Zee: Zero-Effort Crowdsourcing for Indoor Localization

Anshul Rai[†], Krishna Kant Chintalapudi[†], Venkata N. Padmanabhan[†], Rijurekha Sen[‡*]
[†]Microsoft Research India [‡]Indian Institute of Technology, Bombay

ABSTRACT

Radio Frequency (RF) fingerprinting, based on WiFi or cellular signals, has been a popular approach to indoor localization. However, its adoption in the real world has been stymied by the need for site-specific calibration, i.e., the creation of a training data set comprising WiFi measurements at known locations in the space of interest. While efforts have been made to reduce this calibration effort using modeling, the need for measurements from known locations still remains a bottleneck. In this paper, we present Zee – a system that makes the calibration zero-effort, by enabling training data to be crowdsourced without any explicit effort on the part of users.

Zee leverages the inertial sensors (e.g., accelerometer, compass, gyroscope) present in the mobile devices such as smartphones carried by users, to track them as they traverse an indoor environment, while simultaneously performing WiFi scans. Zee is designed to run in the background on a device without requiring any explicit user participation. The only site-specific input that Zee depends on is a map showing the pathways (e.g., hallways) and barriers (e.g., walls). A significant challenge that Zee surmounts is to track users without any a priori, user-specific knowledge such as the user's initial location, stride-length, or phone placement. Zee employs a suite of novel techniques to infer location over time: (a) *placement-independent step counting and orientation estimation*, (b) *augmented particle filtering* to simultaneously estimate location and user-specific walk characteristics such as the stride length, (c) *back propagation* to go back and improve the accuracy of localization in the past, and (d) *WiFi-based particle initialization* to enable faster convergence. We present an evaluation of Zee in a large office building.

Categories and Subject Descriptors

C.2.m [**Computer Systems Organization**]: COMPUTER - COMMUNICATION NETWORKS—*Miscellaneous*

Keywords

Indoor localization, WiFi, crowdsourcing, inertial tracking

*The author was an intern at Microsoft Research India during the course of this work.

MobiCom'12, August 22–26, 2012, Istanbul, Turkey.

1. INTRODUCTION

RF fingerprinting of WiFi signals is a popular approach to indoor localization. Typically, there is an initial training or calibration phase during which received signal strength (RSS) measurements from multiple WiFi access points are recorded at known locations. Then, when a device is to be located, RSS measurements from proximate APs are matched against the training data, either deterministically [3] or probabilistically [32], to estimate location.

The need for calibration is a key bottleneck since it is labour-intensive. Further, it needs to be repeated for each new space and also every time there is a significant change in the space (e.g., when new APs are added or existing ones repositioned). While efforts have been made to reduce the calibration effort using RF modeling, these suffer from various limitations, including the need for at least some data from known locations [9], the need for control over the APs and knowledge of their locations [11], and loss in accuracy because measurements are made at fewer points than ideal to save effort. These limitations also come in the way of a crowdsourcing-based approach to training because, for instance, on a mall floor, the locations of APs installed by multiple providers and stores would not be known, and obtaining a GPS lock might not be feasible at any location.

In this paper we enable *zero-effort crowdsourcing* of WiFi measurements in indoor spaces by developing a system called *Zee* (name derived from the first syllable of "zero"). Our vision is of users carrying smartphones who walk around in the indoor space of interest in normal course (e.g., stroll through a mall), with each user traversing a subset of the paths in the space. We do not assume knowledge of where within the space a user walks or even the starting point of the user's walk. As well, we do not assume knowledge of the placement of a user's smartphone, i.e., whether it is in their hand, shirt pocket, bag, or elsewhere, which also means that we do not know the orientation of the phone relative to the user's direction of motion. All of these elements accord well with the needs of crowdsourcing, where little can be assumed about users, and explicit input or other action from users is best avoided.

The only external input that Zee depends on is a map of the indoor space of interest, which we do not view as onerous since a map would be needed anyway for the purposes of location-based applications such as navigation. Armed with just the map, Zee uses WiFi and inertial sensor measurements crowdsourced from the users' smartphones to automatically infer location over time and thereby construct a WiFi training set (i.e., WiFi RSS measurements annotated with location information).

The key idea behind the automatic inferencing of location in Zee is to combine the sensor information with the constraints imposed by the map (e.g., that a user cannot walk through a wall or other barrier marked on the map), thereby filtering out infeasible locations

over time and converging on the true location. As an example, the inertial sensors such as accelerometer and compass might indicate that the user walked in a zigzag path, taking a certain number of steps in a certain (unknown) direction, then turning 90° to the right and continuing the walk, and finally turning 90° to the left to take a few more paces and then stopping. While the above information does not, by itself, reveal location, it could when viewed together with a floor map. For instance, the map might indicate that there is only one pathway on the floor that could accommodate the kind of zigzag trajectory observed, say the path from the entrance of a mall to a particular store. Thus, at the conclusion of the walk, we can infer that the user's ending location must be the store, and then we can trace back and infer that their starting location must have been the entrance.

To codify the above intuition in Zee, we incorporate the uncertainty arising from sensing and the constraints imposed by the map, into a novel *augmented particle filtering* framework. Whereas particle filtering in the context of localization has typically used particles only to represent the uncertainty in location, we create multi-dimensional particles that also incorporate the uncertainty in other aspects such as the stride length of a user and their direction of walk. Augmented particle filtering then enables the estimation of these latter variables concurrently with the estimation of location. To speed up the convergence of particle filtering, we use two techniques to estimate better priors for the variables being estimated: *placement-independent motion estimation* to estimate the step count and the approximate orientation (or heading offset) of a device relative to the direction of walk, and *WiFi-based initialization* to leverage partial WiFi information to make an initial guess of the location(s) where a device might be. Finally, since the uncertainty in location would tend to reduce with time as a user takes a longer walk with more turns, we use *backward belief propagation* to take advantage of the greater certainty in location at a later point in time to trace back and reduce uncertainty in location at earlier times, post facto.

Concurrently with estimating location, Zee performs WiFi scans and records the results indexed by time. As and when the location estimate for a particular time becomes available, the corresponding WiFi measurement is annotated with the estimated location, thereby adding a record to the WiFi training set. Thus, Zee provides a way to crowdsource WiFi measurements without requiring any explicit effort on the part of users. To evaluate the quality of the crowdsourced training data set, we feed it into Horus [32], a well-known WiFi fingerprinting-based localization technique, and EZ [9], a newer modeling-based technique. We find that with the crowdsourced training data set, Horus and EZ achieve a median localization error of about 3m, which is comparable to the median localization error of 3.5m achieved with a training data set that is explicitly measured.

Thus, Zee offers a truly zero-effort solution for crowdsourcing WiFi data for the purpose of indoor localization, by leveraging the walks that users take through the space of interest in normal course. We view this as a significant contribution of our work, one that could be a key enabler of WiFi-based localization in real-world settings. As well as providing a way of constructing a training data set for later use, another key contribution of Zee is a way to perform accurate tracking of a walking user for the purposes of real-time applications such as indoor navigation.

2. BACKGROUND AND RELATED WORK

Zee draws on prior work in multiple areas, chiefly WiFi-based localization, robotic navigation, and inertial sensing.

2.1 Infrastructure-Based Localization Systems

Early systems required the deployment of special-purpose infrastructure in the indoor space to enable localization. The inability to use GPS indoors has led to myriad approaches based on alternative signals, ranging from infrared [26] to acoustic [27, 19] and visual [28]. There have also been localization systems based on a deployment of RF transmitters and sniffers [15] or RFID [18]. While each of these approaches offers certain advantages (e.g., high accuracy in the case of acoustic ranging), the need for special-purpose hardware and infrastructure is a significant challenge.

2.2 RF Fingerprinting based Localization

Localization based on measuring the RF signal of a wireless LAN has the significant cost advantage of leveraging an existing infrastructure. A popular approach, pioneered by Radar [3], is to employ received signal strength (RSS) based fingerprinting of locations in the space of interest, where typically multiple access points (APs) are heard at each location. While Radar used a simple, deterministic fingerprinting and matching scheme, Horus [32] developed a more sophisticated and accurate approach wherein the RSS measurements corresponding to each location and AP are represented as a probability distribution and matching is performed using the maximum likelihood criterion. SurroundSense [2] extends this idea and builds a map using several features found in typical indoor spaces such as ambient sound, light, color, etc., in addition to WiFi RSS. Several other improvements over and extensions of the basic RF fingerprinting based localization have been proposed, such as the incorporation of mobility constraints [12] and an extension to outdoor settings [8].

The above approaches depend on calibration of the space of interest to construct a training data set comprising RSS measurements at *known* locations. Such calibration tends to be onerous, more so because it has to be repeated for every new space and each time there is a significant change in a given space (e.g., a change in AP placement). Zee is aimed squarely at eliminating the need for such explicit calibration effort.

2.3 Modeling instead of Calibration

An alternative to empirical calibration is to use an RF propagation model to estimate the RSS, P_x, at a given location x based on the the transmit power, P_0, and the distance, d_{0x} between the transmitter and the location x. A popular model is the log-distance path-loss model, which models the RSS (in dBm) as $P_x = P_o - \gamma log(d_{0x}) + N$, where N represents a noise term [20]. Extensions of this model have incorporated the presence of and the attenuation due to obstructions such as walls.

Radar [3] included a model-based variant, which estimated RSS at various locations using knowledge of the AP locations and transmit powers, and a floor map. Radar also proposed the idea of RF environment profiling [4], where measurements between the (stationary) access points is used to characterize the changing RF environment. This idea was leveraged in [17], to develop a zero-configuration localization system, where APs make RSS measurements, with respect to clients and also each other. The measurements made between the APs are used to construct a model that maps RSS to distance. This model is then used locate clients through trilateration. This scheme, however, requires the AP software to be modified and so cannot make use of off-the-shelf APs.

More recently, efforts have been made to use modeling with minimal assumptions. EZ [9] only requires measurements at a few known client locations and WiGem [11] only requires knowledge of AP locations and the ability to measure the client signals as re-

ceived at the APs. The reduced measurement effort with a modeling based approach typically comes at the cost of reduced accuracy.

Zee avoids the need for measurement at any known location or any knowledge or control over AP locations and measurements. This, we believe, makes it much more amenable to a crowdsourcing approach because, for instance, a public space such as a mall might have APs deployed by multiple providers and moreover the absence of GPS coverage might make it challenging to make any measurements at all at known locations. Equally importantly, Zee is able to avoid the loss of accuracy inherent in modeling based on limited data. This is particularly relevant in the context of accurate but measurement-intensive techniques such as Horus [32].

2.4 Alternatives to RSS-based Localization

Several alternatives to RSS have also been considered. In particular, detailed physical layer information has been used to fingerprint both devices [7, 5] and locations [23]. We view the contribution of Zee as being orthogonal to these; the above non-RSS-based approaches also require a training set to be constructed and could benefit from the zero-effort calibration made possible by Zee.

On the other hand, there has also been research on leveraging non-RSS information from RF beacons in a way that does not require any calibration. For instance, [13] simply estimates a client's location as the centroid of the known locations of the APs heard, without regard to the RSS, which leads to loss in accuracy. There has also been work on leveraging time of flight [1] or angle of arrival [31, 22] relative to APs, whose locations are assumed to be known. However, these approaches either require specialized and potentially expensive hardware [1, 31] or require special human effort, e.g., taking a spin while walking [22]. In Zee, we avoid these disadvantages, making our approach more amenable to crowdsourcing albeit with reduced accuracy compared to the approaches based on precise measurement.

2.5 Robotic Navigation

In the robotics community, there has been much work on the Simultaneous Localization and Mapping (SLAM) problem, which dates from the mid-1980s [24, 16]. A robot equipped with sensors, such as laser-based ranging and cameras, is assumed to be exploring the space of interest, e.g., an unmapped building. The space is assumed to have landmarks, which are typically artificially inserted (e.g., barcode pasted on walls or a particular pattern painted on the ceiling). The "mapping" problem is to determine the locations of the landmarks relative to each other whereas the "localization" problem is to determine the location of the robot relative to the landmarks. Estimates are updated based on an action model, which helps relate the new location to the previous location (e.g., based on the number of revolutions of the robot's wheels), as well as sensor data (e.g., visual landmarks captured by a camera sensor).

While the early work on SLAM used Kalman Filters for estimation, subsequent work has been based on Markov localization, which is a better match in practice since it allows the robot's position to be modeled as multi-modal and non-Gaussian probability density functions. Of particular interest to us is the Monte Carlo Localization (MCL), or *particle filtering* based, approach [10], wherein the idea is to represent the belief in the robot's location as weighted random samples, or particles. The particles are then evolved based on the action model and the sensor readings.

In Zee, we build on the key idea of particle filtering by augmenting particles to incorporate not only location but also other unknowns such as the stride length of a particular user. However, in other respects, Zee differs from the SLAM approach. In Zee, it is humans, not robots, that are moving around in the space of interest. This means that we have to contend with all of the complexities associated with human locomotion (e.g., measuring walking is much more complex than odometry based on wheel revolutions). Also, Zee does not depend on additional sensors since these are either not present in typical consumer devices (e.g., laser ranging) or even if present are not amenable to use in a crowdsourcing setting (e.g., users walking through a mall would not normally be taking pictures with their smartphone camera). On the other hand, unlike SLAM, Zee assumes the availability of a map and leverages the constraints imposed by the map for the purposes of localization.

2.6 Inertial Sensing

Finally, we discuss the problem of inertial sensing of human motion for the purposes of localization. This approach is interesting because consumer mobile devices such as smartphones are increasingly being equipped with sensors such as magnetometer (or compass), accelerometer, gyroscope, and barometer. These sensors, respectively, enable measurement of direction, acceleration, rotational velocity, and altitude. Knowing the starting location, a device can, in principle, be tracked using dead-reckoning, wherein the inertial sensor measurements are integrated over time [6]. However, a significant challenge is that even small errors in inertial sensing could be magnified by integration. Zee addresses this problem by leveraging the constraints imposed by the map to filter out erroneous measurements (e.g., a measurement that has a particle passing through an obstruction such as a wall would be filtered out). A complementary approach to prevent the accumulation of errors is proposed in UnLoc [25], where virtual landmarks are created using existing sensing modalities such as WiFi.

Another significant challenge is the complexity of human locomotion. People tend to have different stride lengths. Although there has been work on estimating stride length based on careful modeling of a step using accelerometer data [14], such estimation tends to be sensitive to the placement of the sensor, as does other estimation such as step counting. Indeed, the accelerometer signal tends to be strongest and most distinctive when the sensor is foot-mounted [21], but this does not accord with the typical placement of a device such as a smartphone. Also, in general, the device could be in an arbitrary orientation relative to the user's body and their direction of motion, which also makes estimation challenging.

Some of the above challenges have been addressed in [29, 30]. The use of map constraints for particle filtering, WiFi-based initialization of particles, and augmenting the particles with orientation information are all aspects inherited by Zee. In addition, this prior work also considers transition between floors, which Zee does not in its present incarnation. However, the prior work assumes a foot-mounted IMU that reports step events, the stride length, and the change in heading. These assumptions are problematic in a smartphone-based sensing context, for the reasons noted above.

To avoid assumptions regarding the placement of the IMU or processed step information being returned by it, we employ two techniques in Zee. First, rather than working with sensing thresholds or careful modeling that tend to be placement-dependent, we leverage the fundamental, placement-independent property of walking — its repetitiveness and hence periodicity — to estimate unknowns such as the step count. Second, we augment particle filtering to also estimate unknowns such as the stride length based, in part, on the constraints imposed by the map. In addition, Zee employs a novel backward belief propagation technique to trace back in time and infer location information post facto, thereby enabling more effective WiFi crowdsourcing.

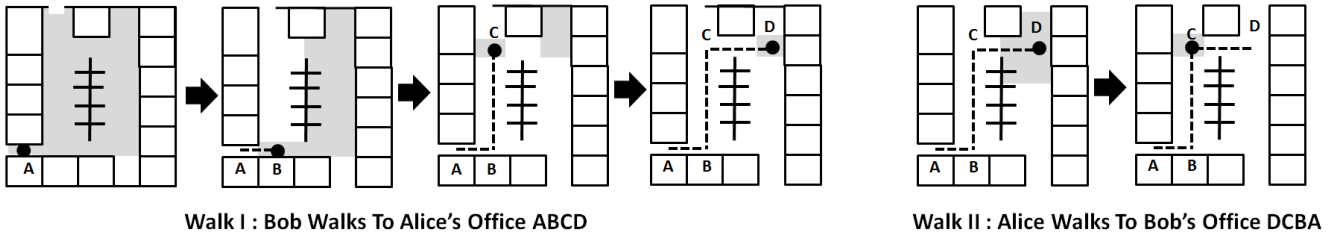

Figure 1: Example Scenario

3. ZEE EXAMPLE SCENARIO

All WiFi-based indoor localization schemes require a training data set — a set of tuples (WiFi measurement,location) of WiFi RSS measurements annotated with the indoor locations where the measurements were made. *The goal of Zee is to enable smartphone-based crowdsourcing of this training data set without requiring any active user participation.* In fact, Zee can run as a background process on smart phones without affecting the user in any way. In this section, we walk through an example scenario to provide an overview of and intuition for how Zee achieves zero-effort crowdsourcing.

Inferring a user's location. Bob is sitting in his office at work at location A (Figure 1). He downloads the Zee client and continues with his work as usual. At some point he decides to walk to Alice's office at location D. Initially, Zee does not know where Bob is located and hence Zee initializes Bob's locations as a probability distribution uniformly across the entire floor, as depicted by the gray region corresponding to Walk I in Figure 1). To walk to Alice's office, Bob takes the path ABCD indicated in Figure 1.

As Bob traverses this path, Zee uses the accelerometer, compass and the gyroscope on Bob's smartphone to continuously infer the direction and distance walked by him. Then using the floorplan of the indoor space, Zee updates the probability distribution of Bob's location by eliminating possibilities that would require him to violate the physical constraints imposed by the floorplan, such as walking through walls. Thus, as Bob reaches point B and then goes on to point C (as depicted in Figure 1), the spread of possible locations that Bob can be at, shrinks. Eventually, when Bob reaches D, Zee is able to narrow down the possible locations of Bob to his correct location. The key reason for this convergence is that there is only one possible path in the shape of ABCD that can be accommodate within the indoor space. *Thus, even without knowing Bob's initial location, but by simply tracking his movements, Zee is able to eliminate all alternative possibilities and eventually determine his location.*

Backward belief propagation. At this point, having narrowed down Bob's location using the sequence of his movements, Zee traces back the entire path taken by Bob and infers post facto that he must have taken the path ABCD. *Thus, Zee can also trace the entire history of locations at which Bob was present.*

Recording WiFi measurements. As Bob was traversing the path from A to D, Zee was also periodically scanning for proximate WiFi Access Points (APs) and recording the Received Signal Strength (RSS) from these APs. Knowing the entire path ABCD taken by Bob, Zee can associate with each RSS measurement the corresponding location on the path ABCD where the measurement must have been made. Thus, Zee obtains its first set of tuples $< RSS\ Measurements, Locations >$. Also, from this point on, having located Bob, Zee can track Bob's future movements and hence his locations. As more WiFi measurements are made, Zee on Bob's phone obtains more location-annotated WiFi measurements over the rest of the day. *Zee is thus is able to obtain location-annotated WiFi measurements from Bob's walks, without having any a priori knowledge of his initial location.*

Using past WiFi measurements to locate subsequent users. At this point Alice learns about Zee from Bob and installs it in her phone. She then decides to walk to Bob's office from her office along the path DCBA. This time, however, the Zee server has obtained some WiFi measurements from Bob's walk. Thus, Zee first performs a WiFi scan and obtains RSS measurements from proximate APs. Rather than initializing Alice's locations uniformly across the entire floor (as in Walk I in Figure 1), Zee uses these RSS measurements and the database to obtain a confined probability distribution, as depicted by the gray region corresponding to Walk II in Figure 1. In other words, the WiFi database obtained from Bob's walk helps Zee to narrow down the possibilities for Alice's initial location. Now as Alice walks, her location estimate converges much more quickly to her true location (by the time she reaches C in Figure 1) than it did in Bob's case. *Thus, each new walk in Zee benefits from the WiFi measurements accumulated from prior walks and in turn benefits the localization of future walks. In fact, after enough walks, Zee will be able to accurately locate a new user simply from a WiFi scan, using WiFi-based localization.*

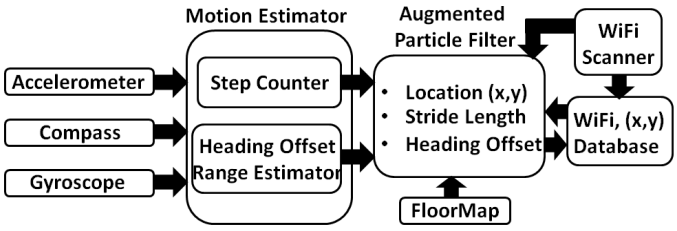

Figure 2: Zee Architecture

4. ZEE ARCHITECTURE

Figure 2 depicts a pictorial overview of Zee's architecture. There are two key components in Zee: *Placement Independent Motion Estimator* (PIME) and *Augmented Particle Filter* (APF). PIME uses mobile sensors such as the accelerometer, compass, and gyroscope to estimate the user's motion. The APF uses the motion estimates from PIME and the floormap as input to track the user's location on the floor.

Placement Independent Motion Estimator (PIME). PIME uses the accelerometer, compass and gyroscope data to perform three key functions. First, it reliably determines whether or not the person is walking. Second, when the user is walking, it generates an

event each time a step occurs and reports this event to the APF. Third, it provides APF with a rough estimate of the *Heading Offset (HO)*, i.e., the angle between the orientation of the phone and the user's direction of motion. The heading offset arises due to the combination of two reasons: (i) in general, the phone might not be oriented along the direction of the user's walk. For example, the user might be walking along the north-south direction while holding the phone laterally, so that it points east-west, which is a common scenario when phone users watch video while walking, and (ii) the presence of magnetic materials often affects the phone's compass. The APF then starts with the rough HO information and refines it to arrive at an accurate HO estimate.

A key feature of PIME is that it is independent of device placement, i.e., *whether the user is carrying the phone in his shirt pocket, trouser pocket, hand,* etc. This is a design requirement for Zee since it must be able to crowdsource in the background without any active user participation. To achieve this, Zee includes novel techniques for placement independent step detection (described in Section 5) and heading offset range estimation (described in Section 6).

Augmented Particle Filter. The key function of the APF is to track the probability distribution of a user's location as he/she walks on the floor. In order to convert steps into distance, the APF needs to estimate the stride length of the user, i.e., the distance traversed per step. Further, to track the user's location, estimating the direction of walking is crucial. While compass measurements provide the phone's orientation, the APF also estimates the HO to correctly compute the user's direction of motion. To simultaneously estimate location, stride length, and heading offset, the APF takes a novel approach – it maintains a four-dimensional joint probability distribution function in the form of a particle filter, comprising 2D location, stride length and, HO, and learns all these values as the user walks on the floor. The APF also implements backward-belief propagation, as described in Section 3.

Creating the WiFi Database. The APF runs belief back-propagation to correct the user's path history. This yields a time-indexed sequence of the user's estimated location during their walk. These location estimates are used to annotate the time-indexed WiFi information that Zee obtains through periodic scans. Thus, Zee's WiFi database comprises WiFi measurement annotated with location information.

WiFi-based initialization in APF. After the WiFi database has been initialized using the data from the first user, for subsequent users, APF uses information from WiFi scans and the current WiFi database, to obtain a confined initial location distribution, as described in Section 8. Confining the initial distribution instead of spreading it uniformly across the floor, enables a quicker convergence to the actual location, as noted in Section 3.

Refinement of the WiFi database. The training data obtained from each subsequent walk is in turn used to refine the existing WiFi database, thus making it more accurate for the next walk. In this manner, both APF-based user tracking and WiFi-based localization work in tandem, benefitting each other and progressively helping refine the WiFi database through crowdsourcing.

5. COUNTING STEPS

As described in Section 4, Zee uses step counting to estimate the distance traversed by the user. There are two tasks for any step counting algorithm: first, to reliably ascertain that the user is indeed walking, and second, to count the number of steps. Zee's step counting algorithm is designed to function irrespective of device placement *i.e.,* how the device is carried by the user.

Typical mobile phone placement scenarios. In order to find out how people typically carry their phones we interviewed 30 employees in an office. Based on our interviews we found that while most men carry their phones in front pant pockets, some carry in their shirt pockets or rear pant pockets and pouches attached to their belt. Women most often carry them in their hands or in handbags and sometimes in pant pockets. Further, both men and women may hold the phone in their hand while using it or to their ear while talking. In designing and evaluating our schemes we used these inputs as a guide.

Data collection. In order to test our schemes, six different people (four men and two women) were given smartphones to collect accelerometer data. Men collected data for five different placements: shirt pocket, pant front pocket, pant rear pocket, in hand while not using the phone, and in hand while using the phone. Women collected data for three different placements: handbag, in hand while using the phone and in hand while not using the phone. For each of these scenarios, data was collected when the user was not walking as well as when the user was walking.

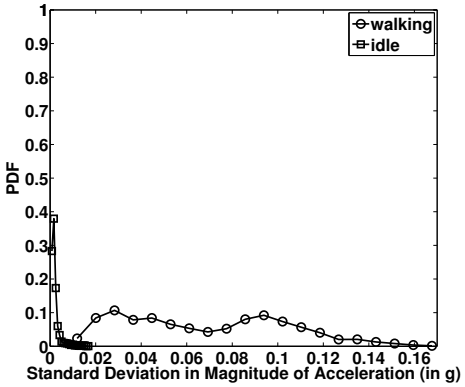

Figure 3: Distribution of standard deviation of acceleration magnitudes during idle and walking states.

Idle versus Motion. When the user is idle, it is expected that their phone will register little acceleration. Thus, the standard deviation in the magnitude of acceleration would be a good indicator of whether there is any movement. Figure 3 depicts the probability density function (PDF) of the standard deviation in the magnitude of acceleration experienced by the phone over a 1s period, in both idle and walking scenarios. The PDF was constructed from a total of 12500 data points for idle and 17000 points for walking. As seen from Figure 3, while the standard deviation is under 0.01g with 99% probability when the user is idle, it is over 0.01g with almost 100% probability when the user is walking. Using the standard deviation by itself, however, is not sufficient to ascertain that the user is walking. For example, sudden movements made by users while they are idle (*e.g.,*hand gestures, turning or shifting in the chair *etc.*) could also result in a large acceleration. To distinguish walking from such sudden movements, we use a novel scheme that exploits a very fundamental property of walking, namely its repetitive nature.

Repetitive nature of walks. Figure 4 depicts the acceleration values seen along the three axes by a mobile phone carried by two different users, one carrying it in his shirt pocket and the other in his front pant pocket. As seen from Figure 4, the acceleration along each axis and in each case exhibits a very repetitive pattern. The repetitive pattern arises because of the rhythmic nature of walking, with a sequence of two steps (one left and one right) constituting the period. While these patterns may be very different across users and placements, *the acceleration pattern for a given user with a*

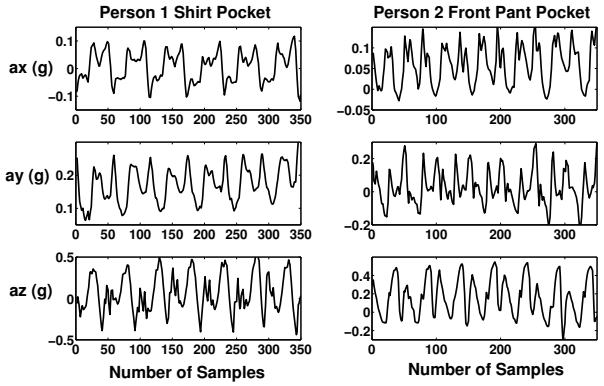

Figure 4: Walking is repetitive

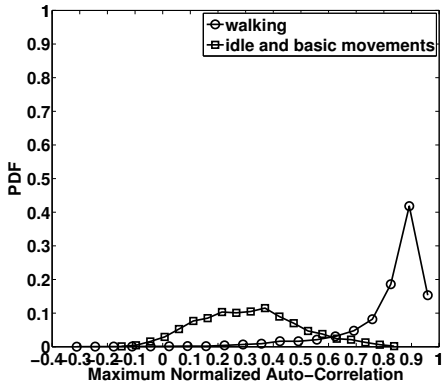

Figure 5: Distribution of Maximum Normalized Auto-Correlation during idle and walking states.

particular device placement repeats. Zee uses this observation to not only count steps but also to ascertain that the user is indeed walking.

Normalized Auto-correlation based Step Counting (NASC). The intuition behind NASC is that if the user is walking, then the auto-correlation will spike at the correct periodicity of the walker. Thus, given an acceleration signal $a(n)$, Zee computes the normalized auto-correlation for lag τ at the m^{th} sample as,

$$\chi(m,\tau) = \frac{\sum_{k=0}^{k=\tau-1} \left[\begin{array}{l} (a(m+k) - \mu(m,\tau)) \\ (a(m+k+\tau) - \mu(m+\tau,\tau)) \end{array} \right]}{\tau\sigma(m,\tau)\sigma(m+\tau,\tau)} \quad (1)$$

In Eqn 1, $\mu(k,\tau)$ and $\sigma(k,\tau)$ are the mean and standard deviation of the sequence of samples $< a(k), a(k+1), \cdots a(k+\tau-1) >$.

When the person is walking and τ is exactly equal to the period of the acceleration pattern, the normalized auto-correlation will be close to one. Since, the value of τ is not known a priori, NASC tries values of τ between τ_{min} and τ_{max} to find the value of τ for which $\chi(m,\tau)$ becomes maximum. Thus,

$$\psi(m) = max_{\tau=\tau_{min}}^{\tau=\tau_{max}} \left(\chi(m,\tau)\right). \quad (2)$$

$\psi(m)$, the maximum normalized auto-correlation, simultaneously provides two pieces of information. A high value (close to 1) suggests that the person is walking and the corresponding value of $\tau = \tau_{opt}$ gives the periodicity of the person's walk.

Since the sampling frequency of our accelerometer was 50Hz, two step duration of most people lies between 40 and 100 samples. Consequently, in our implementation, the initial search window (τ_{min}, τ_{max}) is set to $(40, 100)$. However, once the periodicity of the person's walk is found to be τ_{opt}, the search window is reduced to a few samples around τ_{opt}. In our implementation, after finding the user's periodicity, we used $\tau_{min} = \tau_{opt} - 10$ and $\tau_{max} = \tau_{opt} + 10$. NASC continuously updates the value of τ_{opt} to account for small changes in the user's walking pace.

Figure 5 depicts the distribution of $\psi(m)$ for idle and walking states. The idle state included movements such as hand gestures, transition from sitting to standing and vice versa, and spinning in a chair. As seen from Figure 5, when ψ is 0.7 or higher, the probability that the person is idle is extremely low (less than 1%). Note that NASC will also detect other repetitive activity such as running, but in this paper we did not evaluate such activities.

Walk versus Idle decision in Zee To decide whether or not the user is walking, we use a combination of both standard deviation in the magnitude of acceleration $\sigma_{||a||}$ and the maximum normalized auto-correlation ψ. Zee transitions between the IDLE and WALKING states as follows:

- If $\sigma_{||a||} < 0.01$ then $state = IDLE$.
- Else If $\psi > 0.7$ then $state = WALKING$.
- Else no change in current value of $state$.

Counting steps. Zee uses the periodicity estimated by NASC for step counting. Having estimated τ_{opt}, NASC generates a step occurred event every $\frac{\tau_{opt}}{2}$ samples while the person in the WALKING state.

	Hand While Using	Pant Front Pocket	Pant Back Pocket	Hand Not Using	Shirt Pocket	Hand bag	Over (all)
False +ive	0%	0%	0%	0%	0%	0%	0%
False -ive	2%	0%	0%	0%	0%	0%	0.6%
True +ive	100%	100%	100%	100%	100%	100%	100%
True -ive	98%	100%	100%	100%	100%	100%	99.4%

Table 1: Performance of step counting in Zee

Evaluation of step counting. Table 1 presents the findings from our evaluation of step counting in Zee. A false positive means that an extra step was counted while the user was in idle state, while a false negative means that a step was missed while the user was walking. Table 1 shows that these error rates are very low, often zero, for various placements of the phone across users.

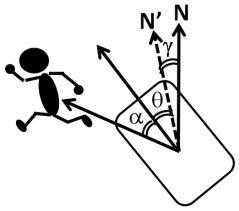

Figure 6: Heading Offset

6. ESTIMATING HEADING OFFSET RANGE

To track the user's path, Zee must estimate the user's direction of walking. The mobile phone's compass provides orientation of

the phone relative to the perceived magnetic north (i.e., the angle θ in Figure 6). In general, however, the compass reading might not be aligned with the direction of motion of the user. We refer to this difference between the compass reading and the direction of motion of the user as the *heading offset (HO)*. HO arises due to a combination of two factors: *magnetic offset* and *placement offset*.

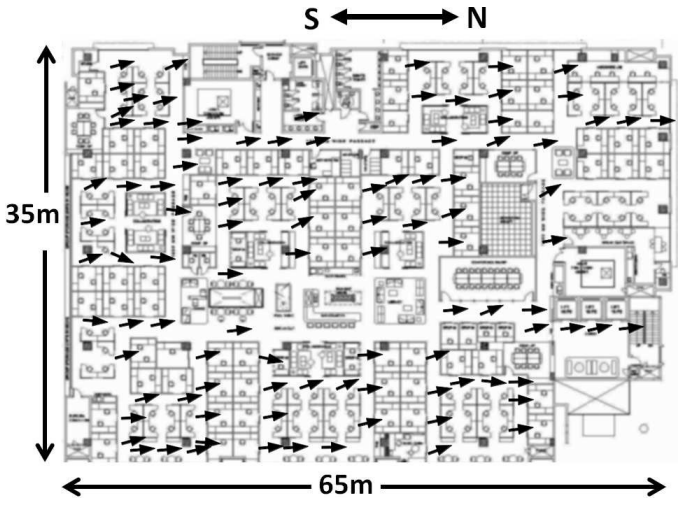

Figure 7: Direction of North as shown by the compass across the floor of a large office

Magnetic Offset: The presence of magnetic materials (e.g., metal) in close proximity of the mobile phone can disturb its perception of North, leading to an offset error in the compass measurement. We refer to this difference between the true north and the north perceived by the compass as the *magnetic offset*. As illustrated in Figure 6, the phone perceives north to be towards N' while the true north is along N. The magnetic offset is depicted as the angle γ in Figure 6. We have found that the magnetic offset is usually a characteristic of a given location, depending on the construction and other materials in the vicinity, and typically remains stable with time. Figure 7 depicts the North direction as measured by a phone's compass at various locations on the floor of a large office building, and Figure 8 shows the distribution of magnetic offset measured at 100 different locations in the floor. The magnetic offset is within $\pm 15^\circ$ in 90% of the locations, and occasionally as high as 30°.

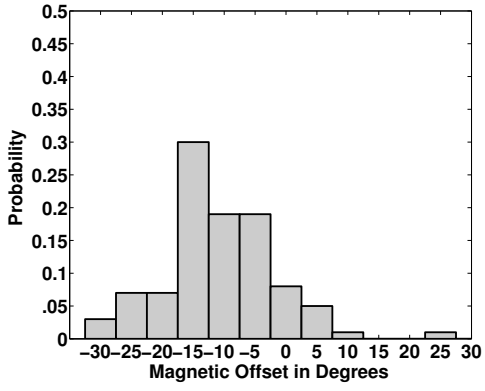

Figure 8: Distribution of magnetic offsets

Placement offset: The phone's compass measures the angle of orientation of the phone with respect to the perceived north. The phone, however, might not be oriented along the direction of motion of the user. For example, when the user walks while watching a video or a photo with the phone held laterally, the orientation of the phone would be at an angle of 90° relative to the direction of motion of the user. We refer to this difference between the phone's orientation and the direction of motion of the user as the *placement offset*. In the example of Figure 6, the placement offset is depicted by the angle α. We found that for most placements such as in the shirt or pant pockets, the placement offset is typically $\pm 45^o$. However, when the phone is placed in a pouch or a handbag, the placement offset can be arbitrary. Nevertheless, the placement offset typically remains unchanged even when the user takes a turn and changes the direction of walking.

Heading Offset: As depicted in the Figure 6, the direction of motion of the user is $\alpha + \theta + \gamma$ relative to the true magnetic north. In other words, the heading offset in this scenario is $\alpha + \gamma$ — the sum of the magnetic offset and the placement offset. While the placement offset is typically constant during a single walk (unless the user changes how he/she is carrying the phone in the middle of the walk), the magnetic offset can change as the user moves through different locations. Consequently, Zee must accommodate both these components of the HO. To estimate HO, Zee takes a two-step approach. It first estimates the HO in a broad range (sector) based on the acceleration experienced by the phone. Then, the APF uses this range of values as the prior for the HO distribution and proceeds to refine its estimate of the HO as the user walks. In the remainder of this section, we describe our technique for estimating the HO range; we defer discussion of the APF to Section 7.

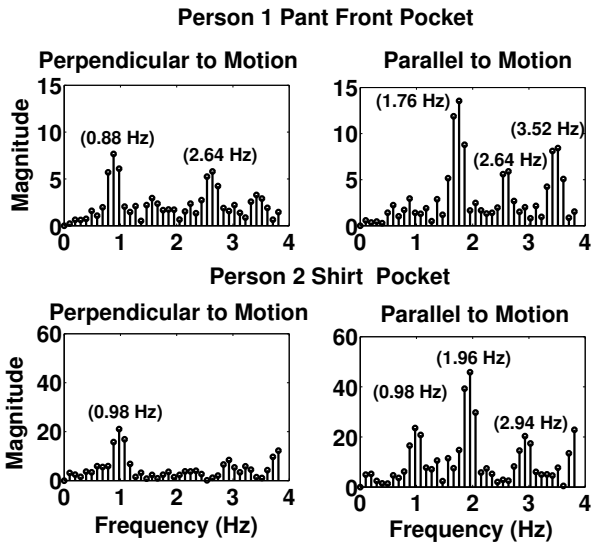

Figure 9: Spectrum of walking

The spectrum of a typical walk. Figure 9 depicts the magnitude of the Fourier transform of the accelerometer signal registered by the phone along directions parallel and perpendicular to the direction of walking, for two different people. Person 1 carries the phone in his shirt pocket while person 2 carries the phone in his front pant pocket. As seen in Figure 9, the dominant frequencies are multiples (harmonics) of a fundamental frequency. For example, in the case of person 1, the fundamental frequency is 0.88 Hz while for the second person it is 0.98 Hz. This fundamental frequency

corresponds to the periodicity of two steps, i.e., the combination of a left step and a right step.

After observing the Fourier transform across several users and placements, we discovered an interesting fact: *the second harmonic (two times the fundamental frequency, corresponding to the periodicity of single steps) is either completely absent or is extremely weak in the accelerations experienced by the phone in the direction perpendicular to the user's walk. It is however always present and dominant in the direction parallel to the user's walk.* This rather curious phenomenon arises for the following reason. In the direction parallel to the walk, each step (left or right) registers as a strong, repetitive acceleration signal. However, this is not so in the perpendicular direction, because the user's sideways or lateral movement (e.g., hip sway) only repeats every two steps. Indeed, as can be seen in Figure 9, the second harmonic frequencies corresponding to 1.76 Hz (person 1) and 1.96 Hz (person 2), while being present in the direction parallel to the user's walk, are absent in the perpendicular direction.

Heading Offset (HO) range estimation. Suppose the magnitude of the second harmonic in the Fourier transform along north is F_y and that along west is F_x. Almost the entire contribution to this harmonic is from the direction of walking, so F_x and F_y must be its components. Thus, $\alpha + \theta + \gamma = \arctan \frac{F_x}{F_y}$ (Figure 6). However, since we work with the (unsigned) magnitude of the Fourier transform, we cannot tell whether the person is walking forward or backward along this angle. Consequently, it is equally likely that $\alpha + \theta + \gamma = \arctan \frac{F_x}{F_y} + 180^o$. Knowing θ from the compass then, the two possible values of $\alpha + \gamma$ (HO) can be estimated.

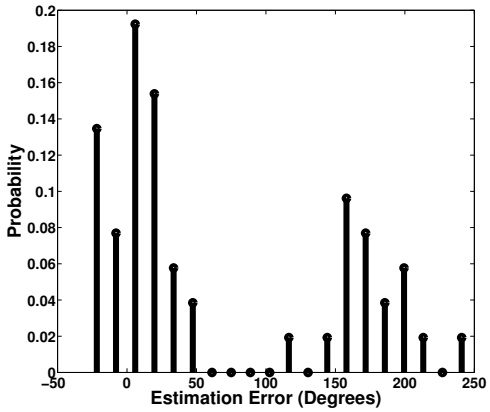

Figure 10: Error distribution of HO estimation

We characterize the error in our HO estimation scheme by testing it on the same walk data collected across six different people, as in Section 5. Figure 10 depicts the probability distribution of the error in the HO estimation. The distribution is bimodal around 0° and 180°, because, as noted above, the HO estimation scheme cannot distinguish between the forward and backward directions. Further, the error spans about 60° around each of 0° and 180°. Thus, the error in our HO estimation is large and by itself cannot be used to track the user. Nevertheless, it helps narrow down the possibilities to two diametrically opposite 90° sectors (expanded from 60° to accommodated the possibility of larger errors) centered about $\arctan \frac{F_x}{F_y}$. The APF then uses this estimate as the prior to efficiently converge on the correct HO, as discussed next.

7. TRACKING USING AUGMENTED PARTICLE FILTER (APF)

In this section we describe how Zee uses APF to track the users' paths as they walk around in an indoor environment.

The key idea. *The key idea of Zee is that as a user continues to walk in an indoor environment, navigating through hallways and turning around corners, the possibilities for the user's path and location shrink progressively.* For example, if a person walks 10m north followed by 12m east and then 20m south in an indoor space with hallways and walls, there will likely be only a few paths in the indoor environment that could accommodate such a walk without having the user run into a wall or other barrier. The longer the walk, the more concentrated the possibilities. As a user walks in the indoor environment while going about their routine, Zee continually eliminates possibilities that violate wall constraints, until eventually only one possibility remains. To drive towards such convergence, Zee maintains a probability distribution of possible locations of the user and updates it for every step taken by the user.

Zee uses the stride length to convert each step into the corresponding distance traversed. However, people generally have different stride lengths, because of differences in height and walking style, so Zee must estimate the stride length of each user, while simultaneously attempting to locate them. Further, as described in Section 6, Zee must also estimate the HO accurately. To accomplish these tasks, Zee augments the standard particle filter to include not only of the location of the user but also the stride length and the HO as two additional unknowns. Thus, Zee continuously maintains and updates a four-dimensional joint probability distribution over the 2D location, stride length, and HO of the user.

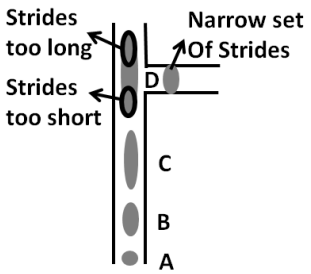

Figure 13: Stride estimation in Zee

Example of stride length estimation. For intuition on how stride length estimation works, consider a simple example, where the user's initial location is assumed to be known but not their stride length. Initially, Zee considers the full range of humanly feasible stride lengths (0.5m to 1.2m, in our current implementation). The user's initial location is shown in Figure 13 as the particle concentration (gray region) labeled A. As the user takes steps and moves to B and then onto C, the gray region becomes elongated. The reason is that since Zee consider a range of possible stride lengths, particles corresponding to a longer stride length travel farther than those with a shorter stride length. At point D, when the user takes a right turn into a passage way, particles with stride lengths that are either too large or too short attempt to move to the right but are hindered by the walls on either side of the passage way. Consequently, as the user walks into the passage way, the possibilities for the user's stride length narrow down. The more the user walks and navigates around corners, the more accurate the estimate becomes. While in this simple example we assumed that the initial location is known, even if the initial location were unknown, Zee can estimate location and stride length simultaneously as the user continues to walk.

In practice, we found that within a single walk, users exhibit up

Figure 11: An example run of Zee

Figure 12: Backward belief propagation in Zee

to $\pm 10\%$ variation in their stride length. To account for this variation, we add a random error δ uniformly distributed in the range of $\pm$ 10% of the stride length (used in Equation 3 below).

Working with heading offset. Zee estimates the HO in a similar manner as it estimates the stride length. Initially, the heading offset is uniformly distributed within the two 90° sectors suggested by the HO range estimation scheme in Section 6. As the user walks, incorrect HO values are eliminated due to the wall constraints. While this approach accounts for the errors in the constant component of HO (due to phone placement), the APF must also account for the changes in HO due to different magnetic offsets at different locations. In order to accommodate this variation, the APF models this component as a compass measurement error — a Gaussian random variable β that is added to the compass measurement (used in Equation 3 below). We found that using a zero mean Gaussian with a standard deviation of 5° typically was sufficient in our floor.

The particle filter. The APF maintains a four-dimensional joint probability distribution as a particle filter, with a set of particles (samples), $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2 \cdots, \mathbf{X}_N)$ representing the probability distribution. Here $\mathbf{X}_i = (x_i, y_i, s_i, \alpha_i)$, with (x_i, y_i) being the 2D location, s_i the stride length, and α_i the placement offset. After the user takes the k^{th} step, the i^{th} particle is updated as,

$$x_i^k = x_i^{k-1} + (s_i + \delta_i)\cos(\alpha_i + \theta + \beta_i) \quad (3)$$

$$y_i^k = y_i^{k-1} + (s_i + \delta_i)\sin(\alpha_i + \theta + \beta_i) \quad (4)$$

As noted earlier, s_i is perturbed by δ_i to account for variation in the stride length, and θ (the compass reading) is perturbed by β_i to account for compass measurement error and variation in the magnetic offset.

After each update, particles are tested to see if they violate any wall constraints. If the line joining (x_i^{k-1}, y_i^{k-1}) and (x_i^k, y_i^k) intersects a wall then the particle is eliminated. In order to replace each eliminated particle, a new particle is randomly chosen from the particle set at the $k-1^{th}$ step and updated. Note that s_i and α_i are not updated, rather incorrect values only get eliminated.

In our implementation we found that within one step, often it is possible to sample the compass several times. Accordingly, we use an average value of all the samples to arrive at the θ for the step. Sometimes, however, when the user turns, θ might change by a large amount within one step. Consequently, if the compass reading changes by more than 20° within a single step, we perform incremental updates by subdividing the step into fractional sub-steps.

An example walk in an office building. Figure 11 shows how Zee works in a large office floor, measuring 65m $\times$ 35m. The user was asked to walk from O to D, along the path shown in Figure 11. There are three turns in the path, at points A, B and C. Initially, particles were instantiated at all possible locations with all possible stride lengths and heading offsets spread uniformly over the two 90° sectors suggested by the HO range estimator. As the user walks to point A and then turns, the spread of particles dramatically decreases, as several possibilities are eliminated. Further, as the user walks towards point B, even more possibilities are eliminated. Finally, as the user navigates around the turn at B, Zee has narrowed down to the user's correct location.

Backward belief propagation. After the turn taken at B, the position of the user is localized to a single concentration of particles. Tracing these particles back in time would, therefore, allow us to infer the past locations of the user accurately. To enable such a trace-back, each particle $\mathbf{X}_i^k$ after the k^{th} step maintains a link to its parent particle, *i.e.,* the particle $\mathbf{X}_j^{k-1}$ that it was generated from. In the backward belief propagation step, out of the set of particles $\mathbf{X}^{k-1} = \{\mathbf{X}_1^{k-1}, \mathbf{X}_2^{k-1}, \cdots, \mathbf{X}_N^{k-1}\}$, the particles with no children surviving in the succeeding step (i.e., in the set $\mathbf{X}^k$), are eliminated. This helps ensure that possibilities that are filtered out in

future steps are not retained in the past steps. Figure 12 depicts the results of applying backward belief propagation on the user's walk along the reverse direction, starting from after the turn at B and going back to the initial location O. As the figure shows, using backward belief propagation, Zee is able to accurately trace the user's path back to the starting location O.

8. PUTTING IT ALL TOGETHER: ZERO-EFFORT CROWDSOURCING

Zee periodically scans for beacons from proximate WiFi Access Points (AP) and records the Received Signal Strength (RSS) measured, along with a timestamp. Also, as Zee tracks the path taken by users, whether in the forward direction or through backward belief propagation, it annotates the path with timestamps, indicating the times at which the user was located at particular points in the path. Thus, Zee can determine where in the floor a certain WiFi measurement was taken and thereby generate location-annotated WiFi measurements of the form (*location, WiFi RSS*). This database of measurements can then be used to locate new users using existing WiFi localization techniques. We now describe the two WiFi localization schemes used in our evaluation.

HORUS. HORUS uses a set of location-annotated RSS measurements to construct a probability distribution, $P(rss_{AP_k} = r|\mathbf{x} = \mathbf{x}_i)$, i.e., the probability of measuring an RSS value of r from access point AP_k at location $\mathbf{x}_i$. $\mathbf{x}$ here is a 2-dimensional location. In order to locate a device using HORUS, a device measures a vector of RSS measurements, $\mathbf{R} = < r_1, r_2, \cdots, r_m >$, where r_i is the RSS from AP_i. The probability of observing $\mathbf{R}$ at a location $\mathbf{x}_i$ is then computed as,

$$P(\mathbf{R}|\mathbf{x} = \mathbf{x}_i) = \prod_k P(rss_{AP_k} = r_k|\mathbf{x} = \mathbf{x}_i) \quad (5)$$

Using Bayesian inference, HORUS computes $P(\mathbf{x} = \mathbf{x}_i|\mathbf{R})$. The location of the device is then estimated as either the *maximum likelihood location* (i.e., the location with the highest probability) or the expectation over all locations (*expected location*).

EZ. EZ relies on a widely-used RF propagation model for WiFi received signal strength (RSS) in indoor environments — the Log Distance Path Loss (LDPL) model. The LDPL model estimates RSS $rss^k_{\mathbf{x}}$ (in dBm) measured at a location $\mathbf{x}$ of the signal from AP_k placed at a location $\mathbf{c}_k$ as,

$$rss^k_{\mathbf{x}} = rss^k_0 - 10\gamma_k \log\left(d\left(\mathbf{x}, \mathbf{c}_k\right)\right) + N(\mathbf{x}). \quad (6)$$

In Eqn 6, rss^k_0 is the RSS from AP_k at the reference distance of 1m, γ_k is the path loss exponent, and $d(\mathbf{x}, \mathbf{c}_k)$ is the distance between locations $\mathbf{x}$ and $\mathbf{c}_k$. $N(\mathbf{x})$ captures the random fluctuations in RSS due to multi-path effects. EZ uses WiFi measurements (a few annotated with location information but most not) from within the indoor space data to construct the LDPL model for each WiFi AP. Then, to locate a device, it converts the RSS measurement obtained from an AP to the estimated distance from that AP using,

$$d(\mathbf{x}, \mathbf{c}_k) = 10^{\left(\frac{rss^k_0 - r_k}{10\gamma}\right)}. \quad (7)$$

Standard trilateration is then used to locate the device, once its distance from three or more APs has been estimated.

Using existing measurement database for subsequent crowdsourcing. In the absence of prior information, Zee starts by spreading the probability distribution of a user's location across the floor. However, once "enough" location-annotated WiFi measurements have been gathered, Zee can use existing localization schemes such as HORUS or EZ to initialize the probability distribution in a more localized area rather than over the entire floor. When using HORUS, Zee draws random samples from $P(\mathbf{x} = \mathbf{x}_i|\mathbf{R})$ and initializes the locations of the particles accordingly. When using EZ, Zee perturbs the measured RSS values to simulate the effect of multipath (using a Gaussian distribution with mean 0dB and standard deviation 5dB) and then generates sample locations for the particle filter. Using such a localized distribution rather than one spread across the entire floor helps speed up the convergence of the particle filter, thereby aiding subsequent crowdsourcing.

Dealing With Paths With Unconverged Particles. Not all paths may lead to a unique location in Zee, e.g., short paths that do not include enough turns to uniquely establish the user's trajectory. In such cases, the spread of particles will typically be large instead of being confined to a small region. Zee identifies such unconverged paths using measures such as the variance in the locations of particles and the number well-separated clusters, and discards the WiFi measurements corresponding to such walks.

9. EVALUATION

We evaluate the performance of Zee in a large office building, with a 65m $\times$ 35m floor plan, as depicted in Figure 14. We seek to answer several questions, including: (a) how well is Zee able to track users, (b) if Zee had been provided the user's initial location or had accurate knowledge of the heading offset, would its performance have improved significantly, (c) how well does Zee estimate quantities such as the stride length and the heading offset of the user, and (d) how well do existing localization schemes perform when using the location-annotated data generated by Zee versus using manually collected data.

Experimental methodology. In a real-world setting, we expect several users to be running the Zee client on their phone as they walk through various sections of an indoor space. Further, users are unlikely to be walking continuously; they would typically walk between locations of interest and dwell at certain locations (e.g., a store) for a significant length of time. However, for our evaluation, we did not have the luxury of several users. Instead, we handed a phone running the Zee client to a user, who kept it with himself for about 15 hours continuously. Our experiment included elements aimed at emulating the real-world characteristics noted above.

To recreate walks with breaks at various locations, the user was asked to walk to various parts of the office floor whenever he had the time and leave the phone there for periods ranging from 20 minutes to a half-hour. Zee was left running on the phone continuously. Upon detecting no walking activity for more than 10 seconds, the accelerometer data was not processed until walking was detected again. WiFi data, however, is more valuable when collected from a single location over an extended period, since it helps capture the variability in the WiFi signal at that location. Consequently, WiFi RSS measurements were collected even during the time when the device was stationary. To record the ground truth for location, the user made note of the locations on a map where he had stopped and the corresponding time periods.

In order to evaluate the benefit of using data from previous users for subsequent data collection, the user in our experiment stopped Zee briefly, walked over to a completely different location on the floor, and then restarted Zee. In this manner four different "walks" were created by restarting Zee four times. Each time, Zee would initialize the user's location based on the WiFi data gathered from the previous walks.

Figure 14 shows the paths covered by the user during the 15-hour period (shown as the dark lines in the figure). Several of the locations were visited multiple times over the course of the experiment.

Zee's tracking performance. We first show the performance of

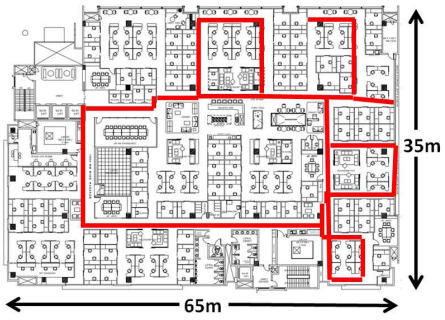

Figure 14: Total area covered during the 15 hrs by the user

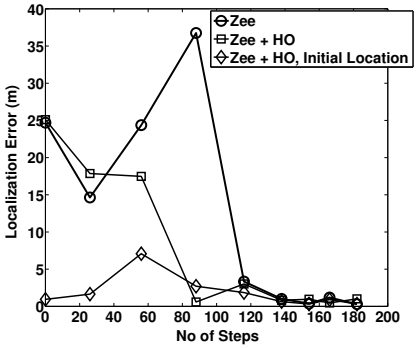

Figure 15: Location errors seen by Zee during walk 1.

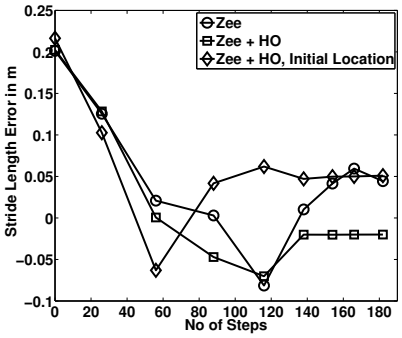

Figure 16: Stride length errors seen by Zee during walk 1.

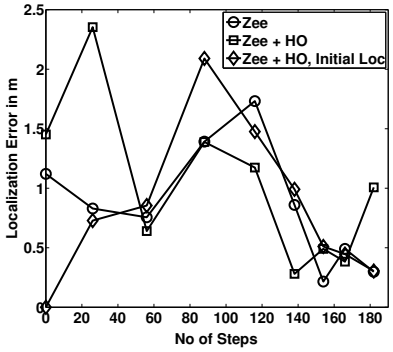

Figure 17: Location errors seen after backward belief propagation by Zee during walk 1

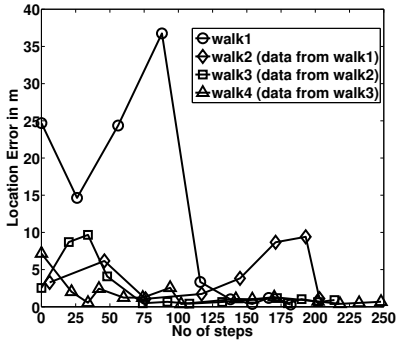

Figure 18: Using WiFi measurements from previous walks to start subsequent walks

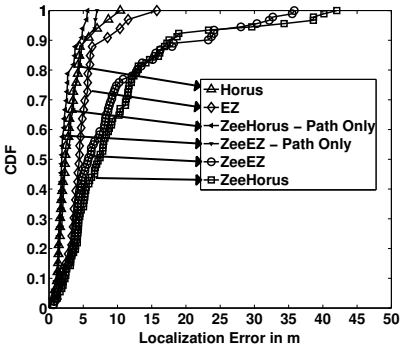

Figure 19: Performance of WiFi localization on Zee data

Zee on the first walk when there was no prior data from the floor. Note that this was not a continuous walk but instead was punctuated by several long stops, each lasting 20-30 minutes, as the user went to different locations on the floor. The walk lasted a total of 3 hours. Figure 15 depicts the location error seen by the particle filter at 9 different checkpoints (i.e., predetermined locations, where the ground truth was recorded) as a function of the step count during the walk. The location error was computed by estimating the user's location as the average location over all surviving particles, and then finding its distance from the ground truth location.

Figure 15 shows the location error with the particle filter run in the forward direction. As can be seen from the curve labeled Zee, initially the average error is extremely high. This is because the initial location is unknown and particles are spread all across the floor. Somewhere between steps 80 and 100, the user took a turn that eliminated all spurious possibilities and the error dropped sharply. At the end of the walk, the location error was under a meter; in other words, Zee's location estimate was correct to the last step!

Figure 17 depicts the location error after backward belief propagation was applied to the walk. As can be seen from the figure, the location error along the entire path is under 2m after this step.

Zee's performance with initial location and HO being known. In this experiment, we tested two different scenarios. In the first scenario, the initial location of the user was kept as unknown while heading offset was treated as known. In the second scenario, both heading offset and initial location were treated as known. Stride length, however, was assumed to be unknown in both cases.

As seen from Figure 15, with the HO known but initial location unknown, Zee converged must faster to the correct location compared to not knowing the HO, which is as expected. When both HO and initial location were known, initially the error increased because the stride length was unknown. However, as soon as Zee learned the stride length, the localization error reduced.

Figure 17 depicts the localization error for both these scenarios after backward belief propagation is applied. As seen from the figure, the location error along the entire path remains under 1m in both the scenarios. Further, the key point to note here is that *Zee was able to perform similarly well despite not knowing HO and initial location as when this information was known a priori, the only difference being that it took longer to converge in the former case, which is as expected.*

Zee's estimation of stride length. To evaluate how well Zee estimates stride length, we first measured the user's stride length by making him walk 20 steps in a straight line and directly measuring the distance covered. Then, during the actual experiment, we estimated the stride length at any step as the average stride length over the surviving particles. Figure 16 shows that the error in stride length estimation drops to around 5cm by the end of the walk.

Using WiFi measurements from previous walks for subsequent walks. For each successive walk, Zee was able to use the WiFi data gathered from the prior walks to estimate the distribution of the user's starting location. To compute this distribution, we used

EZ, as described in Section 8. To emulate a new user for each walk, the stride length and HO were deemed as being unknown at the beginning of each successive walk. Figure 18 shows that by using WiFi information from prior walks, Zee is able to converge much faster and have smaller location errors than otherwise.

Performance of WiFi localization using Zee-based crowdsourcing. To evaluate the performance of existing WiFi-based localization schemes when fed with Zee-based crowdsourced data, we first set up the baseline by collecting WiFi measurements manually at 117 locations, spaced about 3m apart and spread across the floor. At each location approximately 1000 WiFi beacons were collected for each proximate AP. This data was used to train HORUS and EZ. Additionally, we collected data from 91 separate test locations spread across the entire floor. Figure 19 depicts the distribution of localization error seen by EZ and HORUS at these test locations (labeled as EZ and HORUS in the figure).

Next, instead of the data gathered manually, we used the data obtained from Zee over the four walks to train EZ and HORUS, and tested these across all the 91 test locations. The corresponding distributions of localization error are also shown in Figure 19 (labeled as ZeeEZ and ZeeHORUS). We see that while ZeeEZ and ZeeHORUS have almost the same 50%ile errors as EZ and HORUS, respectively, the 80%ile errors are significantly larger. This was because, at the test locations that were far from all of the walk paths (which defined the spatial extent of crowdsourcing in this experiment), ZeeEZ and ZeeHORUS performed much worse than EZ and HORUS (which had the benefit of manually-gathered training data from across the floor).

To evaluate the performance of Zee more fairly, we tested Zee only on the test locations that were within 5m of any of the four walk paths. Figure 19 depicts the corresponding error distributions (labeled as ZeeEZ Path Only and ZeeHORUS Path Only). We see that the 50%ile and 80%ile errors are 1.2m and 2.3m, respectively, which are comparable to those seen when manually-gathered data is used for training (the curves labeled EZ and HORUS). This encouraging result suggests that Zee-based crowdsourcing could be effective and could enable localization with high accuracy, provided the space is well-covered by users during the course of their crowdsourcing walks.

10. CONCLUSION

In this paper, we have presented Zee, a system that enables crowdsourcing of location-annotated WiFi measurements in indoor spaces, using the mobile phones carried by users in normal course. A key attribute of crowdsourcing with Zee is that it is "zero-effort", not requiring any active user intervention in terms of location input, placement of the phone, or other aspect. Zee employs a set of novel techniques to resolve ambiguity in location during crowdsourcing, using inertial and WiFi measurements, and a map of the indoor space as the inputs. The data thus gathered can help train existing WiFi-based localization algorithms. Our evaluation on a large office floor shows that existing WiFi-based localization schemes, both fingerprinting-based and modeling-based, are able to perform accurate localization when trained with data that is crowdsourced automatically using Zee.

11. ACKNOWLEDGEMENTS

We thank Gursharan Sidhu for suggesting that WiFi data could be crowdsourced with the help of inertial tracking. We also thank our shepherd, Romit Roy Choudhury, and the anonymous reviewers for their constructive comments.

12. REFERENCES

[1] Time Domain. http://www.timedomain.com/.
[2] M. Azizyan, I. Constandache, and R. Roy Choudhury. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In *MobiCom*, 2009.
[3] P. Bahl and V. N. Padmanabhan. RADAR: An Inbuilding RF-based User Location and Tracking System. In *INFOCOM*, 2000.
[4] P. Bahl, V. N. Padmanabhan, and A. Balachandran. Enhancements to the RADAR User Location and Tracking System, Feb 2000. Microsoft Research Technical Report MSR-TR-2000-12.
[5] K. Bauer, D. Mccoy, B. Greenstein, D. Grunwald, and D. Sicker. Using Wireless Physical Layer Information to Construct Implicit Identifiers. In *HotPETs*, 2008.
[6] S. Beauregard and H. Haas. Pedestrian Dead Reckoning: A Basis for Personal Positioning. In *WPNC*, 2006.
[7] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless Device Identification with Radiometric Signatures. In *Mobicom*, 2008.
[8] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy Characterization for Metropolitan-scaleWi-Fi Localization. In *MobiSys*, 2005.
[9] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan. Indoor Localization Without the Pain. In *Mobicom*, 2010.
[10] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In *AAAI*, 1999.
[11] A. Goswami, L. E. Ortiz, and S. R. Das. WiGEM : A Learning-Based Approach for Indoor Localization. In *CoNEXT*, 2011.
[12] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical Robust Localization over Large-Scale 802.11 Wireless Networks. In *MobiCom*, 2004.
[13] N. B. John, J. Heidemann, and D. Estrin. GPS-Less Low Cost Outdoor Localization For Very Small Devices. *IEEE Personal Communications Magazine*, 7:28–34, 2000.
[14] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park. A step, stride and heading determination for the pedestrian navigation system. *Journal of Global Positioning Systems*, 3:273–276, 2004.
[15] P. Krishnan, A. S. Krishnakumar, W. H. Ju, C. Mallows, and S. Ganu. A System for LEASE: Location Estimation Assisted by Stationery Emitters for Indoor RF Wireless Networks. In *Infocom*, 2004.
[16] J. Leonard and H. F. Durrant-whyte. Simultanoues Map Building and Localization for an Autonomous Mobile Robot. In *IROS*, pages 1442–1447, 1991.
[17] H. Lim, L. Kung, J. Hou, and H. Luo. Zero-Configuration, Robust Indoor Localization: Theory and Experimentation. In *Infocom*, 2006.
[18] L. Ni, Y. Liu, C. Yiu, and A. Patil. LANDMARC: Indoor Location Sensing Using Active RFID. In *WINET*, 2004.
[19] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *MobiCom*, 2000.
[20] T. Rappaport. *Wireless Communications Principles and Practice*. Prentice Hall, 2001.
[21] P. Robertson, M. Angermann, and B. Krach. Simultaneous Localization and Mapping for Pedestrians using only Foot-Mounted Inertial Sensors. In *Ubicomp*, 2009.
[22] S. Sen, R. R. Choudhury, and S. Nelakuditi. SpinLoc: Spin once to know your location. In *ACM HotMobile*, 2012.
[23] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. Precise Indoor Localization using PHY Layer Information. In *ACM HotNets*, 2011.
[24] R. Smith and P. Cheeseman. On the Respresentation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*, 5, Winter 1986.
[25] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No Need to War-Drive: Unsupervised Indoor Localization. In *Mobisys*, 2012.
[26] R. Want and et al. The Active Badge Location System. *ACM Transactions on Information Systems*, Jan 1992.
[27] A. Ward, A. Jones, and A. Hopper. A New Location Technique for the Active Office. *IEEE Per. Comm.*, 4(5):42–47, 1997.
[28] A. Williams, D. Ganesan, and A. Hanson. Aging in Place: Fall Detection and Localization in a Distributed Smart Camera Network. In *ACM Multimedia*, 2007.
[29] O. Woodman and R. Harle. Pedestrian Localisation for Indoor Environments. In *Ubicomp*, 2008.
[30] O. Woodman and R. Harle. RF-Based Initialisation for Inertial Pedestrian Tracking. *Lecture Notes in Computer Science*, 5538:238–255, May 2009.
[31] J. Xiong and K. Jamieson. ArrayTrack: A Fine-Grained Indoor Location System. In *ACM HotMobile*, 2012.
[32] M. Youssef and A. Agrawala. The Horus WLAN Location Determination System. In *MobiSys*, 2005.