_{j2}), \ldots, (\text{mac}_k, \text{rss}_{jk})\}$$

According to the RSSs of f_i', the order indexes ix can be obtained by sorting the RSSs in descending order.

$$[\text{RSS}_i, ix_{\text{sort}}] = \text{sort}(\{\text{rss}_{i1}, \text{rss}_{i2}, \ldots, \text{rss}_{ik}\})$$

Then, based on ix_{sort}, the RSSs of f_j' is re-ordered, and RSS_j is obtained. The correlation coefficient of f_i and f_j is calculated using the following equation:

$$\text{Corr}_{ij} = \frac{\text{cov}(\text{RSS}_i, \text{RSS}_j)}{\sigma_{\text{RSS}_i} \cdot \sigma_{\text{RSS}_j}} \tag{3}$$

where $cov(\text{RSS}_i, \text{RSS}_j)$ is the covariance of RSS_i and RSS_j, σ_{RSS_i} and σ_{RSS_j} are the standard deviation of RSS_i and RSS_j.

Based on Jac_{ij} and Corr_{ij}, we can determine whether AL i and AL j are collected at the same node. The criterion is that if

$\text{Jac}_{ij} \geq \text{jac}_{th}$ and $\text{Corr}_{ij} \geq \text{corr}_{th}$, AL i and AL j are the same node; otherwise, they are not. In our experiments, jac_{th} is set to 0.6, and corr_{th} is set to 0.6.

2) Spatial Information Based Activity Landmark Clustering: Using the clustering algorithm based on the WiFi fingerprint, some activity landmarks can be clustered correctly. However, some activity landmarks are so close together that the WiFi feature is not able to distinguish them. After clustering the activity landmarks using WiFi features, there may be some mismatching pairs. One new activity landmark may be matched to several nodes of the node database. In order to delete the mismatched landmark pairs, the spatial relationship is used.

When a new trajectory is updated, activity landmarks are detected using an activity detection approach where the WiFi features of the activity landmarks are extracted. Then, we get an activity landmark sequence indexed in chronological order. For example, $\{\text{NAL}_1, \text{NAL}_2, \ldots, \text{NAL}_m\}$ is a activity sequence of a new updated trajectory consisting of m activity landmarks (we use NAL to express the New Activity Landmark extracted from the updated trajectory). Using the clustering algorithm based on WiFi features, there are n $(n < m)$ NALs with similar WiFi features as the nodes in the node database; therefore, one activity landmark may be matched to several NODEs (we use NODE to express nodes in the database). Then, we get n similar NODEs sets for the n NALs, $\{\text{NODES}_1, \text{NODES}_2, \ldots, \text{NODES}_n\}$, where $\text{NODES}_i = (\text{NODE}_1, \text{NODE}_2, \ldots \text{NODE}_k)$, which means that for NAL_i, there are k NODEs in the node database that have similar WiFi features (note that for different NAL the number of the similar NODEs may be different). Since all the nodes in the node database are unique, there is only one or potentially no matching NODE for each NAL. The algorithm for deleting mismatch pairs is detailed as Algorithm 1.

Algorithm 1 Spatial information based activity landmark clustering algorithm

input: $\text{NAL}_{1:n}$
input: $\text{NODES}_{1:n}$
input: NALDB(new AL database)
input: NODEDB(NODE database)
input: n_win
output: M_pairs
definition: $\text{NODES\_can}_{M\times n}$ = NODES candidates which have the similar WiFi features with $\text{NAL}_{1:n}$
$\text{NODES\_can}_{M\times n}$=fun_DF($\text{NAL}_{1:n}$,$\text{NODES}_{1:n}$); //fun_DF means depth-first algorithm
M_pairs = [];
for $i = 1 : M$ **do**
  NODE_can_temp=$\text{NODE\_can\_list}_{M\times n}(:, i)$
  for $j = 1 : n - n$_win $+ 1$ **do**
    flag = 0; //used to flag if the spatial information of the adjacent ALs of NAL and NODE is similar
    for $k = 2 : n$_win **do**
      dist_{NAL}=$\|\text{NAL}_{j-1+k-1}, \text{NAL}_{j-1+k}\|$; //calculate the distance between $\text{NAL}_{j-1+k-1}$ and NAL_{j-1+k}
      $\text{heading}_{\text{NAL}} = \angle(\text{NAL}_{j-1+k-1}, \text{NAL}_{j-1+k})$; //calculate the heading information from $\text{NAL}_{j-1+k-1}$ to NAL_{j-1+k}
      $\text{dist}_{\text{NODE}}$=||NODE_can_temp$_{j-1+k-1}$, NODE_can_temp$_{j-1+k}$||;//calculate the distance between NODE_can_temp$_{j-1+k-1}$ and NODE_can_temp$_{j-1+k}$
      $\text{heading}_{\text{NAL}}$=$\angle$(NODE_can_temp$_{j-1+k-1}$, NODE_can_temp$_{j-1+k}$); //calculate the heading information from NODE_can_temp$_{j-1+k-1}$ to NODE_can_temp$_{j-1+k}$
      $\Delta\text{dist} = \text{dist}_{\text{NAL}} - \text{dist}_{\text{NODE}}$;
      $\Delta\text{heading} = \text{heading}_{\text{NAL}} - \text{heading}_{\text{NODE}}$;
      if $\Delta\text{dist} < \text{dist}_{th}$ **AND** $\Delta\text{heading} < \text{heading}_{th}$ **then**
        flag = flag + 1;
      end if
    end for
    if flag $== n$_win $- 1$ **then**
      M_pairs = [M_pairs; [$\text{NAL}_{j:j+n\_win-1}$, $\text{NODE\_can\_temp}'_{j:j+n\_win-1}$]];
    end if
  end for
end for
M_pairs=unique(M_pairs,'rows'); //remove the duplicates

The inputs of the algorithm includes: $\text{NAL}_{1:n}$, $\text{NODES}_{1:n}$, NALDB, NODEDB, and n_win.

- $\text{NAL}_{1:n}$: n NALs of the m activity landmarks extracted from the new updated trajectory that have similar WiFi features as the activity landmarks in the activity landmark database.
- $\text{NODES}_{1:n}$: n NODE sets consist of several nodes of the database that have the similar WiFi features as $\text{NAL}_{1:n}$.
- NALDB: NAL database, which stores the spatial information between each NAL, including distance and heading information.
- NODEDB: NODE database, which stores the spatial information between each NODE, including distance and heading information.
- n_win: the length of the sliding-window.

The output of the algorithm is M_pairs, which stores the matching pairs of the NALs and NODEs.

The first step of Algorithm 1 is to get $\text{NODES\_can}_{M\times n}$ using the depth-first algorithm from $\text{NODES}_{1:n}$, where $M = \prod_{i=1}^{n} k_i$, k_i is the number of the ith NODE set NODES_i. Then we get M NODESs candidates for $\text{NAL}_{1:n}$. For each NODES, we used a sliding-window to calculate the spatial difference of two sequential NALs and two corresponding NODEs, including distance and heading difference. The length of the sliding-window (n_win) is set to 3 in our experiment. If the difference is less than the threshold, the two sequential NALs and NODEs are determined to be matching pairs, and we add 1 to the flag value (n_{flag}). After comparing n_win activity landmarks, namely n_win $- 1$ sequential activity landmarks, if n_{flag} is equal to n_win $- 1$, the spatial information of n_win NALs is similar to that of n_win NODEs, and we add these NALs and NODEs to the matching pairs. Then, the NAL is given the identification number of the matching NODE.

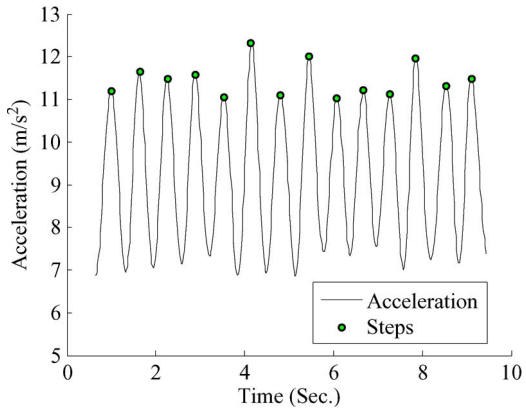

Fig. 7. Step detection result.

VI. Relative Distance Estimation

After activity landmark clustering, all the activity landmarks are clustered into different activity landmark clusters, and each cluster is given an unique identification number. The activity landmark clusters constitute the nodes of the indoor map. In order to construct indoor map, another factor is the distance between every two nodes. This section introduces how to calculate the distance between all the nodes.

A. Distance Between Adjacent Nodes

The distance between two adjacent nodes is calculated using the PDR method. In PDR, the displacement is obtained from the step count and heading information. If the previous location is (x, y), the next location is calculated as

$$(x + sl \cdot sc \cdot \cos(h), y + sl \cdot sc \cdot \sin(h)) \tag{4}$$

where sl denotes the step length, sc the step count, and h the heading. Step count is obtained using the peak detection algorithm introduced in [22]. Before peak detection, the raw acceleration data are preprocessed to filter out irrelevant data using a Butterworth low-pass of order 4, with a cutoff frequency of 10 Hz [22]. The step detection result is shown in Fig. 7. The step length is estimated using the frequency-based model proposed by Cho *et al.* [27]: stride_len $= a \cdot f + b$, where f is the step frequency, and a, b are parameters that can be trained offline. Since we suppose that the indoor map can be characterized by the link-node model, when a pedestrian is walking between two nodes, the heading does not change. Therefore, h is set to zero during the calculation of distance between adjacent nodes.

B. Intersection Angle Estimation

In an indoor map, most of the nodes are non-adjacent; therefore, to calculate the distance between these nodes, the intersection angle between every two links must be known. The intersection angle is the heading change when a pedestrian passes the intersection. The heading value can be obtained using the built-in compass of the smartphone. However, the compass is vulnerable to magnetic interference in indoor environments [2]. In addition to the compass, the gyroscope can also be used to determine the heading change value since the gyroscope can report the relative angular velocity of the smartphone. When integrated over time, the gyroscope yields the relative angular displacement, which can be used to estimate the heading change. However, the integration is affected by noise, causing gyroscope to drift. Borenstein *et al.* proposed a heuristic drift reduction method to address the gyroscope drift by performing the integration only when the gyroscope is changing substantially due to a turn [28], [29]. We utilize the heuristic drift reduction method in our system to estimate the heading changes. Moreover, due to the drift error ϵ, in each interval the output of a gyroscope can be modeled as equation (5) [28].

$$w_i = \tilde{w}_i + \epsilon \tag{5}$$

where w_i is the output of the gyroscope, $\tilde{w}_i$ is the true value, and ϵ is the bias drift. Then, the heading change is calculated by the following equation,

$$\Psi = \sum_{i=t_{\text{begin}}}^{t_{\text{end}}} w_i \cdot T_i \tag{6}$$

where Ψ is the heading change value, T_i is the time interval, t_{begin} and t_{end} are the beginning and ending time of the turn activity, which are obtained based on the turning time obtained by the turn detection algorithm. $t_{\text{begin}} = t - t\_\text{win}$ and $t_{\text{end}} = t + t\_\text{win}$, where t is the turning time and t_win is the time window, which is set to 0.4 second in our experiments.

The bias drift ϵ of the gyroscope can be inferred based on the sum of the inner angles of a closed polygon, which is equal to $180 \cdot (N - 2)$ (N is the number of edges). For example, four nodes constitute a quadrilateral, the sum of the inner angles is 360 degree (ground truth). By integrating of the gyroscope data, we can get an estimated value (containing the bias) of the sum of the inner angles. Then, we can use the ground truth to infer the bias. The inferred bias can be used thereafter.

Another method to calculate the intersection angle of two lines is approximating it as 90-degrees angle [6]. The approximation is based on the observation that the paths in the most office buildings are orthogonal.

C. Relative Distance Estimation

Based on the distances between adjacent nodes and intersection angles, the relative distance between all the nodes can be calculated. If there exists more than one path between two nodes, the mean of the distances of these paths is used as the relative distance. If sufficient trajectories are uploaded, the distance between all nodes can be obtained.

VI. Indoor Mapping

Using the relative distance estimation approach introduced in the previous section, we can find the relative distance matrix of all the nodes. Based on the relative distance matrix, the indoor map can be constructed. This section introduces how to construct the indoor map. The indoor map construction approach consists of two steps: relative indoor map construction and absolute indoor map construction.

A. Relative Indoor Map Construction

Based on the relative distance matrix, ALIMC first generates the relative indoor map using the MDS technique. MDS is commonly used to find a spatial relationship based on dissimilarity information between objects. In this paper, the dissimilarity information is the relative distance between nodes. Using the relative distance matrix as the input, MDS can generate the relative indoor map, for which we can get the relative spatial relationship of all the nodes. The relative indoor map can be used for indoor navigation, since it can provide the relative spatial relationship of the nodes based on the local coordinate system.

B. Converting to the Absolute Indoor Map

In order to convert the relative indoor map to the absolute indoor map, Procrustes analysis is used. Procrustes analysis is commonly used to find the scaling, translation, and rotation to fit a configuration to another as closely as possible [30]. Let Y_m be a vector of estimated positions of m reference points (located by some other indoor localization methods), and assume that real positions of n nonlinearity nodes are known, where n is larger than three, and X_n is a vector of the positions. Based on X_n, a subset vector Y_n can be constructed from Y_m composed of n nodes included in X_n. With X_n and Y_n, a normal Procrustes analysis can be proceed. As a result, scaling (S_n), rotation (R_n), and translation (T_n) parameters for mapping Y_n to X_n can be obtained. Then Y_m can be transformed to the absolute indoor map (Z_m) with the parameters by the following equation:

$$Z_m = S_n \cdot Y_m \cdot R_n + T_n \tag{7}$$

Based on equation (7), the absolute indoor map can be constructed. The reference points can be obtained by other methods, e.g., the last reported GPS location [31].

VII. Experiments

A. Experiment Setup

To evaluate the performance of the proposed system, we implemented ALIMC on three different types of Android phones, including the Nexus S, Nexus 5, and Galaxy 3. The experiments were conducted at two floors of an office building, with a 52.5 m $\times$ 52.5 m floor plan, as shown in Fig. 8. During the experiment, participants held a smartphone in front of themselves and walked normally in the accessible areas of the building. To simulate the crowdsourcing users, participants started at different positions. To evaluate the performance of ALIMC with incremental data, each traces are repeated ten times. In total, 300 user trajectories were collected by three participants using three types of smartphones. In terms of time, these trajectories correspond to 220 minutes of data collection.[2] The collected data includes acceleration data, compass data, gyroscope data, barometer data, and WiFi fingerprints.

[2]200 trajectories are collected on the 14th floor (about 150 minutes).

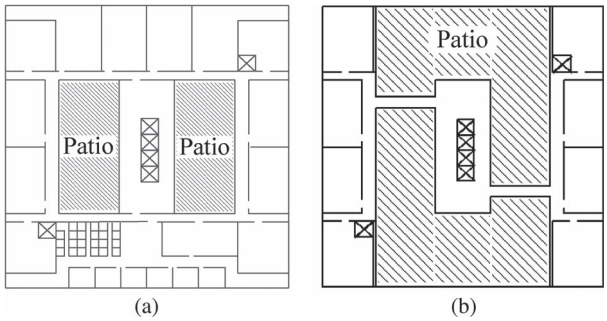

Fig. 8. Experiment environments. (a) 14th floor; (b) 13th floor.

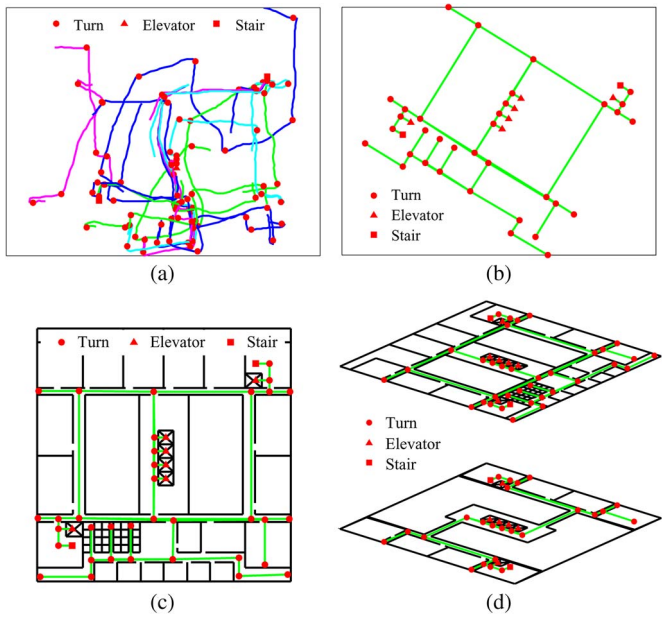

Fig. 9. Outcome of the mapping process. (a) Crowdsourcing trajectories; (b) relative map of 14th floor (2D); (c) mapping result of 14th floor (2D); and (d) mapping result of 13th and 14th floors (3D).

B. Visual Results

Before presenting the quantitative evaluation results, we first give the visual result of ALIMC.[3] Fig. 9 shows the outcome of the mapping process. Part of the trajectories inferred by PDR based on the crowdsourcing data are shown in Fig. 9(a).[4] Based on the crowdsourcing data, ALIMC could construct the indoor map of the environment. The first step is to detect the activities contained in the trajectories. The activity detection results are presented in Fig. 9(a). Attaching these detected activities with the WiFi fingerprints, activity landmarks are generated. Then, ALIMC clusters all the activity landmarks into different clusters and each cluster is treated as a node. After clustering, the relative distance between all the nodes is calculated by the method described in Section V. Based on the distance information, the relative map can be generated using the MDS technique, which is shown in Fig. 9(b). Lastly, the relative map is converted into

[3]In the visual results, the angle estimation method used is to assume that the intersection angle is right-angle.

[4]Actually, the starting points of the trajectories are unknown; to illustrate them better, we translated the starting point to the ground truth. During the experiments, we do not need to know the starting point.

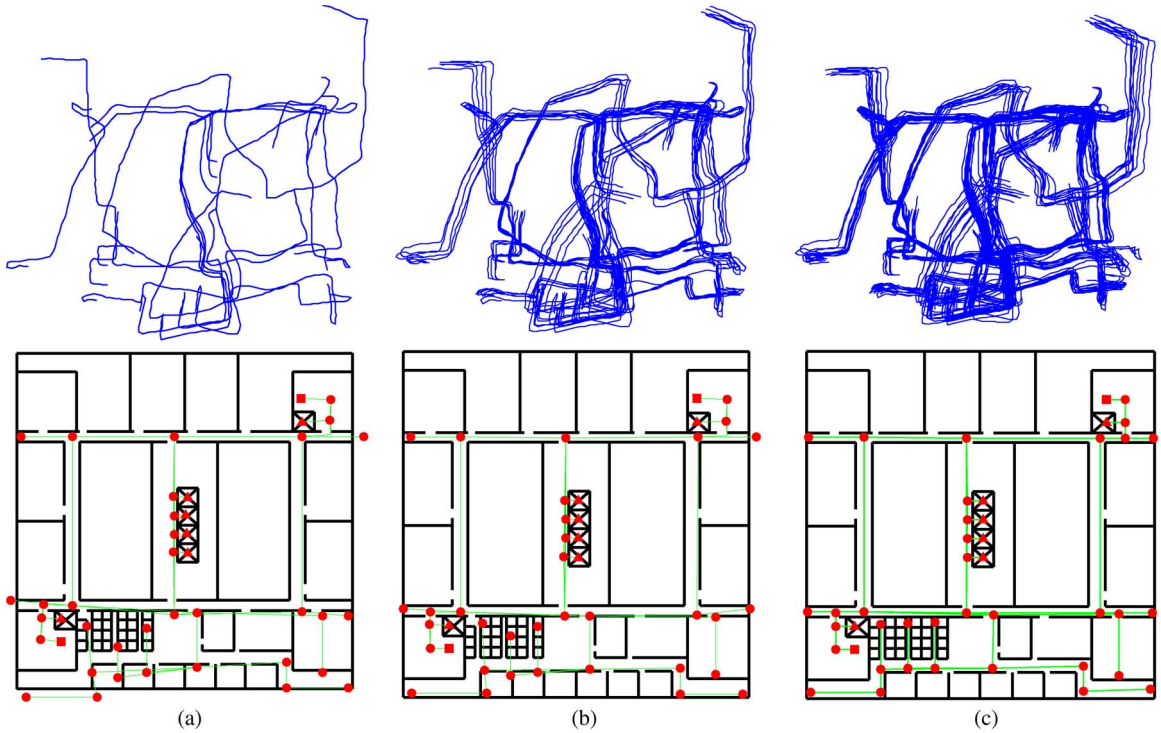

Fig. 10. User trajectories and the inferred indoor maps under different amounts of user trajectories. (a) 15 minutes; (b) 75 minutes; (c) 150 minutes.

an absolute map using the method described in Section VI. The mapping results are shown in Fig. 9(c) (2D) and Fig. 9(d) (3D).

Fig. 10 shows the collected user trajectories and the inferred indoor maps (the 14th floor) under different amounts of user trajectories. As expected, the quality of the resulting map improves with the data amount increases.

C. Metric

To quantify the quality of the inferred pathway map, the following metrics were used [18].

- Graph Discrepancy Metric (GDM): This metric reflects the differences between the nodes of the constructed map and that of the real map. Euclidean distance is used as the difference metric.
- Shape Discrepancy Metric (SDM): This metric quantifies the differences between the shapes of inferred paths and the real ones. To calculate the SDM, the link segments between nodes were uniformly sampled to obtain a series of sample points. The metric is defined as the distance between corresponding sampling points.

D. Performance

1) Performance of Different Angle Estimation Methods: To estimate the intersection angle between two connected links, four methods are used: the first method uses readings from the compass of the smartphone; the second uses the heuristic method introduced in Section V; the third method uses the interior angles of the closed polygon to compensate for the bias drift of the gyroscope; the last method is based on the observation that all the intersections of the office building are right-angle, therefore we assume that the intersection angle is right-angle.

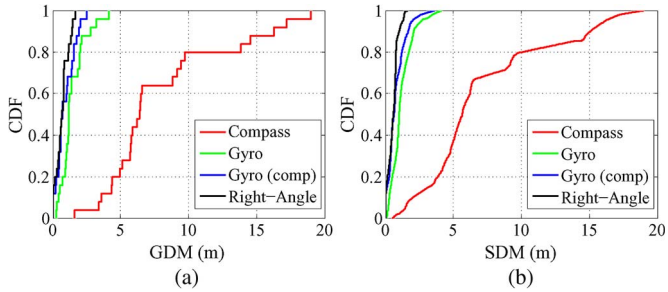

Fig. 11. Map construction performance of angle estimation method. (a) CDF of GDM; (b) CDF of SDM.

Fig. 11(a) and (b) show the Cumulative Distribution Function (CDF) of the GDM and SDM of the constructed map using all the collected data; the angle is calculated by four angle estimation methods. It can be seen that the error of the compass method is much larger than that of the other two methods. By using the heuristic gyroscope method, the map construction error is greatly improved. For GDM, the maximum error is about 4 meters, and the 80 percentile error is around 2 meters; for SDM, the maximum error is about 4 meters, and the 80 percentile error is about 1.8 meters. Using the compensation of the interior angles of the closed polygon, the map construction error can be decreased. The maximum error of the GDM and SDM are about 2.5 meters and 3.5 meters respectively. The 80 percentile error is about 1.6 meters for the GDM, and 1.3 meters for the SDM. If we make the assumption that all the intersections are right-angle, the map construction system can achieve very high accuracy. The maximum error of the GDM and SDM is about 1.6 meter.

2) Performance With Incremental Data: Figs. 12 and 13 show the GDM and SDM with incremental crowdsourcing

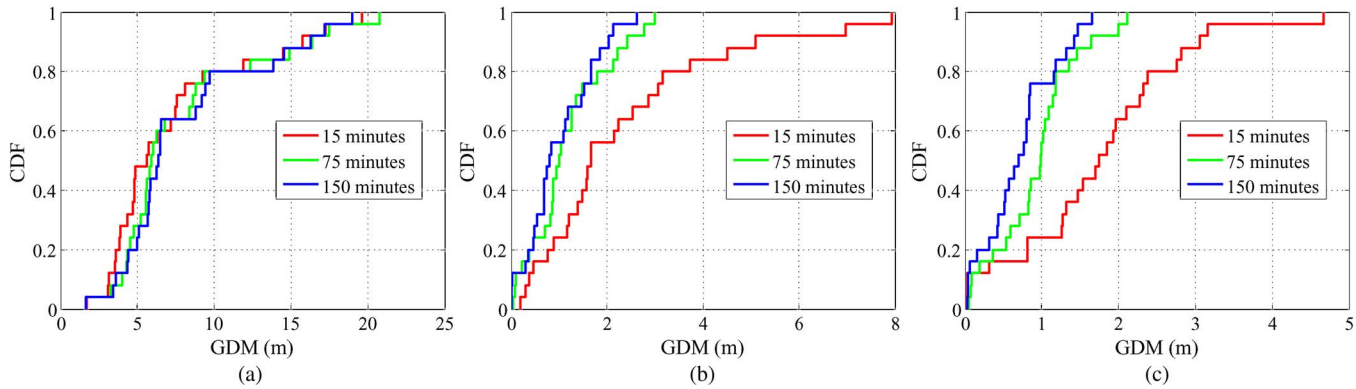

Fig. 12. CDF of GDM with incremental data. (a) Compass; (b) Gyro (comp); (c) Right-Angle.

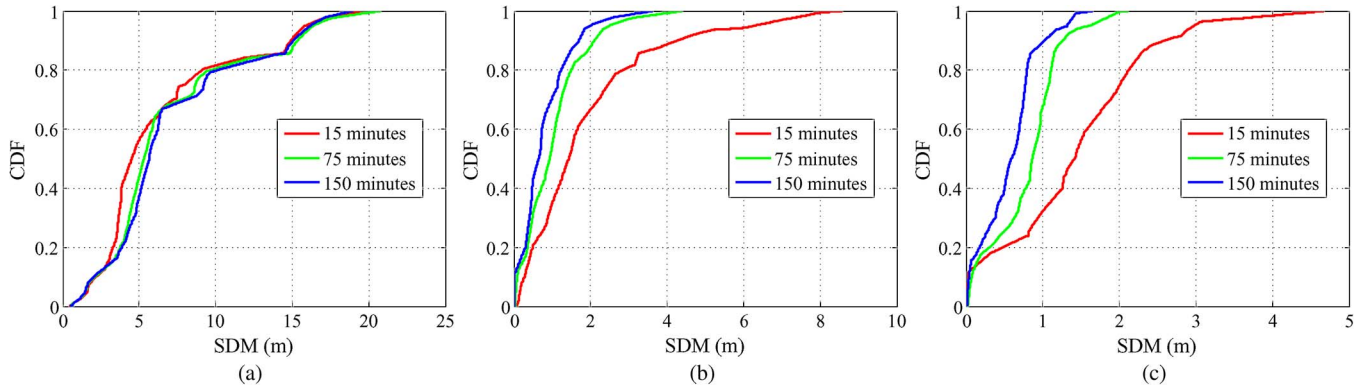

Fig. 13. CDF of SDM with incremental data. (a) Compass; (b) Gyro (comp); (c) Right-Angle.

data in terms of time. We can see that the quality of the constructed indoor map improves as the crowdsourcing data amount increases. The change trend of the gyroscope (with compensation) and right-angle method is more obvious than that of compass method. This is because the error of the compass is location-independent, which is difficult to be compensated for by crowdsourcing. Instead, the error of the gyroscope and right-angle methods can be compensated for by crowdsourcing. From Figs. 12 and 13, it can be seen that the maximum error of the GDM and SDM decreases sharply when increasing the data amount from 15 minutes to 75 minutes. For the gyroscope method, the maximum error of the GDM and SDM is reduced from about 8 meters to around 3 meters; for the right-angle method, the maximum error of the GDM and SDM is reduced from about 4.5 meters to around 2 meters.

VIII. Discussion

A. Comparisons

In contrast with our study, the SLAM-based approaches do not use crowdsourcing data for indoor map construction. We discuss our approach as compared with two state-of-the-art indoor map construction systems: CrowdInside [17] and Walkie-Markie [18]. CrowdInside does not require WiFi coverage and it works well for floorplan building. However, the trace in CrowdInside requires a GPS point as the initial point, which may not be found in practice. In CrowdInside, user motion traces are derived by PDR using the built-in initial sensors in a smartphone. The cumulative error of PDR is eliminated by the virtual landmarks. However, the locations of the virtual landmarks are inaccurate because they are estimated based on the previous traces derived using PDR. Therefore, the user motion traces are not accurate, which reduces the performance of the constructed floor plan. Walkie-Markie uses WiFi-Mark as the landmarks to anchor crowdsourcing trajectories. However, in some office buildings, there may be a higher spatial density of nodes than the spatial resolution provided by WiFi-Marks. For example, the distance between two corners of the office cubicle is usually short, and there may be not WiFi-Mark at the corners; therefore, the indoor map constructed using Walkie-Markie would not include these corners.

B. Link-Node Model

The indoor map constructed by ALIMC is a link-node model, that shows the framework of the indoor environment and can be used for indoor navigation. Moreover, the link-node model indoor map is available to aid indoor localization by the map matching method. For example, the localization approach proposed in [3] is based on the link-node model indoor map. The node attributes of the indoor map can be labeled by ALIMC, such as elevator, stairs, and corners. Some other attributes of the indoor map are also important, e.g., size of the room, width of the corridor. Some related works consider this problem [6],

[17]. The future work will be to extend our proposed ALIMC by mining more knowledge based on the crowdsourcing data.

C. Reference Points

In order to convert the relative map to the absolute map, several reference points (3 are enough) with known coordinates are needed. The reference points can be obtained using GPS data obtained near windows [31], or by some other indoor localization methods, such as by WiFi fingerprinting [32].

IX. Conclusion

In this paper, we propose ALIMC, a activity landmark-based indoor map construction system that can automatically construct an indoor map based on the data collected by smartphones. First, we detect activities based on the crowdsourcing data and use them as activity landmarks. Second, we cluster all the activity landmarks into different clusters, each of which has a unique identification number. Third, we estimate the relative distance between all the activity landmark clusters and get a relative distance matrix. Lastly, we construct the indoor map based on the relative distance matrix. The experiment results in an office building demonstrate the effectiveness of ALIMC, the 80-percentile error of the mapping accuracy is about $0.8 \sim 1.5$ meters.

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, May 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] K.-C. Lan and W.-Y. Shih, "Using smart-phones and floor plans for indoor location tracking," *IEEE Trans. Human-Mach. Syst.*, vol. 44, no.2, pp. 211–221, Apr. 2014.

[4] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 269–280.

[5] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no.4, pp. 839–848, Apr. 2013.

[6] Y. Jiang *et al.*, "Hallway based automatic indoor floorplan construction using room fingerprints," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 315–324.

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

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

[9] T. Brezmes, J.-L. Gorricho, and J. Cotrina, "Activity recognition from accelerometer data on a mobile phone," in *Proc. Distrib. Comput., Artif. Intell. , Bioinformat., Soft Comput., Ambient Assisted Living*, 2009, pp. 796–799.

[10] S. Khalifa, M. Hassan, and A. Seneviratne, "Adaptive pedestrian activity classification for indoor dead reckoning systems," in *Proc. Int. Conf. IPIN*, 2013, pp. 1–7.

[11] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Commun. Surveys Tuts.*, vol. 15, no.3, pp. 1281–1293, 3rd Quarter 2013.

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

[13] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Proc. Autonom. Robot Veh.*, 1990, pp. 167–193.

[14] H. Shin, Y. Chon, and H. Cha, "Unsupervised construction of an indoor floor plan using a smartphone," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no.6, pp. 889–898, Nov. 2012.

[15] M. Angermann and P. Robertson, "Footslam: Pedestrian simultaneous localization and mapping without exteroceptive sensors hitchhiking on human perception and cognition," *Proc. IEEE*, vol. 100, no. Special Centennial Issue, pp. 1840–1848, May 2012.

[16] M. Garcia Puyol, D. Bobkov, P. Robertson, and T. Jost, "Pedestrian simultaneous localization and mapping in multistory buildings using inertial sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1714–1727, Aug. 2014.

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

[18] 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. Implementation*, 2013, pp. 85–98.

[19] C. Luo, H. Hong, and M. C. Chan, "Piloc: A self-calibrating participatory indoor localization system," in *Proc. 13th Int. Symp. Inf. Process. Sens. Netw.*, 2014, pp. 143–154.

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

[21] S. Becker *et al.*, "Combined grammar for the modeling of building interiors," *Proc. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*, 2013, pp. 1–6.

[22] M. Mladenov and M. Mock, "A step counter service for java-enabled devices using a built-in accelerometer," in *Proc. 1st Int. Workshop Context-Aware Middleware Serv., Affiliated 4th Int. Conf. COMSWARE*, 2009, pp. 1–5.

[23] H. Wang *et al.*, "No need to war-drive: Unsupervised indoor localization," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Serv.*, 2012, pp. 197–210.

[24] S. Yang, P. Dessai, M. Verma, and M. Gerla, "Freeloc: Calibration-free crowdsourced indoor localization," in *Proc. IEEE INFOCOM*, 2013, pp. 2481–2489.

[25] L. Chen, H. Wu, M. Jin, and G. Chen, "Homogeneous features utilization to address the device heterogeneity problem in fingerprint localization," *IEEE Sens. J.*, vol. 14, no.4, pp. 998–1005, Apr. 2014.

[26] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes," in *Proc. 15th Int. Conf. Data Eng.*, 1999, pp. 512–521.

[27] D.-K. Cho, M. Mun, U. Lee, W. J. Kaiser, and M. Gerla, "Autogait: A mobile platform that accurately estimates the distance walked," in *Proc. IEEE Int. Conf. PerCom*, 2010, pp. 116–124.

[28] J. Borenstein, L. Ojeda, and S. Kwanmuang, "Heuristic reduction of gyro drift for personnel tracking systems," *J. Navigat.*, vol. 62, no.1, pp. 41–58, Jan. 2009.

[29] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "Sail: Single access point-based indoor localization," in *Proc. 12th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2014, pp. 315–328.

[30] J. Koo and H. Cha, "Unsupervised locating of wifi access points using smartphones," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no.6, pp. 1341–1353, Nov. 2012.

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

[32] P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *Proc. 19th IEEE INFOCOM*, 2000, pp. 775–784.

Baoding Zhou (S'14) received the B.E. degree in communication engineering from Shandong University, Jinan, China, in 2009. He is currently working toward the Ph.D. degree in the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China. Since March 2013, he has been also a Visiting Scholar with the Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen, China. His research interests include indoor localization and navigation, pervasive computing, and intelligent transportation.

Qingquan Li received the Ph.D. degree in geographic information system (GIS) and photogrammetry from 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.

Qingzhou Mao (M'14) received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2008. He is currently an Associate Professor with Wuhan University. His main research fields include satellite navigation system; remote sensing and geographic information system (3S) integration 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.

Wei Tu (M'14) 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 a Postdoctoral Fellow with the Shenzhen Key Laboratory of Spatial Smart Sensing and Service, Shenzhen University, Shenzhen, China. His research interests include spatiotemporal data modeling, spatiotemproal data analysis, and spatiotemproal data mining.

Xing Zhang received the B.E. and Ph.D. degrees in geographic information science from Wuhan University, Wuhan, China. He is currently with the Shenzhen Key Laboratory of Spatial Information Smart Sensing and Services, Shenzhen University. His research interests include mobile navigation, visual cognition, ubiquitous computing, and intelligent transportation.

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 in 2013, respectively.

From October 2011 to November 2012, he was a Visiting Student at National University of Singapore. 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 Assistant Professor with the School of Mobile Information Engineering, Sun Yat-sen University, Zhuhai, China. His areas of interest include computer vision, point cloud processing and unmanned autonomous vehicles.