2012 International Conference on Indoor Positioning and Indoor Navigation, 13-15th November 2012

Classifying and Using Motion in Organic Indoor Positioning

Álvaro Fialho, André M. Cavalcante, Afonso Costa
Nokia Institute of Technology (INdT), Manaus, Brazil
Email: {alvaro.fialho, andre.cavalcante, afonso.costa}@indt.org.br

Jonathan Ledlie
Nokia Research Center, Cambridge, USA
Email: jonathan.ledlie@nokia.com

***Abstract*—Current mobile devices continuously estimate their locations, allowing users to "check in", find nearby friends and interests, and determine routes to their destinations. While underlying satellite, cell, and WiFi-based positioning systems can return an accurate and meaningful position in many cases, extending them to work energy-efficiently, particularly indoors, remains an open problem. In this work, we study energy-efficient and robust human-scale motion classifiers and their use in room-grain, collaborative indoor positioning systems. Previous work on improving energy-efficiency in positioning systems has assumed sensor input from an energy-cheaper alternative: using an accelerometer in lieu of GPS, for example. Unfortunately, even these alternative sensors are not practical for everyday use because of their own energy consumption, at least when sampled continuously. After studying what accelerometer sampling rates are feasible, we compare six methods for motion classification, two of which are new. We find that the existing simple statistical methods are not sufficiently robust with respect to different kinds of movement and different users, because the thresholds between movement and non-movement are too tight. In contrast we find that the two new, more sophisticated models, one based on Page-Hinkley statistics, and the other inspired by the Discrete Fourier Transform, provide a clearer differentiation between the two states. Only the Page-Hinkley-based one is as energy-efficient as the simple statistical methods, however. Through a WiFi geolocation system that relies on motion detection, we show how the choice of the underlying motion classifier can have a significant impact on user-perceived performance.**

***Index Terms*—Motion detection, Page-Hinkley statistics, discrete Fourier transform, energy efficiency, geolocation, crowd-sourcing.**

I. Introduction

Detecting and acting upon human-scale motion has many low-level uses for mobile devices. These range from activating power-hungry sensors that had been quiescent to notifying applications and positioning systems that a meaningful change has occurred. Here we study the problem of using sensor data from an accelerometer to efficiently determine whether the device's user has moved "significantly," where a significant change would require the recomputation of the user's position due to a potential room change. While several pieces of prior work have studied this problem as part of making their positioning systems more accurate or consume less energy, we find that the algorithms used were either quite energy-hungry themselves or are not robust to motion from different people. We examine how to make this human-scale motion detection energy-cheap and handle different people, and study how a crowd-sourced, or *organic* WiFi-based indoor positioning system can benefit from a good motion detector.

978-1-4673-1954-6/12/$31.00 © 2012 IEEE

The chief strategy to make positioning more energy efficient has been to detect quiescence, or the lack of human-scale movement, and turn off or duty-cycle the energy-hungry sensor: GPS or WiFi depending on the positioning method in use. If the device has a good position estimate and the device (person) has not moved, do not do anything. More complex strategies form hybrids of these two sensors with others, particularly the accelerometer and cell radio, to find a low energy cost method for the current precision requirements. For example, if a user's location can be constrained to roads by having the accelerometer's signal reveal the user is traveling in a car, an energy-cheap cell-based position will be sufficient for most applications [1]. We provide a new energy-efficient and accurate method for discerning between quiescence and movement that is applicable to any of the three main types of positioning systems.

A second-order strategy has been to act – to change the behavior of the positioning system – in response to human-scale movement. For example, if the user moves (significantly), then re-activate GPS and, in the meantime, lessen the precision based on the range the user could have moved. These responses to movement are typically specific to the type of positioning system in use: satellite, cell or WiFi. Through an implementation of our motion detector in a crowd-sourced WiFi positioning system [2], we show several new ways to include movement and quiescence directly into its internal behavior, lowering energy consumption and increasing room-grain positioning accuracy.

We first take a detailed, practical look at how make a binary classification of human-scale quiescence and movement. While sensors like the accelerometer are considered energy cheap, they are not that cheap: continuously sampling the accelerometer for five minutes expends more energy than one minute of GPS [3]. Given that the accelerometer too must be sampled, we compare previous techniques for motion detection against two new methods at sparse sampling rates. We find that the methods we propose, based on Page-Hinkley statistics and the Fourier transform, are both more accurate than previous work, but that the Page-Hinkley method is as energy-efficient as the previous work, while the FFT-based method is not.

Next we propose several new methods for using the output of the binary motion classifier in a crowd-sourced WiFi positioning system. In particular, we show how using motion can create more accurate WiFi fingerprints, and how it

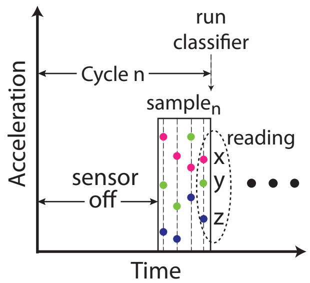


Fig. 1. Our model of energy-efficient accelerometer-based motion classification. Duty-cycling the accelerometer yields a short sample from which the classification is made.

decreases the latency in showing a correct, new estimate to users when they have moved. We also suggest how quiescence, together with estimate confidence, can be used to passively add reinforcing fingerprints to the crowd-sourced database.

This paper makes the following contributions:

- We show that a statistical change-detection algorithm provides more accurate classification of motion detection than previous methods that are based on simpler statistics, and that it is more energy-efficient than equally accurate FFT-based approaches.
- We examine different parameters for accelerometer duty cycling, finding a good trade-off between energy expended and missing user movement.
- We demonstrate the importance of motion detection for both signature creation and use within a crowd-sourced WiFi positioning system through a multi-floor, multi-user experiment.

The first main sections of the paper cover motion/quiescence detection for positioning systems in general: first providing an overview of the problem (Sec. II), then describing several approaches to classifying the signal emerging from a duty-cycled accelerometer, including two new methods (Sec. III). Next, we describe the relevant components of a crowd-sourced WiFi positioning system, focusing on how motion/quiescence can be used. In Section V we compare the different classifiers from Section III and quantify the end-to-end effects of using a good motion detector in a crowd-sourced WiFi positioning system. In Section VI we review prior related work, and in Section VII we describe future work and conclude.

II. Background

As prior work on motion detection has discussed [3], [4], continuous sampling of even relatively low-power sensors like the accelerometer is impractical for wide-spread use. People currently tolerate charging their sensor-rich devices once per day; we do not believe anything more frequent would be acceptable.

We assume that the person whose movement we are detecting is carrying a mobile device that contains a single triaxial accelerometer. At low energy cost, we are trying to provide a binary classification of whether he or she has "moved" from one place to another. Examples of "places" include a room or a section of a large room or a hallway. Because improving an indoor positioning system was our main motivation for this work, we examine motion at the granularity of rooms: should the positioning system believe that the user is in a new room (and act accordingly – e.g. update a map), or not (e.g. scan less). However, we believe that our detection of human-scale movement is applicable to satellite-, cell-, and outdoor WiFi-based positioning.

TABLE I
Definitions

Sampling Period	Length of time accelerometer is on (being sampled)
Off Period	Length of time accelerometer is off
Duty-cycle	Ratio between sampling period and off period (e.g. 5%)
Reading r	Single x, y, z output from the device's accelerometer
Sample $r_1 \ldots r_n \in s$	Time-ordered set of readings from a single sampling period
Cycle	Off period followed by a sampling period
Intra-reading period	Time between two readings (e.g. 10 ms)
Intra-sample period	Time between two samples (e.g. 5 seconds); *i.e.,* length of a cycle
Sliding window size	Number of samples that classifier uses
Metric ρ	Intermediary result from classifier
Threshold τ	Motion detection threshold: if $\rho > \tau$, then moving

We periodically sample the accelerometer: we work with only these readings as opposed to the entire accelerometer signal. As previous work has suggested and as we measure in Section V-A, duty-cycling the accelerometer can lead to an order-of-magnitude energy savings, and reduces the cost of measurement to noise for a typical user. While sampling for longer is more accurate (with diminishing returns), it consumes more energy. The second aspect of energy-efficiency is the classification algorithm itself; in Section V-C, we show that our proposed statistical algorithm is more energy-efficient than the other classifier we propose, while presenting equivalent accuracy and robustness, as discussed in Section V-B.

Figure 1 depicts our model of energy-efficient accelerometer-based motion classification, and Table I provides relevant definitions. The accelerometer is periodically cycled on and sampled. At the end of each sampling period, the collected samples, optionally together with any samples from previous cycles, are handed to a classifier. The classifier outputs "moving" or "quiescent" to any applications that are listening, including the localizer, which then act on this information.

III. Movement Detection Algorithms

Binary classification of a series of accelerometer samples into movement or quiescence is easy when testing and training on a single user, but hard to make robust to different patterns of user motion. Previous work has used variations on simple statistics, such as testing if the variance of a sample or a series of samples is above a threshold. These methods are energy-cheap but prone to false positives and false negatives. After reviewing these in Section III-A, we describe two new alternatives. The first is based on the Page-Hinkley statistical test [5], while the second uses the Discrete Fourier Transform (DFT) [6]. These are shown to significantly outperform previous methods, in particular, providing greater robustness to different kinds of movement from different users. In our evaluation, we show that, while the Page-Hinkley-based method is roughly equivalent to previous work in terms of energy consumption, the DFT-based method (perhaps unsurprisingly) consumes more energy.

A. Detection using Simple Statistics

Much prior work has studied accelerometer-based gait detection (e.g. [7]), and these algorithms could be applied to binary motion detection. However, the motion detection problem is simpler, and consequently should be less energy demanding.

The work by Shafer and Chang [8], Kim et al. [4], and Wang et al. [9], provide the basis for four motion detection variations using simple statistics. As Table I showed, all methods, including our own, produce an intermediate result ρ which is then tested against an empirically-determined threshold τ. "Motion" is emitted when $\rho > \tau$; "quiescence" when not.

Shafer and Chang's detector, which we call *M-SumVar*, is based on the sum of the unbiased variances of the readings from each axis:

$$\begin{aligned}\mathrm{Var}(m_1..m_N) &= \frac{\sum_{i=1}^{N} m_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} m_i\right)^2}{N-1} \\ \rho &= \mathrm{Var}(x_1..x_N) + \mathrm{Var}(y_1..y_N) + \mathrm{Var}(z_1..z_N)\end{aligned} \tag{1}$$

The detectors from Kim et al. and Wang et al. first compute the magnitude of each reading to tolerate random orientations of the device, merging the three axes. Using a method we call *M-Std*, Wang et al. then define their metric ρ to be the standard deviation of the series of readings that constitute the sample under test (Note that Kothari et al. [10] also use *M-Std*).

Instead of the standard deviation, Kim et al. use the variance; we call their method *M-Var*. To include more information from the past, Kim et al. extend their metric to include a time window of samples. Because this extension could be applied to any of the approaches, we treat the use of a time-series of samples as a separate, orthogonal option that could be applied to any algorithm.

Intuitively, a fourth metric could be derived: the use of the sum of the standard deviations given by the three axes, which we will refer to as *M-SumStd*.

To the best of our knowledge, no empirical comparison has justified using one metric or another, including in the papers themselves. These four metrics will be used as baseline for empirical comparison, together with the two new methods which we discuss next.

B. Page-Hinkley Change-detection Statistics

Given a series of observations $o_1, \ldots, o_\ell$, a frequently asked question is whether they come from the same distribution (null hypothesis) or not (*change* hypothesis). In this paper, we propose to tackle the accelerometer-based motion detection task as a change-detection problem: given a series of accelerometer readings (*e.g.,* all the readings from a given cycle), has the situation changed, *i.e.,* has it switched from the moving to the stationary state, or vice-versa?

A standard test for detecting change is the Page-Hinkley (PH) statistic [5], which has been used in contexts such as parameter adaptation in evolutionary algorithms [11] and online learning for dynamic optimization [12]. As in our previous statistics, at time t, PH outputs a metric ρ given a stream of observations $(o_1, \ldots o_\ell)$:

$$\begin{aligned}\bar{o}_\ell &= \tfrac{1}{\ell}\textstyle\sum_{i=1}^{\ell} o_i \\ m_t &= \textstyle\sum_{\ell=1}^{t}(o_\ell - \bar{o}_\ell) \\ M_t &= \arg\max_{\ell=1\ldots t}\{|m_\ell|\} \\ \rho &= M_t - |m_t|\end{aligned} \tag{2}$$

When the metric ρ is greater than some user-specified threshold τ, the PH test states that the *change* hypothesis holds. While PH is generally more robust than the simpler statistics (and continuous to be in the case of motion detection), it is composed of sets of simple computations and is efficient to compute.

PH can be directly applied to the binary motion/quiescence problem with a few refinements. Unlike typical uses of PH, we do not accumulate observations until a change is detected. Instead, we reset the internal variables with each sample. Second, to accentuate the difference between motion and quiescence, we take the square root of $M_t - |m_t|$ as ρ. Lastly, as in Kim et al. and Wang *et al.,* we use the magnitudes of the readings as our observations in order to tolerate random device orientations. When the device is stationary, the metric will be close to zero; when the device is moving, the metric is above a positive threshold. As in the simpler statistics, the only parameter that needs training is the threshold τ.

C. Detection using the Discrete Fourier Transform

The Fourier Transform transforms one function into another, where, for historical reasons, one form is called "frequency domain" and the other the "time domain" (see Lyons [13]). The Fourier Transform, and its discrete analogue, the Discrete Fourier Transform (DFT), are widely employed in signal processing and related fields to analyze the frequencies contained in a sampled signal, to solve partial differential equations, and to perform other operations such as convolutions or multiplying large integers. The transform has gained wide use because it can be computed efficiently in practice [6]. While Mostayed et al. [14] have used the DFT to study abnormal gaits, to the best of our knowledge, there is no previous work applying the Fourier Transform to binary motion classification, a problem for which it seems well suited.

For our classification problem, a DFT analysis can be applied directly to the accelerometer samples as follows. When the device is stationary, the most meaningful acceleration component is due to gravity, i.e. static acceleration. As gravity is a non-periodic field, its transform brings us a concentrated energy around the center frequency – close to 0 Hz. When the device is moving, the acceleration components are due to both gravity and the person's movement, i.e. dynamic acceleration. Because walking is a periodic motion, the transform of the acceleration magnitude samples provides us with other low frequency components with meaningful energy: the total energy into the DFT window is increased when the device is moving compared to when it is static. With this in mind, our DFT-based motion detection model evaluates if there is enough energy produced by movement during the DFT window and then classifies if the user's device is moving or not:

$$\begin{aligned} \bar{\mathbf{A}}_f = & \quad DFT\left[\bar{\mathbf{A}}_{xyz}\right] \\ e = & \quad \textstyle\sum_{i=1}^{n} \bar{\mathbf{A}}_f(i) \\ \rho = & \quad \sqrt{e} \end{aligned} \tag{3}$$

where each DFT window ($\bar{\mathbf{A}}_f$) is generated from the series of n accelerometer magnitudes $\bar{\mathbf{A}}_{xyz}$ within the same sampling period and e quantifies the amount of energy inside the DFT window. When the metric ρ is greater than some user-specified energy threshold τ, the DFT-based motion classification states that the user's device is moving, otherwise it is not.

IV. Using Motion in Organic Localization

While methods for *detecting* human-scale motion can be applied to the three main types of positioning systems – satellite, cell, and WiFi – techniques for *using* the result of that detection are typically more specific. Here we review a WiFi positioning system called Molé [2], and describe several techniques that use motion within it to improve energy-efficiency and accuracy. While we describe these techniques within the context of Molé, most generalize to other (non-crowd-sourced) commercial and academic WiFi positioning systems [15]–[17].

Molé is part of a new class of "organic" WiFi positioning systems that, unlike earlier work, rely on end-users as surveyors [18]–[21]. These systems crowd-source WiFi fingerprints, building up a database that is shared across users. While commercial surveys are often more appropriate for managed spaces, such as airports and malls, the authors of this class of work argue that relying on end-users for fingerprint creation and maintenance is perhaps the only route to extend localization across the "long tail," into people's homes, offices, and other less populated, more private spaces.

As in other organic WiFi positioning system, Molé tries to grow an accurate database of fingerprints. Each fingerprint is a link between radio beacon information that helps uniquely describe a space and a human-understandable label for that space. While some systems in this class use grid points, Molé aims for room-level granularity: its finest level of detail is a room. At any time, a user can add a fingerprint to the shared database, a process called "binding." A bind can either create a new space or add reinforcing fingerprint data to an existing one.

Molé estimates the position of the device using a variable-sized fingerprint: the last k WiFi scans are stored in a FIFO queue. Because a larger number of scans yield better accuracy [16], Molé seeks to collect long sets of scans. The problem with this approach is that, when walking from one room to another, some WiFi scans from the previous room might still be found in the queue, consequently perturbing the fingerprint used for the current position estimate. This can result in a delay while the old scans slowly exit the queue. Previous work using Molé showed that by truncating the scan queue in response to movement, this delay could be eliminated [2].

While using old scans can delay an update for one user, it can also have a more detrimental effect that affects many: bind pollution. It is important that the WiFi scans that make up the fingerprint actually come from the labeled space; otherwise these "polluting" binds will affect accuracy in and nearby that space for all users until several corrections have been made. Interestingly, the same mechanism that lowers update latency also stops bind pollution. When a user first enters a room, old scans in the queue have been dropped: only radio beacon information collected in the intended room becomes a part of the bound fingerprint. We show that by using a high quality motion detector, update latency is low *and* the shared database becomes populated reliably, allowing a crowd-sourced location system to grow rapidly.

V. Evaluation

We examine practical sampling rates, the efficacy of the motion detection algorithms using these rates, and the end-to-end impact of using motion detection within a crowd-sourced WiFi positioning system:

- We show the effect of duty cycling the accelerometer and find that any rate beyond $\approx 5\%$ has a significant effect on overall battery life (§ V-A);
- Using 20 subjects, we compare the accuracy and robustness of the proposed and the baseline motion detection algorithms when trained and tested on different people (§ V-B);
- We analyze the energy expenditure of the algorithms, leading us to select PH as best overall (§ V-C).
- We show the overall effect of using a high quality motion detector in a multi-floor crowd-sourcing scenario (§ V-D).

A. *Efficient Accelerometer Sampling*

We were concerned both with the energy consumption of the motion detection algorithms and with the sampling required to supply those algorithms with data. Here we examine what sampling rates are practical for day-to-day use. To the best of our knowledge, previous work has either assumed that sampling the accelerometer continuously is acceptable or supplied a sampling rate without examining its long term effect. While it may be possible to make other trade-offs with respect to sampling – for example, simply turning off detection for extended periods – the following experiments provide a conservative estimate on the energy required to sample the accelerometer at various rates.

To test the long-term effect of accelerometer sampling, we examined two dimensions of the problem: (a) the energy cost of different but proportional rates of sampling and (b) the cost of different duty cycles. If the answer to question (a) was that there was a high cost to activating and deactivating the accelerometer, then only sampling rates with long periods would be acceptable.

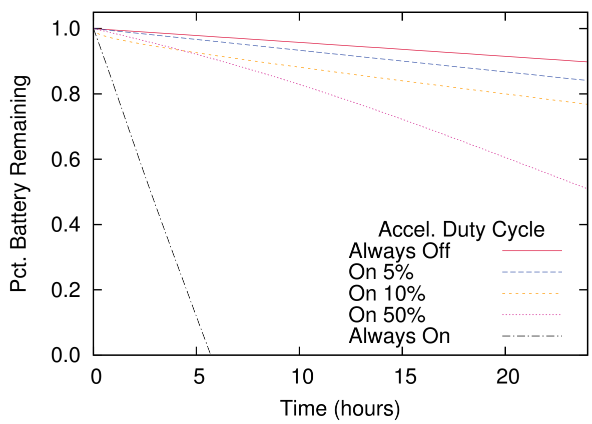


Fig. 2. Short, repeated samples have only a marginal effect on overall battery lifetime for a mobile device. In contrast, continuously sampling the accelerometer adversely affects device lifetimes.

We developed a simple program that turned on and sampled and then turned off the accelerometer repeatedly, at whatever durations were specified. These durations correspond to the accelerometer sampling "on" and "off" durations in Figure 1. We fully charged a set of identical mobile phones, Nokia N900s, and ran our test program with different parameters for at least 24 hours on each device. All user applications and communications were switched off and each device started with at least a 95% full battery, according to our energy-level detector. To account for non-linear power drain, the data are averages across three devices. The program sampled the fraction of battery remaining at one minute intervals.

To compare the effect of different but proportional rates, we set on/off durations to 0.25/1.25, 0.5/2.5, and 1.0/5.0 seconds (*i.e.,* all 20% of the time on). The results agree with Eberle and Perrucci [22], which used different devices. At sampling rates on the order of seconds and fractions of second ($\approx 5 - \frac{1}{5}$Hz), there is no noticeable difference between these rates. This means that the only consideration from above that we need to take care of is (b), the duty cycle ratio.

We used the same simple program to collect data on the long term battery consumption effects of different duty cycle ratios. The data from this experiment is depicted in Figure 2, which shows that, while never cycling on the accelerometer uses the least energy (as one would assume), only sampling at a rate of 5% only consumes marginally more energy. In this experiment specifically, we sampled for 250ms per every five seconds for the 5% rate. We selected five seconds because it appeared to be the upper limit on catching a short movement between offices or adjacent rooms, for example. We confirmed this experiments and the proportionality one through repeated trials.

An alternative to duty cycling is to explicitly set the rate of the accelerometer sampling to be slower. However, a back-of-the-envelope calculation shows that a low duty-cycling approach produces, in general, a per-sample cost lower than using a slower accelerometer sampling at full duty cycle (always on). According to a datasheet from a mobile handset accelerometer manufacturer [23], the consumption rate (in terms of drained electrical current from the battery) and the sampling rate are related in a non-linear way. For instance, at a rate of 4 Hz, the consumption rate is about 44 μA, while for 120 Hz, the corresponding rate is 294 μA. Assuming two approaches, (1) duty-cycle of 5% at 120 Hz and (2) duty-cycle of 100% at 4 Hz to be running over a five seconds cycle, we obtain an energy saving of around 73% using the first approach, while collecting a higher number of observations (30, versus 20 for the first approach).

B. Detection Accuracy and Robustness

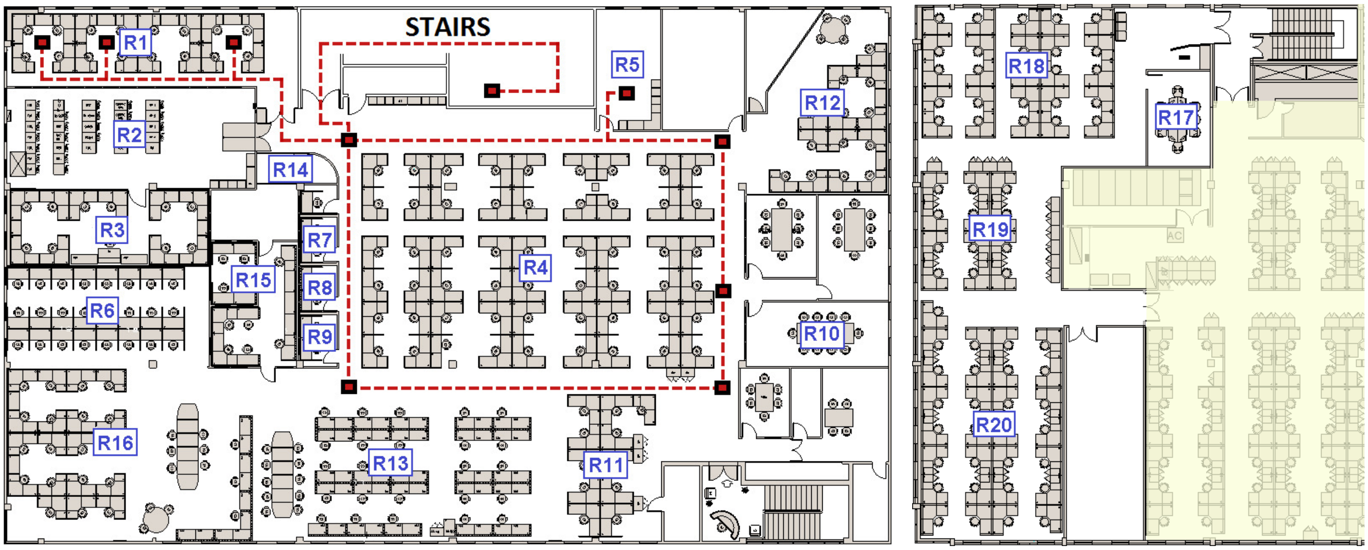
With these low duty cycles in mind, we wanted to study the detection accuracy and robustness of the six algorithms we described in Section III. In particular, given our larger goal of designing an energy-efficient motion detector that would work well for many people, we wanted to ensure that detectors trained on only a few people would be robust enough for testing on a much larger group of different people. To perform this evaluation, we collected a motion trace from twenty volunteers, found a quiescence/moving threshold τ for each method, and compared the algorithms across users.

1) Data Collection: Each of our twenty volunteers was given a mobile device (Nokia N900) which continuously logged accelerometer data. They were asked to follow a pre-specified "track," represented by the red dashed line in Figure 3, and to stop for approximately five seconds at each checkpoint, represented by the black squares. We selected the checkpoints to create many different durations of motion because we wanted to ensure it was correctly detected over longer or shorter periods. Each user took approximately five minutes to walk the track. It comprised many different situations one would face in an office-like environment, such as walking to a neighboring office, walking a few offices away or around the entire lab, and going up and down stairs.

While walking though the track, the volunteers were asked to carry and manipulate the device as naturally and realistically as possible. For instance, some exclusively held the device in their hands, some simply put the device into their pocket, while others used it for calls and games. A researcher followed each volunteer with a second time-synchronized device to log each time the volunteer switched between quiescent and moving states. The acceleration data from the volunteer's device and the motion-state labels from the second device were merged offline for evaluation.

Twenty volunteers had their motion logged in this way. Figure 4 illustrates the accelerometer readings from one volunteer. The variation in the readings during the different motion periods portray how differently a single person could handle the phone and walk in different ways during the course of the experiment.

We had one particularly interesting choice on how to handle human imprecision in the data collection. As the ground-truth timestamp is registered with a precision of milliseconds, there is always a slight positive or negative error from when the volunteer switched from one state to another and when the researcher logged his or her state as switching. Though we considered alleviating these labeling errors by cutting off a few



(a) Lab A Third Floor Map

(b) Lab A Second Floor Map

Fig. 3. Plan of the rooms of Lab A. The red dashed line shows the track used for the collection of motion data used for the empirical comparison of the motion detection algorithms, with the black squares representing the checkpoints where the volunteers should stop for a few seconds. The labeled rooms (e.g., R1) were used in the experiments with the indoor positioning system, which will be discussed later in the paper.

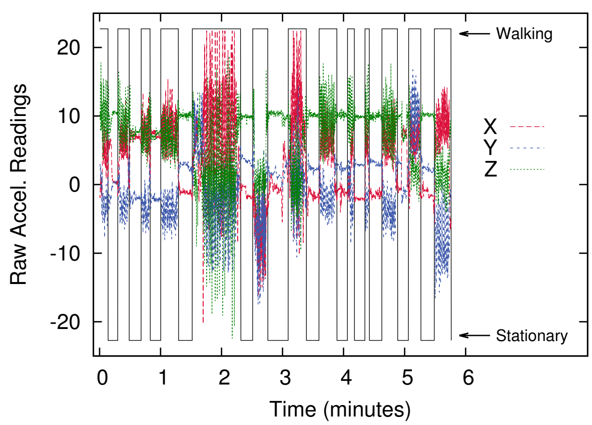


Fig. 4. Ground-truth reference data (black line) and readings from tri-axial accelerometer (red for X, green for Y and blue for Z axes) for a given volunteer following the pre-specified track shown in Fig. 3. Each algorithm attempts to classify the user as walking or stationary given a limited sampling of raw accelerometer inputs.

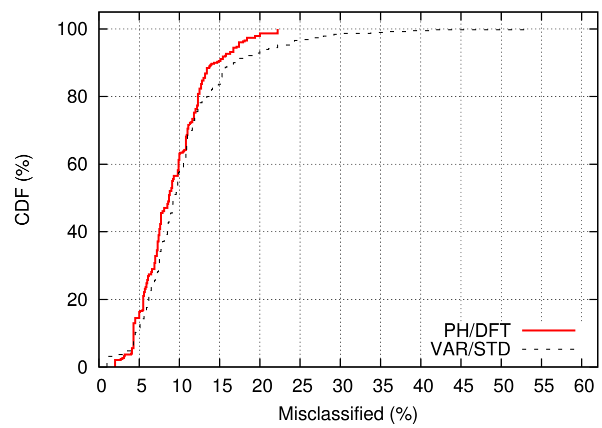


Fig. 5. Cumulative Distribution Function (CDF) of the accuracy of the motion detection algorithms when training (*i.e.,* defining the threshold) with the data from one volunteer and testing on the remaining nineteen. The data show that PH and DFT are more robust to differences in people's movement styles, resulting in much lower misclassification rates for the hardest 20% of cases. Because each group of PH/DFT and VAR/STD/M-Var/M-Std had very similar performance, we illustrate each set as a single line for clarity.

milliseconds before and after each state transition, we decided not to do so as the same data was used for all the methods. In practice, this means that the real detection accuracy of all the methods should be higher than what will be presented next.

2) Threshold Definition: Because we do not anticipate any per-user active learning refinement of the walk detection thresholds, we simply used an offline brute force method to find the optimal threshold for each user/algorithm combination. We first ran the given algorithm on each log, producing a ρ for each sample. This set of ρ values defined the possible range for optimal thresholds. We discretized the range for the metric into n possible thresholds ($n = 500$). These values were tested one by one, and the threshold value achieving the highest accuracy was tracked. In case there was a range of values achieving exactly the same accuracy level, the mean value within this range was used.

3) Robustness: The motion detection methods need to be robust enough in order to work well for many different users under very diverse moving situations, without needing to have its threshold "tuned" for every new user or situation. Motivated by this assumption, we trained the algorithm's threshold over a single log file and then tested it on the 19 remaining log files. This procedure was done iteratively for each logfile, generating a total of 380 accuracy measures (20×(1 training-file × 19 test-files)). The result from each test is the fraction of samples correctly classified when compared to the ground truth for that same time period.

TABLE II
RELATIVE ENERGY COST OF MOTION DETECTION ALGORITHMS

Method	Avg. Time (hours)	Pct. Reduction in Battery Lifetime
M-Std	19.94 ± 1.35	0%
M-Var	19.31 ± 1.30	3.17%
PH	19.26 ± 0.49	3.41%
DFT	16.52 ± 2.72	17.18%

Figure 5 shows the cumulative distribution for these 380 tests. The data show that PH and DFT have similar level of accuracy, outperforming the baseline methods in at least 20% of the cases (*i.e.,* 76 out of the 380 cases) and presenting similar performance otherwise. In addition, for PH, the worst accuracy obtained over all the cases was of around 78%, and for DFT around 70%, while the baseline methods obtained accuracies equivalent to a random choice.

C. Energy Efficiency

Both PH and DFT-based proposed algorithms were shown to outperform the baseline methods in terms of both accuracy and robustness. This is only meaningful, however, if they are found to be energy efficient, *i.e.,* if their use does not hinder the battery lifetime of the mobile device they are running on.

To test this, we left each of the motion detection algorithms running independently on four Nokia N900 devices, which initially had their battery fully charged. The accelerometer readings were processed at the pre-specified duty-cycle ratio, with the accelerometer turned on for 0.25 seconds every 5 seconds (collecting about 25 reading for each sample). Both the time and the current battery capacity were logged every time there was a change in the latter. *M-SumStd* and *M-SumVar* were omitted because we assumed that they would have approximately the same energy consumption profiles as their no-summing counterparts, *M-Std* and *M-Var* respectively. We collected data from each device for 24 hours. In Table II, we show how many hours each of the algorithms took on average to drain 15% of battery while processing accelerometer readings at the pre-defined sampling rate (0.25 secs sampling every 5 secs) with each of the motion detection methods. The Percentage Reduction column shows the loss in terms of energy efficiency when compared to the method that consumes the less energy in average, *M-Std*.

The data show that PH, although significantly outperforming *M-Std* and *M-Var* in terms of motion detection robustness, is highly comparable in terms of energy consumption. Indeed, these three methods present a very similar energy usage, draining 15% of battery after around 19 hours in average. The DFT-based method is much energy-hungrier, as expected, given the well-known high computational cost of computing the corresponding Fourier transform: in average, it takes around 16.5 hours to use 15% of battery, with a standard deviation as high as 2.7 hours. We conclude from these two experiments that PH is the preferred motion detection algorithm out of the six under consideration.

D. Using Motion Detection in Crowd-sourced Indoor Localization

We wanted to evaluate how motion detection could impact the end-to-end room-level accuracy of a crowd-sourced WiFi positioning system. As crowd-sourcing of location data comes into wider use, it is important for end-users to provide high quality data and for them to find the system useful enough for them to want to contribute further.

We described how Molé truncates its time series of scans in response to movement in Section IV. Without motion detection, we conjectured that users would walk into a space, observe an incorrect estimate and make a bind. Unfortunately, the scans that would then make up that bind would erroneously be from where the user had been, not from where he or she was now, a problem we call “bind pollution.”

Here we show how this use of motion detection can have a large effect on overall user-visible performance through a crowd-sourcing experiment. We recruited three volunteers to use Molé in a two story lab with twenty rooms. The volunteers then surveyed the two floors for fourty minutes, splitting their time about 75/25 across the third and second floors, respectively (see Figure 3). Two N900 tablets attached to a clipboard were given to each volunteer. One tablet was running Molé with motion detection disabled and the second ran a Page-Hinkley-based motion detector as described in Section III-B. The three tablets without motion detection pointed to one shared fingerprint database and the three with it pointed to a second, completely separate, one. In this way, we could ensure that each pair of tablets were subjected to nearly identical orientations, displacements, and radio beacon information, and that they were in the same rooms at the same time for the same duration. Before giving the tablets to the volunteers, we performed one bind in Room #1 on the two empty databases. In effect, this initialized the namespace labels (e.g. building, floor, room) on all of the tablets so the volunteers would only need to edit the floor number and the room label. They were given instructions to walk from room to room, editing the estimate when it was wrong or confirming when it was correct. They were told to wait up to 30 seconds (three scans) before making a confirmation of its correctness or creating an edit when it was wrong.

Figure 6 shows how coverage and spot-on hit rates changed for the two approaches during the experiment. We calculated hit rates as a ten minute moving average of spot-on accuracy (i.e. when the volunteer confirmed the estimate); this included the first bind for each room which is, by definition, an incorrect estimate. By using the Page-Hinkley-based motion detection for cleaning the scan queue, Molé was able to converge to 100% accuracy once the whole lab was covered. When no motion detection was used, the scan queue became polluted, so the estimates and contributions were sullied by scans from previously visited rooms. Due to these factors, the spot-on hit rate for the tablets without motion detection was always below 40%.

Beyond the larger quantitative result, the experiment contains two other interesting aspects. First, for both systems, the spot-on hit rate decreased from 15 minutes to 18 minutes.

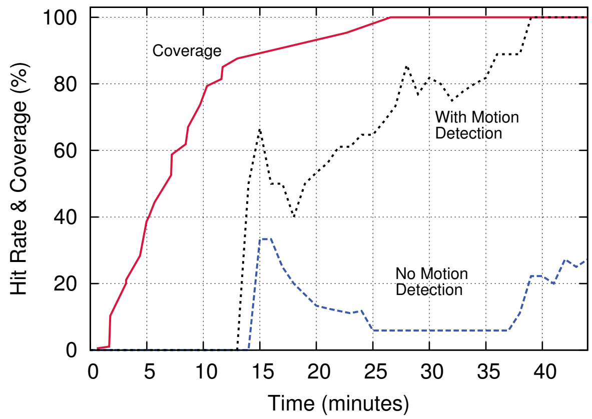


Fig. 6. Three volunteers surveyed a two story lab with twenty rooms. By using the Page-Hinkley-based motion detection for cleaning the scan queue, Molé was able to converge to 100% accuracy once the whole lab was covered.

During this time, the volunteers visited rooms #7, #8 and #9. These rooms are small ($\approx 5.6m^2$) and very close to each other (see Figure 3), with thin plaster walls and glass doors, a situation where WiFi positioning typically has difficulty. Second, all of our volunteers stated that the motion detection was qualitatively superior to their previous experience with Molé, which had used one of the simple statistical detectors. Previously they complained that they often had to vigorously shake the device to have it detect motion, while now they said they did not notice any false positives or negatives.

E. Discussion

The PH-based algorithm is a significant improvement over existing motion detection algorithms in terms of accuracy and robustness (§ V-B), while consuming a similar amount of energy (§ V-C). The DFT-based algorithm was found to be equally accurate and robust, but the high computational cost associated with the computation of the Fourier transform makes it less preferable.

The power of the PH-based motion detection algorithm makes many other improvements in indoor positioning systems possible. As with previous work on using GPS only when necessary (e.g. Jigsaw [24]), motion-based strategies can make Molé and others like it more energy-efficient. For example, we observed that the instance using the detector was correctly recognizing each room after 1-3 scans once the person stopped moving. During the experiment, there were five quiescent periods with approximately eleven WiFi scans each. By limiting the number of scans per quiescent period to three, 40 out of 55 scans would not have been done, *i.e.,* a reduction on the order of 72%, without compromising the accuracy level. Assuming that each WiFi scan costs around 800 mW for a similar mobile device [22], and that this part of the experiment took 15 minutes in total, the use of this simple motion-based strategy would have reduced energy consumption from 200 mWh to 56 mWh.

Another very useful application that we leave as future work is what we refer to as *automatic binding*. By using the (often long) periods that people stay in office-like environments as a trickle of scans that can be added to an existing fingerprint, the shared database can be improved and maintained over time. This could be done by periodically sending "implicit" binds to the server, once a given level of confidence about the current position estimate (or a pre-defined number of scans) is achieved, without any user intervention. Such a strategy would still be more energy-efficient than the current one, as the scans could be done at a much slower rate (*e.g.,* once per minute).

Interestingly, previous work (e.g. Kim et al. [4]) suggested using sliding windows of samples, in order to introduce more readings into the motion detection classifier. Depending on how the window is used, this would mean, for example, that motion is classified based on a moving average of the last n sampling periods. After some preliminary experiments, however, we found that the use of sliding windows simply delayed the onset of motion detection and quiescence without improving accuracy, consequently degrading its overall performance. For this reason, we would discourage its use in this context.

VI. Related Work

We have focused on *classifying* the movement or quiescence of users of crowd-sourced WiFi positioning systems and on *using* this aspect of their state to change system behavior. Work related to this topic can be divided into three categories: (1) different motion detection techniques, which we described in Section III-A, (2) using motion in indoor positioning, and (3) minimizing energy use in positioning systems more generally.

Several pieces of prior work have explored how to augment indoor positioning systems with accelerometer-based motion detectors. You et al. [25] and Shafer and Chang [8] vary the scan rate in response to motion, trading off accuracy and energy use. Both choose to increase scanning rates in response to movement: You et al. do it in proportion to velocity, while Shafer and Chang take a more binary approach, rapidly collecting many scans when movement is detected. Both of these approaches are problematic when considered in the larger context of a user transitioning from indoor positioning, where room grain granularity (or finer) is desired, to walking outdoors or taking other forms of transportation. Both approaches would scan very rapidly – draining the battery – when the user was in a car, for example. Instead we suggest that, after a given velocity, the device should switch to a different positioning method (e.g. cell- or GPS-based), depending on the required accuracy. Kim et al. [4] also use motion detection to save energy: when quiescence is detected and a threshold number of scans has been collected, they stop scanning entirely (We referred to their motion detection algorithm as *M-Var*). Barring collecting background scans for automatic binding (§ V-E), we concur with this strategy. However, Kim et al. only duty cycle the accelerometer at 50%, which severely shortens battery life (§ V-A), and their motion detection algorithm performs poorly compared to Page-Hinkley (§ V-B).

In addition to reducing energy usage, prior work has directly connected users' movements to the location estimate and fingerprint database. You et al. [25] and Gaedeke et al. [26] update the current estimate with sensor-based dead reckoning. This allows the good estimate formed when the user is stationary to be the basis for a new mobile one. This technique is complementary to our motion-based scan list truncation and growth (§ IV): a quiescent device would build up a good estimate based on many scans, which could then be assisted with dead reckoning when the user moved. Bolliger et al. [21] collect scans from long quiescent periods and allow end-users to bind a name to them retroactively ("Where were you from 9-10am?"). In Section V-E, we discussed how estimate confidence, together with quiescence, could be used to automatically reinforce binds that had been made previously.

While we focus on detecting motion using accelerometers in the context of WiFi-based indoor positioning systems, Muthukrishnan et al. [27] and Krumm and Horvitz [28] have instead used variation in the WiFi signals themselves to detect motion. Intuitively this seems like a reasonable approach: since the RSSI values are already available, could they not be used for this purpose as well? Muthukrishnan et al., whose work builds on Krumm and Horvitz, do show that *at high sampling rates* WiFi can be used to detect motion. In Muthukrishnan et al., the authors scan every four seconds and base their detection algorithm both on variation in RSSI values from a given access point and on variation in the number of access points sensed: more variation suggests motion. Unfortunately these high scans rates are incompatible with the goal of putting the energy-consuming location system into a quiescent state; without frequent WiFi scans, it seems challenging to use them to detect motion. In defense of this approach, this work was done before most handheld devices contained accelerometers, making it more applicable for its time.

Another main body of work has examined minimizing energy use in positioning systems in general, typically trading off applications' requirements for the precision of GPS against battery life. Researchers have taken several variations on this theme of balancing between the expenditure and precision of positioning sensors (e.g. GPS vs. GSM) and motion detection ones (e.g. accelerometer [29] [24], microphone [9], cameras [30]). For example, EnTracked estimates the speed of the user with an accelerometer and duty cycles the GPS accordingly [31]. Because it monitors the accelerometer continuously and uses a motion detection scheme similar to *M-Var*, EnTracked would benefit from several of the techniques proposed in this paper. RAPS [3] improves on EnTracked in several ways, including duty-cycling the accelerometer and measuring the user's velocity to activate the GPS only if the estimation confidence drops below a desired threshold.

VII. CONCLUSION

This paper examined energy-efficient binary human-scale quiescence/movement classification and its use in positioning systems. We compared four classifiers based on simple statistics to two new ones in terms of energy-efficiency and robustness. We first explored practical rates for getting accelerometer input into these classifiers, finding that only a low duty cycle was acceptable. Next, we considered the accuracy and robustness of the six classifiers using data from twenty subjects. We found that the two classifiers we proposed, one based on Page-Hinkley statistics and the other on the Fourier transform, were the most robust of the six when trained and tested on different users. We also compared the classifiers in terms of energy-efficiency and found that the five statistical ones were comparable, but that the Fourier transform one was more energy intensive. In our final experiment, we showed the macro-level effects of using motion detection in a WiFi positioning system, finding that a crowd-sourced system without motion detection (and, therefore, with bind pollution) had far lower accuracy than one with a detector. We also discussed how quiescence, together with estimate confidence, could be used to passively add reinforcing fingerprints to the crowd-sourced database, and how sliding windows of samples, while used in previous work, actually delay motion detection without increasing accuracy.

REFERENCES

[1] A. Thiagarajan, L. S. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, Low-Energy Trajectory Mapping for Mobile Devices," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, 2011.

[2] J. Ledlie, J. geun Park, D. Curtis, A. Cavalcante, L. Camara, A. Costa, and R. Vieira, "Molé: a Large-Scale, User-Generated Positioning Engine," in *International Conference on Indoor Positioning and indoor navigation (IPIN)*, Guimarães, Portugal, 2011.

[3] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," in *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, San Francisco, CA, Jun. 2010.

[4] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava, "SensLoc: Sensing Everyday Places and Paths using Less Energy," in *Embedded Networked Sensor Systems (SenSys)*, Zurich, Switzerland, 2010.

[5] D. Hinkley, "Inference about the change point from cumulative sum-tests," *Biometrika*, vol. 58, no. 3, pp. 509–523, 1970.

[6] J. W. Cooley, P. A. W. Lewis, and P. D. Welch, "The Fast Fourier Transform and Its Applications," *IEEE Trans. on Education*, vol. 12, no. 1, pp. 27–34, 1969.

[7] A. Godfrey, R. Conway, D. Meagher, and G. OLaighin, "Direct measurement of human movement by accelerometry," *Medical engineering & physics*, vol. 30, no. 10, pp. 1364–1386, 2008.

[8] I. Shafer and M. L. Chang, "Movement Detection for Power-Efficient Smartphone WLAN Localization," in *Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, 2010.

[9] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition," in *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Kraków, Poland, 2009.

[10] N. Kothari, B. Kannan, and M. Dias, "Robust Indoor Localization on a Commercial Smart-Phone," CMU Robotics Institute, Tech. Rep. CMU-RI-TR-11-27, 2011.

[11] A. Fialho, L. Da Costa, M. Schoenauer, and M. Sebag, "Analyzing Bandit-based Adaptive Operator Selection Mechanisms," *Annals of Mathematics and Artificial Intelligence*, vol. 60, pp. 25–64, 2010.

[12] C. Hartland, N. Baskiotis, S. Gelly, M. Sebag, and O. Teytaud, "Change point detection and meta-bandits for online learning in dynamic environments," in *Conférence francophone sur l'apprentissage automatique (CAp07)*. AFIA, 2007, pp. 237–250.

[13] R. G. Lyons, *Understanding Digital Signal Processing (2nd Edition)*. Prentice Hall, 2004.

[14] A. Mostayed, M. Mynuddin, G. Mazumder, S. Kim, and S. J. Park, "Abnormal gait detection using discrete fourier transform," in *Multimedia and Ubiquitous Engineering (MUE)*, Busan, Korea, Apr. 2008.

[15] "Ekahau positioning engine," ekahau.com.

[16] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical Robust Localization over Large-Scale 802.11 Wireless Networks," in *Proc. of the Tenth ACM International Conference on Mobile Computing and Networking (MOBICOM)*, Philadelphia, PA, Sep. 2004.

[17] M. Youssef and A. K. Agrawala, "The Horus WLAN Location Determination System," in *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Seattle, WA, Jun. 2005.

[18] J. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie, "Growing an Organic Indoor Location System," in *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, San Francisco, CA, Jun. 2010.

[19] E. S. Bhasker, S. W. Brown, and W. G. Griswold, "Employing user feedback for fast, accurate, low-maintenance geolocationing," in *International Conference on Pervasive Computing and Communications (PerCom)*, Mar. 2004.

[20] A. Barry, B. Fischer, and M. Chang, "A long-duration study of user-trained 802.11 localization," in *International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT)*, Orlando, FL, Sep. 2009.

[21] P. Bolliger, K. Partridge, M. Chu, and M. Langheinrich, "Improving Location Fingerprinting through Motion Detection and Asynchronous Interval Labeling," in *Symposium on Location- and Context Awareness (LoCA)*, Tokyo Japan, May 2009.

[22] J. Eberle and G. Perrucci, "Energy measurements campaign for positioning methods on state-of-the-art smartphones," in *Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, Jan. 2011.

[23] M. Kelsey, L. Salhuana, M. Forget, and N. Drew, "MMA7660FC Accelerometer for Mobile Handsets," Freescale Semiconductor, Tech. Rep., 2010.

[24] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The Jigsaw Continuous Sensing Engine for Mobile Phone Applications," in *Embedded Networked Sensor Systems (SenSys)*, Zurich, Switzerland, 2010.

[25] C.-W. You, P. Huang, H.-h. 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*, vol. 6, pp. 1221–1237, 2008.

[26] T. Gädeke, J. Schmid, W. Stork, and K. D. Müller-Glaser, "Pedestrian dead reckoning for person localization in a wireless sensor network," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Guimarães, Portugal, 2011.

[27] K. Muthukrishnan, M. E. Lijding, N. Meratnia, and P. J. M. Havinga, "Sensing motion using spectral and spatial analysis of wlan rssi," in *Second European Conference on Smart Sensing and Context (EuroSSC)*, Kendal, England, Oct. 2007.

[28] J. Krumm and E. Horvitz, "Locadio: Inferring motion and location from wi-fi signal strengths," in *International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004)*, Cambridge, MA, Aug. 2004.

[29] F. Ben Abdesslem, A. Phillips, and T. Henderson, "Less is more: energy-efficient mobile sensing with SenseLess," in *First ACM SIGCOMM Workshop on Networking, Systems, Applications on Mobile Handhelds (MobiHeld)*, Barcelona, Spain, 2009.

[30] F. Aubeck, C. Isert, and D. Gusenbauer, "Camera based step detection on mobile phones," in *International Conference on Indoor Positioning and indoor navigation (IPIN)*, Guimarães, Portugal, 2011.

[31] M. B. Kjaergaard, J. Langdal, T. Godsk, and T. Toftkjaer, "EnTracked: energy-efficient robust position tracking for mobile devices," in *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Kraków, Poland, 2009, pp. 221–234.