A Low-power Fall Detection Algorithm Based on Triaxial Acceleration and Barometric Pressure

Changhong Wang, Michael R. Narayanan, *Member, IEEE*, Stephen R. Lord, Stephen J. Redmond, *Senior Member, IEEE*, Nigel H. Lovell, *Fellow, IEEE*

***Abstract*—This paper proposes a low-power fall detection algorithm based on triaxial accelerometry and barometric pressure signals. The algorithm dynamically adjusts the sampling rate of an accelerometer and manages data transmission between sensors and a controller to reduce power consumption. The results of simulation show that the sensitivity and specificity of the proposed fall detection algorithm are both above 96% when applied to a previously collected dataset comprising 20 young actors performing a combination of simulated falls and activities of daily living. This level of performance can be achieved despite a 10.9% reduction in power consumption.**

I. Introduction

Falls and injuries caused by falls are a major threat to the health of older people. A recent study shows that about 40% of older people aged over 70 years fall at least once per year, and half of these fall two or more times per year [1]. Thus, there is an increasing impetus to develop automatic fall detectors to detect fall events in real-time and send an alarm to a medical care service provider for immediate assistance. However, fall events happen very rarely relative to the duration of the monitoring time, and the battery life of a wearable fall detector is limited. If the fall detector is not efficient in the way it utilizes power, its battery life will be short and users will have to frequently recharge the device. Thus, one important development trend for fall detectors is to incorporate low-power technologies to prolong the battery lifetime.

Currently, low-power technologies have been widely applied to wearable medical devices, providing energy savings through improved design of sensing [2] and processing [3] hardware, data transmission protocols, and software optimization. Coding efficiencies include selecting features with low computational complexity using machine learning [4], and compressing signals to reduce the amount of data transmitted by wireless communication [5].

Despite general work in the area, there are only very limited reports of these power-saving approaches being applied specifically to the field of fall detection [6], [7]. Such reports are mainly focused on reducing the power consumption of wireless transmission by optimizing wireless protocols or managing the wireless communication module based on whether a fall happens or not. To the best of our knowledge, no paper has focused on energy efficiency from the perspectives of sensing and processing kinematic signals generate by fall detectors. In this paper, we propose a low-power fall detection algorithm which dynamically adjusts the sampling rate of the accelerometer and manages the data transmission between the sensors and the controller.

II. Low-power Fall Detection Algorithm

A. Dataset

The dataset used to design and evaluate the proposed algorithm is based on that described previously by our group in [8]. These data were acquired using a triaxial accelerometer and a barometric pressure sensor. The sampling rate of the acceleration data was 40 Hz, and the sampling rate of the barometric pressure data was 1.8 Hz. The noise in the barometric pressure measurement had a root-mean-square error (RMSE) of 1.5 Pa, which equates to a resolution in terms of altitude of about 12.5 cm. The data were obtained from 20 younger actors performing a set of simulated falls and activities of daily living (ADL) (Table I).

Using a 40 Hz sampling rate for the acceleration data is sufficient to reliably capture the impact of a fall [9], but this will result in higher power consumption if the accelerometer continuously samples at this rate. It is ideal for the accelerometer to sample the acceleration at a high rate only when a fall event happens, and to sample the acceleration at a low rate to save power at other times. However, any lower sampling rate must guarantee the accelerometer does not miss features of the falling phase [10] and thus the algorithm must have the capacity to adjust the sampling rate in a timely manner from low frequency to high frequency for the anticipated fall impact phase. The duration of the falling phase of the unbroken fall is expected to be 0.4-0.8s, so the acceleration signal with sampling rate of 5 Hz can provide sufficient information to describe this phase according to the Nyquist–Shannon sampling theorem. For the purposes of the simulations described herein, the acceleration signal with a 5 Hz sampling rate is derived by down-sampling the original acceleration signal with the sampling rate of 40Hz, and the algorithm will be evaluated on these two kinds of acceleration data.

B. Fall Detection Algorithm

The proposed algorithm is an augmented version of the algorithm in [8] based on a digital accelerometer and a barometric pressure sensor, each with their own internal buffer. The algorithm has a parallel structure with two threads.

This research was supported under the Australian Research Council's Linkage Projects funding scheme (project number LP130100159).

C. Wang, M. R. Narayanan, S. J. Redmond, and N. H. Lovell are with the Graduate School of Biomedical Engineering, UNSW Australia. (e-mail: n.lovell@unsw.edu.au)

S. R. Lord is with Neuroscience Research Australia. (e-mail: s.lord@unsw.edu.au)

978-1-4244-7929-0/14/$26.00 ©2014 IEEE

TABLE I. TRIALS CONTAINING EIGHT SIMULATED FALLS AND EIGHT ACTIVITIES OF DAILY LIVING.

Sequence	Category	Instructions
1-3	Fall	Forward, backward, lateral fall, ending lying
4	Fall	Forward fall, ending active lying
5	Fall	Forward fall with attempt to break the fall
6	Fall	Resting against a wall, then sliding vertically down to the end in the sitting position
7-8	Fall with recovery	Forward fall, recovery and walking or standing
9-10	No fall	Sitting on or collpsing into a chair
11	No fall	Climbing into bed
12	No fall	Jump in vertical direction
13	No fall	Pick up something from the floor
14	No fall	Bend down and doing own shoelaces
15	No fall	Taking the lift (one floor, down)
16	No fall	Walk down stairs (6 steps)

One thread is triggered by the acceleration data, and another is triggered by the barometric pressure data. Each thread sets different global software flags based on the features from both acceleration and pressure data.

In the thread triggered by the acceleration data (Fig. 1), in order to reduce power wastage, there are several prerequisites to trigger a higher acquisition rate, including comparing the tilt angle and the signal vector magnitude (SVM) with a predefined threshold. After passing these criteria and increasing the sampling rate, the algorithm will extract different features from the acceleration data, and compare them with predefined thresholds to make the decision whether to set the related flags.

Additionally, the algorithm will also manage the data transmission from the sensor to the controller. Specifically, the term *data transmission* refers to communication between different chips on board along different buses, such as Serial Peripheral Interface (SPI) or Inter-Integrated Circuit (I2C) buses. The controller usually needs be active to participate in the above process of communication. Note that the controller is the major source of energy consumption, frequent data transmissions will increase the time that the controller spends waking from a low-power state; controllers usually operate at high power usages during wake-up at its internal circuits initialize. And a large amount of data transmitted by the buses will also increase the wake-up time of the controller. Thus, it is energy efficient to reduce frequency and amount of data transmission from the sensors to the controller.

The algorithm has a screening step to achieve this goal. In the screening step, the controller only acquires three samples from the sensors and calculates simple features. If the features reflect some signs of a possible fall, the controller will then trigger the acquisition of more information to accurately detect the fall, otherwise the controller will go back to sleep to save power and not acquire subsequent data from the sensor until new data show a sign of a possible fall at a later time.

In more detail, the controller will initially acquire acceleration data at a lower sampling rate. If there is no sign of a possible fall event, the sampling rate will not change. However, when the SVM of the raw acceleration is lower than a threshold (thr_r), the controller will calculate the related features. If the features cannot meet the screening conditions, the controller will return to sleep immediately. After passing all the screening steps, the controller will switch the accelerometer to sample at a higher rate for two seconds, and then extract the body acceleration (BA) and gravitational acceleration (GA) components based on a data window that contains acceleration at the lower sampling rate acquired during a possible "falling" phase and data at a higher sampling rate during a possible "impact" phase. The algorithm updates the upper threshold of tilt angle (thr_ha) according to the GA component. In the BA component, if there is an abnormal SVM spike, the algorithm will set an impact flag. After two seconds of high-speed sampling, the algorithm will adjust the sampling rate of the accelerometer back to the lower rate. Once the impact flag is set, this thread will affect the access behavior of the controller to the barometric sensor's FIFO (a hardware capability of recent barometric pressure sensors). The controller will retrieve the data before and after the point of possible impact, and then calculate the delta pressure (the average differential pressure). If the delta pressure is higher than the threshold (thr_hp), the algorithm will set a ground flag, indicating a possible fall. After setting the ground flag, the algorithm will search for a negative delta pressure peak. If it exists and the tilt angle is below the threshold (thr_ha), the algorithm will set the recover flag.

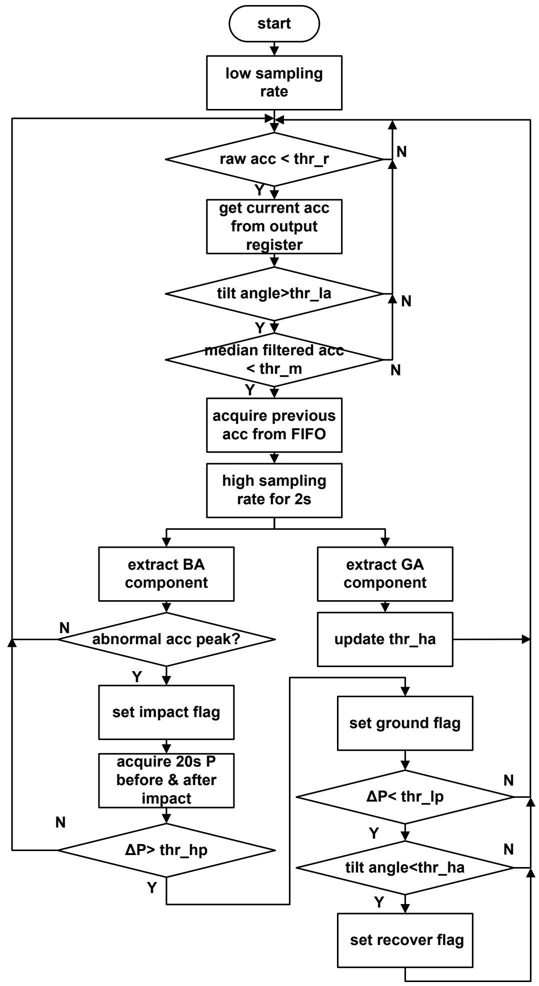

Fig. 1 The processing thread triggered by acceleration, focused on detecting the fall types 1-4 and 7-8 of Table I which have an obvious impact event.

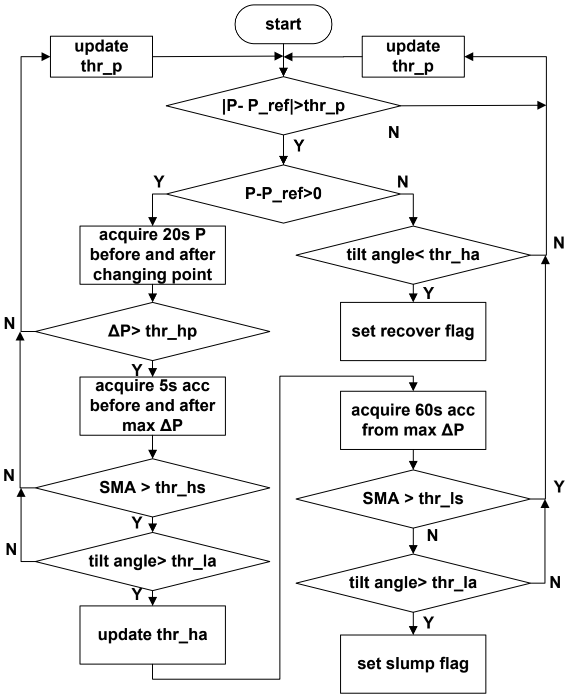

Fig.2 The processing thread triggered by pressure focused on detecing the fall types 5-6 which do not have an obvious impact event.

In the thread triggered by the pressure data (Fig. 2), if the absolute value of the difference between the current pressure data and a pre-configured pressure reference is higher than the threshold (thr_hp), the barometric pressure sensor will trigger an interrupt to the controller, and the controller will be woken to analyze the difference. If the difference is negative, and the direction of the trunk is near vertical, the algorithm will set a recover flag and update the pressure reference. If there is an abnormal positive peak in the delta pressure signal, the algorithm will additionally check the signal magnitude area (SMA) to see whether there is an obvious movement near the time of the local maximum of the delta pressure, since the fall event is expected to occur with increased movement activity. If the features meet the above conditions, the algorithm will monitor the acceleration signal at the lower sampling rate for 60 s after this sudden pressure increase. If the SMA is continuously low, and the direction of the trunk is continuously near the horizontal, it means that the person has been lying on the ground for a long time, and the algorithm will set the slump flag.

Finally, the algorithm will make a judgment as to whether a fall occurred and whether the person has recovered after the fall, based on the status and relative timing of the different flags. If the time between the impact flag and the ground flag is less than 6 s, and there is no recover flag in the following 20 s, the algorithm determines that a fall event has occurred. If the slump flag is set, and there is no recover flag in the following 60 s, the algorithm determines that the person has slumped and lost consciousness.

Fig. 3 shows a representative example of the algorithm dynamically adjusting the sampling rate of the accelerometer and managing data transmission between the sensors and the controller.

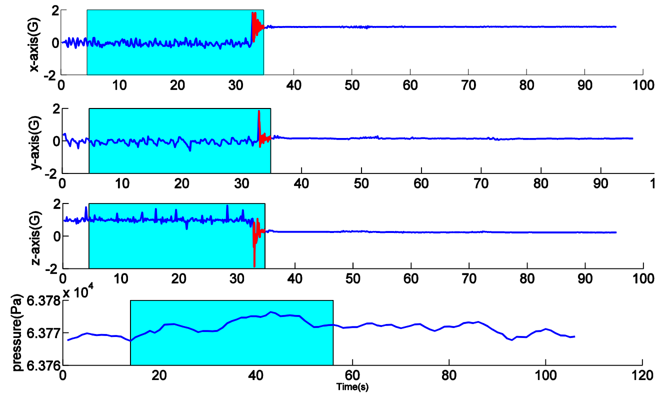

Fig. 3 Acceleration and barometric pressure data from a trial of simulated falls, where the blue curve represents the acceleration signal with a 6 Hz sampling rate, and red signals represents the acceleration signal with a 40 Hz sampling rate. "Transmitting window" refers to the data transmitted from sensors to the controller.

III. Results

The proposed algorithm is evaluated in MATLAB to determine the performance of fall detection algorithm and the reduction in power consumption based on the specific hardware design. The accuracy, sensitivity and specificity for all simulated trials are shown in Table II. The accuracy, sensitivity and specificity of the proposed algorithm are 95.9%, 96.7% and 96.9% respectively.

The performance in terms of energy efficiency is measured by the difference between the average power consumption of sensing and transmitting parts of the system using the original algorithm and the low-power algorithm from each trial. The calculation of power consumption is as follows:

$$P_{s+t} = \frac{\sum_{sensor_i, status_j} P_{i,j} t_{i,j} + P_c \sum_{sensor_i} \frac{D_i}{BR_i}}{t_{total}} \quad (1)$$

where P_{s+t} is the power consumption of sensing and transmitting data around the board, $P_{i,j}$ is the sampling power of a sensor i when it works in a status j (high sampling rate or low sampling rate), $t_{i,j}$ is the corresponding time for which $P_{i,j}$ is consumed, P_c is the power of active controller, D_i is the amount of data generated for sensor i in a trial, BR_i is the byte-rate of the communication protocol (such as SPI) between sensor i and the controller, D_i/BR_i represents the time consumption caused by data transmission, and t_{total} is the duration of a trial.

Here it is assumed that the controller is an MSP430FR5737 (the active current is 1 mA), the accelerometer is an ADXL362 (the current in the measure mode with a sampling rate of 40 Hz is 1.75 μA, the current in the wake-up mode with a sampling rate of 6 Hz is 270 nA), and barometric pressure sensor is a LPS25H (the current is 25 μA), the voltage of the system is 2 V, and the clock frequency of the SPI interface is 1 MHz. As a result, the power consumption using original algorithm and using the low-power algorithm are 57.4 μW and 51.1 μW respectively for each trial, and the reduction of the power consumption is

6.3 μW. The power saving achieved by lowering the sampling rate is 2.9 μW, while the power saving due to a selective data transmission window is 3.4 μW.

IV. Discussion and Conclusion

From the results, the proposed low-power algorithm inherits high sensitivity and specificity from the original algorithm. The proposed algorithm performs slightly better than the original algorithm in terms of specificity, but has a slightly worse sensitivity. The differences in sensitivity and specificity between the proposed algorithm and the original algorithm are less than 1%. In total, the proposed algorithm only decreases in accuracy by 1.0 %. Despite this, the results show that the proposed algorithm successfully captures almost all information related to falls in a short sampling and transmitting window.

Additionally, in terms of power efficiency, the proposed algorithm consumes less than the original algorithm, since the proposed algorithm only samples the acceleration at 40 Hz (high power consumption) when the data meets some triggering conditions, and samples the acceleration at 5 Hz (low power consumption) at all other times. By contrast, the original algorithm continuously samples acceleration at 40 Hz. Besides, only the samples in the transmitting windows which possibly contain fall events will be transmitted to the controller for processing, so the wake-up time of the controller will decrease. The total reduction is 6.3 μW (4200 mAh per year), and it could prolong the lifetime of the fall detector.

To the best of our knowledge, very few papers [11, 12] have focused on low-power algorithms for wearable devices with an accelerometer. Kangas *et al.* reported an algorithm that collects the acceleration in event-trigger mode to minimize energy consumption and data transmission. Thirty samples with a sampling frequency of 6.25 Hz before the activation, 240 samples with a sampling frequency 50 Hz and then 20 samples with a sampling frequency of 6.25 Hz after the activation are collected [11]. However, the paper does not focus on the energy efficiency and only has a qualitative analysis about the decrease of energy consumption of the wireless transmission caused by this collecting method.

French *et al.* reported a predicted sampling algorithm based on the distribution of durations and transition probabilities of specific activities [12]. The accuracy of over 95% is achievable using only 3% of the samples. However, the intended use of the device is quite different from the fall detection task examined in this paper, and involves gait recognition and identification of ADL. Thus, it is acceptable for some latency and a relatively high false negative rate, but by contrast in the task of fall detection, the low false negative rate (high sensitivity) and low latency is a priority due to the criticality of the application. Thus the fall detection algorithm should operate in real-time, incurring the resulting energy costs.

In the future, the performance of the energy efficiency of the proposed algorithm will be tested on a cohort of older subjects, and the trade-off between power consumption and detection accuracy will be studied.

TABLE II. Accuracy, sensitivity, and specificity of the original algorithm and the proposed algorithm.

	Original algorithm	**Low-power algorithm**	**Difference**
Accuracy (%)	96.9	95.9	-1.0
Sensitivity (%)	97.5	96.7	-0.8
Specificity (%)	96.5	96.9	0.4

Reference

[1] K. Delbaere, J. C. T. Close, J. Heim, P. S. Sachdev, H. Brodaty, M. J. Slavin*, et al.*, "A multifactorial approach to understanding fall risk in older people," *Journal of the American Geriatrics Society,* vol. 58, pp. 1679-1685, 2010.

[2] E. I. Shih, A. H. Shoeb, and J. V. Guttag, "Sensor selection for energy-efficient ambulatory medical monitoring," in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, 2009, pp. 347-358.

[3] M. Shoaib, N. Jha, and N. Verma, "A low-energy computation platform for data-driven biomedical monitoring algorithms," in *48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2011, pp. 591-596.

[4] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini*, et al.*, "Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection," in *Wireless Sensor Networks*, ed: Springer, 2008, pp. 17-33.

[5] N. Ravanshad, H. Rezaee-Dehsorkh, R. Lotfi, and Y. Lian, "A level-crossing based QRS-detection algorithm for wearable ECG sensors," *IEEE Journal of Biomedical and Health Informatics,* vol. 18, pp. 183-192, 2014.

[6] M. Benocci, C. Tacconi, E. Farella, L. Benini, L. Chiari, and L. Vanzago, "Accelerometer-based fall detection using optimized ZigBee data streaming," *Microelectronics Journal,* vol. 41, pp. 703-710, 2010..

[7] L. Ren, Q. Zhang, and W. Shi, "Low-power fall detection in home-based environments," in *Proceedings of the 2nd ACM International Workshop on Pervasive Wireless Healthcare*, 2012, pp. 39-44.

[8] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, and N. H. Lovell, "Barometric pressure and triaxial accelerometry-based falls event detection," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 18, pp. 619-627, 2010.

[9] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine,* vol. 10, pp. 156-167, 2006.

[10] C. Becker, L. Schwickert, S. Mellone, F. Bagala, L. Chiari, J. L. Helbostad*, et al.*, "Proposal for a multiphase fall model based on real-world fall recordings with body-fixed sensors," *Zeitschrift Fur Gerontologie Und Geriatrie,* vol. 45, pp. 707-715, 2012.

[11] M. Kangas, I. Vikman, L. Nyberg, R. Korpelainen, J. Lindblom, and T. Jamsa, "Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects," *Gait Posture,* vol. 35, pp. 500-505, 2012.

[12] B. French, D. P. Siewiorek, A. Smailagic, and M. Deisher, "Selective sampling strategies to conserve power in context aware devices," in *11th IEEE International Symposium on Wearable Computers*, pp. 77-80, 2007.