Design and Evaluation of an Indoor Positioning System Framework

Kaifei Chen
Karthik Vadde

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2016-16
http://www.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-16.html

April 13, 2016

Copyright © 2016, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Acknowledgement

This work is supported in part by the National Science Foundation under grant CPS-1239552 (SDB).

Design and Evaluation of an Indoor Positioning System Framework

Kaifei Chen
Computer Science
University of California, Berkeley
kaifei@berkeley.edu

Karthik Reddy Vadde
School of Information
University of California, Berkeley
karthikv@berkeley.edu

Abstract

Numbers of indoor positioning techniques have been proposed, but none of them is generally deployable in most buildings, in spite of their acceptable accuracy in experiments. The primary reason is that, to work properly, all of them have specific assumptions about context, like infrastructure, environment, and user behaviors. However, assumptions of them are almost mutually independent, and their working context are actually detectable. In this paper, we propose a indoor positioning system framework to combine location estimations from multiple positioning systems. Each system is encouraged to implement its own context-awareness mechanism, and needs to provide a confidence about its estimation. The framework maintains credit history for all participating positioning system entities, and fuses their estimations based on their credits and confidences. Our preliminary deployment shows that it is not hard to provide a highly reliable confidence, and using data fusion techniques like Kalman Filter can further reduce errors in the framework's decision.

1 Introduction

Indoor localization is one of essential technologies for many applications, such as energy-efficient buildings, disaster rescue, and indoor navigation. Numbers of Indoor Positioning System (IPS) has been proposed and implemented. One group of them are built upon the idea of fingerprint, which means a signature of environment features consistently and strongly depending on the physical location. Examples include WiFi Received Signal Strength Indicator (RSSI) from all Access Point (AP) in the building [3], FM radio signal features (RSSI, SNR, multi-path, etc.) [5], acoustic background spectrum [16]. Another group of systems look into installing extra infrastructures as beacons in building, where users carry receiver of the signal sent by beacons to form fingerprints. The technologies used by beacons include ultrasound [14, 18], infrared [17], magnetic induction [10], RFID [12], Doppler effect on radio signals [4]. Moreover, rich sensors available on smartphones stimulates researchers to bring the idea of dead reckoning into trajectory estimation and localization [15, 11]. Basically, they use Inertial Measurement Sensors (IMU) embedded in commercial smartphones (e.g. accelerometer, gyroscope, compass, magnetometers) to estimate the velocity and direction of users, and in turn estimate the trajectory and the location given the known start point.

However, no agreement has been made on a general solution as Global Positioning System (GPS) in outdoor localizations [9], which doesn't work well for indoor environments because of the signal attenuation inside buildings. Almost all indoor localization techniques have specific assumptions, which in turn become limitations that thwart their large scale real-world deployments. Firstly, there are general problems for different types of techniques. IMU performs for sporadic movements and suffers from cumulative drift errors. Fingerprint-based methods require the signature at each position to be consistent by time, and the interference should not low as well. But at this stage there is no universal consistent signal. Moreover, beaconing techniques need extra infrastructures that are not common in buildings, which will bring lots of cost on devices, setup, and maintenance. Secondly, some IPSs are designed for specific scenarios. As an example, SurroundSense [2] is designed for shopping mall, where there are always distinguishable features for all adjacent stores and similar types of stores in terms of light, color, and sound.

We argue that their limitations don't mean these systems are useless, because of following points. First, the limitations don't always happen at the same time, and some of them have actually been proven to be almost mutually independent. Liu et al. [5] look at the consistency of fingerprint of WiFi signal and FM signal, and found for most locations, there are at least one consistent signal. Second, it is not hard to detect whether the context satisfies the assumptions. For example, we can regard accelerometer data with very large variance as uncommon or too complex movement from users. The signal consistency can be checked when smartphone detects little movement from accelerometer (i.e. the phone is staying at one location). Third, crowdsourcing from user inputs is also a good ways to determine the context, especially when proceeded in a motivating manner, like games or competitions.

Another reason for the absence of consensus on indoor localization solution is their heterogeneous goals. Some systems pay attentions on the physical precision of the estimated location, such as some Simultaneous Localization And Mapping (SLAM) application employed by robots that aspire very fine grained positioning results. Some other systems only care about the specific locations relevant to the application. A building energy tracking system [6] only cares about the zones around electronic devices. In addition, instead of accurate physical location estimation, some applications need fine-grained room level positioning, which we call as semantic localization. Semantic localization differs from physical localization in the sense that two physically nearby points could locate in two different (and maybe adjacent) rooms, imaging two locations against the same wall form each side; and two points in one room can be apart from each other at meter level, imaging two points locating at two diagonal corners. A consumer behavior analytics system in mall [2, 19] needs accurate estimations of the stores the consumer visited.

In this paper, we propose an IPS Framework, to combine different indoor localization techniques accordingly based on their limitations and context, as well as the requirements of the application. The key point is to assign higher credits and weights to more possibly accurate results, and combine them using appropriate data fusion methods. Each IPS is required to implement its own way of predicting the accuracy of its result, which we call as *confidence*. The overhead for existing IPSs to be integrated is independent of existing implementation, but depends on how accurate they want to the confidence to be. The algorithms to calculate the confidence vary between different techniques, and how well the designers and developers understand the localization methods. As we described above, one important point is many IPSs can be implemented to be context-aware. For example, fingerprint-based confidence could be more confident if the signal is consistent. Based on their confidences and the history performance (i.e. the *credit*), the framework assigns weight to each of them. IPSs with more accurate confidences in history gain higher credits. The idea is to recognize the best results by conservatively trusting their confidence. With different results assigned different weights, we look into how to combine their data to provide more precise and smooth trajectory estimations. Note we are neither trying to denoise the raw sensor data nor improving accuracy for any IPS.

To depict our ideas, we implemented WiFi RSSI-based fingerprint localization and Acoustic Background Spectrum (ABS) fingerprint localization [16] [1] and the prototype of our framework. Both of them are embedded with their own confidence calculation mechanisms. We also implement linear weighted average and Kalman Filter [1] to combine the estimations from WiFiLoc and ABSLoc. Based on the fingerprint database we build for a university building, we evaluate our confidence algorithms and framework with three trajectories, along which both WiFi and ABS fingerprints are sampled. The results show that the framework can trust the confidence, and hence yield results no worse than the best estimations among WiFiLoc and ABSLoc.

In summary, our contributions are:

- We point out that most IPS can do context-awareness, and hence provide an accurate confidence of their localization results.
- Given each IPS can predict their result accuracy, we propose the idea of IPS framework that accordingly adjusts and uses appropriate IPSs under different circumstances.
- With real implementations, we prove the correctness and feasibility of the confidence algorithms and framework.

The rest parts of this paper are organized as follows: in Section 2, we introduce some IPSs briefly, and talk about related works in combining IPSs. The system architecture is described in Section 3. We discuss the ideas of calculating confidences with detailed examples in WiFiLoc and ABSLoc in Section 4, and data fusion methods to combine multiple sources of estimated locations in Section 5. The evaluations are described and analyzed in Section 6. We talk about our future work and conclude the paper in Section 7 and 8 respectively.

2 Related Work

Some former work of combing multiple indoor localization techniques resulted to significant performance improvements. Azizyan et al. take WiFi, sound, light, and color features from mobile phone sensors as signatures, and sequentially and gradually filter estimated position candidates with multiple signature-based techniques, each of which uses the output candidates set of the former one as input [2]. The improvement brought about by this cascaded methodology is proved to increase with the number of available sensors increases in general. However, they also observe sound filter sometimes rule out the correct positions, which gives more motivation for us to look into the determinability of the intermediate accuracy before blindly combing them.

Chen et al. [5] combine WiFi and FM signal indicators as one signature. They found the interferences to WiFi and FM signals causing erroneous results happen independently. Thus, using integrated signature almost remove all the errors, drastically increasing the localization accuracy from around 80% up to 98%. We believe this phenomenon is also applicable to other localization techniques, and will bring more benefits if more appropriate intelligence are introduced.

However, no comprehensive and general study has been done for combing multiple IPSs. Rai et al. [15] build an fully-automated indoor localization system called Zee, which features no training phase, and the system converge pretty neatly by combining IMU-based dead-reckoning and WiFi signature-based localizations. Zee uses estimated trajectory to determine the positions on the map, and records the WiFi signatures simultaneously to build the signature database from zero. WiFi signatures in history are used to calibrate the localization methods. This is a good example of

[1] For ease of discussion, "WiFiLoc" and "WiFi RSSI-based fingerprint localization" are used interchangeably in this paper, so as "ABSLoc" and "Acoustic Background Spectrum fingerprint localization".

combining different techniques, but they don't have in-depth investigation on general combinations.

Paiidya et al. [13] introduce the idea of combining different wireless IPSs because they are not avaliable everywhere, and the coverage of collection will provide a more pervasive services. However, they only use averaging without weights and have no discussions about the rationale. Experiments show that improvements of midpoint of results from K Nearest Neighbor (KNN) and triangulation of Bluetooth signal range from $2\% - 52\%$.

In [8], Gwon et al. propose a Selective Fusion Location Estimation (SELFLOC) algorithm and Region of Confidence (RoC) algorithm. Both of them are data fusion methods of IPSs. SELFLOC is essentially a linear weighted averaging calculation. However, they consider little about the nature of the mechanism of IPSs, and therefore don't discuss how and what weights should be assigned to each IPS, let alone the context awareness, which we will describe in this paper. On the other hand, RoC can only be used for location determination during triangulations, which is just a little portion among all IPSs. Similar to SELFLOC, they didn't dive into how to eliminate the erroneous IPSs in triangulations using context awareness. Another problem of triangulations is that universal obstacles in commercial buildings almost preclude accurate line-of-sight distance measurements, which is crucial in triangulations. This is also the primary reason that we don't use triangulations in our WiFiLoc.

David et al. [1] developed a decentralized multi-sensor target tracking system with a distributed information flow and data fusion among the participating the platforms. They analyzed three data fusion techniques associated measurement fusion, tracklet fusion and track to track fusion. Of these techniques, the track to track fusion is of great interest to us. Track to track data fusion technique is used to combine the data from multiple sensors assuming trackers on all platforms use the same information to start with at same time t_0. We use similar approach to indoor localization however with a centralized server for combining data for multiple tracks.

3 System Architecture

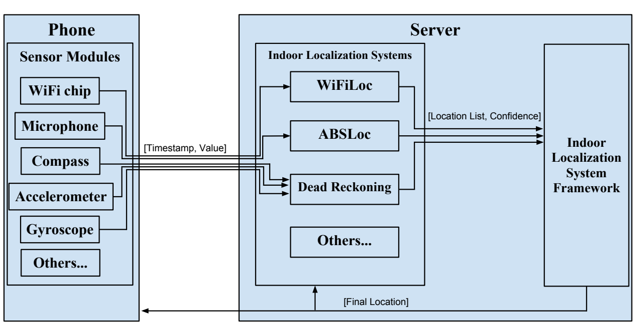


Figure 1. System Architecture

In this Section, we describe the system architecture of the IPS framework in detail. The point is to design a system with both efficiency and extensibility. The efficiency comes from various aspects including hardware, algorithm, data transmission, etc. We leave the efficiency as issues to each of these elements, and emphasize on the extensibility. As a framework, it should require as little overhead as possible for a new module who wants to plug in. Fig. 1 shows the overall architecture of our systems. We have sensor modules that are available on most off-the-shelf smartphones. All data from the smartphones are pushed to or pulled by different IPSs. And all IPSs provide their list of most possible locations and their confidences to the IPS framework. The framework determine and return the final results back to smartphone.

3.1 Sensor Modules

Sensors in Fig. 1 are most commonly available in commercial smartphones. But we don't limit the range of sensors or devices we can use. They are just data sources of specific IPSs, and have little to do with the framework that only looks at the output from IPSs. For example, one can install magnetic signal beacons and have user carry a pre-registered magnetic receiver [10]. One important idea here is to isolate the sensor modules from IPSs. IPSs can get raw data, but don't need to get into details of sensor module implementations. Some protocols have been proposed to abstract heterogeneous stream/time-series data [7]. But these are out of the range of this paper.

Same type of sensors from different manufactures may have significant differences with respect to performance, such as consistency, accuracy, speed, and durability. But this is not the worry of the framework either. As we will discuss later, the framework maintains a history of performance of different IPS systems, so the error-prone sensors or devices will lose the credits and have little impact on the final results.

3.2 Indoor Positioning Systems

In spite that IPSs are depicted as running on server in Fig. 1, practically there should be no restrictions on the implementations. Currently we implemented WiFiLoc and ABSLoc in Matlab on server. But we can totally distribute them. In particular, we may put the database of fingerprints on other servers if the size of database increases. Smartphone can also do preliminary computations to refine the data before sending them to server for fine grained fingerprint matching, and hence reduce the network throughput. For example, it can use last outdoor GPS readings to decide the part of building it should be, and only look at the fingerprints around that area.

WiFiLoc takes the set of RSSI values from all WiFi APs as a signature.[2] Signatures on selected locations, which we call fingerprints, are collected in advance and stored in database. When a user or application query the estimated location, WiFiLoc grabs the current signature, and calculates distances between it and fingerprints in database. The distances could be any appropriate distances that can be applied to an array of values[3], such as Euclidean distance or Manhattan distance. Basically, the corresponding location of the most similar fingerprint will be regarded as the estimated location. Alternatives may take the list of near fingerprints and their locations, and use the mean value of the location list as

[2] There are other IPSs that use signal attenuation to estimate distances from user and APs, and perform triangulation on the map to do positioning

[3] Depending on the nature of the array of values, it can, but not restricted to, be a vector in a vector space.

final estimation. This process is called KNN averaging, and is commonly used in fingerprint-based IPS.

ABSLoc [16] is also a fingerprint-based IPS. The difference between ABSLoc and WiFiLoc is the nature of fingerprint. Based on the idea that each room in a building has its unique pattern of background noise, thanks to the uniqueness of environment and sound attenuation through walls. ABSLoc extract the frequency spectrum of background noise as the fingerprint for each room. In our prototype, we produced fingerprints for points rather than rooms, because it will not reduce room-level localization accuracy and it simplifies data fusion. We will come back to this in Section 6.

Dead-reckoning is to estimate current position based on the known start point and the velocity, time, direction along past trajectory. With richer availability and better performance of IMU on smartphones, we can measure the speed and orientation of the user, and perform a dead reckoning calculation. The downside of dead reckoning is that it subjects to cumulative errors. So the calibrations with help from other IPSs appears very important. In our system, the calibrations happen when the framework returns the final location to all plugged-in IPSs.

In addition to the original designs of these IPSs, we propose each of them to implement a confidence calculation mechanism, in which they can use sensor data to learn the context and analyze the intermediate or ultimate results, to indicate their confidences about the accuracy of their estimations. More details will be discussed in Section 4.

Similar to sensor modules, IPSs are isolated from the framework. In particular, IPS framework use one generic interface with all IPS and obtain estimated location list and confidence from each of them. Hence, any detail of IPS is hidden from framework. However, the manifestations of locations varies. They can be room numbers, or coordinates based on the map. The framework needs to understand both of them and handle the differences. The details will be discussed in Section 5.1.3.

3.3 Indoor Positioning System Framework

IPS framework is in charge of 1) collecting estimations and confidences from IPSs, 2) maintaining a credit based on the history performance of each entity of IPSs, and 3) figuring out a good way to fuse data from multiple sources to generate a final estimated location. We discuss each of them as follows.

Estimation and confidence collections could be either event-driven or command-driven, which means IPSs can report to framework on locations updates, or framework can retrieve these information from IPS on demand. We implemented out prototype as the latter case, so estimations from both IPSs have the same timestamp, and in turn simplify the comparisons between them in Section 6.

To reduce the impacts from ill-performed entities of IPS resulted from various reasons, including malfunctioning sensors, nasty implementation, or coarse nature of the algorithms, the framework log the performance and evaluate the credits of them. With different requirements for framework, each entity may have a credit for each zone in a building, or even for each time span in a day or a week. The reason is that some IPSs may perform well under specific circumstances. For example, ABSLoc tends to perform better among physically separated rooms than open areas. And WiFiLoc may perform better during early morning when there are less interferences.

There are a bunch of techniques to execute data fusion on multiple locations. The most simple one could be linear weighted average values of each dimension of coordinates. Nevertheless, we can bring more intelligence into this step. For example, Kalman Filter [1] could be used to iteratively eliminate the errors along the trajectory, considering restrictions like pedestrian speed. Moreover, other information, such as physical obstacles and user movement pattern, can also be integrated into the framework. After combing different locations, the framework sends the final decision back to all IPSs as feedback and smartphone as query result.

4 Confidence

Aside from typical localization mechanisms, we advocate all IPSs also implement their mechanisms to estimate their result accuracy, which is the confidence we talked about. We look into confidence in this Section. Because the confidences depend on IPSs themselves, to deliver our ideas, we discuss possible confidence calculation algorithms for the IPS involved in our prototype, namely WiFiLoc and ABSLoc, and extend the discussion a little in other IPSs. To generalize, we require all confidences C_i $(i = 1, 2, 3, ..., n)$ to be real numbers in $[0, 1]$.

4.1 WiFiLoc

As a fingerprint-based IPS, WiFiLoc can employ generic ideas of confidence calculation for fingerprint-based IPS. One of them is to see the signature distance between the sampled signature and nearest fingerprint in database. Smaller nearest distances ought to yield higher confidence generally. Furthermore, many fingerprint-based IPSs average more than one locations with near enough fingerprints to avoid extreme errors in estimation. Hence, the sparsity, or the variances of each dimensions in the location list, also imply the potential accuracy of the estimations. It could be more confident if its location list converge on the map.

On the other hand, IPS will potentially provide more precise confidence if it takes relevant context into consideration. For WiFiLoc, one can determine whether the signal is consistent at one position by recording multiple signatures. It can be done during the the database building phase, in which the system administrator collects takes charge. But a more effortless way is to have the users' application check the signal consistency when little movement in IMU is detected (i.e. the smartphone is halting at one point). Other indicators of confidence include the background noise or interferences, and the network delay, and numbers of users for each AP. These information are all observable with existing infrastructure, but how exactly they influence the accuracy of the estimations require more in-depth research.

Moreover, if the designer and implementer go further, and make WiFiLoc maintain a history of the consistency of each AP, or even their performances under different circumstances (e.g. numbers of clients, interferences, network throughput), it may perform better context awareness, and provide much more accurate confidences. These details is hard to be considered in the framework, therefore the responsibility is on each IPS.

4.2 ABSLoc

ABSLoc is also a fingerprint-based IPS, therefore the generic ideas described for fingerprint-based IPS before are applicable as well for ABSLoc.

Apparently, ABSLoc can also take context information as input to confidence calculation. If it can identify open areas and separated rooms areas in the building, which may require input from users or administrator, it can be less confident if most of the nearest fingerprints are located among open areas. ABSLoc can also log its performances based on the feedbacks from the framework at each positions, and tag the positions where it always yield inaccurate estimations as open areas. One point to emphasize here is that, the idea for history performances in framework and in each IPS are different. For framework, it has not idea about the reasons that cause the coarse estimations in IPS, which could be either bad hardware or software, especially if the framework doesn't have history with fine enough granularity in terms of time and space. However, if these inaccuracy could be identified by IPSs themselves by low confidences before sending to framework, it avoids them harming their credits, and preserves their opportunities to contribute to the whole system.

According to the evaluation part in [16], ABSLoc only work properly in relative quiet environments. Fortunately, we can determine whether the audio record is noisy by audio signal processing features, such as its variances and mean value. Based on the noisiness of the record, ABSLoc can definitely indicate higher confidence if the sound is quieter, compared to a preset performance threshold.

4.3 Other Indoor Positioning Systems

Aside from incomplete confidence calculation methods above, other IPSs also have great potential in predicting there result accuracy using confidence. Take Dead Reckoning as an example, we can infer from [15, 11] that their IMU-based trajectory estimations only work with "normal" movements of phones, which is a limited set of pre-defined patterns to which all movements have to be matched. But we actually cannot guarantee that all users will use phone "normally". It is possible that the user swings her or his phone during gaming, or have weird movements when they are shooting a video using phone. Those anomalies could actually be detected. Fortunately, most IMU-based IPS have a possibility of their estimation accuracy, which tends to be low with "abnormal" situations, but few systems make use of that.

Potentially there are numerous ways to compute a confidence. Theoretically, the better we understand the nature of the IPS and related hardware and software, the more accurate confidence calculation algorithm we could come up with. However, perfect confidence calculation is not our goal. Our key point is we can make the most of the information we gain, and provide a better positioning results with cooperated efforts from multiple IPSs.

5 Data Fusion

With multiple sources of estimated location list and confidences, the framework takes charge of combing them and generating a final decision of location. Too achieve this, it needs to judge and weight between multiple proposes from IPSs. Unfortunately the confidences provided along with location lists by IPSs can only act as clues rather than determinants, because of different understandings, implementations, and natures among IPSs. We propose an empirical method, which keeps a credit based on performance history for each IPS entity. After that, we look into ideas of fusing multiple location estimations, to finalize more practical results in the sense of reasonable human movements. We use Kalman Filter as our key solution for data fusion, which is proved to fulfill our expectations as we will describe in Section 6.

5.1 Preparation

5.1.1 *Performance History and Credit*

The framework keeps track of all plugged-in IPSs, and logs their performances to build their credits. Basically, if one IPS always provides accurate estimations, which means the estimations have small distances to the final decision of the framework, when they meanwhile gives high confidences, it gains credits. Low confidences don't help gain or loss credits. And erroneous estimations with high confidence will harm the credits. Note that a bad estimation coming with high confidence will not determinably ruin the final decision, because the framework will not only look at on confidences, but other parameters like credits and human movement restrictions.

One important functionality that the framework has is to figure out what granularity each credit should represent in terms of time and space. As we described in Section 3.3, one IPS entity may perform quite differently at different places and time, because of their specific assumptions on indoor environments. For instance, ABSLoc assumes separated rooms in buildings, but open areas like cubicles are actually not uncommon. To make sure the bad performances of an IPS under inappropriate circumstances don't prevent it from contributing in their adept scenarios, the framework can use clustering techniques, such as k-means, to divide an entity into several ones with several credit records.

5.1.2 *Weight Assignment*

Weights represent to what extent the framework trust the estimations from an IPS. As weights are directly used for estimations combination, so we require them to be normalized, which means all weights W_i $(i = 1,2,3,...,n)$ for n IPSs meet the condition:

$$\sum_{i=0}^{n} W_i = 1,\ W_i > 0 \tag{1}$$

There are also several ideas to assign weights. Most simple one should be multiplying credit and the confidence, and normalizing the results as weight for each IPS. Moreover, as the purpose of weights is to highlight the most possibly accurate result, and reduce the impact from erroneous ones, when the framework detects large sparsity among estimated locations, it may consider amplifying the differences between all weights, to potentially reduce the effects from less-trusted IPS.

5.1.3 *Unify Location Granularities*

Another issue faced by the framework is that different IPSs actually provide location estimations at different granularities. In particular, some IPSs like ABSLoc aim at room-level semantic estimations, but dissimilarly, WiFiLoc only produce coordinates as locations. First of all, the framework has to understand both semantic and physical representatives on the map. We don't dive into the details of map implementation in this paper. As far as we are concerned, the incongruity of location manifestations forces the framework to compromise if it wants to use dominant data fusion techniques. We propose two ways to unify the granularity as follows.

- **Location Granularities Boost:** To convert estimated room from semantic estimations to coordinates, framework can uniformly select points in a room at a proper granularity. However, it is not a good idea to choose the mean value of these points as estimation from one IPS, because it always leads to the central point of the room. Instead, the framework divides these points and regards each of them as estimation from one IPS entity with the same confidence as the original one.
- **Location Granularities Reduction:** Conversely, the framework can also semantically use room where the mean value of the coordinate list from one IPS to represent its estimation. However, combing semantic locations requires different data fusion techniques. We leave this as our future work.

5.2 Fusion Methods

With weights and locations prepared, the framework can combine them and give the final decision of the estimation. We only talk about data fusion methods after using "Location Granularities Boost", which generates coordinates as inputs rather than semantic representatives.

5.2.1 Linear Combination

The most simple fusion is linear weighted average. We denote L_i $(i = 1,2,3,...,n)$ as the mean location of location list from the ith IPS using KNN averaging, and W_i $(i = 1,2,3,...,n)$ as its weight. Then the final estimated location L is

$$L = \frac{\sum_{i=0}^{n} (W_i \cdot L_i)}{n} \tag{2}$$

, where W_i $(i = 1,2,3,...,n)$ are normalized as we described before. However, this combination only follows the mathematical principles without any consideration of human movement restrictions, leaving us plenty rooms for improvements.

5.2.2 Kalman Filter

Compared to linear weighted average, Kalman Filter considers the rationality in the trajectory. We used a simple form of Kalman Filter to combine the data. Kalman Filter provides Bayesian recursive estimate of state space X_k using knowledge of previous state $[X_1, X_2, X_3, ..., X_{k-1}]$ in a linear state space system with Gaussian noise. Since the project focus was on building the framework rather than Kalman Filter itself we used a basic form of Kalman Filter with certain assumptions. Firstly, a persons movement in a building is a linear system where he is moving with a constant velocity.

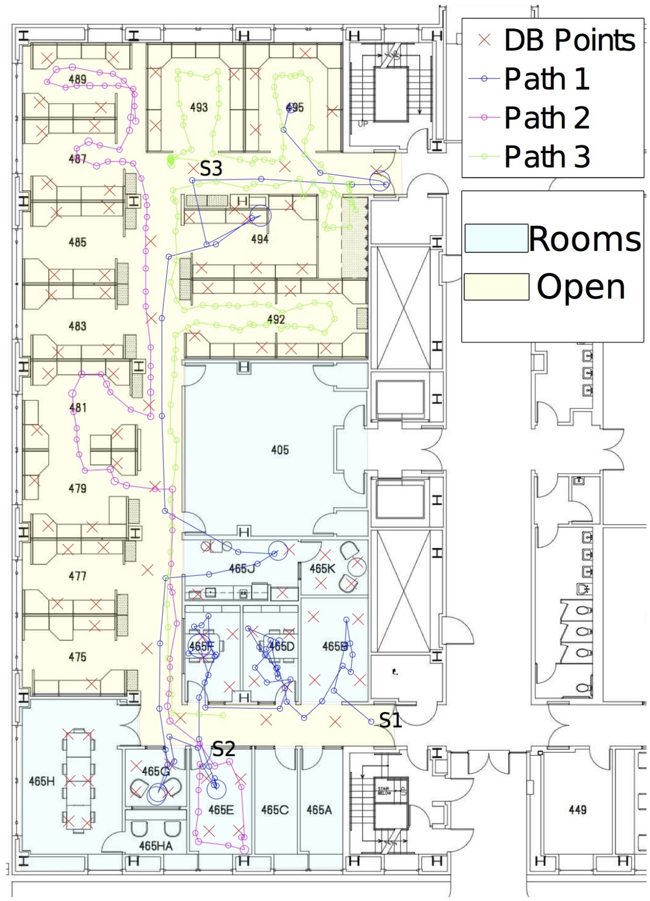


Figure 2. Database Points and Ground Truth Paths

This is a simplest case of a persons movement in building. So a simple variation of Kalman filter would suffice for data fusion. Secondly, the measurements are taken from right top corner of the floor map. Position of both sensors are considered to be fixed at $(0,0)$ position.

Each sensor module feeds the Kalman Filter with location list and confidence value. Another important assumption is that the sensors are synched meaning they provide measurement of target state at same time interval. Using the linear system equations for position estimation and the measurements from sensors Kalman Filter corrects the errors in sensor measurements. We combine the data using weighted average after the estimation. Weights are calculated based on the confidence value supplied to the framework. We plan to extend this work using non-linear Kalman Filters variants such as Extended Kalman Filter (EKF) or Unscented Kalman Filter (UKF).

6 Evaluation

We evaluate our system based on 3 trajectories in university building. Particularly, we firstly look into confidences about the results for each technique, namely WiFiLoc and ABSLoc. Results shows that each technique can provide fairly accurate confidence on their estimation accuracy. Thereafter, we analyze the accuracy improvements of different combination methods given the estimated points and confidences. We employ simple weighted average method as well as Kalman Filter. Thanks to the error deduction feature of Kalman Filter, it can remove erroneous spikes in all estimated paths. Note that we emphasize on the error reduction ability of our framework, rather than accuracy improvement, which may be the goal for each technique themselves. In other words, the framework are trying to avoid trusting and using, but not able to reduce, the errors that inherently exist in sensor data and localization techniques.

6.1 Experiment Setup

We implemented WiFiloc and ABSLoc, both of which are comprised of a training phase and a localizing phase. All training and localizing data are collected in advance, and post-processed in MATLAB 2012b.

In our WiFiLoc, we wrote an application for LG Revolution VS910 with Android 2.3.4 to collect WiFi RSSI. Every entry in raw data is a $< MAC\ address, RSSI >$ tuple. Each WiFi signature is a list of tuples. And the application collects RSSIs for each available AP around every 800 milliseconds, which in turn forms a signature. In WiFiLoc database, every fingerprint is combined with a coordinate on the map. And new signature are compared wit all fingerprints, and in turn result to a most-possible coordinate list.

For ABSLoc, we used the default sound recorder in Apple iPhone 5 with iOS 6.01, which yields .m4a audio files. The record files last more than 60 seconds for each point. To make the audio file easy to analyze, each of them is converted to .wav files in Audacity (Specifically, the file type is "WAV (Microsoft) signed 16 bit PCM"). However, instead of room-level semantic localization, we also bind a ABS fingerprint in ABS database to a coordinate for two reasons. First, using fine-grained doesn't harm the accuracy at room level, because accurate estimated location based on coordinates can also be converted to accurate room estimation. Second, using coordinates for both WiFiLoc and ABSLoc greatly simplifies and reduces our data fusion efforts.

6.1.1 Training

In training phase, we collects both WiFi and ABS signatures at specific points in RADLab. The 98 red cross markers in Fig. 2 indicate the locations of training data. For every point, we record more than 60 seconds audio signal and WiFi RSSI values. The binding between WiFi or ABS fingerprint and coordinate on the map are stored in WiFiLoc database and ABSLoc database respectively. The time spent on collecting all data in database spans one week.

6.1.2 Localizing

Fig. 2 depicts the 3 paths we conducted for evaluation. Every marker on the path indicates both a sampled WiFi and a sampled ABS signature. There may be signature sampling happened between two consequent markers. The size of marker represents the time the user spent at that point. Both path 1 and path 2 start from the bottom part of the map, whereas path 3 starts from the top part. Start points are tags as S1, S2, and S3 in Fig. 2 respectively. To help discussions in Section 6.2 about how the environments influence ABSLoc's performances, we also mark blocked rooms and open areas on the map.

With the signatures sampled along the path, WiFiLoc and ABSLoc start positioning these signatures as we described in Section 3.2. We use Euclidean distance as signature distance. In WiFiLoc, the default value for RSSI is set as $-150\ dBm$ if it is missing in either signatures compared. Among the coordinates sorted by their corresponding signature distances, WiFiLoc and ABSLoc set a threshold

$$T = \alpha \cdot D_{min} \tag{3}$$

, where D_{min} denotes the minimal signature distance, and select all coordinates whose corresponding signature distances are less than T as estimated location list. Currently, we only use the sparsity of this location list to determine the confidence for both WiFiLoc and ABSLoc. The more the location list converges, the higher the confidence goes. In particular, the confidence C_i is

$$C_i = \frac{1}{2 \cdot e^{\frac{Std_x}{StdBaseline_x}}} + \frac{1}{2 \cdot e^{\frac{Std_y}{StdBaseline_y}}} \tag{4}$$

, in which Std_x and Std_y denote the standard deviations on each dimension of coordinates, and $StdBaseline_x = 100\ (inch)$ and $StdBaseline_y = 100\ (inch)$. Those numbers are empirically set, and are possible to be tuned for a better performance. However, that is not our emphasis in this paper.

The framework obtains their location lists and confidences, and assigns normalized weighted W_k based on C_i ($i = 1, 2, 3, ..., n$) using following formula,

$$W_k = \frac{(C_k)^3}{\sum_{i=0}^{n} (C_i)^3} \tag{5}$$

The purpose of using 3rd power computation is to enlarge the differences between IPSs with relatively large differences in confidences. In other words, it is used for highlighting the higher confidences between WiFiLoc and ABSLoc. With weights assigned, the framework applies both linear weighted average and Kalman Filter to the mean coordinates of both location lists from WiFiLoc and ABSLoc.

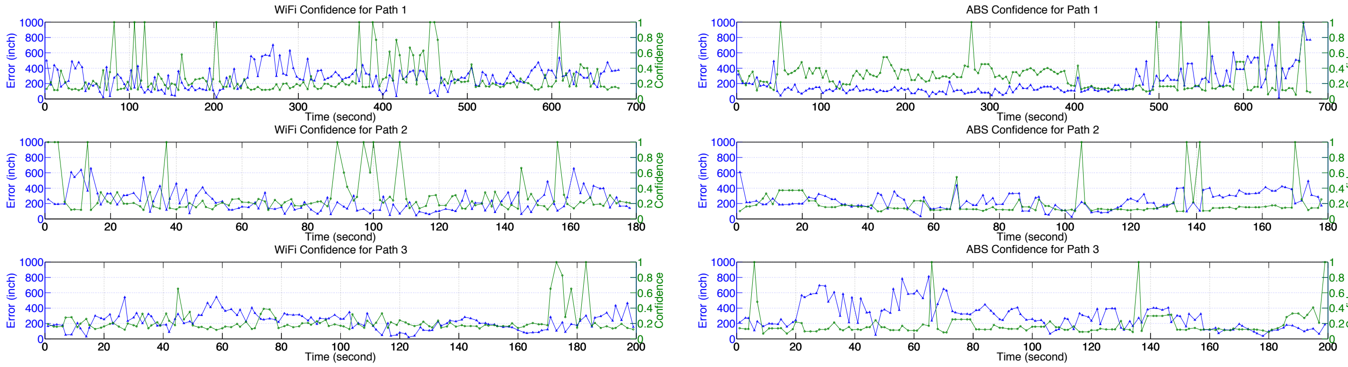
We thereby evaluate this prototype with respect to the confidence accuracy, data fusion methods, and other selected parameters, such as fingerprint number at each point, and total points number in database.

6.2 Confidence Accuracy

Empirically, we select 3 WiFi fingerprints and 1 ABS fingerprint at each position in database. And we choose $\alpha = 1.3$ for both WiFiLoc and ABSLoc.

Fig. 3 depicts the confidences and errors, which is the physical distance from estimated location to the ground truth location, for WiFiLoc and ABSLoc in all 3 paths. The x-axis is the timeline along the path, and every marker corresponds to a marker in Fig. 1. For path 1, it samples a WiFi and ABS signature nearly every 4 seconds, and path 2 and 3 sample every 1 or 2 seconds. We eliminate the samples between adjacent markers along the curve, to ensure the results clear as well as representative.

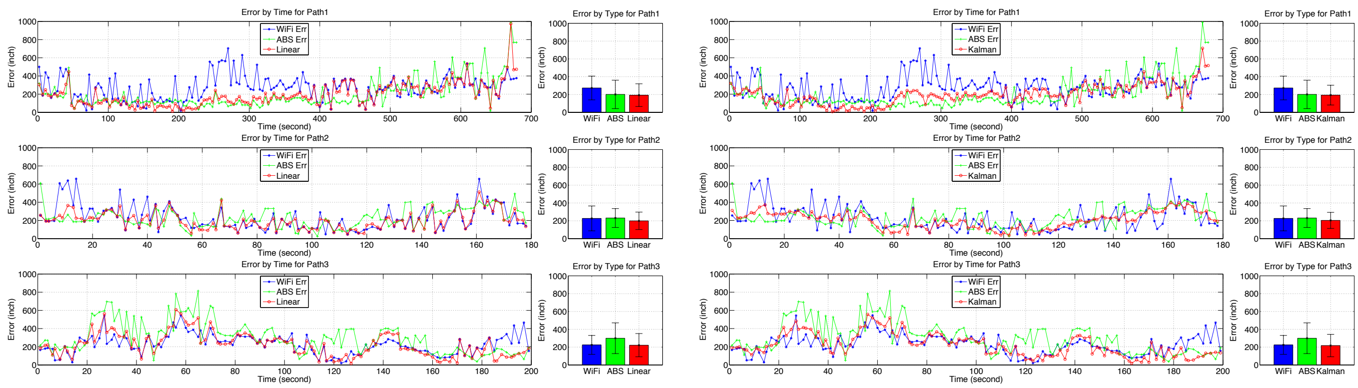
According to Fig. 3, in most cases when the confidence is high, the errors are low, with only one exception in AB-



(a) WiFiLoc Confidence

(b) ABSLoc Confidence

Figure 3. Confidence Accuracy of WiFiLoc and ABSLoc



(a) Linear Weighted Average

(b) Kalman Filter

Figure 4. Performance Improvements after Using Linear Weighted Combination and Kalman Filter

SLoc at the end of path 1. Hence, the framework should trust high confidence, and lean the final decision to the very confident IPS. However, high confidences don't happen frequently, but fortunately they happen independently, which means we can potentially have at least one confident IPS every time if there are enough number of them. We can hardly prove theoretically their independence, but intuitively WiFi signal should have nothing to do with acoustic background noise. Experiments can be conducted to empirically prove the mutual independence, such as [5].

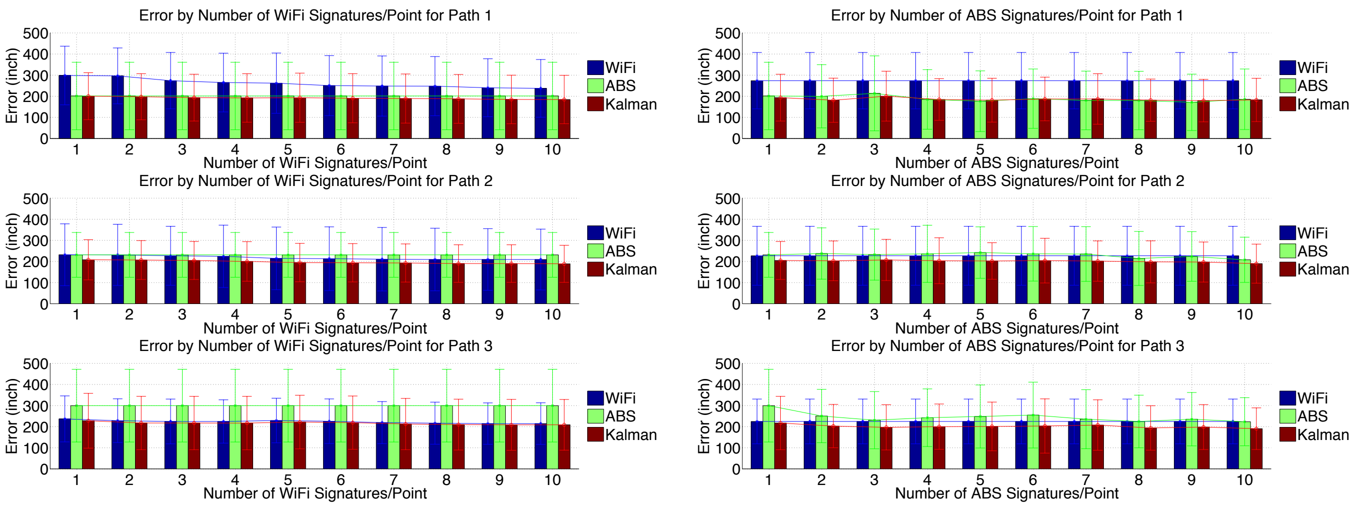
Another importance observation from path 1 in Fig. 3(b) is that ABS works well among blocked rooms, which ends around 470 seconds after the start of the trajectory, and suddenly varies drastically after the user entering the open area. Meanwhile, the confidences of ABSLoc are also generally relatively higher in separated rooms than in open area. This phenomenon is not obvious in path 2 and 3 because they are almost always in open area. If the open areas and blocked areas are tagged on the map, by either crowdsourcing from users or manual operations from administrator, we can give ABS a better confidence computation compared to only using location list sparsity.

6.3 Data Fusion Methods

We depicts the performance of linear weighted average and Kalman Filter in all 3 paths in Fig. 4. The same as Fig. 3, the x-axis is the timeline of paths, but the y-axis only shows the errors. On the right part of each small summary figure is the mean value and standard deviation of WiFiLoc, ABSLoc and the final location from data fusion method.

We firstly look at the small summary figures in Fig. 4. As we can see, ABSLoc is more accurate than WiFiLoc in path 1, because path 1 is mostly in separated rooms, but performs worse in path 2 and 3, where the users are mostly in open areas. So it is more clear here that the environment impacts IPS as we suppose in theory. Most importantly, no matter which IPS performs better, and no matter what combination data fusion method is employed, the framework can almost give final results that have similar accuracy as the better one among WiFiLoc and ABSLoc. The final nearly 200-inch errors result from the inaccuracy of each IPS themselves, for which the framework has no responsibility to improve.

If we go into more details, we can see both linear weighted average and Kalman Filter tend to have final decisions with errors near the more accurate one among IPSs, especially in the first half part of path 1. However, at the end of path 1, because ABSLoc has both high error and high confidence, the linear weighted average method generates a sharp spike at that point. But as Kalman Filter takes the human movement patterns into consideration, we can see it reduces the spike by 50%. Moreover, Kalman Filter eliminates most



(a) WiFiLoc Fingerprint Number

(b) ABSLoc Fingerprint Number

Figure 5. Error Trends by the Number of Signatures for Every Point in Database

spikes existing in linear weighted average results. Therefore, we can conclude that Kalman Filter ties linear weighted average in average accuracy improvements, and performs better in terms of spike avoidance.

6.4 Fingerprint Number Per Point

For any one of the 98 locations in database, we collected multiple fingerprints. In WiFiLoc, 60-second log generally contains more than 70 signatures. And a 60-second acoustic record in ABSLoc can also be divided to several small frames. Because we collect the ABS data all in quiet environments, the background noise can also be extracted from a even 5-second frame. However, using all fingerprints for every location appeared to be a bad idea on account of the unbearable response time. For example, if we use 10 WiFi fingerprints at each location, it will take WiFiLoc up to 10 seconds to localize one signature. So we vary the number of fingerprints per point, and generate Fig. 5.

In Fig. 5, the x-axis is the number of fingerprints at each location in database, and y-axis is estimation errors. The mean value and standard deviation of errors of WiFiLoc, ABSLoc, and results of Kalman Filter are all shown for each x-axis value. In Fig. 5(a), the ABS fingerprints number per point is 1. And in Fig. 5(b), the WiFi fingerprints number per point is 3. These are just empirical values.

Based on Fig. 5, neither WiFi nor ABS fingerprints numbers has obvious influence on the performances. Some subtle exceptions happen when WiFi signatures number changes from 2 to 3 in path 1 in Fig. 5(a), and when ABS signatures number changes from 1 to 2 in path 3 in Fig. 5(b). However, on the whole we cannot see any trend at least in our deployment. So it actually doesn't matter how many fingerprints we have for each point, let alone that the framework is not focusing on improving the accuracy for each IPS.

6.5 Points Number in Database

To figure out how many location do we need to build a sufficient database forRADLab, we remove data from database, and plot the trends of errors in Fig. 6. Note that

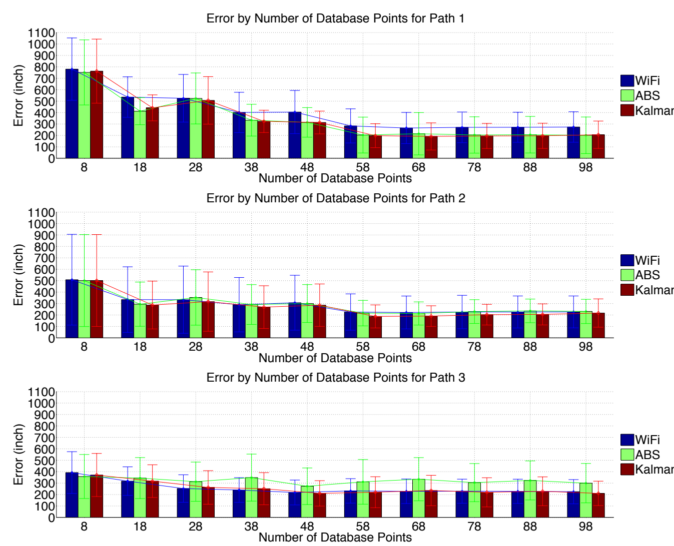


Figure 6. Error Trends by the Number of Database Points

we only change signature number per point for either IPS in Fig. 5, but we change the database point number for both of them every time in Fig. 6. The reason is that we don't want to observe the trends influenced by each IPS in the former case, but removing one point in database leads to removing both WiFiLoc and ABSLoc records at the same time. When removing points from database, we make sure removed points are physically evenly distributed on the map.

In Fig. 6, we can see that number of database points influence the performance when it is less than a threshold, which is 58 in this case. In other words, we over-sampled fingerprints in RADLab by nearly $(98-58)/58 \approx 69\%$.

7 Future Work

Based on our current prototype, we plan to go steps further and build a complete and easily deployable indoor localization system. To fulfill this, we plan to add more IPSs to our system, such as dead-reckoning, and infrastructure-based beaconing system like infrared and ultrasound. For most IPSs, there are still many details we can look at that contains huge rooms of improvements. For example, among all points and their fingerprints stored in database, we can look at how to choose the least and optimized ones based on features of signature themselves, like signature distance over real distance, to provide both fast and accurate fingerprint-based localization system.

Another important functionality we should primarily focus on is how to implement room-level semantic localization, and how to do data fusion on semantic level. One information that should be considered is physical restrictions of human movements, such as wall, to understand the data with more intelligence.

After all systems working well, we may consider implementing automatic database building, which has been proposed and implemented in [15]. Currently, there are still insufficient work on this, which is crucial to the system deployability in a building, especially those without skilled administrator.

To go further, automatic building generation is also interesting and promising. [15] requires users to input a building map, which may be a huge adversary for users to use the system. Crowdsourcing techniques can be investigated to achieve this goal.

8 Conclusions

With more and more indoor positioning techniques springing up, there are still no consensus on a generic solution. There is a huge gap between this fact and that all proposed IPSs are proven to work acceptably well with appropriate assumptions or under specific circumstances. However, their assumptions and circumstances vary a lot, and mostly independent with each other. On the other hand, plenty sensors are involved in each IPS, which actually can be used not only for localization, but for context awareness.

In this paper, we propose an IPS framework, which combines multiple localization systems, and takes charge of deciding a final estimated location. Each IPS are responsible for providing a confidence about their estimation as accurate as possible. The framework manages all plugged-in IPSs, builds credit history for them, assigns weights to each proposed estimation, and uses proper data fusion techniques to make the decision.

We deployed the prototype in a university building, which contains WiFiLoc, ABSLoc, and a framework. Experiment results show that IPSs can generate trustable confidences, which in turn benefits the data fusion. The framework can always generate decision with nearly same errors as the better IPS. Compared to simple linear weighted averaging method, Kalman Filter outperform in the sense of reducing drastic errors. In summary, there are still huge potential in the old indoor localization area. As future work, we plan to go deeper into a bunch of dominant IPSs, and build a plug-and-play indoor positioning system that is applicable in any building.

9 References

[1] D. Akselrod, A. Sinha, and T. Kirubarajan. Information flow control for collaborative distributed data fusion and multisensor multitarget tracking. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(4):501–517, 2012.

[2] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 261–272. ACM, 2009.

[3] P. Bahl and V. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. IEEE, 2000.

[4] H. Chang, J. Tian, T. Lai, H. Chu, and P. Huang. Spinning beacons for precise indoor localization. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 127–140. ACM, 2008.

[5] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. Fm-based indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 169–182. ACM, 2012.

[6] Y. Cheng, K. Chen, B. Zhang, C. Liang, X. Jiang, and F. Zhao. Accurate real-time occupant energy-footprinting in commercial buildings. 2012.

[7] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler. smap: a simple measurement and actuation profile for physical information. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 197–210. ACM, 2010.

[8] Y. Gwon, R. Jain, and T. Kawahara. Robust indoor location estimation of stationary and mobile users. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1032–1043. IEEE, 2004.

[9] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. Global positioning system. theory and practice. *Global Positioning System. Theory and practice. J.. Springer, Wien (Austria), 1993, 347 p.*, 1, 1993.

[10] X. Jiang, C. Liang, K. Chen, B. Zhang, J. Hsu, J. Liu, B. Cao, and F. Zhao. Design and evaluation of a wireless magnetic-based proximity detection platform for indoor applications. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, pages 221–232. ACM, 2012.

[11] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. 2012.

[12] L. Ni, Y. Liu, Y. Lau, and A. Patil. Landmarc: indoor location sensing using active rfid. *Wireless networks*, 10(6):701–710, 2004.

[13] D. Pandya, R. Jain, and E. Lupu. Indoor location estimation using multiple wireless technologies. In *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, volume 3, pages 2208–2212. IEEE, 2003.

[14] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM, 2000.

[15] A. Rai, K. Chintalapudi, V. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 293–304. ACM, 2012.

[16] S. Tarzia, P. Dinda, R. Dick, and G. Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 155–168. ACM, 2011.

[17] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.

[18] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *Personal Communications, IEEE*, 4(5):42–47, 1997.

[19] B. Zhang, K. Chen, Y. Cheng, C. Liang, X. Jiang, and F. Zhao. Location-log: Bringing online shopping benefits to the physical world with magnetic-based proximity detection. 2012.