

Research Article

Interrupt-Based Step-Counting to Extend Battery Life in an Activity Monitor

Seung Young Kim and Gu-In Kwon

School of Computer and Information Engineering, Inha University, Incheon 402-751, Republic of Korea

Correspondence should be addressed to Gu-In Kwon; gikwon@inha.ac.kr

Received 11 May 2015; Revised 16 June 2015; Accepted 23 June 2015

Academic Editor: Toshiyo Tamura

Copyright © 2016 S. Y. Kim and G.-I. Kwon. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Most activity monitors use an accelerometer and gyroscope sensors to characterize the wearer's physical activity. The monitor measures the motion by polling an accelerometer or gyroscope sensor or both every 20–30 ms and frequent polling affects the battery life of a wearable device. One of the key features of a commercial daily-activity monitoring device is longer battery life so that the user can keep track of his or her activity for a week or so without recharging the battery of the monitoring device. Many low-power approaches for a step-counting system use either a polling-based algorithm or an interrupt-based algorithm. In this paper, we propose a novel approach that uses the tap interrupt of an accelerometer to count steps while consuming low power. We compared the accuracy of step counting and measured system-level power consumption to a periodic sensor-reading algorithm. Our tap interrupt approach shows a battery lifetime that is 175% longer than that of a 30 ms polling method without gyroscope. The battery lifetime can be extended up to 863% with a gyroscope by putting both the processor and the gyroscope into sleep state during the majority of operation time.

1. Introduction

The growing interest in health has motivated the development of smart accessories that monitor the wearers' physical activities, including walking, running, sitting down, and lying in bed. Recently there has been a lot of research on method of counting steps using a three-axis accelerometer in association with different algorithms, including these found in commercially produced pedometers [1].

An activity monitor or digital pedometer typically reads an accelerometer sensor periodically, at a frequency of 30–50 Hz to measure the number of steps taken in walking or running. In a polling algorithm of this sort, the monitoring processor wakes up from its sleep state every 20–30 ms, reads the sensor data, and then runs the step-counting algorithm. This approach is 99.5% accurate for walking speeds over 90 m/min, which is usually considered satisfactory. However, polling consumes more power than the alternative event-based interrupt approach [1], in which the system only wakes from its sleep state when the acceleration crosses some

threshold. Research shows the acceleration change in human walking is between 0.4 and 0.8 g along the vertical axis [2].

Because an activity monitor or pedometer is usually worn on the wrist, on ankle, or on a belt around the waist, the system needs to have a form factor similar to that of a watch. Nevertheless, users expect a battery life of a week at least. For example, the electrical specification of the Misfit Shine, which is a highly optimized commercial activity monitor, indicates that a 3 V CR2032 battery, with a capacity of approximately 200 mAh, lasts three months. To obtain this kind of battery life, a system needs to stay in its sleep state for as long as possible, and the all electrical components have to be low-power types. For instance, the MSP430 or STM32L MCUS only draws a few nA in sleep mode [3].

In this paper, we propose a low-power approach to step-counting which uses the tap interrupt which is provided by several accelerometer chips, including the most popular Analog Devices ADXL345 and Freescale MMA8451. This interrupt is triggered by an acceleration exceeding a preprogrammed value. To the best of our knowledge, a tap interrupt

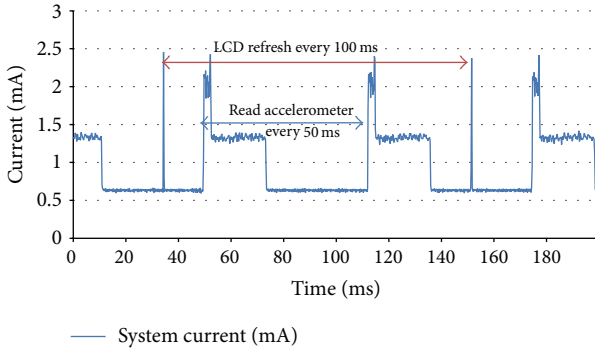


FIGURE 1: System current when polling at 20 Hz.

function has not previously been used for step-counting. We have compared the power consumption of a system running our algorithm and running a polling algorithm. We measured average power usage with and without gyroscope, which can draw a lot of power. The accuracy of our tap interrupt algorithm ranges between 83.5% and 90.25%, depending on the weight of the users, but the battery life is extended by factors ranging up to 8.6 when compared with polling. Comparing these figures with the performance of commercial pedometers suggests that our algorithm is suitable for devices in which battery life is paramount.

2. Methods

2.1. Step-Counting by Polling an Accelerometer. A polling algorithm uses a timer-based routine to wake the accelerometer sensor to read accelerations and then return it to sleep state to save power [4]. We experimented with a polling algorithm running on 20 Hz and measured the system power consumption. As shown in Figure 1, the system wakes at every 50 ms and reads the sensor which produces a peak in power consumption; then it processes the sensor data for about 20 ms which also uses significant power consumption. The number of steps is displayed on an LCD every 100 ms.

The power consumption in sleep state P_s is 1.91 mW (P_s); the power P_a used for reading accelerometer is 6.04 mW; and the processor running the polling algorithm uses a power P_p of 3.95 mW when it wakes up at a polling frequency of f . The power consumption of the system P_{system} can be expressed as follows:

$$P_{\text{system}} = f \times V \times \left\{ (P_a \times T_a) + (P_p \times T_p) + \left(P_s \times \left(\frac{1}{f} - (T_a + T_p) \right) \right) \right\}, \quad (1)$$

where P_a is power consumption during reading of the accelerometer, P_p is power consumption running the step-counting algorithm, P_s is power consumption in sleep state, T_a is processing time required to read the reading accelerometer, T_p is processing time required by the polling algorithm, f is frequency of polling, and V is supply voltage.

TABLE 1: Averages value of power consumption and processing times.

P_a	P_p	P_s	T_a	T_p
6.04 mW	3.95 mW	1.91 mW	8.97 mW	62.19 mW

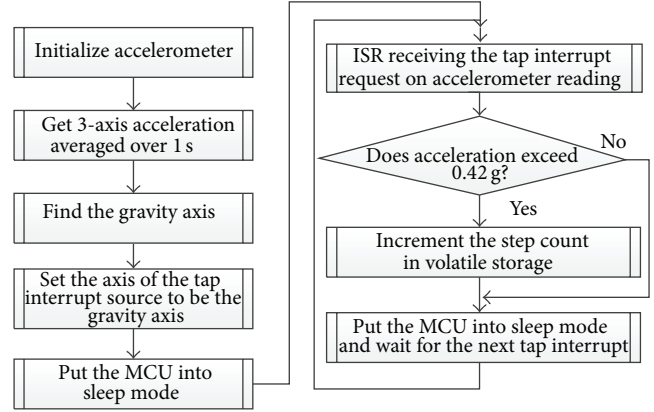


FIGURE 2: Algorithm for counting steps using the tap interrupt facility of the ADXL345 accelerometer.

Table 1 shows the average power consumption of system for 1 second.

It is clear from (1) that actually energy can be saved by running the frequency at which the step-counting algorithm is waken up. But accuracy declined at frequency below 20 Hz [5], even though a human cannot walk or run at over 5 Hz. This motivates the development of an interrupt-based algorithm.

2.2. Step-Counting Using a Tap Interrupt. The tap interrupt provided by the ADXL345 accelerometer chip can reduce the processing time required for counting steps, hence saving power.

Figure 2 shows our step-counting algorithm which uses this interrupt. An interrupt is triggered by acceleration along one of the accelerometer three fixed axes which exceed a specific threshold. To avoid unexpected interrupt, our algorithm finds the gravity axis among three axes by averaging the acceleration along each axis over 1 second and selects the axis whose average is around -1.0 g for interrupt source axis to trigger the tap interrupt. The interrupt service routine (ISR) handling the tap interrupt checks both the tap interrupt flag and the data-available flag before it decides to count a step. And this interrupt can be triggered by accelerometer even in its sleep mode and we can put processor also in its sleep to save system power.

Figure 3 shows the power consumption of the system over 1 second using the tap interrupt technique: one step is taken and one power peak occurs. During the resulting wake-up period the system consumes 5.92 mW for 12.6 ms and most of the time the system stays in sleep state to save the power. We will present results showing that the tap interrupt algorithm does save system-level power and assess its accuracy in Section 3.

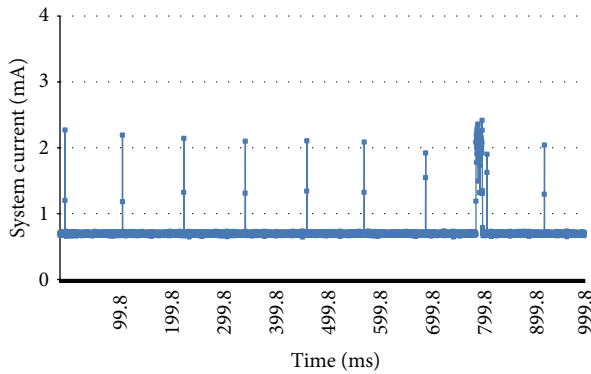


FIGURE 3: Current consumption of the system running the tap interrupt algorithm for 1 second.

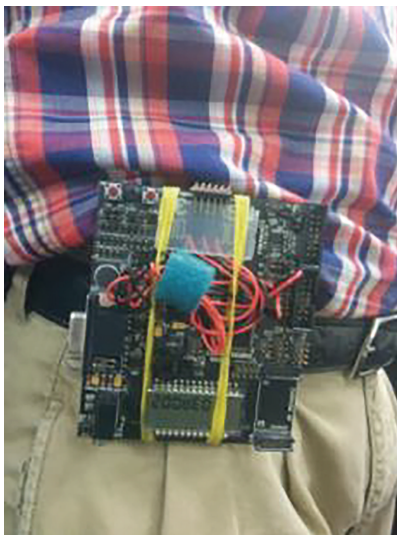


FIGURE 4: Experimental pedometer and battery worn on a belt.

2.3. System Implementation. We implemented our step-counting algorithm in a pedometer, shown in Figure 4, constructed from a Texas Instruments MSP430F4618 experimenter's board, populated with an ADXL345 and a ITG3200 gyroscope. Both sensors are connected to the I2C bus on MSP430F4618, and this I2C bus was configured to run at 100 kHz to save power. All unnecessary components were removed from the board to avoid spurious power consumption.

The ADXL345 was configured to run at a sampling rate of 200 Hz, at which frequency it drew $145 \mu\text{A}$ [6]. The ITG3200 was put into its sleep state immediately after initialization, where it draws just $6 \mu\text{A}$; when operating normally it draws 6.5 mA [7], and this power consumption is too great for a 200 mAh battery-powered system. The ADXL345 needs to be operating normally so that it can wake the MCU up by means of an interrupt. The MCU is put into LPM4 mode, in which it typically draws $0.3 \mu\text{A}$, against $400 \mu\text{A}$ in active mode at room temperature [8]. To compare the power consumed by a polling algorithm and our tap interrupt algorithm, both algorithms were implemented in the single source code

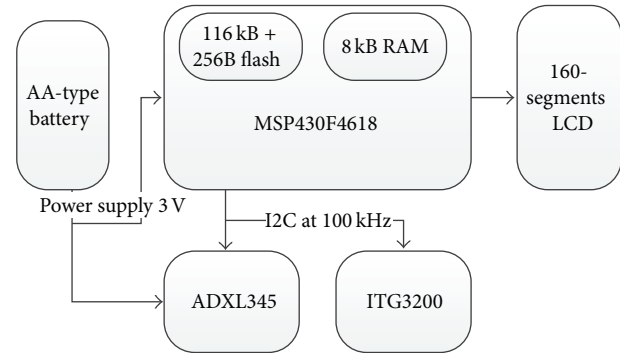


FIGURE 5: Block diagram of an experimental pedometer implemented on an MSP430F4618 experimenter's board, with an accelerometer and a gyroscope.

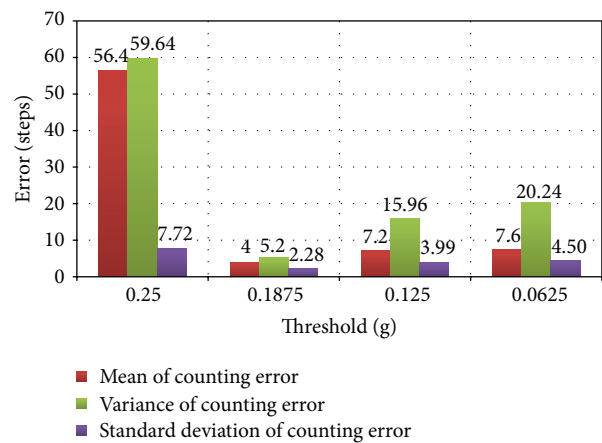


FIGURE 6: Mean and standard deviation of the errors in step count for different acceleration thresholds for the tap interrupt.

file, and one algorithm was selected at compilation time. The step count is displayed by an LCD driven by the LCD controller built into the MSP430F4618. Figure 5 shows the block diagram of this system.

3. Experimental Results

3.1. Step-Counting Accuracy. The threshold value of acceleration that triggers a tap interrupt needs to be carefully selected to balance accuracy against the powers saved by the MCU in its sleep state. We searched for an optimal acceleration threshold by measuring the error in step-counting using thresholds between 0.0625 g and 0.25 g (Figure 6).

The procedure involved 10 trials of 40 steps by 65 kg adult. The smallest error corresponds to an acceleration threshold of 0.187 g, and this threshold uses further experiments. All experiments were approved by the Inha University Ethics Committee.

The accuracy of our algorithm was compared with that of an Omron HJ303 pedometer (which uses triaxis technology), the Runtastic (the most downloaded Android pedometer app), and Pedometer (implemented by Levente Bagi based on a public step-counting algorithm). These tests were

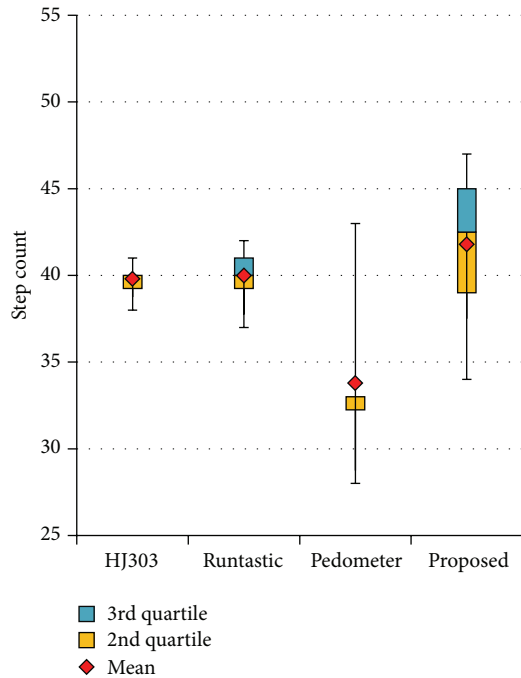


FIGURE 7: “Box and whiskers” plot showing the mean, standard deviation, and quartile of results from four pedometers.

conducted at walking speed of 80 m/min, because many commercial pedometers are accurate at this speed [9]. Each test involved 10 trials of 40 steps by each of two adults, weighing 65 kg and 72 kg. Because accuracy can be affected by the position in which a pedometer is worn, all the devices were attached to the wear’s belt above their right hip.

The result for the 65 kg walks is shown as a “box and whiskers” plot in Figure 7 of the 10 time trials.

The mean step counts (and standard deviations) for our device, the HJ303, the Runtastic, and the Android pedometer app were 41.8 (4.46), 39.8 (0.92), 40.0 (1.63), and 33.8 (3.90), respectively. Our pedometer is less accurate than the commercial pedometers but performs considerably better than the Android app.

We conducted a further experiment on our devices, involving walkers with a wider range of weights, again involving 10 trials of 40 steps. The accuracy of the results varied between 83.5% and 90.25% as shown in Table 2. Schneider et al. [10] suggested an error of up to 10% is acceptable when a pedometer is used to measure the wearer’s activity over long periods. This is exactly the sort of application in which a long battery life is essential.

3.2. Energy Consumption. To measure the system power consumption, we connected our pedometer to a Monson Solutions Power Monitor, connected in turn to a PC running Power Tool 4.0.4. The output voltage of the monitor was set to 3.0 V, and the current across a 0.56 Ω resistor was measured every 200 μ s [11]. We performed tests with the tap interrupt algorithm and the polling algorithm, with the frequency of polling set to 20, 32, and 100 Hz.

TABLE 2: Accuracy of our pedometers with walkers of different weights.

Weight (kg)	35	44	63	65	65	72	94
Accuracy (%)	87.5	83.5	90.25	90.5	90.0	83.75	88.0

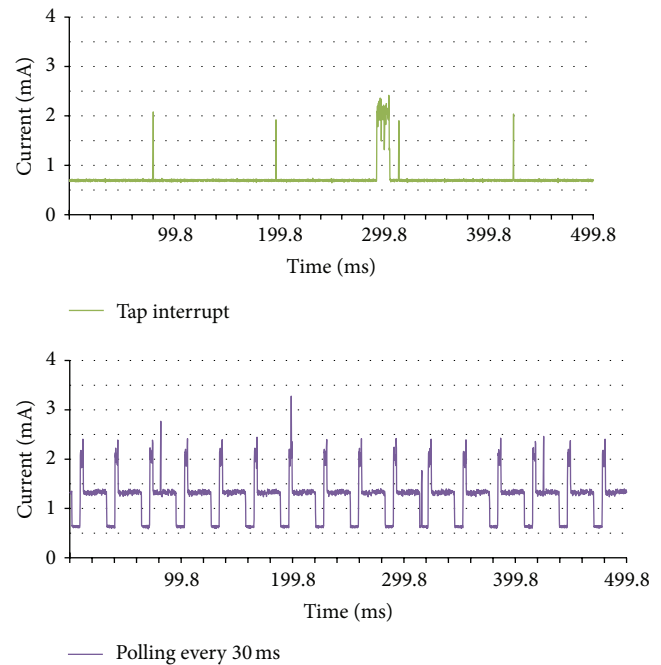


FIGURE 8: Power drawn by the system over 500 ms, running tap interrupt and polling algorithms. The results at 100 ms intervals correspond to LCD refresh, and the wide peak in the upper chart is the tap interrupt service routine.

Figure 8 shows the patterns of current consumption over 500 ms for the tap interrupt algorithm and the polling algorithm running at 32 Hz. The tap interrupt produces peak of 1.98 mA while the polling algorithm has 16 reads of 1.732 mA.

3.3. Gyroscope Power Consumption. The accelerometer alone is sufficient to count steps, but the monitoring of other activities requires a gyroscope to be used in addition. Brajdic and Harle evaluated several more algorithms for waking detection and step-counting using accelerometer, gyroscope, and both [12]. This increases power consumption by a factor of about 5 times. Figure 9 shows the pattern of current consumption when running the tap interrupt algorithm with and without the gyroscope. The gyroscope is only waken from sleep state when an interrupt is received from the accelerometer. The ETHOS system [3] draws 7.4 mA where the gyroscope is enabled, and the estimated battery life of the 4.2 V 200 mAh Li-ion battery is 27 hours. Table 3 compares the battery life for the polling and the interrupt algorithm with and without gyroscope. Our tap interrupt algorithm extends battery life by 175%, compared with polling without

TABLE 3: Power consumptions and expected battery life using the polling and tap interrupt algorithms.

Supply voltage	Measure time (ms)	Consumed energy (uAh)	Average current (mA)	Battery life (minutes)
2.99 V				
Polling at 30 ms	60936	20.95	1.24	9690.6
Polling at 50 ms	60519	15.75	0.94	12808.8
Tap interrupt	60012	11.92	0.72	16775.4
Polling at 30 ms with gyroscope	60885	18.99	6.35	1890.6
Tap interrupt with gyroscope	61083	12.47	0.74	16324.8

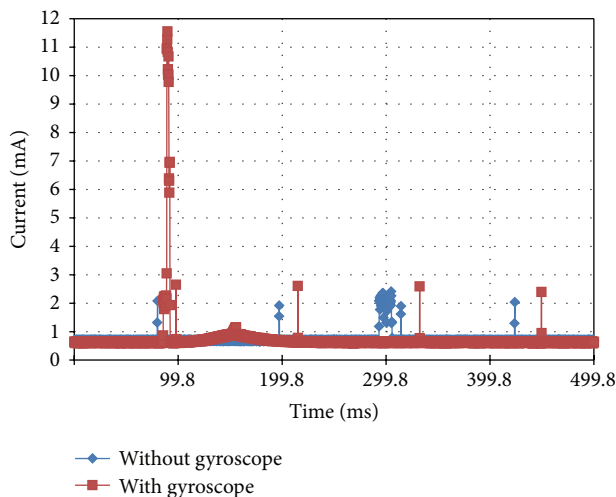


FIGURE 9: Current drain over 500 ms for the tap interrupt algorithm, with and without a gyroscope. The ITG3200 gyroscope draws approximately 6.5 mA in active mode, which results in a large difference in power consumption between the polling and interrupt algorithms.

a gyroscope. With the gyroscope, the battery life can be extended by up to 863% by putting both the processor and gyroscope into sleep state for most of the time.

4. Discussion

This paper proposes a new approach for a low-power step-counting algorithm by using a tap interrupt algorithm and we conducted power profiling to analyze power consumption of popular polling algorithm.

By reducing wake-up frequency from the sleep state and shortening process time at every wake-up, we can reduce the average power consumption. By using the threshold interrupt of the accelerometer, the tap interrupt algorithm can contribute to a longer battery life for activity monitoring devices. A minimum of 30 minutes of daily physical activity, such as walking, is now considered necessary to maintain fitness

[13]. Patients with conditions such as obstructive pulmonary disease may require long-term activity monitoring, but this does not need to be very accurate. In future work, we plan to investigate whether the tap interrupt can be used to detect unexpected fall for elders or to recognize posture in humans and animals in long-term application in which frequent battery recharging is inconvenient [14].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (no. 2015R1A2A2A01003501). This work was supported by Inha University.

References

- [1] C. G. Ryan, P. M. Grant, W. W. Tigbe, and M. H. Granat, "The validity and reliability of a novel activity monitor as a measure of walking," *British Journal of Sports Medicine*, vol. 40, no. 9, pp. 779–784, 2006.
- [2] R. Cross, "Standing, walking, running, and jumping on a force plate," *American Journal of Physics*, vol. 67, no. 4, pp. 304–309, 1999.
- [3] H. Harms, O. Amft, R. Winkler, J. Schumm, M. Kusserow, and G. Troester, "ETHOS: miniature orientation sensor for wearable human motion analysis," in *Proceedings of the 9th IEEE Sensors Conference (SENSORS '10)*, pp. 1037–1042, IEEE, Kona, Hawaii, USA, November 2010.
- [4] A. Krause, M. Ihmig, E. Rankin et al., "Trading off prediction accuracy and power consumption for context-aware wearable computing," in *Proceedings of the IEEE International Symposium on Wearable Computers (ISWC '05)*, pp. 20–26, October 2005.
- [5] H. Junker, P. Lukowicz, and G. Tröster, "Sampling frequency, signal resolution and the accuracy of wearable context recognition systems," in *Proceedings of the 8th International Symposium on Wearable Computers (ISWC '04)*, pp. 176–177, November 2004.
- [6] N. Zhao, "Full-featured pedometer design realized with 3-axis digital accelerometer," *Analog Dialogue*, p. 44–06, 2010.
- [7] K. Leuenberger and R. Gassert, "Low-power sensor module for long-term activity monitoring," in *Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '11)*, pp. 2237–2241, September 2011.
- [8] T. Hornos, *Wireless ECG/EEG with the MSP430 microcontroller [M.S. thesis]*, University of Glasgow, Glasgow, UK, 2009.
- [9] S. E. Crouter, P. L. Schneider, M. Karabulut, and D. R. Bassett Jr., "Validity of 10 electronic pedometers for measuring steps, distance, and energy cost," *Medicine & Science in Sports & Exercise*, vol. 35, no. 8, pp. 1455–1460, 2003.
- [10] P. L. Schneider, S. E. Crouter, and D. R. Bassett Jr., "Pedometer measures of free-living physical activity: comparison of 13 models," *Medicine and Science in Sports and Exercise*, vol. 36, no. 2, pp. 331–335, 2004.

- [11] L. Zhang, B. Tiwana, R. P. Dick et al., "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proceedings of the 8th IEEE/ACM International Conference on Hardware/Software-Co-Design and System Synthesis (CODES+ISSS '10)*, pp. 105–114, October 2010.
- [12] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*, pp. 225–234, September 2013.
- [13] R. R. Pate, M. Pratt, S. N. Blair et al., "Physical activity and public health. A recommendation from the Centers for Disease Control and Prevention and the American College of Sports Medicine," *The Journal of the American Medical Association*, vol. 273, no. 5, pp. 402–407, 1995.
- [14] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, "Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information," in *Proceedings of the 6th International Workshop on Wearable and Implantable Body Sensor Networks (BSN '09)*, pp. 138–143, IEEE, Berkeley, Calif, USA, June 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

