2013 IEEE International Conference on Pervasive Computing and Communications (PerCom), San Diego (18--22 March 2013)

CLIPS: Infrastructure-free Collaborative Indoor Positioning Scheme for Time-critical Team Operations

Youngtae Noh[†], Hirozumi Yamaguchi[§], Uichin Lee[‡], Prerna Vij, Joshua Joy, Mario Gerla
[†]Cisco Systems Inc. [§]Osaka University [‡]KAIST UCLA
[†]ynoh@cisco.com, [§]h-yamagu@ist.osaka-u.ac.jp, [‡]uclee@kaist.ac.kr, {pvij, jjoy, gerla}@cs.ucla.edu

***Abstract*—Indoor localization has attracted much attention recently due to its potential for realizing indoor location-aware application services. This paper considers a time-critical scenario with a team of soldiers or first responders conducting emergency mission operations in a large building in which infrastructure-based localization is not feasible (e.g., due to management/installation costs, power outage, terrorist attacks). To this end, we design and implement a collaborative indoor positioning scheme (CLIPS) that requires no preexisting indoor infrastructure. We assume that each user has a received signal strength map for the area in reference. This is used by the application to compare and select a set of feasible positions, when the device receives actual signal strength values at run time. Then, dead reckoning is performed to remove invalid candidate coordinates eventually leaving only the correct one which can be shared amongst the team. Our evaluation results from an Android-based testbed show that CLIPS converges to an accurate set of coordinates much faster than existing non-collaborative schemes (more than 50% improvement under the considered scenarios).**

I. Introduction

Location-based services have received a lot of attention in recent years as they can deliver customized services based on people's locations. Outdoor services can be efficiently delivered with standard localization techniques supported in off-the-shelf mobile devices (e.g., GPS, cell-tower localization). In contrast, indoor location services generally require some form of infrastructure, e.g., Wi-Fi access points [25], [10], [5], [1], [2], [8], acoustic beacons [9], [26], [28], and RFID tags [17], [30], [8], [24] with quite a bit of customization.

Our work is motivated by a time-critical indoor scenario with a team of soldiers or first responders conducting emergency mission operations (e.g., firefighting, rescue, or urban military operations). Fast and accurate localization would help team members navigate an area of interest and sharing situational-awareness would facilitate in successful mission operations. Infrastructure-based localization, however, is often not the best solution in emergency scenarios for several reasons. The team may not have enough time to (install and) configure localization infrastructure such as Wi-Fi and acoustic beacons. In reality, maintaining such infrastructure for the mission operations in every building a priori may be neither practical nor economically feasible. In some emergency cases, infrastructure may not be even available to begin with due to a power outage or terrorist attacks.

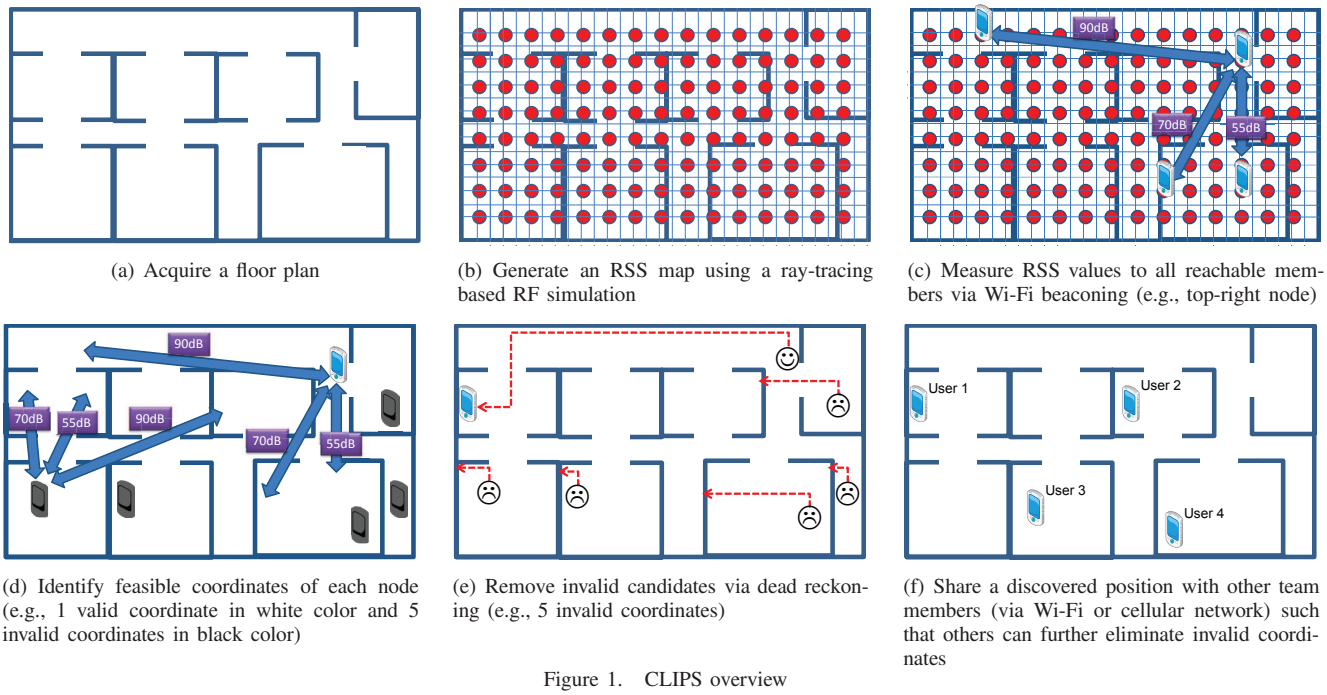
This work was supported in part by the KDDI foundation and the CPS-IIP Project (FY2012 - FY2016) in the research promotion program for national level challenges by the Ministry of Education, Culture, Sports, Science and Technology (MEXT). This work was supported in part by the National Research Foundation of Korea Grant funded by the Korean Government (MEST) (NRF-2012R1A1A1008858). This work was supported in part by the Health Guardian- A Gateway to Networked Wellness (NSF-0917408) and closing the loop between traffic/pollution sensing and vehicle route control using traffic lights and navigators (NSF-1111971).

In such a scenario, it is imperative to use infrastructure-free localization. The most common solution is inertial navigation (also known as dead reckoning) that tracks the current position of a user by constantly monitoring heading changes and traveled distance with inertial sensors (e.g., gyroscope and accelerometer) [29]. However, inertial navigation often suffers from slow convergence and large errors due to relative lack of indoor position landmarks, with detrimental effect on time-critical indoor missions. It is possible that the rate of increment in position error is higher than the rate at which it is fixed with a floor map (e.g., map matching with turn detection) [20], [15], [11]. This problem will be even more pronounced when a team explores a large and complex building (e.g., entering an underground parking lot and then moving to a locus of events at the fifth floor). For timely localization, it is well known that we need to employ additional mechanisms (e.g., manual inputs, infrastructure support); unfortunately, existing solutions are less appealing to the emergency scenario under consideration.

To mitigate this problem, we propose a collaborative indoor positioning scheme (CLIPS) that leverages peer-to-peer Wi-Fi beaconing and accurate dead reckoning. In our scheme, mobile users measure the received signal strength (RSS) values from other peers. Given that for safety operations a response team can access a floor plan, we propose to build a realistic signal strength estimation map a priori (e.g., using a wireless signal propagation simulator such as ray tracing). This map allows the application to search for a set of possible coordinates in which the calculated RSS values match the observed RSS values. To handle potential wireless signal fluctuations, we relax that a match can happen when the difference of RSS values is within the threshold value, a tunable system parameter (called a slack variable). Although this process will initially produce a large number of false candidate positions, they can be efficiently eliminated with dead reckoning; i.e., as a user moves along the corridors, invalid candidates will quickly lead to dead-ends. Every peer will repeat this procedure, and thus, each team member can quickly locate his position; in general, the larger the number of members, the faster the convergence time. Whenever all members have acquired their position fixes, Wi-Fi beaconing can be suspended to conserve battery; dead reckoning continues to track each user's current position. In the subsequent processes, we can drastically reduce the search space (and convergence time) as it is sufficient to consider the close-by locations to a node's current position (e.g., with fixed radius).

The main contribution of this paper is to show the feasibility of infrastructure-free indoor localization by lever-

978-1-4673-4575-0/13/$31.00 ©2013 IEEE



(a) Acquire a floor plan

(b) Generate an RSS map using a ray-tracing based RF simulation

(c) Measure RSS values to all reachable members via Wi-Fi beaconing (e.g., top-right node)

(d) Identify feasible coordinates of each node (e.g., 1 valid coordinate in white color and 5 invalid coordinates in black color)

(e) Remove invalid candidates via dead reckoning (e.g., 5 invalid coordinates)

(f) Share a discovered position with other team members (via Wi-Fi or cellular network) such that others can further eliminate invalid coordinates

Figure 1. CLIPS overview

aging collaborative Wi-Fi beaconing and dead reckoning. Our field experiment results with Android smartphones confirm that CLIPS provides accurate position fixes much faster than non-collaborative schemes; i.e., travel distance to acquire an accurate fix was less than half of that in a non-collaborative scheme. In the following sections, we provide a brief overview of CLIPS (Section II) and detail the core components (Section III). Then, we report the test-bed experiment results in Section IV. Finally, we conclude the paper in Section VI.

II. CLIPS Overview

Our objective is to help a group of rescuers localize themselves on an unfamiliar floor-plan without any infrastructure support. We assume that the team has acquired a floor plan a priori (say as in Figure 1(a)). The floor-plan is processed to form an overlay of an $N \times N$ grid as shown in Figure 1(b). The grid dimensions are determined by the granularity of location information required in the applications for missions operations; we use a 2m $\times$ 2m grid in our experiments. Also, for every coordinate on the floor, we compute Received Signal Strength (RSS) values from every other coordinate on the floor, using a ray-tracing based RF propagation simulation and generate an RSS map of N $\times$ (N-1) dimension. We assume that the rescue team members have homogeneous mobile devices and download the floor plan and this estimated RSS map a priori.

Figure 1(c) depicts what happens once the team members enter the target building/floor. Each mobile node performs periodic Wi-Fi beaconing through which RSS values of reachable team members can be observed. These RSS values are then matched against the RSS map downloaded a priori, and this will generate a list of all feasible coordinates on the floor. We will present the detailed matching algorithm in Section III-B. After finding these feasible coordinates, we perform dead reckoning to eliminate all the false positives from the list. By tracking heading changes and distance walked over a map, we update every feasible coordinate. However, if it leads to a dead end, we eliminate the coordinate. A simple example is given in Figure 1(e). We repeat this process until there is only one coordinate remaining. Once a user localizes himself correctly, he can broadcast his position on the map to the rest of the nodes (as shown in Figure 1(f)) such that they can further eliminate invalid coordinates in their list. This also allows each team member to know the position taken by every other member.

III. CLIPS System Details

We illustrate the core components of CLIPS, namely (1) floor plan pre-processing and RSS map generation, (2) feasible coordinate estimation and location convergence, and (3) accurate map matching with dead-reckoning.

A. RSS Map Generation

Our positioning scheme requires a floor map for each site. Such a floor map can be obtained from floor plan information or Geographic information system (GIS) data. We assume that the emergency team can access a digital map from a service provider who recognizes a given floor plan image and builds a digital floor map—architectural floor plan image recognition has been extensively studied in the field of GIS [13]. Recently, similar services have been launched such as "Google Maps Floor Plans" for WLAN indoor positioning that accept users' uploaded floorplan images [3].

Next, an RF propagation model (e.g., the log-distance path loss model) can be used to predict RSS values for each coordinate on the floor plan and get the pathloss simulation data. Using this model reduces the number of RSS measurements significantly compared to RF fingerprinting schemes, albeit at the expense of degraded localization accuracy. Since RF propagation characteristics vary widely (especially indoor), the model parameters would have to be estimated

specifically for each indoor space of interest. For example, Anderson et al. presented measured data for 2.5GHz in-building partition loss [7].

Ray-tracing has also been widely used, and some tracing models consider the details of a place, such as walls, windows, doors as well as desks, chairs and the thickness of walls to pursue the best accuracy. It approximates the radio propagation with a finite number of isotropic rays emitted from a transmitting antenna by a ray imaging technique. In this technique, the transmitter is assumed to be reflected at each surface around it to produce image transmitters, the reflected rays to the receiver from the real transmitter are considered as 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. The major drawback of such techniques is, however, expensive computation complexity. Readers may refer to detailed ray-tracing technique in [19], [27], [14], [22], [18]

All models could be used in our simulation. However, in order to pursue reasonable balance between effort of floor map modeling, simulation time and accuracy, we take the following modeling and simulation: we consider all the solid lines in a floorplan as walls where windows are regarded as walls and doors are not drawn (we assume they are always open since mobile nodes walk through), and all the spaces surrounded by the solid lines as accessible spaces. With this indoor modeling, we use a simplified (reduced) 3-ray tracing in the experiments (provided by Wireless InSite software [4]) where diffraction along obstacles are considered. In practice, however, the measured signal strengths tend to fluctuate due to small scale fading. While this phenomenon can be mitigated by conducting multiple measurements to find the average RSS values, a node may still suffer from temporal fluctuation (thereby having chances of excluding the true position). To solve this problem, we introduce a slack parameter α (usually $\pm$13dB in our setting). We will further justify this approach with measurement results in Section IV.

B. Feasible Coordinate Estimation

From each beacon log $(i, j, ss)^t$ of node j where ss is a received signal strength from node i at time t, node j estimates the path loss value in this communication (denoted as $m(i, j)$). For this purpose, we assume that transmission power and other factors such as sender antenna gain/loss and environmental noise are almost common and constant on all mobile nodes. We note that these values may be hardware-dependent, but pre-measurement before localization (before mission starts) is effective to know such values.

1) Positioning problem formulation: We focus on the positioning activity of node j. Given a floor plan of *fp* with *N* grids, let $L : N^2 \rightarrow \mathcal{R}^+$ denote a path loss matrix among N points where element $(u, v) \in N^2$ is the simulated path loss from points u to v. Also, let M_j denote the set of node j and the nodes that node j has a beacon log at time t. Let $m : M_j \times \{j\} \rightarrow \mathcal{R}^+$ denote the set of path loss measurements of node j at time t (*i.e.* $m(i, j)$ is the measured path loss value from nodes i to j).

The positioning problem of node j is to find node j's position from the N points with the least distance from the true position. Our approach is to find such a position by using the positioning function $p : M_j \rightarrow N$ with the least "path loss matching error" between m and L. The objective function is defined to minimize such pathloss matching error as follows.

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

We note that if path loss increases as the square of distance (like free-space attenuation), we should find such p that has the least square deviations. Nevertheless, we adopt this simple sum of deviations since such path loss characteristics are highly situation-dependent, and it is therefore hard to accurately determine such an attenuation coefficient.

As we addressed earlier, we need to accommodate measurement errors caused by fluctuation of RSS values. We should allow some deviation parameter α where a measured path loss l' and a simulated path loss l are regarded identical *iff* $l' \in [l-\alpha, l+\alpha]$, and choice of appropriate α will further be investigated in Section IV. Then using function z where $z(l', l) = 1$ *iff* $l' \in [l - \alpha, l + \alpha]$ and $z(l', l) = 0$ otherwise, we may use the following objective function instead of (1).

$$\max \sum_{i \in M_j} z(m(i, j), L(p(i), p(j))) \quad (2)$$

This leads us to find p that maximizes the number of path-loss-matched edges in m and L.

2) Algorithm and Complexity: Assume graph $G = (M_j, m)$ and complete graph $H = (N, N^2)$. The above optimization problem is the *maximum common subgraph isomorphism problem* (MCSI problem) in graph theory, which finds an induced graph G' of G in H with objective function (2). 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 $v \in N$, calculation of objective function (2) in this case can be induced to the following problem; (i) for each edge $(i, j) \in m$ find all possible edges $(u, v) \in N^2$ that satisfies $z(m(i, j), L(u, v)) = 1$, and (ii) find the positioning function $p : M_j \rightarrow N$ from (i) with objective function (2). Part (i) needs exhaustive tests of edge pairs in G and H respectively, and part (ii) can be induced to the *maximum bipartite matching problem* in the graph where M_j and N are bipartite vertices and possible allocations of nodes in M_j to points in N found in part (i) as well as (j, v) are edges. The computation complexity of part (i) is $O(|M_j||N|)$ and that of part (ii) is also $O(|M_j||N|)$ by the path matching algorithm [12]. Thus, the complexity of our optimization algorithm is given as $O(|M_j||N|^2)$ to apply the above parts (i) and (ii) to all points in N. The pseudocode is given below.

C. Location Convergence via Indoor Path Tracing

Our indoor path tracing mechanism, over a floor plan, uses the smartphone's sensors to collect movement direction and distance, which is known as dead reckoning. The challenge is to use unreliable inertial sensors in smartphones (e.g., magnetometer, accelerometer) to accurately track a users path.

We use Android's heading and magnetic field sensor to identify direction of the movement. During our initial implementation, we observed compass readings have some bias in addition to the fluctuation caused by even a slight sway, irregularity in motion and by magnetic fields in the

Algorithm 1 Positioning algorithm for node j

procedure find_possible_positioning_functions (M_j, m, N, L)
$k \leftarrow 0$; $F = \emptyset$
for each $v \in N$ **do**
  $E \leftarrow (j, v)$
  for each pair of edges $(i, j) \in m$ and $(u, v) \in N^2$ **do**
    $E \leftarrow E \cup (i, u)$ if $z(m(i, j), L(u, v)) = 1$
  end for
  $E' \leftarrow$ bipartite graph matching result for $((M_j, N), E)$
  if $|E'| > k$ **then**
    $F \leftarrow E'$; $k \leftarrow |E'|$
  else if $|E'| == k$ **then**
    $F \leftarrow F \cup E'$
  end if
end for
return F as a set of positioning functions
end procedure

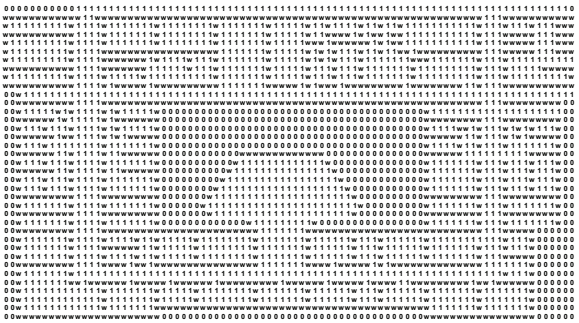


Figure 2. CS department building map: 1 (accessible), 0 (inaccessible), and W (wall)

surroundings. In order to overcome these errors and to accurately detect significant heading changes at a corner while taking a turn, we use a recently proposed method by Constandache *et. al.* [11]; and use the following condition for turn detection:

$$Avg(t_{(i+1)}) - Avg(t_i) \geqq \frac{StdDev(t_i) + StdDev(t_{i+1})}{2} \quad (3)$$

where $Avg(t_i)$ denotes the average compass readings over a t_i time period, $StdDev(t_i)$ denotes 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 number of steps taken and the step stride length. According to [11], the technique of double-integrating acceleration readings could induce a large error even with small distance travelled. Our algorithm to calculate user step count reads accelerometer data continuously, filters out the noise and infers an increment in step count based on changes in the observed readings.

We converted an example floor plan into a $N \times N$ matrix as shown in Figure 2 (2m $\times$ 2m grids). This map comprises of symbols $\{1, 0, W\}$, where 1 represents accessible points, 0 represents inaccessible points, and W is for walls. After the initial Wi-Fi scan, all possible 1s on the grids are enlisted. Every time a distance equal to that between two coordinates on the grid ($2m$) is traveled by the user, the path tracking module records user's direction of movement. Based on the observed movements, each coordinate in the list is updated and coordinates which fall on a "0" or "W" are eliminated. This process of elimination continues iteratively as the user moves around by changing directions and eventually narrows down to a single point on the grid which represents the users true position. Once one user locates himself correctly, he can broadcast his position to other team members such that they can further eliminate their invalid coordinates faster.

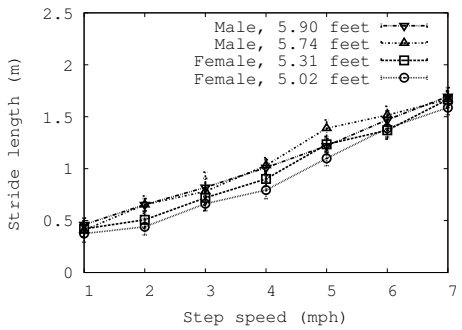


Figure 3. Stride length as a function of speed of walking (different gender and height)

Accurate estimation of distance walked: Experimentally we observed that walking speed also plays a crucial role in the calculation of the stride length. Therefore, we incorporated the moving speed and the corresponding stride length into our system. Our profiling approach consisted of two modes. In the profile creation mode, a user trains the system for his specific stride lengths for different walking speeds. In our experiments, we created four separate profiles for users of different heights and gender. The participants walked $100m$ in different speeds to train the system. This calibration process can be automated with GPS: a user simply walks a straight line of 100 meters. By walking a long distance, we can effectively eliminate the impact of GPS errors. Figure 3 shows the results of our experiments. It can be observed that independent of the height and gender, a user's stride length increases as the walking speed increases. It is also closely related to the user's walking styles. For example, a female of height 5.83 feet walking at a speed of 5 mph has similar stride length as that of a male of height 5.90 feet. This example justifies why it is necessary to use a personalized step profile to obtain more accurate distance traveled. It is noteworthy that height is not directly related to stride length, as claimed in [6].

Table I and Table II show the error percentage (distance deviation from the ground truth) in distance traveled for average (statistical) stride length and profiling stride length, respectively. According to [6], the average stride length for a male of height 5.83 feet is given as $0.737m$ (fixed value regardless of walking speed). Similarly for a female of height 5.5 feet, it is given as $0.637m$. Since stride length naturally increases with walking speed, this fixed stride length based approach tends to become more inaccurate at higher speeds increasing the error percentage. In contrast, our profiling based approach does well here. For instance, for a male of height 5.83 feet, the stride at moderate speed was observed to be 0.79, 1.39m at fast speed and 1.67m at running speed. When this information was used, the difference between the true distance and the calculated distance (as $stridelength \times stepcount$) dropped drastically. This dynamic adaptation is very important for our target scenario since the firefighters or rescuers could be walking and running intermittently for emergency operations.

Table I
A MALE'S (5.83 FEET) AVERAGE STRIDE LENGTH AND STEP-PROFILING RESULTS AT THREE DIFFERENT STEP SPEED LEVELS

Speed		Mod	Fast	Run
Distance (m)		100	100	100
Lap Time (sec)		74.5	44.7	31.9
Observed # of step		127	77	68
# of Step (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
A FEMALE'S (5.5 FEET) AVERAGE STRIDE LENGTH AND STEP-PROFILING RESULTS AT THREE DIFFERENT STEP SPEED LEVELS

Speed		Mod	Fast	Run
Distance (m)		100	100	100
Lap Time (sec)		74.5	44.7	31.9
Observed # of step		121	77	75
# of Step (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

IV. TESTBED EXPERIMENTS

A. Experiment Setup

To test overall system performance in the real-world scenarios, we performed field tests in the CS department building. We first generated a preprocessed floor plan with a grid overlay as shown in Figure 4. We performed field tests by varying the number of team members ranging from 2 to 9. We used stride length profiles for accurate dead reckoning. The same floor plan was used for path tracking (Figure 2). To verify how the system performance is influenced by 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), we deliberately chose three representative routes. These routes are $Route\ \#1$, $Route\ \#2$ and $Route\ \#3$ shown in Figure 4. Each route starts from the same starting point but while $Route\ \#1$ comprises of a long straight line path with just a single turn, $Route\ \#2$ contains more turns with long enough straight line paths between two consecutive turns. $Route\ \#3$ contains many turns with very short straight line paths after each turn.

B. Experimental Results

We analyzed the performance of CLIPS by evaluating the following: (1) impact of the team size and the slack value (2) convergence delay variation when taking different routes, (3) convergence accuracy with and without stride length profiles, and (4) overall convergence delay.

Impact of team size and slack value: Figure 5 shows feasible coordinate ratio with a different number of peers. We also varied the slack value of α to find a good estimate. Note that the value should be small enough to eliminate more infeasible points but big enough to not to miss the true location. The feasible coordinates ratio is calculated by dividing the number of feasible coordinates by the total number of coordinates in the floor plan (48×48 coordinates). While the α values showed a positive relationship with feasible coordinates ratio, we observed that the number of false positives decrease as we increase the number of peers. Figure 6 illustrates the *hit ratio* that is the probability of containing the true positive position in the matched outcomes. As Figure 5 shows, we varied the number of team members and also the size of α value to observe the system behavior in different cases. As we increase α, this probability reaches 1. Another interesting observation is that a smaller team (with few members) requires smaller size of α value in order to have the current position in the initial scan. For example, a one-member team with α value of 9dB was observed to scan the current position with 100% accuracy, while an 8 members team requires the α value of 15dB to have the same probability.

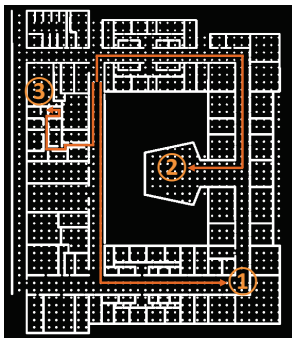


Figure 4. Preprocessed floorplan and Route $\#1, \#2$, and $\#3$

Convergence speed when taking different routes: Among the overall system performance metrics, convergence speed to the unique point accurately is one of key factors for the proposed scheme. We investigated how this factor is affected by the characteristics of routes and by the use of peer to peer exchange of RSS. As illustrated earlier, we carefully chose three representative routes, namely $Route\ \#1$ (including long straight line and a turn), $Route\ \#2$ (including moderate straight line path and few turns), and $Route\ \#3$ (including short straight line and many turns) to see which of the two factors, having long straight line, or the number of turns, affects the convergence speed more. We measured 20 times on each route and evaluated the success ratio of two different stride approaches.

Figure 7 shows the fraction of feasible coordinates as a function of traveled distance. $Route\ \#1$ shows rapid drop of infeasible coordinates ratio in the beginning, but after $20m$, the ratio decreases slowly. This was because there are multiple points on a straight line remained as candidates. Nonetheless, when a turn is taken after $60m$, our algorithm converges. $Route\ \#2$ shows the slowest convergence speed as it contains relatively small number of straight lines and turns. $Route\ \#3$ shows the fastest convergence speed as it contains many turns. Thus, based on our observation, a complicated route tends to expedite the process of convergence to a unique point. We noted that even without peer to peer RSS exchange and matching, the convergence results showed the similar trends, but in all three routes, this caused the user to travel longer distance to converge to a unique point. Figure 8 shows the distance

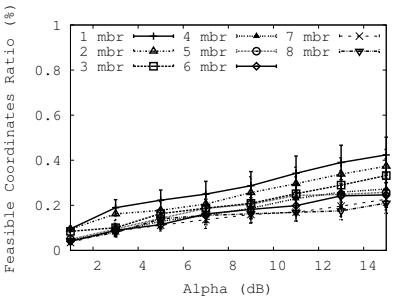


Figure 5. Feasible coordinate matching ratio

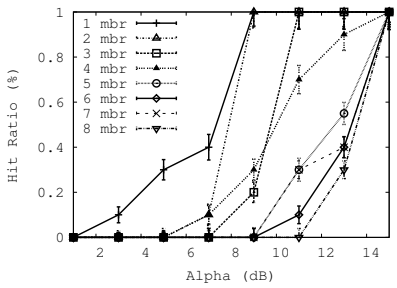


Figure 6. Probability of containing the current position

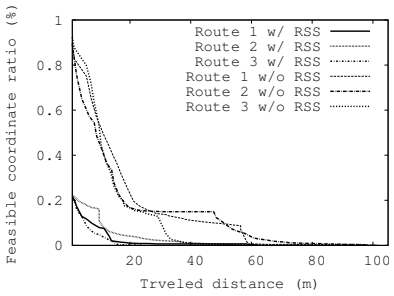


Figure 7. Convergence speed based on different routes with and without beacon RSS

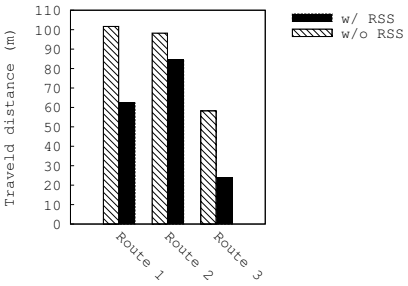


Figure 8. Distance travelled to converge unique point with and without beacon RSS

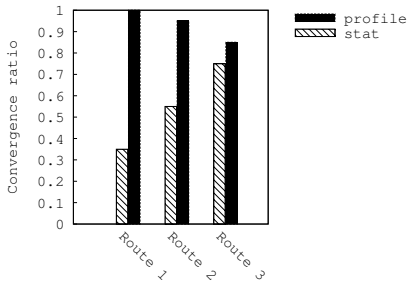


Figure 9. Convergence success ratio comparison

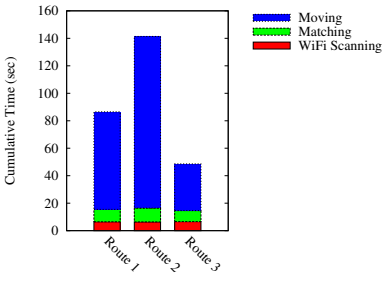


Figure 10. Cumulative latencies of three modules for different routes

traveled on the different routes till convergence, with and without RSS.

Convergence accuracy with stride length profiles: Accurate dead reckoning with stride length profiling is another key feature of the proposed scheme. To evaluate this, we performed the field tests: 20 times per route using two different stride length approaches, namely average (fixed) stride length and variable stride length with profiling. Since users are navigating indoors, it is highly necessary to have accurate displacement logging in the map shown in Figure 2. Unlike the average stride length approach, our step profiling uses dynamic stride length to calculate traveled distance. Eliminating false positives on the floor plan based on using this scheme would be more accurate. For instance, at the end of a corridor, both late and early, than required, changes of direction will cause failure of convergence to a unique position. Figure 9 shows the convergence ratio of two different mechanisms (fixed vs. variable), namely average stride length and step profiling approaches, which is well matched with distance deviations for both male and female shown in Table I and II, respectively. Since step profiling provides accurate distance measure so it provides 100% hit ratio on $Route\ \#1$ but slightly degraded performance on $Route\ \#3$. This can be attributed to the error in compass readings, caused by the magnetic field variation in surrounding. Even when profiling calculated the distance travelled correctly, error in orientation caused the inaccurate results. Note that to eliminate these errors, gyroscope and accelerometer can be used in addition to the orientation sensor as explained in [29], which is part of our future work. In our experiments, the average stride length approach significantly underestimated the actual distance traveled in a long straight corridor, resulting poor performance (32%). In the case of $Route\ \#3$, the use of average stride length showed relatively reliable performance due to many turns in the path.

Overall convergence delay of CLIPS: Figure 10 shows the aggregate time taken to converge to a unique point with step profiling in the three routes. The initial Wi-Fi scanning and matching took almost constant time for all the three scenarios. The delay difference mainly comes from the fact that users are traveling different routes. Overall, the delay is dominated by the traveling time, and Wi-Fi scanning and matching delay are much smaller when compared to the traveling delay. Figure 8 shows that Route 2 requires longer distance trip to converge to the unique point, which consequently takes the longest times compared to the other two routes.

V. Related Work

We consider two types of localization schemes that are closely related to CLIPS, namely inertial sensor based and model based localizations. For a complete survey of existing schemes, readers may refer to recent survey papers [16], [21].

Inertial sensor based localization: For a robot to navigate through an indoor environment, it must have the ability to determine its current location. Initial approaches provisioned the robot with a map of the indoor environments, allowing it to determine its location by comparing its observed environment to the map (e.g., using ultrasound or LADAR sensors). A significant step in the area of robotics was Simultaneous Localization and Mapping (SLAM) [20], which allowed a robot to build a map of the indoor environments (in terms of walls and other obstructions) while simultaneously determining its location with respect to the constructed map. Martin *et. al.* proposed an application for indoor localization with smartphones which use only the hardware embedded within the phone and integrating both online and offline phases of

RSS fingerprinting within the same device [23]. As shown earlier, the main drawback of these approaches is that they mostly suffer from slow convergence and large cumulated errors due to relative lack of position fixes indoors, which is detrimental to the time-critical indoor missions. To the best of our knowledge, our work is the first of its kinds that leverages both collaborative Wi-Fi beaconing and dead reckoning to drastically reduce the convergence time.

RF model based localization: An RF propagation model such as the log-distance path loss (LDPL) model can be used to predict RSS at various locations in the indoor environment. The advantage of using these models is that it reduces the number of RSS measurements dramatically compared to RF fingerprinting schemes, albeit at the cost of decreased localization accuracy. Since RF propagation characteristics vary widely, the model parameters would have to be estimated specifically for each indoor space in question. ARIADNE [18] uses sniffers at known locations but makes use of a more sophisticated ray-tracing model based on detailed indoor maps and uses simulated annealing to estimate radio propagation parameters. Given the signal measurements for a mobile, a proposed clustering algorithm searches that signal strength map to determine the current mobile's location. As discussed earlier, existing schemes require some kinds of infrastructure (e.g., fixed Wi-Fi APs, sniffers) which may not be available in an emergency scenario under consideration. The main departure is that CLIPS leverages mobile Wi-Fi beacons (i.e., other team members) to determine a set of coordinates and then employs dead-reckoning over a map to reduce the set and to pinpoint the real positions.

VI. Conclusions

We proposed a novel infrastructure-free collaborative indoor positioning system called CLIPS. Given that for emergency operations, the floor map and blueprints of a building are accessible, we proposed to build a realistic received signal strength (RSS) map using ray-tracing. When the mission starts, each team member uses the periodic peer-to-peer Wi-Fi beaconing to identify a set of feasible coordinates consistent with the RSS map. CLIPS can quickly remove invalid candidate coordinates and converge to a user's current position via dead reckoning over a floor map and information sharing of discovered coordinates. Our evaluation results with testbed experiments confirmed that CLIPS provides accurate localization with much lower position fix delay when it is compared with a non-collaborative scheme.

References

[1] Apple, MobileMe, Web Site. http://www.me.com.

[2] Google, Latitude, Web Site. https://www.google.com/latitude.

[3] Google Maps Floor Plans, Web Site. http://maps.google.com/help/maps/floorplans/.

[4] Insite, Web Site. http://www.remcom.com/wireless-insite.

[5] Skyhook, XPS, Web Site. http://www.skyhookwireless.com.

[6] Step size in pedometers, Web Site. http://walking.about.com/cs/pedometers/a/pedometerset.htm.

[7] C.R. Anderson and T.S. Rappaport. In-building wideband partition loss measurements at 2.5 and 60 GHz. *Wireless Communications, IEEE Transactions on*, 2004.

[8] P. Bahl and V.N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Infocom*, 2000.

[9] Gaetano Borriello, Alan Liu, Tony Offer, Christopher Palistrant, and Richard Sharp. WALRUS: wireless acoustic location with room-level resolution using ultrasound. In *MobiSys*, 2005.

[10] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. Indoor localization without the pain. In *MobiCom*, 2010.

[11] Ionut Constandache, Xuan Bao, Martin Azizyan, and Romit Roy Choudhury. Did you see Bob?: human localization using mobile phones. In *MobiCom*, 2010.

[12] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 3nd edition, 2001.

[13] Philippe Dosch, Karl Tombre, Christian Ah-Soon, and Gerald Masini. A complete system for the analysis of architectural drawings. *International Journal on Document Analysis and Recognition*, 2000.

[14] A. Falsafi, K. Pahlavan, and Ganning Yang. Transmission techniques for radio LAN's-a comparative performance evaluation using ray tracing. *Selected Areas in Communications, IEEE Journal on*, 1996.

[15] Brian Ferris, Dieter Fox, and Neil Lawrence. WiFi-SLAM using Gaussian process latent variable models. In *IJCAI*, 2007.

[16] Yanying Gu, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. *Communications Surveys Tutorials, IEEE*, 2009.

[17] J.R. Guerrieri, M.H. Francis, P.F. Wilson, T. Kos, L.E. Miller, N.P. Bryner, D.W. Stroup, and L. Klein-Berndt. RFID-assisted indoor localization and communication for first responders. In *EuCAP*, 2006.

[18] Yiming Ji, Saâd Biaz, Santosh Pandey, and Prathima Agrawal. ARIADNE: a dynamic indoor signal map construction and localization system. In *MobiSys*, 2006.

[19] H. Kim and H. Ling. Electromagnetic scattering from an inhomogeneous object by ray tracing. *Antennas and Propagation, IEEE Transactions on*, 1992.

[20] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *IROS*, 1991.

[21] Hui Liu, H. Darabi, P. Banerjee, and Jing Liu. Survey of Wireless Indoor Positioning Techniques and Systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 2007.

[22] Matthias Lott. On the Performance of an Advanced 3D Ray Tracing Method. In *Proc. of European Wireless & ITG Mobile Communication*, 1999.

[23] Eladio Martin, Oriol Vinyals, Gerald Friedland, and Ruzena Bajcsy. Precise indoor localization using smart phones. In *MM*, 2010.

[24] L.M. Ni, Yunhao Liu, Yiu Cho Lau, and A.P. Patil. LANDMARC: indoor location sensing using active RFID. In *PerCom*, 2003.

[25] Hyojeong Shin, Yohan Chon, Kwanghyo Park, and Hojung Cha. FindingMiMo: tracing a missing mobile phone using daily observations. In *MobiSys*, 2011.

[26] Stephen P. Tarzia, Peter A. Dinda, Robert P. Dick, and Gokhan Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *MobiSys*, 2011.

[27] R.A. Valenzuela. Ray tracing prediction of indoor radio propagation. In *PIMRC*, 1994.

[28] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 1992.

[29] Oliver Woodman and Robert Harle. Pedestrian localisation for indoor environments. In *UbiComp*, 2008.

[30] Guang yao Jin, Xiao yi Lu, and Myong-Soon Park. An indoor localization mechanism using active RFID tag. In *Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2006.