HiMLoc: Indoor Smartphone Localization via Activity Aware Pedestrian Dead Reckoning with Selective Crowdsourced WiFi Fingerprinting

Valentin Radu and Mahesh K. Marina
The University of Edinburgh

***Abstract*—The large number of applications that rely on indoor positioning encourages more advancement in this field. Smartphones are becoming a common presence in our daily life, so taking advantage of their sensors can help to provide ubiquitous positioning solution. We propose HiMLoc, a novel solution that synergistically uses Pedestrian Dead Reckoning (PDR) and WiFi fingerprinting to exploit their positive aspects and limit the impact of their negative aspects. Specifically, HiMLoc combines location tracking and activity recognition using inertial sensors on mobile devices with location-specific weighted assistance from a crowd-sourced WiFi fingerprinting system via a particle filter. By using just the most common sensors available on the large majority of smartphones (accelerometer, compass, and WiFi card) and offering an easily deployable method (requiring just the locations of stairs, elevators, corners and entrances), HiMLoc is shown to achieve median accuracies lower than 3 meters in most cases.**

I. INTRODUCTION

Indoor mobile phone localization is gaining a lot of attention these days due to the increasing number of location-based services and applications that require accurate positioning or continuous tracking inside buildings. These applications can span from indoor navigation [1] to monitoring different aspects of the environment like the WiFi coverage [2] and can be used in many indoor spaces like offices, shopping malls and airports.

Dead reckoning and WiFi fingerprinting are well known approaches for indoor localization but each has its own advantages and limitations. While dead reckoning based schemes naturally enable continuous location tracking, error accrual over time is a major concern; moreover, dead reckoning in indoor environments with complex movement patterns is relatively more challenging. WiFi fingerprinting based localization approach is an attractive alternative as it can leverage the existing WiFi infrastructure (that is commonplace nowadays in most indoor environments) as well as exploit the presence of WiFi interfaces on smartphones. But the WiFi fingerprinting approach is not suitable for continuous location tracking of a mobile user because WiFi scanning operations are relatively quite power hungry. Also the applicability and effectiveness of WiFi fingerprinting is dependent on a number of factors including WiFi AP density, spatial differentiability and temporal stability of the radio environment.

We propose HiMLoc, a novel solution that synergistically uses Pedestrian Dead Reckoning (PDR) and WiFi fingerprinting to exploit their positive aspects and limit the impact of their negative aspects. Specifically, HiMLoc combines location tracking and activity recognition using inertial sensors on mobile devices with location-specific weighted assistance from a crowd-sourced WiFi fingerprinting system via a particle filter. HiMLoc uses the most common sensors available on the large majority of smartphones: accelerometer, compass, and WiFi card.

Our novel integration of dead-reckoning with WiFi fingerprinting is based on the observation that some spaces in a building are more accurately localizable with WiFi fingerprinting than others, which is a consequence of differences in spatial differentiability of the WiFi environment among these spaces due to building structure and radio signal propagation effects. To exploit this observation, we associate a weight for the WiFi fingerprinting component in a particle filter that influences the extent to which it is relied on in the hybrid localization. This weight is in turn inversely proportional to similarity area metric computed by comparing a run-time WiFi fingerprint with fingerprint database — smaller similarity area results in a higher weight and vice versa.

To ease deployment, HiMLoc requires just a small set of parameters specific to the new building, like position of stairs, position of elevators, position of main entrances and height of each floor. Moreover, WiFi fingerprinting component is crowd-sourced to adapt with infrastructure and environment changes and fast convergence towards increased location accuracy. Unlike other particle filter systems that require a detailed knowledge of the building layout, like the exact position of each wall and dimensions, to restrain the particles, our system uses distances to known reference points (corner, stairs, elevators and WiFi estimations) to determine the weights of particles.

Experimental evaluation of HiMLoc using Android phones shows that median location accuracy of under 3 meters is achievable even with complex movement within a building (e.g., going between floors using stairs and elevators). The evaluation was performed for two cases of carrying a phone, in hand and in pocket, with expectedly better results seen for the first. We evaluate the performance of HiMLoc by deploying it in a new building other than the one used for training the activity classifiers with positive results. The synergistic integration of the WiFi and PDR components is also revealed by our evaluation of HiMLoc spanning multiple floors within a building. On one hand, WiFi fingerprinting component provides the PDR component with an additional source for reference points for intermediate recalibrations. On the other hand, WiFi fingerprinting yields more accurate results

This work was supported in part by a Cisco Research Award.

978-1-4673-1954-6/12/$31.00 © 2012

when considering the knowledge of floor, which the PDR is able to identify via its activity classifier.

In the next section we present the background and related work. In section III we describe the design and implementation of HiMLoc, followed by its evaluation in section IV. We discuss related issues and directions for future work in section V and conclude in section VI.

II. Background and Related Work

A. Pedestrian Dead Reckoning

With the continuous miniaturization of sensors and the richness of applications they enable, their incorporation in modern phones is now indispensable. Taking advantage of their presence, recent years have seen an emerging class of location tracking systems that use inertial sensors to perform dead reckoning on mobile phones. These systems have the advantage that very little physical infrastructure is required for them to function.

Pedestrian Dead Reckoning (PDR) technique works by estimating successive positions starting from a known location, based on a way of estimating the traveled distance and the direction of walking. A solution to determine the traveled distance is to count the number of steps and estimate their length. Most typical step detection implementations are based on analyzing the acceleration data [3], [4], [5], but data from other sensors have also been tried, like angular velocity [6], [7], [8] and magnetometer data [9], or combination of these [10]. Using the acceleration magnitude, steps detection is performed through techniques like peak detection, which looks for peaks in the acceleration magnitude caused by the leg carrying the sensor touching the floor [11]; zero crossing, which monitors the acceleration value zero crossings [12]; and auto-correlation, by taking advantage of the repetitiveness of human walking [13]. The traveled distance can also be estimated, either by observing the rotation of the hip [14], or by estimating the length of the step. Probably the easiest way to estimate the step length is to appreciate it as a linear function of the frequency of stepping [15].

The other important component of the PDR is direction, which can be obtained by a compass or a gyroscope. The presence of a compass on a smartphone is more common than having a gyroscope. But compass indications are subject to magnetic interference inside buildings. Afzal et al. showed that these interferences can sometimes result in a direction deviation from the compass of up to 100° [16]. However, our experience was that under the normal conditions of human walking not too close to walls or other metal structures along the way, magnetic interferences are typically isolated and tolerable.

Common presence of sensors such as accelerometer and compass in smartphones have made PDR an attractive technique for mobile phone localization [17]. While most systems use PDR for outdoor tracking in conjunction with a map [18], others such as GAC [19] combine it with occasional GPS correction for energy-efficient location tracking on roads. A well-known limitation of PDR schemes is that error can get accumulated over time unless it is corrected in between.

The steady increase in performance of inertial sensors opened the opportunity for their use inside buildings with smartphones [20], [1], [18]. All of these systems have an increasing error accumulation if they are not periodically adjusted. Assisting the system with corrections from beacons has been experimented in [1]. For an easier deployment, activity recognition together with some knowledge of the building layout can provide some error correction points [20].

B. Activity Recognition

Gusenbauer et. al, introduced Pedestrian Dead Reckoning with Activity Classification, designed to navigate a person in an underground parking lot in [20]. Thus, they only consider the case of a person walking with the phone in hand and ahead of the user, not exploring other cases of carrying the phone and assuming no WiFi coverage in those environments. Ftrack [21] also uses an activity classifier to perform floor detection, having just a limited number of activities that can recognize, like movements on stairs and in elevator.

However, this is still not enough for a robust localization system. In general, the activity classifier cannot know all the possible movements that a user may perform and any large deviations from the training set can lead to confusion in the system.

We recognize and factor in the fact that activity classifier may not always provide an accurate result. This would be particularly true when the activity classifier is trained by a small group of users and needs to recognize the activities of a large number of diverse users. Through a particle filter we limit the effect of bad classifications by considering all the other activities with lesser weight, according to the classifier's confidence; this helps the system to recover in cases of wrong classifications.

C. Particle Filter

A Particle Filter is a numerical approximation to a Bayesian filter [22]. It has a number of 'particles', each representing a virtual position with its own weight to describe the likelihood of the user having that position. Particle filters are usually used in PDR system to incorporate maps in the system. Particles move independently on the floor plan and when they cross a wall they are eliminated, assigning higher weights to the other particles following the constraints imposed by the floor plan [6]. The only problem with this way of using Particle Filter is that a very detailed model of the building is required at deployment time, which is hard to obtain. In our case, the particle filter has the role of fusing activity classification and PDR estimation from inertial sensors with an independent location estimation from the WiFi fingerprinting positioning component.

D. WiFi Fingerprinting

WiFi fingerprinting is a well-known localization technique that can exploit the presence of WiFi interfaces now common on smartphones. WiFi infrastructure is also prevalent these days in many indoor environments. Early WiFi fingerprinting systems such as RADAR [23] and Horus [24] rely on an initial training phase to construct fingerprint database for use as a reference in the positioning phase later but training phase can be quite time consuming and expensive. More recent WiFi fingerprinting systems make this training phase automated via crowdsourcing using mechanisms of increasing sophistication (e.g., Redpin [25], OIL [26], WiFi-SLAM [27], Zee [28]).

While these systems work well with a sufficient number of samples, it is still a challenge to know which runtime fingerprints stand a good chance to provide a more accurate location estimation than others. Using just one fingerprint on the go requires a way to rapidly determine the value brought by each scan.

WiFi fingerprinting can be quite expensive from an energy consumption perspective if solely relied on for continuous location tracking. Another more obvious disadvantage of WiFi fingerprinting is that it works only where there is WiFi coverage. There are however usually some areas inside buildings not generally considered for Internet connectivity requirements like the stairs, toilets and some corridors. Despite this, WiFi fingerprinting can offer the needed correction for a PDR based system where available and if used judiciously as we show with HiMLoc.

E. Hybrid Localization Solutions

Hybrid localization approaches that combine PDR with WiFi fingerprinting try to avoid the disadvantages of either of those two individual approaches: PDR have enough correction instances to reduce the error accumulation in the navigational component and there always is a location estimation no matter whether is WiFi signal coverage or not.

Combining PDR with WiFi fingerprinting has been considered recently in [29] and [30]. The UnLoc system [29] combines the use of inertial sensors (accelerometer, compass, gyroscope) with the notion of natural and organic landmarks that are learnt over time for indoor navigation. While UnLoc looks to find WiFi landmarks based on the set of APs it sees, in [30] the use of WiFi fingerprinting is used only in the location where maximum signal strength is seen, to correct PDR at those points. While both [29] and [30] use basic PDR scheme, HiMLoc incorporates a more sophisticated version with activity recognition capability that would be needed in more complex environments (e.g., multi-floor buildings with elevators and stairs to move between floors). Moreover, unlike [29] and [30], HiMLoc uses only accelerometer and compass for the PDR which are present in almost every smartphone, thus achieving greater applicability. HiMLoc is presented at a high level in its initial form in [2] in the context of Pazl mobile crowdsensing based indoor WiFi monitoring system. The current paper provides a detailed design and evaluation of HiMLoc.

WiFi-SLAM [27] is a pioneer in bringing the robotics technique of SLAM (Simultaneous Localization and Mapping) into PDR. By using a detailed model of the building layout, their PDR implementation can track a person inside the building and collect WiFi scans to build the radio map at the same time. Their high accuracy is achieved by using specialized hardware. Similarly, Zee [28] learns the WiFi environment by using a PDR assisted by particle filter, in a crowd-sourcing manner. Unlike Zee and WiFi-SLAM, HiMLoc does not need a very detailed building model (the exact location of each wall); instead a few natural landmarks (position of elevators, stairs and corners) and some parameters of the building (height of each floor) are sufficient for HiMLoc to obtain a good level of localization accuracy. Another approach presented by Faragher et al. [31] was to use smartphones to collect acceleration data in order to estimate the movements using a Distributed Particle Filter Simultaneous Localization and Mapping (DPFSLAM). They relied on WiFi signal opportunistically, just to identify those places where the user has been before. Their experiment setup consisted of a single floor in an office building, with no intention of using landmarks like elevators and stairs and movements between floors.

Our system builds on these modern solutions and takes them one step closer towards an easily deployable and widely applicable indoor localization system.

III. Design and Implementation

A. *HiMLoc Hybrid Localization Mechanism Overview*

HiMLoc is illustrated in Figure 1. Phone's sensors (accelerometer, compass and WiFi card) collect sensor data (acceleration, orientation and WiFi scans) to be used as direct input to the system. The Activity Classification component determines what activity the user is performing within a short interval of time by sampling the Acceleration data. If the estimated activity can be performed in just a very limited number of places inside a building, like going up and down the stairs or taking an elevator, then Map Knowledge can assist to determine these possible locations. Acceleration and Orientation are used in the Pedestrian Dead-Reckoning (PDR) component to track the continuous movement. Finally, if a WiFi Scan is available, it is used to extract a runtime WiFi fingerprint. Such a fingerprint is compared with those in a fingerprint database (created via crowd–sourcing). Estimations of these components are merged by the Particle Filter to obtain a single estimation for the whole system. At the end of this process, if WiFi Scan information is available, it is annotated with the estimated location and used to update the fingerprint database.

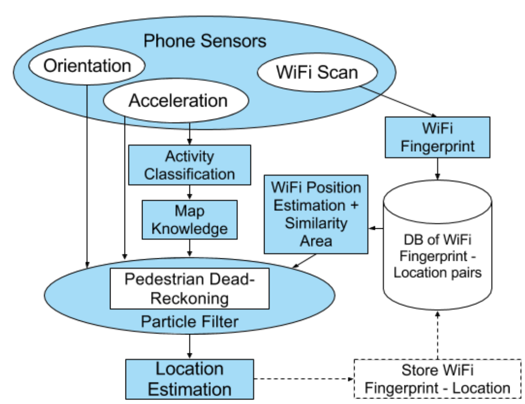


Fig. 1: Schematic of HiMLoc hybrid localization mechanism.

Next we present the two main components of HiMLoc: the Pedestrian Dead Reckoning driven by Activity Classification for continuous tracking and the WiFi fingerprinting component.

B. Pedestrian Dead Reckoning

The PDR estimates successive positions of a moving pedestrian starting from a known position through estimations of traveled distance and direction of movement. HiMLoc uses this method to track the position of a person when walking. But in order to know what activity the user is performing, HiMLoc relies on an activity recognition phase performed by the Activity Classification component.

Based on the detected activity, the system chooses how to interpret user's movements. HiMLoc needs this component to distinguish between vertical movements (going up/down stairs and elevators) and horizontal movements (walking). With the help of Map Knowledge, activity recognition can provide even more information about the user's location. Certain activities like going up or down stairs or taking an elevator can be performed only at a limited set of known locations inside a building. Getting the activity right has the effect of providing the needed periodic correction to the PDR in order to reduce the accumulating error caused by noisy sensors and other interferences over long tracks.

The most suitable sensor for activity recognition is the accelerometer as it is an inertial sensor permitting energy-efficient sampling at a high rate for continuous tracking. Most activities are performed similarly every time and their acceleration patterns can make them recognizable. All smartphones sense the acceleration on three axes orthogonal to one another. Considering that the sensitivity of the sensor is the same on all three axes, the acceleration magnitude will always indicate the same values, no matter the orientation of the phone:

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2} - g \tag{1}$$

where g is the Earth gravity, ax, ay and az represent the acceleration received on the Cartesian axes Ox, Oy and Oz respectively.

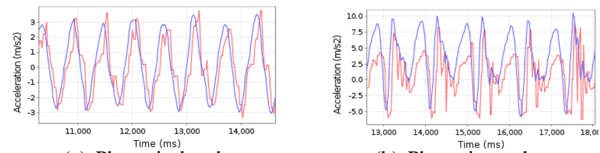
HiMLoc was designed to permit two ways of carrying the phone: in pocket and in hand. For the case with the phone in pocket we chose to investigate using the front pocket of the trousers. In the case of carrying the phone in hand we considered it to be straight in front of the user like for navigation purposes. A common aspect between these two cases is that the phone can be considered static relative to the user's body.

The system was trained to recognize the following activities: stationary, walking, elevator going up, elevator going down, going up on stairs, going down on stairs, opening and closing doors. Each of these were considered in the two scenarios: carrying the phone in hand and in pocket. The PDR component reacts differently to each of these activities.

Horizontal movements

If the activity performed by the user is determined to be walking, either with the phone in pocket or with the phone in hand, the user's movement is tracked on a horizontal plane, using traveled distance estimation and direction. Next we present how these estimations are obtained.

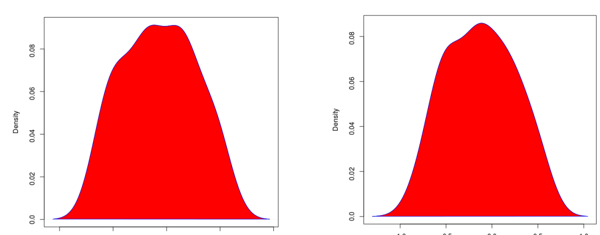
Figure 2(a) presents the acceleration magnitude pattern of walking with the phone in hand. The red curve indicates the raw acceleration and the blue curve is the same acceleration after adding a weighted average smoothing filter. Each step leaves the signature of a high spike in acceleration, caused by the heel touching the ground, followed by a deceleration. To estimate the traveled distance, HiMLoc first smooths the acceleration to eliminate some of the noise, then applies a zero crossing method to count the number of steps. In the case of walking with the phone in pocket, the same technique of counting the number of steps is used, but because the vibrations are more intensive when holding the phone in pocket, a low-pass filter is also used.



(a) Phone in hand. (b) Phone in pocket.

Fig. 2: Acceleration pattern (raw acceleration with red and filtered acceleration with blue) when walking.

Step length is computed as a linear function of stepping frequency [17]. HiMLoc computes the traveled distance as the sum of each step's length. This solution gives good results, but has its limitations. We conducted an experiment to evaluate the efficiency of this method of distance estimation on a window size 3.2 seconds of uniform walking. Doing several walks at different speeds we observed deviations of the expected distance from the actual traveled distance. The density of these deviations is represented in Figures 3(a) and 3(b). We observed errors of up to 15% that can have negative effect on the accuracy of the system. Our solution was to enforce the particle filter to correct for this deviations from the exact distance, as it will be presented later in the Particle Filter section.

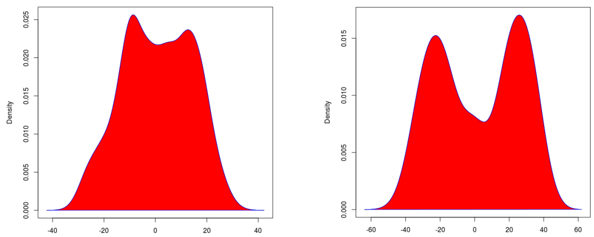


(a) Phone in hand. (b) Phone in pocket.

Fig. 3: Deviations of the estimated distance from the real traveled distance

The direction of movement also needs to be estimated. Considering that each smartphone has a compass, we chose this sensor to provide the direction. It is true that compasses are sometimes affected by magnetic interferences inside a building caused by the building structure and electric equipments, but we observed these interferences to be just isolated and not very disturbing when the person is moving at normal walking speed. Using a time frame to average the compass indications can eliminate some of the local interferences.

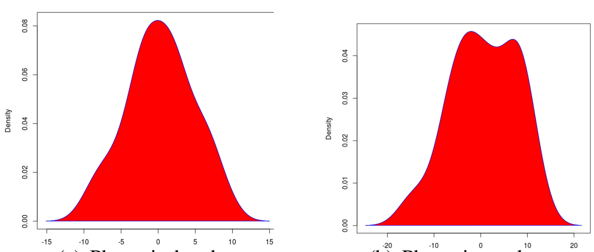
Evaluating the compass sensor on a long walk, we have observed that the human body has a slight rotation when stepping which is detected by the compass. Figures 4(a) and 4(b) show the compass deviation distribution in an interval of 3.2 seconds, capturing on an average 6 steps of walking. This rotation is more obvious with the phone in pocket (Figure 4(b)) as the hips tend to rotate much more than the upper body.



(a) Phone in hand. (b) Phone in pocket.

Fig. 4: Deviations of the compass indication from the true direction of movement

But choosing a good size window to average the compass data can overcome this rotation in order to provide a more reliable direction of movement. A window size of 3.2 seconds usually captures 6 steps of movement at average walking speed, which allows for every two consecutive steps to cancel each others rotations. This can be observed from Figures 5(a) and 5(b), where the compass indication is averaged over the time window and compared to the true direction of movement.



(a) Phone in hand. (b) Phone in pocket.

Fig. 5: Deviations of window averaged direction from the true direction of movement

HiMLoc considers the phone to have a static position relative to the body throughout the movement. To compensate any deviation of the phone from the user's frame orientation, a correction angle is determine after the initial few steps on the corridor, when we have the information of the corridor orientation from the Map Knowledge, or after two landmarks where we know the position of each landmark on the map, by assuming the walking movement to be in a straight line.

The distance and direction corrections are considered in the Particle Filter when choosing a distance and direction for each particle to progress the PDR.

If the compass deviation suddenly gets close to a right angle, the system infers that the user has left the corridor, either to go into a room or made a turn to another corridor. This event is considered as encountering a landmark and the position of the closest one is used to correct the system as it will be described in the Particle Filter section.

Vertical movements

Elevator movements present a specific pattern, with significant accelerations when the elevator starts and stops. Figure 6 presents these two events of the elevator denoted by the two large spikes in opposing directions. The number of floors ascended or descended by the elevator can be determined from the difference of times between the start and the stop of the elevator movement. In both cases of carrying the phone (pocket and hand), the elevator acceleration presents similar patterns.

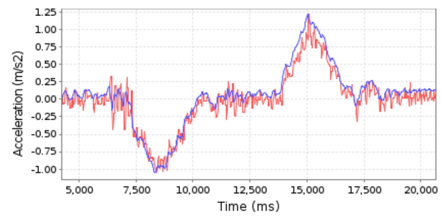


Fig. 6: Elevator acceleration showing a large spike at start followed by an opposing spike when stopping.

For the activities of going up and down the stairs, a similar method of step counting is used. By counting the number of stairs ascended or descended, the new level can be accurately determined as it is presented in the evaluation section. Figure 7 presents the acceleration magnitude for the activity of going down on stairs.

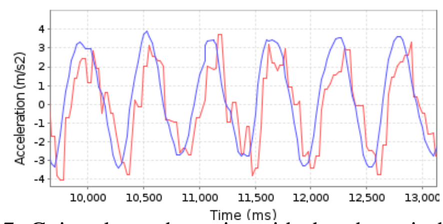


Fig. 7: Going down the stairs with the phone in hand.

Classification performance

The Weka[1] machine learning software was used to classify the acceleration samples into activities. The training set consisted of 176 instances of activities from two participants manually annotated with the right activity, each activity having at least 6 instances. These activities were: stationary, walking, going up on stairs, going down on stairs, going up by elevator, going down by elevator, opening and closing doors. All these activities were considered for both cases with the phone in pocket and with the phone in hand. Features were selected from the time domain (mean, variance, standard deviation, first integral (velocity), second integral (distance) and interquartile range) and from the frequency domain (energy and entropy) of the acceleration magnitude. Using Weka's cross-validation option, we compared two window sizes 128 and 256 samples and three classifiers, J48, Naive-Bayes and FT (Table I). These three classifiers had the best performance out of the classifiers implemented in Weka. Even though the 256 window size had a slightly better performance, we decided to use a window size of 128 samples because it allows more granular position estimations. The chosen classifier was Naive-Bayes because

[1] http://www.cs.waikato.ac.nz/ml/weka/

TABLE I: Weka classifiers performance with cross-validation.

	J48	Naive-Bayes	FT
128 window-size	70.5%	81.7%	80.5%
256 window-size	74.2%	85.3%	81.9%

of its good activity classification performance and faster run times.

The confusion matrix for Naive-Bayes showed that 10% of the activities of going down on stairs were classified as walking and another 10% as opening a door, while 5% of walking was classified as going down the stairs, 15% of the activities of going up on stairs were classified as walking. This was for the ideal case where activities were captured in the sampled window separately from other activities. In practice, it is common for more activities to be captured in the same window of 128 samples (3.2 seconds at a frequency of one sample every 25msec), so the rate of bad classifications may be higher in practice. To prevent these wrong classifications from having a significant negative effect on the location tracking, we need to assist the system with an independent component. For this we employ a WiFi fingerprinting based localization component which is described next.

C. WiFi Localization Component

This can be seen as a stand alone component but in HiMLoc we used it to complement the PDR estimation through a particle filter.

At run time, the vector of top five strongest APs and their signal strength values are selected and compared to the fingerprints in the database. The closest matching fingerprints are selected using Euclidean distance in the signal space (as in [32]). Fingerprints are stored in the database in groups representing cells. Each cell has the size of 1x1m and together they form the grid covering a floor plan. To support continuous update of the training set of WiFi fingerprints, all fingerprint are annotated with the time when they were collected. Newer fingerprints get a higher priority in fingerprint selection thus creating a simple solution to infrastructure change adaptation. The centroid of the closest three fingerprints gives the location estimation of the component.

We identified that the position estimation with WiFi fingerprinting is not spatially uniform, some areas having a higher accuracy of localization than others. Figure 8 indicates regions with a high (green) and low accuracy (red), based on the distance between the estimation and the ground truth.

In order to know when a WiFi location estimation is reliable, we introduce the notion of *similarity area of a WiFi fingerprint*, which is the area described by all points in the fingerprint database with a fingerprint *close* to the one at runtime. A threshold for the Euclidean distance in the signal space between the runtime fingerprint and each fingerprint in the database is used to define closeness. We set this threshold empirically to 12.5 in our implementation. The area spanning all close points determines the similarity area. Figure 9 shows the correlation between the estimation error and the similarity area.

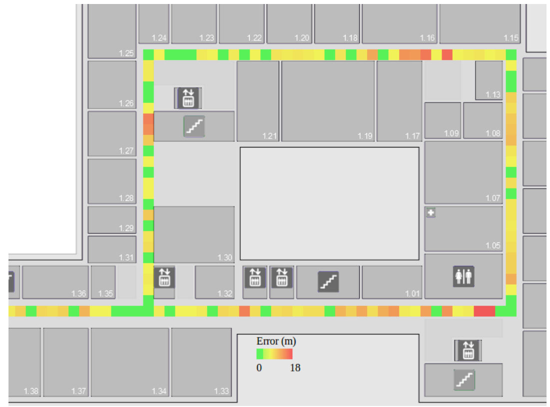


Fig. 8: Spatial distribution of WiFi fingerprinting based location estimation errors on the floor plan.

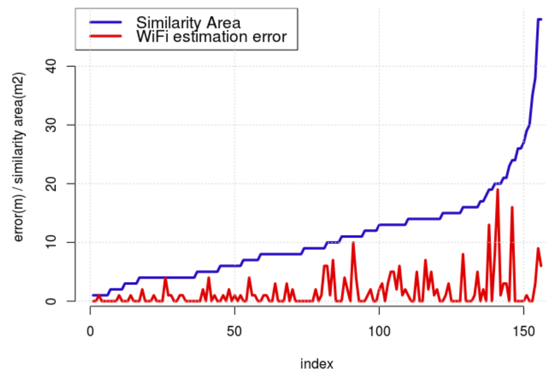


Fig. 9: Correlation between the estimation error and the similarity area.

We observed that the errors of estimation are much lower when the similarity area is small. While the errors are not necessarily larger when the similarity area is higher, they are more variable than to the left of the chart, so our solution is to consider the estimations with a low similarity area as offering a higher certainty of their indication. In fact, having a small similarity area is an indicator that the fingerprint is well distinguishable from other fingerprints and similar fingerprints can be found in just a small area in the building. HiMLoc assigns higher weights to the estimations with a low similarity area as they are considered to be more accurate.

D. Particle Filter

HiMLoc uses a Particle Filter to integrate all estimations from Activity Classifier, Map Knowledge, WiFi positioning component and PDR's variables (distance and direction). The role of the particle filter is to correct these estimations that are possibly affected by noise. This is done by investigating all possible activities based on their probability, determining the possible distance deviation and compass deviation in each time window.

Each particle has its own PDR component where it chooses an activity for each time window based on the probabilities provided by the Activity Classifier for each activity, a distance deviation for walking in the time window and a compass deviation. The compass deviation at the window level (Figures

5(a) and 5(b)) and the distance deviation (Figures 3(a) and 3(b)) can be tightly fitted by a normal distribution. Based on their observed behavior in practice, the probability of encountering any deviation is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{2}$$

where, x is the chosen deviation and μ is the mean and σ the standard deviation of observed model.

Based on the probability, each particle selects its own correction values to compensate for the estimated value. In turn, this probability will affect the weight of the particle. The activity recognition variable gets its probability from the classification confidence of the Activity Classifier.

The other purpose of the Particle Filter is to prevent the system from getting lost when the PDR starts accumulating errors. When there is an external assistance, for instance a position is indicated by the Map Knowledge (e.g. because of a corner), particles weights are updated inverse proportional to the distance between the particle's position and the assistance indicated position. In the case of the WiFi component estimations, the confidence of the estimation is determined based on the similarity area. As it can be observed from Figure 9, when similarity area is small, the errors of WiFi location estimation tends to be small, so we want to assign a higher confidence to those estimations. An exponential model provides the confidence of WiFi location estimations by indicating high confidence when the accuracy area is small and low confidence when the similarity area is high. The weight of each particle is updated based on WiFi estimation confidence and on the distance between the position of the particle and the WiFi estimation.

So, the weight of a particle is updated as a sum of all the weights of the probabilistic variables:

$$w_i = w_0 + w_a + w_o + w_d + w_f \tag{3}$$

where w_i is the final weight, w_0 is the initial weight of the particle and w_a, w_o, w_d, w_f are the weights computed for the particle's variables (activity selection, orientation, distance and WiFi fingerprinting based fix assistance if available) based on their likelihoods.

The life cycle of the Particle Filter begins with all the particles having the same weight at the starting point. There are three steps repeated by the Particle Filter in a loop:

- selection of particles. At the start of the iteration, some particles are sampled to progress and create the new group of particles. This selection is done based on their weight.
- weight update based on the variables selected by the PDR. Each particle randomly creates its own set of variables and progresses the particle, updating its weight accordingly.
- observations about the environment update the particles' weight. If there is an external contributor like the Map Knowledge or the WiFi positioning, particles closest to the specific positions get higher weights.
- weight normalization. The weight of all particles are normed to sum up to one.

E. Implementation

HiMLoc was implemented as a system with two parts: a mobile application that collects data from the phone sensors; and a server side application that receives and processes this data. With the phone application designed to run on a large number of smartphones, we chose the Android platform and evaluated our implementation using HTC Nexus One phones. For increased availability with concurrent access, the server side application runs as a cloud app on the Google App Engine platform.

Phone sensors are sampled only when the user carrying the phone moves. When the phone is static, the compass and radio card are disabled to save energy. Only the accelerometer is left on to run at a lower frequency just to sense when the user is moving again. Acceleration, orientation and WiFi scans are locally stored to be uploaded opportunistically to the server: when the phone is charging, when WiFi connectivity is available, or when an upload is requested by the user.

The frequency of WiFi scans was chosen to be one scan every 20 seconds, which is a compromise between keeping the energy consumption low, with each WiFi scan imposing an extra energy consumption on the phones, but also gather enough data to assist the PDR estimation more often.

IV. Evaluation

In this section we present our evaluation of the floor detection method as part of the PDR component and the evaluation of HiMLoc in three different scenarios.

A. Floor detection

There are two types of movements that HiMLoc interprets: vertical movements and horizontal movements. The vertical movement is described by the movement of the elevator and going up and down the stairs. The immediate effect of correctly estimating the vertical movements is determining the change in floor level.

We evaluated the performance of the PDR floor detection in two different buildings in the University of Edinburgh: Informatics Forum (IF) and Appleton Tower (AT), each with their own different characteristics. The Activity Classifier was trained in a single building and used to recognize the performed activities in both buildings.

Table II presents the performance of stair counting in the case of using the stairs with the phone carried in hand. These numbers indicate the performance of stair counting as an average of 5 independent movements between a number of levels indicated for each building. The observation was that even if in some cases the stair counting mis-performed by a few stairs, the number of these wrongly counted steps was substantially smaller than half of the number of stairs between two levels, and so the level identification was not affected. In all the evaluation scenarios, the system indicated the correct floor. The same performance was achieved in the case of carrying the phone in pocket as indicated in Table III.

TABLE II: Stair Counting performance when using the stairs with the phone in hand.

No of floors	Actual number	Counted going up	Counted going down
IF 1 floor	24	24	24.8
IF 2 floors	50	48.8	49.6
IF 4 floors	102	97.4	100.6
AT 1 floor	29	28	28.3
AT 2 floors	59	57.6	56.6
AT 4 floors	119	115.6	118

TABLE III: Stair Counting performance when using the stairs with the phone in pocket.

No of floors	Actual number	Counted going up	Counted going down
IF 1 floor	24	25.2	26.8
IF 2 floors	50	52.4	52.2
IF 4 floors	102	100.8	105.8
AT 1 floor	29	28	30
AT 2 floors	59	60	56.6
AT 4 floors	119	120	115.3

The elevator vertical movement was similarly evaluated, this time looking at the time between the elevator peaks, representing the start and stop of elevator movement as observed in Figure 6. Table IV presents the times between the elevator peaks measured by the system when the elevator was moving a number of floors as specified. The observation is that floor detection is possible because the time between two floors is more than 2000ms, for both of the two buildings, while the maximum deviation of the time between peaks from what was expected was less than half of the time between two consecutive floors.

TABLE IV: Elevator times between floors.

No of floors	Going up-avg time (ms)	Going up-max time deviation (ms)	Going down avg time (ms)	Going down-max time deviation (ms)
IF 1 floor	2804.3	120.6	2662.8	63.83
IF 2 floors	4262	288	4318	285
IF 4 floors	8765	310	8865.4	290.4
AT 1 floor	2587.75	288.75	2643.75	393.2
AT 2 floors	5306.5	306.5	5431.5	368.5
AT 4 floors	11230.25	318.75	10899.75	401.25

Performance of floor detection in the case of elevator movements was again 100%. It should be noted that this evaluation was performed when the user was static. Any extra movements from the user might not allow the best peak detection, but at the same time, this assumption is plausible since the elevator doors are closed while the elevator is moving.

B. Localization Accuracy in Different Scenarios

To evaluate the performance of HiMLoc, we put the system to the test in three different scenarios. First scenario was designed to evaluate the performance of the HiMLoc system on one floor of an office environment where frequent landmarks were present, corners and WiFi assistance, with a large training set of WiFi Fingerprint-Location pairs. The second scenario was to evaluate HiMLoc performance for movements that span multiple floors. And the third scenario was to monitor HiMLoc's performance evolution during deployment in a new environment.

Single floor of an office building

For this experiment we used the Informatics Forum, which is a modern office building. To train the WiFi fingerprinting component, we collected multiple fingerprints on the first floor annotating them with their precise location. This was done in a crowd-sourced manner, data being collected my multiple users to be joined in a single database on the server side application. There are already solutions available that can automate this process much faster, like WiFi-SLAM [27], but we chose this approach to avoid the complexity of other systems and to have a higher confidence on the training set for the WiFi localization component.

To evaluate the accuracy of HiMLoc, we selected a track of about 100m on the corridor with a number of 20 reference points representing entrances to offices adjacent to the corridor. Localization error was determined as the Euclidean distance between the known position of these reference points and HiMLoc's location estimation at the time of encounter. We compare the two cases of carrying the phone in pocket and in hand. Results for localization error with HiMLoc for these two cases are reported earlier in [2] (see Fig. 12). Those results essentially show that the case of carrying the phone in hand always has a higher location estimation accuracy (median error under 2m with phone in hand vs. median error between 2-3m for the phone in pocket case) as counting the number of steps with the phone in pocket is a relatively harder task.

For the following experiments we considered only the case of carrying the phone in hand.

Moving between floors

In the second experiment we included the second floor of the same building too. Starting from the same starting point on the first floor, the track went up the stairs and followed the second floor corridor similar to the first floor track. This experiment was designed to evaluate the training set of the WiFi component when moving between floors. In the first instance we had all the WiFi fingerprints from the entire building in a single training set. The effect of this was a lot of confusion in the WiFi component of HiMLoc, making mistakes between floors (Figure 10). As a consequence, we decided to rely on the PDR to estimate the floor and use only the fingerprints from the same floor as training set for the WiFi component.

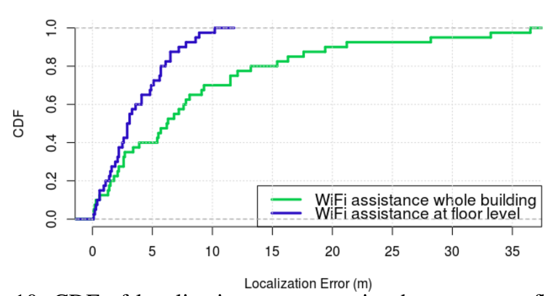


Fig. 10: CDF of localization errors moving between two floors.

We then wanted to evaluate the performance of the hybrid approach compared with each of the two localization solutions alone: PDR and WiFi fingerprinting. WiFi fingerprinting alone cannot perform where there is no proper WiFi coverage and continuous scanning has negative implications on the battery life. The average energy used while performing WiFi scans is about 260mW, whereas the accelerometer needs 3mW and the compass 60mW, on a Nexus One phone. HiMLoc uses the cheaper sensors (compass and accelerometer) for continuous sensing and occasionally WiFi scans, with the effect of reducing the power consumption of the system.

To evaluate the improvement of HiMLoc over PDR with Activity recognition alone, we performed another experiment over two floors in the same building. The track involved walking on the corridor at the first floor, going up on stairs to the second floor, walking on the corridor at the second floor, walking in a large open space, resting on the couch, walking on the corridor again, taking the elevator back to the first floor and walking back to the starting point. Using this track, we compared the performance of the PDR with activity recognition alone with the performance of HiMLoc (Fig. 11). We can see that HiMLoc performs better as median error but also having lower errors overall, due to occasional assistance from WiFi fingerprinting when there are long periods of no assistance from Map Knowledge in the PDR.

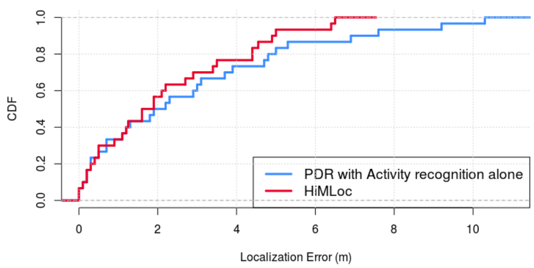


Fig. 11: Comparison between PDR alone and HiMLoc.

Deploying the system in a new building

For this experiment we chose a large open floor in a different building from the office building used before. This building is used for lectures, group reunions and other student activities. It has the first two floors joined in a large open space in the middle of about 600 m^2 with lecture theaters and a coffee area on sides. It has stairs on two sides and an elevator to reach the second floor where there is a half open corridor to facilitate access to some more lecture rooms. This was ideal to evaluate the case of deploying to a new environment with fewer landmarks from the building structure.

After inputting the system's parameters, like the location of stairs, elevators and corners, the system was ready for test. In the first instance there were no WiFi fingerprints in the training set, so the system was running only on PDR. After an hour of continuous movements in the open area, using the stairs and the elevator between the first two levels, the system had collected a number of 200 WiFi fingerprints to be used as training set. Figure 12 presents the performance of the system at start and after an hour.

We observed that the performance of the system improved over time. After collecting WiFi fingerprints for just an hour the system had a median improvement of about 20%. This performance improvement over time is obtained from the higher number of assistance opportunities for the PDR, with the WiFi component having a denser training set.

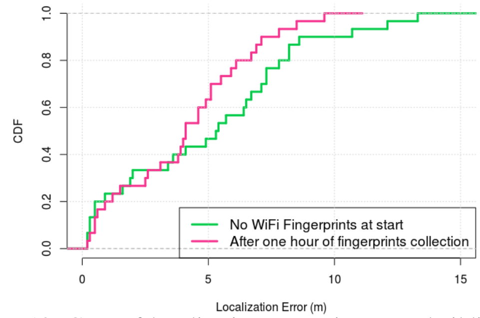


Fig. 12: CDF of localization errors in a new building.

Our system was designed to be available to most smartphone users, utilizing just compass, accelerometer and WiFi card, to facilitate a crowd-source built WiFi training set by more people moving around the building with their daily activities. This allows a faster convergence to a more accurate estimation, making HiMLoc even more easier to deploy.

V. Discussion and Future work

Probably the most important resource of smartphones is their battery. Long running applications have to consider their impact on this resource and to reduce energy consumption as much as possible. But there is always a trade-off between low energy consumption and system's performance. It is the case with localization on smartphones as well. Continuous use of WiFi scans consumes more energy than continuous sampling of acceleration data. With HiMLoc we aimed to reduce the number of WiFi scans but also keep providing some to reduce the error accumulation in the PDR caused by drift and noisy readings. In this trade-off between location accuracy and battery consumption we found that one WiFi scan every 20 seconds is ideal for our cause. This was empirically determined based on analysis of the time between landmarks in one of the buildings we investigated. We consider this to be a feature imposed by the building, as well as the application's purposes. If the application requires high location accuracy, then this frequency can be increased to provide assistance for the PDR more often or decreased if the application constraints are not very strict. In the end, the localization system should be adapted base on the purpose of its application. We leave the investigation of this energy-accuracy tradeoff determined by application requirements for future work.

In two of our evaluation scenarios we had access to a crowd-sourced WiFi site survey of the Informatics Forum building. This was collected with the help of a group of users inputing their location on an map through a graphical interface while the application was scanning the WiFi environment. In our third scenario we considered a naive way of building the WiFi database, where all the WiFi scans are annotated with the estimated location from the system and stored in the database, without any filtering or creating relations between fingerprints. More sophisticated solutions, like [27] and [28] do that, but they require more detailed building models, whereas HiMLoc was conceived as a easy deployable solution, with just a small set of parameters: location of stairs, elevators, corners, entrances and height of floors. In future work we will investigate more effective crowdsourced WiFi fingerprinting that is inline with our goals, i.e., to create an easy to deploy system which can be used by many people.

The two cases we considered for carrying the phone: in the front pocket of trousers and in hand in front of the user, were chosen as consequence of an initial investigation of how people carry their phone and also the need to have the phone in a static position relative to the user's body so that the phone can detect user's movements more accurately. We trained the Activity Classifier with samples of activities in one building and used these in both buildings of the experiment. However, some people may prefer to carry their phones differently, like in a bag or purse, but these cases are very hard for location systems that use inertial sensors because their position is not fixed and the acceleration detected by the phone is a mixture between the bag's movement and the free movement of the phone inside the bag. Some possible work around this problem would be to learn the patterns of movements on the way, with more assistance from independent references, like landmarks or WiFi, knowing that people tend to keep a uniform motion when walking. We leave this investigation for future work.

VI. Conclusions

Smartphones equipped with several sensors and network interfaces aid in indoor phone localization. In this paper, we have presented HiMLoc, a hybrid indoor location tracking solution that integrates Pedestrian Dead Reckoning with indoor landmarks detection and WiFi fingerprinting. The main advantage of this solution is that it offers easy deployment as it relies on only a small set of building parameters (e.g., location of elevators, stairs and corners and distance between floors) and can provide good estimation for most smartphones by using just three of the most common sensors present on smartphones: accelerometer, compass and WiFi card. Our integration of PDR with WiFi fingerprinting based estimations is performed by a particle filter and is based on the concept of similarity area for WiFi fingerprints. Very distinct fingerprints over a small area tend to provide very good location estimation accuracy as do fingerprints obtained from the same floor. Evaluations show that HiMLoc achieves median location error less than 3 meters in most cases. In future work we will investigate the trade-off between localization related energy consumption and desired localization accuracy as determined by application requirements.

References

[1] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. Did you see Bob?: Human Localization using Mobile Phones. In *ACM MobiCom*, 2010.

[2] V. Radu, L. Kriara, and M. K. Marina. Pazl: A Mobile Crowdsensing based Indoor WiFi Monitoring System. In *Proc. IEEE CNSM*, 2013.

[3] R. Stirling, J. Collin, K. Fyfe, and G. Lachapelle. An Innovative Shoe-Mounted Pedestrian Navigation System. In *GNSS*, 2003.

[4] B. Krach and P. Roberston. Cascaded estimation archi- tecture for integration of foot-mounted inertial sensors. In *Proc. IEEE Position Location and Navigation Symposium*, 2008.

[5] N. Castaneda and S. Lamy-Perbal. An improved shoe- mounted inertial navigation system. In *Proc. IEEE IPIN*, 2010.

[6] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In *Proc. ACM UbiComp*, 2008.

[7] F. Cavallo, A. Sabatini, and V. Genovese. A step toward GPS/INS personal navigation systems: real-time assessment of gait by foot inertial sensing. In *Proc. IEEE Conference on Intelligent Robots and Systems*, 2005.

[8] L. Ojeda and J. Borenstein. Non-GPS navigation with the personal dead-reckoning system. In *Proc. SPIE*, 2007.

[9] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara. A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU. In *IEEE International Symposium on Intelligent Signal Processing*, 2009.

[10] E. Foxlin. Pedestrian Tracking with Shoe-Mounted Iner- tial Sensors. In *IEEE Computer Graphics and Applications*, 2005.

[11] L. Fang, P. Antsaklis, L. Montestruque, M. McMickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, M. Xie, and X. Xie. Design of a Wireless Assisted Pedestrian Dead Reckoning SystemThe NavMote Experience. In *IEEE Trans. Instrum. Meas.*, 2005.

[12] P. Goyal, V. J. Ribeiro, H. Saran, and A. Kumar. Strap-down Pedestrian Dead-Reckoning system. In *Proc. IEEE IPIN*, 2011.

[13] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, M. Schiek, S. Leonhardt, T. Falck, P. Mahonen, and R. Magjarevic. 4th International Workshop on Wearable and Implantable Body Sensor Networks. In *Springer Berlin Heidelberg*, 2007.

[14] H. WeinBerg. AN-602: Using the ADXL202 in Pedometer and Personal Navigation Applications. In *Analog Devices, Tech. Rep.*, 2002.

[15] S. Yang and Q. Li. Ambulatory walking speed estimation under different step lengths and frequencies. In *Proc. IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2010.

[16] M. H. Afzal, V. Renaudin, and G. Lachapelle. Assessment of indoor magnetic field anomalies using multiple magnetometers. In *Proc. ION GNSS*, 2010.

[17] R. Harle. A Survey of Indoor Inertial Positioning Systems for Pedestrians. *IEEE Communications Surveys & Tutorials*, 2013.

[18] I. Constandache, R.R. Choudhury, and I. Rhee. Towards Mobile Phone Localization without War-Driving. In *IEEE INFOCOM*, 2010.

[19] M. Youssef, M. A. Yosef, and M. El-Derini. GAC: Energy-Efficient Hybrid GPS-Accelerometer-Compass GSM Localization. In *Proc. IEEE GLOBECOM*, 2010.

[20] D. Gusenbauer, C. Isert, and J. Krsche. Self-Contained Indoor Positioning on Off-the-Shelf Mobile Devices. In *IEEE Indoor Positioning and Indoor Navigation (IPIN)*, 2010.

[21] H. Ye, T. Gu, X. Zhu, J. Xu, X. Tao, J. Lu, and N. Jin. FTrack: Infrastructure-free Floor Localization via Mobile Phone Sensing. In *IEEE Percom*, 2012.

[22] J. Hightower and G. Borriello. Particle Filters for Location Estimation in Ubiquitous Computing : A Case Study. In *Computing*, 2004.

[23] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *IEEE INFOCOM*, 2000.

[24] M. Youssef and A. K. Agrawala. The Horus WLAN Location Determination System. In *ACM MobiSys*, 2005.

[25] P. Bolliger. Redpin – Adaptive, Zero-Configuration Indoor Localization through User Collaboration. In *Proc. ACM MobiCom MELT Workshop*, 2008.

[26] J. Park et al. Growing an Organic Indoor Location System. In *Proc. MobiSys*, 2010.

[27] B. Ferris, D. Fox, and N. Lawrence. WiFi-SLAM Using Gaussian Process Latent Variable Models. In *Proc. IJCAI*, 2007.

[28] A. Rai et al. Zee: Zero-Effort Crowdsourcing for Indoor Localization. In *Proc. ACM MobiCom*, 2012.

[29] H. Wang et al. No Need to War-Drive: Unsupervised Indoor Localization. In *Proc. MobiSys*, 2012.

[30] Y. Kim, H. Shin, Y. Chon, and H. Cha. Smartphone-based Wi-Fi tracking system exploiting the RSS peak to overcome the RSS variance problem. *Elsevier Pervasive and Mobile Computing*, 9(3), Jun 2013.

[31] R. M. Faragher, C. Sarno, and M. New. Opportunistic Radio SLAM for Indoor Navigation using Smartphone Sensors. In *Proc. IEEE Position Location and Navigation Symposium (PLANS)*, 2012.

[32] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from Mere Connectivity. In *Proc. ACM MobiHoc*, 2003.