

Walk Detection and Step Counting on Unconstrained Smartphones

Agata Brajdic

Cambridge University Computer Laboratory
ab818@cam.ac.uk

Robert Harle

Cambridge University Computer Laboratory
rkh23@cam.ac.uk

ABSTRACT

Smartphone pedometry offers the possibility of ubiquitous health monitoring, context awareness and indoor location tracking through Pedestrian Dead Reckoning (PDR) systems. However, there is currently no detailed understanding of how well pedometry works when applied to smartphones in typical, unconstrained use.

This paper evaluates common walk detection (WD) and step counting (SC) algorithms applied to smartphone sensor data. Using a large dataset (27 people, 130 walks, 6 smartphone placements) optimal algorithm parameters are provided and applied to the data. The results favour the use of standard deviation thresholding (WD) and windowed peak detection (SC) with error rates of less than 3%. Of the six different placements, only the back trouser pocket is found to degrade the step counting performance significantly, resulting in undercounting for many algorithms.

Author Keywords

Inertial sensing; Dead reckoning; Context-aware computing; Pedestrian Model; Smartphones

ACM Classification Keywords

H.5.m. Information Systems: Information Interfaces and Presentation—*Miscellaneous*; I.6.4 Computing Methodologies: Simulation and Modeling—*Model Validation and Analysis*

INTRODUCTION

With the increasing ubiquity of smartphones, users are now carrying around a plethora of sensors with them wherever they go. These platforms are enabling technologies for ubiquitous computing, facilitating continuous updates of a user's context. Smartphones are already touting built-in pedometers for ubiquitous activity monitoring and there is a growing interest in how to use the same inertial sensors to build Pedestrian Dead-Reckoning (PDR) systems for indoor location tracking.

Unfortunately there is a significant computational load associated with continuous sensor data processing, particularly for PDR. Modern smartphone CPUs can perform the necessary processing in real time, but only at the cost of high power

consumption and hence reduced battery life. We must therefore apply PDR techniques only when the user is moving, for which a continuous walk detection algorithm is required. Once a walk is detected, PDR algorithms require a robust estimate of the number of steps taken, for which a more sophisticated algorithm may be deployed.

Walk detection (WD) and *step counting* (SC) tasks have received much prior attention, but were often studied under laboratory conditions and tested on a relatively small number of subjects. Moreover, there is currently no detailed understanding of how well WD and SC algorithms work when applied to smartphones in typical, unconstrained use.

In this paper we seek efficient and robust WD and SC algorithms for unconstrained smartphones. We survey various WD and SC algorithms from the literature and evaluate them under different smartphone placements. We define a metric that allows uniform comparison of the algorithms, from which we draw recommendations for the design of future smartphone-based PDR systems. Our paper distinguishes itself from existing work in the following ways:

- we use commodity smartphones to gather the data;
- we do not fix the smartphone to the user, allowing it to be used naturally;
- we analyse a large set of 130 sensor traces from 27 distinct users, with a range of walking speeds; and
- we provide a fair, quantitative comparison of the standard algorithms in one place.

The annotated accelerometer datasets used in this study are freely available to all researchers and can be found at:

<http://www.cl.cam.ac.uk/~ab818/ubicom2013.html>

BACKGROUND AND RELATED WORK

There is a wealth of studies on the use of body-worn inertial sensors for detecting and characterising walk-related activities. This work has appeared in various domains including medical [3], context-aware computing [15], biometric identification [1] and PDR [17]—each with different goals and methodologies.

In the medical domain, the main interests are detection of specific gait events (e.g., falls) [46] and discrimination of walking patterns [24]. Detailed gait characterisation is sought and consequently sensors are usually firmly attached at pre-defined locations. The work in the context-aware computing domain is usually concerned with classification of a wider range of activities [4, 27, 49] from sensors attached at convenient positions (e.g., the waist or wrist). These approaches typically consider walk detection; step counting is rarely of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
UbiComp'13, September 8–12, 2013, Zurich, Switzerland.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-1770-2/13/09...\$15.00.
<http://dx.doi.org/10.1145/2493432.2493449>

interest. By contrast, gait-based biometrics focuses on more detailed gait analysis and often incorporates some sort of gait cycle detection [1, 14]. On smartphones, these topics are now receiving much attention for inertial-based activity recognition [30, 42] and biometric identification [29, 52, 57].

In the PDR field, walk detection is essential to avoid applying expensive analysis algorithms during periods of non-motion, and step counting is necessary for many algorithms that assume a step length and estimate a movement [6, 16, 58]. Initial PDR systems concentrated on analysing foot-mounted sensors, for which walk detection and step counting are relatively straightforward. However, for smartphone-based pedestrian localisation [33, 47], walk detection and step counting are much less well understood.

Algorithms

The simplest WD and SC techniques threshold some property of the accelerometer signal [16, 25, 48]. This is particularly effective when sensing at the foot, where heel strikes result in large, short-lived accelerations. As with any thresholding technique, difficulties arise in selecting the optimal threshold value, which can vary between users, surfaces and shoes [14]. For an unconstrained smartphone, the freedom of movement might additionally affect the property of the accelerometer signal and mistakenly trigger the threshold.

Many algorithms search instead for the periods inherent in the cyclic nature of walking [1, 18, 24]. Technically the cycle is across a *stride* (two steps), although a sensor sited along the body's mid-axis can exhibit a one-step period if the user's gait is symmetric. Typical stride frequencies are around 1–2 Hz, a range that few activities other than walking exhibit.

Peak detection [25, 48] or zero-crossing counting [6, 16] on low-pass filtered accelerometer signals can be used to find specific steps. Methods such as *Pan-Tompkins* and *dual-axis* [59] apply this independently to globally vertical acceleration, which provides better performance [40], but depends on being able to isolate the orthogonal accelerations in the global frame. This is difficult to do on a smartphone that is not firmly attached to the body. Instead, the signal magnitude is often substituted.

The short-term Fourier transform (STFT) [55] can be applied to evaluate the frequency content of successive windows of data [5]. However, this windowing introduces resolution issues [32]. Wavelet transforms (CWT/DWT) [36, 54], which repeatedly correlate a 'mother' wavelet with the signal, temporally compressing or dilating as appropriate, are more suitable [5, 43, 57] as they can capture sudden changes in the acceleration, but are known for being computationally more expensive [12].

A less costly option is to use auto-correlation to detect the period directly in the time domain. The cost of this approach can be further reduced by evaluating the auto-correlation only for a subset of time lags corresponding to the frequencies of interest [35, 47].

A disadvantage of these algorithms is that they will be triggered by any motion with a similar periodicity to walking.

An alternative is to attempt to *recognise* each step within the data. A stride template can be formed offline and cross-correlated with the online signal [1, 40, 59]. Better results can be achieved by allowing a non-linear mapping between two signals computed by the dynamic time warping (DTW) [50], but at the price of a considerable computational overhead. DTW-based matching can be further improved by the phase registration technique [35], at even more cost.

Feature Classification

Many previous studies leverage machine learning techniques to classify activities based on features extracted from the accelerometer signal. However, neither a single feature nor a single learning technique has yet been shown to perform the best [10, 20, 46].

Commonly used features include mean [4, 27, 29, 45, 49, 52, 57], variance or standard deviation [29, 31, 42, 45, 49, 52, 57], energy [4, 49, 52, 57], entropy [4, 42], correlation between axes [4, 45, 49], and FFT coefficients [26, 27]. Features obtained by dimensionality reduction (e.g., principal component analysis) perform below average [39]. See [12] for a detailed overview.

Most studies use these features for training classifiers such as neural networks [29, 30, 39, 45], support vector machines (SVMs) [19, 41, 51] or Gaussian mixture models (GMMs) [2, 21, 57], or give them as an input to non-parametric classification methods such as k-nearest neighbour [45, 52]. These techniques can be effective (particularly the Bayesian ones), but are supervised, so offline training is required—often for each individual and each carrying position.

Unsupervised techniques are used much less (notably, self-organising maps [23] and k-means clustering [8, 9]) with the exception of hidden Markov models (HMMs) [37, 38, 41, 44], which can be used both for classification and in an unsupervised fashion. HMMs are particularly interesting as they are trained on a *sequence* of features, which tends to improve the accuracy as temporal correlations get captured in the model.

Evaluation Methodologies

Comparing algorithms across studies is difficult for two reasons: firstly different evaluations use different sensor attachments (foot, hip, hand, pocket, etc); and secondly the evaluations rarely feature a representative sample of phone users. Large studies are available in the medical domain (e.g. [40]), but the corresponding data are usually collected under conditions that do not generalise to an unconstrained smartphone. Conversely, the evaluation in [47] puts few constraints on the users (who are free to use the smartphone as they wish), but only evaluates using a small number of users. They report 100% success rates that we have been unable to reproduce.

Difficulties also arise due to differences in the perceived applications. A gait analysis system, for example, rarely requires a walk detection algorithm since there is a medical practitioner to start and stop data collection manually. Similarly, many trials in the PDR field involve asking a small number of users to walk 'naturally' for a few minutes in an area clear of obstructions. Estimates of step or stride counts

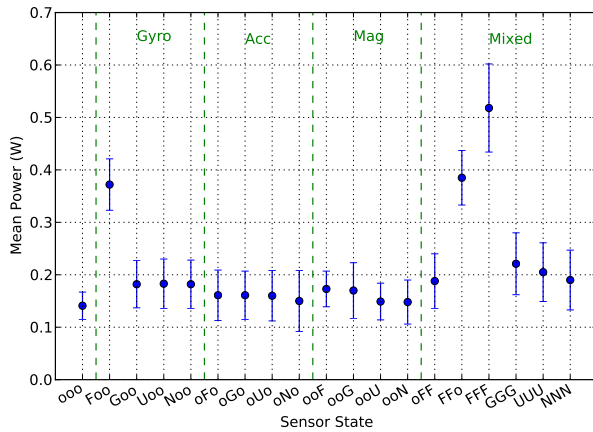


Figure 1. Experimental power consumption of a Nexus S smartphone. The sensor state is represented by three letters for the gyroscope, accelerometer and magnetometer. The letters signify the Android sensor power states: o (off), N (normal), G (Game), U (UI), F (Fastest).

	Walk detection	Step counting
Time domain	MAGN_TH	
	ENER_TH	
	STD_TH	
		WPD
Freq. domain		MCC
	NASC+STD_TH	NASC
		DTW
Feat. clust.	STFT	STFT
	CWT	CWT
	DWT	DWT
	HMM	HMM
	KMC	KMC

Table 1. The algorithms tested in this work. We use the above abbreviations when referring to the algorithms in further text.

can then be made by dividing the duration of the walk by the observed step period. However, such estimates may be coarse since an average step period is used (e.g. [47]).

REPURPOSING SMARTPHONES

Repurposing smartphones for ubiquitous sensing is challenging due to their battery constraints and multi-purpose nature. We observed that each embedded sensor has a similar power draw if used individually, but using combinations of sensors costs significantly more (see Figure 1). Thus, we use solely the accelerometer (applicable even to early smartphones).

Smartphones are not dedicated sensing devices and they suffer from dropped samples and significant jitter in signal timestamps. Moreover, their nonrigid attachment (they may be carried in front pockets; back pockets; shirt pockets; hands; bags and more) and freedom of motion violate many of the assumptions made in previous step detection work.

ALGORITHMS TESTED

We tested variants of ten algorithms for WD and SC tasks. The WD task was to extract the period(s) during which walking was occurring; the SC task was to count the number of

steps within a well-defined period of walking. We took the algorithms operating in the time domain (Thresholding, Windowed Peak Detection, Mean Crossing Count, Normalised Autocorrelation SC and Dynamic Time Warping) and in the frequency domain (Short Term Fourier Transform and Continuous/Discrete Wavelet Transforms) directly from the literature. For feature clustering, we adapted two representative algorithms (Hidden Markov Models and K-Means Clustering) for our specific WD and SC tasks.

Time Domain

Thresholding

Walk Detection: We considered threshold-based walk detection (i.e., treating all readings as walk so long as the value of the variable of interest lies above a predefined threshold) when applied to acceleration magnitude (MAGN_TH), energy of the acceleration signal (ENER_TH, calculated over a window of size $ener_{win}$) and its standard deviation (STD_TH, calculated over a window of size std_{win}).

Step Counting: N/A.

Windowed Peak Detect (WPD)

Walk Detection: N/A.

Step Counting: We smoothed the acceleration magnitude using a centred moving average (window size $MovAvr_{win}$) and applied a windowed peak detection (window size $Peak_{win}$) to find the peaks associated with the heel strike.

Mean Crossings Counts (MCC)

Walk Detection: N/A.

Step Counting: We smoothed the acceleration magnitude using a centred moving average (window size $MovAvr_{win}$), sliced into windows of width $Mean_{win}$, and counted each positive crossing of the window mean as a step.

Normalised Autocorrelation SC (NASC)

Walk Detection: We took the NASC algorithm from [47]. When the standard deviation threshold, σ_{thresh} , was exceeded, we evaluated the normalised auto-correlation over a 2 s window for a series of appropriate time lags (τ_{min} to τ_{max}). If the maximum of the auto-correlation exceeded a threshold, R_{thresh} , the user was asserted to be walking. The walking state ended when the standard deviation fell back below σ_{thresh} .

Step Counting: We continually computed the normalised autocorrelation for rolling 2 s time windows. For each window, we took the time lag corresponding to the maximum autocorrelation as the stride period and the fractional steps computed. The step count was the sum of these fractional values.

Dynamic Time Warping (DTW)

Walk Detection: N/A.

Step Counting: We used DTW [28] to match a gait template with its occurrences in the acceleration signal (the non-linear mapping established by DTW accounts for changes in the walk speed or the signal shape). We manually identified optimal gait templates for every subject and repeatedly applied DTW subsequence matching to the smoothed signal to extract individual steps. We varied minimal step length ($step_{min}$) and maximal distance between two steps/strides

($step_{max}/stride_{max}$). This amounted to the best-case scenario of the algorithm's performance as the gait templates would normally be extracted by another preprocessing step.

Frequency Domain

Short Term Fourier Transform (STFT)

Walk Detection: We used STFT [55] to split the signal into successive time windows of size DFT_{win} and labelled each as walking if it contained significant (greater than some threshold DFT_{thresh}) spectral energy at typical walking frequencies, $freq_{walk}$.

Step Counting: We used STFT to compute a fractional number of strides for each window by dividing the window width by the dominant walking period it detected. These fractional values were then summed to estimate the strides taken.

Continuous/Discrete Wavelet Transform (CWT/DWT)

Walk Detection: We extracted walking periods by thresholding (threshold $ener_{thresh}$) the ratio between the energy in the band of walking frequencies $freq_{walk}$ and the total energy across all frequencies [5, 43], where the energy was computed using CWT/DWT [36, 54].

Step Counting: We zeroed all CWT/DWT coefficients outside of the $freq_{walk}$ band and inverted the transform so that the resulting signal does not contain a DC component and retains only a walk information. Individual steps were then extracted using mean-crossings as per the MCC algorithm.

Feature Clustering

The two feature clustering techniques we tested, Hidden Markov Models (HMMs) and K-Means Clustering (KMC) [7], are representative of sequential (resp. static) clustering. We did not consider classification methods (such as e.g., SVMs) as these require supervised learning. We chose KMC in favour of self-organising maps due to its lesser computational cost and better overall accuracy [53]. For similar reasons (in addition to the sequential nature of HMMs), we favoured HMMs over GMMs (cf. [38]).

The features we used as an input were signal mean, signal energy and standard deviation in the time domain, and dominant frequency and spectral entropy in the frequency domain. We did not use individual FFT coefficients as under unconstrained movements dominant frequencies tend to vary, similarly as across different types of ambulation [20]. For earlier reasons, we also did not include dimensionality reduction nor more exotic features (e.g. weightlessness [19]).

Hidden Markov Model (HMM)

Walk Detection: We trained a two-state (walk/idle) HMM with Gaussian emissions in an unsupervised fashion using the Viterbi algorithm [56]. We alternated the selection of feature vectors passed as an input into the model and compared the results by performing *leave-one-out* cross-validation. We used *tied* covariance matrices, which are the same for all states of the HMM (to allow learning HMMs from less data but also to avoid overfitting).

Step Counting: When applied to the walk portion of the signal, an HMM can be trained to distinguish between different phases of the gait period such as heel-off, toe-off, heel strike,

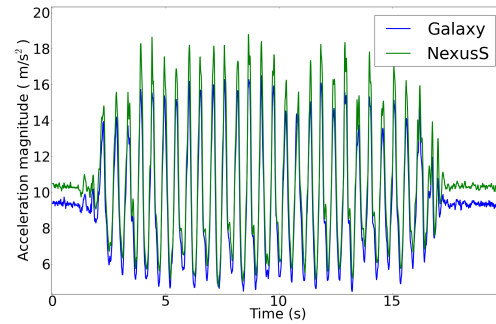


Figure 2. Acceleration magnitudes obtained from two smartphones with different accelerometer sensitivities. The magnitudes are slightly displaced, but their shapes are almost identical.

and foot stationary [37]. We experimented with various numbers of hidden states but found that a simple HMM with two hidden states produced equivalent step counting results. The model would regularly assign one state to the positive hill and the other to the negative valley of the step period even in the presence of significant noise. To accommodate variations in walking speed, we trained the HMM dynamically over a rolling time window ($Train_{win}$).

K-Means Clustering (KMC)

Walk Detection: KMC can be straightforwardly used for WD (see e.g. [9]), however on our dataset we observed significantly poorer performance than for other techniques, so we did not include it in our study.

Step Counting: We partitioned feature vectors over a rolling time window ($Train_{win}$) into two clusters (hill/valley) using Lloyd's algorithm [34]. We restarted the algorithm 50 times with different centroid seeds to avoid falling into local minima. The prediction was achieved by calculating the closest cluster. As for HMM, we performed *leave-one-out* cross-validation to compare different selection of features.

EXPERIMENTAL METHOD AND DATA COLLECTION

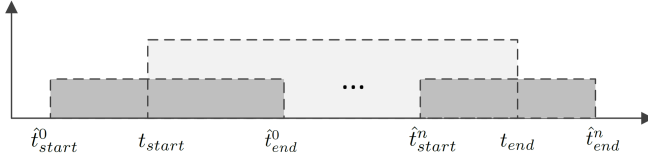
If smartphones are to be used for walk detection and/or step identification, it is important to relax the constraints on user motion during testing. Away from the laboratory, users are unlikely to keep the smartphone in the same place on the body, and they must be expected to do more than walk. We explored several common phone placement scenarios: hand-held, handheld plus interaction (e.g. typing a message), front trouser pocket, back trouser pocket and in a backpack or handbag. We also avoided the use of a treadmill as in some previous studies and encouraged users to walk at varying speeds in order to get more realistic walking samples.

Our tests involved gathering accelerometer data from a smartphone (Galaxy Nexus GT-I9250 running Android 4.1.1, with the embedded Bosch BMA220 accelerometer sampled at 100 Hz) given to a set of subjects who were asked to perform a simple walking trial. All participants were given the same smartphone model to make the data uniform, but we expect this not to affect conclusions of our study (see Figure 2).

Each trial involved the user carrying one or more smartphones over a straight-line route by hand, in a pocket, in a back-

Gender	Age [yrs]	Height [cm]
F: 9	15-19: 9	150-159: 3
M: 18	20-29: 18	160-169: 5
		170-179: 11
		180-189: 8

Table 2. Statistics about subjects participating in our data collection.

Figure 3. Illustration of intervals in the definition of ϵ_+ and ϵ_- .

pack or in a handbag. We divided the route into three equal segments and instructed the subjects to walk the first as they pleased, speed up in the second, and slow down in the third. The subjects were free to interpret the speeds as they saw fit. Data were logged to the smartphones for offline analysis and a video camera was used to record each trial for later event verification. We refer to the data collected from a specific walk as a *trace*.

In all, our study used 27 subjects recruited from both genders, a variety of age groups, and a range of heights (see Table 2 for statistics). A total of 130 sensor traces were collected.

RESULTS

Ground Truths

We established a baseline walk period for each of the 130 traces. This was achieved by manually finding the walk-start (t_{start}) and walk-end (t_{end}) events from a combination of trace inspection and video review. Each trace sample was labelled as walking or non-walking. We also counted the steps taken using the same approach. For some traces there was ambiguity over the number of steps taken (some users shuffled at the start or end, for example)—we excluded all traces where there was not a clear step count, leaving 80 traces with known counts.

Parameter Optimisation

Each of the algorithms considered in this study is associated with a number of parameters affecting its behaviour and performance. We optimised these parameters for our specific dataset by performing an exhaustive search across all feasible values. For each choice of values, we evaluated the WD and SC errors for all traces and then applied the Friedman test [13] with the Iman and Davenport correction [22] to test for statistical significance.¹ Performing this optimisation means we can treat our subsequent results as an upper bound to the accuracy.

Optimisation for WD

We optimised WD parameters using the manually-determined ground truth walk periods. Each algorithm identified n disjoint walking periods (ideally $n=1$ but this was not always true), with the i^{th} period spanning a time range $[\hat{t}_{start}^i, \hat{t}_{end}^i]$.

¹The Friedman test has been suggested as the preferred test for statistical comparison of more methods over multiple datasets [11].

For $\hat{t}_{start}^0 \leq t_{start} \leq \hat{t}_{end}^0$ and $\hat{t}_{start}^n \leq t_{end} \leq \hat{t}_{end}^n$ (see Figure 3 for this case; other cases are similar) we defined the false positive error (ϵ_+), false negative error (ϵ_-) and total error (ϵ_{total}) as:

$$\begin{aligned}\epsilon_+ &= (t_{start} - \hat{t}_{start}^0) + (\hat{t}_{end}^n - t_{end}), \\ \epsilon_- &= \sum_{i=1}^n (\hat{t}_{start}^i - \hat{t}_{end}^{i-1}), \\ \epsilon_{total} &= \epsilon_+ + \epsilon_-.\end{aligned}$$

The optimal parameter values for various WD algorithms determined using these metrics and the Friedman test are detailed in Table 3.

Algorithm	Optimal parameter values
MAGN_TH	$magn_{thresh} = 10.5$
ENER_TH	$ener_{thresh} = 0.04, ener_{win} = 1 \text{ s}$
STD_TH	$\sigma_{thresh} = 0.6, std_{win} = 0.8 \text{ s}$
NASC+STD_TH	$R_{thresh} = 0.4, \tau_{min} = 0.4 \text{ s}, \tau_{max} = 1.5 \text{ s}, \sigma_{thresh} = 0.6, std_{win} = 0.9 \text{ s}$
STFT	$DFT_{win} = 0.7 \text{ s}, DFT_{thresh} = 20, freq_{walk} = (0.01 \text{ Hz}, 7 \text{ Hz})$
CWT	Difference of Gaussians (DOG) wavelet with $\sigma = 6, freq_{walk} = (1 \text{ Hz}, 5 \text{ Hz}), ener_{thresh} = 0.45$
DWT	Haar wavelet, $freq_{walk} = (0.5 \text{ Hz}, 4 \text{ Hz}), ener_{thresh} = 0.0095$
HMM	Feature vectors: magnitude, energy; $ener_{win} = 0.5 \text{ s}$

Table 3. Optimal parameter values for the WD algorithms

Optimisation for SC

For step counting, our error metric was the magnitude of the difference between the estimated step count and ground truth. We ranked the parameter sets and applied the Friedman test as above to find the optimal set (see Table 4). The $MovAver_{win}$ value corresponds to the centred moving average window size, and $Peak_{win}$ ($Mean_{win}$) to the size of the window where we search for the peak (i.e. minimum distance between two subsequent local maxima) or mean, respectively. We note that the latter optimal window size seems to be approximately equal to the average step period. The meaning of other parameters in Table 4 is the same as in Table 3.

We note in passing that algorithms with multiple parameters often exhibit local minima in the error vs parameter plots, and care must be taken if optimising them without performing a full search of the parameter space. Figure 4, for example, shows the relative error of the WPD algorithm with respect to the $Peak_{win}$ and $MovAver_{win}$ parameters. This exhibits a strong ‘saddle’ shape due to small spans giving spurious peaks (and hence steps) and large spans inadvertently merging multiple peaks.

Algorithm Comparison

Using the optimal parameters from the previous section, we compared the accuracy of the WD and SC algorithms when

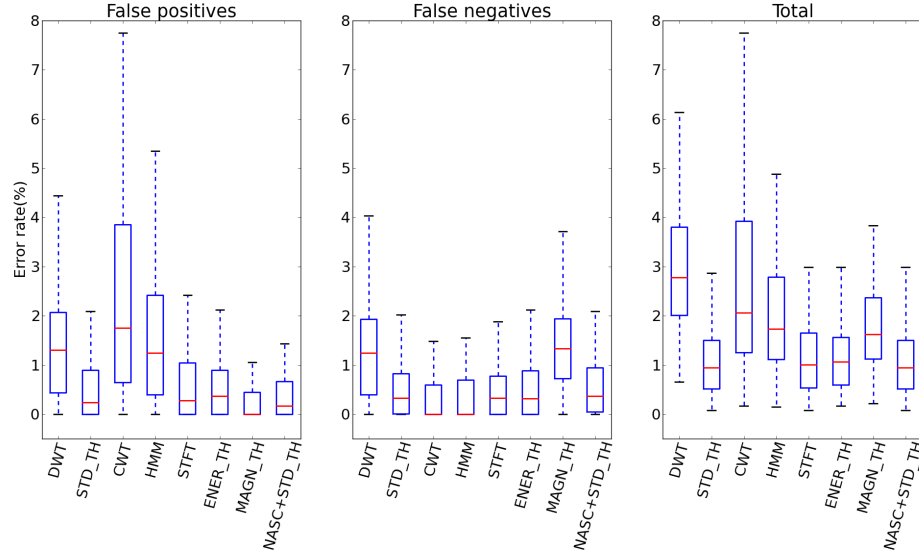


Figure 5. Walk detection results

Algorithm	Optimal parameter values
WPD	$MovAvr_{win} = 0.31$ s, $Peak_{win} = 0.59$ s
MCC	$MovAvr_{win} = 0.29$ s, $Mean_{win} = 0.5$ s
NASC	$\tau_{min} = 0.48$ s, $\tau_{max} = 0.8$ s
DTW	$MovAvr_{win} = 0.3$ s, $step_{min} = 0.25$ s, $step_{max} = 5.5$ s, $stride_{max} = 6$ s
STFT	$DFT_{win} = 4$ s, $freq_{walk} = (0$ Hz, 2.5 Hz)
CWT	Morlet wavelet with $\sigma = 6$, $freq_{walk} = (0.3$ Hz, 2.5 Hz)
DWT	Haar wavelet, $freq_{walk} = (0.03$ Hz, 3 Hz)
HMM	Feature vector: smoothed magnitude, $MovAvr_{win} = 0.3$ s, $Train_{win} = 1$ s
KMC	Feature vector: smoothed magnitudes, $MovAvr_{win} \in \{0.2, 0.3, 0.5\}$ s, $Train_{win} = 5$ s

Table 4. Optimal parameter values for the SC algorithms

applied to our traces. As such, these results present the algorithms in their best light: optimised parameters and semi-contrived datasets with a limited number of actions.

Walk Detection

By comparing the ground truth sample labels to those output by the algorithms we classified each sample as a true positive, a true negative, a false positive, or a false negative. For each trace we computed the number of false positive samples, n_+ ; the number of false negative samples, n_- ; and the total number of erroneously labelled samples, $n_{total} = n_+ + n_-$. Figure 5 illustrates the distributions of these quantities across all traces, presented relative to the number of samples in the true walk period, n_{walk} —e.g. $n_+/n_{walk} \times 100\%$.

We observe that the error rates amongst the algorithms were broadly comparable, with median values less than 2% and similar false positive and false negative rates. The two ex-

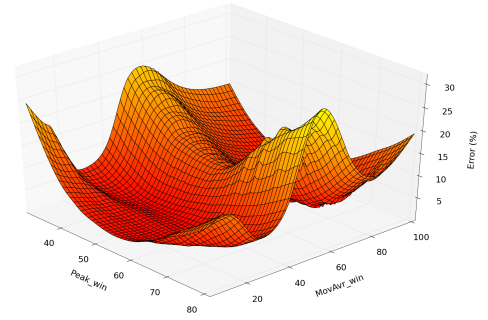


Figure 4. The saddle-shaped error vs parameter plot for the WPD algorithm. With the $Peak_{win}$ being too small, we detect more peaks (i.e. steps) than there actually are, so the error increases. If the $Peak_{win}$ is too big, we lose some peaks that are valid steps (since there can be only one peak per window), so the error again increases.

ceptions are for the wavelet transforms, which both exhibited high false positive rates. Careful review of the individual results revealed that the $ener_{thresh}$ threshold was at fault since finding a universal threshold was not possible due to the different walking speeds giving very different energies. In particular, the slow walk section often gave a transform with very little energy at the frequencies of interest. Our optimisation was therefore forced to choose between a high threshold (resulting in failure to capture many low-energy slow walks) and a low threshold (resulting in many spurious walking results). The results of Figure 5 reveal a tendency for the latter situation. The other algorithms coped much better with the wide range in signal energies.

However, all of these algorithms are essentially activity detectors that do not explicitly ‘recognise’ walking—indeed NASC was proposed as an extension to standard deviation threshold-

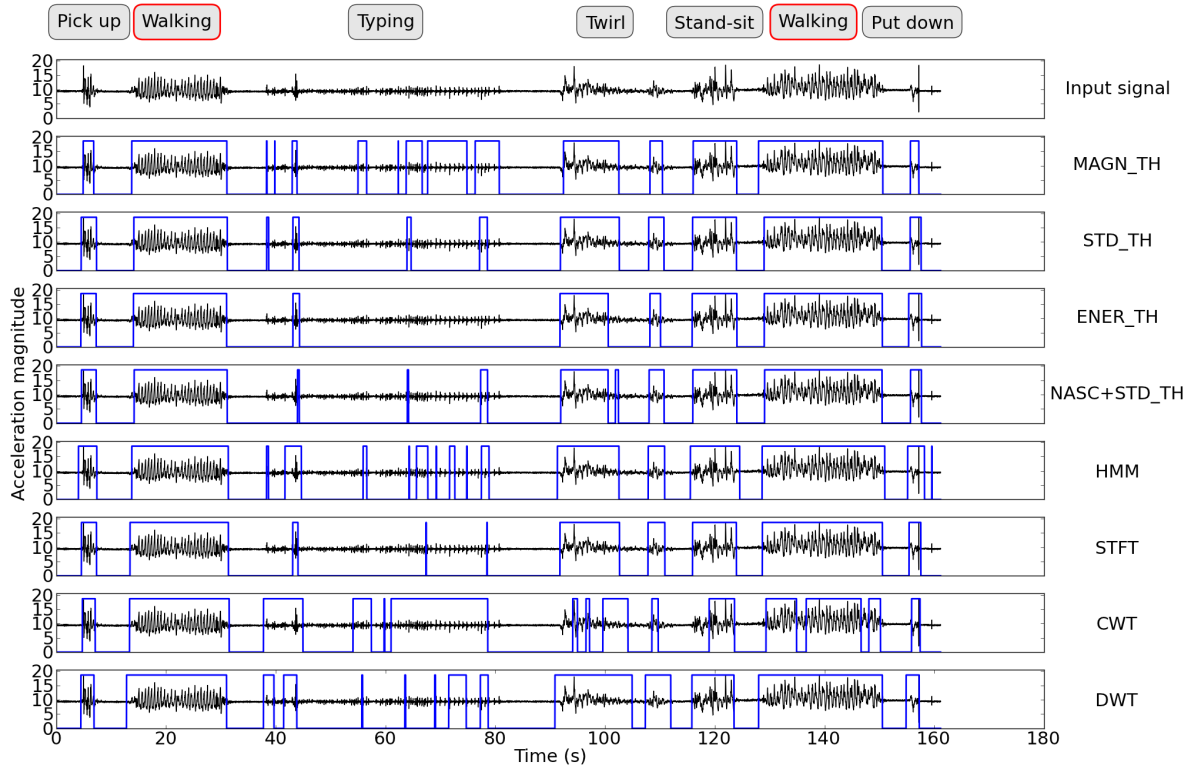


Figure 6. Walk detection amongst a variety of activities

ing that was more robust. We applied the algorithms to a more complex trace that involved a greater deal of activities chosen to repeat those in [47]—i.e. picking up and putting down the phone, typing, twirling on a rotating office chair, transitioning between standing and sitting, walking and idle. Figure 6 illustrates the output from the WD algorithms. We observe that all but the CWT algorithm correctly identified walking (i.e. gave true positives and no false negatives), but that these were mixed amongst a significant number of false positives. Therefore any walk detection algorithm for an unconstrained smartphone will have to implement a robust false positive rejection algorithm that can be applied whenever these simpler algorithms indicate walking.

Step Counting

We measured the accuracy of each SC algorithm by comparing the estimated step count (c_{est}) with the ground truth values (c_{gt}). Figure 7 illustrates the results across the 80 traces with reliable step counts using an error rate metric defined as:

$$\frac{c_{est} - c_{gt}}{c_{gt}} \times 100\%.$$

We observe that all of the algorithms had a median error rate of less than 3% and were more inclined to undercount than overcount. We attribute this to the fact that the algorithms are unlikely to find multiple cycles where one exists, but may only find one cycle when multiple exist. This is particularly likely at the start and end of a walk, where the steps typically have different properties (lower energy, longer duration, etc).

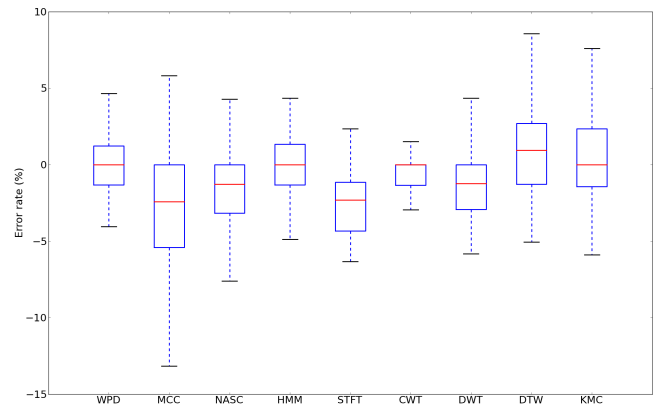


Figure 7. Step counting errors

Smartphone Placement

To illustrate the effect of smartphone placement, Figure 8 and Figure 9 give more detailed views of the data underlying Figure 7. Figure 8 presents the same box-whisker plots but separated by placement; Figure 9 gives the error distribution with each column colour-coded to indicate the proportion attributable to a specific placement.

All but one placement shows the expected tendency towards undercounting. Carrying a phone within a handbag, however,

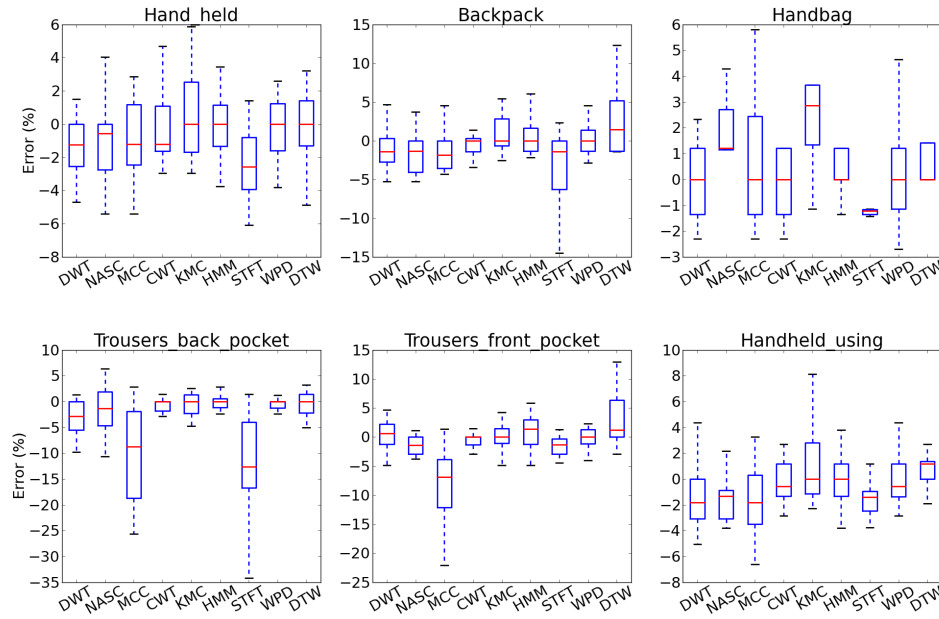


Figure 8. Step counting errors by placement

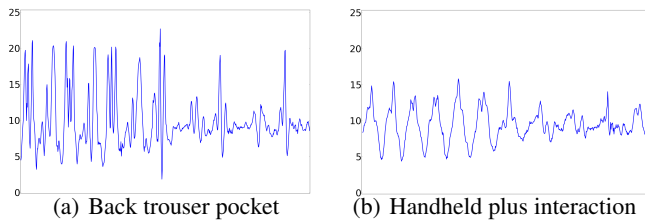


Figure 10. Time-aligned segments of traces for the ‘back trouser pocket’ and ‘handheld plus interaction’ phone placement scenarios. The fast-paced portion in (a) exhibits perturbations that caused undercounting for the STFT, MCC, and NASC algorithms.

had more of a tendency to overcount. We attribute this to the vertical bounce being amplified and propagated by the handbag straps, producing occasional extra oscillations in the acceleration trace. This aside, the majority of placements have little effect on the SC performance. A notable exception is the back trouser pocket, which significantly increased the likelihood of undercounting for the STFT, MCC, and NASC algorithms. We observe that the traces for the back trouser pocket often exhibited a less cyclic nature compared to other placements (see Figure 10). We attribute this to the relaxing of the gluteus maximus during the walking cycle, which may introduce extra oscillations that perturb the underlying trace periodicity.

CONCLUSIONS

This work has provided a detailed and fair comparison of a range of walk detection and step counting algorithms in one place for the first time. The evaluations were based on a large dataset of 130 walk traces from 27 people collected using smartphone accelerometers in six different positions.

The best performing algorithms for walk detection were the two thresholds based on the standard deviation (STD_TH) and the signal energy (ENER_TH), STFT and NASC, all of which exhibited similar error medians and spreads. We observed that most of the tested algorithms can reliably detect walking within a trace that contains only periods of idle and walking activity. Given the relative computational complexities, then, we favour the thresholding techniques (particularly on standard deviation).

The overall best step counting results were obtained using the windowed peak detection (WPD), HMM and CWT algorithms, each with a median error of $\sim 1.3\%$. Given its relative simplicity, the WPD would seem to be an optimal choice when we are confident that walking is occurring.

However, applying SC on a section of a trace that is actually a false positive from the WD algorithm, we are likely to obtain an erroneous step count. This is a serious issue since the previous section highlighted the high probability of false positives from WD algorithms. We have trialled a number of possible solutions to this, including applying multiple step counting algorithms to look for (a lack of) consistency amongst their outputs. However, the results have not been a significant increase in SC accuracy.

In conclusion, we determined that none of the algorithms is 100% reliable and this is significant for the PDR community. In order to robustly track based on the output from these algorithms it will be necessary to account for missing and extra steps when step counting and false positives in walk detection. Where probabilistic methods are used, it should be possible to account for these errors, usually at the cost of extra processing (for example, the use of a particle filter, which would need more particles to adequately model the error).

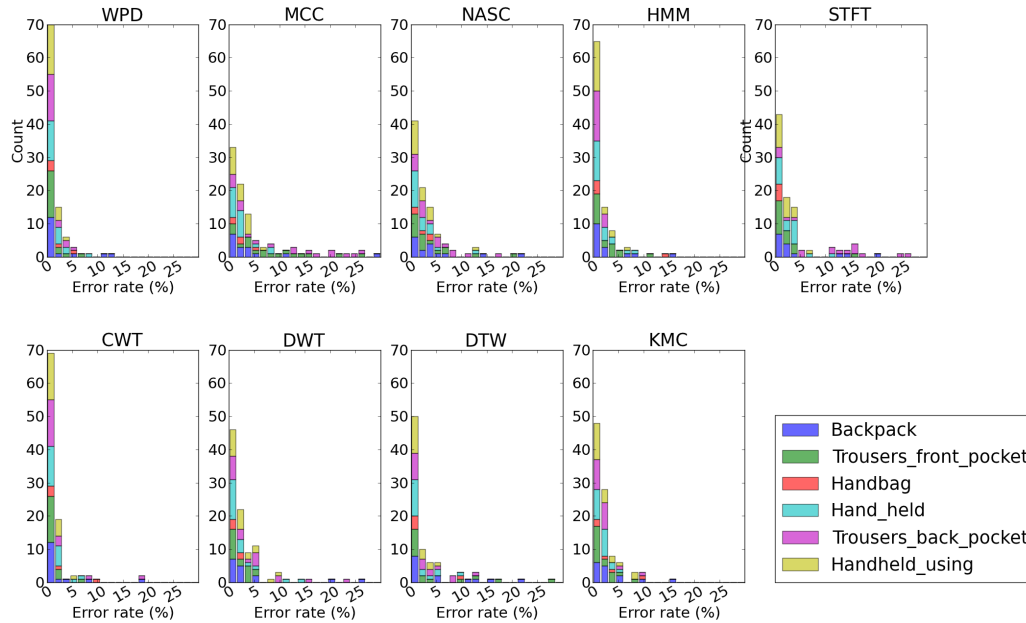


Figure 9. Step counting error distributions

Recommendations

Based on our results we offer the following recommendations:

- A straightforward thresholding of the accelerometer standard deviation will robustly and cheaply detect periods of walk. It will, however, also produce significant false positives in general use.
- A windowed peak detection algorithm is overall the optimal option for step counting regardless of smartphone placement.
- When evaluating any system that uses walk detection or step counting, it is important to use the inputs from a variety of people and carrying positions, and to consider the effect of typical non-walking actions.

In the future we aim to evaluate the use of different or additional sensors for walk detection and step counting, as well as evaluate the performance of the algorithms described here when incorporated as the first stage of a PDR system.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous UbiComp'13 shepherd and reviewers for helpful comments and suggestions, the student Bono Xu who helped us recruit the participants and conduct the experiments, and the many volunteers who assisted us in gathering the large dataset.

REFERENCES

1. H. J. Ailisto, M. Lindholm, J. Mantjarvi, E. Vildjiounaite, and S. M. Makela. Identifying people from gait pattern with accelerometers. In *SPIE*, volume 5779, page 7, 2005.
2. F. Allen, E. Ambikairajah, N. Lovell, and B. Celler. An adapted gaussian mixture model approach to accelerometry-based movement classification using time-domain features. In *EMBS '06*, pages 3600–3603, 2006.
3. A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. J. M. Havinga. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *ARCS '10*, pages 167–176, 2010.
4. L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *PERVASIVE '04*, pages 1–17, 2004.
5. P. Barralon, N. Vuillerme, and N. Noury. Walk detection with a kinematic sensor: Frequency and wavelet comparison. In *EMBS '06*, pages 1711–1714, 2006.
6. S. Beauregard. A helmet-mounted pedestrian dead reckoning system. In *IFAWC '06*, pages 1–11, 2006.
7. C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007.
8. S. Chen, J. Lach, O. Amft, M. Altini, and J. Penders. Unsupervised activity clustering to estimate energy expenditure with a single body sensor. In *BSN '13*, 2013.
9. B. Choe, J.-K. Min, and S.-B. Cho. Online gesture recognition for user interface on accelerometer built-in mobile phones. In *ICONIP '10*, pages 650–657, 2010.
10. W. Dargie. Analysis of time and frequency domain features of accelerometer measurements. In *ICCCN '09*, pages 1–6, 2009.
11. J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
12. D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.
13. M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11:86–92, 1940.
14. D. Gafurov and E. Snekenes. Towards understanding the uniqueness of gait biometric. In *FG '08*, pages 1–8, 2008.
15. A. R. Golding and N. Lesh. Indoor navigation using a diverse set of cheap, wearable sensors. In *ISWC '99*, pages 29–36, 1999.

16. P. Goyal, V. Ribeiro, H. Saran, and A. Kumar. Strap-down pedestrian dead-reckoning system. In *IPIN '11*, pages 1–7, 2011.
17. R. Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys & Tutorials*, PP, 2013.
18. Harvey WeinBerg. AN-602: Using the ADXL202 in Pedometer and Personal Navigation. Technical report, Analog Devices, 2002.
19. Z. He, Z. Liu, L. Jin, L.-X. Zhen, and J.-C. Huang. Weightlessness feature - a novel feature for single tri-axial accelerometer based activity recognition. In *ICPR '08*, pages 1–4, 2008.
20. T. Huynh and B. Schiele. Analyzing features for activity recognition. In *sOc-EUSAI '05*, pages 159–163, 2005.
21. R. Ibrahim, E. Ambikairajah, B. Celler, and N. Lovell. Time-frequency based features for classification of walking patterns. In *DSP '07*, pages 187–190, 2007.
22. R. Iman and J. Davenport. Approximations of the critical region of the friedman statistic. *Communications in Statistics – Theory and Methods*, 9:571595, 1980.
23. T. Iso and K. Yamazaki. Gait analyzer based on a cell phone with a single three-axis accelerometer. In *Mobile HCI '06*, pages 141–144, 2006.
24. J. J. Kavanagh and H. B. Menz. Accelerometry: a technique for quantifying movement patterns during walking. *Gait & posture*, 28(1):1–15, 2008.
25. J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park. A Step, Stride and Heading Determination for the Pedestrian Navigation System. *Journal of Global Positioning Systems*, 3(1-2):273–279, 2004.
26. T. Kobayashi, K. Hasida, and N. Otsu. Rotation invariant feature extraction from 3-d acceleration signals. In *ICASSP '11*, pages 3684–3687, 2011.
27. A. Krause, D. P. Siewiorek, A. Smailagic, and J. Farrington. Unsupervised, dynamic identification of physiological and activity context in wearable computing. In *ISWC '03*, pages 88–97, 2003.
28. J. B. Kruskal and M. Liberman. The symmetric time-warping problem: from continuous to discrete. In *Time Warps, String Edits, and Macromolecules - Theory and Practice of Sequence Comparison*, 1983.
29. J. Kwapisz, G. Weiss, and S. Moore. Cell phone-based biometric identification. In *BTAS '10*, pages 1–7, 2010.
30. J. R. Kwapisz, G. M. Weiss, and S. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explorations*, 12(2):74–82, 2010.
31. S. W. Lee and K. Mase. Activity and Location Recognition Using Wearable Sensors. *IEEE Pervasive Computing*, 1(3):24–32, 2002.
32. J. Lester, C. Hartung, L. Pina, R. Libby, G. Borriello, and G. Duncan. Validated caloric expenditure estimation using a single body-worn sensor. In *UbiComp '09*, page 225, 2009.
33. F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *UbiComp '12*, pages 421–430, 2012.
34. S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
35. Y. Makihara, T. N. Thanh, H. Nagahara, R. Sagawa, Y. Mukaigawa, and Y. Yagi. Phase registration of a single quasi-periodic signal using self dynamic time warping. In *ACCV '10*, pages 667–678, 2010.
36. S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1999.
37. A. Mannini and A. Sabatini. A hidden Markov model-based technique for gait segmentation using a foot-mounted gyroscope. In *EMBC '11*, pages 4369–4373, 2011.
38. A. Mannini and A. M. Sabatini. Accelerometry-based classification of human activities using markov modeling. *Comp. Int. and Neurosc.*, 2011.
39. J. Mantyjarvi, J. Himberg, and T. Seppanen. Recognizing human motion with multiple acceleration sensors. In *SMC '01*, volume 2, pages 747–752 vol.2, 2001.
40. M. Marschollek, M. Goevercin, K.-H. Wolf, B. Song, M. Gietzelt, R. Haux, and E. Steinhagen-Thiessen. A performance comparison of accelerometry-based step detection algorithms on a large, non-laboratory sample of healthy and mobility-impaired persons. In *EMBS '08*, volume 2008, pages 1319–22, 2008.
41. C. Nickel, H. Brandt, and C. Busch. Benchmarking the performance of SVMs and HMMs for accelerometer-based biometric gait recognition. In *ISSPIT '11*, pages 281–286, 2011.
42. C. Nickel, H. Brandt, and C. Busch. Classification of acceleration data for biometric gait recognition on mobile devices. In *BIOSIG '11*, pages 57–66, 2011.
43. M. Nyan, F. Tay, K. Seah, and Y. Sitoh. Classification of gait patterns in the timefrequency domain. *Journal of Biomechanics*, 39(14):2647 – 2656, 2006.
44. T. Pfau, M. Ferrari, K. Parsons, and A. Wilson. A hidden markov model-based stride segmentation technique applied to equine inertial sensor trunk movement data. *Journal of Biomechanics*, 41(1):216 – 220, 2008.
45. S. Pirttikangas, K. Fujinami, and T. Nakajima. Feature selection and activity recognition from wearable sensors. In *UCS '06*, pages 516–527, 2006.
46. S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton. Activity identification using body-mounted sensors – a review of classification techniques. *Physiological Measurement*, 30(4):R1+, 2009.
47. A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee - zero-effort crowdsourcing for indoor localization. In *Mobicom '12*, page 293, 2012.
48. C. Randell, C. Djallil, and H. Muller. Personal position measurement using dead reckoning. In *ISWC '03*, pages 166–173, 2003.
49. N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. In *AAAI '05*, pages 1541–1546, 2005.
50. L. Rong, D. Zhiguo, Z. Jianzhong, and L. Ming. Identification of individual walking patterns using gait acceleration. In *ICBBE '07*, pages 543–546, 2007.
51. B. Santhirayagam, D. Lai, C. Jiang, A. Shilton, and R. Begg. Automatic detection of different walking conditions using inertial sensor data. In *IJCNN '12*, pages 1–6, 2012.
52. P. Siirtola and J. Rönning. Recognizing human activities user-independently on smartphones based on accelerometer data. *IJIMAI*, 1(5):38–45, 2012.
53. P. (Sundar) Balakrishnan, M. Cooper, V. Jacob, and P. Lewis. A study of the classification capabilities of neural networks using unsupervised learning: A comparison with k-means clustering. *Psychometrika*, 59(4):509–525, 1994.
54. C. Torrence and G. P. Compo. A practical guide to wavelet analysis. *Bull. Amer. Meteor. Soc.*, 79(1):61–78, 1998.
55. M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. Prentice Hall PTR, 1995.
56. A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.
57. J.-H. Wang, J.-J. Ding, Y. Chen, and H.-H. Chen. Real time accelerometer-based gait recognition using adaptive windowed wavelet transforms. In *APCCAS '12*, pages 591–594, 2012.
58. O. Woodman and R. Harle. Pedestrian Localisation for Indoor Environments. In *UbiComp '08*, 2008.
59. A. Ying, Hong, C. A. Silex, A. Schnitzer, A., S. A. Leonhardt, and M. Schiek. Automatic Step Detection in the Accelerometer Signal. In S. Leonhardt, T. Falck, and P. Mähönen, editors, *BSN '07*, 2007.