Social-Loc: Improving Indoor Localization with Social Sensing

Junghyun Jun* peterjun@sutd.edu.sg
Yu Gu* jasongu@sutd.edu.sg
Long Cheng*, Banghui Lu*† long_cheng@sutd.edu.sg
Jun Sun* sunjun@sutd.edu.sg
Ting Zhu‡ tzhu@binghamton.edu
Jianwei Niu† niujianwei@buaa.edu.cn

*Singapore University of Technology and Design, Singapore
†Beihang University, China
‡State University of New York, Binghamton, USA

ABSTRACT

Location-based services, such as targeted advertisement, geo-social networking and emergency services, are becoming increasingly popular for mobile applications. While GPS provides accurate outdoor locations, accurate indoor localization schemes still require either additional infrastructure support (e.g., ranging devices) or extensive training before system deployment (e.g., WiFi signal fingerprinting). In order to help existing localization systems to overcome their limitations or to further improve their accuracy, we propose *Social-Loc*, a middleware that takes the potential locations for individual users, which is estimated by any underlying indoor localization system as input and exploits both social *encounter* and *non-encounter* events to cooperatively calibrate the estimation errors. We have fully implemented Social-Loc on the Android platform and demonstrated its performance on two underlying indoor localization systems: Dead-reckoning and WiFi fingerprint. Experiment results show that Social-Loc improves user's localization accuracy of WiFi fingerprint and dead-reckoning by at least 22% and 37%, respectively. Large-scale simulation results indicate Social-Loc is scalable, provides good accuracy for a long duration of time, and is robust against measurement errors.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication

General Terms

Algorithms, Experimentation, System

Keywords

Indoor Localization, Social Interaction, Middleware

SenSys'13, November 11–15, 2013, Rome, Italy.

1. INTRODUCTION

Personal mobile devices such as smartphones have become popular in the past several years. Many mobile applications, such as Facebook, Twitter, Google+ and Foursquare, have revolutionized the way a person interacts with the surrounding environment [10, 20, 38, 2]. For many of these mobile applications, users' locations are critical to the proper functioning of applications. Since people spend most of their time indoors (either in offices or at home), there is an increasing demand for accurate indoor localization systems. However, GPS-based localization systems [4] are not suitable for indoor environments due to attenuation and the scattering of microwave signals. Therefore, different indoor localization technologies are necessary. The existing state-of-the-art solutions can be broadly categorized into four groups: 1) Proximity-based; 2) Range-based; 3) Signal-fingerprint-based; and 4) Dead-reckoning.

Although indoor localization has been well-studied, challenging issues remain in many aspects. Proximity-based localization [32, 12] requires less computation, but it provides only *coarse-grained accuracy* unless the density of reference nodes is high. Range-based localization, such as measuring Received Signal Strength Indication (RSSI), Time-Of-Arrival (TOA) [22], Time Difference Of Arrival (TDOA) and Angle-Of-Arrival (AOA), estimates the distances between a mobile device and multiple reference nodes so as to triangulate the mobile device's location. RSSI-based ranging localization [33] can be easily achieved without involving additional hardware, but provides an *unstable lower accuracy*, while alternative ranging methods [37, 5, 14] can achieve higher accuracy at the price of requiring specialized hardware, thus incurring *higher system and infrastructure costs*. Signal-fingerprint-based approaches [6, 29, 18] require *a lengthy and extensive training phase* to construct a signal fingerprinting database for every individual's indoor locations of interest, and such a fingerprinting database can change over time. [25, 31]. Furthermore, even with the movement of users in the indoor environment, the performance of signal-fingerprint-based systems could be affected [30]. Although Dead-reckoning-based systems [8, 24, 35] normally require no extra infrastructure support and no intensive training phase, they suffer from *lower localization accuracy*, especially after an extended period of time. Device-free passive localization [18] is mainly used for intrusion and anomaly detection or border protection.

In order to assist existing indoor localization systems to overcome their limitations, we explore the social sensing of mobile users in the indoor environment as virtual sensors. Specifically, we introduce a middleware, called *Social-Loc*, which exploits social events, such as the encountering multiple users, and uses them as virtual sensors to cooperatively refine the errors in the potential locations estimated from underlying indoor localization systems. The refined potential locations are then used by the underlying localization system to further enhance their performance. More interestingly, Social-Loc also utilizes events such as the non-encounter of users in the indoor environment to further improve the localization accuracy. Our contributions are as follows:

- We introduce Social-Loc, a novel middleware for existing indoor localization systems that exploits social sensing among mobile users as virtual sensors. To our best knowledge, this is the first work that provides a generic means of utilizing social interactions to improve existing indoor localization accuracy.
- To improve system stability, we formulate encountering and non-encountering of mobile users as stochastic events and propose a probabilistic approach to cooperatively calibrate their individual potential locations based on encounter and non-encounter events.
- We fully implement our design on the Android platform and perform both testbed experiments and large-scale simulations under two commonly used indoor localization schemes: particle-filter-based Dead-reckoning and WiFi fingerprinting. The testbed results, as well as large-scale simulations, demonstrate that Social-Loc is able to enhance underlying indoor localization accuracy and maintain its accuracy for a long duration.

The rest of this paper is structured as follows: The basic design is introduced in Section 2. Section 3 describes our advanced design. Section 4 reports our system implementation. Section 5 and Section 6 discuss our experiment and simulation results. Section 7 discusses security and privacy issues of Social-Loc. Section 8 discusses the related work. Finally, Section 9 concludes the paper.

2. BASIC DESIGN

Social-Loc is a transparent middleware that sits between basic indoor localization systems and location based applications. It aims to enhance the localization accuracy by utilizing social sensing among co-located mobile users.

Figure 1 shows the overall system architecture of Social-Loc. As shown in Figure 1, underlying indoor localization systems, such as particle-filters-based Dead-reckoning [1, 9, 28] and Fingerprinting [6, 29], localize indoor users and provide their potential locations to the Social-Loc system. The potential locations are a set of discretized geographical locations with nonzero probability of containing the user's current location. For example, potential locations from WiFi fingerprinting are a set of reference locations that are associated with the user's fingerprinting measurements.

The social sensing unit monitors social interactions among all users and identifies the encountered users. The detail of encounter detection is presented in Section 4.1.1. Social-Loc utilizes the users' encounter information to derive sets

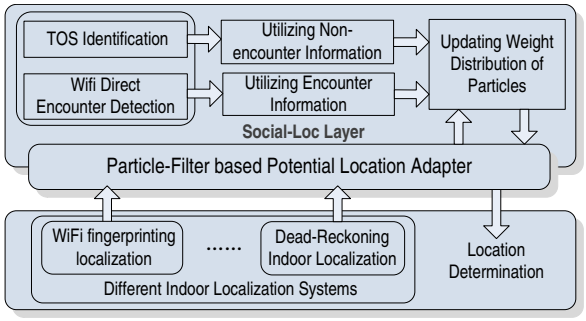


Figure 1: Social-Loc architecture overview

of non-encountered users who are currently not co-located, but still present in the same indoor environment.

There are two key components in Social-Loc: the *Potential Location Calibrator* and the *Potential Location Adapter*. First, the potential location adapter receives every user's potential location set from an underlying localization system and presents it as a distribution format. Then, the potential location calibrator probabilistically refines the potential locations based on encounter and non-encounter information from the social sensing unit. The underlying indoor localization system retrieves the refined potential location from the potential location adapter and then estimates the user's current location.

In the following sections, we introduce how Social-Loc utilizes both user encounter and non-encounter events to calibrate the estimation errors and improve the accuracy of underlying indoor localization schemes in detail. For the purpose of illustrating the key ideas of Social-Loc, we first present our design in a deterministic manner on cooperatively improving the potential locations of mobile users. In Section 3, we extend this basic idea to probabilistically update the weight of potential locations.

2.1 Utilizing Encounter Information

In an indoor environment, mobile users would normally encounter many other users over time, and such user encounter events are able to significantly help Social-Loc to improve the localization accuracy. Suppose the underlying indoor localization system initially estimates the location of user i at time t as a set $P_i(t) = \{p_1, p_2, \ldots, p_n\}$, where $p_1, p_2, \ldots, p_n$ are n potential locations. Assuming two mobile users i and j encounter each other at time t, then their potential location sets $P_i(t)$ and $P_j(t)$ can be reduced to:

$$P_i(t) = P_j(t) = P_i(t) \bigcap P_j(t) \quad (1)$$

This is because whenever two users encounter one another, this event relates them in space and time. As the intersection of two sets normally has much smaller cardinality than the original sets, it is clear to conclude that encounter information can improve the accuracy of localization.

To further illustrate the utilization of encounter information, we use the example in Figure 2(a). Suppose the underlying indoor localization scheme estimates the potential location sets of user A and user B as $P_A = \{1, 5, 7\}$ and $P_B = \{3, 5, 9\}$, respectively. As shown in Figure 2(a), both A and B are co-located at location 5; then Social-Loc discovers the encountered users and applies Equation (1). Consequently, both A and B can successfully localize to one specific location $P_A = P_B = \{1, 5, 7\} \bigcap \{3, 5, 9\} = \{5\}$.

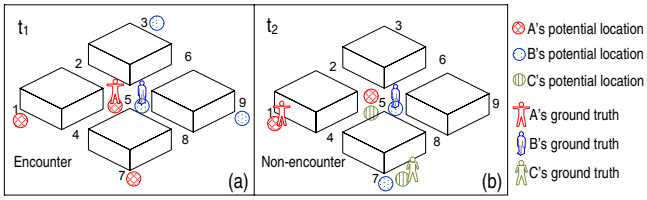


Figure 2: Examples illustrating utilization of encounter and non-encounter

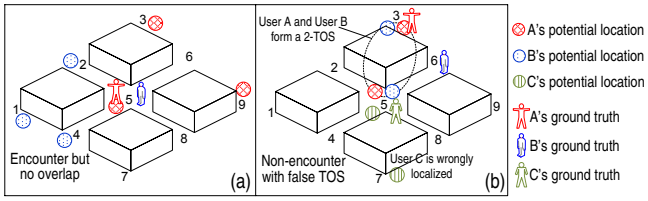


Figure 3: Examples illustrating problems of deterministically utilizing encounter and non-encounter

Here, we emphasize that intersecting potential location sets for encountered users do not always result in one specific location. Depending on the localization accuracy of the underlying localization system, it may contain no or multiple locations. This motivates us to design a probabilistic approach of utilizing users encounter information in Section 3.2.

2.2 Utilizing Non-Encounter Information

In contrast to encountering other users, it is more common for a user *not to* encounter other users over a period of time. Interestingly, in this work we reveal that such non-encounter information can also improve localization accuracy.

The most intuitive method to utilize non-encounter information is to remove overlapping elements from potential location sets of non-encountered users. For example, assume the potential location set for user A and user B is $P_A = \{1, 4, 6\}$ and $P_B = \{2, 5, 6\}$, respectively. If user A does not encounter user B we can remove the overlapping location $P_A \bigcap P_B = \{6\}$ from both P_A and P_B. However, this intuitive approach to utilize non-encounter information is incorrect. For example, user A may actually locate at location 6, while user B locates at location 5. Even though they do not encounter each other, we cannot simply remove the overlapping elements from their potential location sets.

To illustrate the main idea of how to effectively utilize non-encounter information, we use the example shown in Figure 2(b). The potential location set generated by the underlying indoor localization scheme for each user A, B and C is $P_A = \{1, 5\}$, $P_B = \{5, 7\}$ and $P_C = \{5, 7\}$ respectively. Users B and C do not encounter each other, but they both potentially locate at location 5 or 7. Therefore, we can infer that there is exactly one user locating at each location 5 and 7, but we can not distinguish whether it is user B or user C. Furthermore, since user A also encounters neither user B nor user C, we are certain that user A does not locate at location 5 or 7. Otherwise, user A would have encountered either user B or C. Consequently, user A can be localized to location 1 by removing the impossible location 5.

To generalize the above inference process, let us assume that there are n users who *do not encounter each other*, and their estimated potential location set at current time t is $P_1(t), P_2(t), \ldots, P_n(t)$. If we have

$$|P_1(t) \bigcup P_2(t) \bigcup \cdots \bigcup P_n(t)| = n, \tag{2}$$

then we can infer there is exactly one user at each of these n locations. For such a set of n users, we call them **n-Total Occupancy Set** (n-TOS). Take the example shown in Figure 2(b): users B and C form a 2-TOS since $|P_B \bigcup P_C| = 2$. Furthermore, for a given number n, there may exist multiple n-TOS involving different n users in the system.

Therefore, to fully utilize non-encounter information, assuming there are currently n users in the system, we need to first find all existing total occupancy sets such as 1-TOS, 2-TOS and so on, up to n-TOS at time t. Given a specific i-TOS, where $i = 1, 2, \ldots, n$, let the unions of all potential location sets for these i users be U_i. For each user j in the system who does not encounter any of the i users in this specific i-TOS at time t, we can update his or her potential location set as follows:

$$P_j(t) = P_j(t) - P_j(t) \bigcap U_i \tag{3}$$

Clearly, in this way, we improve the accuracy of the localization by effectively removing the impossible locations from the potential location sets of individual users. While utilizing TOS effectively improves the performance of indoor localization systems, we observe it is not trivial to find all TOSes in the system. In fact, we have proven this problem is NP-Complete. In Section 4, we show it is not necessary to find all TOSes in practical systems and discuss how to find useful TOSes in detail.

3. ADVANCED DESIGN

In Section 2, we introduce the main idea of our Social-Loc design in a deterministic manner to remove impossible potential locations. However, such a deterministic approach is error-prone if the original potential location sets or social sensing services contain errors. In this section, we first discuss the limitations of the deterministic approach with several examples and then extend the basic design to probabilistic approaches.

3.1 Limitations of the Basic Design

If the ground truth locations of mobile users are always included in their potential location sets generated by the underlying localization system, Social-Loc can effectively apply the deterministic method introduced in the previous section to improve the localization accuracy. However, due to various measurement and estimation errors (e.g., drift errors from inertial sensing), the estimated potential location sets from the underlying localization system may not even contain the ground truth locations of mobile users [28]. Under such scenarios, if we just directly apply the deterministic Social-Loc design, we may obtain very limited performance gain or even performance deterioration.

For example, as shown in Figure 3(a), user A and user B encounter each other at location 5, and their potential location sets[1] are $P_B = \{1, 2, 4\}$ and $P_A = \{3, 5, 9\}$ respectively. However $P_A \cap P_B = \emptyset$, then we can easily infer that the potential location set for user A or user B contains the error, but we can not further identify which user's potential location set contains error. To make the case even worse, if

$P_A = \{1, 3, 4\}$ and $P_B = \{3, 5, 9\}$ respectively, by applying the deterministic encounter method in the previous section, we incorrectly localize both user A and user B to location $P_A \bigcap P_B = \{3\}$ while the ground truth location is 5.

In terms of the deterministic non-encounter method, we refer to an example shown in Figure 3(b). Suppose users A, B and C do not encounter each other and their potential location set is $P_A = \{3, 5\}$, $P_B = \{3, 5\}$, and $P_C = \{5, 7\}$, respectively. Clearly, based on the definition of Total Occupancy Set (TOS), user A and user B form a 2-TOS as $|\{A, B\}| = |P_A \bigcup P_B| = 2$. By applying the deterministic non-encounter method, we would localize user C to location $P_C = \{5, 7\} - \{5, 7\} \cap \{3, 5\} = \{7\}$. However, as shown in Figure 3(b), the ground truth location for user C is location 5. From this example we can see that due to the errors in the original potential location sets, directly applying the deterministic non-encounter method has the possibility to erroneously remove the ground truth location of user C.

To resolve these issues, we introduce probabilistic methods that utilize the encounter and non-encounter information in the following sections. Conceptually, instead of completely removing potential locations, the probabilistic methods reduce the probability of those unlikely locations in the potential location sets such that the weight of ground truth location is always nonzero.

3.2 Probabilistically Utilizing Encounter Information

In the basic design section, each potential location in P_i is treated equally. However, for most existing indoor localization systems, users would have different probabilities in different potential locations [1, 28]. Therefore in this section, we extend the original potential location set by associating a weight to each potential location. Let Ω be a set of all potential locations in an indoor environment. w_i^u denotes a weight associated with a user u at the potential location $p_i \in \Omega$ and it represents the probability of user u currently located within the potential location p_i. Then $S_u = \{p_i \in \Omega | w_i^u > 0\}$ represents a set of feasible potential locations of a user u. For each user u, her potential location set is represented as:

$$P_u = \{(p_i, w_i^u) | p_i \in S_u\} \tag{4}$$

Then if two users A and B encounter each other, we set the potential location sets for user A and B as the union between S_A and S_B, denoted by $S_{AB} = S_A \bigcup S_B$. For the weight w_i^{AB} of each potential location $p_i \in S_{AB}$, we calculate it with the following equation:

$$w_i^{AB} = \begin{cases} w_i^A w_i^B & , p_i \in S_A \bigcap S_B \\ \delta w_i^A & , p_i \notin S_A \bigcap S_B \ \ \& \ \ p_i \in S_A \\ \delta w_i^B & , p_i \notin S_A \bigcap S_B \ \ \& \ \ p_i \in S_B \end{cases} \tag{5}$$

where $\delta = \min(\{w_i : p_i \in S_{AB}\}, \frac{1}{|\Omega|})$. Essentially in Equation (5), if a location p_i appears in both users A's and B's potential location sets, we calculate their joint probability $w_i^A w_i^B$. On the other hand, if the location p_i appears only in user A's or B's potential location set, we still consider it as a candidate location by assigning a smaller weight to p_i.

For the updated potential location set

$$P_{AB} = \{(p_i, w_i^{AB}) | p_i \in S_{AB}\} \tag{6}$$

[1]For brevity, we use P_i without t to represent the potential location set of a user i at a certain time in this paper.

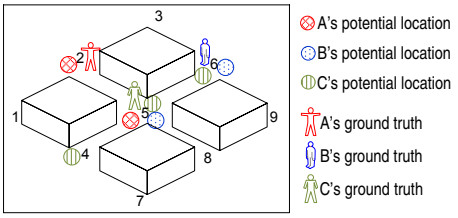


Figure 4: An example illustrating a problem of utilizing non-encounter information by a joint probability distribution

we normalize the weights of individual potential locations in S_{AB} as:

$$P_{AB} = \{(p_1, \frac{w_1^{AB}}{\sum_{i=1}^{|P_{AB}|} w_i^{AB}}), \ldots, (p_m, \frac{w_{|P_{AB}|}^{AB}}{\sum_{i=1}^{|P_{AB}|} w_i^{AB}})\} \tag{7}$$

Now let us revisit the example in Figure 3(a) with the newly introduced probabilistic method. Suppose, $P_A = \{(1, 0.3), (2, 0.3), (4, 0.4)\}$ and $P_B = \{(3, 0.1), (5, 0.7), (9, 0.2)\}$. Based on Equations (5) and (7), we have the normalized potential location set: $P_{AB} = \{(1, 0.15)\ , (2, 0.15), (3, 0.05), (4, 0.2), (5, 0.35), (9, 0.1)\}$. From the above results, we can see that even though the original estimated potential location set for user A does not contain the ground truth location 5, we can recover the ground truth location 5 by cooperatively updating their potential locations by the probabilistic method.

3.3 Probabilistically Utilizing Non-Encounter Information

In this section, we first illustrate that updating the weight of potential locations by computing the joint probability of non-encountered users does not always work. For example shown in Figure 4, three non-encountered users A, B, and C are located at 2, 6, and 5, respectively. Their underlying indoor localization system provides their individual potential locations $P_A = \{(2, 0.6), (5, 0.4)\}$, $P_B = \{(5, 0.3), (6, 0.7)\}$, and $P_C = \{(4, 0.1), (5, 0.6), (6, 0.3)\}$. The joint probability of user C locating at 5 is $0.6 \times 0.6 \times 0.7 = 0.252$. Similarly, the joint probabilities of user C locating at 4 and 6 are 0.19 and 0.02 respectively. After normalization, the updated potential location set of user C is $P_C^* = \{(4, 0.41), (5, 0.54), (6, 0.05)\}$. This example demonstrates that intuitively applying joint probability actually reduces the weight of the ground truth location of user C from 0.6 to 0.54 and increases the weight of unlikely potential locations of user C.

In this section, we extend the deterministic design discussed in Section 2.2 with the probabilistic method for 2-TOS and generalize the method to n-TOS case. Conceptually, probabilistic methods reduce the weights of unlikely potential locations instead of completely removing potential locations of users who satisfies n-TOS condition.

3.3.1 *Probabilistically Identifying 2-TOS*

Suppose two non-encountered users A and B with their potential location sets P_A and P_B are provided from an underlying indoor localization. Assume the top-2 potential locations for users A and B with respect to their weights are $S_A^2 = \{p_i, p_j\}$ and $S_B^2 = \{p_m, p_n\}$, where superscript 2 denotes they are top-2 potential locations. To identify whether or not user A and user B can form a 2-TOS, we first check

whether the top-2 potential location sets for user A and user B satisfy the following condition:

$$|S_A^2 \bigcup S_B^2| = |\{p_i, p_j\}| = 2 \tag{8}$$

If users A and B satisfy condition (8), we then calculate the *reliability* of 2-TOS they form (i.e., how likely user A and user B will locate at different locations within their joint top-2 potential locations).

Let E_2 denote the event where users A and B co-locating inside the same potential locations. $\bar{E_2}$ is the complement of E_2 which denotes the event where users A and B locate in two different potential locations such that they do not encounter. We have the non-encounter probability $p(\bar{E_2})$ as

$$\begin{aligned} &p(\bar{E_2}) = 1 - p(E_2) \\ &= 1 - \sum_{p_i \in S_A \bigcap S_B} w_i^A \cdot w_i^B \end{aligned} \tag{9}$$

Note that in order to calculate the encounter probability between users A and B, we must include *all* potential locations between user A and user B ($S_A \cap S_B$), not just those limited to the top-2 potential encounter locations.

Let $R_2(A, B)$ represent the reliability of the 2-TOS between user A and B, which is defined as the probability that user A and B locate within $\{p_i, p_j\}$ but they each locate at different locations (denoted as the event O_2). We can calculate the reliability of this 2-TOS based on the Bayesian conditional probability as:

$$\begin{aligned} &R_2(A, B) = p(O_2|\bar{E_2}) \\ &= \frac{p(\bar{E_2}|O_2) \cdot p(O_2)}{p(\bar{E_2})} = \frac{p(O_2)}{p(\bar{E_2})} \quad = \frac{w_i^A \cdot w_j^B + w_j^A \cdot w_i^B}{1 - \sum\limits_{p_k \in S_A \bigcap S_B} w_k^A \cdot w_k^B} \end{aligned} \tag{10}$$

where $p(O_2|\bar{E_2})$ is the conditional probability of O_2 given users A and B do not encounter each other. In Equation (10), $p(\bar{E_2}|O_2) = 1$ as the probability that two users A and B do not encounter each other under the condition that they each locate at a different location is 100%. Therefore, $p(\bar{E_2}|O_2)p(O_2) = p(O_2)$. $p(O_2) = (w_i^A \cdot w_j^B + w_j^A \cdot w_i^B)$ in Equation (10) denotes the probability that two users locate at different locations.

After obtaining $R_2(A, B)$, we utilize it to reduce the weight of unlikely potential locations for other users. For a user C who does not encounter both user A and user B, if $S_C \cap \{p_i, p_j\} \neq \emptyset$, we use the following equation to update the weight of a potential location $p_k \in S_C \cap \{p_i, p_j\}$:

$$w_k^C = w_k^C \cdot (1 - R_2 + \delta) \tag{11}$$

where $\delta = \min(\{w_i : p_i \in S_{AB}\}, \frac{1}{|\Omega|})$.

In Equation (11), we reduce the original weight of location p_i according to the reliability of the 2-TOS. The advantage of this weight assignment is that user C could avoid losing her ground truth location from the potential location set if the 2-TOS formed by users A and B is unreliable. After updating all relevant potential locations for user C, then we normalize all associated weights in P_C.

The δ is introduced in the Equation (11) to handle a situation in which the ground truth location is completely lost from the potential location set generated from the underlying indoor localization system. In this situation, Social-Loc can use this δ to recover the lost ground truth location later by utilizing encounter and non-encounter events.

To illustrate the advantage of probabilistically applying 2-TOS, we revisit the example in Figure 3(b). Suppose, $P_A = \{(3, 0.7), (5, 0.3)\}$, $P_B = \{(3, 0.5), (5, 0.5)\}$, and $P_C = \{(5, 0.6), (7, 0.4)\}$. Between users A and B, $P(\bar{E_2}) = 0.7 \cdot 0.5 + 0.3 \cdot 0.5 = 0.5$ since (p_3^A, p_5^B) and (p_5^A, p_3^B) are two possible cases of event $\bar{E_2}$ can occur. Based on Equation (10), we have the reliability of 2-TOS formed by users A and B: $R_2(A, B) = 1$. However, this is incorrect since the ground truth location 6 of user B was lost from the original potential location set generated by the underlying indoor localization system. In such case, Social-Loc still considers potential location 5 as a candidate location of user C by multiplying $\delta = 1/9$ to w_5^C such that its updated weight is 0.07.

3.3.2 Probabilistically Identifying n-TOS

After describing how to probabilistically identify 2-TOS, in this section we generalize the process for n-TOS. Suppose the top-n potential locations of user A is $S_A^n \subset S_A$, where $n = |S_A^n|$. We first search all other users who do not encounter user A and have the possibility to form n-TOS with user A. For user B with top-n potential locations S_B^n, if $|S_A^n \cup S_B^n| = n$, user B is considered a candidate to form a n-TOS with user A. If there are $m \geq n-1$ candidates to form n-TOS whose top-n potential locations are fully overlapped with S_A^n, then we need to check all $\binom{n}{m}$ possible combinations, calculate their corresponding n-TOS reliabilities and select the combination with the highest reliability.

Let $\bar{E_n}$ denote the event that n candidate users do not encounter each other for one of the candidate combinations. Since directly computing the probability of $\bar{E_n}$ is challenging, we calculate the probability that any two out of n users encounter at a potential location (E_2), up to n users encounter at a potential location (E_n). Then, we can obtain:

$$\begin{aligned} p(\bar{E_n}) &= 1 - \sum_{m=2}^{n} \sum_{k=1}^{\binom{n}{m}} p(E_m^k) \\ p(E_m^k) &= \sum_{p_i \in \bigcap_{k=1}^{m} S_k} \prod_{j=1}^{m} w_i^j \end{aligned} \tag{12}$$

where k is the index of all combinations when selecting any m users out of n users. $\sum_{k=1}^{\binom{n}{m}} p(E_m^k)$ is the encounter probability of any m users out of n users, given m specific users and their potential location set $\{P_1, P_2, \ldots, P_m\}$.

Let O_n denote the event that n users locate at n different locations within $\bigcup_{k=1}^{n} S_k^n$. Given n users $U^n = \{u_1, \ldots, u_n\}$ and n locations, there are $n!$ different permutations of locating n users at n different locations. Let $Z_i = \{z_1, \ldots z_n\}$ represent one of the permutation sequences of n locations. Therefore, there are at most $n!$ cases when O_n can happen. Since it is impractical to search for all $n!$ cases of O_n, in Section 4.1.2, we discuss how to practically utilize TOSes.

The probability for the event O_n is calculated as

$$p(O_n) = \sum_{i=1}^{n!} \prod_{z_j \in Z_i} w_{g(z_j, u_j)}^{u_j}, \forall u_i \in U^n. \tag{13}$$

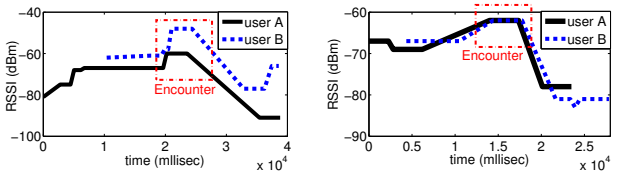
where $g(z_j, u_j)$ is the location index for $p_{g(z_j,u_j)} \in S_{u_j}^n$ for user u_j in the $z_j \in Z_i$. For example, if

$$P_A^2 = \{(p_1, w_1^A), (p_2, w_2^A)\}, \quad Z_i = \{p_2, p_1\}$$

then, $w_{g(z_1,A)}^A$ represents w_2^A.

Finally, to calculate the reliability of a n-TOS, we use the following formula:

$$\begin{aligned} &R_n(A, U^n) = p(O_n|\bar{E_n}) \\ &= \frac{p(\bar{E_n}|O_n) \cdot p(O_n)}{p(E_n)} = \frac{p(O_n)}{p(E_n)} \end{aligned} \tag{14}$$



(a) Samsung Galaxy Nexus, Android 4.1 (b) Samsung nexus S, Android 4.0

Figure 5: Encounter Detection

If $R_n(A, U^n)$ is the largest among all possible $\binom{n}{m}$ n-TOS involving user A, then we use this n-TOS to update the weights of potential locations for other users. The process is similar to 2-TOS cases in Equation (11).

4. SOCIAL-LOC IMPLEMENTATION

To illustrate the application of Social-Loc, we build a reference system on the Android platform and implement two underlying indoor localization schemes: particle-filter-based Dead-reckoning and WiFi fingerprinting. We first describe the implementation of the *Potential Location Calibrator*. Next, we describe the implementation of the *Potential Location Adapter* for each underlying localization system.

4.1 Potential Location Calibrator

Essentially, the potential location calibrator refines the user's potential locations based on the social interactions.

4.1.1 Encounter Detection

Refining potential locations requires effective and reliable detection of encounters among mobile users to function properly. In our implementation, we utilize built-in WiFi direct mode on the Android[1]as our encounter detection mechanism. During the operation, each mobile phone periodically scans for the beacon messages transmitted from other mobile phones. We detect an encounter event between two mobile users based on the presence of a peak RSSI from both encountered users in the sequence of received beacon messages. Intuitively, every encounter event should generate a peak RSSI since RSSI in general increases when two mobile users approach each other and decreases when they depart after the encounter. Figure 5 shows an RSSI trace of received beacon messages collected by mobile users in our experiment. The beacon messages are from users' smartphones in WiFi direct mode. It was observed during the experiment that the time of encounter events corresponds to the presence of RSSI peaks in Figure 5.

Since typical communication range of WiFi-direct is 200m, we set the RSSI-peak threshold to ensure an encounter happens within a potential location (about 3m×3m for Dead-reckoning and 5m×5m for WiFi fingerprinting). When two users are geographically close to each other, but separated by an obstacle, our encounter detection implementation considers this as a non-encounter based on RSSI-peak threshold.

[1]Unfortunately, WiFi direct mode in Android cannot simultaneously operate with regular WiFi. Therefore, we could not implement this in a single smartphone. We worked around this issue by using two smartphones per user during our experiments: One smartphone with WiFi direct mode and another smartphone providing the user's potential locations to Social-Loc and scanning WiFi-direct.

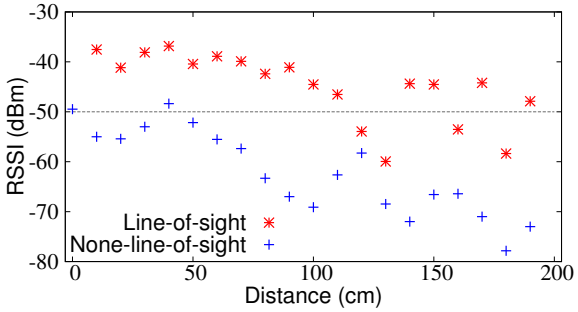


Figure 6: WiFi-direct RSSI variation between two users

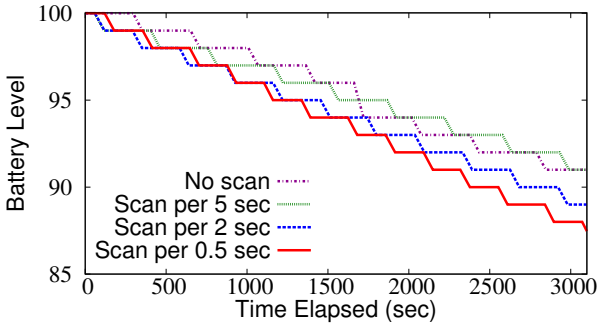


Figure 7: Smart Phone Energy Level

Figure 6 shows the changes of RSSI with respect to distance. For the non-line-of-sight case, there is a 15cm thick concrete wall between two users. Our observation shows, RSSI-peak threshold of -50dBm detects an encounter as long as two users are within 3m range and distinguishes it from the non-line-of-sight case. As long as encounter detection is reliable, Social-Loc can maintain good indoor localization accuracy by utilize non-encounters even if the encounter rate is low. Therefore, our encounter detection method detects few reliable encounters rather than many erroneous encounters.

In our experiment, when the scanning period is set to 2 sec/scan the encounter detection probability was almost 100% regardless of smartphone models as shown in Figure 5(a) and Figure 5(b). For our experiment, 100% accurate encounter detection was necessary for a small number of users. However, this increases average energy consumption of smartphones by 24% as shown in Figure 7. In practice, we expect that the population density of any indoor environment would be much higher than our experimental settings, and our large-scale simulation in Section 6.6 shows Social-Loc can tolerate detection errors up to 50% in such a case. Therefore, Social-Loc should still perform effectively when the encounter detection runs at a low scaning rate of 5 sec/scan. Figure 7 shows the energy cost of encounter detection is negligible at this rate.

4.1.2 Practically Finding TOSes

As discussed in Section 2.2, Social-Loc uses Total Occupancy Sets (TOS) for the non-encounter events to reduce the weights of users' unlikely locations. Since finding all TOSes is NP-Complete, here we discuss how to practically find TOSes that can effectively improve the localization accuracy in the system.

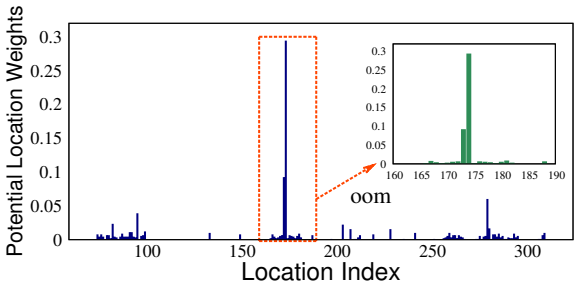


Figure 8: Weight distribution of potential locations

First of all, although there are many potential locations, often there are only a few effective potential locations near a user's ground truth location. Figure 8 shows the weight distribution of potential locations estimated from the Social-Loc experiment after a representative user walked for nine minutes. It is clear from Figure 8 that only the top-2 potential locations near the user's ground truth location are the ones that effectively estimate the ground truth. Therefore, it is not necessary to search for all n-TOSes in practice.

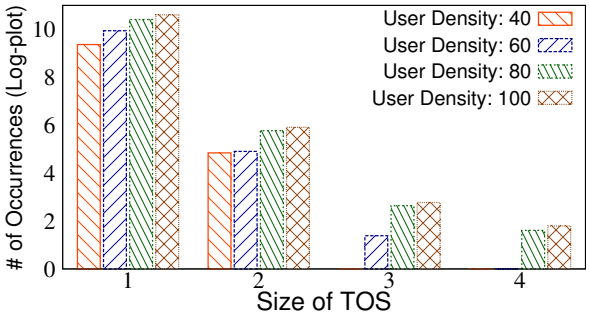


Figure 9: Distribution of TOS in log-plot. TOS larger than 5 never occur in the case of 100 users in the Dead-reckoning simulation.

To investigate how often n-TOSes with different n values occur, we exhaustively search all n-TOSes with n up to 5 in a Social-Loc simulation with 100 users and show the results in Figure 9. The average occurrences of 1-TOS, 2-TOS, 3-TOS and 4-TOS for 30 minutes are 40319, 368, 16, and 6 respectively. In Social-Loc, 1-TOSes represent users who are localized to a single potential location. High occurrences of 1-TOS in Figure 9 are due to Social-Loc's effectively localizing users to a single potential location. From Figure 9, it is clear that the occurrences of 3-TOS and above are very low. Therefore in a practical system implementation, it is sufficient to consider and search for up to 2-TOS with $O(n^2)$ complexity. To find all n-TOSes up to $n = 2$, we follow the procedures described in Section 3.3.

4.1.3 *Utilizing Encounters and Non-Encounters*

The key mechanisms for Social-Loc to improve localization accuracy are to increase the weights of more likely potential locations and decrease the weights of less likely potential locations. After obtaining encounter information and identifying TOSes, we update the weights of potential locations for individual users as described in Section 3.

4.2 Potential Location Adapter

In Social-Loc, the potential location adapter handles the translation of potential locations from the underlying indoor localization system to a unified format for the Social-Loc and applies the updated potential locations to an underlying localization system. Since the design of this adapter depends on the underlying indoor localization system, we implement particle-filter-based Dead-reckoning and WiFi fingerprinting. We choose these two as underlying indoor localization since they are the most representative categories of an indoor localization system. However, we argue that better underlying indoor localization would only benefit Social-Loc since Social-Loc is designed to leverage on the users with good localization accuracy to reduce localization errors of other users upon their encounters. We provide implementation detail in the next two sections.

4.2.1 *WiFi Fingerprinting*

To demonstrate the application of Social-Loc, we implement WiFi fingerprinting based on Horus [36] as one of our underlying indoor localization systems. Specifically, we collect WiFi measurements manually at 89 reference locations, spaced about 3m apart and spread across the floor. Floor layout is shown in the Figure 10. Here, we denote these 89 reference locations as set X. At each reference location, 1000 WiFi beacons are collected for each proximate access point (AP) using the smartphones. The average number of proximate APs at each reference location is five in our testbed. This data is used to train Horus system for generating radio map. The radio map is represented as a set of reference locations. The RSSI measurement corresponding to each reference location and each AP at each reference location is presented as a non-parametric distribution (normalized).

The users scan WLAN periodically for available APs and collect their signal strength vector $s = (s_1, \ldots, s_k)$. In Horus, the discrete space estimator localizes the user to a location $x \in X$ where probability $P(x|s)$ is maximum. It is shown in Horus that location x maximizing $P(x|s)$ is equal to location x maximizing $P(s|x)$ based on the Bayes's theorem.

Generating Potential Locations from Horus. The potential location adapter directly translates 89 reference locations $X = \{x_1, \ldots x_{89}\}$ in the radio map as total potential locations $\Omega = \{p_1, \ldots p_{89}\}$ for Social-Loc. Each reference location represents a small area of the building floor. Any user inside that area is localized to that reference location.

Given a signal strength vector $s = (s_1, \ldots, s_k)$ from user u, the discrete space estimator in Horus first computes probability $P(s|x) = \prod_{i=1}^{k} P(s_i|x)$, for all locations $x \in X$ in the radio map. Before the estimator computes for the $x \in X$ which maximizes $P(x|s)$, potential location adapter interrupts Horus to obtain a set of $P(x|s)$ for $x \in X$. The weight w_u^i of each potential location $p_i \in \Omega$ represents $P(x|s)$ after normalization.

Once all the weights are assigned to the potential locations they are forwarded to the potential location calibrator component of Social-Loc, which refines their weights.

Applying Updated Potential Locations to Horus. Resuming the regular online operations of Horus for a user u requires probability $P(x|s)$ for $x \in X$ and the previous estimated location. After refining potential locations, the potential location adapter redistributes the weight of each potential location $p_i \in P_u$ to its corresponding reference

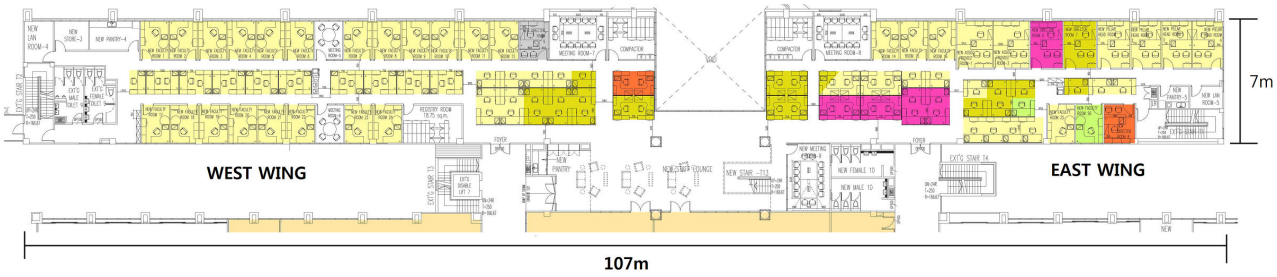


Figure 10: Floor Map

locations $x_i \in X$. Also, it resets the previously estimated location to be a reference location that has the maximum weight before resuming the interrupted Horus.

4.2.2 *Dead-Reckoning*

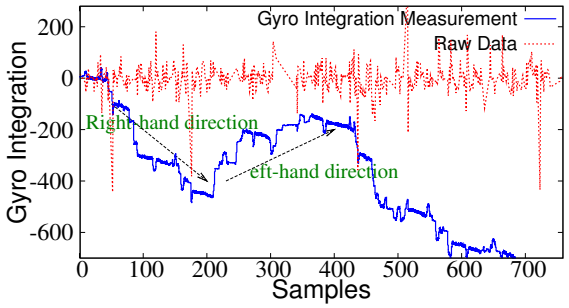


Figure 11: Turning event detection

To demonstrate the effectiveness of Social-Loc, we also design and implement a particle filter-based Dead-reckoning scheme as another underlying indoor localization system. Specifically, we use the built-in accelerometer and gyroscope sensors on the smartphones to measure the inertial changes and perform particle-filter-based location estimation.

Moving distance is estimated from a pedometer which counts the number of steps that a user has walked based on the periodic changes in the vertical direction of the accelerometer [23]. We have tested the accuracy of our pedometer under three different conditions, including putting the smartphone in the hand, in the front-pocket, and in the back-pocket of six different individuals. The accuracy of our pedometer under these three conditions is 99.2%, 97.8%, and 94.5%, respectively.

After obtaining the number of steps that a user has walked, the moving distance is estimated by multiplying the average stride length to the step count. In this work we manually measure the average stride length of each participant before the experiment. The average stride length of these participants is between $0.65m$ and $0.8m$.

We estimate the user's moving direction by double integrated gyroscope sensor readings from our experiment. Although it is impossible to obtain the precise turning angle, we can still estimate the turning directions based on the changes in the slope of gyroscope sensor readings. In Figure 11, a decreasing slope indicates a right turn and an increasing slope indicates a left turn. However, when the smartphone is inside the pocket, the constant changes in the orientation of the phone cause a high chance of false-positive and false-negative turnings in the measurement.

After obtaining the moving distance and turning information, we estimate the most likely location of individual users by particle-filter. Particle-filter estimates the current location of a mobile user based on her previously estimated location, her moving distance and the direction obtained from inertial sensors. The estimated location is represented with a set of weighted particles.

Generating Potential Locations from Particle Filter. Conceptually, the weight of each particle represents the probability that the user resides at that particular particle's location. However, in an indoor environment it is more effective to estimate the user's ground truth location based on the density of particles around that location since any particles entering the inaccessible areas will be considered ineffective and their weights would be set to zero until resampling of particles. Specifically, the user's ground truth location is more likely to be found in the area where the density of particles is high. Therefore, the potential locations P_A of user A and their associated weights are derived from the density of particles around them. Here, we divide the indoor environment into a set of $2 \times 2m^2$ rectangular grids and assign potential location weight w_i^A of location $p_i \in P_A$ by total weights of particles residing in p_i. Since some particles are ineffective, w_i^A is normalized based on Equation (7).

Applying Updated Potential Locations to Particle Filter. After the potential location update, potential location adapter redistributes particles to the potential locations based on their updated weights. For example, assume the potential location set for user A at time t_0 is $P_A(t_0) = \{(p_1, 0.6), (p_2, 0.4)\}$, and the new potential location set at time t_1 is $P_A(t_1) = \{(p_1, 0.75), (p_2, 0.25)\}$ after applying Social-Loc. To reflect this change of weights for various potential locations, we redistribute 75% of the particles to location p_1 and the remaining 25% of the particles to location p_2 by copying the live particles (particles with weights greater than zero) near the locations p_1 and p_2.

5. EXPERIMENTAL EVALUATION

In this section we present an experimental evaluation of Social-Loc on top of two underlying indoor localization systems: Dead-reckoning and WiFi fingerprinting.

5.1 Experiment Setup

Both experiments are conducted on one floor of our office building; the floor map is shown in Figure 10. The area of the floor is $107m \times 12m$. Although our testbed is a long straight corridor that contains few corners, this layout benefits neither WiFi fingerprinting nor Dead-reckoning. The reason is our testbed is wide open corridor where RSSI

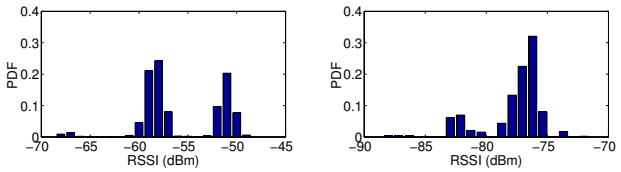
Figure 12: Experiment Snapshot

radio fingerprint's similarities between many pairs of reference locations are high and having fewer corners actually increases the accumulative errors of the particle-filter-based Dead-reckoning.

Six people participate as mobile users in the Dead-reckoning and WiFi fingerprinting experiment for a period of 25 minutes. In order to obtain the ground truth location for individual mobile users, we place 89 RF-tags uniformly across the corridor and meeting rooms at the experiment site. To record the ground truth location, each user holds a smartphone with near field communication (NFC) module on his hand to scan the RF-tags along the walking path. Whenever a user encounters another user, he or she scans the RF-tag placed on the other user's arm to collect the ground truth encounter information. Before the experiment, we synchronize the time of all the smartphones to record the ground truth encounter times. Figure 12 shows a snapshot during the experiment.

5.2 WiFi Fingerprinting

5.2.1 *Average Localization Error Over Time*



(a) AP 1 (b) AP 2

Figure 14: Examples of the normalized signal strength histograms from two different access point in the testbed

In Figure 13(a), we show the average cumulative localization errors of WiFi fingerprinting and Social-Loc for all users over time. Since we assume that the initial starting location of each user is known, everyone starts with zero localization error. For all users, the average cumulative localization accuracy of WiFi fingerprinting converges to 11m overtime. This 11m is worse than the localization accuracy reported in Zee [23] and Horus [36]. This is due to dynamic power control in our WLAN network in which RSSI measured at a given location does not follow any commonly known parametric distributions (e.g. Gaussian Distribution). The evidence of this dynamic power control is shown in Figure 14.

Nevertheless, Social-Loc further improves system average accuracy by 45%. In Figure 13(b), we show average localization errors of each individual user. The average performance

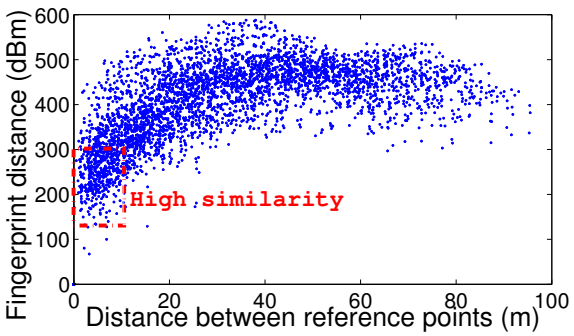


Figure 15: Similarity between two reference locations

of WiFi fingerprinting is around 10m and its variance is as large as 20m. The large localization errors are mostly due to power control. However, we have observed that users are often localized to different nearby reference points for each consecutive fingerprinting measurement. These small variances (< 5m) are due to similarity between two fingerprinting measurements from two nearby reference locations, respectively. Figure 15 shows the evidence of this similarity. Social-Loc effectively eliminates these jumps between nearby reference locations by reducing the weight of impossible potential locations based on users' encounter and non-encounter events. As a result, Social-Loc limits this large variance and improves average localization accuracy of all six users by at least 22%.

5.2.2 *Instantaneous Localization Error Over Time*

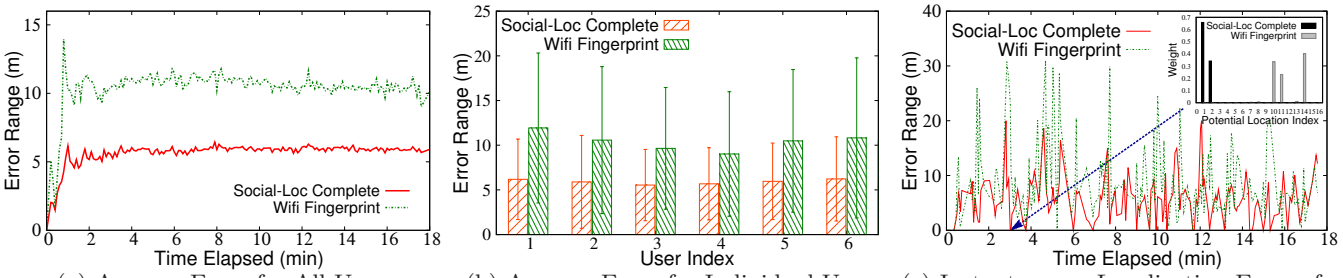
In Figure 13(c), we show the instantaneous localization error of a representative user. Figure 13 shows Social-Loc improves the localization accuracy of WiFi fingerprinting in many instances. Especially at the 3-minute mark, this user encounters another user whose localization error is close to zero. Social-Loc utilizes this event and calibrates the potential locations. The evidence of this effect is shown in the small embedded figure of Figure 13. The embedded figure shows that Social-Loc shifts the distribution of poorly estimated potential location set $\{10, 11, 14\}$ to correct potential locations $\{1, 2\}$.

5.3 Dead-Reckoning

5.3.1 *Average Localization Error Over Time*

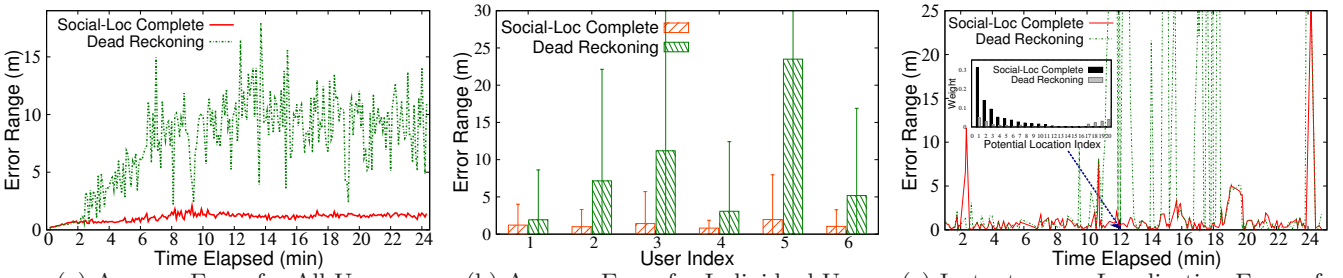
In Figure 16(a), we show the average cumulative localization errors of Dead-reckoning and Social-Loc for all six users. It shows Social-Loc can improve the localization accuracy of Dead-reckoning overtime and maintain this good accuracy during the entire 25 minutes of our experiment. In practice, maintaining good localization accuracy for a long duration of time is as important as low localization errors.

Figure 16(b) shows the localization accuracy of each individual user in the experiment. In contrast to WiFi fingerprinting, localization accuracy of Dead-reckoning varies significantly among different users. For example, user 1 and user 5 in Figure 16(b) show localization accuracy of 1.9m and 23.5m respectively. This is due to the difference in their moving patterns and walking trajectories overtime. In such a case, Social-Loc can cooperatively calibrate potential locations of user 5 with poor localization accuracy by utilizing



(a) Average Error for All Users (b) Average Error for Individual Users (c) Instantaneous Localization Error of a Representative User

Figure 13: Social-Loc Complete vs. Wifi Fingerprinting



(a) Average Error for All Users (b) Average Error for Individual Users (c) Instantaneous Localization Error of a Representative User

Figure 16: Social-Loc Complete vs. Dead Reckoning

user 1 with good localization accuracy. As a result, user 5 improves her average localization accuracy by 91%. Even for user 1, Social-Loc further improves her average localization accuracy by 37%.

5.3.2 *Instantaneous Localization Error Over Time*

In Figure 16(c), we show instantaneous changes in localization accuracy of Dead-reckoning and Social-Loc. From Figure 16(c), we notice huge spikes that go up as high as $50m$. These indicate the times when Dead-reckoning incorrectly estimates users' ground truth locations. Once the ground truth location is lost it take either long time for Dead-reckoning to correct this error or often it never recovers the ground truth location.

On the other hand, Figure 16(c) shows Social-Loc successfully maintains instantaneous localization errors of six users below $5m$ for 90% of the time. Especially at the 12-minute mark, Social-Loc improves the accuracy from 50m to 2m by utilizing encounter events to calibrate the potential locations. The evidence of this effect is shown in the small embedded figure of Figure 16(c). The embedded figure shows that Social-Loc shifts the distribution of the poorly estimated potential locations set to the correct potential locations set and their weights are concentrated within a smaller potential locations set. Concentrating the weight distribution to a small number of potential locations is a key advantage of Social-Loc. This helps underlying Dead-reckoning to keep more particles alive near the user's ground truth location for a longer duration of time.

6. SIMULATION-BASED EVALUATION

To evaluate the performance of Social-Loc in a large-scale indoor environment with a large number of mobile users ($>$ 10 users), we simulate Social-Loc with the Dead-reckoning as an underlying indoor localization. We simulate Social-Loc to Dead-reckoning because an accurate WiFi fingerprinting model for simulation is not available. We compare the performance of Social-Loc with the following approaches:

- **Social-Loc-Complete:** Social-Loc utilizing both encounter and non-encounter information for localization.
- **Social-Loc w/ Encounter:** Social-Loc utilizing only encounter information for localization.
- **Dead-reckoning:** A typical inertial localization method using the particle-filter-based dead-reckoning.

6.1 Simulation Setup

In our simulation, we varied the number of users from 5 to 300 in a square area of $144m \times 144m$ with 100 rooms. Square rooms $10m \times 10m$ in size are regularly placed in a grid-like topology, and adjacent rooms are separated by corridors of $2m$ wide.

Mobility Model: Random waypoint mobility model with resting is implemented to simulate users' mobility with average velocity of 1.44m/sec (approximately equal to 2 steps/sec). Each user starts walking towards a randomly selected destination along the corridor. Once the user reaches the destination, he or she rests for some exponential random time. When the resting is over, the users start walking again toward another randomly selected destination. This process is repeated until the 3000 seconds simulation ends.

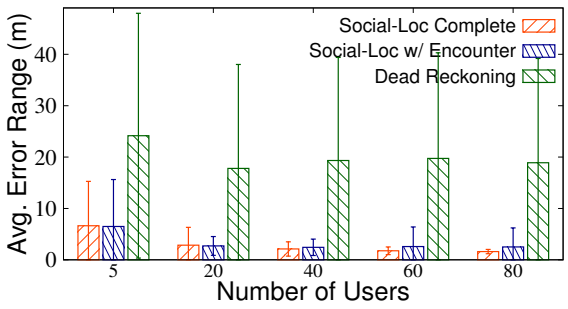


Figure 17: Number of Users vs. Localization Error

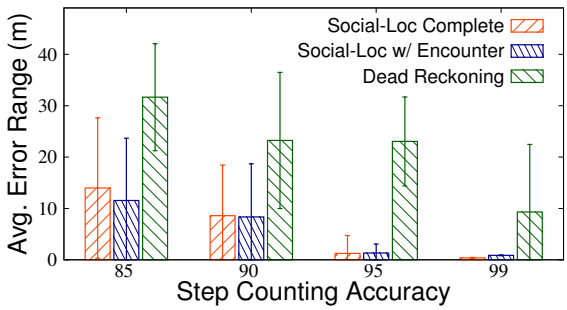


Figure 18: Step Counting Accuracy vs. Localization Error

Error Model: Moving trajectories of users are estimated based on the number of steps taken and turning angles. However, in practice these values are prone to measurement errors. In our simulation, we introduce error models for these values to closely reflect the data collected from our experiment. The step-counting errors are geometrically distributed random integer numbers with a step-error probability of 1% to 15%. The 15% is our worst bound step-counting errors, since our implemented pedometer accuracy varies from 90% to 99%. Turning angle error is modeled as zero mean Gaussian distributed random variable with different variances and added to each step. Small turning errors from zero mean Gaussian distributed random variable with small variance is used to reflect turning errors during walking straight. Large turning errors from zero mean Gaussian distributed random variable with a larger variance is used to reflect turning errors during the actual turnings.

6.2 Impact of Number of Users

The performance of Social-Loc greatly depends on the encounter and non-encounter events in the system. Although exact encounter rate depends on many factors, such as floor layout, mobility patterns and so on, the number of users in the system is generally positively correlated with the frequency of user encounters. Therefore, in this section, we investigate the impact of number of users on the localization accuracy. Figure 17 shows the average localization errors when we use the basic Dead-reckoning, Social-Loc with only encounter information, and Social-Loc-Complete. From Figure 17, we observe that the average localization errors for the Dead-reckoning are around $20m$ and independent of number of users, while Social-Loc takes the advantage of the increasing number of users. For Social-Loc with encounter, the average localization error drops 67% when the number of users increases to 80. For Social-Loc-Complete, the average localization error drops 77% when the number of users increases to 80.

From Figure 17, we can also observe that when the number of users in the system is relatively small (e.g., less than 20), the performance between Social-Loc with only encounter and Social-Loc-Complete is very similar. This is because if there is a small number of users, the user encounter rate reduces. Consequently, the location weights would be more uniformly spread out among all potential locations. In this case, Social-Loc can not effectively utilize non-encounter events since it is also difficult to form many 1-TOSes or 2-TOSes. When the number of users is greater than 20 Social-Loc can greatly take the advantage of non-encounter events. We can see this happening in Figure 17 where the average localization error for Social-Loc-Complete is about 36% smaller than that of Social-Loc with encounter when there are 60 users.

6.3 Impact of Step-counting Accuracy

From our experiment, we find that the pedometer accuracy is below 97.8% when the smartphone is inside the user's front pocket. However, the step-counting errors can cause poor estimation in the step length and make the underlying Dead-reckoning system miscalculate the turning time. For example, a ground truth turning happens after a certain number of steps, but the turning is detected a few steps earlier or later due to the step-counting error. In this case, the Dead-reckoning system can incorrectly assign a lower probability to the potential location containing the user's ground truth location. In this simulation, we show Social-Loc calibrates these errors in the potential locations of the Dead-reckoning system.

To investigate the impact of step-counting accuracy, we set gyroscope error and stride length error to zero. Figure 18 shows that when step-counting accuracy decreases from 99% to 95% the localization error of Dead-reckoning increases sharply whereas the localization error of both Social-Loc and Social-Loc with encounter remains below 2m. When the step counting accuracy decreases from 99% to 85%, the localization error for Social-Loc-Complete increases from 0.36m to 13.9m. From Figure 18, we can see that both Social-Loc approaches can improve the localization accuracy by 53% compared to particle-filter-only design even when the step counting accuracy is only 85%. Since the step-counting accuracy in our experiment is around 95%, this result is consistent with our experiment results in Section 5.

6.4 Impact of Stride-length Measurement Errors

In the Dead-reckoning, the moving distance for a mobile user is estimated by the number of steps that a user walks multiplied by the stride length. As described in Section 4.2.2, we estimate the number of steps that a user has walked by analyzing data from the accelerometer sensor. However, a person's stride is usually multi-modal: The stride length varies depending on how fast he/she is walking.

Therefore, in this section we study the impact of errors in the stride-length measurement on the localization accuracy. Figure 19 shows the effect on the localization error by varying the errors in each stride-length measurement. Essentially, the larger error is added to the moving distance

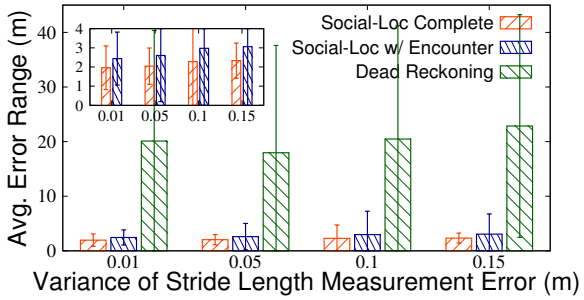


Figure 19: Variance of Stride Length Measurement Error vs. Localization Error

by increasing the standard deviation of the stride-length measurement error model, and this stride-length error accumulates over time if there is no reference existing in the system to correct the error. For example, the localization error of Social-Loc-Complete only increases from 1.96m to 2.33m when the stride-length measurement error increases from 0.01m to 0.15m. Interestingly, in Figure 19, we observe the impact of such stride-length measurement errors is relatively small.

During our empirical experiments, the stride-length error normally is around 0.03m per step. From these results, we can conclude that our Social-Loc design is insensitive to the stride-length errors.

6.5 Impact of Gyroscope Errors

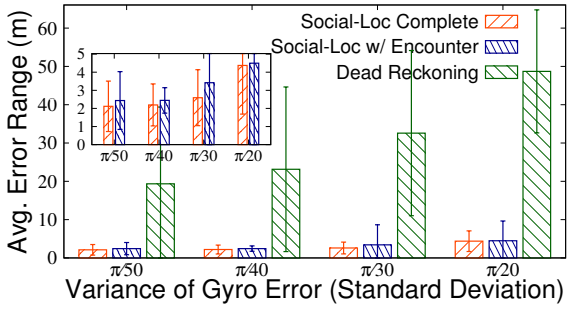


Figure 20: Variance of Gyroscope Error vs. Localization Error

Social-Loc uses gyroscope sensor readings to detect turning information of mobile users. In this section, we study the impact of gyroscope errors on the localization accuracy. In the case of Dead-reckoning, Figure 20 shows that increasing gyroscope errors lead to larger localization errors. This is because a larger gyroscope error leads to more false positive turnings. Any false positive turning error can reduce many particles near the ground truth location. For example, a false positive turning can be declared while the user is walking along a straight corridor with walls on either side. If this happens, some particles that take a turn would get lost when they cross the walls. However, from Figure 20 we can see that the Social-Loc-Complete maintains relatively low localization errors of 5m, which is 10 times less than Dead-reckoning since false positive turnings can be calibrated whenever the users encounter each other. This result indicates that Social-Loc improves the robustness of underlying Dead-reckoning against the measurement errors.

6.6 Impact of Encounter Detection Error

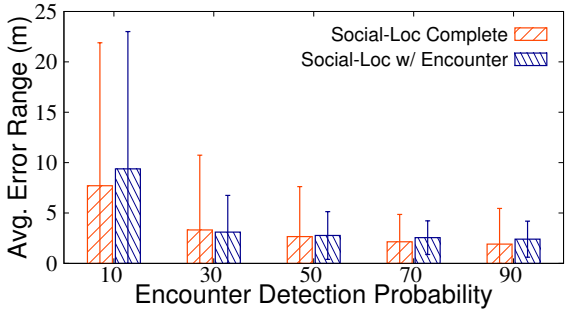


Figure 21: Encounter Detection Error vs. Social-Loc Localization Error

In Section 4.1.1, we implement accurate encounter detection using WiFi direct on smartphone. Normally encounter events are detected when they are geographically close to each other($<$ 3m). However, encounter detection errors may happen if WiFi Direct signals do not generate obvious peaks from two encountered users due to signal interference or multipath effect. In this simulation, we artificially drop some encounter events based on a specified encounter detection probability to study whether Social-Loc can tolerate any encounter detection errors. Figure 21 shows that Social-Loc is robust against encounter detection errors until encounter detection probability falls below 30%.

6.7 Scalability of Social-Loc

Social-Loc is a middleware designed to work in real-time with any practical indoor localization systems. Therefore, its computation time is another key requirement. Table 1 shows the calibration time of Social-Loc utilizing both encounters and non-encounters for a large number of users. The calibration time is the total computation time divided by simulated real-world time (3,000 secs.). The computer used for this test has Intel i7 3.4GHz CPU and 8G RAM. The statistic shows that the population density of Shibuya, one of the busiest shopping districts in Tokyo, is 13,540 persons per km^2, which is equivalent to 331 people in our simulation setup. Considering this real fact, 1554/3000 sec ($\approx$ 0.5 sec) computational time of Social-Loc to calibrate the potential locations of 300 users is reasonable for any practical location-aware applications.

In case of higher population densities where more than 300 users are in 144m x 144m, we can still handle these situations by subdividing the area into smaller grids and computing the Social-loc for each sub-area with users' potential locations in that sub-area only. This does not affect the performance of Social-Loc. The reason is when population is as high as 331 users dividing the area into 4 equal sectors would still contain an average of 82 users in each sector. We have shown in Figure 17 that 80 users in a 144m x 144m area is enough to keep the average localization accuracy of Social-Loc below 3m. Therefore, 83 users in 72m x 72m would not decrease the performance of Social-Loc.

Number of users	100	200	300
Calibration time (sec)	432/3000	890/3000	1554/3000

Table 1: This table shows computation time of Social-Loc

7. SECURITY AND PRIVACY ISSUES

Social-Loc is built on top of centralized indoor localization systems where a central server runs the localization system and produces potential locations. Social-Loc collects encounter information of all users and refines their potential locations. Many security schemes have been proposed for the centralized indoor localization systems such as [13].

Social-Loc is a middleware and applying it to underlying indoor localization systems neither reduces their security nor exposes the privacy of users. This is because Social-Loc does not explicitly require a user's geographical location information. The basic operations of Social-Loc is the set operations such as the intersection of potential location sets in Equation (1) and searching for n-TOSes in Equation (3). Therefore, an index of potential locations and their corresponding weight is sufficient for Social-Loc to function properly. As long as the underlying indoor localization system is secure, Social-Loc does not weaken its security and privacy of user.

8. RELATED WORK

Indoor localization in low-power mobile networks has been an active area of research due to its wide application domains [3, 19, 15]. Due to space constraints, we focus here on localization systems that are more relevant to Social-Loc.

Signal-fingerprint-based approaches work in two stages, calibration (offline phase) and localization (online phase). They first construct a signal fingerprint database from various sources, such as WiFi [36], sound [27], cellular [34] and FM [21] etc., for individual locations of an indoor environment. Then they match the observed signal to the most likely signal fingerprint in the database to estimate an object's location. Some notable solutions include Horus [36], Sensloc [16], SurroundSense [1], ABS [27] et al. Since interference always affects the observed signal strength [11], signal-fingerprint-based localization suffers from the instability in positioning accuracy, where we explore Social-Loc to improve the localization accuracy.

In Dead-reckoning, a user's current position is calculated by the previously determined position and various sensory inputs. NavShoe [8] is a representative inertial navigation system and achieves good performance with shoe-mounted sensors. FootSLAM [24] and [35] use foot-mounted inertial measurement unit (IMU) and particle filters to determine users' trajectories. SMART [39] combines signal fingerprinting with Dead-reckoning method. These Dead-reckoning-based schemes provide reasonable performance with an intrinsic assumption of fixed orientation of IMU, i.e., the performance is strongly dependent on the precision and mounted positions of inertial sensors. Therefore, they are not suitable for practical usage of pervasive mobile computing devices such as smartphones in the pocket. Under such conditions, Dead-reckoning suffers from poor estimation of user locations due to systematic biases and random noises. Consequently, it may generate multiple potential locations at every time instance. In Social-Loc, we apply social interactions to remove the impossible potential locations.

Recently, there have been several preliminary studies on utilizing neighborhood links to enhance position estimation. As far as we are aware, Social-Loc is the first work utilizing both user encounter and non-encounter information for enhancing accuracy of underlying indoor localizations. Beside Social-Loc's utilizing non-encounter information, there are several key differences in the way Social-Loc utilizes encounter information compared to existing works.

Encounter-based sensor tracking (EBT) [26] is a relative localization system utilizing users' encounter events. EBT first maps a user's trajectory and distance between two encountered users to a graph and finding an embedding of this graph in the plane.

Another tracking service utilizing user-intersections is Escort [7]. It uses mobile phone inertial sensors and opportunistic user-intersections to develop an electronic escort service. Escort maintains a single estimated track for each user and performs the inertial drift cancellation by using users' past intersections as routing points between any pair of persons in the vicinity. Different from both EBT and Escort, Social-Loc probabilistically updates the weight of potential locations of encountered users.

Similar to Social-Loc, Kloch et al. [17] utilizes user's encounter information in the indoor localization by updating their joint probability distribution. However, Social-Loc represents continuous location space as a set of discrete potential locations and updates their weights by utilizing both encounter and non-encounter information. This representation allows Social-Loc to be used as a transparent middleware for improving the accuracy of different underlying indoor localization schemes.

9. CONCLUSION

In this paper, we introduce Social-Loc, an indoor localization middleware that exploits social interactions among mobile users as virtual sensors. The key insight for Social-Loc to improve the localization accuracy is its capability to utilize both encounter and non-encounter information among mobile users. Based on the input from underlying indoor localization systems, Social-Loc further probabilistically increases the weights of more likely potential locations and decreases the weights of less likely potential locations by utilizing encounter events and Total Occupancy Sets (TOS) for non-encounter events.

We have fully implemented the Social-Loc on the Android platform with two reference indoor localization systems: particle-filter-based Dead-reckoning and WiFi fingerprinting. Then, we conduct extensive evaluations through both experiments and simulations. The results show that Social-Loc is able to significantly improve the accuracy of underlying indoor localization systems and achieves good accuracy.

Acknowledgments

We thank Pedro Jose Marron (our shepherd) and the other SenSys reviewers for their many useful comments. This work was supported by Singapore-MIT IDC IDD61000102a, IDG31100106a, NRF2012EWT-EIRP002-045, NSF CNS-1217791, BU Academic Program and Faculty Development Fund, NSFC61170296, and NECT-09-0028.

10. REFERENCES

[1] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *MobiCom '09*, 2009.

[2] N. Banerjee, S. Agarwal, P. Bahl, R. Chandra, A. Wolman, and M. Corner. Virtual compass: relative

positioning to sense mobile social interactions. In *Pervasive'10*, 2010.

[3] G. Borriello, M. Chalmers, A. LaMarca, and P. Nixon. Delivering real-world ubiquitous location systems. *Communications of the ACM*, 48(3), Mar. 2005.

[4] B. Buchli, F. Sutton, and J. Beutel. Gps-equipped wireless sensor network node for high-accuracy positioning applications. In *EWSN '12*, 2012.

[5] H.-l. Chang, J.-b. Tian, T.-T. Lai, H.-H. Chu, and P. Huang. Spinning beacons for precise indoor localization. In *SenSys '08*, 2008.

[6] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *MobiCom '10*, 2010.

[7] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. Did you see bob?: human localization using mobile phones. In *MobiCom '10*, 2010.

[8] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *Computer Graphics and Applications*, 25(6), nov.-dec. 2005.

[9] C. R. Gallistel and A. P. King. *Dead Reckoning in a Neural Network*. Wiley-Blackwell, 2009.

[10] S. Guha, K. Plarre, D. Lissner, S. Mitra, B. Krishna, P. Dutta, and S. Kumar. Autowitness: locating and tracking stolen property while tolerating gps and radio outages. In *SenSys '10*, 2010.

[11] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *MobiCom '04*, 2004.

[12] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *MobiCom '03*, 2003.

[13] J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *MobiSys '04*, 2004.

[14] X. Jiang, C.-J. M. Liang, K. Chen, B. Zhang, J. Hsu, J. Liu, B. Cao, and F. Zhao. Design and evaluation of a wireless magnetic-based proximity detection platform for indoor applications. In *IPSN '12*, 2012.

[15] D. H. Kim, K. Han, and D. Estrin. Employing user feedback for semantic location services. In *UbiComp '11*, 2011.

[16] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *SenSys '10*, 2010.

[17] K. Kloch, P. Lukowicz, and C. Fischer. Collaborative pdr localisation with mobile phones. In *ISWC'11*, 2011.

[18] A. E. Kosba and M. Youssef. RASID: A robust wlan device-free passive motion detection system. In *PerCom '12*, 2012.

[19] Y. Lee, S. Lee, B. Kim, J. Kim, Y. Rhee, and J. Song. Scalable activity-travel pattern monitoring framework for large-scale city environment. *IEEE Transactions on Mobile Computing*, 11(4):644–662, 2012.

[20] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *SenSys'08*, 2008.

[21] A. Popleteev, V. Osmani, and O. Mayora. nvestigation of indoor localization with ambient fm radio stations. In *PerCom '12*, 2012.

[22] J. Qiu, D. Chu, X. Meng, and T. Moscibroda. On the feasibility of real-time phone-to-phone 3d localization. In *SenSys '11*, 2011.

[23] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: zero-effort crowdsourcing for indoor localization. In *Proc. MobiCom ï£¡ï£¡12*, 2012.

[24] P. Robertson, M. Angermann, and B. Krach. Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors. In *Ubicomp '09*, 2009.

[25] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. You are facing the mona lisa: spot localization using phy layer information. In *MobiSys '12*, pages 183–196, 2012.

[26] A. Symington and N. Trigoni. Encounter based sensor tracking. In *MobiHoc '12*, 2012.

[27] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *MobiSys '11*, 2011.

[28] A. Thiagarajan. *Probabilistic Models For Mobile Phone Trajectory Estimation*. PhD thesis, Massachusetts Institute of Technology, 2011.

[29] A. W. T. Tsui, W.-C. Lin, W.-J. Chen, P. Huang, and H.-H. Chu. Accuracy performance analysis between war driving and war walking in metropolitan wi-fi localization. *Transactions on Mobile Computing*, 9(11), Nov. 2010.

[30] S. Wagner, M. Handte, M. Zuniga, and P. Marron. On optimal tag placement for indoor localization. In *PerCom '12*, 2012.

[31] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: unsupervised indoor localization. In *MobiSys '12*, 2012.

[32] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *Trans. Inf. Syst.*, 10(1), Jan. 1992.

[33] K. Whitehouse and D. Culler. A robustness analysis of multi-hop ranging-based localization approximations. In *IPSN '06*, 2006.

[34] T. Wigren. Adaptive enhanced cell-id fingerprinting localization by clustering of precise position measurements. *Transactions on Vehicular Technology*, 56(5), Sept. 2007.

[35] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In *Ubicomp '08*, 2008.

[36] M. Youssef and A. Agrawala. The horus wlan location determination system. In *MobiSys '05*, 2005.

[37] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala. Pinpoint: An asynchronous time-based location determination system. In *MobiSys '06*, 2006.

[38] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda. Swordfight: Enabling a new class of phone-to-phone action games on commodity phones. In *MobiSys'12*, 2012.

[39] P. Zhuang, D. Wang, and Y. Shang. Smart: Simultaneous indoor localization and map construction using smartphones. In *IJCNN'10*, 2010.