# Infrastructure-Free Collaborative Indoor Positioning Scheme for Time-Critical Team Operations

Youngtae Noh, *Member, IEEE*, Hirozumi Yamaguchi, *Member, IEEE*, and Uichin Lee, *Member, IEEE*

***Abstract*—Indoor localization is the key infrastructure for indoor location-aware applications. In this paper, we consider an emergency scenario, where a team of soldiers or first responders perform time-critical missions in a large and complex building. In particular, we consider the case where infrastructure-based localization is not feasible for various reasons such as installation/management costs, a power outage, and terrorist attacks. We design a novel algorithm called the collaborative indoor positioning scheme (CLIPS), which does not require any pre-existing indoor infrastructure. Given that users are equipped with a signal strength map for the intended area for reference, CLIPS uses this map to compare and extract a set of feasible positions from all positions on the map when the device measures signal strength values at run time. Dead reckoning is then performed to remove invalid candidate coordinates, eventually leading to only correct positions. The main departure from existing peer-assisted localization algorithms is that our approach does not require any infrastructure or manual configuration. We perform testbed experiments and extensive simulations, and our results verify that our proposed scheme converges to an accurate set of positions much faster than existing noncollaborative solutions.**

***Index Terms*—Distributed indoor positioning systems, indoor navigation, infrastructure-free indoor localization, time-critical team operations.**

## I. Introduction

LOCATION-based services basically aim to deliver customized services based on people's locations. In the outdoor environment, location-based services can be relatively easily delivered with well-known location services such as global positioning system (GPS) and cell-tower localization. However, such technologies do not work well, and existing indoor location services usually require dedicated infrastructure indoors such as acoustic beacons [1]–[3], Wi-Fi access points (APs) [4]–[10], and radio-frequency identification (RFID) tags [9], [11]–[13].

In this paper, we consider a time-critical indoor scenarios with teams of soldiers or first responders performing emergency missions (e.g., firefighting, urban military, and search/rescue missions). If accurate and fast location services are available, the team members can easily access a region of interest for mission operations. However, in emergency scenarios, infrastructure-based location services may face several problems. Deploying infrastructures such as acoustic, Wi-Fi, and Bluetooth beacons may not be feasible in time-critical missions. In practice, it might be neither economically feasible nor practical to maintain such localization infrastructure for emergency missions in every building in advance. In some instances, infrastructure might not even be available because of a power outage or terrorist attacks.

To summarize, our localization problem for time-critical mission operations has several unique constraints: no existing infrastructure can be used for localization (infrastructure-free) and team members cannot manually configure localization schemes (configuration-free). The most widely-accepted solution is inertial navigation (i.e., dead reckoning)—a user's current position is tracked with inertial sensors (e.g., gyroscopes and accelerometers), by monitoring movement trajectories and possibly matching a user's current position in a map.

However, the well-known problems of inertial navigation include large errors and slow convergence since indoor position landmarks are lacking; these problems may cause detrimental effect for time-critical indoor scenarios. When a team explores a large and complex building, this problem becomes even worse. For example, a team may enter an underground parking structure and then take an elevator to a locus of events on the fifth floor. The incremental rate in the position error could be higher than the rate at which the position can be fixed using map matching with turn detection [14], [15]. For accurate indoor localization, it has been well-accepted that we must make use of additional mechanisms such as manual configuration and infrastructure support. While peer-assisted localization schemes collaboratively use peer measurements, they generally require infrastructure support to improve localization accuracy. For example, Liu *et al.* [16] used peer-based acoustic ranging to improve the accuracy of

Manuscript received April 9, 2014; revised September 20, 2014, February 22, 2015, October 16, 2015, and February 17, 2016; accepted October 3, 2016. Date of publication November 17, 2016; date of current version February 14, 2018. This work was supported in part by the Japan Society for the Promotion of Science KAKENHI under Grant JP15H02690 and Grant JP15K12019, in part by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education under Grant NRF-2016R1C1B2011415, in part by the Inha University Research under Grant 54465-01, in part by the KUSTAR-KAIST Institute, and in part by the Institute for Information and Communications Technology Promotion funded by the Korea Government (MSIP) (Development of Device Collaborative Giga-Level Smart Cloudlet Technology) under Grant B0101-15-1272. This paper was recommended by Associate Editor N. N. Xiong. *(Corresponding authors: Youngtae Noh and Uichin Lee.)*

Y. Noh is with the Department of Computer Science and Information Engineering, Inha University, Incheon 402-751, South Korea (e-mail: ytnoh@inha.ac.kr).

H. Yamaguchi is with the Department Information Networking, Osaka University, Osaka 565-0871, Japan (e-mail: h-yamagu@ist.osaka-u.ac.jp).

U. Lee is with the Department of Industrial and Systems Engineering, KAIST, Daejeon 34141, South Korea (e-mail: uclee@kaist.edu).

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/TSMC.2016.2615652

radio-frequency (RF) fingerprint-based positioning. In addition, peer nodes can collaboratively perform simultaneous localization and mapping (SLAM) by combining received signal strength (RSS) measurements from APs close to the peers [17]–[21]. Unfortunately, existing solutions are not particularly suitable for our emergency scenario because of its configuration-free and infrastructure-free requirements.

In this paper, we propose a localization algorithm called collaborative indoor positioning scheme (CLIPS). Our algorithm leverages peer-to-peer Wi-Fi beaconing and accurate smartphone dead reckoning. In CLIPS, obtained signal strength (RSS) values are measured from other mobile peers. Given that a response team can obtain a floor-plan for safety operations, we can build a realistic RSS estimation map in advance. For example, we can use a realistic wireless signal propagation model like ray-tracing. This estimation map allows the team members to find a set of feasible coordinates in which the estimated RSS values are matched with the measured RSS values at run time.

To deal with wireless signal variations due to channel fading, we ensure that a position match can happen when RSS value differences are within the threshold value. This process can initially produce many false candidates, but false candidates can be easily removed by leveraging dead reckoning. That is, invalid candidates will lead quickly to dead ends as a user moves along the corridors. This process will be repeated such that each member's real position can be estimated rapidly. In general, as the number of members increases, the convergence time decreases as well. In order to conserve battery, Wi-Fi beaconing can be stopped when nodes? Position fixes are entirely acquired; each user's current position will be tracked by dead reckoning continues. In the next processes, the search space (and convergence time) will be drastically reduced, because it is enough to only consider the locations closest to a node's current location (e.g., within a fixed radius).

Concretely, this paper makes the following contributions.

1) We propose CLIPS, which uses collaborative Wi-Fi beaconing and smartphone dead reckoning. The CLIPS novel positioning algorithm does not require any infrastructure support or manual configuration, which are the main departures from existing peer-assisted-based localization schemes [5], [16], [19]–[21].
2) We demonstrate the feasibility of CLIPS. The field test confirms that CLIPS can accurately acquire position fixes significantly faster than conventional approaches that are based noncollaborative schemes; e.g., the required travel distance to obtain an accurate fix took less than half that of a noncollaborative scheme (a more than 50% improvement in the considered scenarios).
3) We further perform an extensive simulation study to investigate the impact of various parameters (e.g., the slack variable, number of nodes, mobility patterns, and location sharing) on the system performance under various scenarios. Overall, we find similar trends to those in the test-bed experiments. In particular, we confirm that convergence speed exponentially decreases with the team size, RSS matching contributes to a more than 50% accuracy improvement, and sharing location updates significantly improves convergence speed.

The rest of this paper is structured as follows. We present related work in Section II. In Section III, we overview CLIPS and in Section IV, we review the core elements of CLIPS. We illustrate our evaluation results based on the testbed experiments and simulations in Sections V-A and B, respectively. We discuss the improvements and limitations of the current work in Section VI and this paper is concluded in Section VII.

## II. Related Work

We first provide a general overview of indoor positioning. CLIPS leverages multiple techniques, i.e., RF modeling, P2P communication (peer assisted), and dead reckoning, for infrastructure-free instant localization. Thus, we detail each of these key components and illustrate how CLIPS combines them.

### A. Indoor Positioning Overview

Indoor positioning has been intensively investigated recently to provide accurate position information in indoor environments, where GPS is not accessible. Existing indoor localization schemes can be classified into two categories, namely, infrastructure-based and infrastructure-free systems. Previous indoor positioning schemes used specialized distance sensing devices, such as RFID tags [13], ultra-sound transmitters/receivers [1], [22], [23], and infrared IR beacons/receivers [3]. For example, Want *et al.* [3] proposed Active Badge, where users wear a small badge that periodically transmits a unique infrared signal, and each room has networked sensors on the ceiling for accurate positioning. Similarly, ultrasound-based ranging is used in existing systems, e.g., Cricket [22], Active Bat [23], and WALRUS [1]. LANDMARC [13] uses RFIDs, where active tags periodically transmit signals, and readers measure the signal strength for localization. Recently, researchers also showed that frequency-agile wireless networks (i.e., in-band frequency hopping) can be used for distance estimation. ToneTrack estimates the time-of-arrival (ToA) of a mobile node's transmission at pairs of APs in the network by analyzing the correlation between incoming signals on different subcarriers [24]. It then compares the time differences of the ToA readings across pairs of APs to estimate and refine the mobile node's location.

In contrast to these systems, which require additional infrastructure installation, researchers have also proposed to leverage existing RF infrastructure, such as Wi-Fi and cellular communications. In RADAR [9], an RSS map is built *a priori* (using Wi-Fi signal strength vectors in the map). For a given signal strength vector, the location of a node is then estimated using a nearest neighborhood matching algorithm in the map. In Horus [25], a stochastic description of the RSS map is used for maximum likelihood-based location estimation. Cellular signals also have been used for localization [26], e.g., signal-strength fingerprints from multiple cell towers. However, if RF infrastructure is uniformly deployed, it would be difficult to build a dense fingerprint map across an entire building.

To solve this nonuniform map problem, Modellet [27] unified a model-based and fingerprint-based approach such that a set of high-quality fingerprints are used to populate virtual fingerprints at positions without samples via a model-based approach.

Recent commercial solutions tend to use multiple location sources, such as GPS, cell towers, and Wi-Fi hotspots; e.g., SkyHook [6], Google Latitude [8], and PlaceEngine [28]. Recent studies have attempted to use other types of fingerprints. In SurroundSense [29], environment sensing data (e.g., ambient sound and light conditions) were used for indoor place detection. Likewise, given that an acoustic background spectrum is fairly unique in each place, BatPhone [2] uses this fingerprint for place recognition. Sextant [30] leverages environmental landmarks such as store logos. It uses image processing and databases to identify the landmarks and lets smartphones obtain relative position measurements. However, all these schemes require building fingerprint databases and mostly rely on infrastructure (e.g., Wi-Fi hotspots); thus, they are not suitable for infrastructure-free instant localization. A more complete survey of existing localization schemes can be found in recent surveys [31], [32].

### B. Model-Based Localization

Log-distance path loss (LDPL) as an RF propagation model, can be used to forecast RSS at various indoor locations [5], [33]. A clear benefit from this model is that it obviates the need for an RSS map building process at the expense of decreased localization accuracy. Due to large diversities of RF propagation characteristics, the model parameters would have to be estimated particularly for each indoor space of interest. In TIX [34], knowing the transmit power and locations of all Wi-Fi APs, the APs are modeled to measure the RSS of the beacons from nearby APs, and linear interpolation finally estimates the RSS at every location in the indoor space. ARIADNE [35] also deploys sniffers at known locations, but employs a more sophisticated ray-tracing model based on detailed indoor maps and uses simulated annealing to estimate radio model parameters. Lim *et al*. [36] employed Wi-Fi sniffers at known locations, and the LDPL model was used to construct an RSS map. When there are peer mobile devices nearby, EZ can collect RSS measurements of APs at the peer devices to find LDPL radio model parameters—each node's position and radio parameters are treated as unknown variables that can be uniquely identified by using the RSS measurements of APs at multiple nodes [5].

CUPID is based on the angle and distance estimation between an AP and a mobile node [37]. To accurately estimate distance with the path loss model in an environment with multipath fading, CUPID differentiates direct path (line-of-sight) signals from reflected path signals via the power delay profile of the channel state information. Direct path signal strength information is then used to empirically find path loss exponents. An angle-of-arrival (AoA) calculation that is based on the phase difference between the signals that arrive at two antennas is also affected by multipath fading, because direct and reflected paths result in different angles. CUPID mitigates this problem by leveraging a user's mobility; i.e., it computes distance traveled via dead reckoning, and then a cosine rule is used to estimate the change in the AoAs of direct path signals. Conventional localization requires multiple APs for multilateration, but as shown earlier, RF infrastructure may not be uniformly distributed. To mitigate this problem, Mariakakis *et al.* [38] proposed a solution called SAIL that only uses a single AP. When a user is mobile, it continuously measures the distance between the client and AP. At the same time, it estimates a user's distance traveled via dead reckoning and the overall compass heading during that travel (this obviates the need for AoA estimation). Thus, a user's location can be geometrically determined using a single AP.

More recently, to cope with random signal fluctuations, He *et al.* [39] proposed a novel indoor localization approach, named as Wi-Dist, which fuses noisy wireless fingerprints with uncertain mutual distances given by their bounds. Another method along similar lines, Wu *et al.* [40] proposed the automatic and continuous radio map updating service, which takes advantage of the static behavior of mobile devices. This system uses off-the-shelf mobile devices as movable reference points. These devices collect the up-to-date fingerprints when they are static at certain locations in real-time and are accurately pinpointed by a novel trajectory-matching algorithm. The underlying RSS relationship among neighboring locations is stable over time with the help of newly collected data from the reference points.

Considering time-critical scenarios as discussed earlier, the main departure of CLIPS from previous solutions is that it leverages mobile Wi-Fi beacons (i.e., those of emergency team members) to determine a set of feasible coordinates from all coordinates on an intended map, and then employs dead reckoning over the map to eliminate invalid candidate locations and finally pinpoint the correct/real positions.

### C. Peer-Assisted Localization

The high density of smartphones/hand-held devices in public spaces enables the peer-assisted localization of nearby peer devices. Proximity sensing schemes, such as Hummingbird [41] and NearMe [42], provide relative localization of nodes with an accuracy of 30–100 m. Hummingbird uses dedicated RF ranging methods, whereas NearMe uses Wi-Fi fingerprints to detect whether two nodes are within short range (if there is at least one AP in common) or long range (if they can reached by hopping through APs with overlapping coverage) of each other. Bluetooth (e.g., BlueHoo [43]) and cellular signals (e.g., People-Tones [44]) can be used for proximity sensing nearby devices, but its positioning accuracy is largely dependent on radio characteristics. Apple's iBeacon uses Bluetooth low energy, in which beacon nodes periodically send unique IDs (say every 100 ms), for proximity sensing. Upon receiving an ID beacon, nearby devices can look up the ID to determine the device's physical location or to trigger associated services. Similar to NearMe, iBeacon categorizes proximity into three classes: 1) immediate (located within a few centimeters); 2) near (located within a couple of meters); and 3) far (located further than 10 m away).

In contrast to Wi-Fi and Bluetooth, acoustic ranging methods (e.g., Beep Beep [45]) can provide much higher accuracy than other radio-based methods. Recently, Banerjee *et al.* [46] proposed the Virtual Compass, where multiple radios are used to improve the ranging accuracy (i.e., Wi-Fi and Bluetooth), and Vivaldi-based distributed localization is used to produce the relative positions of nodes. Higuchi *et al.* [47] proposed "stop-and-go" collaborative localization that leverages accurate ranging methods among mobile peers to estimate their moving states. As a result, more stationary nodes can be chosen as relative anchors to improve accuracy. Liu *et al*. [16] used both Wi-Fi-based positioning and acoustic ranging among peers such that acoustic-based relative positions are mapped into the *absolute positions* by carefully referencing Wi-Fi-based localization results. EZ can be considered a peer-assisted localization because the RSS measurements of APs at the peer devices are used for localization [5]. Furthermore, as shown later, multiple nodes can collaboratively perform SLAM methods [17]–[21], which can be also considered as peer-assisted localization. This paper can also be considered as peer-assisted localization in that RSS measurement among peers are used for identifying feasible coordinates in the RF map, but the key difference from earlier work is that our approach supports absolute localization and uses collaborative dead reckoning, in which each node performs map matching to eliminate infeasible coordinates and shares this information for fast convergence.

### *D. Robot Localization and Dead Reckoning*

For the robot indoor navigation, a robot basically requires an ability to determine its real location. Simple approaches provision the robot with an indoor map that allows the robot to determine its real location by comparing its observed environment to the map (using, e.g., ultrasound or LADAR sensors). Significant advances in robot mobility research include SLAM [14], which allows a robot to build a map of the indoor environment (in terms of walls and other obstructions), and, at the same time, determine its position based on the constructed map. In order to determine the distance having moved between measurement points, a robot typically uses an onboard odometer. For pedestrian navigation, Woodman and Harle [48] showed that the use of a foot-mounted inertial unit and particle filter can provide accurate position estimation. Lan and Shih [49] improved the accuracy of pedestrian dead reckoning by proposing an elaborate gait analysis technique for accurate step recognition and step length estimation as well as a turn detection method for map matching based on gyroscope data. MapGENIE [50] is recent work on building indoor maps using pedestrian dead reckoning. It relies on crowdsourcing to collect the traces of residents. In addition, Sorour *et al.* [51] proposed joint indoor localization and radio map construction. The aim of this paper is to reduce extensive deployment efforts for Wi-Fi radio map construction. This paper differs from these earlier studies in that we use P2P Wi-Fi fingerprinting and dead reckoning for fast location convergence in time-critical mission operations.

Recent studies have also explored the use of wireless infrastructure to better sense the environment [15]. Similarly, Martin *et al.* [52] proposed a smartphone indoor localization that uses only the smartphone sensors and integrates both online and offline RSS fingerprinting phases in a local smartphone. Kim *et al.* [17] proposed a method for improving the localization accuracy of dead reckoning, where multiple nodes collaboratively build the global RSS map, and drift errors are corrected by referring to this map. SmartSLAM implements SLAM to automatically construct an indoor floor-plan and radio fingerprint map using off-the-shelf smartphones [18]. The cost of RSS map building can be significantly reduced by leveraging the fact that RSS fingerprints are geographically connected as a user is moving, and thus, a high-dimensional fingerprint space can be formed. LiFS [21] uses multidimensional scaling to map fingerprint space into 2-D space, which can be easily mapped onto a given map. UnLoc [19] leverages the fact that motion sensor and Wi-Fi readings collected by multiple users can be clustered to generate landmarks through which the drift errors of dead reckoning can be corrected. PiLoc [20] uses trajectory based clustering to improve performance, i.e., trajectories are first clustered based on AP similarity, path segments (turn and line vectors) are then clustered based on trajectories and RSS signal similarity, and path segments are finally merged to build floor-plans. These RSS map-based methods cannot be used in our scenarios because of their dependence on Wi-Fi infrastructure. Furthermore, landmark generation and trajectory clustering basically assume that there is sufficient sensor data collected from mobile users, but this assumption does not hold in time-critical mission operations.

## III. CLIPS Overview

We would like to emphasize that this paper considers emergency indoor scenarios with first responders performing emergent missions (e.g., firefighting, urban military, or rescue operations) or teams of soldiers. In this scenario, accurate and fast localization would facilitate team members to easily navigate a region of interest, and sharing situational-awareness would finally lead to successful mission completion. However, we recognize there are limits to the accuracy of RSS-based map mapping—it does not deterministically identify exact locations, although we have introduced a slack variable to accommodate RSS variance and estimation errors and produce an acceptable number of feasible coordinates/positions. Because of this design choice, we need an additional component that can uniquely lead to a user's real location/position from the feasible coordinates. The most common solution is inertial navigation (i.e., dead reckoning), which tracks a user's current position by continuously monitoring heading changes and distance traveled with inertial sensors (e.g., gyroscopes and accelerometers), and finally leads to current position of the user in the map.

We assume that a team has obtained a floor-plan in advance [an example is shown in Fig. 1(a)]. It is preprocessed to build an overlay of an $M \times N$ grid, as shown in Fig. 1(b). The grid dimensions are determined by the granularity of the location

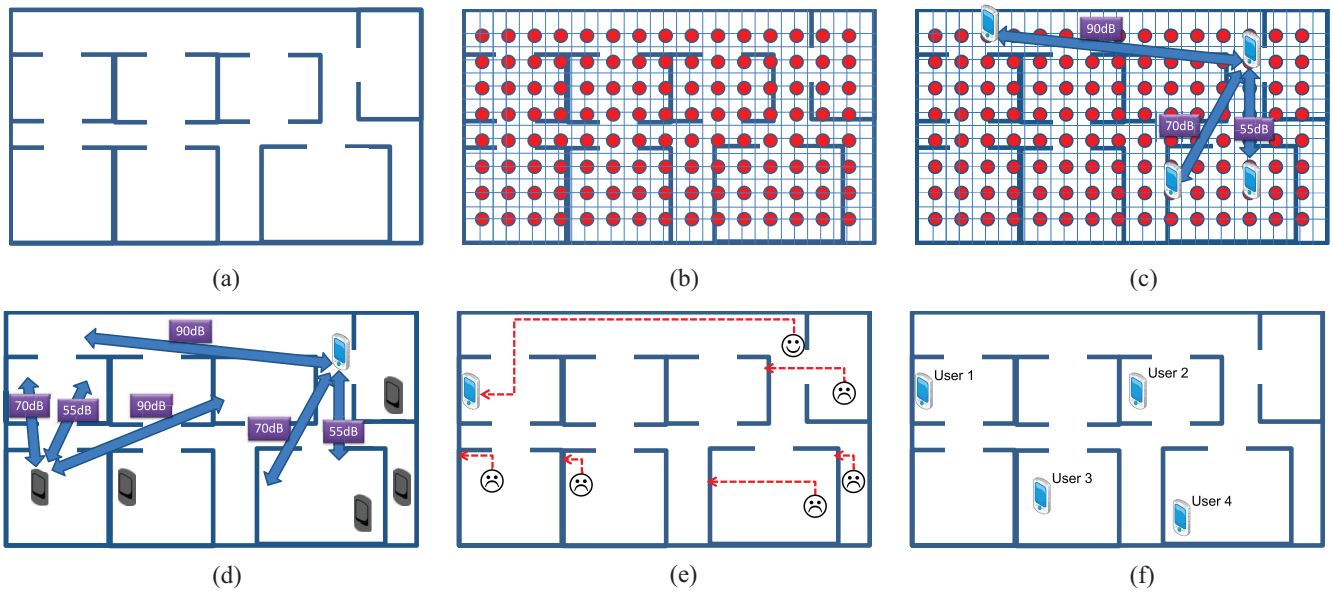


Fig. 1. CLIPS overview [53]. (a) Acquiring the floor-plan. (b) Generating an RSS map using a ray-tracing-based RF simulation. (c) Measuring the RSS values to all reachable members via Wi-Fi beaconing (e.g., top-right node). (d) Identifying the feasible coordinates of each node (e.g., one valid and five invalid coordinates indicated by the white and black phones, respectively). (e) Eliminating invalid candidates via dead reckoning (e.g., all five invalid coordinates). (f) Sharing the discovered position with other team members (via a cellular network or Wi-Fi) such that others can further remove invalid coordinates.

information required for mission operations; we use a $2 \times 2$ m grid in our field test. In addition, for all coordinates on the floor, we calculate the path loss values from every other coordinate on that floor, using an RF ray-tracing model, and generate an *RSS map* of $M \times (N-1)$ dimensions.

Fig. 1(c) shows what would happen once a team of four members step on the floor in the intended building. Each mobile node performs periodic Wi-Fi beaconing through which the RSS values of the reachable team members can be observed. As shown in Fig. 1(d), these RSS values are then matched against the RSS map downloaded in advance, and this generates a list of all feasible coordinates on the floor. We present the detailed matching algorithm in Algorithm 1 (see Section IV-B). After obtaining these feasible coordinates, team members employ smartphone dead reckoning to eliminate all false positives from the list. According to heading changes and distance traveled on the map, each member applies the displacement on the list of all feasible coordinates. In the meantime, if the coordinates lead to a dead end, those invalid coordinates are eliminated from the list. An example is shown in Fig. 1(e). This process is repeated until each member gets an unique coordinate from the list. As shown in Fig. 1(f), once each member localizes herself correctly, she can broadcast her position on the map to the remaining nodes such that she can help others to remove invalid coordinates on their lists quickly. This also allows each team member to know the position of all other members, and helps to facilitate the team mission.

## IV. CLIPS System Details

In this section, the main components of CLIPS will be provided: 1) floor-plan preprocessing and RSS map generation; 2) feasible coordinate estimation; and 3) map matching via indoor path tracking.

### A. RSS Map Generation

CLIPS requires a floor-map for each site, which can be obtained from geographic information system (GIS) data. We assume that a team under time-critical indoor operation can access a digital map from a service provider that recognizes a given floor-plan image and builds a digital floor map—architectural floor-plan image recognition has been studied in the field of GIS [54]. Similar services have recently been launched, such as "Google Maps Floor Plans," for indoor wireless LAN positioning, which obtain floor-plan images uploaded by users [55], and floor level information can be accurately identified by using existing algorithms (e.g., FTtrack [56]).

Next, the LDPL model as an RF propagation model is employed to forecast RSS values for all coordinates on the intended floor-plan, and to obtain path loss data after simulation. Using this model reduces the number of RSS measurements significantly compared to RF fingerprinting schemes, albeit at the expense of degraded localization accuracy. Given that RF propagation characteristics vary widely (especially indoors), the model parameters would have to be estimated specifically for each indoor space of interest. For example, Anderson and Rappaport [57] presented measured data for a 2.5 GHz in-building partition loss.

Ray-tracing has widely been used because ray-tracing models consider the details of a place, such as thickness of walls, doors, and windows as well as desks and chairs to achieve high fidelity. In general, ray-tracing approximates radio propagation using a finite number of isotropic rays emitted from a transmitting antenna by a ray imaging technique and can deliver high fidelity. In this technique, the transmitter is assumed to be reflected at each surface around it to produce image transmitters; the rays reflected to the receiver from the real transmitter are considered to be direct paths from the mirror

images of the true transmitter. Based on geometrical optics, each ray from the transmitter to the receiver can be exactly determined. However, the major drawback of such techniques is their expensive computational complexity. Readers can refer to the detailed ray-tracing techniques in [35] and [58]–[63]

All models can be used in our extensive simulation. However, in order to pursue a reasonable balance amongst the effort of floor map modeling, and accuracy, and simulation duration, we use the following modeling and simulation: we consider all the solid lines in a floor-plan as walls, all the spaces surrounded by the solid lines as accessible spaces, and all windows are regarded as walls. With this indoor modeling, we use a simplified (reduced) 3-ray tracing in the experiments (provided by Wireless InSite software [64]), where diffraction along obstacles is considered. In practice, however, the measured signal strengths tend to fluctuate because of small-scale fading. This phenomenon can be mitigated by conducting multiple measurements to determine the average RSS values, but a node might still suffer from short-term fluctuations (thereby removing the true position). Moreover, floor-plans that are partially outdated because of recent remodeling, missing obstacles (e.g., furniture or refrigerators), human movements, and different Wi-Fi chipsets can also affect the RSS values. To address these problems, we introduce a slack parameter $\alpha$ (usually $\pm 13$ dB in our settings) to reduce the sensitivity to RSS measurements on the overall system performance. We also believe that the calibration process of Wi-Fi chipsets will further opt-out false-positive positions by decreasing the $\alpha$ value; however, this will be a part of our future work. We justify this approach with test bed and simulation results in Section V-A.

### *B. Feasible Coordinate Estimation*

Node $j$ estimates the path loss value in this communication [denoted as $m(i,j)$] from each beacon log $(i,j,ss)^t$ of node $j$, where $ss$ is the RSS from node $i$ at time $t$. To this end, we assume that transmission power and other factors, such as sender antenna gain/loss and environmental noise, are common and almost constant on all mobile nodes. We note that these values might be hardware-dependent,[1] but premeasurement before localization (before a mission starts) is an effective way to determine such values.

*1) Positioning Problem Formulation:* In this section, we focus on the positioning activity of node $j$. Given a floor-plan of $fp$ with $N$ grid points, let $L : N^2 \rightarrow \mathcal{R}^+$ denotes a path loss matrix among the $N$ points, where element $(u,v) \in N^2$ is the simulated path loss from points $u$ to $v$. In addition, let $M_j$ denotes the set of node $j$ and the nodes from which it received beacons at time $t$. Let $m : M_j \times \{j\} \rightarrow \mathcal{R}^+$ denotes the set of path loss measurements of node $j$ at time $t$, where $m(i,j)$ is the measured path loss value from nodes $i$ to $j$.

The node $j$'s positioning problem is to select the grid point that has the minimum distance from the true position of node $j$.

[1]We introduce a slack parameter $\alpha$ and perform experiments to find emphatically correct value (shown in Section V-A2, $\pm 13$ dB in our settings) to reduce the dependency as we cannot completely consider potential changes of floor-plans, missing obstacles (e.g., furniture, refrigerator, etc.), human movements, and different Wi-Fi chipsets to obtain RSS values.

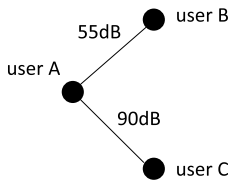


Fig. 2. Example of pass loss measurements.

Our approach is to locate such a position by finding the positioning function $p : M_j \rightarrow N$ with the least "path loss matching error" between $m$ and $L$. The objective function is minimizing such path loss matching error as follows:

$$\min \sum_{i \in M_j} |m(i,j) - L(p(i), p(j))|. \tag{1}$$

We note that we may use different objective functions depending on signal attenuation at the target site. For example, if the path loss increases as the power of the distance, we may use the following function instead of (1):

$$\min \sum_{i \in M_j} (m(i,j) - L(p(i), p(j)))^{\gamma} \tag{2}$$

where $\gamma$ is the pass loss exponent (e.g., $\gamma = 2$ in the free space attenuation model). More general propagation models exist. For indoor space, there is a well-known model in ITU-R P.1238-8 [65]. Assuming 2.4 GHz propagation on the same floor with the reference distance 1 m, the P.1238-8 model has the form "$30 \log_{10} d - 39.60$." By transforming this log-based model into the exponent-based model, the coefficient "30" corresponds to the pass loss exponent $\gamma = 3$ in the propagation model. We note that shadowing loss and multipath fading are not usually considered in the general (i.e., site-independent) propagation models because of their time-varying and situation-dependent features and we therefore ignore them in the objective function. Nevertheless, the path loss characteristics are highly situation-dependent, and it is not straightforward to choose the best-fit exponent. Consequently, we adopt the simplest form for (1) considering the simplicity of its calculation.

As addressed above, we need to accommodate measurement errors caused by the fluctuation of RSS values. We should allow some deviation parameter $\alpha$ such that a measured path loss $l'$ and a simulated path loss $l$ are regarded as identical if $l' \in [l - \alpha, l + \alpha]$. The choice of an appropriate $\alpha$ is further investigated in Section V-A. Using function $z$, where $z(l', l) = 1$ if $l' \in [l - \alpha, l + \alpha]$ and 0 otherwise, we can then use the following objective function instead of (1):

$$\max \sum_{i \in M_j} z(m(i,j), L(p(i), p(j))). \tag{3}$$

This leads us to find the $p$ that maximizes the number of path loss-matched edges in $m$ and $L$. A simple example is given in Figs. 2 and 3. Fig. 2 shows the path loss measurements of the users, and Fig. 3 shows an RSS map generated from a floor-plan by a simplified 3-ray trace model. Using (1) as an objective function, $p = \{userA : point3, userB : point2, userC : point1\}$ minimizes the

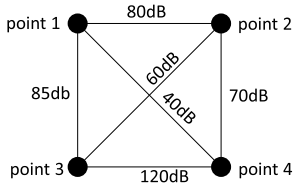


Fig. 3. Example of an RSS map generated from a floor-plan.

**Algorithm 1** Positioning Algorithm for Node $j$

```
1: procedure find_possible_positioning_functions (M_j, m, N, L)
2: k ← 0; F = ∅
3: for each v ∈ N do
4:     E ← (j, v)
5:     for each pair of edges (i, j) ∈ m and (u, v) ∈ N^2 do
6:         E ← E ∪ (i, u) if z(m(i, j), L(u, v)) = 1
7:     end for
8:     E' ← bipartite graph matching result for ((M_j, N), E)
9:     if |E'| > k then
10:        F ← E'; k ← |E'|
11:    else if |E'| == k then
12:        F ← F ∪ E'
13:    end if
14: end for
15: return F as a set of positioning functions
16: end procedure
```

path loss matching error (i.e., $|55-60|+|90-85|$). However, to relax sensitivity and find a larger set of feasible points, we can use (3). Using this objective function with the slack value $\alpha = 10$, the path loss matching error obtained by positioning function $\{userA : point2, userB : point3, userC : point1\}$ becomes equivalent to that obtained by another positioning function $\{userA : point3, userB : point2, userC : point1\}$. Thus, the results will contain a larger set of feasible points.

*2) Algorithm and Complexity:* Assume a graph $G = (M_j, m)$ and a complete graph $H = (N, N^2)$. The above optimization problem is the maximum common subgraph isomorphism (MCSI) problem in graph theory, which finds an induced graph $G'$ of $G$ in $H$ with the objective function in (3). Although the general MCSI problem is known to be NP-hard, our problem is a special class of MCSI in that we have a graph $G$ with a star topology centered at node $j$. Therefore, if $j$ is allocated to point $v \in N$, the calculation of the objective function in (3) can be reduced in this case to the following problem: 1) for each edge $(i, j) \in m$ find all possible edges $(u, v) \in N^2$ that satisfy $z(m(i, j), L(u, v)) = 1$ and 2) find the positioning function $p : M_j \rightarrow N$ from 1) with the objective function in (3). Part 1) needs to exhaustively test all edge pairs in $G$ and $H$, respectively, and part 2) can be reduced to the *maximum bipartite matching problem* on the graph, where $M_j$ and $N$ are bipartite vertices, $(j, v)$ are edges, and all possible allocations of the nodes in $M_j$ to points in $N$ found in part 1) are also edges. The computation complexity of part 1) is $O(|M_j||N|)$, and that of part 2) is also $O(|M_j||N|)$ by the path matching algorithm [66]. Thus, to apply parts 1) and 2) to all points in $N$, the complexity of our optimization algorithm is $O(|M_j||N|^2)$. The pseudocode is given below.

The algorithm works as follows. In the loop between lines 3 and 14, first, it is examined whether a pair of an edge from node $j$ on the path loss measurement graph and an edge from point $v$ on the simulated path loss graph satisfies function $z$ or not in the loop between lines 5 and 7. Then the bipartite matching algorithm is applied to set $E$ of feasible edges in line 8. Finally, the optimality of this result is examined compared with the best record in terms of the number of matched edges between lines 9 and 13.

### C. Location Convergence via Indoor Path Tracking

The mechanism for our indoor path tracking (i.e., dead reckoning) over a floor-plan uses commercial off-the-shelf (COTS) smartphone sensors to obtain movement distance and heading changes. The major challenge here is to use the unreliable inertial sensors in smartphones (e.g., magnetometers and accelerometers) to accurately track a user's path.

We use Android's magnetic field sensors and heading to identify the direction of movement. During our initial implementation, we observed that the readings from a compass have an internal bias in addition to the fluctuations caused by even a slight sway, irregular motion, or by magnetic fields in the surroundings. We use a method recently proposed by Constandache *et al.* [67] to overcome above errors and detect significant heading changes accurately at a corner while taking a turn and further use the following condition for more accurate turn detection:

$$\text{Avg}\big(t_{(i+1)}\big) - \text{Avg}(t_i) \geqq \frac{\text{StdDev}(t_i) + \text{StdDev}(t_{i+1})}{2} \quad (4)$$

where $\text{Avg}(t_i)$ denotes the average compass readings over a $t_i$ time period, $\text{StdDev}(t_i)$ denotes the standard deviation of compass reading during $t_i$, and $G$ denotes a guard factor.

Next, we compute the distance traveled by a user as a product of the number of steps taken and step stride length. This is because, according to [67], the technique of double-integrating acceleration readings could induce a large error even over a small distance. Our algorithm continuously reads accelerometer data to calculate user step count while it filters out the noise. Then, it infers an increment in step count based on changes in the observed readings.

We converted a sample floor-plan into an $N \times N$ matrix, as shown in Fig. 4. Here, each coordinate point is 2 m away from its neighbors. This map comprises just the symbols $\{1, 0, W\}$, where "1" indicates accessible points, "0" indicates inaccessible points, and "$W$" represents walls. After the initial Wi-Fi scan, all possible 1 s on this grid are listed. Every time a distance equal to the distance between two coordinates on the grid (2 m) is traveled by the user, the path tracking module records the user's direction of movement. Based on the observed movements, each coordinate in the list is updated, and coordinates that fall on a 0 or $W$ are eliminated. This process of elimination continues iteratively as the user moves while changing direction, and eventually concretizes to a single point on the grid that represents the user's true position. Once one user locates himself/herself correctly, he/she can broadcast his/her position to other team members so that they may eliminate their invalid coordinates faster.

*1) Accurate Estimation of Distance Walked:* Experimentally, we observed that walking speed also plays a crucial role

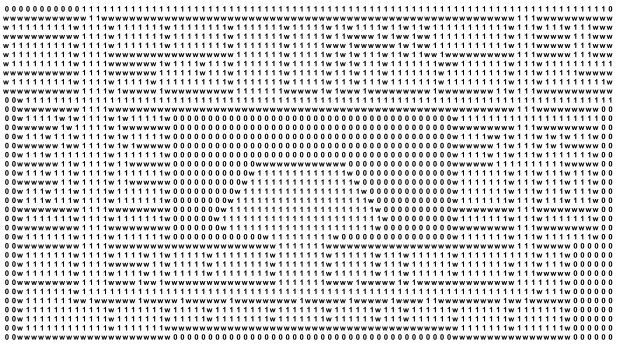

Fig. 4. Computer science department building map: 1 (accessible), 0 (inaccessible), and *W* (wall) [53].

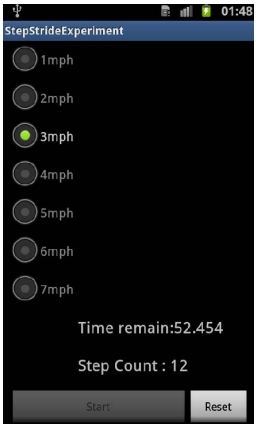


Fig. 5. Step profiling App: generating a user's stride length at 3 mph.

in the calculation of stride length. Therefore, we incorporated walking speed and the corresponding stride length into our system. Our profiling approach consists of two modes: 1) profile creation and 2) profile usage.

Each user trains the system for his specific stride lengths based on his/her different walking speeds in the profile creation mode. The graphical user interface for this is shown in Fig. 5. It shows different speed options of 1 mph, 2 mph, and so on up to 7 mph. It also shows the corresponding lap time. For our tests, we generated four separate profiles for users of different genders and heights. The test participants walked a test area of length 100 m to train the system for different speeds, i.e., step frequencies (by time-limiting the distance to be covered) and corresponding stride lengths.

Fig. 6 depicts our experiments results. A user's stride length increases, independently of height and gender, as the walking speed increases. The stride length is also closely related to the user's walking style. For instance, a female of height 5.83 ft walking at a speed of 5 mph has a similar stride length as that of a male of height 5.90 ft. This example emphasizes why it is required to use step profiling for each user in order to obtain a more accurate traveled distance. We note that, according to the human kinematics literature [68], [69], it has been shown that human walking patterns are unique to individuals and do not vary significantly over time. Further, walking patterns can be well modeled based on walking speed and stride length, which provides a theoretical basis for this paper.

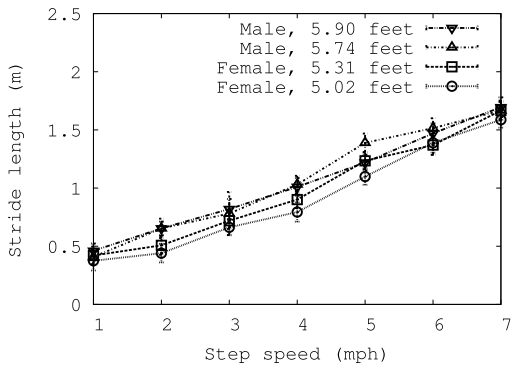


Fig. 6. Stride length as a function of walking speed (for different genders and heights) [53].

TABLE I
AVERAGE STRIDE LENGTH AND STEP-PROFILING RESULTS FOR A MALE (5.83 ft) AT THREE DIFFERENT STEP SPEEDS

| Speed | | Mod. | Fast | Run |
|---|---|---|---|---|
| Distance (m) | | 100 | 100 | 100 |
| Lap Time (s) | | 74.5 | 44.7 | 31.9 |
| Observed # of Steps | | 127 | 77 | 68 |
| # of Steps (Ground Truth) | | 127 | 84 | 71 |
| Male Stat. | Step Len. (m) | 0.737 | 0.737 | 0.737 |
| | Dist Err. | 6.4 | 42.04 | 49.884 |
| Male Profile | Step Len. (m) | 0.79 | 1.28 | 1.42 |
| | Dist Err. | 0.33 | 1.44 | 3.44 |

TABLE II
AVERAGE STRIDE LENGTH AND STEP-PROFILING RESULTS FOR FEMALE (5.5 ft) AT THREE DIFFERENT STEP SPEEDS

| Speed | | Mod. | Fast | Run |
|---|---|---|---|---|
| Distance (m) | | 100 | 100 | 100 |
| Lap Time (s) | | 74.5 | 44.7 | 31.9 |
| Observed # of Steps | | 121 | 77 | 75 |
| # of Steps (Ground Truth) | | 125 | 86 | 77 |
| Female Stat. | Step Len. (m) | 0.673 | 0.673 | 0.673 |
| | Dist Err. | 18.93 | 48.179 | 49.525 |
| Female Profile | Step Len. (m) | 0.82 | 1.32 | 1.42 |
| | Dist Err. | 0.78 | 1.64 | 5.08 |

In the profile usage mode, the system uses the existing profile reference data to dynamically obtain a user's stride length from his/her current walking speed. Tables I and II compare the error percentage (relative deviation from the ground truth) in distance traveled by a male user and female user, respectively, for average (statistical) stride length and profiling stride length. According to [70] based on the average stride length, the stride for a male of height 5.83 ft is a single fixed value (0.737 m) for all walking speeds. Similarly, for a female of height 5.5 ft, it is 0.637 m. Because stride length naturally becomes larger with walking speed, stride analysis becomes more inaccurate at higher speeds, causing the error percentage increase. However, profiling the stride length is sufficient here. For example, for the stride length of a male of height 5.83 ft was observed at moderate, fast, and running speeds to be 0.79, 1.39, and 1.67 m, respectively. With profiling information, the difference between the ground truth and calculated distance (as $stridelength \times stepcount$) dropped significantly. This improved the system performance tremendously. A dynamic system is

more suitable for our target scenario, where a firefighter could be running and walking intermittently.

To summarize, our profiling method consists of two modes, namely, profile creation mode and usage mode. In the profile creation mode, a user walks a test road for a given walking speed (i.e., step frequency), and the average stride length is then estimated by dividing the road length by the step count. A user's reference database is built by collecting stride length data for different speeds. In the profile usage mode, the system dynamically obtains a user's stride length from the reference database by simply determining the user's current walking speed. Because walking patterns are unique to individuals and do not vary significantly over time [68], [69], this manual calibration can be performed once when a user initializes the system's mobile client. Note that it is possible to automate this calibration process, and thus, the mobile system can periodically trigger calibration. For example, when a user walks outside, it can record GPS tracks along with the walking data. We then can find a set of straight line segments from the GPS traces. The walking data collected while a user is traveling along the straight line segments can then be used to estimate stride length as we did in manual profiling.

## V. Evaluation

We evaluated CLIPS from an Android-based test bed to experiment our system's overall performance in real-world scenarios, which is presented in Section V-A. To verify the general system performance of CLIPS in diverse locations (with floor-plans obtained from UCLA, Osaka University, and KAIST), we further performed extensive simulations and present the results in Section V-B. Both experimental and simulation results demonstrate that CLIPS lead to an accurate position significantly faster than existing noncollaborative schemes.

### A. Test Bed Experiments

*1) Experiment Setup:* We conducted our field tests in the computer science department building at UCLA. As shown in Fig. 7, we first generated a preprocessed floor-plan with a grid overlay. We used stride length profiles for accurate dead reckoning with varying numbers of team members, ranging from two to nine. For path tracking (Fig. 4), the same floor-plan was used. We deliberately chose three representative routes to further evaluate how the nature of a route traveled by a user (i.e., the number of corner turns and the length of a straight line taken by a user) can influence the system performance; the performance results with random paths are reported in the simulation section. As shown in Fig. 7. *Route* 1–3 start from the same starting point, but while *Route* 1 contains a long straight path with just a single turn, *Route* 2 comprises of more turns with long enough straight paths between two consecutive turns. *Route* 3 includes many turns with very short straight paths after each turn.

*2) Experimental Results:* We evaluate CLIPS's performance with the followings: 1) the impact of the team size and the slack value; 2) convergence speed variation when taking different routes; 3) convergence accuracy with and without stride length profiles; and 4) overall convergence delay between CLIPS and smartphone dead reckoning.

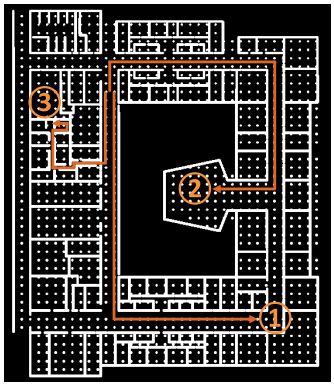


Fig. 7. Preprocessed floorplan and routes 1–3 [53].

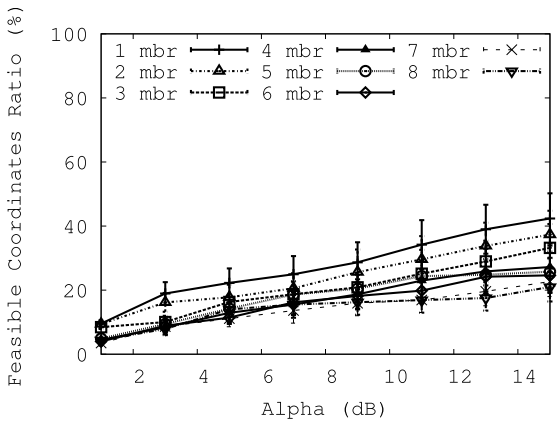


Fig. 8. Matching outcome ratio for different numbers of team members [53].

*a) Impact of team size and slack value:* Fig. 8 shows the feasible coordinate ratio (FCR) for different numbers of peers. We also varied the slack value $\alpha$ to determine its optimum value. Note that we introduced $\alpha$ to cope with the difference between the floor-plan and reality, radio signal fluctuation over time, and Wi-Fi chipset disparity. Its value should be small enough to eliminate most infeasible points, but large enough to not miss the true location. The FCRs are calculated by dividing the number of outcomes by all the coordinates in the floor-plan ($48 \times 48$ coordinates). While the $\alpha$ values showed a positive relationship with the FCR, we observed that the number of false positives decreases as we increase the number of peers. Fig. 9 illustrates the *hit ratio*, which is the probability that the matching outcomes contain the true positive position. As Fig. 8 shows, we varied the number of team members as well as $\alpha$ to observe the changing system behavior in different cases. The probability that a current point is scanned becomes 100% as the $\alpha$ has been increased. Another interesting observation is that a smaller team (with few members) requires a smaller $\alpha$ value to contain current position with the initial scan. For instance, a one-member team with an $\alpha$ value of 9 dB was observed to scan the current position with 100% probability, whereas an 8-member team requires an $\alpha$ value of 15 dB to obtain the same probability.

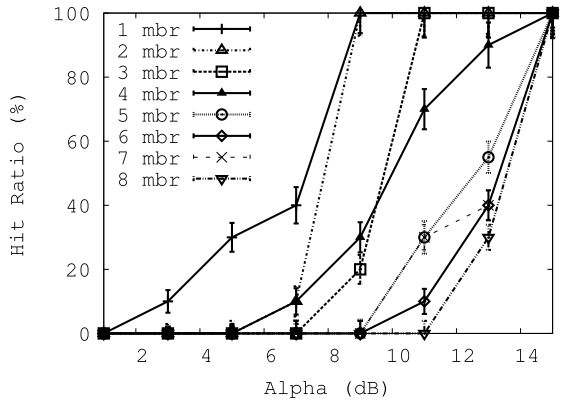


Fig. 9. Probability of obtaining the current point for different team sizes as a function of slack variable $\alpha$ [53].

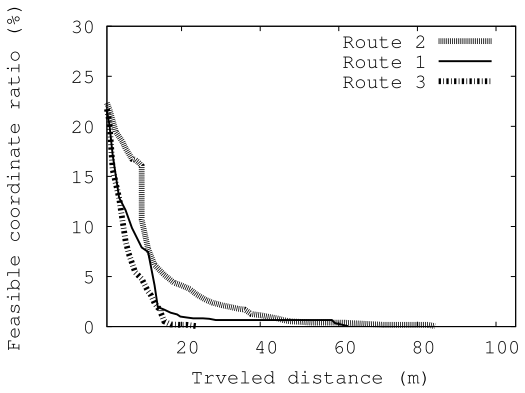


Fig. 10. Convergence speed of different routes [53].

*b) Convergence speed for different routes:* Of all the system performance metrics, accurate convergence speed to a unique point is one of the most important for the proposed scheme. We investigated how this factor is affected by different routes and by the utilization of peer-to-peer RSS exchanges and matched it with simulation data. To verify this, we carefully chose three representative routes: *Route* 1 (long straight paths and a turn); *Route* 2 (moderately long straight paths and a few turns); and *Route* 3 (short straight paths and many turns) to see which of the two factors, long straight paths or the number of turns, affects the convergence speed more. We measured 20 trials on each route and evaluated the success ratio of two different stride approaches. Fig. 10 shows the convergence to the unique point as a function of the distance traveled. *Route* 1 shows a rapid drop in FCR in the beginning because of its distinct route feature, but after traveling 20 m, the route showed relatively slow convergence. This was because multiple points on the straight path still remain as candidates. However, when a turn is taken after 60 m, it converges to a unique point. *Route* 2 shows the slowest convergence speed. This is because this route contains relatively moderate events, such as moderately long straight paths and turns. *Route* 3 shows the fastest convergence speed to a unique point. This is because this route contains many turns. Thus, based on our observation, a complicated route tends to expedite the process of convergence to a unique point.

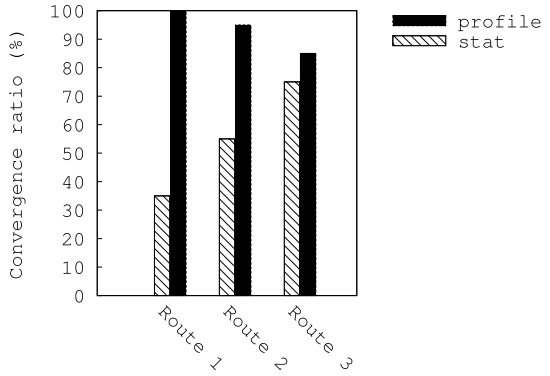


Fig. 11. Convergence success ratio of two stride length estimation methods for three different routes [53].

*c) Convergence accuracy with stride length profiling:* Accurate dead reckoning with stride length profiling is CLIPS's one of import features. In order to evaluate this feature, two different stride length approaches, namely, average stride length (fixed) and stride length profiling (variable), have been tested 20 times per route. Given that the users are navigating indoors, it is highly necessary to have accurate displacement logging, as in the map shown in Fig. 4. Our step profiling uses dynamic stride length to calculate the traveled distance unlike the average stride length approach. Eliminating false positives on the floor-plan using this scheme is more accurate. For instance, at the end of a corridor, changes in direction that are both earlier and later than required will cause the convergence to a unique position to fail. Fig. 11 shows the convergence ratio of the two different mechanisms (fixed versus variable), which are well matched with distance deviations for both the male and female users shown in Tables I and II, respectively. Step profiling provides 100% accuracy for *Route* 1 while slightly degraded performance for *Route* 3 is observed. This can be attributed to the error in compass readings caused by the magnetic fields in the surroundings. Even when profiling correctly calculates the distance traveled, the error in orientation can cause inaccurate results. To eliminate these errors, a gyroscope and an accelerometer can be used in addition to the orientation sensor, as explained in [48]. We will incorporate this feature in the future. In our experiments, the average stride length approach significantly underestimated the actual distance traveled in a long, straight corridor, resulting in poor performance (32%). However, for a path such as that of *Route* 3, the average stride length showed relatively reliable performance. This could be because the overall distance error is refreshed after each turn on this route.

*d) Overall convergence delay of CLIPS:* The aggregated time taken to lead to a unique point with step profiling for each of the three routes is shown in Fig. 12. The delay difference mainly occurs because users are traveling different routes while the initial Wi-Fi scanning and matching took almost constant time for all three scenarios. Overall delay is dominated by the traveling time while the Wi-Fi scanning and matching delay are considerably smaller comparatively.

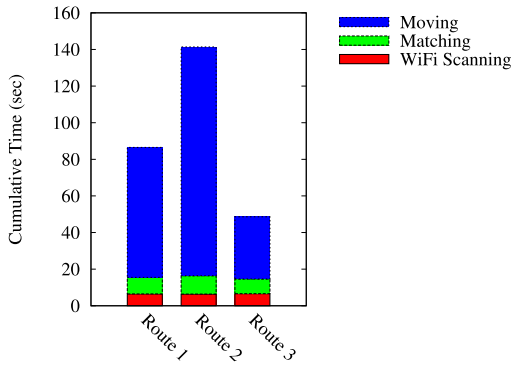


Fig. 12. Cumulative latencies of three modules for different routes [53].

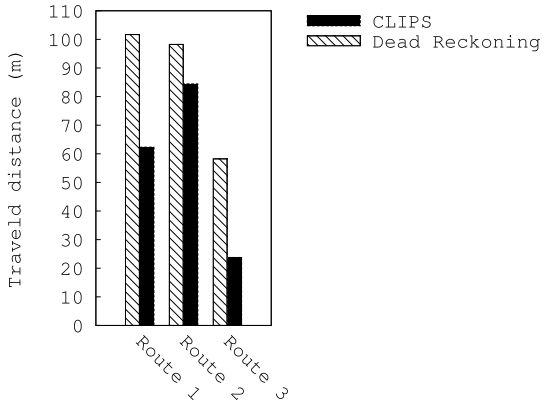


Fig. 13. Distance that must be traveled to converge to a unique point on three different routes. CLIPS versus smartphone dead reckoning [53].

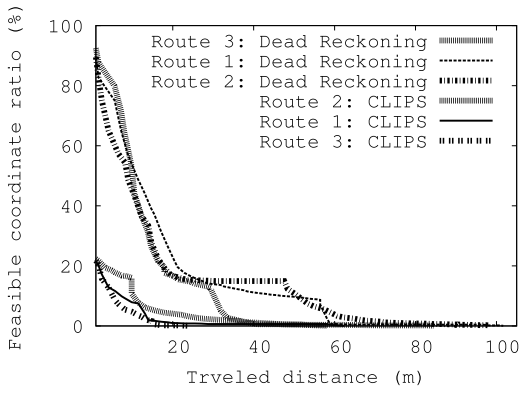


Fig. 14. Convergence speed. CLIPS versus smartphone dead reckoning.

*e) Smartphone dead reckoning versus CLIPS:* To evaluate how RSS matching can affect the overall system performance, we compared dead reckoning using COTS smartphone sensors with CLIPS. Fig. 13 shows the exact distances that had to be traveled by each system to converge to a unique location on each of the three routes. We observed that for all three routes, smartphone dead reckoning required the user to travel a longer distance in order to converge to a unique point. Fig. 14 shows that more detailed, feasible coordinates are lost as a function of the distance traveled. We note that smartphone dead reckoning shows that *Route* 2 requires a longer

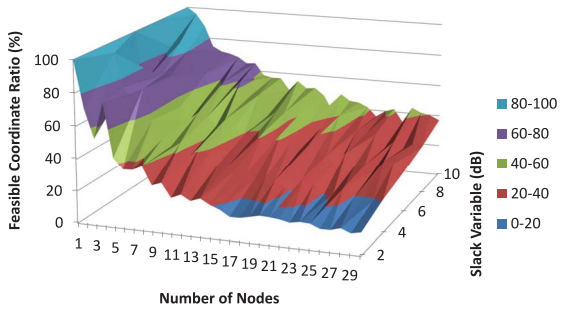


Fig. 15. Effect of slack variable $\alpha$ on feasible coordinates.

distance to converge to a unique point, which consequently takes the longest time compared to the other two routes. This is because smartphone dead reckoning shows more dependency on routes, as one route contains more turns to expedite convergence process to the unique point. However, CLIPS starts from a different set of feasible points; therefore, it is less dependent on turns. Considering the results of Figs. 12 and 13, we conclude that Wi-Fi scanning and matching improve the system performance with relatively little cost.

## B. Simulations

*1) Simulation Setup:* To test the general system performance at different locations, we further evaluated the two major components of CLIPS, namely, matching RSS with path loss estimation and dead reckoning via network simulations using Qualnet. For the simulations, floor-plans were obtained from UCLA, Osaka University, and KAIST. To generate path loss estimation on a floor-plan, the wireless signal propagation was computed by Wireless Insight with a 3-D model of the floor-plan (by CoCreate modeling Software [71]). We used $N \times M$ grid topologies, in which the side length of each grid is 2 m. The measured RSS was randomly generated from the simulated RSS with Gaussian error $N(0$ dB, 5 dB) (mean 0 and variance $=$ 5 dB). To verify the sensitivity of the matching between simulated and measured RSS, we set the default value of $\alpha$ to 10 dB; we also assumed that dead reckoning is accurate. In our simulations, we adopted random room search mobility (RRSM) to emulate a team member's mobility. Initially, each node is located at a randomly chosen position. Each node chooses a destination room randomly and moves to the destination through a passage. When it reaches the destined room, it again randomly picks and moves to the next destination room. The nodal speed was set to 2.5 m/s. Unless otherwise specified, we report an average value of 50 runs with RRSM to mitigate any dependence on specific route paths and to understand general system behavior.

## C. Simulation Results

*1) Effect of Slack Variable $\alpha$ on Feasible Coordinate and Hit Ratios:* Figs. 15 and 16 examine the effects of the slack variable $\alpha$ on the FCR and the hit ratio, respectively. To verify these, we varied the number of deployed nodes from 1 to 30, and changed the slack variable from 2 to 10. Fig. 15 shows

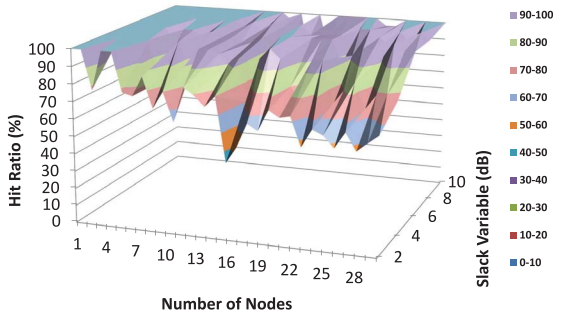


Fig. 16. Effect of slack variable $\alpha$ on hit ratio.

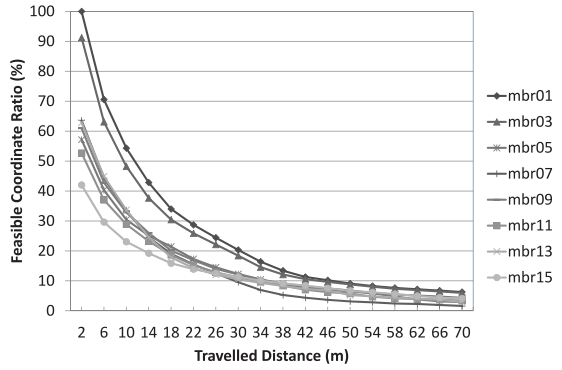


Fig. 17. Effect of team size on convergence speed.

TABLE III
POSITIONING ERRORS ($m$)

| Traveled Distance (m) | | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 |
|---|---|---|---|---|---|---|---|---|---|
| w/ RSS | Average | 19.85 | 17.01 | 14.31 | 10.55 | 6.82 | 4.14 | 3.31 | 2.21 |
| | Std Dev. | 6.04 | 6.61 | 6.54 | 5.73 | 4.32 | 2.54 | 1.85 | 1.40 |
| w/o RSS | Average | 40.98 | 36.42 | 28.36 | 24.95 | 20.84 | 12.35 | 8.70 | 4.55 |
| | Std Dev. | 1.05 | 1.09 | 1.29 | 2.22 | 1.70 | 3.07 | 3.28 | 1.24 |

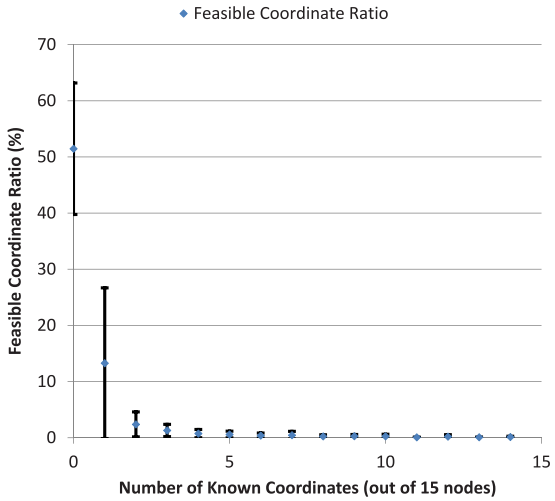


Fig. 18. Effect of known coordinate landmarks on convergence speed and accuracy.

that all coordinates are considered as feasible in the single node case. This is because the node can be deployed in any place on the passage and cannot verify feasibility with itself. As we increased the number of deployed nodes, the FCR decreased; the greater number of nodes deployed, the smaller the set of feasible matches on the map. Note that the number of coordinates increases as we increase $\alpha$. Fig. 16 also examines the effect of $\alpha$ on the hit ratio. This figure represents the performance accuracy of CLIPS based on the slack variable, coupled with Fig. 15. The greater the number of coordinates considered feasible thanks to the larger $\alpha$ value, the higher the hit ratio. By performing this simulation, we confirmed that the relationship between the number of deployed nodes and FCR as well as between the slack variable and hit ratio are well matched with the test bed results.

*2) Effect of Team Size on Convergence Speed:* Fig. 17 reports how different numbers of team members can affect the convergence speed. All eight cases, from 1- to 15-node teams, show correlated behavior: FCR decreases as the traveled distance increases. We also note that teams with a large number of members start with a smaller FCR, which means that a large team will require less traveled distance (and time) to converge to unique coordinates.

*3) Positioning Errors:* Table III shows the average positioning errors and their standard deviations. Positioning errors are calculated as the average distance error from the feasible coordinates to true coordinates. CLIPS (with RSS) has a clear advantage over the scheme without RSS in terms of the average errors, but it has larger standard deviations. However, these deviations converge and become smaller after at least 50 m has been traveled. The positioning error of CLIPS is finally reduced to 2.2 m with $\sigma = 1.4$ m, which is reasonably small.

*4) Sharing Location Updates:* After a user has determined the current location, periodically sharing this information with fellow team members (via ad hoc communications or a centralized server) helps the team to eliminate false coordinates. To verify the improvement in performance caused by this location sharing, we evaluated the FCR based on the number of determined locations. In Fig. 18, we present the feasible coordinates of users as a function of the number of users with known coordinates (acting as mobile anchors). The graph shows that the feasible coordinates exponentially decrease with the number of users with known coordinates. Even for a single node, the FCR is reduced from 51% to 13%, and with an additional node, it is further reduced to 2.3%. This result shows how much the overall system performance can be improved by sharing determined locations.

## VI. DISCUSSION AND FUTURE WORK

### A. Aggressive Information Sharing

In our simulation results, we show that when nodes share discovered location information (act as mobile anchors), this can significantly reduce the size of feasible coordinates. Another way of reducing the size of feasible coordinates is to share the current progress of localization with other members (i.e., sharing the feasible coordinates of a node with the rest of the nodes). If node $A$ receives the feasible coordinate of another node $B$, node $A$ can further eliminate the infeasible coordinates of node $B$ in its feasible coordinates. We leave this simple improvement as part of our future work.

### B. Robust Dissemination

For beaconing and location sharing, we use best effort delivery with user datagram protocol broadcasting. Wireless errors can be effectively handled with periodic packet broadcasting. In some cases, location updates might need to be delivered over multiple hop. By piggybacking discovered location information in a packet, the location information of group members can be eventually disseminated to the entire group.

### C. Exploiting Environmental Signatures

Mobile devices with their sensor measurements are dead reckoned while, in order to detect unique environmental signatures within the building (e.g., Wi-Fi RSS drops and magnetic fields), the same measurements are leveraged. Similarly to the recent work by Wang *et al.* [19], these unique signatures can be used to fix errors caused by the dead reckoning, which in turn improves the location accuracy of these signatures. However, surveying such environmental signatures takes considerable time. One possible solution is to use an ad hoc deployment of mobile beacons during the emergency operations to create "ad hoc environmental signatures," which is an idea inspired by iBeacon scenarios. When a team of first responders arrives at the floor of interest, they could scatter mobile beacons throughout the site. Then, CLIPS can leverage both mobile and stationary beacons. As shown in our simulation results, we can clearly see that as the number of beacons increases, we can significantly reduce the feasible coordinates, and sharing update information will further improve the convergence speed. We leave the examination this hybrid scenario as a part of our future work.

### D. Path-Loss Simulation of Random Coordinates

A random instead of uniform distribution of coordinates could be beneficial to remove flip and rotation issues in matching. The density of such points is also an important parameter for understanding the tradeoff between accuracy and computation complexity. It might be possible to aggregate two points that have similar propagation characteristics (i.e., points in the same small room) to reduce complexity and expedite convergence to the unique point. To do this automatically, we need to verify whether two nearby points match after mapping.

### E. Simultaneous Multifloor Operations

If there are other responders located on different floors, their Wi-Fi signals may interfere with those located on the current floor. In this scenario, we can use the following strategies. One approach is to remove nodes whose RSS values are lower than some threshold value. This is based on the fact that if mobile users are located on different floors, their RSS values could be significantly lower than those of the others. In addition, the mobility of users can be considered: i.e., it is highly unlikely that a user's RSS trace measured on a given floor matches with that on a different floor. Note that we can further extend our algorithm by simultaneously considering multiple floors and quickly determining which floor map is the best candidate. Another approach is to use an extra sensor, namely, a barometer (or altimeter), which is widely available in recent smartphones (e.g., Google Nexus 5). According to Muralidharan *et al.* [72], we can accurately detect whether users are located on the same floor by whether their atmospheric pressure values are within some threshold range (e.g., for every 10 m in height, there is a pressure difference of 1.2 hPa). Furthermore, when a user moves across floors, the change of pressure value is significant enough to detect this event.

## VII. Conclusion

In this paper, we consider time-critical indoor scenarios such as search and rescue missions in a large building, where traditional infrastructure-based localization schemes are not feasible. For this scenario, we developed CLIPS, a novel infrastructure-free, collaborative localization system. Unlike infrastructure-based approaches, CLIPS leverages peer-to-peer beaconing and floor maps, which are readily available for time-critical team operations. In CLIPS, mobile nodes initially estimate candidate coordinates in the floor maps by exchanging beacon signals and signal space maps. Mobility of users is then tracked with dead reckoning, and this information is used to validate whether a movement on the map is feasible when those candidate coordinates were considered as starting positions. This approach can easily remove all the invalid coordinates, thereby quickly leading to position fixes. We evaluated the proposed system in both Android-based testbed experiments and extensive simulations. Our results confirmed that CLIPS can provide accurate localization performance and achieve much lower delay for position fixes.

## References

[1] G. Borriello, A. L. Liu, T. Offer, C. Palistrant, and R. Sharp, "WALRUS: Wireless acoustic location with room-level resolution using ultrasound," in *Proc. MobiSys*, Seattle, WA, USA, 2005, pp. 191–203.

[2] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum," in *Proc. MobiSys*, Bethesda, MD, USA, 2011, pp. 155–168.

[3] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, no. 1, pp. 91–102, Jan. 1992.

[4] H. Shin, Y. Chon, K. Park, and H. Cha, "FindingMiMo: Tracing a missing mobile phone using daily observations," in *Proc. MobiSys*, Bethesda, MD, USA, 2011, pp. 29–42.

[5] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. MobiCom*, Chicago, IL, USA, 2010, pp. 173–184.

[6] *Skyhook, XPS, Web Site*. [Online]. Available: http://www.skyhookwireless.com

[7] *Apple, MobileMe, Web Site*. [Online]. Available: http://www.me.com

[8] *Google, Latitude, Web Site*. [Online]. Available: https://www.google.com/latitude

[9] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. INFOCOM*, Tel Aviv, Israel, 2000, pp. 775–784.

[10] A. LaMarca *et al.*, "Place lab: Device positioning using radio beacons in the wild," in *Proc. Pervasive*, Munich, Germany, 2005, pp. 116–133.

[11] J. R. Guerrieri *et al.*, "RFID-assisted indoor localization and communication for first responders," in *Proc. EuCAP*, Nice, France, 2006, pp. 1–6.

[12] G.-Y. Jin, X.-Y. Lu, and M.-S. Park, "An indoor localization mechanism using active RFID tag," in *Proc. SUTC*, Taichung, Taiwan, 2006, pp. 40–43.

[13] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," in *Proc. PerCom*, Fort Worth, TX, USA, 2003, pp. 407–415.

[14] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Proc. IROS*, Osaka, Japan, 1991, pp. 1442–1447.

[15] B. Ferris, D. Fox, and N. D. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proc. IJCAI*, Hyderabad, India, 2007, pp. 2480–2485.

[16] H. Liu *et al.*, "Push the limit of WiFi based localization for smartphones," in *Proc. Mobicom*, Istanbul, Turkey, 2012, pp. 305–316.

[17] Y. Kim, Y. Chon, and H. Cha, "Smartphone-based collaborative and autonomous radio fingerprinting," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 1, pp. 112–122, Jan. 2012.

[18] 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.

[19] H. Wang *et al.*, "No need to war-drive: Unsupervised indoor localization," in *Proc. MobiSys*, Ambleside, U.K., 2012, pp. 197–210.

[20] C. Luo, H. Hong, and M. C. Chan, "PiLoc: A self-calibrating participatory indoor localization system," in *Proc. IPSN*, Berlin, Germany, 2014, pp. 143–154.

[21] 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.

[22] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. MobiCom*, Boston, MA, USA, 2000, pp. 32–43.

[23] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," *IEEE Pers. Commun.*, vol. 4, no. 5, pp. 42–47, Oct. 1997.

[24] J. Xiong, K. Sundaresan, and K. Jamieson, "Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization," in *Proc. MobiCom*, Paris, France, 2015, pp. 537–549.

[25] M. Youssef and A. K. Agrawala, "The horus WLAN location determination system," in *Proc. MobiSys*, Seattle, WA, USA, 2005, pp. 205–218.

[26] A. Varshavsky, E. de Lara, J. Hightower, A. LaMarca, and V. Otsason, "GSM indoor localization," *Pervasive Mobile Comput.*, vol. 3, no. 6, pp. 698–720, Dec. 2007.

[27] L. Li *et al.*, "Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service," in *Proc. MobiCom*, Maui, HI, USA, 2014, pp. 459–470.

[28] *Koozyt, Inc., PlaceEngine, Web Site*. [Online]. Available: http://www.placeengine.com/showe/about

[29] M. Azizyan, I. Constandache, and R. R. Choudhury, "SurroundSense: Mobile phone localization via ambience fingerprinting," in *Proc. MobiCom*, Beijing, China, 2009, pp. 261–272.

[30] R. Gao *et al.*, "Sextant: Towards ubiquitous indoor localization service by photo-taking of the environment," *IEEE Trans. Mobile Comput.*, vol. 15, no. 2, pp. 460–474, Feb. 2016.

[31] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 13–32, 1st Quart., 2009.

[32] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[33] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo, "Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure," *Wireless Netw.*, vol. 16, no. 2, pp. 405–420, Feb. 2010.

[34] Y. Gwon and R. Jain, "Error characteristics and calibration-free techniques for wireless LAN-based location estimation," in *Proc. MobiWac*, Philadelphia, PA, USA, 2004, pp. 2–9.

[35] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, "ARIADNE: A dynamic indoor signal map construction and localization system," in *Proc. MobiSys*, Uppsala, Sweden, 2006, pp. 151–164.

[36] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo, "Zero-configuration, robust indoor localization: Theory and experimentation," in *Proc. INFOCOM*, Barcelona, Spain, 2006, pp. 1–12.

[37] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding WiFi localization," in *Proc. MobiSys*, Taipei, Taiwan, 2013, pp. 249–262.

[38] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "SAIL: Single access point-based indoor localization," in *Proc. MobiSys*, Bretton Woods, NH, USA, 2014, pp. 315–328.

[39] S. He, S.-H. G. Chan, L. Yu, and N. Liu, "Fusing noisy fingerprints with distance bounds for indoor localization," in *Proc. Infocom*, Hong Kong, 2015, pp. 2506–2514.

[40] C. Wu *et al.*, "Static power of mobile devices: Self-updating radio maps for wireless indoor localization," in *Proc. INFOCOM*, Hong Kong, 2015, pp. 2497–2505.

[41] L. E. Holmquist, J. Falk, and J. Wigström, "Supporting group collaboration with interpersonal awareness devices," *Pers. Technol.*, vol. 3, no. 1, pp. 13–21, 1999.

[42] J. Krumm and K. Hinckley, "The NearMe wireless proximity server," in *Proc. UbiComp*, Nottingham, U.K., 2004, pp. 283–300.

[43] *Bluehoo, Web Site*. [Online]. Available: http://bluehoo.com

[44] K. A. Li, T. Y. Sohn, S. Huang, and W. G. Griswold, "Peopletones: A system for the detection and notification of buddy proximity on mobile phones," in *Proc. MobiSys*, Breckenridge, CO, USA, 2008, pp. 160–173.

[45] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "BeepBeep: A high accuracy acoustic ranging system using COTS mobile devices," in *Proc. SenSys*, Sydney, NSW, Australia, 2007, pp. 1–14.

[46] N. Banerjee *et al.*, "Virtual compass: Relative positioning to sense mobile social interactions," in *Proc. Pervasive*, Helsinki, Finland, 2010, pp. 1–21.

[47] T. Higuchi, S. Fujii, H. Yamaguchi, and T. Higashino, "Mobile node localization focusing on stop-and-go behavior of indoor pedestrians," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1564–1578, Jul. 2014.

[48] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proc. UbiComp*, Seoul, South Korea, 2008, pp. 114–123.

[49] 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.

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

[51] S. Sorour, Y. Lostanlen, S. Valaee, and K. Majeed, "Joint indoor localization and radio map construction with limited deployment load," *IEEE Trans. Mobile Comput.*, vol. 14, no. 5, pp. 1031–1043, May 2015.

[52] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, "Precise indoor localization using smart phones," in *Proc. MM*, Florence, Italy, 2010, pp. 787–790.

[53] Y. Noh *et al.*, "Clips: Infrastructure-free collaborative indoor positioning scheme for time-critical team operations," in *Proc. PerCom*, San Diego, CA, USA, Mar. 2013, pp. 172–178.

[54] P. Dosch, K. Tombre, C. Ah-Soon, and G. Masini, "A complete system for the analysis of architectural drawings," *Int. J. Doc. Anal. Recognit.*, vol. 3, no. 2, pp. 102–116, 2000.

[55] *Google Maps Floor Plans, Web Site*. [Online]. Available: http://maps.google.com/help/maps/floorplans/

[56] H. Ye *et al.*, "FTrack: Infrastructure-free floor localization via mobile phone sensing," in *Proc. PerCom*, Lugano, Switzerland, 2012, pp. 2–10.

[57] C. R. Anderson and T. S. Rappaport, "In-building wideband partition loss measurements at 2.5 and 60 GHz," *IEEE Trans. Wireless Commun.*, vol. 3, no. 3, pp. 922–928, May 2004.

[58] H. Kim and H. Ling, "Electromagnetic scattering from an inhomogeneous object by ray tracing," *IEEE Trans. Antennas Propag.*, vol. 40, no. 5, pp. 517–525, May 1992.

[59] R. A. Valenzuela, "Ray tracing prediction of indoor radio propagation," in *Proc. PIMRC*, The Hague, The Netherlands, 1994, pp. 140–144.

[60] A. Falsafi, K. Pahlavan, and G. Yang, "Transmission techniques for radio LAN's-a comparative performance evaluation using ray tracing," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 3, pp. 477–491, Apr. 1996.

[61] M. Lott, "On the performance of an advanced 3D ray tracing method," in *Proc. EW*, Munich, Germany, 1999, pp. 113–118.

[62] K. A. Remley, H. R. Anderson, and A. Weisshaar, "Improving the accuracy of ray-tracing techniques for indoor propagation modeling," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2350–2358, Nov. 2000.

[63] I. Vlasenko, I. Nikolaidis, and E. Stroulia, "The smart-condo: Optimizing sensor placement for indoor localization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 3, pp. 436–453, Mar. 2015.

[64] *Insite, Web Site*. [Online]. Available: http://www.remcom.com/wireless-insite

[65] ITU-R, "Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 300 MHz to 100 GHz," Recommendation ITU-R P.1238-8, Geneva, Switzerland, 2015.

[66] T. H. Cormen, C. E. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: McGraw-Hill Higher Educ., 2001.

[67] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, "Did you see Bob?: Human localization using mobile phones," in *Proc. MobiCom*, Chicago, IL, USA, 2010, pp. 149–160.
[68] R. Margaria, *Biomechanics and Energetics of Muscular Exercise*. Oxford, U.K.: Clarendon Press, 1976.
[69] A. Zijlstra, E. D. de Bruin, N. Bruins, and W. Zijlstra, "The step length-frequency relationship in physically active community-dwelling older women," *Eur. J. Appl. Phys.*, vol. 104, no. 3, p. 427, 2008, doi: 10.1007/s00421-008-0795-6.
[70] *Step Size in Pedometers, Web Site*. [Online]. Available: http://walking.about.com/cs/pedometers/a/pedometerset.htm
[71] *Parametric Technology Corporation, Web Site*. [Online]. Available: http://www.ptc.com/products/creo-elements-direct/modeling-express/
[72] K. Muralidharan, A. J. Khan, A. Misra, R. K. Balan, and S. Agarwal, "Barometric phone sensors: More hype than hope!" in *Proc. HotMobile*, Santa Barbara, CA, USA, 2014, Art. no. 12.

**Youngtae Noh** (M'07) received the B.S. degree in computer science from Chosun University, Gwangju, South Korea, in 2005, the M.S. degree in information and communication from the Gwangju Institute of Science Technology, Gwangju, South Korea, in 2007, and the Ph.D. degree in computer science from the University of California at Los Angeles, Los Angeles, CA, USA, in 2012.

He is an Assistant Professor with the Department of Computer Science and Information Engineering, Inha University, Incheon, South Korea. He was a Staff Member with Cisco Systems, San Jose, CA, USA, and a Postdoctoral Research Associate with Purdue University, West Lafayette, IN, USA. His current research interests include data center networking, wireless networking, and mobile computing.

**Hirozumi Yamaguchi** (M'02) received the B.E., M.E., and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1994, 1996, and 1998, respectively.

He is currently an Associate Professor with Osaka University. His current research interests include design and development of mobile and pervasive systems.

**Uichin Lee** (M'06) received the B.S. degree in computer engineering from Chonbuk National University, Jeonju, South Korea, in 2001, the M.S. degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2003, and the Ph.D. degree in computer science from the University of California at Los Angeles, Los Angeles, CA, USA, in 2008.

He is an Associate Professor with the Department of Knowledge Service Engineering, KAIST. He was a Technical Staff Member with Bell Laboratories, Murray Hill, NJ, USA, and Alcatel-Lucent, Boulogne-Billancourt, France, until 2010. His current research interests include social computing systems and mobile/pervasive computing.