Fall Detection Algorithm Based on Triaxial Accelerometer and Magnetometer

Tianjiao Shi, Xingming Sun, Zhihua Xia, Leiyue Chen, and Jianxiao Liu

***Abstract*—Fall is a precipitous drop from a height, or from a higher position, which may be accompanied by injuries. This is one of the most dangerous and fearful situation in the elderly living. This is the reason, fast and early detection of the fall is very important to save and rescue the people and avoid the badly prognosis. In this article we are presenting a threshold-based fall detection algorithm that processes data from common sensors in modern smart phones, such as triaxial accelerometer and magnetometer in order to detect falls. The algorithm uses Signal Vector Magnitude (SVM) peak value, base length and post-impact velocity to distinguish falls from most of daily activities. However, the SVM curve in a period produced by running is similar to a fall (a running curve can be regarded as the combination of multiple fall curves). Accordingly, residual movement is taken into account to identify running and fall. In addition, the vertical acceleration is observed to increase detection accuracy. In the experiments, the data are collected by simulating fall in four directions: forward, backward, left and right. The simulations are conducted by young people. The final experiment includes data from 120 simulated falls and 150 daily activities. Compared with previous methods, the proposed method achieves higher sensitivity and specificity.**

***Index Terms*—Accelerometer, Fall Detection, Magnetometer, Residual Movement, Smart phone**

I. Introduction

FALL is a fearful and dangerous in the elderly people's daily life. Each year, there are more than one third of people aged over 65 years old falling [1], [2]. The immediate rescue to fall would greatly alleviate the hurt. The automatic detection of fall events could help reducing the response time and significantly improve the prognosis of fall victims.

Nowadays, a number of fall detection methods have been proposed [3]. According to the methods of data acquirement, fall detection methods can be divided into three categories. The first category can be called context-aware method which usually utilizes pressure sensor to detect the people whether fall or not [4], [5]. This kind of method will be out of work if the pressure is not under the user's weight after fall. The second category is camera-based method which is installed as an important part of in-home auxiliary system [6], [7], [8], [9], [10], [11], [12], and the fall events are detected by analysing the video pictures [13], [14], [15], [16], [17], [18], [19] with classification techniques. The whole system is usually expensive and only works at the video-surveillance place. The third category is the acceleration-based method which is now the most popularly used one [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. At the beginning, the acceleration-based system uses body-attached sensors such as accelerometers and gyroscopes to acquire kinetic data from human motion. Through it can detect the fall event, it is not portable. The modern smart phones have been equipped with various build-in sensors which can be directly utilized for fall detection [25], [26], [27], [28], [29], [30]. In this paper, we proposed a threshold-based fall detection method by using the triaxial accelerometer and magnetometer which are common sensors in most kinds of modern smart phones. Four features are extracted from accelerometer and magnetometer raw data to distinguish fall event from activities of daily living (ADL) such as walking, going upstairs, going downstairs, and jumping. In addition, the algorithm also takes residual movement into account after the fall to improve the accuracy as a running curve can be regarded as the combination of multiple fall curves. As long as all feature values meet the condition of the threshold values we set, we can recognize the fall.

The remainder of this paper is structured as follows. In section II, we present an overview of related works. We introduce the algorithm design in Section III and implementation in Section IV. In Section V, we evaluate our system with extensive experiments. Section VI summarizes the conclusion of the paper.

II. Related Works

As mentioned above, the existing fall detection methods can be divided into three categories. As the development of smart phones, the acceleration-based fall detection systems can be conveniently applied in real world applications, and thus attract more and more attentions.

At the beginning, researchers proposed some acceleration-based fall detection methods by attach the acceleration sensor on tester's body. It is quite inconvenience, but provides preliminary knowledge for us to refer to. Yang et al. [20] used the two wireless acceleration sensors embedded with Naive Bayes algorithm to implement a wearable real-time fall detection system which has an advantage both in accuracy performance and model building time. The system attached sensors to the chest and the thigh provides acceleration information to detect forward, backward, leftward and rightward falls. Wang et al. [24] proposed a low-power fall detection algorithm based on triaxial accelerometer and barometric pressure signals. The algorithm dynamically adjusts the sampling rate of an accelerometer and manages data transmission

Manuscript received June 11, 2015; revised October 06, 2015. This work is supported by the NSFC (61173141, 61232016, 61572258, 61502242, U1405254, U1536206, 61173136, 61373133), 201301030, 2013DFG12860, BK20150925, Fund of Jiangsu Engineering Center of Network Monitoring (KJR1308, KJR1402), Fund of MOE Internet Innovation Platform (KJRP1403), CICAEET, and PAPD fund. Zhihua Xia is supported by Jiangsu Government Scholarship for Overseas Studies (JS-2014-332).

Tianjiao Shi, Xingming Sun, Zhihua Xia and Leiyue Chen are with Jiangsu Engineering Center of Network Monitoring, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, China. (e-mail: tianjiao_shi@163.com, sunnudt@163.com, xia_zhihua@163.com, chenly_cool@163.com).

Jianxiao Liu is with College of informatics, Huazhong Agricultural University, China (e-mail: liujianxiao321@163.com).

between sensors and a controller to reduce power consumption. Pham et al. [23] presented a fall alert system using an accelerometer sensor for elderly people based on threshold algorithm. The algorithm calculates Signal Magnitude Area (SMA) to distinguish fall events from daily activities. If this value is greater than the predefined threshold, the system recognizes as a fall and then displays a warning.

For the reason that the update of Micro Electro Mechanical System (MEMS) motion sensors technology and the popularity of mobile phones which are equipped with 3-axis accelerometers and other motion sensors, there is no doubt that smart phones have become ideal devices for fall detection which are popular in researchers in recent years.

Dai et al. [28] designed a fall detection system, called PerFall, which is the first pervasive fall detection system using mobile phones. The algorithm is based on accelerometer thresholds: the total acceleration and the vertical acceleration need to compare with predefined thresholds which is adjusted with collected data. Sposaro et al. [30] proposed a fall alert system called iFall which used triaxial accelerometers for fall detection. The algorithm is based on acceleration magnitude thresholds and timeouts detection. If a fall is detected, the system will send a request for help to the caregivers. Tiwari et al. [25] presented a detection mechanism for free fall condition using an Android based smart phone with an in-built triaxial accelerometer. In this algorithm value of acceleration is analysed for the time it was close to 0. In this time we obtain the net displacement. If the obtain value of displacement crosses a threshold predefined value than free fall condition is confirmed.

III. ALGORITHM DESIGN

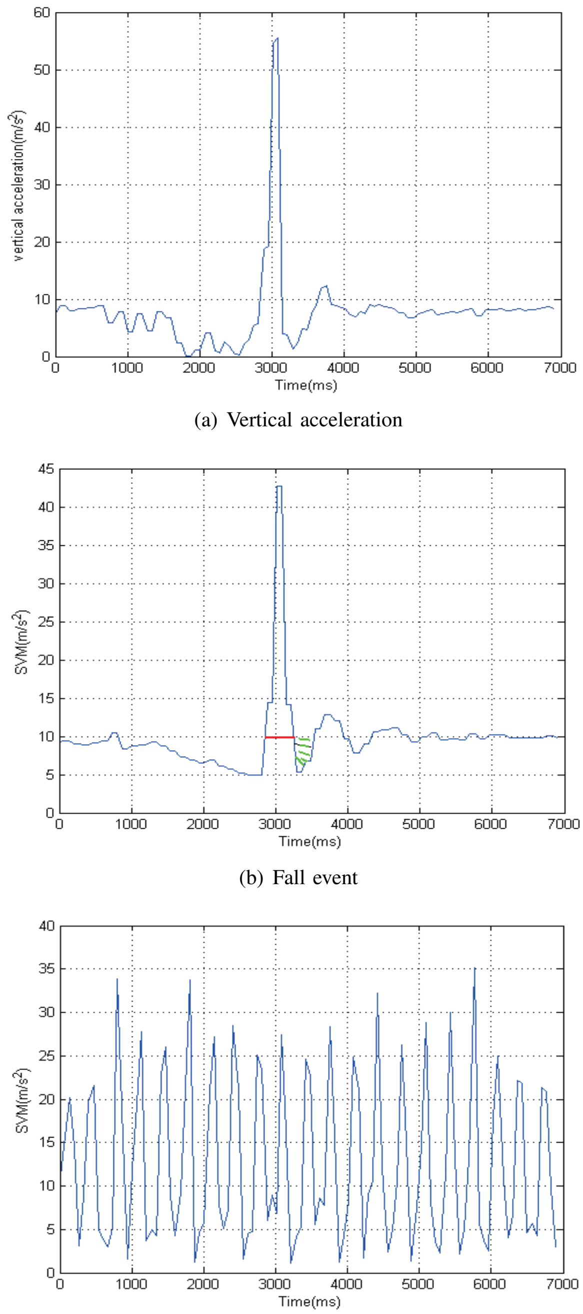
In this section, we propose a threshold-based fall detection algorithm which mainly utilizes four features which are identified from the analysis of the signal vector magnitude (SVM) to recognize the fall event from human activities. The four features are the SVM peak value, the base length of the triangle, post-impact velocity and residual movement respectively. In addition, the vertical acceleration is added to the approach as a verified feature. The threshold values are chosen based on the data set including different types of fall data (including forwards, backwards, sideways) and ADL (including walking, running, jumping, going downstairs and going upstairs), as is shown in Fig. 1(c) and Fig. 2.

A. Algorithm Features and Thresholds

Human body movement can be reflected by acceleration. When people is walking, going downstairs, going upstairs, jumping, running, or falling, acceleration data varies. For simplification, SVM is calculated as follow to eliminate the sensitivity of mobile phone in aspect of position and direction.

$$\mathbf{SVM} = \sqrt{A_x^2 + A_y^2 + A_z^2} \tag{1}$$

where A_x, A_y, and A_z represent the accelerating speeds of the X, Y, and Z axes of the accelerometer respectively. SVM values are shown in Fig. 1, the horizontal axis represents the time (ms) and the vertical axis represents the SVM values(m/s^2).The algorithm defines the three features mentioned in [31] to distinguish fall event from ADL below:



(a) Vertical acceleration

(b) Fall event

(c) Running activity

Fig. 1: This figure shows the waveforms of (a) vertical acceleration, (b) fall event, and (c) running activity. The horizontal axis represents the time (ms) and the vertical axis represents the SVM values (m/s^2).

1) SVM peak value (denoted as P): Most falls have a higher SVM value than daily activities as shown in Fig. 1(b), so the SVM peak value is a good choice in the design of the fall detection algorithm. From the Fig. 2(a), (c), and (d), we can know that the waveforms of walking, going downstairs and going upstairs and find that SVM peak values of them are all lower than a given thresholds. So, it can detect falls totally but there are still some ADL mistaken for fall.

2) The base length of the triangle (denoted as B): It is formed by the peak value and the 10 (m/s^2) horizontal axis as shown in the short solid line of Fig. 1(b). When a fall happens, there is a short time impact and the shock is quickly absorbed by the human body. Therefore, the base length should be smaller in case of a fall compared to the case of daily activities.

3) Post-impact Velocity (denoted as V): As what is shown in the shaded area of Fig. 1(b). When the event meets P and B threshold, it can be deemed to a possible fall. The velocity after the impact is utilized to distinguish sudden moves, like jumping, quickly sitting on a bed or physical exercise. By the analysis of the experiment data, *P*, *B*, and *V* values which produced the optimal results in terms of accuracy are selected in Table I:

TABLE I: Fall detection algorithm features and associated optimal thresholds

No	Features description	Code	Threshold	Fall sign
1	Peak value for SVM	P	23.4 (m/s^2)	$\geq$
2	Base length	B	0.5 s	$<$
3	Post-impact velocity	V	0.5 m/s	$<$

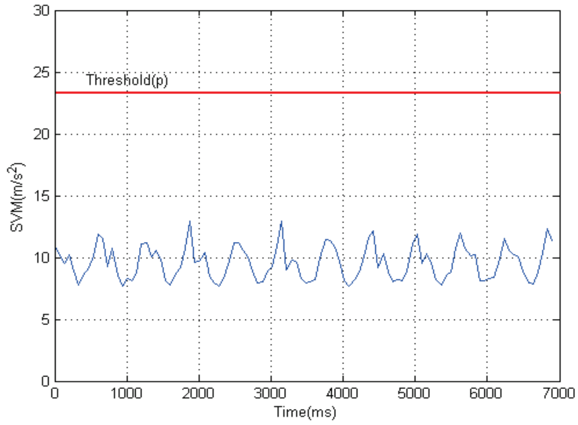
Although the three features mentioned above can be used to separate fall event from most of ADL, running activity cannot be distinguished. So, the algorithm takes residual movement into account to recognize the running activity.

4) Residual movement: When the fall happens, the first peak value reaches the threshold we set. And then, the second peak value follows which lower than the threshold we set. At last, the acceleration back to normal. However, when we are running, the peak values all exceed the pre-set threshold and the time interval between the peak values is smaller than the one in fall experiment. The waveforms of running activity and fall event are shown in Fig. 1(b) and (c). Furthermore, we set a variable *count* in our algorithm, which increases every time in condition that four features mentioned above are satisfied in order to describe the residual movement [32]. If the values of *count* are between 10 and 15, algorithm takes the event as the fall. Otherwise, algorithm takes the event as running.

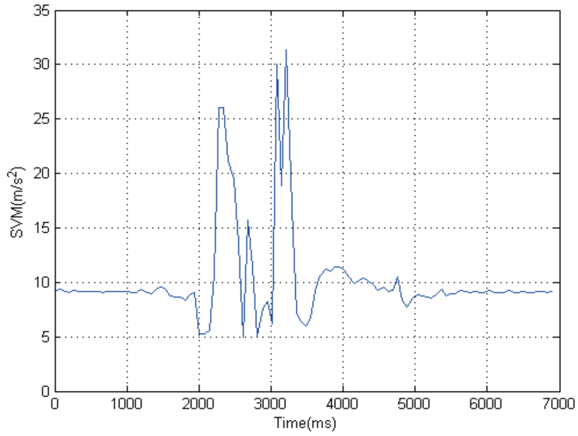
5) Vertical acceleration: Additionally, fall detection algorithm adds the vertical acceleration as a verified procedure. Before and after the fall, the waveform of the vertical acceleration is similar to SVM's. In this study, we utilize the vertical acceleration value $|A_v|$ [33] as a verified procedure. And the vertical acceleration can also take running-to-walking as non-fall to improve the accuracy. The vertical acceleration value $|A_v|$ which is computed as:

$$|A_v| = |A_x \sin\theta_z + A_y \sin\theta_y - A_z \cos\theta_y \cos\theta_z| \qquad (2)$$

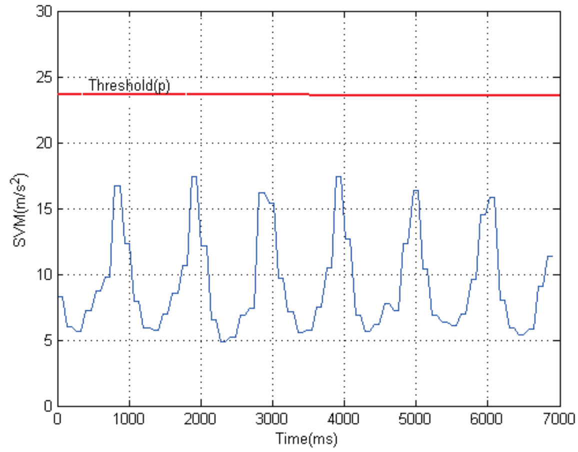
where θ_y and θ_z denote its pitch and roll values which is determined by a mobile phone's accelerometer and magnetometer sensor. If the value of $|A_v|$ is larger than the threshold $Th_{|A_v|}$ we pre-set, it will be considered as a possible fall. In the algorithm, we set $Th_{|A_v|}$ value for 30. Before and after a fall event, the vertical acceleration value will change which is illustrated in Fig. 1(a).



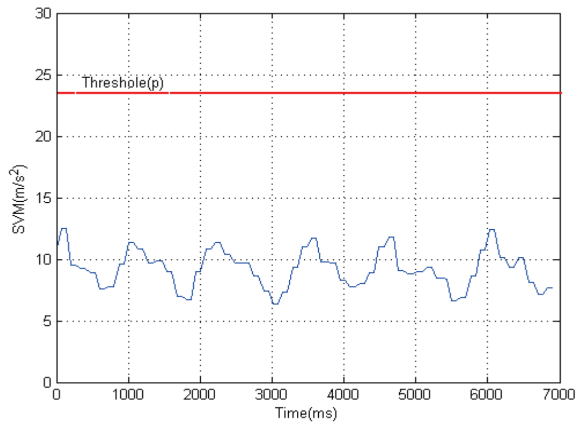
(a) Walking



(b) Jumping



(c) Going downstairs



(d) Going upstairs

Fig. 2: Activities of daily living.

IV. IMPLEMENTATION

We have developed fall detection system which is implemented in the smart phone using the algorithm proposed in this paper. The system mainly consists of three main screens as shown in Fig. 3(a), (b), and (c). Our application is simple, smart and efficient as its biggest advantage. Fig. 3(a) shows the main screen of the system and the raw data of sensors. Fig. 3(b) depicts some parameters of fall detection algorithm including SVM threshold, vertical acceleration threshold and sensor speed value, which can be set by the participant. Fig. 3(c) depicts some parameters of warning message including the telephone number of the emergency contact, time-out and SMS content which can be set by the participant before monitoring application. Once a fall condition is detected, the system will send a warning message. The message contains the participant's longitude, latitude, detailed address and fall information to emergency contact as shown in Fig. 4.

V. EXPERIMENTS

We evaluate the effectiveness of the proposed algorithm with extensive experiments. In this section, we briefly introduce our experimental environment and restricted condition. Then we introduce how the data are collected. In addition, we also present performance of system and compare it with that of existing algorithms. Finally, we present resource consumption of the system.

A. Experimental Environment

In our experiments, we used the smart phone built-in three-axis accelerometer and magnetometer to realize the fall detection algorithm. The sampling frequencies of the accelerometer and magnetometer values are both 15Hz. Furthermore, results indicate that fall detection using a triaxial accelerometer attached to the waist or head is more efficient [34] from previous studies. So, all subjects are required to attach the smart phone to the waist to get data compared to attach to the head which is more convenient in our study.

B. Data Collection

Since elderly people are not suitable for simulating falls, our data is derived from healthy young subjects who volunteer to perform the simulate falls and the ADL tasks.

We selected 10 young subjects, including 6 male and 4 female, with an average age of 25 $\pm$ 3 years, average height of 170 $\pm$ 10 cm and an average weight of 70 $\pm$ 12 kg. Each participant performed four types of falls (including forwards, backwards, sideways). The simulation of each kind of fall was conducted for 2 minutes and was repeated three times. In addition, each participant performed five types of ADL tasks (including walking, running, jumping, going downstairs and going upstairs) and each kind of ADL was conducted for 5 minutes and was repeated three times. In total, we obtain data for 120 falls and 150 ADL.

C. Detection Performance

We measure the detection performance in terms of sensitivity and specificity. The sensitivity defined in (3) is used to measure the algorithm's ability to determine real falls from the set of simulated falls.

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \tag{3}$$

where TP is the true positive representing a fall occurs and the system properly detects it. FN is the false negative represent a fall occurs but the system does not announce it.

On the other hand, the specificity defined in (4) is used to measure the algorithm's ability to mistake normal activities for falls.

$$Specificity = \frac{TN}{TN + FP} \times 100\% \tag{4}$$

where TN is the true negative representing a non-fall movement is performed and the system does not announce a fall. FP is the false positive represent the system announces a fall although it did not occur.

- Thresholds Robustness

To evaluate how robust these thresholds are, we can vary these thresholds and the algorithm's performance which can be re-evaluated. By the analysis of the experiment data, the thresholds listed in Table I produce optimal results. Modifying these thresholds can degrade performance. However, it is important to see how much the algorithm's performance will change.

First, *P*, *B* and *V* values were changed so that the risk of false positive or true negative could increase. For instance, if the peak value *P* becomes smaller, more daily activities may be identified as falls and false positives will increase. Similarly, if the peak value *P* becomes greater, more falls may be identified as daily activities and true negative will increase.

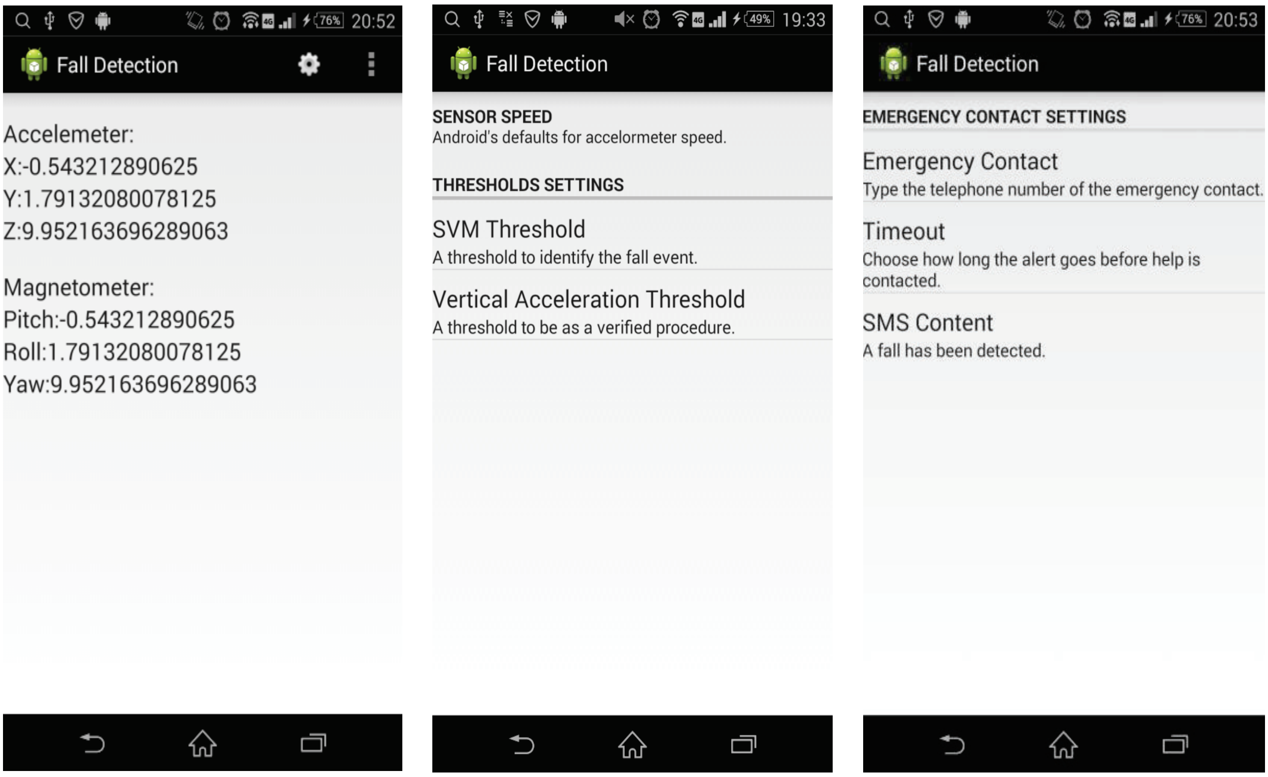
When the thresholds have a 5% variation to increase the false positive, two jumping events and two running events were identified as falls. This led to a specificity of 92.67%. When the thresholds have a 10% variation for increasing, false positive was generated by the algorithm, thus obtaining 88.67% specificity.

The thresholds were also modified so as to increase the true negative. When a 5% variation for reducing the threshold, three falls were misinterpreted as daily activities, which led to 88.83% sensitivity. Similarly, When a 10% variation for reducing the threshold, four fall events weren't detected, which led to 87.5% sensitivity. Table II shows the results obtained using the thresholds variations presented. In general, the algorithm provided satisfactory results.

TABLE II: Algorithm performance using different threshold values

Threshold values	Algorithm performance (%)	
	Sensitivity	Specificity
Optimal values(see Table I)	90.83	95.33
5% variation from optimal values so as to increase the risk of false positive	90.83	92.67
10% variation from optimal values so as to increase the risk of false positive	90.83	88.67
5% variation from optimal values so as to increase the risk of true negative	88.83	95.33
10% variation from optimal values so as to increase the risk of true negative	87.50	95.33

- Locations and Sensors



(a) Main screen (b) Threshold settings (c) Emergency contact settings

Fig. 3: User interfaces in fall detection system.

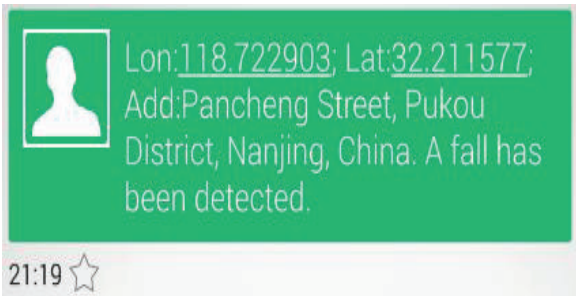


Fig. 4: The content of a warning massage.

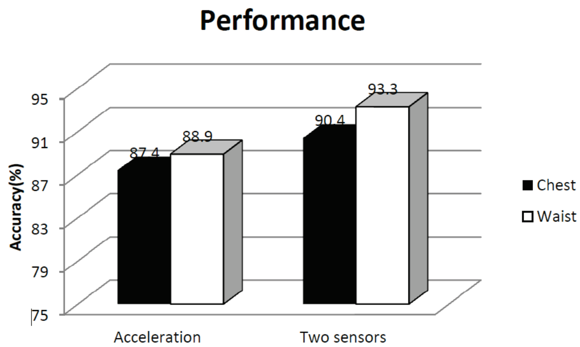


Fig. 5: Accuracy variance of different location and different numbers of sensors.

Due to smart phone internally installs multiple sensors, fall detection algorithm uses only acceleration and magnetometer. Firstly, the experiment evaluates the accuracy of acceleration in the chest and waist. From Fig. 5, it can be seen that there is the better accuracy in the chest with accuracy of 90.4%. Obviously, the movement of chest is the better indication of fall detection. Secondly, when smart phone is placed in the waist, the accuracy with one sensor is lower than algorithm with two sensors. So, in this paper, both acceleration and magnetometer are used to detect the fall event.

- Performance Comparison

By a series of experimental tests, the experimental results are shown in Table III below. Table III shows the number of samples each activity in parenthesis. It can be observed that the accuracy of this algorithm is quite high as shown in Table III. Among experimental data, walking, going down-stairs, and going upstairs are correctly identified and recognition rate achieve the 100% in our data set. There are five jumping events mistaken for fall and two running events mistaken for fall. The main reason of the phenomenon is that young subjects are different from older people in simulating jumping and running activities. Finally, our algorithm can recognize 109 falls from 120 simulated fall events.

TABLE III: The result of the experiment tests

	Samples					
	Walking (30)	Going down-stairs (30)	Going up-stairs (30)	Jumping (30)	Running (30)	Fall (120)
Fall	0	0	0	5	2	109
Non-Fall	30	30	30	25	28	11

Both Cao's method [35] and Fang's method [36] are compared to our method. Table IV lists the result of the three methods and shows the best part in our algorithm's result. Cao's method is based on three axis accelerations and only considers the resultant acceleration values. The reason that we compare with Cao's method is to illustrate that two sensors can obtain more information to distinguish the fall event. As for the Fang's method, it also puts the device in the waist and we can compare the sensitivity and specificity when two devices are put in the similar position in aspect of performance. By contrast, we find that our algorithm has better performance.

TABLE IV: The result of the three methods

	Ours	Cao	Fang
Sensitivity (%)	90.83	92.75	72.22
Specificity (%)	95.33	86.75	73.78

D. Power Consumption

To test power consumption, we fully charged the two SONY Z3 phones and then monitored the power states continuously for 6 hours respectively. One of them was remained without the fall detection application, while the other ran the application continuously. Fig. 6 presents the two curves of battery level states versus time during the time period of 6 hours. If the application keeps running normally until the battery power is exhausted, it will last more than 20 hours.

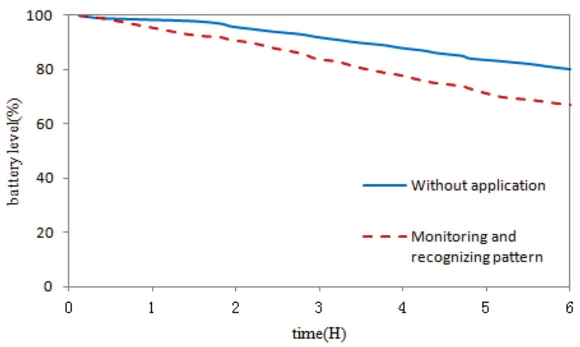


Fig. 6: Power consumption: the solid line shows the battery levels when the phone runs without the fall detection application; the dash line presents the battery levels when the monitoring daemon and the pattern recognition in the fall detection system run.

VI. Conclusion

This paper proposed a fall detection algorithm that utilizes the smart phone with built-in accelerometer and magnetometer sensor to recognize the fall event from the human activities. We firstly select ten subjects simulate ADL and fall events to collect raw data which is to observe the difference in human activities and get the option thresholds. Considering the trade-off between recognition accuracy and computing load, the sampling rate of 15Hz is adopted by the algorithm. In addition, the algorithm uses the peak value of SVM to detect fall events totally but there are still some ADL mistaken for fall. In order to improve the accuracy, base length of the triangle (B) and post-impact velocity (V) are made the distinction between a real fall and some ADL, like jumping, quickly sitting on a bed or physical exercise. However, when the subject is running, the waveform of which is similar to the waveform of the fall event meets the thresholds above. So, our approach considers the residual movement to distinguish with the real fall event and running activity. Finally, the algorithm uses the vertical acceleration to monitor the subject's body posture for verification.

References

[1] W. H. O. Ageing and L. C. Unit, *WHO global report on falls prevention in older age*. World Health Organization, 2008.

[2] S. Yoshida-Intern, "A global report on falls prevention epidemiology of falls," *WHO. Geneva*, 2007.

[3] R. Luque, E. Casilari, M.-J. Morón, and G. Redondo, "Comparison and characterization of android-based fall detection systems," *Sensors*, vol. 14, no. 10, pp. 18543–18574, 2014.

[4] M. Alwan, P. J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, "A smart and passive floor-vibration based fall detector for elderly," in *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, vol. 1, pp. 1003–1007, IEEE, 2006.

[5] H.-W. Tzeng, M.-Y. Chen, and M.-Y. Chen, "Design of fall detection system with floor pressure and infrared image," in *System Science and Engineering (ICSSE), 2010 International Conference on*, pp. 131–135, IEEE, 2010.

[6] E. Vlaeyen, M. Deschodt, G. Debard, E. Dejaeger, S. Boonen, T. Goedemé, B. Vanrumste, and K. Milisen, "Fall incidents unraveled: a series of 26 video-based real-life fall events in three frail older persons," *BMC geriatrics*, vol. 13, no. 1, p. 103, 2013.

[7] L. Alhimale, H. Zedan, and A. Al-Bayatti, "The implementation of an intelligent and video-based fall detection system using a neural network," *Applied Soft Computing*, vol. 18, pp. 59–69, 2014.

[8] J.-L. Chua, Y. C. Chang, and W. K. Lim, "A simple vision-based fall detection technique for indoor video surveillance," *Signal, Image and Video Processing*, vol. 9, no. 3, pp. 623–633, 2013.

[9] R. Cucchiara, A. Prati, and R. Vezzani, "A multi-camera vision system for fall detection and alarm generation," *Expert Systems*, vol. 24, no. 5, pp. 334–345, 2007.

[10] D. Anderson, J. M. Keller, M. Skubic, X. Chen, and Z. He, "Recognizing falls from silhouettes," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pp. 6388–6391, IEEE, 2006.

[11] W.-Y. Shieh and J.-C. Huang, "Speedup the multi-camera video-surveillance system for elder falling detection," in *Embedded Software and Systems, 2009. ICESS'09. International Conference on*, pp. 350–355, IEEE, 2009.

[12] C.-W. Lin and Z.-H. Ling, "Automatic fall incident detection in compressed video for intelligent homecare," in *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pp. 1172–1177, IEEE, 2007.

[13] Z. Pan, Y. Zhang, and S. Kwong, "Efficient motion and disparity estimation optimization for low complexity multiview video coding,"

[14] B. Chen, H. Shu, G. Coatrieux, G. Chen, X. Sun, and J. L. Coatrieux, "Color image analysis by quaternion-type moments," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 124–144, 2015.

[15] F. Nie and P. Zhang, "Fuzzy partition and correlation for image segmentation with differential evolution," *IAENG International Journal of Computer Science*, vol. 40, no. 3, pp. 164–172, 2013.

[16] Y.-S. Huang and Y.-J. Wang, "A hierarchical temporal memory based hand posture recognition method," *IAENG International Journal of Computer Science*, vol. 40, no. 2, pp. 87–93, 2013.

[17] M. F. Hashmi, A. R. Hambarde, and A. G. Keskar, "Robust image authentication based on hmm and svm classifiers," *Engineering Letters*, vol. 22, no. 4, pp. 183–193, 2014.

[18] M. H. Taieb, J.-Y. Chouinard, and D. Wang, "Low complexity rate estimators for low latency wyner-ziv video decoders," *Engineering Letters*, vol. 21, no. 1, pp. 1–9, 2013.

[19] R. Pérez-Aguila, "Enhancing brain tissue segmentation and image classification via 1d kohonen networks and discrete compactness: an experimental study," *Engineering Letters*, vol. 21, no. 4, pp. 171–180, 2013.

[20] X. Yang, A. Dinh, and L. Chen, "Implementation of a wearerable real-time system for physical activity recognition based on naive bayes classifier," in *Bioinformatics and Biomedical Technology (ICBBT), 2010 International Conference on*, pp. 101–105, IEEE, 2010.

[21] T. Zhang, J. Wang, L. Xu, and P. Liu, "Fall detection by wearable sensor and one-class svm algorithm," in *Intelligent Computing in Signal Processing and Pattern Recognition*, pp. 858–863, Springer, 2006.

[22] H. Gjoreski, S. Kozina, M. Gams, and M. Lustrek, "Rarefallreal-time activity recognition and fall detection system," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, pp. 145–147, IEEE, 2014.

[23] P. T. Phu, N. T. Hai, and N. T. Tam, “A threshold algorithm in a fall alert system for elderly people,” in *5th International Conference on Biomedical Engineering in Vietnam*, pp. 347–350, Springer, 2015.

[24] C. Wang, M. R. Narayanan, S. R. Lord, S. J. Redmond, and N. H. Lovell, “A low-power fall detection algorithm based on triaxial acceleration and barometric pressure,” in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pp. 570–573, IEEE, 2014.

[25] R. Tiwari, A. K. Singh, and S. N. Khan, “Using android platform to detect free fall,” in *Information Systems and Computer Networks (ISCON), 2013 International Conference on*, pp. 161–163, IEEE, 2013.

[26] W. Wibisono, D. N. Arifin, B. A. Pratomo, T. Ahmad, and R. M. Ijtihadie, “Falls detection and notification system using tri-axial accelerometer and gyroscope sensors of a smartphone,” in *Technologies and Applications of Artificial Intelligence (TAAI), 2013 Conference on*, pp. 382–385, IEEE, 2013.

[27] Y. Kim, S. Kim, D. Kang, H. Park, N. Kim, S. H. Yang, and Y. Kim, “A simple falling recognition scheme for a human body by using mobile devices,” in *Proceedings of the 1st International Conference on Advanced Information and Comuter Technology (AICT), Warsawa, Poland*, pp. 14–17, 2013.

[28] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan, “Perfalld: A pervasive fall detection system using mobile phones,” in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pp. 292–297, IEEE, 2010.

[29] M. V. Albert, K. Kording, M. Herrmann, and A. Jayaraman, “Fall classification by machine learning using mobile phones,” *PloS one*, vol. 7, no. 5, p. e36556, 2012.

[30] F. Sposaro and G. Tyson, “ifall: an android application for fall monitoring and response,” in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 6119–6122, IEEE, 2009.

[31] M. Dumitrache and S. Pasca, “Fall detection algorithm based on triaxial accelerometer data,” in *E-Health and Bioengineering Conference (EHB), 2013*, pp. 1–4, IEEE, 2013.

[32] P. Kostopoulos, T. Nunes, K. Salvi, M. Deriaz, and J. Torrent, “Increased fall detection accuracy in an accelerometer-based algorithm considering residual movement,”

[33] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan, “Mobile phone-based pervasive fall detection,” *Personal and ubiquitous computing*, vol. 14, no. 7, pp. 633–643, 2010.

[34] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, “Comparison of low-complexity fall detection algorithms for body attached accelerometers,” *Gait & posture*, vol. 28, no. 2, pp. 285–291, 2008.

[35] Y. Cao, Y. Yang, and W. Liu, “E-falld: A fall detection system using android-based smartphon,” in *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pp. 1509–1513, IEEE, 2012.

[36] S.-H. Fang, Y.-C. Liang, and K.-M. Chiu, “Developing a mobile phone-based fall detection system on android platform,” in *Computing, Communications and Applications Conference (ComComAp), 2012*, pp. 143–146, IEEE, 2012.