Step Detection through Ultra-Low Complexity Zero Crossing Analysis

Fabian Hölzke, Jakob Heller, Salo A. Deatcu, Frank Golatowski, Dirk Timmermann

Institute of Applied Microelectronics and CE
University of Rostock,
Rostock, Germany
Email: fabian.hoelzke2@uni-rostock.de

***Abstract*—Step detection is a common application in battery driven wearables. It enables fitness tracking as well as indoor localization. However, current state of the art approaches heavily rely on spectral properties of the acceleration signal or decisions-trees comparing peaks and valleys, using various thresholds and timings. This requires accurate AD conversion as well as complex calculations to the disadvantage of battery life. Consequently, an ultra-low complexity step detector with competitive accuracy is desirable. We propose a zero-crossing interval and Bayesian-analysis based step detection algorithm which requires minimal computation at runtime, using a-priori knowledge from pre-computed statistical analysis. We compare our approach to a classifier that uses the more accurate but costly spectral properties of the data. The statistical analysis for pre-computation as well as evaluation is done using the annotated sensor data of the OU-ISIR Gait Database. Our evaluation shows the presented method outperforms classification with spectral features and delivers a step count accuracy that is competitive with state of the art commercial products.**

I. Introduction

Step detection in smartphones and other wearable devices has become ubiquitous. Here, step counting commonly serves to track or encourage exercise. A more complex use case for step detection is localization in buildings, when GPS can not be used reliably. The step count, together with walking direction and an estimation of step length, enables the tracking of the users' movement starting from a known initial position and orientation.

In order to detect steps, the signal, e.g. of an accelerometer, needs to be continuously monitored for features that indicate a step. This is typically done by a sliding window approach with a certain overlap. The window overlap ensures that a step event is contained, at least for the most part, within one window of sensor data. In general, one may analyze inertial data in the time domain and/or frequency domain for step detection. For instance, for each window one may detect peaks in the time domain [1] or evaluate certain parts of the frequency spectrum of a signal [2] to detect the step event. These spectral markers rely on detecting spectral peaks within the narrow frequency band that is influenced by human motion. For each overlapping window, a new transformation of the signal to the spectral domain needs to be computed. Consequently, parts of the data (within the overlapping interval) are at least transformed twice to produce the signal spectrum of the propagated window. Keeping in mind that fast Fourier transform (FFT) has a computational complexity of $\mathcal{O}(n \log n)$, we see potential for optimization. Since step detection is usually done on battery powered wearable devices, power consumption of the step detection system is an important factor and should be minimized. Simple spectral discrimination techniques, such as zero crossing intervals, typically feature lower complexity and power consumption compared to spectral analysis such as FFT. Consequently, an alternative approach is to estimate the leading frequency through the zero crossings alone. This approach is called zero crossing interval analysis (ZCIA). It has been successfully implemented for seizure prediction based on EEG data [3] and voice activity detection in audio recordings [4]. The connection of zero crossing intervals with the spectrum of the signal is a long established subject of research as shown in [5] and [6]. While ZCIA is not a direct representation of the actual spectrum, it is correlated to the dominant frequency in a way that allows for step detection without explicit transformation to the frequency domain.

ZCIA may be implemented completely in hardware, minimizing actual computation: It only requires $1-$bit AD conversion, i.e. checking weather the sensor output of a wearable accelerometer is above or below a set threshold. This can be typically realized with a primitive comparator circuit or a slightly more complex Schmitt trigger, if noise tolerance is desired. The binary output of this $1-$bit AD converter circuit is then analyzed for timing intervals between edges of the converted signal to produce the zero crossing intervals for further analysis. Additionally ZCIA allows updating the zero crossings of a window using only the new data from the window propagation, without recalculation of the window as a whole. Moreover a-priori knowledge of the underlying application, e.g. the crossing intervals when taking a step, allows checking the measured intervals against the known interval distribution of the desired event. Consequently, a measured interval can be directly associated with the likelihood of representing the desired event by a simple look-up operation from memory directly producing a confidence measure for the step event.

Since the above mentioned factors contribute to lower power consumption, the power draw of step detection/counting systems may be significantly decreased by implementing a zero crossing based approach if comparable accuracy can be achieved. Consequently the aim of this work is to investigate and evaluate a zero crossing based approach for step detection using a given dataset.

This paper is structured as follows: Section II describes related work regarding UWB and PDR based positioning and the combination of both approaches. The dataset enabling the Bayesian analysis and final evaluation is presented in Section III. Following the pre-processing of the data is described in Section IV while the actual approach for step detection is outlined in V. The evaluation approach is described in Section VI and the experimental results are presented in Section VII. Final conclusions are drawn in Section VIII.

II. Related Work

A step counting algorithm estimates the number of steps a user has taken based on event detection in inertial sensor data. A multitude of approaches to achieve this have been proposed. Threshold based approaches use one or multiple thresholds applied to the sensor data to detect steps. While this is the simplest approach, selecting a threshold that works in all cases is not trivial. Implementations can be found in [7] and [8]. Peak detection leverages peaks in the accelerometer signal that are produced when taking a step. To remove the impact of device orientation, the magnitude of the accelerometer signal should be used [1]. Auto correlation leverages the periodic nature of walking motion, computing the auto correlation of an inertial signal produces sharp peaks with every step. In [9], this is used to count steps in combination with walking detection based on the signals standard deviation and the maximum magnitude of the signals auto correlation. Step frequency estimation can be used if the walking duration and step frequency can be accurately determined. One may estimate the step count by simply multiplying the time spent walking with the step frequency [2].

Zero crossings are employed for step counting in [10]. Thresholds around the zero line are used to facilitate counting of zero crossings in high amplitude parts of the signal. The step count is then estimated by linear regression on the number of zero crossings in a signal. Further works combine multiple features of the sensor data such as the detection of zero crossings within an allowed interval, combined with the a threshold for signal variance [11]. Another approach is the detection of peaks above a certain threshold followed by a zero crossing [12]. In [13], ZCIA is used for event detection in satellite imagery. Machine learning techniques have shown promising results recently for the use of step event detection [14], [15]. Other than accelerometer data, gyroscopic data may also be used to register the periodic swing of legs or arms [16], [17]. Indoor localization is achieved by the combination of step event detection with step length estimation and heading estimation [18], [19].

III. Data Preparation

The following sub-chapters describe data used in this work and its' pre-processing for experimental validation.

A. *Dataset of true step events*

The inclusion of a-priori knowledge for the likelihood estimation of the step event necessitates a sufficiently large dataset. This work uses data collected by the Institute of Scientific and Industrial Research (ISIR) at Osaka University (OU). Specifically the Similar Action Inertial Dataset from the OU-ISIR Gait Database [20]. This dataset includes triaxial accelerometer and gyroscope data recorded from three inertial measuring units (IMUs) at 100 Hz. The IMUs are positioned on the subject's left and right side of the waist as well as the lower back using a belt.

Subjects with an even gender ratio and ages between 2 and 78 walked through a course which included level ground, stairs, and slopes.

The labels of the dataset include the interval of single steps, as well as the mode of walking (level walk, upstairs, downstairs, slope-up, slope-down). Furthermore, some sections of the data are labeled as invalid. These invalid sections may include walking and other actions and are not used in this work, because the specific action for each section is unknown. The data is provided in text files labeled with the subject ID which is used to obtain the subject's age and gender.

The sensor position and orientation is static, therefore we are able to choose an optimal sensor axis at design-time, as proposed in [2]. Excluding rotation, the hip mainly moves up and down when walking. Therefore the highest acceleration is experienced by the vertical axis and the acceleration in the horizontal plane is comparatively small. The vertical axis corresponds to the y-axis for all three IMUs. Indeed, testing different approaches confirmed that the y-axis of the accelerometer yields the best detection performance. Differences based on the choice between the left, center, and right IMU were minimal, so the center IMU was used for all further considerations.

In order to accurately evaluate and train the step detector, a dataset of positive and negative examples is needed. The positive examples are extracted from the previously described dataset. Non-overlapping windows of data from the sections marking a valid step are extracted until the desired number or positive examples is reached.

B. *Generation of non-step examples*

Negative examples, such as standing, gesturing, or other explicitly labeled non-walking actions, are not part the OU-ISIR Gait Database. Instead white Gaussian noise is used to approximate other actions that may be sampled by the IMU. Consequently, the noise is generated with a standard deviation of 0.4, which is the same as an average step of the training set, and an offset of -1 due to gravity. It is subjected to the same pre-processing described in Section IV. As the mean of the noise signal is non-zero the low-pass filter of the pre-processing leads to a ramp up from zero at the beginning of the generated counter examples. This is remedied by creating a longer signal and discarding the initial samples. The seed of the pseudo random number generator is retained in order to reproduce the set of negative examples during the development process. We employ a 1:1 balance of positive and negative examples for training and evaluation.

IV. Data pre-processing

Two factors were considered for pre-processing of sensor data: Human walking mainly generates acceleration signals in a range of 4 Hz to 6 Hz [21] and high frequency components produces a high number of zero crossings that interfere with the detection method. Consequently, all data is low-pass filtered by a 5th order Butterworth filter with a cutoff frequency of 6 Hz.

To remove the gravity from a given signal x we take the mean of the window $\overline{x}$ and subtract it from the signal. Furthermore we add an offset δ to the zero line, as this will avoid the generation of zero crossings for very low amplitude signals. In hardware, this would be realized by an analog high pass filter, followed by a constant level shift by δ. Subtracting this offset from the signal effectively moves the zero line by the desired amount.

$$x' = x - \overline{x} - \delta \tag{1}$$

Each sign change in x' represents a zero crossing. The index of the sample before the sign change is considered the zero crossing index. One could further improve the precision of the zero crossing detection by estimating the instance between the two samples that constitutes the sign change. However, since we are using data at a relatively high sample rate of 100 Hz, this is unlikely to meaningfully impact the detection performance and is thus not used in our case.

V. Step detection through Zero Crossing Interval Analysis

We employ a probabilistic approach to step detection: A window of acceleration data is analyzed for zero crossings. For each window a sample of consecutive zero crossing indices is generated. The probability if this sample was also observed for a step or for noise is then compared. Using Bayes' theorem, we can determine the probability if this window of data is indeed generated by a step.

Following, the probability for a step event given the timing of two consecutive zero crossings is denoted as $P(S|zc_1 \cap zc_2)$. The steps event is denoted by S, the case of noise by N and the position of the n-th zero crossing in a window by zc_n.

Bayes' rule states:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}. \tag{2}$$

We can apply this to the problem at hand:

$$P(S|zc_1 \cap zc_2) = \frac{P(zc_1 \cap zc_2|S) * P(S)}{P(zc_1 \cap zc_2)} \tag{3}$$

Here $P(S)$ refers to the prior probability that a sample belongs to the set of steps (without any knowledge about the zero crossing positions). For the data used, this is assumed to be 50% for both $P(S)$ and $P(N)$ (balanced dataset). The denominator describes the probability that a certain combination of zero crossing timings occurs, whether generated by step data or noise:

$$\begin{aligned} &P(zc_1 \cap zc_2) \\ &\quad = P(zc_1 \cap zc_2|S)P(S) + P(zc_1 \cap zc_2|N)P(N) \end{aligned} \tag{4}$$

This leaves $P(zc_1 \cap zc_2|S)$ and $P(zc_1 \cap zc_2|N)$ as the unknowns. Since we cannot assume statistical independence between the zero crossing positions, we have to compute the conditional probability for the second zero crossing based on the first:

$$P(zc_1 \cap zc_2|S) = P(zc_2|S \cap zc_1)P(zc_1|S) \tag{5}$$

The equation for $P(zc_1 \cap zc_2|N)$ is analogous to (5).

The conditional probabilities of zero crossing placements zc_1 and zc_2 are determined by first computing the respective probability density function (PDF) using kernel density estimation (KDE) with a Gaussian kernel on the training dataset. Zero crossings can only occur at integer values, so the characterization as a probability density function is, strictly speaking, not correct. Therefore, the PDF is sampled at integer values to produce an approximation of the respective probability mass functions (PMF).

For $P(zc_1|S)$ we simply collect all the position of the first zero crossing in every step of the training set and determine the PMF. To estimate $P(zc_2|S \cap zc_1)$ the second zero crossings have to be grouped according to the positions of the corresponding first zero crossing. Then the PMF is determined on the groups of zc_2 corresponding to each zc_1.

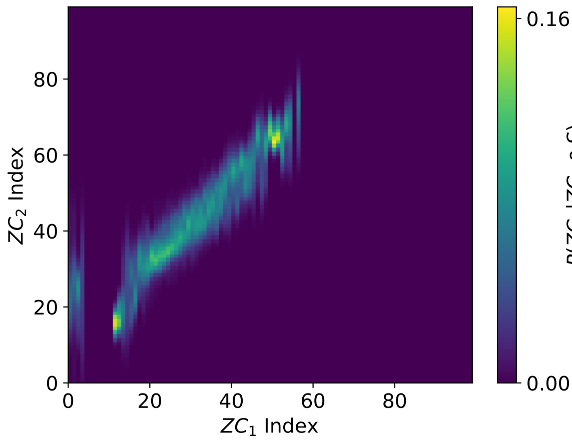
Furthermore, in order to perform a KDE multiple unique samples are needed and some zero crossings may occur rarely or not at all in the training set, so this condition is not always satisfied. Consequently, the PMF is set to zero in its entirety, if there is no zc_1 for an index or if there is not more than one zc_2 for one zc_1.

The result of these computations using is visualized in Fig. 1. The visualization reaffirms the previous assumption that the position of the second zero crossing is not statistically independent from the first, as the plot shows a linear relationship between the placement of both zero crossings. Furthermore. the fact that these probabilities show different properties for steps and noise supports the assumption that they can be used to discriminate between the two cases.

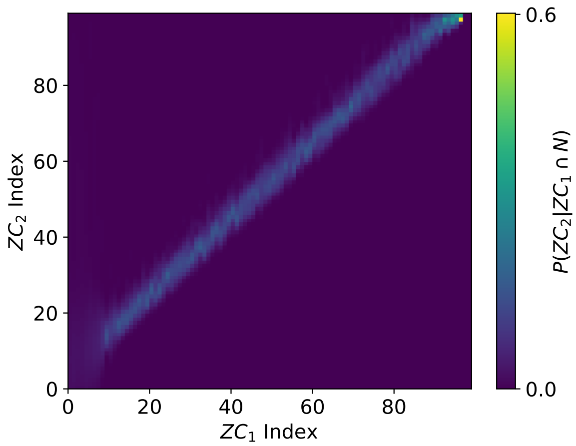
Finally, with we compute the probability of a combination of zero crossings being generated by a step:

$$\begin{aligned} &P(S|zc_1 \cap zc_2) \\ &\quad = \frac{P(zc_2|S \cap zc_1)P(zc_1|S)P(S)}{\begin{array}{r} P(zc_2|S \cap zc_1)P(zc_1|S)P(S) \\ + P(zc_2|N \cap zc_1)P(zc_1|N)P(N) \end{array}} \end{aligned} \tag{6}$$

This probability is computed for every combination of zero crossings in the training set, generating the final matrix of probabilities as shown in Fig. 2. Some values from the PMFs that are used to compute (6) are very small. These double precision floating point values are set to zero if they fall below



(a) Steps



(b) Noise

Figure 1: Conditional probability for second zero crossings. a) shows probability of a zero crossing occurring at a certain index after an initial zero crossing generated by a step, while b) shows the probability of a second zero crossing generated by noise

a threshold ε. This ε is the lower limit ensuring $1.0+\varepsilon \neq 1.0$. Resulting from this, a sharp drop of the probability is visible in Fig. 2 for the upper bounds of zc_2. This has no significant effect on the performance of the step detection method.

A window of sensor data can now be evaluated by applying the above-mentioned pre-processing and determining the first two zero crossings for the data. A look-up operation on the pre-computed matrix shown in Fig. 2 returns the probability if the examined window of data is belonging to a step event.

In order to detect single steps, a sliding window is propagated in one sample steps along a sequence of data. In each window propagation, the first and second zero crossing zc_1 and zc_2 is determined to find $P(S|zc_1 \cap zc_2)$ as described in (6). A step is detected if $P(S|zc_1 \cap zc_2)$ exceeds a threshold th_{step} for a certain count of consecutive windows. These thresholds

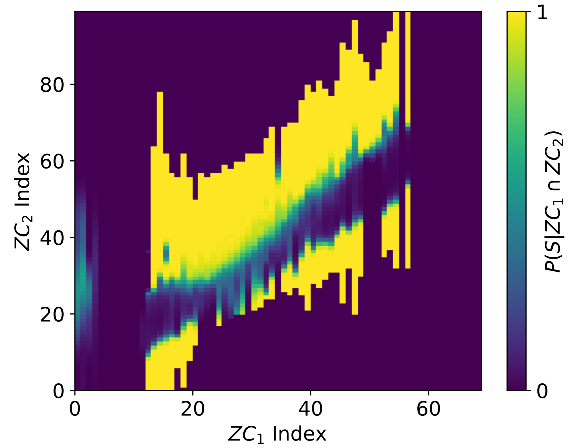


Figure 2: Step Probability based on the index of two consecutive zero crossings

are found by maximizing the detector performance in the training set and will be detailed in Section VI. The exact size of the sliding window has little effect on the accuracy or complexity of the step detector. Only new samples after a window propagation need to be checked for a zero crossing while the known zero crossing indices are simply incremented by the window propagation step size. Finally a new look-up in the probability matrix is done, the value is compared against th_{spec} and the counter is either incremented or reset if th_{spec} is not exceeded anymore.

A detailed evaluation of this approach is presented in the following sections.

VI. Experimental setup

We verify the proposed method in a two-fold way. First, the ability to discern walking from random noise is validated. Second, the actual ability to determine the correct count of steps from a sequence of true walking is determined.

The basis of the evaluated step detector is a matrix of predetermined probabilities as outlined in the previous section. The same matrix is used for both experiments. It is generated using a training set of 7000 examples, containing each: A sequence of data containing exactly one true step and a sequence of Gaussian noise. One noise sequence consist 100 samples with constant standard deviation σ_n. For comparison: The training steps consist on average of 52.83 acceleration samples with a standard deviation of 0.4 g. The longest step spans 106 samples. The probability threshold for a window of data to hint at a step th_{step} is tested for values from 0.05 to 0.95 while the number of consecutive windows exceeding th_{step} needed to actually count a step is fixed at 5. A range of values for σ_n from 0.1 to 1.0 g was evaluated using the same constant σ_n for the generation of each probability matrix as well as the corresponding test sequences for both experiments. The level shift δ is fixed at 0.2 g.

The evaluation of walking detection uses an evaluation set of 3000 sequences of equal length for each: true walking and noise. The walking sequences are extracted as non-overlapping windows of 100 samples from previously unused walking examples containing multiple steps. The noise examples are generated with the same parameters as in the training set. An example is counted as walking if at least one step is detected in it. However, in order to guarantee that zero crossings which occur near the end of a sequence are evaluated across the whole window length of the step detector, the evaluated sequence is extended using the value of its last entry. We compare the walking detection using zero crossings with a method using the actual spectral markers of the FFT of the window of data, similar to [2]. However, we were not able to replicate the exact spectral properties that were observed there. Instead, we are classifying a signal as *walking* if the energy of the peak between 1 Hz and 4 Hz exceeds a multiple of the average energy up to 10 Hz. The performance for several settings of this multiplier th_{spec} will be discussed in the next section.

The evaluation of step counting is straight forward. Uninterrupted sequences of true walking are extracted from the training set. Again, the ends of these sequences are extended to ensure that zero crossings near the end of a sequence are evaluated across the whole step detector window length. The step detector is applied to the walking sequence and the resulting step count is compared to the true step count. The test set consist of 8700 true steps from 281 participants.

VII. Experimental results

For the spectral classifier (SP), a high th_{spec} results in better sensitivity and specificity, while the opposite applies to precision. A equilibrium of 0.89 for all three measures is reached at a th_{spec} of about 5.90. The performance measures are constant for every setting of σ_n.

In contrast, the performance of the zero crossing approach (ZC) depends highly on σ_n and the choice of th_{step}. At σ_n of 0.45 g and lower, ZC performs better than SP for adequate choices of th_{step} that result in an equilibrium between the three performance measures. However, at higher σ_n, th_{step} also needs to be higher to maintain superior performance. A higher th_{step} lessens the sensitivity, while precision and specificity improve.

The receiver operating characteristic (ROC) for ZC and SP at σ_n of 0.4 g is shown in Fig. 3. The curve for ZC shows superior performance for sensitivities above 0.8. For th_{step} at 0.5, sensitivity, precision and specificity are at 0.97, 0.77 and 0.70, indicating a sufficient capability to count actual steps but also a tendency for false positives when presented with noise. A more balanced performance is reached at 0.8 for th_{step} with sensitivity, precision and specificity at 0.93, 0.89 and 0.89.

The second evaluation concerns only true walking examples. The absolute average error in the step count is 4.99 % for a th_{step} of 0.5 and 8.04 % for a th_{step} of 0.8. The result are competitive with commercially available pedometers [22], [23]. Fig. 4 shows the distribution of step count errors for the evaluated participants and a th_{step} of 0.5. Here, there is no error for 98 out of 281 participants. Unsurprisingly, a th_{step} of 0.8 shows a tendency to under-count. Fig. 5 shows the determined step probability over time for an example of walking.

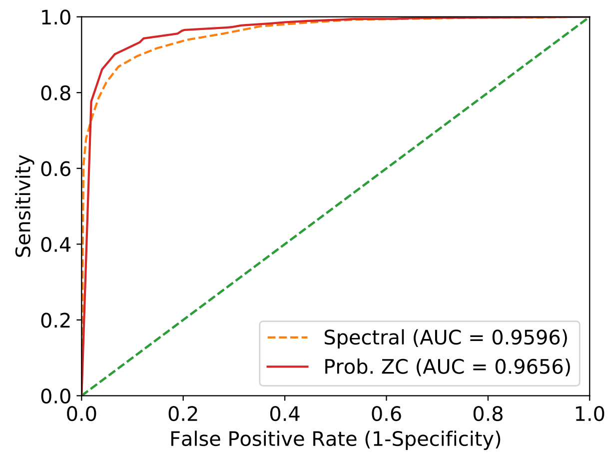


Figure 3: ROC of the SP and ZC classifier with σ_n of 0.4 g, including the area under curve (AUC)

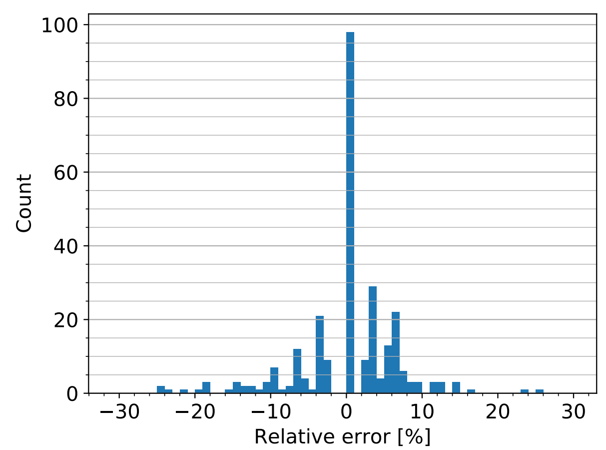


Figure 4: Histogram of step count error for 281 participants at a th_{step} of 0.5 and σ_n of 0.4 g.

VIII. Conclusion

In this paper, we describe a Bayesian estimator for step counting based on zero crossing interval analysis and demonstrate its capability using data of the OU-ISIR Gait Database. We demonstrate the statistical discrimination of zero crossing vectors, differentiating between taking a step and random noise. The extraction of the zero crossing interval vector requires no explicit AD-conversion and the step probability is retrieved at runtime from a pre-computed matrix. Additionally and in contrast to a FFT based spectral approach,

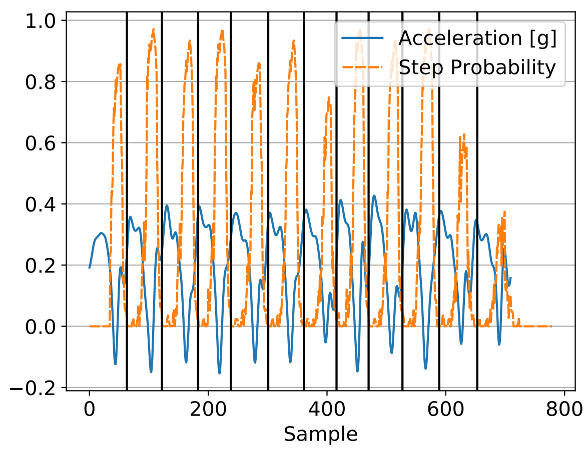


Figure 5: A walking sequence with the step probability over time, including one missed step at the end. Individual steps are separated by vertical lines. Trained using a σ_n of 0.4 g

the evaluation of sliding windows of data is done by adding and evaluating new values without explicit re-computation of the window as a whole. This fact leads to an extremely low computational complexity and thus a potentially low power consumption. We compare the results with a spectral estimator and show that the presented method performs competitively to the state of the art. The quantity of possible energy savings has to be shown in future work.

A fundamental limitation of this analysis is the synthetic nature of the noise counterexamples. Further analysis with data of actual scenarios of daily life and work is needed. It should also be noted that the the participants of the dataset are highly diverse, with ages ranging from toddlers to seniors. We suspect that certain age groups show differing walking motion. How this affects step detection in detail will be investigated in the future as well.

Acknowledgements

This work has been achieved in the European ITEA project ”OPTimised Industrial IoT and Distributed Control Platform for Manufacturing and Material Handling (OPTIMUM) and has been funded by the German Federal Ministry of Education and Research (BMBF) under reference number 01IS17027. We want to thank all partners in the OPTIMUM project for the stimulating discussions and their contributions to the project. All project partners can be found on https://itea3.org/project/optimum.html.

Furthermore, we would like to thank Osaka University for the kind provision of the dataset. Without it, this work would have not been possible.

References

[1] A. Abadleh, E. Al-Hawari, E. Alkafaween, and H. Al-Sawalqah, “Step detection algorithm for accurate distance estimation using dynamic step length,” *Proceedings - 18th IEEE International Conference on Mobile Data Management, MDM 2017*, pp. 324–328, 2017.

[2] X. Kang, B. Huang, and G. Qi, “A novel walking detection and step counting algorithm using unconstrained smartphones,” *Sensors (Switzerland)*, vol. 18, no. 1, jan 2018.

[3] A. S. Zandi, R. Tafreshi, M. Javidan, and G. A. Dumont, “Predicting temporal lobe epileptic seizures based on zero-crossing interval analysis in scalp EEG,” *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10*, no. October, pp. 5537–5540, 2010.

[4] L. R. Rabiner and M. R. Sambur, “An Algorithm for Determining the Endpoints of Isolated Utterances,” *Bell System Technical Journal*, vol. 54, no. 2, pp. 297–315, 1975.

[5] B. Kedem, “Spectral analysis and discrimination by zero-crossings,” *Proceedings of the IEEE*, vol. 74, no. 11, pp. 1477–1493, 1986.

[6] Y.-P. Wang, J. Chen, Q. Wu, and K. R. Castleman, “Fast frequency estimation by zero crossings of differential spline wavelet transform,” *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 8, p. 624376, 2005.

[7] M. Alzantot and M. Youssef, “UPTIME: Ubiquitous pedestrian tracking using mobile phones,” *IEEE Wireless Communications and Networking Conference, WCNC*, pp. 3204–3209, 2012.

[8] W. Y. Hu, J. L. Lu, S. Jiang, W. Shu, and M. Y. Wu, “WiBEST: A hybrid personal indoor positioning system,” *IEEE Wireless Communications and Networking Conference, WCNC*, pp. 2149–2154, 2013.

[9] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, “Zee: Zero-effort crowdsourcing for indoor localization,” *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, pp. 293–304, 2012.

[10] J. Seo, Y. Chiang, T. H. Laine, and A. M. Khan, “Step counting on smartphones using advanced zero-crossing and linear regression,” *ACM IMCOM 2015 - Proceedings*, no. November 2017, 2015.

[11] S. Ayub, X. Zhou, S. Honary, A. Bahraminasab, and B. Honary, “Indoor pedestrian displacement estimation using smart phone inertial sensors,” *International Journal of Innovative Computing and Applications*, vol. 4, no. 1, pp. 35–42, 2012.

[12] W. Chen, Z. Fu, R. Chen, Y. Chen, O. Andrei, T. Kroger, and J. Wang, “An integrated gps and multi-sensor pedestrian positioning system for 3d urban navigation,” in *2009 Joint Urban Remote Sensing Event*. IEEE, 2009, pp. 1–6.

[13] K. Tejaswi, S. Rao, T. Nair, and A. Prasad, “Gpu accelerated automated feature extraction from satellite images,” *International Journal of Distributed and Parallel Systems*, vol. 4, 04 2013.

[14] G. Rodríguez, F. Casado, R. Iglesias, C. Regueiro, and A. Nieto, “Robust step counting for inertial navigation with mobile phones,” *Sensors*, vol. 18, no. 9, p. 3157, 2018.

[15] S. Vandermeeren, S. Van de Velde, H. Bruneel, and H. Steendam, “A feature ranking and selection algorithm for machine learning-based step counters,” *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3255–3265, 2018.

[16] V. Pham, D. Nguyen, N. Dang, H. Pham, V. Tran, K. Sandrasegaran, and D.-T. Tran, “Highly accurate step counting at various walking states using low-cost inertial measurement unit support indoor positioning system,” *Sensors*, vol. 18, no. 10, p. 3186, 2018.

[17] H.-h. Lee, S. Choi, and M.-j. Lee, “Step detection robust against the dynamics of smartphones,” *Sensors*, vol. 15, no. 10, pp. 27 230–27 250, 2015.

[18] P. Davidson and R. Piché, “A survey of selected indoor positioning methods for smartphones,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1347–1370, 2016.

[19] L. E. Díez, A. Bahillo, J. Otegui, and T. Otim, “Step length estimation methods based on inertial sensors: a review,” *IEEE Sensors Journal*, vol. 18, no. 17, pp. 6908–6926, 2018.

[20] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, “Similar gait action recognition using an inertial sensor,” *Pattern Recognition*, vol. 48, no. 4, pp. 1289–1301, 2015.

[21] R. Khusainov, D. Azzi, I. E. Achumba, and S. D. Bersch, “Real-time human ambulation, activity, and physiological monitoring: Taxonomy of issues, techniques, applications, challenges and limitations,” *Sensors*, vol. 13, no. 10, pp. 12 852–12 902, 2013.

[22] B. Presset, B. Laurenczy, D. Malatesta, and J. Barral, “Accuracy of a smartphone pedometer application according to different speeds and mobile phone locations in a laboratory context,” *Journal of Exercise Science & Fitness*, vol. 16, no. 2, pp. 43–48, 2018.

[23] F. Ehrler, C. Weber, and C. Lovis, “Influence of pedometer position on pedometer accuracy at various walking speeds: a comparative study,” *Journal of medical Internet research*, vol. 18, no. 10, p. e268, 2016.