

MoLoc: On Distinguishing Fingerprint Twins

Wei Sun*, Junliang Liu*, Chenshu Wu†, Zheng Yang†*, Xinglin Zhang* and Yunhao Liu†*

*Hong Kong University of Science and Technology

†School of Software, TNLi, Tsinghua University

Email: {sunw1989, talentjl, wucs32, hmilyyz, zhxlinse, yunhaoliu}@gmail.com

Abstract—Indoor localization has enabled a great number of mobile and pervasive applications, attracting attentions from researchers worldwide. Most of current solutions rely on Received Signal Strength (RSS) of wireless signals as location fingerprint, to discriminate locations of interest. Fingerprint uniqueness with respect to locations is a basic requirement in these fingerprinting-based solutions. However, due to insufficient number of signal sources, temporal variations of wireless signals, and rich multipath effects, such requirement is not always met in complex indoor environments, which we refer to as *fingerprint ambiguity*. In this work, we explore the potential of leveraging user motion against fingerprint ambiguity. Our basic idea is that user motion patterns collected by built-in sensors of mobile phones add to the diversity built by RSS fingerprints. On this basis, we propose MoLoc, a motion-assisted localization scheme implemented on mobile phones. MoLoc can easily be integrated in existing localization systems by simply adding a motion database that is constructed automatically by crowdsourcing. We conducted experiments in a large office hall. Experiment results show that MoLoc doubles the localization accuracy achieved by the fingerprinting method, and limits the mean localization error to less than 1m.

Index Terms—Indoor Localization; User Motion; RSS Fingerprint; Crowdsourcing

I. INTRODUCTION

Context awareness has become an essential part in many pervasive computing applications, where location [1] acts as a principal factor. Provided with reliable and accurate location information, users are able to navigate in shopping malls [2], search events of interest in the vicinity [3], and experience various ways of social participation [4] and entertainment [5].

While outdoor localization solutions are mainly dominated by GPS, indoor localization remains a great challenge due to the limited visibility of GPS satellites. Early work relies on pre-installed infrastructure or specialized hardware such as ultrasound devices [6] and RFID [7], making them neither scalable nor universal. Therefore, researchers have diverted their attention to widely available radio frequency (RF) signals (*e.g.*, WiFi [8], GSM [9], FM [10]) and popular off-the-shelf devices (*e.g.*, mobile phones [11]). Under this context, fingerprinting-based approaches are the most prevalent among the localization techniques used. They typically consist of two phases: site survey and localization. During the site survey, each reference location is profiled by the Received Signal Strength (RSS) from signal sources as its fingerprint (*e.g.*, WiFi signal strengths from multiple Access Points, APs). A database of location-fingerprint mappings is thereafter con-

structed. In the localization phase, when a user sends a location query with her current RSS fingerprint, localization algorithms retrieve a matched location from the fingerprint database and return it to the user.

A basic requirement to ensure accuracy for fingerprinting-based approaches is to obtain *discriminative* fingerprints among all reference locations. That is, dissimilarities among the fingerprints in the database should be pairwise large enough so that no mismatch would occur. However, meeting such requirement is nontrivial due to insufficient number of signal sources, temporal variations of wireless signals, and rich multipath effects indoors caused by walls, furniture, and human. As a result, two distinct locations may possess similar fingerprints, making them difficult to distinguish. We regard this problem as *fingerprint ambiguity*. Fingerprint ambiguity may lead to degradation of localization performance, since a user may be mistakenly recognized at another location that matches her current RSS fingerprint. In particular, it is in fact the root cause of large errors in WiFi localization, as reported in [12]. On the other hand, it is intractable as we are unable to eliminate the uncertainty and instability of signal propagation in the air. Previous proposals fail to identify or devote little effort to addressing fingerprint ambiguity, leaving it an unsolved problem.

To tackle this, we propose MoLoc, a MOTion-assisted indoor LOCALization scheme implemented on off-the-shelf mobile phones. The intuition behind MoLoc is simple: user motion patterns collected by built-in sensors of mobile phones add to the diversity built by RSS fingerprints. By exploiting relative location measurement between each pair of adjacent locations, we may be able to achieve adequate diversity and thus distinguish among multiple locations with similar fingerprints. On this basis, MoLoc works as follows. A user queries the fingerprint database with her current RSS fingerprint collected by her mobile phone. The system obtains a set of location candidates by calculating dissimilarities between her fingerprint and those in the database. Each candidate is evaluated with the assistance of a set of former candidates, current motion measurements, and a motion database storing relative location measurements between adjacent locations. The evaluation process ranks these candidates and returns the top one as the current location estimate.

Despite the simple idea, two challenges underlie the design of MoLoc. (1) How to effectively construct the motion

database? Manual configuration requires a great amount of time and effort, while computing by location coordinates may result in unreliable measurements without regard to walls or obstacles invisible from the floor plan. Instead, we adopt crowdsourcing, which has been proved to be effective and efficient in constructing fingerprint databases [13], [14]. As the data collected from users are too noisy to be directly used, we develop several data sanitation techniques to purify them. (2) How to evaluate location candidates combining user motion and the motion database? A simple way is to sum up the dissimilarities aggregated from RSS fingerprints, direction, and offset measurements. However, this could result in biased evaluation, that is, the measurement with wider range gets more important. To eliminate such bias, we design a probabilistic algorithm taking advantage of the mutual independence between RSS fingerprints and user motion.

Three main advantages make MoLoc a promising solution to indoor localization. First, experiment results show that it greatly reduces the large errors caused by fingerprint ambiguity, and hence improves localization accuracy. Second, MoLoc accelerates the convergence of accurate localization. Even if the initial location estimate is inaccurate, a user only needs to walk pass few reference locations before the first accurate localization, after which the average localization accuracy is over 90% (see Section VI). Third, MoLoc is highly compatible with existing fingerprinting-based localization systems. The only extra effort is to construct a motion database, which is crowdsourced and automatic. Note that MoLoc is not meant to bear down preceding proposals but to shed light on a train of thought augmenting localization accuracy with user motion.

Overall, we summarize our main contributions as follows:

- *We identify the problem of fingerprint ambiguity and explore the potential to resolve it by leveraging user motion.* We observe that user motion can act as a ubiquitous feature to distinguish locations with similar RSS fingerprints. Built-in sensors in mobile phones, such as digital compass and accelerometer, make user motion available for indoor localization.
- *We propose MoLoc, a motion-assisted indoor localization scheme implemented on off-the-shelf mobile phones.* It combines user motion patterns with RSS fingerprints to address fingerprint ambiguity. We adopt a crowdsourcing approach to construct a motion database and design a probabilistic algorithm to evaluate location candidates. To our knowledge, MoLoc serves as an early attempt at exploiting user motion in the localization phase, while that in the site survey phase has been well studied.
- *We implement a prototype of MoLoc and deploy it in an office hall covering over 650m².* The motion database achieves small errors in both direction and offset measurements, demonstrating its validity. The localization algorithm doubles the localization accuracy achieved by the fingerprinting method, and limits the mean localization error to less than 1m.

The rest of this paper is organized as follows. Section II describes the related work inspiring us. We present the potential gain from user motion for localization and the system architecture of MoLoc in Section III. The motion database construction and the motion-assisted localization algorithm are detailed in Section IV and V, respectively. Section VI gives the system performance. Finally, we conclude our work in Section VII.

II. RELATED WORK

Indoor localization has been extensively studied in the literature of pervasive and mobile computing. A variety of approaches have been proposed, which can be roughly classified as those with infrastructure, RSS fingerprinting and modeling, and multi-modal sensors. We review the state-of-the-art work and discriminate MoLoc from them in this section.

Localization with infrastructure. Early attempts enable indoor localization by specialized infrastructure. For example, Active Badge [15] relies on infrared beacons. Cricket [6] achieves high accuracy using ultrasound devices installed on ceilings. LANDMARC [7] attaches active RFID tags at known locations as landmarks. PinPoint [16] works with a kind of specialized hardware, which they refer to as PP2. Despite high accuracy, these systems inevitably incur high hardware cost or deployment overhead, which limits their ubiquitous popularity.

Localization with RSS fingerprinting and modeling. Recognizing the drawbacks of infrastructure dependence, researchers have devoted substantial effort to exploiting widely available radio such as WiFi [8], GSM [9] and FM [10]. Under this context, the localization techniques can be well divided into two categories: RSS fingerprinting and modeling. On one hand, RSS fingerprinting is to profile each reference location with its RSS from all APs as the fingerprint (known as the site survey), and to localize users based on user-collected fingerprints. There has been a plethora of research work adopting this idea, such as RADAR [8], Horus [17], and PlaceLab [18]. Several recent proposals, including WILL [19], LiFS [14] and Zee [13], aim at minimizing the site survey effort. On the other hand, RSS modeling estimates a user's location with various models, *e.g.*, RF propagation models [20], [21] to estimate distance away from an AP or Bayesian probabilistic models [22] to capture the relationships between different nodes in the network. In general, RSS fingerprinting methods assume that fingerprints are dissimilar among reference locations, while RSS modeling methods assume that the models reflect the truth. However, both assumptions are difficult to hold ideally.

Localization with multi-modal sensors. We have witnessed a popular trend in incorporating data from multi-modal sensors into localization. For example, SurroundSense [11] performs logical location estimation based on WiFi and ambient features including light, color, and sound, but it may require time-consuming and labor-intensive fingerprinting process for these features. Liu *et al.* [23] exploit accelerometer-assisted localization based on a hidden Markov model (HMM).

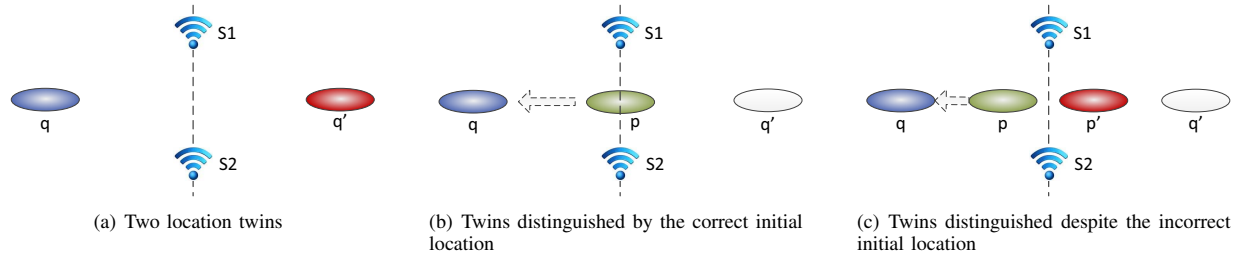


Fig. 1. Potential gain against fingerprint ambiguity by user motion.

We argue that this system is prone to initial localization error intrinsic to HMM, and the high computational overhead may drain off the battery quickly. Most recently, Liu *et al.* [12] investigate the performance of WiFi localization for smartphones, and propose a peer-assisted algorithm leveraging physical constraints imposed by neighboring peers. One flaw in their work is that peer involvement is sometimes neither available nor desirable.

MoLoc leverages the idea of using multi-modal sensors, but different from the preceding work, it is an infrastructure-free, efficient, and self-contained solution to indoor localization for mobile phone users. MoLoc is targeted at mitigating fingerprint ambiguity and can be built atop existing fingerprinting-based localization systems, regardless of fingerprint types. Due to the prevalence of WiFi, in this paper we narrow down our discussion to WiFi only.

III. MOLOC OVERVIEW

To ensure high localization accuracy and achieve small errors, fingerprinting-based localization systems have to respond to *fingerprint ambiguity* where two distinct or even highly spaced locations have similar RSS fingerprints, as observed by [12]. Instead of using acoustic ranging, we build our system named *MoLoc* on top of various built-in sensors of mobile phones. In particular, relative direction and offset between adjacent locations, measured by digital compass and accelerometer respectively, add to the fingerprint diversity to differentiate RSS-similar locations. In this section we start with an illustration of potential gain from user motion against fingerprint ambiguity, followed by the system architecture.

A. Potential Gain from User Motion

We walk through a simple example to illustrate fingerprint ambiguity and to study how user motion contributes to resolving it. Consider the simple setting in Figure 1(a). It is a relative open space with two APs (*i.e.*, S_1 and S_2) placed away from each other. Two locations q and q' are symmetric with respect to the line connecting S_1 and S_2 . We assume RSS fingerprints at q and q' are both collected during the site survey process. As these two locations are in similar conditions relative to both APs (*e.g.*, same distance), their corresponding RSS fingerprints may be quite alike. Now suppose a user is at q and inquires for

her location by her RSS fingerprint. Due to the close similarity of the RSS fingerprints between q and q' , it is almost equally possible to estimate her location at either one, making them undistinguishable by simply RSS fingerprints.

Next we consider the scenario in Figure 1(b) when combined with user motion. Suppose a user was previously at p (marked with green), which is on the line S_1S_2 . She then moved to left and arrived at q (marked with blue). Due to p 's uniqueness in distances to both APs, we suppose she obtained a correct initial location, *i.e.*, she was initially localized at p . This initial location estimate, together with her motion information that her moved to the left, leads to a much higher possibility that she is now at q rather than at q' . By this way q and q' can be distinguished with the ground truth location more probable to be the location estimate.

Though such motion-assisted localization seems to be effective on first impression, a natural problem arises – what if a user has never been to such a unique location or has an incorrect initial location estimate? We show by Figure 1(c) that even in this case, user motion can still help pilot the localization to correct estimation if we maintain a set of former location candidates. Suppose a user was initially at p , then moved left and arrived at q . At first a set of location candidates composed of p (marked with green) and its mirrored location p' (marked with red) were obtained by the initial RSS fingerprint. Typically they are of almost equal probability to be initial location estimate but somehow p' was returned, resulting in an incorrect estimation. Now a new set of location candidates containing q and q' are obtained by the new RSS fingerprint. On one hand, although q' is much closer to p' , it is to the right of both p and p' (previous candidates), the opposite direction of her motion. This makes it less possible to be the current location estimate. On the other, her motion highly matches the direction and distance between p and q , suggesting that q is more eligible for the current estimate. Overall, q may be selected out in spite of inaccurate initial location estimation.

The above simple example shows that *user motion has potential benefits for resolving fingerprint ambiguity, thus improving localization accuracy*. Note that the scenario we discussed is too ideal to be true in real-world applications. The number of APs is usually more than two in general indoor environments. Temporal variations of WiFi signal and envi-

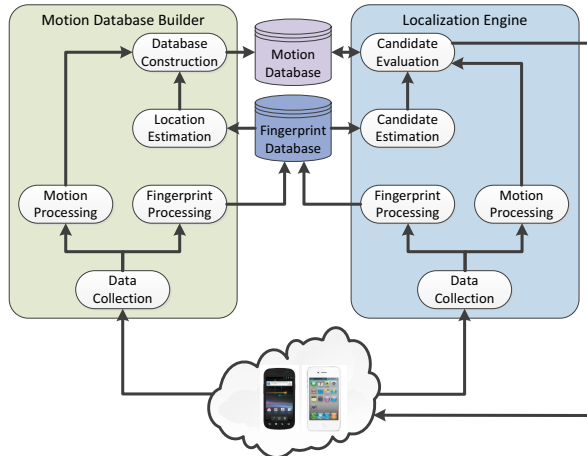


Fig. 2. MoLoc architecture.

ronmental dynamics such as crowd activity further complicate the scenario in question. But still, the potential gain brought by user motion for localization has greatly stimulated us to examine in deep and finally led to MoLoc. An experimental study of MoLoc and how user motion improves localization in real-world applications is presented in Section VI.

B. System Architecture

Inspired by the potential gain from user motion for resolving fingerprint ambiguity, we design MoLoc, a MOTion-assisted LOCALization system implemented on off-the-shelf mobile phones. Figure 2 presents the overall architecture of MoLoc. Its working process is divided into three phrases: *fingerprint database construction*, *motion database construction*, and *motion-assisted localization*. We give brief descriptions of these three phrases and present their details in the subsequent sections.

Fingerprint database construction is also known as the site survey process for fingerprinting-based localization. In general, RSS of WiFi signals from multiple APs is recorded as the fingerprint at each reference location. All these fingerprint-location mappings form together the fingerprint database. Traditionally, the site survey process is done by engineers, time-consuming and labor-intensive. Several proposals have been made to reduce the time and effort in the site survey. For example, EZ [20] is among the earliest that require no site survey. More recently, WILL [19], LiFS [14] and Zee [13] are proposed to construct the fingerprint database by crowdsourcing. As this is not the main contribution of MoLoc and ample studies have been made on it, we skip this part in subsequent sections. In our current implementation we adopt traditional methods, and leave the newly proposed methods for future investigation.

Motion database construction is a process that obtains relative location measurement between each pair of adjacent

locations. To achieve efficiency and consistency, we opt to adopt crowdsourcing to train the motion database. Specifically, a user’s mobile phone periodically samples RSS fingerprints from all APs, and built-in sensors, including digital compass and accelerometer, measure the direction and acceleration readings. The fingerprint processing unit sends a fingerprint to the fingerprint database, which returns a matched location estimate, as fingerprinting-based localization does. In addition, the motion processing unit extracts the direction and offset measurements from the sensor readings, and feeds them to the database construction unit together with the location estimate. The database construction unit verifies the correctness of these data and filters out outliers. Finally the processed relative location measurements are stored in the motion database. As we will see in Section V, the validity of motion database determines how much gain we can get for motion-assisted localization. The motion database construction will be elaborated in Section IV.

Motion-assisted localization is to localize a user by her RSS fingerprint and motion information. Similar to the motion database construction, RSS fingerprints are periodically sampled and user motion is recorded by digital compass and accelerometer. However, rather than return a rough location estimate, the fingerprint database yields a set of *location candidates* stored by a candidate estimation unit. These candidates with fingerprint dissimilarities, direction and offset measurements, are fed to a candidate evaluation unit, which evaluates the probability of each candidate to be the current location estimate. The candidate with the highest probability is returned back to the user, and all these candidates are retained for localization next time. The details of the localization algorithm assisted by user motion will be presented in Section V.

The key novelty of MoLoc design is *efficiently integrating user motion into localization algorithm*, so as to enhance localization performance. Though WILL [19], LiFS [14] and Zee [13] have demonstrated the effectiveness of user motion in reducing effort of site survey, few work explores the opportunity to leverage user motion in localization. MoLoc serves as an early attempt to this open field. Besides, considering the prevalence of fingerprinting-based solutions to indoor localization, most of current localization systems can easily be upgraded to the one similar to MoLoc. Finally, to achieve energy efficiency, we make a compromise on the delicacy of the localization algorithm but still provide a significant gain over traditional fingerprinting methods. An evaluation of MoLoc will be shown in Section VI.

IV. MOTION DATABASE CONSTRUCTION

We present in this section the way how the motion database is constructed. Two principles are proposed as the criteria for assessing different construction methods, among which the crowdsourcing way outperforms others. An detailed description of the construction process by crowdsourcing, including

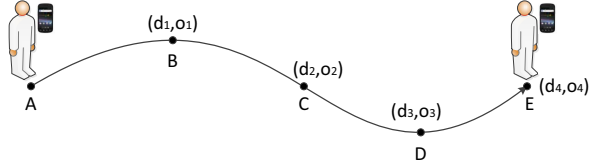


Fig. 3. Collecting direction and offset measurements between adjacent locations.

data collection and data sanitation, will be introduced in the following. Finally we will show the database structure.

A. Principles and Methods

We define the relative location measurement (RLM) between two adjacent locations as the *direction* and *offset* on walkable paths. RLM of each pair of adjacent locations is stored in a motion database. Here we state two principles we should follow when constructing the motion database. First, *the principle of efficiency*: the construction should be as efficient as possible so that it does not require too much time or human effort. A great deal of time consumption and labor work will no doubt hinder the deployment of such systems. Second, *the principle of consistency*: the RLMs in the motion database should reflect real patterns of user motion as much as possible. For example, straight-line distance between two locations may not necessarily be used as their corresponding offset, since the direct path may be unavailable because of obstacles such as walls, doors and furniture.

Bearing these two principles in mind, we start to discuss possible approaches to the motion database construction. An intuitive way is by manual configuration: engineers measure RLMs between each location and its adjacent ones, and store them into the motion database. While this approach achieves high accuracy, it is inevitably time-consuming and labor-intensive, violating the principle of efficiency. An alternative to this may be calculating RLMs given a map with the coordinates of all reference locations. This can be done by computer programs and hence significantly reduces time and human effort. However, it may mistakenly view as adjacent two locations, which are geographically close but in fact separated by a wall or obstacles invisible in the map. In this case, the principle of consistency gets infringed. Instead, we note the popularity of dealing with tough problems by crowdsourcing in both industry (e.g., Amazon Mechanical Turk [24]) and academia (e.g., LiFS [14] and Zee [13]). In the context of motion database construction, crowdsourcing not only requires little human effort, but also provides measurements from walkable paths. Therefore, we opt to construct the motion database by crowdsourcing.

B. Crowdsourcing for User Motion

Suppose a user with a mobile phone randomly walks in an indoor environment where the fingerprint database is

already available. Her phone periodically samples RSS fingerprints, and obtains corresponding location estimates by sending queries with fingerprints to the fingerprint database. Meanwhile, built-in sensors in her mobile phone, namely digital compass and accelerometer, record during each localization interval the direction and acceleration readings used to compute RLMs of adjacent locations. An illustration of this scenario is shown in Fig. 3. She moves from location A to E passing B , C and D in succession. After the first localization interval, the RSS fingerprint collected *might* correctly localize herself at B . The RLM during this interval, denoted as (d_1, o_1) , are also calculated from sensor readings. This process works for subsequent intervals. Altogether, these location estimates with corresponding RLMs are applied to constructing the motion database that presents RLMs of adjacent locations. Note that the validity of the motion database relies on two assumptions: (1) the location estimates are correct, and (2) the RLMs are accurate. However, as exemplified in Section III and measured in [12], localization by RSS fingerprints may yield large errors due to fingerprint ambiguity. Moreover, the RLMs measured are usually inaccurate because of biased and noisy sensors. We first elaborate how to obtain location estimates and RLMs, and then show how to address the case where these assumptions do not hold.

1) *Data Collection*: We represent an RLM between two adjacent locations i and j as an ordered pair $r_{i,j} = \langle d, o \rangle$, where i indicates the starting location, j the ending location, d and o the direction and offset measurement, respectively.

Location Estimate. This is done by the location estimate unit. The starting location i and ending location j are both estimated by RSS fingerprints. Suppose the number of APs is n . Then an RSS fingerprint of WiFi signal can be represented as

$$F = (f_1, f_2, \dots, f_n)$$

where f_i denotes the RSS value from the i th AP. To estimate a user's location by her RSS fingerprint, we adopt Euclidean distance to measure the *dissimilarity* between two fingerprints, similar to [19]. Formally, given two fingerprints $F = (f_1, f_2, \dots, f_n)$ and $F' = (f'_1, f'_2, \dots, f'_n)$, the dissimilarity $\phi(F, F')$ between them is given by

$$\phi^2(F, F') = \sum_{i=1}^n (f_i - f'_i)^2 \quad (1)$$

We calculate the dissimilarities between the fingerprint collected by a user and each one in the fingerprint database. The location whose corresponding fingerprint achieves the least dissimilarity is returned as her estimated location. Put it mathematically, we define $\psi(D)$ as the set of all fingerprints in the fingerprint database D and $\varphi(F)$ the corresponding location of the fingerprint F . Then the location estimate l with respect to the user fingerprint F is given by

$$l(F) = \varphi \left(\arg \min_{F' \in \psi(D)} \phi(F, F') \right) \quad (2)$$

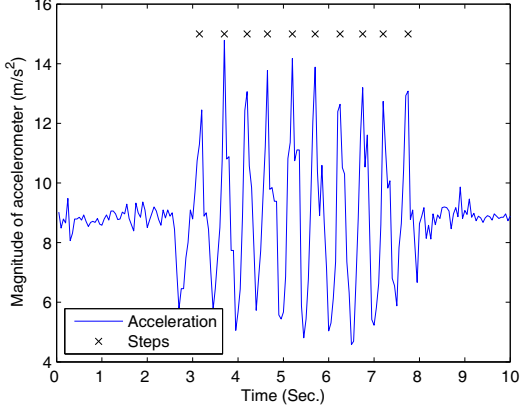


Fig. 4. Acceleration signatures of 10 steps (each marked with a cross).

Direction. This is one of the two functions provided by the motion processing unit. The direction d of user motion is measured by digital compass integrated in mobile phones. Note that readings from digital compass do not necessarily indicate motion directions. In fact, they just reflect the degree of angle between phone orientation and magnetic north. With a typical handheld style when a user is surfing on the Internet, text messaging or microblogging, compass readings may roughly accord with motion directions. However, they may differ significantly from or even be opposite to motion directions when she is making a call, playing games or watching videos. Here we take credits from Zee [13] that achieves placement-independent orientation estimation. Specifically, the motion direction can be obtained by estimating the heading offset between the compass reading and the motion direction, regardless of phone placement.

Offset. This is another function of the motion processing unit. In order to estimate the offset o , we exploit the repetitive walking patterns [13], [14] displayed by acceleration readings from accelerometer. Figure 4 shows the acceleration readings when a user is walking 10 steps. Apparently a repetitive pattern can be observed and identified. The distance traversed during this period can be calculated as the product of step count and step size. Existing works relying on step counting for offset estimation use integral steps. We refer to it as *Discrete Step Counting (DSC)*. DSC may miss out the motion during the odd time, *i.e.*, the time before the first recognized step and after the last one. This may result in one or two steps unaccounted, intolerable as a localization interval (*e.g.*, 3s) generally contains a few steps. Therefore, we explore to apply *Continuous Step Counting (CSC)* by leveraging relatively stable period of walking patterns. Put it concretely, we first detect whether a user is walking throughout an interval of time. If she is, we detect each step within the interval. The period of her walking pattern is calculated by dividing the time covering all the detected steps by the integral step count. Thus the odd time equals to the difference between the whole

interval and the time covering all steps. Finally, we divide the odd time by the period and get what we call *decimal steps*. The decimal and integral steps, together with the step size derived from individuals height and weight [25], estimate the offset of the user motion.

2) *Data Sanitation:* Given the RLMs crowdsourced by users, we now consider how to use them efficiently to construct the motion database and to address the error and noise caused by inaccurate location estimation and biased noisy sensors. This is accomplished by the database construction unit.

Data reassembling. We first assume *mutual reachability* between any two adjacent locations, *i.e.*, if a user can walk from location i to its adjacent location j , then vice versa by the reverse direction and the same offset. This assumption usually holds in indoor environments. Under this assumption, we reassemble all RLMs with the location of smaller ID as the start. In other words, for an RLM $r_{i,j} = \langle d, o \rangle$, if $i.ID > j.ID$, we replace it with its mirror RLM $r_{j,i} = \langle d', o \rangle$ where

$$d' = d + 180^\circ \pmod{360^\circ}.$$

These reassembled RLMs are fed to the data filter introduced in the following. In such a manner, we use an RLM in two ways (forward and backward directions) to shorten the time for training the motion database.

Data filtering. The location estimation may be erroneous due to fingerprint ambiguity. Moreover, biased and noisy sensors (*i.e.*, digital compass and accelerometer) contribute to unreliable RLMs as well. Therefore, outliers need to be filtered out before the RLMs collected are put into use. We observe from a preliminary experiment that the direction and offset measurements calculated by digital compass and accelerometer do not vary too much from the ones by the map. Typically the errors are bounded by 20° in direction and $3m$ in offset. Furthermore, the distribution of the direction and offset measurements can be well fit to Gaussian distributions as similar to [25]. Therefore, we get rid of outliers in two levels of granularity. In a coarse-grained manner, we compare each RLM between location i and j ($i \leq j$) with that calculated by their corresponding coordinates. Those RLMs are discarded if the corresponding differences are beyond certain threshold values (*e.g.*, 20° in direction, $3m$ in offset). In a fine-grained manner, we model the remaining direction and offset measurements as Gaussian distributions determined by their means $\mu_{i,j}^d, \mu_{i,j}^o$, and standard deviations $\sigma_{i,j}^d, \sigma_{i,j}^o$, respectively. We filter out those measurements if the differences compared with their means are larger than certain thresholds (*e.g.*, two times of the standard deviations). To reflect the direction and offset in reverse order (from j to i), we set

$$\begin{aligned} \mu_{j,i}^d &= \mu_{i,j}^d + 180^\circ \pmod{360^\circ}, \\ \mu_{j,i}^o &= \mu_{i,j}^o, \\ \sigma_{j,i}^d &= \sigma_{i,j}^d, \\ \sigma_{j,i}^o &= \sigma_{i,j}^o, \end{aligned}$$

according to the mutual reachability. Note that there might be more accurate ways to obtain RLMs. For example, we may achieve highly accurate direction estimation by using gyroscope and advanced filtering techniques such as the Kalman filter. However, it may suffice to validate RLMs through the method discussed above, as reported in Section VI. We leave this opportunity for investigation in the future.

C. Database Structure

We organize the motion database as an $n \times n$ matrix M , where n is the number of the reference locations. For each entry $M_{i,j}$, it stores a quadruple of the means and standard deviations of direction and offset between location i and j , i.e., $(\mu_{i,j}^d, \sigma_{i,j}^d, \mu_{i,j}^o, \sigma_{i,j}^o)$.

V. MOTION-ASSISTED LOCALIZATION

After the motion database construction is done, MoLoc turns into its serving stage. When a user initiates a localization request, her mobile phone generates RSS fingerprints and the built-in sensors in her mobile phone, including digital compass and accelerometer, start to record her motion information. When the fingerprint database receives the location query, it yields a set of location candidates based on the dissimilarities between her current fingerprint and those in the fingerprint database. The candidate with the least dissimilarity is returned as the initial location estimate. During the next localization interval (e.g., 3s), she may move to another location. A new RSS fingerprint is obtained and the direction and distance measurements during this interval are recorded. Likewise, the new RSS fingerprint generates a new set of location candidates. However, rather than return the candidate with the least dissimilarity as the new location estimate, MoLoc evaluates the possibility of each candidate being the ground truth by *considering the reachability from past location candidates through such direction and offset measurements*. The candidate with the highest probability is returned as the new location estimate, and all the candidates are retained for localization in the future. We next present in details how location candidates are calculated and user motion is applied to assisting location estimation.

A. Fingerprint Matching

The candidate estimation unit retrieves location candidates from the fingerprint database. Similar to the location estimation in Section IV, the location candidates are selected based on fingerprint dissimilarities. But rather than return only one location, a set L of k location candidates with corresponding probabilities are obtained. The selection of the location candidates and the computation of corresponding probabilities goes as follows. First, the corresponding fingerprints of the location candidates are k -nearest neighbors to the user-collected fingerprint. Specifically, let F be the RSS fingerprint in question, x the location estimate, and $L = \{l_1, l_2, \dots, l_k\}$. For each i

($1 \leq i \leq k$), l_i must satisfy the following condition

$$\sum_{F' \in \psi(D)} I(\phi(F, F') \leq \phi(F, \varphi^{-1}(l_i))) < k \quad (3)$$

where $I(e)$ is an indicator function returning 1 when its parameter e is true. We abbreviate $\phi(F, \varphi^{-1}(l_i))$ as m_i for subsequent use. Second, we observe that the value of m_i reflects to some extent the probability of location l_i being the location estimate: smaller dissimilarity suggests higher probability. Therefore, we empirically [8] calculate the probability of each candidate l_i given the fingerprint F as

$$P(x = l_i | F) = \frac{1/m_i}{\sum_{j=1}^k 1/m_j} \quad (4)$$

B. Motion Matching

With the motion database, we are able to evaluate the possibility that a user walks from one location to another through certain direction and offset. Given direction d and offset o , we define $P_{i,j}(d, o)$ as the probability that a user can walk from location i to j through such direction d and offset o . As d and o are independent from each other, we adopt a similar technique used in [26] and measure $P_{i,j}(d, o)$ as

$$P_{i,j}(d, o) = D_{i,j}(d)O_{i,j}(o) \quad (5)$$

where

$$D_{i,j}(d) = \int_{d-\alpha/2}^{d+\alpha/2} \frac{\exp\left\{-\frac{(x - \mu_{i,j}^d)^2}{2(\sigma_{i,j}^d)^2}\right\}}{\sigma_{i,j}^d \sqrt{2\pi}} dx,$$

$$O_{i,j}(o) = \int_{o-\beta/2}^{o+\beta/2} \frac{\exp\left\{-\frac{(x - \mu_{i,j}^o)^2}{2(\sigma_{i,j}^o)^2}\right\}}{\sigma_{i,j}^o \sqrt{2\pi}} dx.$$

$D_{i,j}(d)$ is a discretization of the Gaussian distributions of the direction between two adjacent locations i and j , while $O_{i,j}(o)$ is that of the offset. α and β are two parameters that determine the discretization interval.

The motion matching can be further extended to the case when we consider a set of possible starting locations. Let x denote the starting location and $S = \{i_1, i_2, \dots, i_k\}$ be the candidate set from which x can take values. Given direction d and offset o , the probability $P_{S,j}(d, o)$ that a user can walk from the locations in S to j through such direction d and offset o is calculated as

$$P_{S,j}(d, o) = \sum_{i_k \in S} P(x = i_k) P_{i_k,j}(d, o) \quad (6)$$

C. Localization Algorithm

Given the fingerprinting matching and the motion matching, we come to our motion-assisted localization algorithm. To refrain from draining off battery power, we try to develop an efficient algorithm that minimizes the computational complexity so as to save energy. Here the main idea is to *consider both the fingerprint matching and the motion matching by exploiting their independence*. Specifically, let

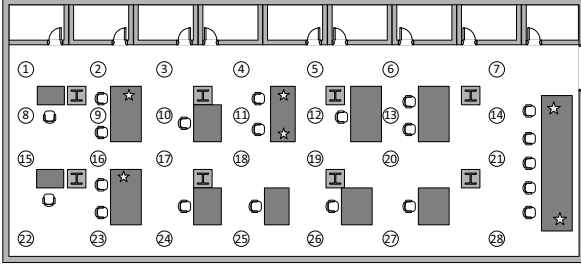


Fig. 5. Floor plan of the office hall.

$L = \{j_1, j_2, \dots, j_k\}$ be the set of current location candidates and $L' = \{i_1, i_2, \dots, i_k\}$ the one obtained from the last localization. Given the current RSS fingerprint F , the direction d and offset o during the localization interval, we calculate the probability $P(x = j_m | L', F, d, o)$ that the user is at each candidate location j_m as

$$P(x = j_m | L', F, d, o) = \frac{P(x = j_m | F)P(x = j_m | L', d, o)}{N} = \frac{P(x = j_m | F)P_{L', j_m}(d, o)}{N} \quad (7)$$

where N is a normalizer such that

$$\sum_{m=1}^k P(x = j_m | F, d, o) = 1.$$

With the probability computed for each of the current candidate locations, MoLoc returns the one with the highest probability as the location estimate. The newly obtained candidates with corresponding probabilities are retained for localization in future.

In summary, with RSS fingerprints, direction and offset measurements, MoLoc maintains a set of candidate locations by measuring how RSS fingerprints match the fingerprint database and how user motion matches the motion database. MoLoc is an infrastructure-free solution to indoor localization. It achieves high efficiency for its low computational overhead. It is also self-contained as it requires no interaction with other users. Finally, most of existing localization systems can easily be upgraded to the ones similar to MoLoc to improve localization performance. An evaluation of MoLoc in real-world experiments will be presented in the next section.

VI. EVALUATION

We aim at answering several questions in this section: (1) Is the motion database constructed by crowdsourcing valid, in the sense of reflecting the measurements from real walkable paths? (2) How much can MoLoc improve localization accuracy over WiFi fingerprinting? (3) What are the reasons behind the localization improvement? We describe the experiment settings first, followed by the performance evaluation.

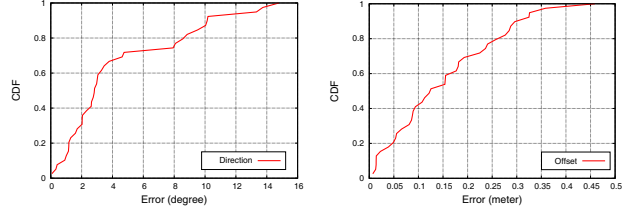


Fig. 6. Errors in the motion database

A. Experiment Settings

We implemented a prototype of MoLoc on a Google Nexus S phone, which is Android OS-based and integrated with digital compass and accelerometer. The phone performed a full scanning of WiFi signals twice per second. The digital compass and accelerometer recorded direction and acceleration readings 10 samples per second, respectively.

We evaluated the performance of MoLoc in a large office hall as shown in Figure 5. The experimental area is $40.8m \times 16m$ with columns, partition boards, and furniture such as shelves, desks, and chairs. We deployed only 6 APs as it is usually the case that only a few APs can be detected simultaneously. These APs are sparsely placed where each of their corresponding locations is marked with a star in Figure 5. The signals of all APs covered the whole office hall.

To study the localization performance, we applied a trace-driven approach to collecting and analyzing data. Specifically, we took 60 RSS samples at each of the 28 locations (marked with circles and location IDs). A quarter of these samples were collected when we faced to the north, east, south, west, respectively. We randomly selected 40 samples for the fingerprint database construction, 10 for location estimate when constructing the motion database and the remaining 10 for localization with MoLoc.

We collected the motion traces from 4 users with diverse height and walking speed. Each of them randomly walked along the aisles over half an hour. Overall, we collected 184 traces covering over 30 times of each reference location. Among them, 150 traces are used to train the motion database, and the remaining 34 traces are used for localization. We required the users to provide a feedback (e.g., make a mark) whenever they walked pass a reference location. Note that such requirement is not realistic in real-world scenarios as users usually have no knowledge of the reference locations in advance. It was used for reporting localization accuracy only.

B. Performance Evaluation

We first validate the motion database, and then analyze localization accuracy and errors to justify MoLoc's performance. In particular, we evaluate the localization accuracy in both overall scenario and that with large errors, in the presence of different number of APs. The convergence of accurate

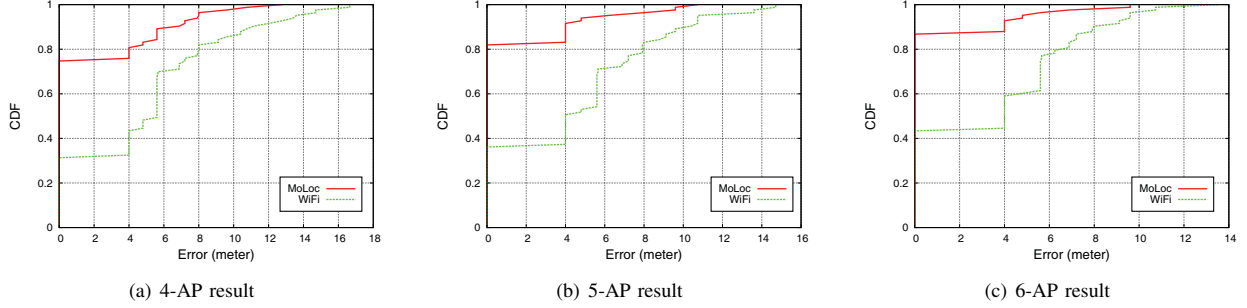


Fig. 7. Average performance of MoLoc over the WiFi fingerprinting method.

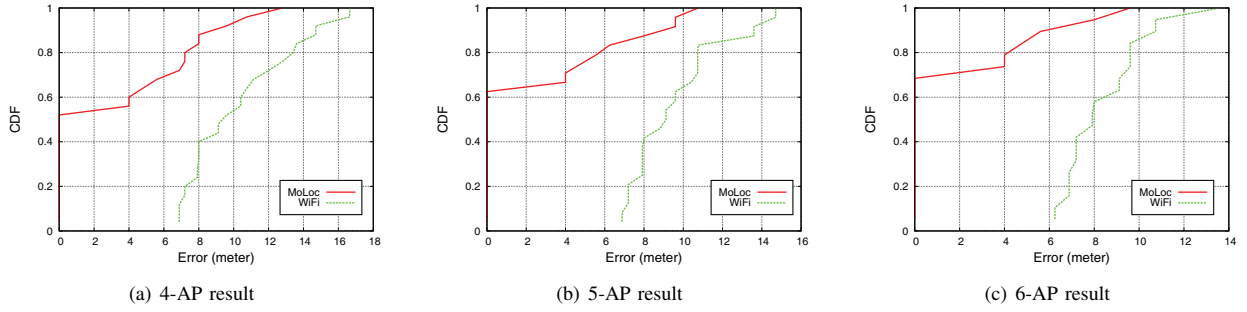


Fig. 8. Performance of MoLoc at the locations where the WiFi fingerprinting method has large errors.

localization (*i.e.*, localization times until stable accuracy) with statistics such as mean and maximum errors is also listed. We select the WiFi fingerprinting method as the baseline similar to [12].

1) *Validity of the Motion Database*: We first show that the direction and offset measurements in the motion database accord well with walkable paths between adjacent locations. We compute the direction errors by comparing the direction measurements in the motion database with those calculated by location coordinates. Figure 6(a) depicts the cumulative distribution function (CDF) of the direction errors. We observe that the crowdsourced direction measurements achieve a median error of only 3° , while the maximum is limited to 15° . This demonstrates the reliability of the motion database in direction measurements. For the relative large errors, we find that most likely it is due to the data reassembling: reversing directions generally brings in bias errors of 10° to 20° with our mobile phone.

Similarly, we depict the CDF of the offset errors in the motion database as shown in Figure 6(b). Compared with the ground-truth distances, the offset measurements have a median and maximum error of $0.13m$ and $0.46m$, respectively. Even the maximum error is smaller than a normal step size (usually between $0.7m$ and $0.8m$), suggesting that the step counting method can be well applied to measuring offset.

2) *Overall Localization Accuracy*: With the reliable motion database, we next study the overall localization accuracy of MoLoc. The accuracy is measured by localization error, *i.e.*,

the distance between the estimated location and the ground-truth one. Based on the standard deviations of the direction and offset measurements in the motion database, we set α and β , the Gaussian discretization intervals, as 20° and $1m$, respectively. The CDFs of the overall localization errors using 4, 5 and 6 APs are illustrated in Figure 7. Evidently, MoLoc outperforms the WiFi fingerprinting method to a significant extent in all cases. In particular, MoLoc achieves an average localization accuracy of 75%, 82%, and 86%, while the WiFi fingerprinting method achieves that of only 31%, 36%, and 43%, respectively. Note that such high accuracy is achieved with very limited number of APs (6 at most). This observation indicates that the location distinction by MoLoc does not rely much on dense deployment of APs (*e.g.*, 14 in [12]). Besides, MoLoc reduces the maximum error by around $4m$ in all cases. We discover an important insight into the still existing large errors of MoLoc: they are caused primarily by initial errors when user motion is not available yet.

3) *Dealing with Large Errors*: We conduct further investigations of the factors that contribute to the improved accuracy. From the localization results by the WiFi fingerprinting method, we find several pairs of locations (*e.g.*, 2 and 15, 10 and 27, 13 and 26) that exhibit similar fingerprints and thus lead to large errors. Therefore, we extract these locations where the WiFi fingerprinting localization has errors over $6m$, and see how MoLoc perform on these locations. Figure 8 presents the result. On average, MoLoc reduces the average and maximum errors at these locations by around $6.8m$ and $4m$, respectively. Such a great reduction in large errors gives

TABLE I
PERFORMANCE STATISTICS FOR CONVERGENCE OF ACCURATE
LOCALIZATION

Setting	EL	Accuracy	Mean error	Maximum error
4-AP WiFi	3.28	34%	4.91	16.64
4-AP MoLoc	1.57	89%	0.67	7.92
5-AP WiFi	2.71	39%	4.33	14.7
5-AP MoLoc	1.42	93%	0.36	6.25
6-AP WiFi	2.25	48%	3.27	13.6
6-AP MoLoc	1.13	96%	0.22	6.88

the first contributor to the improved accuracy.

4) *Convergence to Accurate Localization*: We next extract those traces that have erroneous initial estimates, and measure the convergence to accurate localization. We do this by calculating on average how many times of erroneous localization (EL) before the first accurate one, and summarizing the statistics, including accuracy, mean error and maximum error (in meters), for subsequent localization. Table I presents a summary with 4, 5, and 6 APs, respectively. In all cases, MoLoc approximately halves the number of EL required for the WiFi fingerprinting method. It also improves the accuracy of subsequent localization to around 90% or more, while that of the WiFi fingerprinting method is less than 50% to its best. The mean and maximum errors of subsequent localization also get reduced by 4.24m and 8.72m, 3.97m and 8.45m, 3.05m and 6.72m for 4, 5, and 6 APs, respectively. We infer from these statistics that MoLoc can accelerate the convergence to accurate localization and achieve high accuracy for subsequent localization.

VII. CONCLUSION

Traditional indoor localization systems mostly rely on RSS fingerprints to discriminate locations, which is insufficient in many cases. In this paper, we present MoLoc, a motion-assisted indoor localization scheme implemented on mobile phones. The main idea is to leverage user motion in localization. We adopt a crowdsourcing approach to construct a motion database storing direction and offset measurements between adjacent locations, and design a probabilistic algorithm to evaluate each location candidate. MoLoc is prototyped and deployed in an office hall. The experiment results show that with only 6 APs, MoLoc doubles the localization accuracy achieved by the fingerprinting method, and limits the mean localization error to less than 1m.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their constructive comments. This work is supported in part by the NSFC Major Program under grant 61190110, NSFC under grant 61171067, 61133016, 61272429, and 61272466, National Basic Research Program of China (973) under grant No. 2012CB316200, and the NSFC Distinguished Young Scholars Program under grant 61125202.

REFERENCES

- [1] Y. Liu, Z. Yang, X. Wang, and L. Jian, "Location, localization, and localizability," *Journal of Computer Science and Technology*, vol. 25, no. 2, pp. 274–297, Mar. 2010.
- [2] "PointInside," <http://www.pointinside.com/>.
- [3] "Happening," <http://www.happening-app.com/>.
- [4] "Facebook FriendShake," <http://www.facebook.com/friendsshake>.
- [5] "GeoCaching," <http://www.geocaching.com/>.
- [6] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," in *Proceedings of ACM MobiCom*, 2000.
- [7] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," *Wireless Networks*, vol. 10, no. 6, pp. 701–710, Nov. 2004.
- [8] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of IEEE INFOCOM*, 2000.
- [9] A. Varshavsky, E. de Lara, J. Hightower, A. LaMarca, and V. Otsason, "GSM indoor localization," *Pervasive and Mobile Computing*, vol. 3, no. 6, pp. 698–720, Dec. 2007.
- [10] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in *Proceedings of ACM MobiSys*, 2012.
- [11] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense: mobile phone localization via ambience fingerprinting," in *Proceedings of ACM MobiCom*, 2009.
- [12] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of WiFi based localization for smartphones," in *Proceedings of ACM MobiCom*, 2012.
- [13] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *Proceedings of ACM MobiCom*, 2012.
- [14] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proceedings of ACM MobiCom*, 2012.
- [15] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, Jan. 1992.
- [16] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala, "PinPoint: An asynchronous time-based location determination system," in *Proceedings of ACM MobiSys*, 2006.
- [17] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proceedings of ACM MobiSys*, 2005.
- [18] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powlledge, G. Borriello, and B. Schilit, "Place lab: device positioning using radio beacons in the wild," in *Proceedings of PERVASIVE*, 2005.
- [19] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: Wireless indoor localization without site survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, Apr. 2013.
- [20] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proceedings of ACM MobiCom*, 2010.
- [21] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo, "Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure," *Wireless Networks*, vol. 16, no. 2, pp. 405–420, Feb. 2010.
- [22] D. Madigan, E. Einahrawy, R. Martin, W.-H. Ju, P. Krishnan, and A. S. Krishnakumar, "Bayesian indoor positioning systems," in *Proceedings of IEEE INFOCOM*, 2005.
- [23] J. Liu, R. Chen, L. Pei, W. Chen, T. Tenhunen, H. Kuusniemi, T. Kröger, and Y. Chen, "Accelerometer assisted robust wireless signal positioning based on a hidden Markov model," in *Proceedings of IEEE/ION PLANS*, 2010.
- [24] "Amazon Mechanical Turk," <https://www.mturk.com/mturk/>.
- [25] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, "Did you see Bob?: human localization using mobile phones," in *Proceedings of ACM MobiCom*, 2010.
- [26] 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 *Proceedings of ACM MobiCom*, 2004.