Movement Detection for Power-Efficient Smartphone WLAN Localization

Ilari Shafer
ilari.shafer@alumni.olin.edu

Mark L. Chang
mark.chang@olin.edu

Olin College of Engineering
1000 Olin Way
Needham, MA 02492

ABSTRACT

Mobile phone services based on the location of a user have increased in popularity and importance, particularly with the proliferation of feature-rich smartphones. One major obstacle to the widespread use of location-based services is the limited battery life of these mobile devices and the high power costs of many existing approaches.

We demonstrate the effectiveness of a localization strategy that performs full localization only when it detects a user has finished moving. We characterize the power use of a smartphone, then verify our strategy using models of long-term walk behavior, recorded data, and device implementation. For the same sample period, our movement-informed strategy reduces power consumption compared to existing approaches by more than 80% with an impact on accuracy of less than 5%. This difference can help achieve the goal of near-continuous localization on mobile devices.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Miscellaneous; C.3 [**Special-Purpose and Application-Based Systems**]

General Terms

Design, Experimentation, Measurement

Keywords

Localization, mobility, power-efficiency, smartphone, experimental evaluation, accelerometer

1. INTRODUCTION

The continued spread of mobile and ubiquitous computing has brought with it a desire to determine the location of portable devices and their users. Using the contextual information provided by location offers the possibility of coordinating the behavior of devices or their environments. Location-based services, which include asset tracking and routing [8], person tracking [23], phone behavior modification [24], and advertising [16], constitute a large and growing market [19].

One particular focus of localization work is in indoor environments, where many traditional location sources like GPS and cellular networks have poor coverage or are unavailable [11]. To overcome these obstacles and provide high precision indoors, systems have emerged based on radio frequency ID (RFID) tags, infrared transceivers, custom radio systems, and 802.11 WLAN infrastructure [20]. Among these technologies, positioning based on 802.11 is an attractive solution and often offers the lowest setup cost: many facilities are already equipped with access points, 802.11 devices are an inexpensive commodity, and many network users already use the platform.

A number of obstacles lie in the path of effective WLAN positioning, including system accuracy, training or site surveys, calibration, coverage, and the observance of users' privacy [18]. One design goal that has not received much attention is the device power consumption of a localization system. Energy efficiency is key for mobile platforms, which typically have highly constrained battery resources and a need or desire to operate independently for extended periods. In particular, power conservation is an increasingly valuable aspect for recent feature-rich smartphones, [7] for which many localization applications are designed.

Here, we propose a method for WLAN localization that is supplemented with accelerometer data to localize only when a user moves, thereby reducing power consumption. Since this sensor requires much less power than an 802.11 radio and its associated computational work, the system provides the same quality of location data (provided a sufficiently accurate movement-detection metric) for a longer time.

2. RELATED WORK

Indoor localization is a well-studied problem in the literature. Early localization systems such as the Active Badge (Want et al [25]) used custom hardware to detect wearable devices, and many frameworks today use specialized tags [8, 23]. Bahl et al, in developing RADAR, were among the first to demonstrate the possibility of using 802.11 networks to provide localization based on triangulation using the received signal strength indicator (RSSI) [2], and today WLAN information is even used commercially by providers like Skyhook Wireless [22].

Existing research efforts have often observed that the battery life of mobile devices that localize regularly is poor; Hightower et al find that a battery life of 10 hours is "less than desirable" [12]. Work on consumer mobile devices has been more severely limited by power usage. For example, PlaceSense (while not focusing on battery life) drained the battery of a smartphone within 4-5 hours [15].

Noting the energy conservation needs of localization applications, a number of approaches have developed to produce lower power consumption. Many aim to predict the user's mobility: the velocity history of a device can serve as a heuristic to inform sampling frequency. Xu et al simulate such a strategy, and observe a tradeoff between missing changes in location and energy conservation for any such scheme [27]. Chen et al observe the variety of efforts to reduce the power consumption of hardware on mobile phones, and instead focus on a software-based approach using access point clustering to reduce the computational and communication cost of localization [6].

From the sensor side of our approach, accelerometers have been used in localization applications to be the sole provider of data or augment the information provided to a localization service. Since many modern smartphones have a variety of built-in sensors, typically including an accelerometer, it is attractive to use the sensor either passively (i.e. to sense the environment) or actively (i.e. to determine the response to vibration). Woodman and Harle use accelerometers as part of an inertial measurement unit and exemplify the dead-reckoning approach to positioning [26], whereas SurroundSense [1] proposes using accelerometers in conjunction with light and sound information to create a location fingerprint.

Most similar to our work is the approach taken by You et al [28], which focuses on using accelerometer data to estimate the mobility of a user. Much like our approach, the authors use an accelerometer to help determine when localization should occur. However, the approach adapts its sample period to expected movement, which risks "nonconformance" due to missed samples. We aim to develop a strategy that only misses localization when movement misclassification occurs. Additionally, whereas You et al use 802.15.4-based MICAz sensor nodes and custom transmitters, we are interested in the impact on consumer WLAN devices such as commercially-available smartphones.

3. LOCALIZATION STRATEGY

Our goal in creating a localization strategy involves meeting multiple requirements. Our goals are fourfold:

3.1 Goals

1. *Reduces the power consumption of localization:* This is the primary goal of our work.

2. *Runs on consumer smartphones:* We aim to develop a strategy that can be applied to popular, functioning WLAN localization systems like those described in [22] and [4].

3. *Works with existing localization frameworks:* We would like to use an approach that augments existing WLAN services, so that increasing energy efficiency only involves modification of client-side software. For example, although moving location computation to the client reduces energy-consumptive communication [6], it requires system-wide modification and does not scale well to large environments.

4. *Does not introduce "non-conformance" error:* In contrast to approaches that increase the sample period, we develop a technique that does not miss movement due to prediction error.

These requirements inform our design choices. In order to meet (2) above, we perform our analysis consistently using the HTC Dream (aka G1) [14], which is a prototypical smartphone based on the Android software platform. To satisfy (3), we consider the system described in [4], which provides an existing on-campus localization service. Although it is not currently trained for smartphone localization, it allows us to produce the precise set of events necessary to localize.

3.2 Strategy

The strategy we develop aims for simplicity: in essence, we localize when we detect a user has moved to a new location. To do so, we poll the accelerometer of a device at a fixed period to determine whether a user is walking or not. We only perform a full localization when a user was previously walking and is no longer walking. This behavior is summarized in Fig. 1.

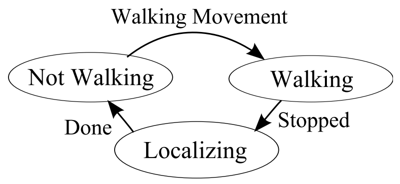

Figure 1: ***System State Diagram.*** **Information about user movement from the accelerometer mediates transitions to and from the walking state.**

A full localization event for the WLAN framework we use (represented by "Localizing") requires information about the device's location in signal space and a communication to map to physical space. In detail, a full localization comprises:

- Waking the device's WLAN radio
- Initiating a scan for nearby access points and waiting for it to complete
- Assembling a list of the RSSI and identifier for each discovered access point (a "fingerprint")
- Associating with an access point to establish a WLAN data connection
- Sending the fingerprint to a server running the localization service over the connection
- Receiving an identifier for a physical location from the server

We obtain information about user movement by polling the accelerometer periodically and analyzing the resulting data for movement events. See Sec. 6.1 for further explanation of the walk detection process, and Fig. 7 for a depiction of the alignment of localization with respect to movement.

Although our strategy uses a fixed sample rate to determine movement, we note that it does not preclude the addition of other approaches. For example, power savings from hardware-level WLAN optimization would decrease the cost of each full localization and the strategy as a whole.

4. POWER PROFILING

An analysis of the power behavior of the strategy outlined above depends upon an understanding of the energy consumption of localization, accelerometer polling, and movement detection. We describe our approach to studying these factors here.

4.1 Current Measurement

One way of determining the power profiles of WLAN access, accelerometer polling, and device sleep is through direct measurement of the current and voltage delivered by its battery. Since battery voltage over short timescales is relatively constant, instantaneous power is simply given by $P(t) = V \cdot I(t)$.

To measure voltage, we bring the device battery to the fully-charged state and measure a steady 4.15V across it. Then, we construct the current measurement arrangement shown in Fig. 2, in which a 0.1Ω Ohmite 12FR010 current sense resistor lies between battery and device. We use a Keithley 2400 single-channel source-measure unit (SMU) as the sensor labeled "V". To calibrate for the additional resistance of our arrangement, we use the SMU to perform a sweep over the voltage range we measure and linearize the resistance we observe with a small current offset.

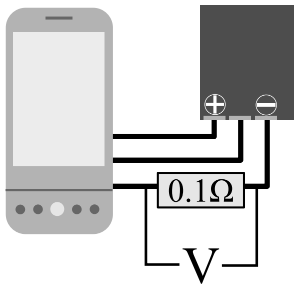

Figure 2: *Power Measurement Arrangement.* Potential difference across the small fixed resistance is measured by a source-measure unit with data logging capabilities.

To capture power consumption information for each of the activities we need for localization, we focus on measuring power for each of the following scenarios:

- *Completely Off*: Device has been manually shut down
- *Sleep*: When the device is not in use, it falls back to a sleep mode during which certain events (like calls or scheduled wakeups) can bring it back online.
- *Screen On*: The screen consumes a great deal of power, and since our test for CPU usage requires the screen to be on, this value serves as a baseline to subtract from "Heavy CPU"
- *Heavy CPU*: We produce full CPU usage by running a compute-intensive task on the device.
- *Heavy WLAN*: We attempt to determine WLAN radio power consumption by using the radio to continuously transmit and receive HTTP requests for email.
- *Accelerometer*: We write an application that performs the intermittent accelerometer polling required for movement detection and measure its consumption.

Table 1 summarizes our findings and compares them with rough estimates of their relative values. These estimates are obtained from a resource in the Android OS source code that provides values for screen, Bluetooth, WLAN, CPU, DSP, cell radio, and GPS power consumption for a generic device [10]. Many of our test conditions are best represented by a combination of these values; for example, the "Heavy CPU" task is performed at full screen brightness (screen.full) and places a full load on the CPU (cpu.full). Our measurement setup creates an additional load, which is reflected by a small measured power consumption even when the phone is completely off.

Activity	**Measured** (mW)	**Scaled** / Unscaled [type] (unspecified power units)
Completely Off	**4.18**	**4.18** / 0 [none]
Sleep	**8.63**	**8.43** / 1.6 [cpu.idle]
Screen On	**317**	**311** / 116 [cpu.idle+screen.full]
Heavy CPU	**799**	**677** / 254 [cpu.full+screen.full]
Heavy WLAN	**754**	**799** / 300 [cpu.normal+wifi.active]
Accelerometer	**321**	**269** / 100 [cpu.normal]

Table 1: *Power Measurements.* The measurements we obtain for localization-specific behaviors have similar relative values to rough estimates from power data for a generic device. For comparison, we produce the "Scaled" values by scaling the range of the unscaled Android power profile data to the power values we measure. These unscaled data are generated from combinations of CPU, WLAN, and screen power consumption.

The primary observation we make from these results is that device sleep is very inexpensive in terms of power use: its consumption is nearly two orders of magnitude smaller than heavy CPU usage. A device that remains in sleep would take more than 20 days to completely drain its battery. Additionally, polling for movement, which only requires CPU usage and the minimal power drain of the accelerometer, is less costly than performing wireless localization — particularly since it also takes less time (see Sec. 6.1).

Although these data confirm the general principle that polling for movement requires significantly less power than localization, a more specific analysis is necessary for the power consumption of WLAN scanning and access. Unlike accelerometer readings, which have a fairly constant power use, the WLAN radio has multiple levels of power usage, which depend upon the details of establishing a connection and transmitting data [3]. As shown in Fig. 3, we observe these variations in power use when obtaining a trace from our test setup for a single WLAN data request. This nonuniformity motivates our desire to measure average-case WLAN energy consumption for localization with a rundown analysis.

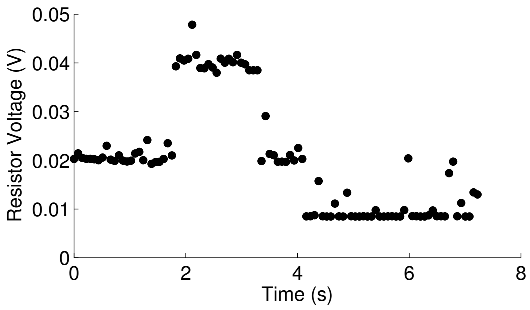

Figure 3: *Variable WLAN Power Consumption.* WLAN access and scanning has multiple power states, three of which are visible in this power trace of a single HTTP GET. Voltage is recorded every 0.073s.

We use the current measurements from Table 1 to confirm the measurements we make in the next section and to inform our choice of additional infrequent scanning to our strategy (see Sec. 7.2). Our measurements of device sleep power establish the value we use for inter-event power consumption.

4.2 Rundown Analysis

To obtain a more comprehensive portrait of power consumption, we collect information on the time required to drain the device's battery for localization (WLAN scanning and data transmission) and movement sensing. Although this process takes much longer and is less precise than current measurement, it adds average-case information about localization that we could not otherwise obtain. Our primary goal in performing this alternate analysis is to determine the relative cost of accelerometer polling and WLAN localization, which complements the relationship to sleep and CPU cost obtained above.

A key requirement of rundown analysis is a battery sensor that is consistent and accurate, though not necessarily precise. Fortunately, the HTC Dream device we use exposes a hardware battery sensor to the Android API that can meet these requirements with minimal compensation. This API reports a percentage of battery charge to user code.

To test the power characteristics of movement sensing and localization, we develop an Android application that performed the same duties as the final strategy, but that only performs one or the other activity repeatedly. Based upon the measurements in Sec. 4.1, we focus on the *per-event* energy expenditure of accelerometer and WLAN activity, since device sleep is a constant contributor that only depends on inter-event time.

Fig. 4 illustrates traces for three rundown trials at the same event rate — two for WLAN scanning to demonstrate the correspondence between repeated measurements, and one for WLAN localization. The battery sensor itself has a nonlinear sensed-battery versus time characteristic for a constant power draw. To compensate for this nonlinearity, we average all of the rundown trials that use constant power to find a aggregate battery consumption curve. We then compute the numerical inverse of this curve and use it as a lookup table for battery readings from the device. In essence, we find a discrete function that maps from average measured battery reading to a linear profile.

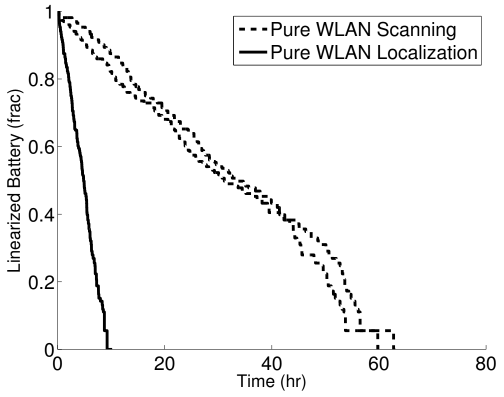

Figure 4: *Sample Battery Rundown.* Battery drain traces are shown for localization and WLAN scanning, both at 0.05Hz. Though inexact, there is a clear correspondence between traces from the same activity (as shown here, Pure WLAN Scanning). Additionally, state of charge vs time is sufficiently linear to be useful.

The rundown analysis indicates that the mapped battery measurements, while not perfectly linear, are sufficiently consistent for use in measuring state of charge. These measurements, in conjunction with the data from Table 1, form a basis for evaluating our localization strategy.

5. SIMULATION

To analyze the power and accuracy characteristics of our strategy under a variety of conditions, we simulate its long-term power consumption. The empirical data gathered from the current measurement and rundown profiles of the phone enables us to perform this simulation. Each time full WLAN localization, WLAN scanning, or movement sensing occurs, the simulation computes power cost as:

$$Cost = \frac{t - t_{last}}{TimeToSleepDrain} + EventCost$$

This model of battery consumption uses a $TimeToSleepDrain$ that would be required to fully drain the battery according to the current measurements in Table 1 and an average $EventCost$ taken from the additional data in rundown trials.

We verify that the simulation is an accurate model by running it for the same behaviors we use for rundown analysis. Fig. 5 depicts the concordance of the simulation results with experimental data. For this particular comparison, we run the experimental WLAN localization until the battery is depleted (which causes the device to shut down) and run accelerometer polling for 29 hours.

6. IMPLEMENTATION

To evaluate the real-world performance of our strategy, we write an Android application to evaluate the functionality, accuracy, and power consumption of our localization strategy. It must be able to detect user movement and exhibit consistent power consumption.

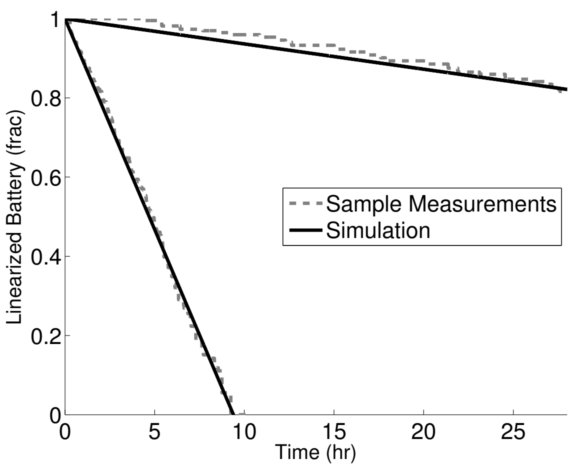

Figure 5: ***Simulation of Power Profile.*** **We run a simulation of power consumption for two simple behaviors: use the WLAN radio to localize at a fixed rate, and poll for accelerometer data at the same rate. These simulated results match experimental results for the same behaviors.**

6.1 User Movement

A key requirement of accelerometer-based movement detection is that it reliably captures periods when the user is moving. Fortunately, considerable work has been done in detecting gait from accelerometer measurements [9]. One finding we incorporate is the length of time we should sample to produce useful information about movement. We opt to sample for 350ms at the fastest update rate delivered by the software (approximately 35Hz), although the sorts of accelerometer data we expect vary based upon the location of the phone on a user's body.

The details of gait capture are in fact not critical for the movement detection algorithm, since we do not actually need to detect gait — rather, we need a boolean output of whether the user is walking. To meet this requirement, and to reduce the computation associated with sensing movement, we use a metric for movement that is the sum of the unbiased variance of X, Y, and Z acceleration:

$$\mathrm{Var}(m_1..m_N) = \frac{\sum_{i=1}^{N} m_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} m_i\right)^2}{N-1}$$

$$\text{Metric} = \mathrm{Var}(x_1..x_N) + \mathrm{Var}(y_1..y_N) + \mathrm{Var}(z_1..z_N) \qquad (1)$$

We compute the sum of squares and the sum in the variance computation as data collection proceeds, which allows us to avoid storing measurements. Also notable is that since Eq. 1 does not depend upon the mean of the accelerations; the phone can be held in any orientation. To verify that this metric suffices for movement detection (and as part of the process of selecting it), we record accelerometer data for a number of user activities. We log data to a file while holding the phone, typing on its physical keyboard, interacting with the screen, resting with the phone in a pocket, and walking.

Fig. 6 contains empirical cumulative distribution functions for walking behaviors and non-walking behaviors. It is evident that a threshold on this metric can be a good predictor of walking movement; we choose one that includes all walk events we obtained. We intentionally choose such a low threshold since excluding walking events means missing desired localization events — a situation we do not want to introduce. The impact of the classification accuracy is discussed further in Sec. 7.2.

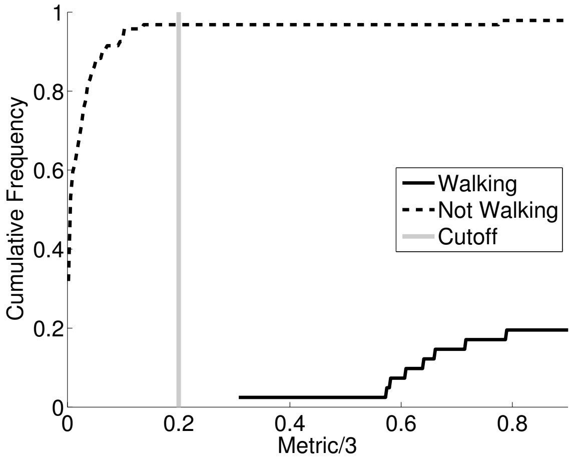

Figure 6: ***Movement Detection Threshold.*** **We aggregate all data from non-walking behaviors and walking, then compute the metric in Eq. 1. A clear distinction between walking and non-walking events is discernible, and we choose a cutoff threshold accordingly.**

6.2 Application

We design our application to be minimally dependent upon other variable power costs. To control the impact of other intermittent sources of power consumption, we stop tasks that may run based on other inputs from the environment.

To do so, we disconnect the device from the cellular network by removing its SIM card and remove the association with an email account that is typically carried by Android phones. Display power consumption is negligible, since we write the application as a background service that does not wake the screen. Also, we stop other background processes that are not part of the default operating system. These techniques mimic the use of smartphones in previous studies [15], and also ensure that our impact on the phone is limited to a standard user-level Android application.

7. RESULTS

We evaluate our movement-informed localization strategy on the basis of its long-term accuracy and power savings. These results are gathered from a simulation that is based on the data we describe in Sec. 4. To demonstrate that the strategy functions in practice, we first describe the accuracy and power savings of an implementation.

7.1 Functionality

While walking around two floors of an indoor space, we run our application on a phone that is carried in a user's pocket. Concurrently on a second mobile device, the user records "ground truth" data of when he is actually walking. We sample for movement with a fixed period, and record when the localization strategy determines that a full localization is necessary.

As shown in Fig. 7, the periods of time that the application identifies as "walk" intervals are the same as those recorded by the user, and differ in time by less than one sample period. Localization events, as expected, occur on the transition from walking to not walking. It is clear that the accelerometer-based strategy uses much less power than the battery consumption we find from localizing every sample period. We discuss power usage more thoroughly in Sec. 7.3 below.

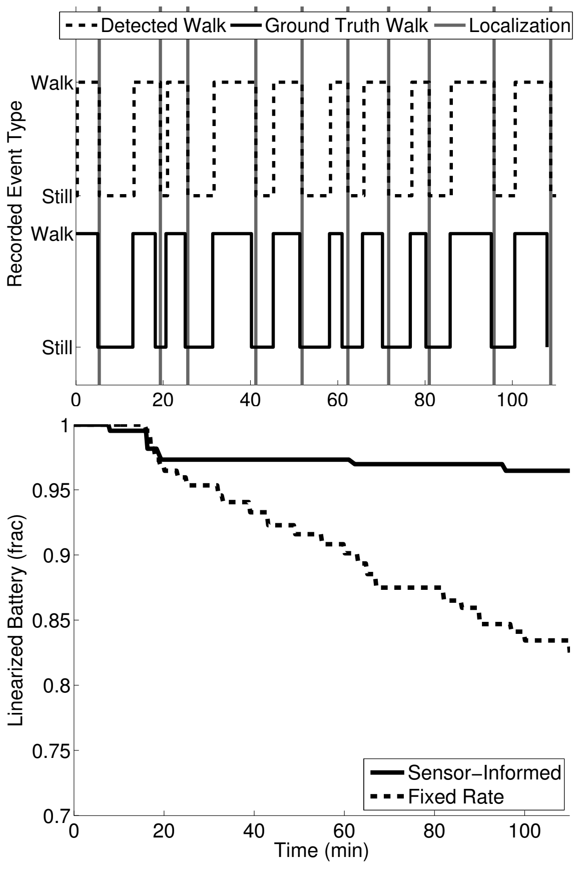

Figure 7: *Implementation Logs.* Periods of actual walking ("Ground Truth Walk") are detected by our application ("Detected Walk") and result in localization events on walk to still transitions ("Localization"). Power consumption for the accelerometer-informed strategy ("Sensor-Informed") is much lower than naïve localization ("Fixed Rate") with the same sample period (once every 20 seconds).

7.2 Accuracy

We define the accuracy of our localization strategy as the fraction of changes in location that are correctly identified. To study this metric of system performance, we simulate our strategy with walking profiles generated from observations of human walking patterns. Since the accuracy of the underlying WLAN localization system is not our focus, we remove it from our analysis: we do not track the actual location of the user.

Because a walking to not walking transition triggers a localization, there are two circumstances that can cause a missed localization. Either an entire walking period can be identified as non-walking (which may occur if no movement sample occurs within the walk), or an entire non-walking period can be considered a walk. Therefore, for our purposes of accuracy analysis, a basic long-term portrait of a human activity profile consists of the distribution of walk durations and the durations between walks (sedentary periods).

Fortunately, recent work by Chastin and Granat in quantification of human walking points indicates that the sedentary periods between walks are well modeled by a power law distribution [5]. The authors study walking patterns of individuals who are active, sedentary, and of limited mobility. We use their model for healthy active adults as a worst case for missing a span of non-walk periods, and we sample sedentary duration from a Pareto distribution with the parameters they observe. Notably, the nature of sedentary time is much less important than walking time, since sedentary periods are over an order of magnitude longer than walking times and error from missing a non-walk requires missing the entire interval.

Since accuracy is much more sensitive to the length and distribution of walking periods, we do not use a single profile but instead analyze accuracy for four diverse distributions of walk time: constant, uniform, Pareto, and normal. Each of these distributions has a minimum walk length of consideration $T_{\min} = 6$s based on the accuracy of our underlying localization system. We select a duration for each that matches total walk time to the mean of a typical adult. In detail, we choose an expected value T for each such that

$$T \cdot E[\,\mathit{WalkingPeriodsPerDay}] = E[\,\mathit{WalkTimePerDay}]$$

where we obtain $\mathit{WalkingPeriodsPerDay}$ from the distribution for sedentary periods, and we select $\mathit{WalkTimePerDay}$ based on a study by Bates et al of weekly walking time for 6626 U.S. adults [17].

Name	**Sampling Function (sec)**
Constant	$T_{\min} + T$
Uniform	$T_{\min} + U(0,1) \cdot 2T$
Pareto	$T_{\min} + (U(0,1))^{-2/3} \cdot T/3$
Normal	$T_{\min} + \max\left(\mathcal{N}(T, 144)\right)$

Table 2: *Walk Time Distributions* We use multiple models of walking time to assess strategy accuracy. $U(a,b)$ represents a uniform random number between a and b, and $\mathcal{N}(\mu,\sigma^2)$ a normally-distributed random number with mean μ and variance σ^2. The Pareto distribution used for sampling has $\alpha = 1.5$.

Using this definition of accuracy and the distributions in Table 2, we consider two primary sources of error: missing an event due to *not sampling*, and *misclassifying* events. We do not adapt the sampling period since our goal is to not miss localization due to prediction error. Therefore, any error due to not sampling will be produced by checking for movement with a period longer than the shortest walk time. This shortest period is $T_{\min}+T$ for the constant distribution and $T_{\min}$ for the others shown in Table 2; periods below this threshold produce no error from a failure to sample for movement. For other sampling periods, Fig. 8 shows a simulation of the dependence of accuracy on sampling period assuming perfect classification. The choice of a low sample period is necessary to maintain accuracy.

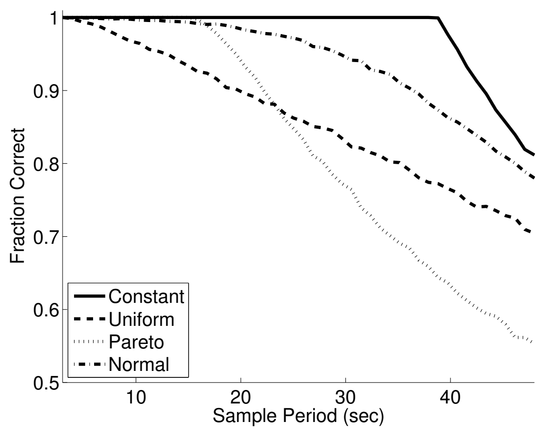

Figure 8: ***Impact of Sample Period on Accuracy.*** **The impact of sample period on localization accuracy as defined in Sec. 7.2 is shown for the four synthetic walking patterns shown in Table 2. Though the dependence varies, only short sampling periods provide high accuracy.**

Such short sampling periods are similar to those used in previous systems. Kim et al [15] sample once every 10 seconds, and Want et al intentionally save power by sampling once every 15 seconds [25], noting that the period is long but sufficient for office environments. You et al [28] use an adaptive sampling period with a non-conformance rate that is typically 10% to 17%, but still poll for movement as often as 2-4 seconds.

Given that a short sampling period is necessary for accuracy, we are most interested in the impact of misclassification. Misclassifying an entire walk is the primary error introduced by the movement-sensing system, and a significant impact of classification error on accuracy would run counter to our goal of not introducing additional error. Fortunately, two properties of movement sensing cause a small impact. First, as we observe in Sec. 6.1, a simple metric can be very accurate in classifying walking/non-walking intervals — the threshold we choose does not misclassify any walk events in our sample dataset. Even for greater misclassification rates, though, the strategy will often sample again within a walk. The probability of an incorrect classification is then the product of multiple incorrect classifications. As simulated and illustrated in Fig. 9, the accuracy of the strategy is robust to more incorrect classification than we observe in our tests.

One additional accuracy-related concern is the outcome if a localization is missed and a user does not move for a long period of time. In this case, the strategy will not relocalize until another walk event occurs. To avoid this situation, the system can check the user's location in signal-space using a WLAN scan on an infrequent basis when no walking is detected. Although a single WLAN scan consumes approximately 2.2 times as much power as a movement detection, it consumes 10 times less than full localization. The combination of infrequent scanning and relatively low per-event cost produces a low energy impact, as discussed below.

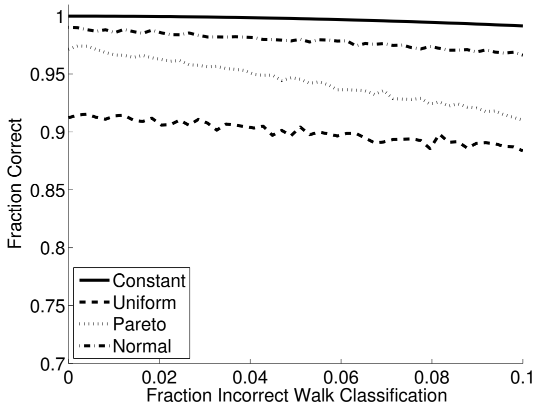

Figure 9: ***Impact of Misclassification on Accuracy.*** **Localization accuracy is not highly dependent upon walk misclassification in the low range we experience, even for relatively long sampling periods. Here, a sampling period of 17.9s is used for each walk time distribution.**

7.3 Power

We evaluate the power consumption of our strategy using the simulation techniques described in Sec. 5 for the same distributions of walk and non-walk times used for the accuracy analysis above. To obtain a worst-case scenario for power consumption, we use a 0% misclassification rate; this ensures that a maximum number of energy-expensive localization events are performed after performing movement detection. We also include a WLAN scan to localize the phone in signal space every five minutes to avoid the potential long periods of incorrect location information introduced by any missed localizations.

Since the strategy parameter that affects power consumption the most is sample period, we compute power usage as a function of sample period for each of the distributions in Table 2. Our baseline for comparison is full (naïve) localization with the same period. Fig. 10 presents the reduced power cost for the movement-sensing assisted strategy as compared to baseline energy expenditure. For the short sampling periods of interest, we observe a computed reduction in power consumption that exceeds 80%. The comparative savings are reduced for longer sample periods primarily due to the cost of inter-event phone sleep, which causes a tapering of device lifetime.

We also investigate the power behavior of our localization strategy for a few individual activity profiles to confirm that our aggregate model for walk times encompasses actual walks. A few data samples provided by PAL Technologies [21] serve this purpose well. These profiles are recorded using a custom device and an algorithm that isolates intervals of sitting, standing, and walking. For example, Fig. 11 shows a simulation of power consumption for the walk pattern of a 41-year-old executive. The power savings simulated for the HTC Dream when using a 20 second sample period (92.8%) and a 10 second sample period (94.3%) are in line with the savings predicted by the walk model and shown in Fig. 10.

Data from other smartphones indicate comparable power savings. We perform rundown profiles of the HTC Magic (aka MyTouch 3G), generate a battery sensor linearization,

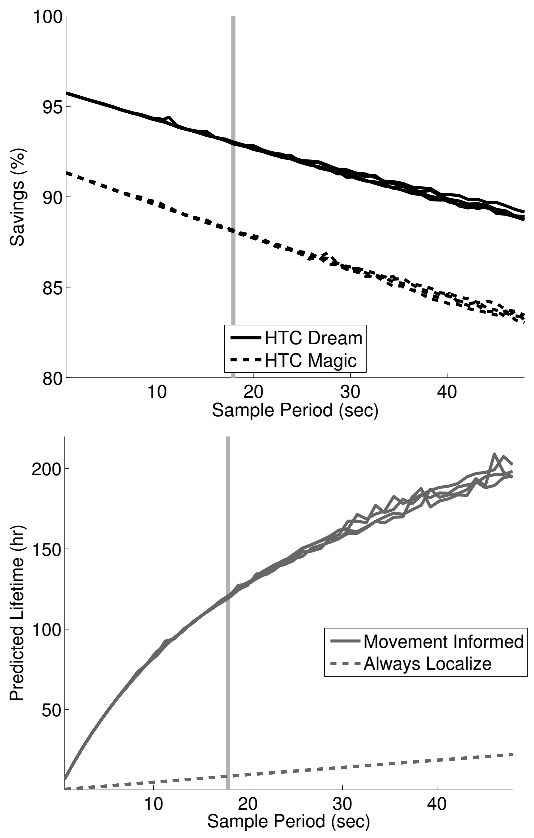

Figure 10: ***Power Savings.*** **Power consumption for our localization strategy is much lower than that for localization for the same sample period. All distributions for walk time are overlaid for both our strategy ("Movement Informed") and naïve localization ("Always Localize"). The light gray vertical bar indicates the sampling period of 17.9s that is used in Fig. 9.**

and conservatively assume that its sleep lifetime is the same as the HTC Dream we discuss earlier (even though its actual sleep lifetime is longer) [13]. The simulated power savings for this device are illustrated in Fig. 10. Our initial profiling of another smartphone (the Motorola Droid) shows similar promise for power savings, although further work is needed to fully characterize the device, in part since its battery sensor provides a much coarser measurement.

For both the short duration implementation results shown in Fig. 7 (which show a power savings of 79%, possibly due to the association costs we discuss in Sec. 8.2), and in the simulated long-term behavior in Fig. 10 and Fig. 11, we observe a considerable power savings for our strategy. These results indicate great promise for movement-informed localization on consumer smartphones.

7.4 Comparison with Other Methods

Since we focus on improving the battery life of smartphones when localizing, the accuracy and power use of other localization systems on these devices ought to be considered. Unfortunately, a direct comparison is problematic, since the goals and requirements of the technologies differ. For example, GPS-based localization is largely orthogonal to our goals: although its accuracy can be 1-5m outdoors [20], it is much less effective indoors, requires a longer time to associate than WLAN and consumes greater than 50% more power than WLAN while active [10].

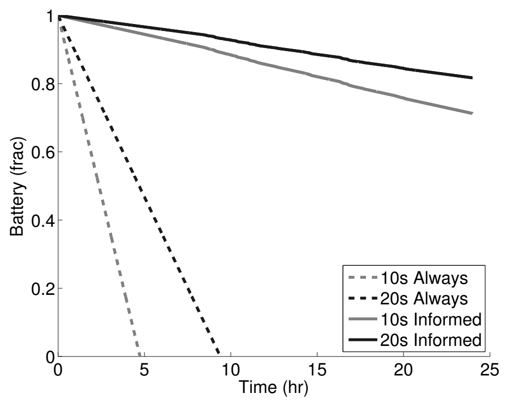

Figure 11: ***Example Walk Power Use.*** **We use data on walk periods to compute the energy expenditure of always localizing ("Always") and our strategy ("Informed") for two different sampling periods. No events are missed by either sampling period.**

A more comparable localization mechanism on consumer devices uses the identity of towers on the cellular network. Typical accuracy for these systems is very low (on the order of 50-200m). Prototype systems have achieved a resolution as fine as 2.5m in the best case, but require a dense set of neighboring cells [20]. Accuracy for WLAN-based systems is typically much greater: for example, the underlying framework we use is accurate to within 10m for 94.9% of localizations [4], and multiple systems with greater complexity and training are consistently accurate to within less than 3m [18–20]. The relative energy consumption is dependent upon the type of network used: for a single full localization requiring 1-2KB of transfer, power consumption based on the findings of [3] would be 7J for WLAN, 4J for GSM, and 13J for the 3G radios popular in smartphones.

Efforts most closely related the present work show considerable promise but are not directly comparable. For example, Chen et al reduce CPU power consumption required for offline localization, which is unlike our server-based localization framework [6]. The work of You et al mentioned earlier achieves up to a 68.92% reduction in power consumption as compared to periodic sampling as compared to the greater than 80% reduction we predict. However, these results are not directly comparable: the authors concurrently focus on improving accuracy through predictive sampling. Additionally, both studies use wireless sensor nodes, which have different power characteristics than the smartphones we study. In this spirit, since our emphasis is on consumer devices and their users, we focus our comparison on improvement for their situation in the next section.

7.5 Predicted Benefits

We intentionally study the power consumption of our strategy without other user applications running on the device to control our experiment. However, actual users of a localization service will likely perform other activities like making calls. Power usage on smartphones in particular is heavily dependent on activity: as easily drawn from the component power consumption values in Table 1 and observed by users, device lifetime can range from less than 10 hours for continuous heavy network and media use to longer than 48 hours for intermittent general use.

To capture the expected impact of user activity, we predict the lifetime of the device by simulating additional power consumption alongside localization. Since battery life varies so widely based on activity, we choose a range of expected battery lifetimes (24-48 hours) for user activity only based on typical user behavior of charging every day or every other day. We then find the average power consumption that produces these lifetimes, and use it as the power use between localizations in a simulation with a 20s movement detection period. Clearly, movement detection and user activity will overlap at times with such a model; however, since movement detection takes only 350ms of every 20s period (1.75%), we find that the effect of overlap on lifetime is less than 4%.

Device Activity	**Lifetime (hours)**
Only Sleep	493
Movement-Informed Localization	130
Always Localize (only WLAN)	8.7
Typical User Activity	24–48
Typical User Activity + Movement-Informed Localization	20–37
Always Localize (only WLAN) + Movement-Informed Localization	6.5–7.5

Table 3: *Predicted Lifetime with Other Activity.* Using movement-informed localization in the presence of other power consumption remains advantageous. We use a sample time of 20s for both localization strategies.

Table 3 summarizes the impact of our movement-informed localization strategy on device lifetime. Using our strategy with the 20s sample period used in our studies of power produces a 130h lifetime in the absence of user activity. It would allow users to keep to the same charge schedule: for a one-day original lifetime, it only reduces device lifetime by 4h, and reduces it by 11h for a 48h original lifetime. This time window allows for much more usable localization applications than using existing WLAN-only localization approaches: a 6.5–7.5h lifetime demands multiple charges per day. Therefore, we find that users of our target devices (smartphones) stand to benefit from localization services built around movement-informed localization while feeling little impact on their usage patterns. Some optimizations and study, detailed below, could produce even further benefits.

8. LIMITATIONS AND FUTURE WORK

Although the power savings afforded by movement detection are considerable for minimal cost in accuracy, there are considerations that limit its use. Perhaps the most significant is the potential for a long span of incorrect location information due to a missed localization. Intermittent scanning, as described in Sec. 7.2, can ameliorate this concern, but hardware to perform continuous sensing would improve both power use and accuracy.

8.1 Continuous Sensing Hardware

One of the current limitations of performing movement sensing on smartphones like the one we use is the necessity of waking the CPU to obtain data from the accelerometer. This property causes power usage that, while not as large as localization, is still much larger than device sleep. Additionally, the need to poll the accelerometer makes it infeasible to perform continuous movement sensing. In contrast, many sensor nodes have the capability to wake only when a given sensor's measurement exceeds a threshold value [8].

Small changes to smartphone hardware could allow for wake-on-accelerometer behavior, thereby permitting lower power use and higher accuracy through continuous sensing. For example, the smartphone we study contains a Qualcomm PM7500 power management IC that manages waking the CPU. It contains its own housekeeping ADC, which could be used in place of the ADCs on the CPU for low-power sensing and triggering the CPU to wake if necessary.

8.2 WLAN Association Cost

We observed a power savings for an actual implementation that was slightly lower than predicted from measurements and simulation. The likely source of this cost is the additional energy expenditure needed to associate with a new WLAN access point. When walking, we model actual behavior by walking between different locations, which had different access points that require authentication. Future work could reduce this reassociation cost — perhaps by maintaining a unauthenticated network limited to localization or performing improved caching of connection information at the session layer.

8.3 Integration with Outdoor Localization

Our focus in this work is on saving power for WLAN localization. Similar movement-sensing approaches are also applicable in outdoor environments, where WLAN localization is often not possible or not preferable. The power cost of common outdoor localization schemes based on GPS and cellular signals is also high, and could benefit from sensor-informed approaches like the one we develop here. Extending the present work to a comprehensive power study and localization system that includes multiple techniques could make it suitable for deployment among a much broader audience. Such a deployment would provide the opportunity to study implementation power behavior on phones that undergo typical use. This work would help address the need for deeper study of the accuracy and power use of the strategy under population workloads.

9. CONCLUSION

We describe an approach to WLAN localization that aims to reduce power consumption. By only localizing when a user has moved, our strategy leverages less energy-expensive sensors to achieve the same sample period as fixed rate localization with a much lower power cost. Our target platform is the recent collection of consumer smartphones, which provide WLAN radios and an accelerometer that we use for movement sensing.

Our study of the power consumption of movement sensing and localization on a smartphone allows us to simulate our strategy for a number of movement profiles. We show that for three cases — long-term models of human walking, recorded activity profiles, and actual implementation —

movement-informed localization reduces power consumption drastically with only a small impact on accuracy. We also show that our methodology can be successfully combined with normal phone user patterns with acceptable power impact and minimal impact on users' charge schedules. Small changes at the hardware level and expansion to other forms of localization could produce even further improvements for this strategy, and lead to power saving for location-based services and their users.

Acknowledgment

The authors would like to thank Andrew Barry, Noah Tye, and Brad Minch for their thoughtful input. Ilari Shafer is supported by an Olin Scholarship.

References

[1] M. Azizyan and R. R. Choudhury. SurroundSense: mobile phone localization using ambient sound and light. In *ACM SIGMOBILE Mobile Computing and Communications Review*, volume 13, pages 69–72, 2009.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. IEEE Infocom 2000*, pages 775–784, 2000.

[3] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 280–293, 2009.

[4] A. Barry, B. Fisher, and M. L. Chang. A long-duration study of user-trained 802.11 localization. In *Mobile Entity Localization and Tracking in GPS-less Environments (MELT)*, pages 197–212, Sept. 2009.

[5] S. Chastin and M. Granat. Methods for objective measure, quantification and analysis of sedentary behaviour and inactivity. *Gait & Posture*, 31(1):82–86, 2010.

[6] Y. Chen, Q. Yang, J. Yin, and X. Chai. Power-efficient access-point selection for indoor location estimation. *IEEE Transactions on Knowledge and Data Engineering*, 18(7):877–888, Jul. 2006.

[7] G. B. Creus and M. Kuulusa. *Mobile Phone Programming*, chapter 25: Optimizing Mobile Software with Built-in Power Profiling, pages 449–462. Springer Netherlands, 2007.

[8] Ekahau, Inc. Ekahau Products. `http://www.ekahau.com/products.html`.

[9] A. Godfrey, R. Conway, D. Meagher, and G. ÓLaighin. Direct measurement of human movement by accelerometry. *Medical Engineering & Physics*, 30:1364–1386, 2008.

[10] Google Inc. Android source code. `http://source.android.com/download`.

[11] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34:57–66, 2001.

[12] J. Hightower, G. Borriello, and R. Want. SpotON: An indoor 3D location sensing technology based on RF signal strength. Technical Report 2000-02-02, Univ. Washington, Feb. 2000.

[13] HTC Corp. HTC Magic. `http://www.htc.com/www/product/magic/specification.html`.

[14] HTC Corp. T-Mobile G1. `http://www.htc.com/www/product/g1/specification.html`.

[15] D. H. Kim, J. Hightower, R. Govindan, and D. Estrin. Discovering semantically meaningful places from pervasive RF-beacons. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 21–30, 2009.

[16] B. Kölmel and S. Alexakis. Location based advertising. In *The First International Conference on Mobile Business*, 2002.

[17] J. Kruger, S. A. Ham, D. Berrigan, and R. Ballard-Barbash. Prevalence of transportation and leisure walking among U.S. adults. *Preventive Medicine*, 47(3):329 – 334, 2008.

[18] J. Letchner, D. Fox, and A. Lamarca. Large-scale localization from wireless signal strength. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2005.

[19] H. Lim, L. Kung, J. C. Hou, and H. Luo. Zero-configuration, robust indoor localization: theory and experimentation. In *Proceedings of IEEE INFOCOM*, pages 123–125, 2006.

[20] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 37(6):1067–1080, Nov. 2007.

[21] PAL Technologies LTD. Complete record for subject i session 1 day 1. `http://www.paltech.plus.com/examples/index.html`, Aug 2001.

[22] Skyhook Wireless, Inc. `http://www.skyhookwireless.com/`.

[23] Sonitor Technologies, Inc. `http://www.sonitor.com`.

[24] two fourty four a.m. LLC. Locale. `http://www.twofortyfouram.com/`.

[25] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 40(1):91–102, Jan. 1992.

[26] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing*, volume 344, pages 114–123, 2008.

[27] Y. Xu, J. Winter, and W. Lee. Prediction-based strategies for energy saving in object tracking sensor networks. In *Proceedings of the 2004 IEEE International Conference on Mobile Data Management*, 2004.

[28] C.-W. You, P. Huang, H. hua Chu, Y.-C. Chen, J.-R. Chiang, and S.-Y. Lau. Impact of sensor-enhanced mobility prediction on the design of energy-efficient localization. *Ad Hoc Networks*, 8(8):1221–1237, Nov. 2008.