SAIL: Single Access Point-Based Indoor Localization

Alex Mariakakis
University of Washington

Souvik Sen
HP Labs

Jeongkeun Lee
HP Labs

Kyu-Han Kim
HP Labs

ABSTRACT

This paper presents *SAIL*, a Single Access Point Based Indoor Localization system. Although there have been advances in WiFi-based positioning techniques, we find that existing solutions either require a dense deployment of access points (APs), manual fingerprinting, energy hungry WiFi scanning, or sophisticated AP hardware. We design SAIL using a single commodity WiFi AP to avoid these restrictions. SAIL computes the distance between the client and an AP using the propagation delay of the signal traversing between the two, combines the distance with smartphone dead-reckoning techniques, and employs geometric methods to ultimately yield the client's location using a single AP. SAIL combines physical layer (PHY) information and human motion to compute the propagation delay of the direct path by itself, eliminating the adverse effect of multipath and yielding sub-meter distance estimation accuracy. Furthermore, SAIL systematically addresses some of the common challenges towards dead-reckoning using smartphone sensors and achieves $2 - 5$x accuracy improvements over existing techniques. We have implemented SAIL on commodity wireless APs and smartphones. Evaluation in a large-scale enterprise environment with 10 mobile users demonstrates that SAIL can capture the user's location with a mean error of 2.3m using just a single AP.

Categories and Subject Descriptors

H.3.4 [**Information Systems and Retrieval**]: Systems and Software; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

Keywords

Indoor location; Smartphones; Dead-Reckoning; Time-of-Flight; Sensing

1. INTRODUCTION

Precise indoor localization has received extensive interest due to the demand in location-based services. Innovative approaches [1] are constantly raising the bar; however, while trying to choose a positioning technique for real-world deployment that is both low cost and accurate, we find that the choices are quite limited. Accurate indoor positioning can be achieved using manual fingerprinting [2–6] or additional infrastructure [7–10], both of which are known to be costly propositions. Crowdsourcing solutions reduce the cost of fingerprinting [11, 12], but are slow to adapt to changes in the environment and depend on the willingness of the users to share sensitive sensor and location information.

Recently WiFi-based indoor localization has been revived by innovative triangulation and multilateration-based schemes [9, 13–15]. These approaches do not require any manual fingerprinting, but assume a high density of APs. For instance, EZ [14] leverages more than 100 APs and Arraytrack [9] requires several sophisticated WiFi APs with $7 - 8$ antennas. Given the pervasiveness of WiFi, the density requirement would seem to be an acceptable restriction. However, while trying to deploy our recent WiFi-based approach CUPID [15], we uncovered several practical limitations. First, multilateration requires distance estimation from at least $4 - 5$ reasonably strong WiFi APs with known locations, which are often unavailable at the edge of the enterprise network, in the developing world, and in small business locations. Second, even within the core of the enterprise network, it is difficult to find $4 - 5$ strong APs on the same channel. This is because nearby APs reside on different channels after the advent of the 802.11ac standard, which allows 21 channels in the 5GHz frequency band, and advancement in distributed channel assignment algorithms. This imposes a requirement on client devices to frequently scan across different channels to find a reliable set of APs. WiFi scanning is an energy hungry operation and can reduce the battery life of mobile devices by more than $2 - 3$ times [16], even if the scanning operation is invoked once every 10 seconds for continuous location tracking. Third, regular data communication cannot happen during the scanning operation, impacting the user experience, particularly for real-time traffic such as VoIP. To address these limitations, we need a positioning system that does not rely on multiple APs, thereby avoiding channel switching and network disruptions at the client. To achieve this, we design an *accurate single-AP localization system that works with commodity APs and avoids fingerprinting or crowdsourcing.*
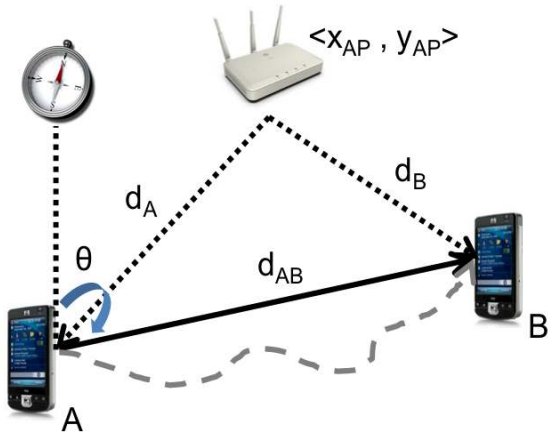
Figure 1: The triangle formed between an AP and a user walking from location A to location B.

Figure 2: Wireless signal traverses through multiple paths, one direct, and a few reflected paths.

In search of a single-AP localization solution, we studied the possibility of combining the distance and the angle of the client from the AP to determine her location. However, we found that the angle estimation granularity with recent 3-antenna commodity APs is only 60 degrees [15], resulting in large location estimation errors of more than 7m on average. Instead of relying on coarse-grained angular information, we make a unique geometric observation that enables single AP-based positioning by leveraging user mobility.

We propose a scheme based on the triangle depicted in figure 1. Let us assume that the user has a smartphone connected to an AP, with a known location, and walks from location A to location B, both of which are unknown. It may be possible to estimate the distance of the user from the AP at these locations (d_A, d_B in figure 1) using WiFi. It may also be possible to compute the user's displacement between location A and B (d_{AB}), by using her phone's motion sensors – a method called *dead-reckoning*. Accurate estimation of d_A, d_B and d_{AB} will yield a unique triangle; however, this is not enough to find the location of the user. The triangle can be rotated around the AP in any direction and will still satisfy the side length constraints. However, if we can measure the overall compass heading of the user during the path AB (θ in figure 1), it is possible to determine the orientation of the triangle in the 2-D plane, ultimately yielding the location of the user.

To realize the above scheme, it is important to accurately estimate the distance of the client based on WiFi. It may be possible to estimate the distance using signal strength (RSSI), but RSSI is known to perform poorly indoors. CUPID [15] reduces the effect of multipath on signal strength, but is fundamentally susceptible to indoor shadowing. For example, whenever the AP and the client are in two separate rooms, most of the wireless paths are absorbed, causing a large distance estimation error of more than 10m. Rather than using signal strength, we find that the signal propagation time between the AP and the client is far less susceptible to shadowing. We use the Time-of-Flight (ToF) information from commodity WiFi APs that capture the round trip signal propagation time between the AP and the client. ToF is susceptible to multipath because it captures the propagation time of the
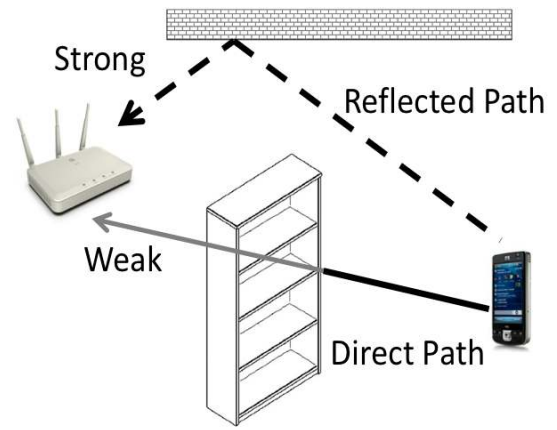
signal traversing through the strongest path and not necessarily the direct path [1](figure 2). Whenever the direct path is relatively weak due to a blockage, ToF is biased by stronger reflected components that traverse longer distances than the direct path. We exploit PHY layer information called *Channel Impulse Response (CIR)* to detect the presence of a stronger reflected path and appropriately correct the ToF value to obtain the arrival time of only the direct path. We further improve ToF-based ranging accuracy by carefully considering human mobility, ultimately reducing the median distance estimation error to 0.8m from 7m while using RSSI.

Our scheme requires precise dead-reckoning using smartphone motion sensors to determine the displacement of the user between two locations (e.g. A and B in figure 1). Although dead-reckoning techniques are quite mature [17], we find that existing techniques either impose restrictions on the user (e.g., holding the phone in a specific position), or employ map matching techniques to deal with the errors. Our new techniques are not without restrictions, but we believe that they are a step in the right direction. We design an accelerometer-based walking distance estimation algorithm that does not require user input for calibration. We detect random phone orientation changes using the smartphone's accelerometer and harness the gyroscope to identify physical turns. Our design also requires the compass heading of the user as she walks between two different locations. The compass on commodity smartphones is known to work poorly indoors due to ferro-magnetic and electrical interference [12,18]. We show that by accounting for abnormalities in the magnetometer, it is possible to determine when the compass reading is most reliable. The compass reading alone is not enough because the phone may be held in such a way that the heading of the user does not correspond to heading of the phone. We employ the accelerometer to measure this offset and thereafter obtain the correct heading.

We combine the client's distance, dead-reckoning, and heading estimates geometrically (figure 1), yielding her location using a single WiFi AP. We prototype and deploy our system

---

[1]Direct path signal is the signal component that traverses along the straight line joining the client to the AP

called *SAIL* (Single AP-based Indoor Localization), using an HP enterprise AP and commodity smartphones. Testbed results from a $30,000\text{ft}^2$ floor demonstrate that SAIL can achieve a median localization error of 2.3m using only a single AP. In addition to high accuracy and our deployment in multiple enterprises, we find that relying on only a single AP enables indoor location-based services in the developing world, small and medium business, etc.

Our main contributions are summarized as follows:

- **We utilize channel impulse responses and human mobility to eliminate the effect of multipath in ToF-based distance estimation:** By accounting for multipath, SAIL reduces the average error of ToF-based ranging from 5m to 0.8m.

- **We demonstrate how WiFi-based distance estimation can be improved by exploiting human mobility:** Our solution uses Kalman filtering to reduce the distance estimation error and adjusts its predictions after detecting physical turns using the smartphone's gyroscope.

- **We identify the opportunity to improve inertial dead-reckoning techniques using accelerometer hints:** SAIL distinguishes between most random orientation changes and physical turns. Furthermore, SAIL estimates the user's walking distance without any explicit inputs. In our experiments, these improvements reduced the mean dead-reckoning error of existing schemes from 7m to 4m.

- **We use the compass only when it is reliable to determine the user's heading:** SAIL reduces the heading estimation error to 5.9^o from 27.4^o when using raw compass values. It also identifies and nullifies the offset between the user's heading and the smartphone's heading.

- **We implement and demonstrate our solution using commodity WiFi APs and smartphones:** Our system exploits the PHY layer information obtained from Atheros wireless chipsets and is currently deployed at multiple enterprise locations.

In the following sections, we elaborate on our key intuitions and develop insights to design the SAIL system, followed by performance evaluation.

2. DISTANCE ESTIMATION USING WIRELESS TIME-OF-FLIGHT

This section elaborates on the key issues with Time-of-Flight-based distance estimation. We perform measurements in a busy office environment using HP MSM 460 APs using Atheros 9590 chipset and use these measurements to develop our key algorithms.

2.1 Wireless Signal Propagation

A wireless signal traverses in all radial directions and reflects off walls, furnitures, and other objects. Due to reflections, multiple copies of the same signal arrive at the receiver, each undergoing different delay and attenuation – a phenomena which is commonly called multipath. We define the *direct*
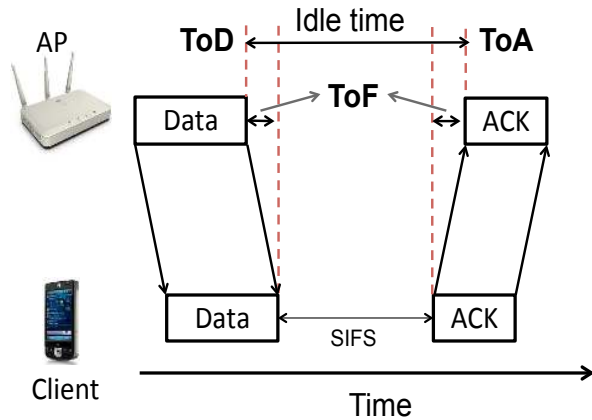

Figure 3: Computation of ToD and ToA in the PHY layer using data-ACK exchange.

path as the straight line joining the transmitter and the receiver. A wireless signal is composed of a direct path and other reflected components, each of which suffers its own delay as it propagates between the transmitter and the receiver.

Time-of-Flight (ToF) is defined as the round trip propagation time of a signal transmitted between the AP and the mobile device. WLAN ToF measurements are usually based on echo techniques [19]. These techniques employ a data-ACK exchange between the AP and the mobile device, relying on the extraction of timestamps from the main WLAN clock. The Atheros chipset can precisely compute the *Time-of-Departure (ToD)* of a data packet when it is sent out in the air at the PHY layer (figure 3). On correct reception of the packet, the client waits for an SIFS duration (a fixed value according to 802.11 standard) and starts responding with an ACK packet. The chipset also reports an estimated *Time-of-Arrival (ToA)* of the ACK packet at the AP as shown in figure 3. Both the ToD and ToA values are reported in terms of clock cycles using the 88MHz WLAN clock. Observe that the ToD and ToA values are recorded entirely at the AP and do not require any explicit synchronization between the mobile client and the AP. Moreover, since ToD and ToA are computed locally in the PHY layer of the AP, it is affected by neither clock drift between the AP and the mobile device nor the load on the main network processor.

The difference between ToA and ToD captures the channel idle time (t_{idle}) between the data and the ACK packets (figure 3). It is possible to estimate the ToF between the AP and a client based on t_{idle} because it only captures the ToF and the SIFS duration. Of course, the estimated ToF is susceptible to errors due to preamble synchronization and multipath. However, once the ToF is correctly determined, the distance between the AP and the client can be easily calculated as:

$$distance = c * ToF/2 \qquad (1)$$

where c is the speed of light and the ToF is expressed in seconds. Since the Atheros chipset employs a 88MHz WLAN clock, the resolution of distance estimation using ToF can be less than 1.7m.

2.2 Effect of Preamble Detection on ToF

It is important to precisely determine the ToA of the ACK packet; otherwise, even a small discrepancy can lead to large distance estimation errors. The ToA is determined based on preamble detection in the PHY layer; however, it is difficult to precisely detect the arrival of the preamble with nanosecond precision because of the limited bandwidth of the WLAN chipset. Figure 4(a) shows that ToA values can be quite random due to uncertainties in preamble detection.
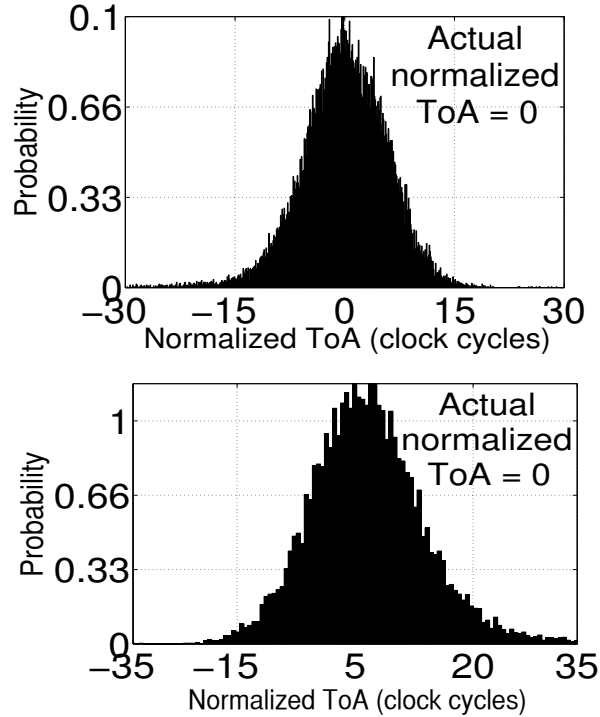

Figure 4: Distribution of normalized ToA values at the AP (a) for a static client, (b) for the same client when the LoS is blocked by a human standing in between (NLoS).

To address the randomness in preamble detection, we observe from figure 4(a) that the resultant error follows a Gaussian distribution whose mean is close to zero. This is because ToA estimates collected at different instances are subjected to independent sources of noise due to preamble detection. Thus, we use a Kalman filter [20] to minimize the effect of noise. The Kalman filter can optimally minimize the mean square error of the estimated distance based on ToF. Since the client can be mobile, we use a two-state Kalman filter that tracks both the estimated distance of the client as well as her relative speed w.r.t. the AP. In principle, the Kalman filter uses its relative speed estimate to predict the distance of the client. It then uses the current ToF-based distance value to update its distance and relative speed estimate. We will discuss in Section 5 how the Kalman filter is further optimized by correcting the relative speed estimate using dead-reckoning hints. Figure 5 shows that by addressing the errors due to preamble detection, it is possible to reduce the ToF-based distance estimation error. However, we find that the distance estimation error is still high in case of NLoS (non-LoS) environment because of the adverse effect of multipath. SAIL applies multipath correction to reduce the error due to multipath as described next.
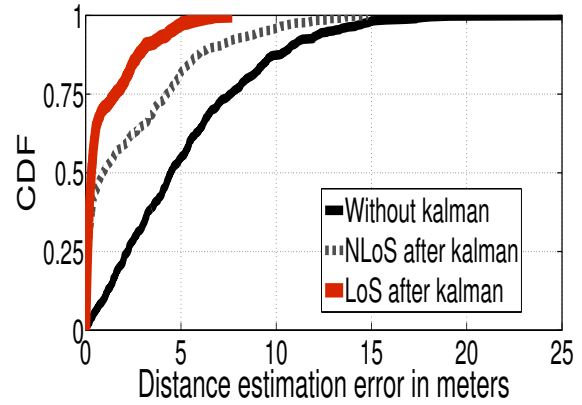

Figure 5: CDF of distance estimation error before and after applying Kalman filtering techniques for LoS and NLoS scenarios.

2.3 Effect of Multipath on ToF

Because of indoor multipath, multiple copies of the same signal arrive at the AP. The PHY layer reports the ToA value corresponding to the strongest arriving signal path, not necessarily the direct path, which has an effect on ToA estimation. ToA values can have lower error in Line-of-Sight (LoS) environments where the direct path signal is also the strongest arriving signal (figure 4(a)). However, in NLoS (non-LoS) environments where the direct path is blocked by a wall or the human carrying the mobile device, the estimated ToA may correspond to a later arriving stronger reflected path, causing a positive bias in the ToA readings (figure 4(b)). If it is possible to somehow determine the arrival time of only the direct path, distance estimation based on ToF can be significantly improved.

The direct path signal can be identified by using a PHY layer information called *Channel Impulse Response (CIR)* [15, 21]. The CIR captures the energy of the different wireless propagation paths, both direct and reflected, incident at increasing delays. Since the direct path traverses the minimum distance amongst all the received paths, it will likely appear in the earliest component of the CIR. Figure 6 shows the CIR at two different clients which are equidistant from the AP. For the first client (figure 6(a)), the direct path does not pass through obstructions, and thus yields the strongest component. However, for the second client (figure 6(b)), the direct path's trajectory is blocked by a human and hence is weaker than a later arriving reflected component. In this case, the PHY layer may report the arrival time of the reflected path as the ToA, resulting in a positive bias.

Our previous work, CUPID [15], exploited the energy of the direct path (EDP), extracted from the earliest component of the CIR to compute the distance between the AP and a client; however, its accuracy was limited because signal strength is inherently susceptible to indoor shadowing. Rather than depending on signal strength, we study the possibility of improving ToF-based ranging using CIR information. We find that by using the CIR, it is possible to estimate the positive bias in the ToA that is caused by a stronger reflected path. The difference in delay between the strongest and first component of the CIR captures the positive bias in the ToA (if any). In the
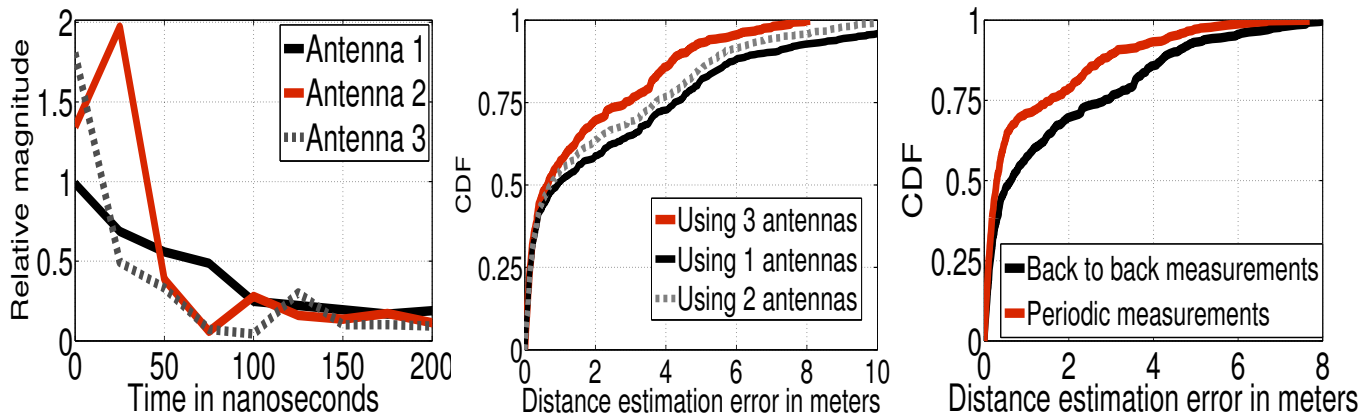
Figure 7: (a) Extracted CIR from 3 **antennas. (b) Distance estimation performance improves due to multipath correction using multiple antennas. (c) Performance with distributing 10 ToF measurements over 1s.**
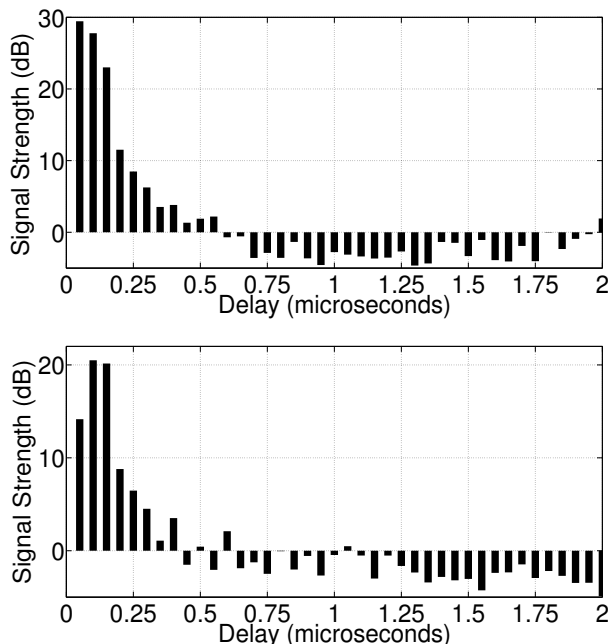

Figure 6: Estimating direct path using CIR: (a) Client is visible to the AP (LoS). (b) Client's direct signal path is blocked (NLoS).

NLoS scenario of figure 6(b), the stronger reflected path arrived 50ns later than the first (direct) path. Thus, the ToA can be reduced by 50ns to capture the arrival of the direct path in this example. However, the CIR-based correction depends on the resolution of the CIR itself. The resolution of the CIR is inversely proportional to the communication bandwidth. For example, the resolution of the CIR is only 25ns for an ACK packet received over a 40MHz bandwidth, causing distance estimation errors of up to 7.5m. As a consequence, we find that the above CIR-based correction only reduces the average distance estimation error marginally. We overcome the limitation of coarse CIR resolution by exploiting multiple antennas and human mobility as described next.

Exploiting multiple antennas: A mobile device typically transmits an ACK packet by using a single antenna, but most APs are MIMO capable and hence receive the ACK over multiple antennas. Since MIMO antennas are typically spaced apart by at least half a wavelength (6cm at 5.2GHz frequency band), the CIR varies across antennas (figure 7(a)). Nevertheless, the PHY layer reports a single ToA value of the ACK packet, even when multiple antennas are present. This is because the PHY layer synchronizes with the strongest arriving signal across all the antennas and reports the corresponding ToA. To reduce the multipath error by exploiting multiple antennas, we determine the relative arrival time (t_s) of the strongest component across all the CIRs (e.g., 2nd component of the 2nd antenna in figure 7(a)), and thereafter reduce the reported ToA by t_s. In figure 7(b) we apply the above technique to reduce the distance estimation error in NLoS scenarios where the direct path is blocked by a wall or a human being. Since increasing the number of antennas at the AP effectively increases the multipath detection resolution, we find that ToF-based distance estimation error decreases with increasing number of antennas. By correcting the error due to multipath using 3 antennas, we were able to reduce the mean distance estimation error from 2.8m to 1.6m.

Exploiting human mobility: We leverage natural human mobility to further reduce the effect of multipath. We observe that whenever the mobile device moves, the direct-path signal is usually stable, whereas reflected paths are quite random [9]. Therefore, if we compute multiple ToF snapshots over a short period, it may be possible to further reduce the error due to multipath. The multipath coherence time under mobility is typically more than 10ms [22]. Thus, it is difficult to exploit the effect of mobility on multipath by scheduling a few back-to-back measurements. Instead, we space the same measurements equally within a short time interval (1s in our implementation) to capture the arrival of the stable direct path. Figure 7(c) shows that by distributing the ToF measurements, SAIL is able to better deal with multipath reflections and hence improves the distance estimation accuracy. The performance can improve by increasing the number of ToF measurements per second; however, even with only 10 packets per second, the mean distance estimation error becomes less than 1m using an AP with 3 antennas.
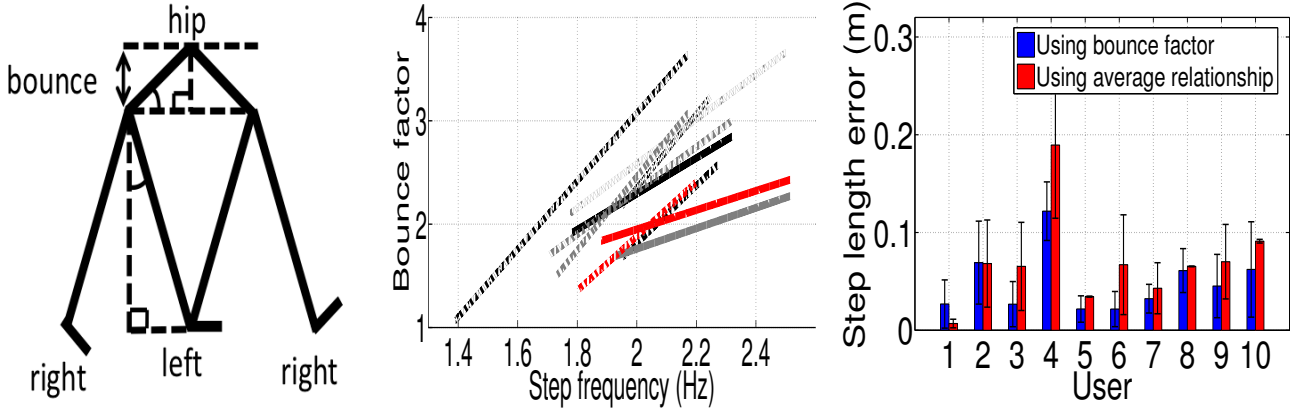
Figure 8: (a) Similar triangles are formed between the user's legs and hips while walking. (b) Step frequency vs. bounce factor relationship for ten users. (c) Step length determination accuracy across ten users.

3. SMARTPHONE DEAD-RECKONING

To enable indoor positioning using a single AP, SAIL needs to compute the displacement of the user as she walks between two locations. We use the smartphone's accelerometer to compute the walking distance and utilize the gyroscope to determine orientation changes, as described next.

3.1 Walking Distance Estimation using Accelerometer

To understand the issues with accelerometer-based walking distance estimation, we performed experiments in a 200ft corridor with 10 users for more than 4 hours. We find that when a person is walking, their body exhibits a natural periodic bounce that can be detected by the accelerometer. Counting the number of valleys in the acceleration magnitude signal allows us to track the number of steps taken by the user [12, 23]. The distance traveled by the user can be computed by multiplying the number of steps with the user's step length. We find that the step length of a user is directly proportional to her step frequency [23], which can be determined by counting the number of steps taken over a time interval. However, the step frequency vs. step length relationship varies across users (figure 9). This happens because step length depends on the physical build of the user (e.g., taller users typically take longer steps), which is difficult to ascertain without explicit user inputs [18] or personalization [23].

Rather than establishing the step length vs. step frequency relationship on a per user basis [23], we observe that users with similar physical builds are likely to exhibit similar walking behaviors. In other words, two users with similar physical build should exhibit similar lines in figure 9. We look to the vertical bounce of the user's hips as a quantifiable measurent of the user's build. Figure 8(a) illustrates how the bounce is directly proportional to step length through the geometry of similar triangles. The magnitude of the bounce manifests itself through the accelerometer. The accelerometer reports the total acceleration $\langle a_x, a_y, a_z \rangle$ experienced by the phone, which is the sum of the acceleration due to gravity $\langle g_x, g_y, g_z \rangle$ and the acceleration due to the force exerted by the user. The bounce is proportional to the variation of the acceleration component that is parallel to the gravity vector. Thus we
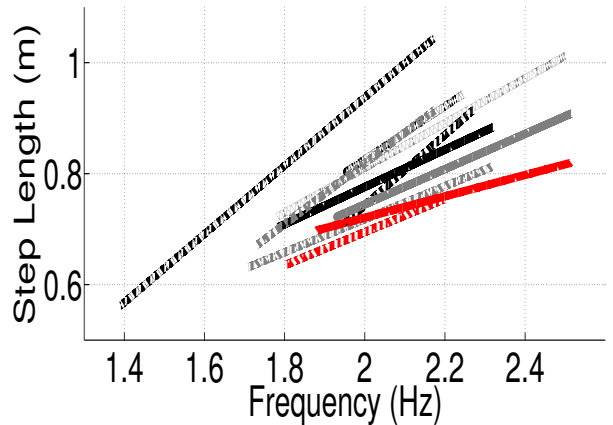

Figure 9: Step length vs. step frequency relationship for ten users.

define the standard deviation of the projections of the total acceleration onto the gravity vector as the *bounce factor*, BF:

$$BF = \text{std}\left(\frac{\langle a_x, a_y, a_z \rangle \cdot \langle g_x, g_y, g_z \rangle}{\langle g_x, g_y, g_z \rangle} \right) \quad (2)$$

Of course, the bounce factor is linearly related to step frequency (figure 8(b)) – the faster a person walks, the more their hips bounce. The position of the phone on the body also has an effect on the phone's acceleration. For instance, a phone in the user's pocket experiences more acceleration than a phone in the user's hand. SAIL's step length detection models only consider acceleration data when the phone is in the user's pocket because of the tight coupling between the phone and the user's leg motion. When a new user is tracked for the first time by SAIL, a constant step length is assumed. Once she places her phone in her pocket, it computes her step frequency and bounce factor. Thereafter, it identifies previous users who demonstrated the same bounce factor at the new user's step frequency. These users presumably share a similar build as the new user; hence, we assume that they share similar step length vs. step frequency lines, so we use an average of the ones in our model for the new user. The newly created

step length vs. step frequency line can then be used to determine her step length independent of where she keeps the phone since step frequency is not affected by various poses.

Figure 8(c) displays the results of our step length estimation algorithm through cross-validation analysis across ten users. Our algorithm generally performs better than using an average step length vs. step frequency fit, which is agnostic of the physical build of the user.

3.2 Phone's Orientation Tracking using Gyroscope

Walking distance is not enough to find the displacement of the user between her past and current location. Since the user may take turns between those locations, orientation information is also necessary. Orientation changes can be detected using the smartphone compass, but it is known to be vulnerable to indoor magnetic interference [12]. Orientation changes can be determined from the gyroscope, which reports the relative angular velocity of the phone. When integrated over time, the gyroscope yields the relative angular displacement, which can be used to detect the turns made by the user. This integration is affected by noise, causing the gyroscope to drift. The gyro drift can be addressed by performing the integration only when the gyroscope is changing substantially due to a turn [24]. However, not all the turns observed by the gyroscope correspond to physical turns made by the user. For instance, whenever the user raises her phone to take a phone call, the gyroscope may report a turn, although the user may have been walking straight during that time. Unless such events are detected, SAIL may report a turn whenever the user is toying with their phone.

We define a *pose change* as an event during which the user changes her phone's orientation without taking a physical turn (figure 10). We observe that during a pose change, the direction of the gravity vector with respect to the phone changes appreciably. For example, when the raises the phone to her ear to answer a phone call after texting with her friend, the direction of the gravity vector changes from the -z-axis to somewhere on the xy-plane. We use this observation to detect pose changes. If the direction of the gravity vector changes abruptly, we attribute any changes in the gyroscope reading during the same interval as a pose change and reset the integration of gyroscope values. Because gyroscope readings during that interval are ignored, any turns made by the user during a pose change are missed by the system. While these assumptions are less than ideal, reliable compass measurements can be used to correct the user's heading later on. Table 1 presents the results for pose change detection for 300 pose changes between five common poses. In a separate experiment, we also found that the false positive rate of the pose change detection scheme is close to zero. We conclude that regular walking seldom leads to any pose changes because the latter require a drastic change in the direction of the gravity vector.

3.3 User's Heading Estimation using Compass

Apart from the displacement of the user between her past and current location, SAIL also requires her overall compass head-
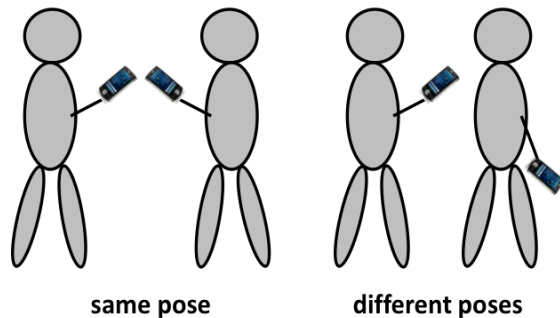

Figure 10: Pose changes occur during specific user initiated changes in phone orientation.

	Different poses				
	Head	Ear	Swinging	Purse	Pocket
Hand	X	100%	100%	100%	100%
Ear	100%	X	100%	100%	100%
Swinging	100%	100%	X	95%	100%
Purse	100%	100%	95%	X	100%
Pocket	100%	100%	85%	90%	X%

Table 1: Accuracy results for transitions between five different poses.

ing to determine the orientation of the triangle in figure 1 in the two-dimensional plane. The compass reading is based on the phone's magnetometer, which is seldom reliable indoors due to close proximity to ferromagnetic materials and electrical devices (figure 11(a)). Nevertheless, there are moments when the compass is accurate. If we can somehow detect these instances, we may be able to use the compass, along with the gyroscope, to determine the user's absolute heading.

In the absence of magnetic interference, the magnetometer should report the ideal geomagnetic field at a particular location. The World Magnetic Model [25] maintains an updated survey of this information every five years. If the user's magnetometer does not agree with the ideal geomagnetic conditions at her rough GPS location according to the survey, we may infer that there is some sort of electrical or ferromagnetic interference affecting the compass readings. Figure 11(b) shows the component of the magnetometer reading towards the earth from two different experiments. We observe that whenever the declination of the magnetic field was weak (i.e., magnitude towards earth was close to 0T), the compass was unreliable.

Not only are the magnitude and direction of the magnetometer readings important, but so too are the stability of those measurements. If the user is walking straight and the magnetometer is rapidly fluctuating between values, something is disturbing the nearby magnetic field and rendering any measurements useless. We gather the two above observations to design a *compass confidence metric* that ranges from 0 to 1, with higher values indicating more reliable readings. If m_{exp} represents the ideal magnitude of the geomagnetic field according to the World Magnetic Model and m_{meas} represents the magnitude reported by the magnetometer and projected in the direction of the ideal geomagnetic field vector, the compass confidence metric (M) is defined as:
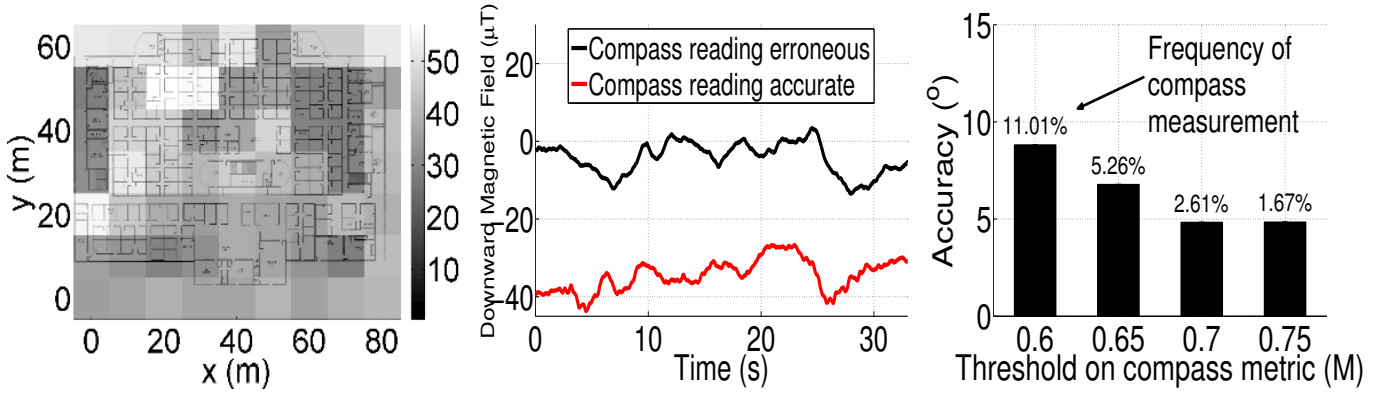
Figure 11: (a) Variation in compass error for different parts of a floor. **(b)** Magnetometer readings from two different locations with different compass accuracy. **(c)** Compass error as a function of confidence threshold.
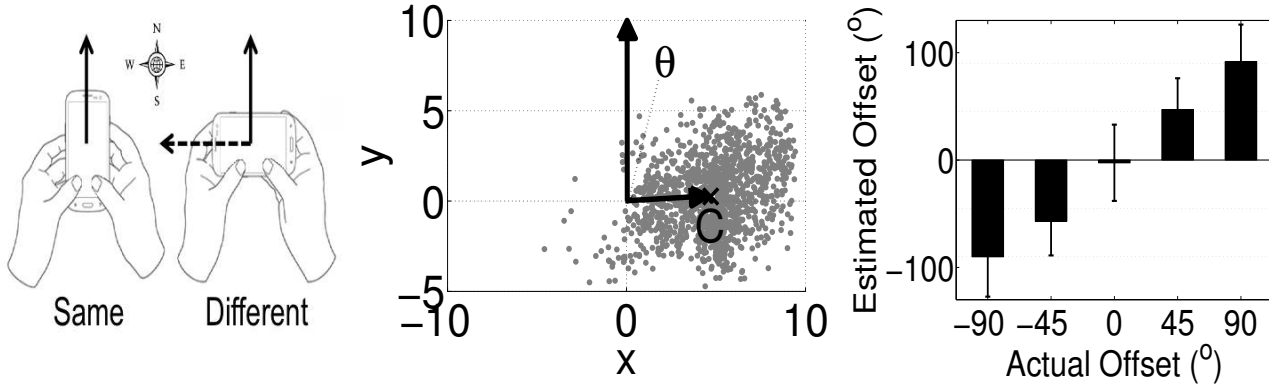

Figure 12: (a) The phone's heading is not always the same as the user's heading. **(b)** Force distribution while the phone is held in landscape mode. **(c)** Compass offset estimation accuracy.

$$M = \frac{1}{2}\left(\mu_1 \frac{dm_{meas}}{dt} + \mu_2 |m_{meas} - m_{exp}|\right) \quad (3)$$

where μ_1 and μ_2 are empirically determined constants. Intuitively, a higher threshold on the compass confidence metric will yield a more reliable compass (figure 11(c)). When the threshold is increased to 0.75, the median error for compass readings becomes $5°$. However, higher compass reliability comes at the cost of reliable reading frequency (figure 11(c)). While more frequent compass readings would be desirable, SAIL requires an accurate compass reading early in the user's path at the very least. After a reliable compass reading, the gyroscope can be used to track the user's turns, and in turn, absolute heading direction. Of course, reliable compass readings later on can be used to update the user's heading. In fact, more periodic compass readings are required if the user changes the phone's pose while turning since SAIL ignores gyroscope readings during pose changes.

Compass Offset Correction

The compass value indicates the phone's heading and not necessarily the user's heading, causing an offset in the compass reading. For example, if the user holds her phone in landscape mode (figure 12(a)), any accurate compass reading would be $90°$ off of the user's heading. When the phone is in a more arbitrary orientation (e.g., pocket, purse, ear), finding the offset between the phone's and the user's heading, called the *compass offset*, is not trivial. Previous research has mostly assumed that the compass offset is zero, or address this problem indirectly by assuming that a map of the floorplan is available. SAIL does not make any such assumptions and addresses this issue by exploiting the smartphone's accelerometer.

We observe that as the user walks forward, she exerts a force that manifests itself on the accelerometer. The heading of the user with respect to the phone may be found by determining the component of that force that is parallel to the ground. Figure 12(b) shows the distribution of these force vectors for an experiment where the user held the phone in landscape mode. Observe that the distribution of the force lies primarily along the +x-axis of the phone, indicating that the user's walking force was being exerted to the longer side of the phone. The angle of the vector, formed by joining the origin and the centroid of the force distribution (θ in figure 12(b)), is the heading of the user relative to the phone. Since the +y-axis is representative of the phone's heading, the compass value can be compensated by using the angular difference between the vector C and the +y-axis. The same process applies for more random phone orientations. Figure 12(c) shows the accuracy of the offset detection for five different angles. Between the five cases, the average error was $4°$, with the worst error of $12°$ appearing when the phone was held $45°$ counterclockwise with respect to the user.

4. LOCATION ESTIMATION USING TRIANGLES

SAIL finds the user's location by forming a triangle between the user's past and current unknown location coordinates, and the known location of the AP to which her smartphone is connected (figure 1). It uses ToF to compute the distances, d_A and d_B, of the user from the AP at two unknown locations A and B respectively. It uses dead-reckoning to calculate the distance between locations A and B, called d_{AB} in figure 1. The length constraints (d_A, d_B, and d_{AB}) determine a unique triangle, but are not enough to find the exact coordinates of the locations A and B. The triangle can be rotated at any angle around the AP, creating a large number of possibilities for locations A and B. However, the absolute heading of the line AB, as suggested by the user's compass, can determine the orientation of the triangle in the 2-D plane, ultimately yielding the user's location.

Even with the constraints discussed above, there can be ambiguity in location determination. Given a triangle that satisfies the above conditions, a second possibility can be made by reflecting that triangle across a line that is parallel to the side AB and passes through the AP coordinate (figure 13). If the user is walking straight, these triangles are identical and it is not possible to break the ambiguity. However, once the user turns, observe that the user is moving closer to the AP in one of the possibilities and further away in the other. This indicates that the distance of an intermediate point D from the AP will be different for the two candidate triangles (d_1 and d_2 in figure 13). Thus, the ambiguity can be broken by inspecting the WiFi distance at the intermediate point D and picking the triangle that satisfies this distance. In the case of figure 13, the top triangle will be selected if the distance estimate at the intermediate point of the trace is small. To avoid selecting the wrong triangle, we consider the intermediate instance when the WiFi distance measurement has the highest confidence.
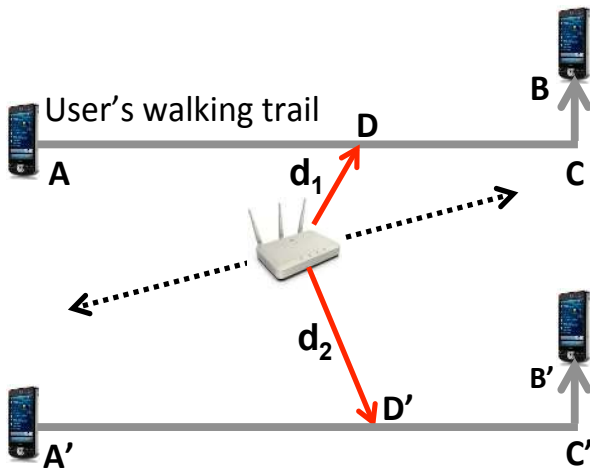

Figure 13: Two possible rotations of the triangle can still satisfy the side length and heading constraints.

5. IMPLEMENTATION AND EVALUATION

This section presents the implementation details of SAIL. We discuss SAIL's system details and end the section by evaluating SAIL's positioning performance.

5.1 CIR and ToF Processing

We obtain the CIR from the *Channel State Information* (CSI), available in the PHY layer. The Atheros 9590 chipset can export the CSI of any received packet. The reported CSI is a matrix containing one complex number per subcarrier and per receive antenna at the AP. Since the CSI captures the wireless channel in the frequency domain, an Inverse Fast Fourier Transform (IFFT) provides the CIR which captures the multipath components in the time domain. The chipset also reports the Time-of-Departure (ToD) value, in terms of clock cycles, in the transmit-complete descriptor of any data packet. Similarly, it exports the Time-of-Arrival (ToA) value of the ACK packet in its receive descriptor. For each data-ACK exchange, we estimate the ToF by using the ToA and ToD, as well as the chipset's clock frequency. We estimate the distance between the AP and a client based on ToF and further correct it using the computed CIR.

5.2 Configuring Kalman Filter States

The Kalman filter uses the past distance and relative speed estimates to correct the errors in current estimates. We initialize the filter's distance estimate using the first distance value available from the AP and the relative speed estimate as zero. Kalman filtering generally performs well, except in cases when the user is close to the AP and takes a turn. This is because the relative speed of the user w.r.t. the AP changes abruptly when she turns, confusing the Kalman filter and in turn injecting large errors in distance estimation. To solve this problem, we note that at any turn, it is possible to estimate the new relative speed by considering the user's relative speed right before the turn and the orientation change of the user from her gyroscope. Thus, whenever the user takes a physical turn, we correct the Kalman filter's relative speed estimate to account for the abrupt change. In section 5.4, we will show how this optimization can further reduce SAIL's distance estimation error.

5.3 SAIL System Details

Figure 14 plots the overall architecture of SAIL. As the user walks, the phone collects accelerometer, gyroscope, and magnetometer measurements. The phone performs calculations in real-time at every step to determine the user's step length and heading. A userspace process at the AP receives this step and heading information from the phone. It implements SAIL's distance estimation as well as triangle-based location determination. The AP schedules 10 probe packets per second to obtain the ToF estimates for the client. We use standard NULL data packets as probes because of their short length (60 bytes). Of course, if there is existing data traffic for the client, probe packets are not required. The AP receives the ToF, CSI, RSSI values from the driver and corrects the ToF using CSI to obtain the client's distance. SAIL is primarily targeted to track the location of a user as she walks. Initially, SAIL can only track the *change* in the user's location. Once she turns, however, the AP can decide between the two possible triangles and return a location estimate to the phone. From then on, the AP computes triangles only when it is confident about the WiFi-based distance estimation. Otherwise, it only uses dead-reckoning from the past estimated location and updates the user's location accordingly.
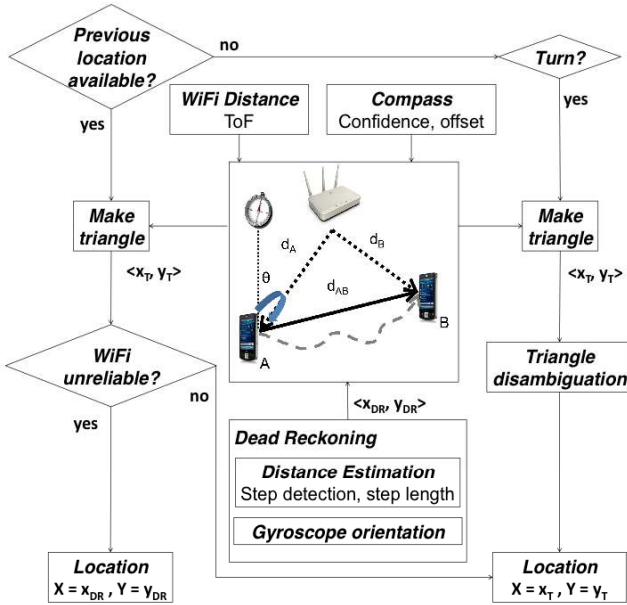
Figure 14: The architecture of SAIL.

5.4 Evaluation Results

We evaluate SAIL using HP MSM 460 APs, with Atheros 9590 chipset tuned at 5.805GHz frequency using a 40MHz bandwidth. We recruited 10 users to evaluate SAIL and used 20 different types of mobile devices. Our evaluation includes Android-based mobile devices such as the Samsung Galaxy S3, Samsung Galaxy S4, Nexus 4, Sony Xperia Z, HTC One and iOS-based mobile devices such as the iPhone 5 and iPhone 5s.

Methodology

We design real-life experiments in an office environment with a single AP installed at a known location (figure 15). Users walked around arbitrarily in the building for an hour during normal office hours, with the smartphone in their pants pocket, covering approximately $30,000$ft^2. As each user walks, the AP estimates and stores her location coordinates, which we use for our evaluation. We made separate arrangements to collect ground truth since GPS is not available indoors. Briefly, we pasted numbered markers at known locations. Whenever the user walked through a marker, she recorded its number and the current time. Since the locations of the markers are known, we did interpolation between the markers using step-count as the ground truth. The distance between the ground truth and the estimated location is SAIL's instantaneous localization error.

WiFi-based Distance Estimation

SAIL computes the distance of the client based on the propagation delay of only her direct path, called time-of-flight (ToF). Figure 16(a) shows the performance of SAIL's distance estimation error by using 10 packets per second and correcting the multipath infused error by using CIRs from 3 antennas at the AP. Further, by correcting the relative speed according to the Kalman filter whenever the user takes a phys-
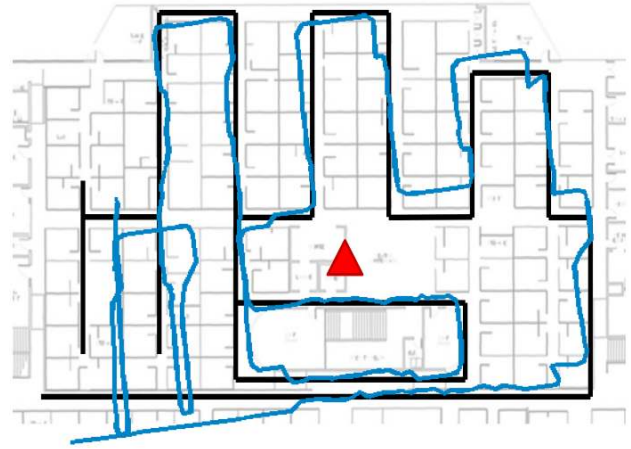

Figure 15: Evaluation floor plan with ground truth (black line), estimated walking path (blue line), and AP location (red triangle). The estimated path is slightly tilted due to compass error.

ical turn, SAIL reduces the average distance estimation error to 0.8m. This is a step forward from current wireless ToF-based distance estimation schemes that suffer large errors (figure 16(a)) from being multipath and mobility agnostic [19]. Figure 16(a) also studies if signal strength can be used to determine the client's distance. We found that RSSI performs poorly because it is highly susceptible to indoor multipath and shadowing [15,21]. Recently, CUPID [15] reduced the effect of multipath on signal strength-based distance estimation by using the energy of the direct path (EDP). However, EDP is susceptible to shadowing caused by blockage of the signal paths between the AP and the client. EDP also suffers from large errors when the direct path is weak, and thus performs worse than ToF with an average distance estimation error of 5.4m. SAIL's ToF-based approach can precisely find the distance of the client, even when her direct path signal is weak. If the direct path does not exist due to complete blockage, ToF will indeed overestimate the client's distance. It is difficult to evaluate whether the direct path signal exists in our measurements. However, we find that the distance estimation accuracy falls sharply when the signal strength of a client at the AP is below 10dB (figure 16(b)), indicating the likely absence of a direct path.

Effect of Device Heterogeneity on Distance Estimation

The idle time between data frames and ACKs must be correctly accounted for in ToF calculations. Our measurements showed that some chipsets add an offset to the SIFS time. We found that this offset is not random, but is actually a manufacturer dependent *constant* value [19]. The first bits of the ACK MAC address identify the chipset manufacturer and thus can be used to look up and correct the predetermined offset. All mobile devices carrying the same chipset have the same offset. Because the offset is a chipset-specific parameter and is not affected by antenna configuration or phone orientation, we found that mobile devices of the same type have very similar distance estimation error (table 2). Even for mobile devices using different chipsets, once the predetermined offset
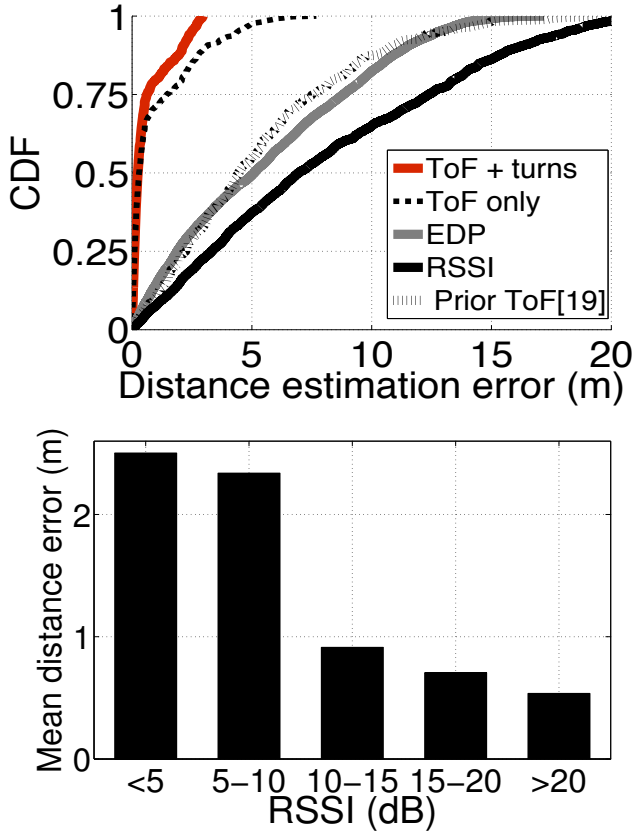
Figure 16: (a) Distance estimation error using ToF, RSSI, and EDP [15]. (b) Distance estimation improves with signal strength.

is corrected, the average distance estimation error is less than a meter (table 3).

	Distance estimation error (meters)			
Samsung Galaxy S4	0.71	0.68	0.75	0.63
iPhone 5	0.9	0.81	0.83	0.94

Table 2: Performance using 4 iPhone and 4 Samsung phones of the same type.

Galaxy S3	Xperia Z	iPhone 5S	HTC One	Nexus 4
0.93m	0.64m	0.82m	0.7m	0.9m

Table 3: Distance estimation error using different types of mobile devices.

Dead-Reckoning Performance

We use dead-reckoning techniques to determine the displacement of the user between two locations (e.g. A and B in figure 13). To evaluate the performance of dead-reckoning, we consider the distance between two random locations of the user spaced more than 20m apart. We find that existing dead-reckoning techniques are affected primarily by poor step length estimation (figure 17(a)). While most existing approaches using map matching techniques to deal with the er-
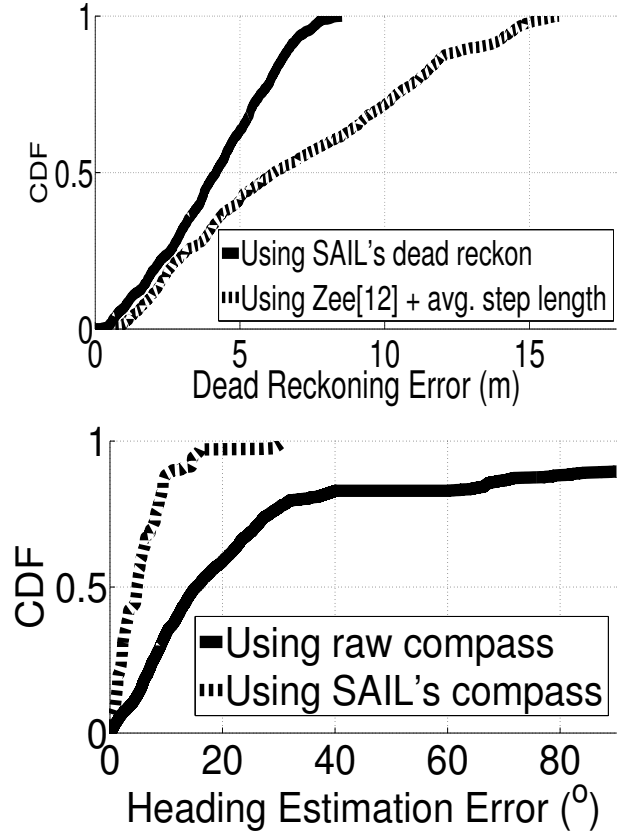

Figure 17: (a) CDF of dead-reckoning error between locations spaced more than 20m apart. (b) Heading estimation error can be large unless correct compass readings are carefully selected.

rors in step length estimation, SAIL does not impose any such restriction. By determining the user's physical build from her accelerometer readings, SAIL estimates her step length more accurately, resulting into significantly less estimation error.

SAIL also computes the user's absolute heading to determine the orientation of the triangle in figure 1. Smartphone compass-based heading estimation can be quite erroneous due to presence of electromagnetic materials indoors (figure 17(b)). SAIL improves heading estimation by harnessing the compass only when it is reliable. On average, SAIL reduces the heading estimation error to 5.9^o from 27.4^o while using the raw compass values.

Location Estimation

By combining WiFi distance and dead-reckoning, SAIL can successfully determine a client's location using a single AP. We evaluate the localization performance of the system by varying specific parameters and replacing our techniques with existing methods. Figure 15 and 18 shows SAIL's instantaneous location estimation error over a duration of 10 minutes. SAIL starts computing the user's location as soon as she makes the first turn. After the user's location has been determined for the first time, SAIL computes the triangle only when the WiFi distance is deemed reliable (RSSI > 10dB). Otherwise, it continues to dead-reckon from the past estimated location. We
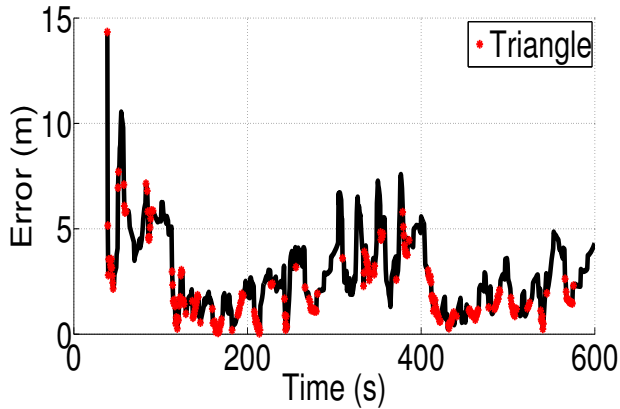
Figure 18: SAIL's location estimation error over time.

find that the errors are typically less than 5m except for a few locations where the client is far away from the AP. This is because SAIL's distance estimation is less accurate when the signal strength is weak, perhaps due to a less prominent presence of the direct path.

Impact of periodic WiFi probing: In general, SAIL probes the client every second and gathers the ToF and CIR values to compute her distance from the AP. Each such probe consists of 10 measurement packets resulting into a background traffic overhead of approximately 0.2% per client. It is important to address this overhead since it can grow quickly with the number of clients. To reduce the overhead, we study the possibility of reducing the frequency of probing in figure 19(a). We find that computing the client's distance at a reduced frequency affects our localization performance. However, with only 10 probe packets every 3.5 seconds, SAIL's average location estimation error is still less than 4m. Of course, SAIL can leverage any existing traffic between the AP and the client to further reduce the location estimation error.

Comparison with existing schemes: Location estimation based on SAIL's triangle can be performed by using existing dead-reckoning and WiFi-based distance estimation techniques at the cost of accuracy. Figure 19(b) shows how the performance is affected by using EDP-based [15] WiFi distance estimation, but maintaining SAIL's dead-reckoning techniques. EDP does not capture the client's distance as accurately as ToF does, increasing the median location estimation error to 7m. Figure 19(b) also evaluates if existing dead-reckoning schemes [18] can perform as well when augmented with SAIL's ToF-based distance estimation. We found that, unlike SAIL, existing schemes often get confused by erroneous compass measurements, resulting into large errors in location estimation.

It is also possible to find the location of the user by computing her angle-of-arrival (AoA [9,15]) at the AP and combining that information with the ToF-based distance estimate. AoA-based techniques suffer from multipath and their accuracy is fundamentally limited by the number of antennas (e.g., $60°$ granularity with recent 3-antenna APs). Thus, even after combining accurate distance estimation using ToF, AoA-based schemes such as CUPID [15] do not perform as well as SAIL.
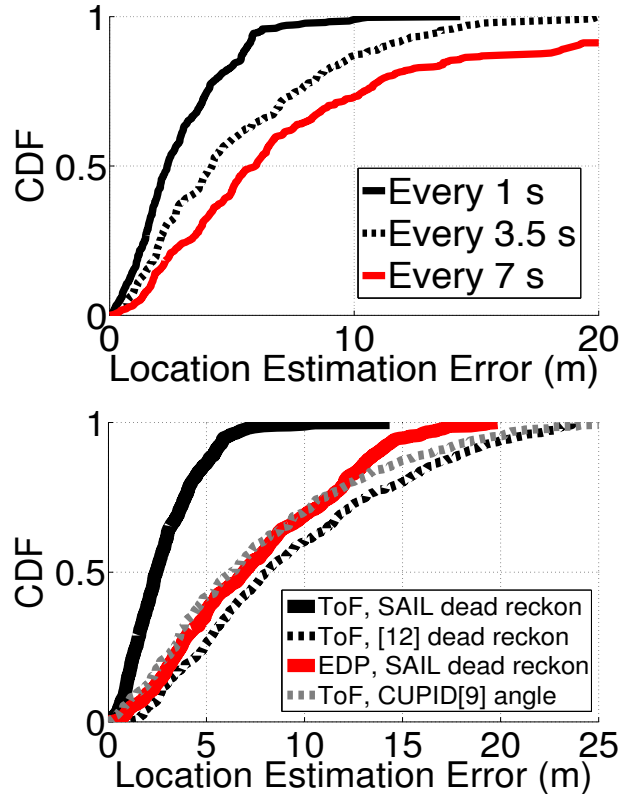

Figure 19: (a) CDF of location estimation error for different frequency of WiFi distance probing. (b) Comparison of SAIL's location estimation performance.

6. RELATED WORK

The indoor localization literature is vast [1, 26, 27], including techniques using fingerprinting [2, 5, 6, 28, 29], crowdsourcing [11, 12, 18, 23, 30], triangulation and trilateration [8, 9, 14, 15, 31, 32]. In the interest of space, we heavily subsample this literature, focusing on systems related to SAIL.

RF signal strength-based: RSSI is a poor estimator of distance because it is susceptible to indoor multipath reflections and shadowing [15, 21]. It may be possible to combine measurements from a large number of APs [14] to address the issues with RSSI, but comes at the cost of frequent scanning at the client or channel hopping at the APs. Fingerprinting techniques [2–5, 33] also rely on the availability of several APs, but provide better accuracy. However, they need periodic wardriving because of environmental and automatic WLAN configuration changes, making them an expensive proposition. FILA [21] and CUPID [15] attempt to improve WiFi-based distance estimation using PHY layer information, but do so with limited success.

Time-based: Apart from signal strength, there have been attempts to harness the time-of-flight of wireless signal propagation. Most techniques either require expensive clock synchronization between the AP and the client or additional hardware modules [10, 34, 35]. Echo techniques based on packet exchanges were first proposed in [36] and refined in [37] and [19]. However, unlike SAIL, none of the previous ap-

proaches address the effect of multipath in WiFi ToF-based distance estimation. Centaur [13] supplements RSSI with acoustic ranging. Cricket [38] and AHLoS [39] mark the difference in propagation delays between RF signals and ultrasound to determine location. Whether using signal strength or time-of-flight to compute location, all of the above solutions require measurements from multiple access points. This either comes at the cost of battery life for the user's smartphone due to excessive scanning or increased overhead at the APs due to frequent channel hopping and coordination packets. SAIL avoids these costs by leveraging only a single AP, yet still providing high accuracy.

Angle-of-arrival-based techniques use multiple antennas to estimate the angle at which the signal is received, and then triangulate the location of the signal source [9, 15, 40–43]. However, acquiring accurate angle estimation requires large antenna arrays, known antenna spacings, and synchronized CSI reports across antennas [15]. Even with access to raw signal information and using 8 antennas [9, 41], AoA techniques can be derailed in multipath rich indoor environments. SAIL's localization performance will certainly improve when accurate angle estimation is available alongside distance estimation. However, the angle estimation granularity with recent 3-antenna 11n APs is only 60 degrees [15], adding little benefit to SAIL. Moreover, while SAIL may be able to operate strictly at the ToF-capable mobile device (section 7) without requiring any explicit user turns [44], angle estimation requires multiple antennas – unforeseeable for mobile devices in recent future.

Smartphone dead-reckoning: Dead-reckoning has often been used to improve the performance of WiFi-based localization [17]. Evennou et al. [45] combine WiFi fingerprinting and dead-reckoning to refine a map-based particle filter, but do so at the cost of expensive wardriving with knowledge of the building floorplan. UnLoc [18] uses WiFi as a dimension for generating landmarks that can correct dead-reckoning traces over time; however, UnLoc restricts the position of the phone since it relies on gyroscope readings for dead-reckoning and landmark matching. Dead-reckoning has been addressed on its own as well. Existing literature has attempted to eliminate the fixed step-length constraint by either requiring user personalization or leveraging periodic GPS reading to calibrate the correlation function [46]. Park et al. [47] apply machine learning to both estimate the user's walking speed and classify between four different pose classes. While their approach is robust for the trained poses, we suspect that previously unseen poses would lead to misclassification. Instead of looking for changes in pose classification, SAIL looks for large changes in the direction of the gravity vector. Heading estimation is another important aspect of dead-reckoning. Since the compass is unreliable for accurate estimation [12], existing techniques [12, 23, 30, 45, 48] correct the compass heading using map-based particle filtering. These systems rely on indoor walking paths like long hallways for corrections to be made, rendering them ineffective in open spaces.

7. DISCUSSION AND LIMITATIONS

Acquiring CSI/ToF: SAIL leverages CSI and ToF information, already calculated in the WiFi chipset hardware for channel estimation, calibration, etc. Recent 802.11n/ac systems report CSI estimation to drivers for advanced beamforming

and MIMO [22, 49, 50]. ToF has also been available in various opensource WiFi platforms and used for distance estimation [19, 37].

Step length training: SAIL's automated step length estimation relies on the training data gathered from users with various physical builds. We performed this training through our own experiments, but we could have also gathered the data by observing GPS readings as the users walks outdoors [46]. SAIL does not require extensive training data collection since step lengths tend to have correlations between users with similar physical builds [51].

Dead-reckoning during pose changes: Pose changes indicate moments when the system is unsure about the dead-reckoning, so physical turns may be missed during these times. The compass may be able to correct the user's heading intermittently, but the orientation estimation algorithm needs to be extended for continuous dead-reckoning.

Diversity in human mobility: We recognize every user interaction with the phone is yet to be tested. Examples include when the user is walking backwards and when the phone is free to move in a user's backpack. Even so, SAIL applies fewer constraints upon the user mobility than its predecessors.

Interaction between AP and mobile device: For our implementation, we combined the dead-reckoning on the phone and WiFi distance estimation at the AP. However, if the right set of APIs are exposed, ToF-based distance computation can also be performed at the mobile device, allowing location estimation strictly at the client.

8. CONCLUSION

In this paper, we have presented SAIL, a system that methodically addresses many of the challenges towards practical WiFi-based positioning using only a single AP. While there have been numerous attempts to use pervasive WiFi for localization, most proposals impose limitations – extensive bootstrapping or dense AP deployments – that hinder their widespread adoption. To overcome the limitations, SAIL employs a single AP and still yields a decent positioning accuracy by improving WiFi ToF-based ranging and smartphone dead-reckoning. SAIL exploits PHY layer information to eliminate the effect of multipath in ToF-based ranging. SAIL advances smartphone dead-reckoning by designing robust techniques for step length, heading estimation, and pose change detection. To our best knowledge, SAIL is the first attempt to meet the single AP design goal using commodity APs and smartphones. The techniques proposed in SAIL are now operational in an enterprise network and achieve a median error of 2.3m.

9. ACKNOWLEDGMENT

10. REFERENCES

[1] Y. Gu et al. A survey of indoor positioning systems. *IEEE Communications Surveys & Tutorials*, 2009.

[2] V. Bahl et al. RADAR: An in-building rf-based user location and tracking system. In *INFOCOM*, 2000.

[3] S. Sen et al. Spot localization using phy layer information. In *MobiSys*, 2012.

[4] Yu-Chung et al. Accuracy characterization for metropolitan-scale wi-fi localization. In *MobiSys*, 2005.

[5] M. Youssef and A. Agrawala. The horus WLAN location determination system. In *MobiSys*, 2005.

[6] Y. Chen et al. Fm-based indoor localization. In *MobiSys*, 2012.

[7] G. Borriello et al. Walrus: wireless acoustic location with room-level resolution using ultrasound. In *MobiSys 2005*.

[8] P. Hu et al. Pharos: Enable physical analytics through visible light based indoor localization. 2013.

[9] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. In *USENIX NSDI*, 2013.

[10] M. Youssef et al. Pinpoint: An asynchronous time-based location determination system. In *ACM Mobisys 2006*.

[11] Z. Yang et al. Locating in fingerprint space: wireless indoor localization with little human intervention. In *MobiCom*, 2012.

[12] A. Rai et al. Zee: Zero-effort crowdsourcing for indoor localization. In *MobiCom*, 2012.

[13] R. Nandakumar et al. Centaur: locating devices in an office environment. In *MobiCom*, 2012.

[14] K. Chintalapudi, A. Iyer, and V. Padmanabhan. Indoor localization without the pain. In *MOBICOM*, 2010.

[15] S. Sen et al. Avoiding multipath to revive inbuilding wifi localization. In *MobiSys*, 2013.

[16] A. Anand et al. A quantitative analysis of power consumption for location-aware applications on smart phones. In *ISIE 2007*. IEEE.

[17] Robert Harle. A survey of indoor inertial positioning systems for pedestrians. 2013.

[18] H. Wang et al. Unsupervised indoor localization. MobiSys, 2012.

[19] D. Giustiniano et al. Caesar: carrier sense-based ranging in off-the-shelf 802.11 wireless lan. In *ACM CoNEXT 2011*.

[20] Greg Welch and Gary Bishop. An introduction to the kalman filter, 1995.

[21] K. Wu et al. Fila: Fine-grained indoor localization. In *INFOCOM*, 2012.

[22] D. Halperin. *Simplifying the Configuration of 802.11 Wireless Networks with Effective SNR*. PhD thesis, University of Washington, 2012.

[23] F. Li et al. A reliable and accurate indoor localization method using phone inertial sensors. 2012.

[24] Johann Borenstein et al. Heuristic reduction of gyro drift for personnel tracking systems. *Journal of Navigation*, 62(1):41, 2009.

[25] Stefan Maus, Susan Macmillan, Susan McLean, Brian Hamilton, A Thomson, M Nair, and C Rollins. *The US/UK world magnetic model for 2010-2015*. NOAA National Geophysical Data Center, 2010.

[26] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 2001.

[27] G. Borriello et al. Delivering real-world ubiquitous location systems. *Communications of the ACM*, 48(3):36–41, 2005.

[28] J. Hightower et al. Learning and recognizing the places we go. In *UbiComp 2005*.

[29] Y. Gao et al. Zifind: Exploiting cross-technology interference signatures for energy-efficient indoor localization. *Scanning*, 120:140, 2013.

[30] J. Hightower et al. Particle filters for location estimation in ubiquitous computing: A case study. *UbiComp 2004*.

[31] N. Banerjee et al. Virtual compass: relative positioning to sense mobile social interactions. *Pervasive Computing*, 2010.

[32] K.K. Chintalapudi et al. Ad-hoc localization using ranging and sectoring. In *INFOCOM 2004*.

[33] Yi-Chao Chen et al. Sensor-assisted wi-fi indoor location system for adapting to environmental dynamics. In *MSWIM 2005*.

[34] M. Ciurana, F. Barcelo, and F. Izquierdo. A ranging system with ieee 802.11 data frames. In *RWS*, 2007.

[35] Golden S.A. et al. Sensor measurements for wi-fi location with emphasis on time-of-arrival ranging. *IEEE TMC*, 2007.

[36] D. McCrady et al. Mobile ranging using low-accuracy clocks. *IEEE Transactions on Microwave Theory and Techniques*, 38(6), 2000.

[37] C. Hoene et al. Four-way toa and software-based trilateration of ieee 802.11 devices. In *PIMRC 2008*.

[38] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MOBICOM*, 2000.

[39] A. Savvides et al. Dynamic fine-grained localization in ad-hoc networks of sensors. In *ACM MobiCom*, 2001.

[40] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AoA. In *IEEE Infocom*, 2003.

[41] J. Xiong and K. Jamieson. Towards fine-grained radio-based indoor location. In *HotMobile*. ACM, 2012.

[42] L. Cong et al. Hybrid tdoa/aoa mobile user location for wideband cdma cellular systems. *IEEE Transactions on Wireless Communications*, 1(3):439–447, 2002.

[43] A. Tarighat et al. Improved wireless location accuracy using antenna arrays and interference cancellation. In *ICASSP*, 2003.

[44] Z. Zhang et al. I am the antenna: accurate outdoor ap location using smartphones. In *ACM Mobicom*, 2011.

[45] Frédéric Evennou et al. Advanced integration of wifi and inertial navigation systems for indoor mobile positioning. *Eurasip journal on applied signal processing*, pages 164–164, 2006.

[46] D.K. Cho et al. Autogait: A mobile platform that accurately estimates the distance walked. In *PerCom*, 2010.

[47] J.G. Park et al. Online pose classification and walking speed estimation using handheld devices. In *UbiComp*, 2012.

[48] V Fox et al. Bayesian filtering for location estimation. *Pervasive Computing, IEEE*, 2(3):24–33, 2003.

[49] E.H. Ong et al. Ieee 802.11 ac: Enhancements for very high throughput wlans. In *IEEE PIMRC 2011*.

[50] Y. Xiao. Ieee 802.11 n: enhancements for higher throughput in wireless lans. *Wireless Communications, IEEE*, 2005.

[51] J. Ahn et al. Long-range correlations in stride intervals may emerge from non-chaotic walking dynamics. *PloS one*, 8(9), 2013.