ASEE 2014 Zone I Conference, April 3-5, 2014, University of Bridgeport, Bridgeport, CT, USA.

Fall Detection Using Single Tri-Axial Accelerometer

Suleman Belal Kazi, Sherjeel Sikander, Sadia Yousafzai
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology
Topi, Swabi, Pakistan

***Abstract*— This paper describes a mobile phone based system which implements a fall detection algorithm using a mobile phone's built-in accelerometer which can detect falls with a high degree of accuracy. The application developed can then notify predefined guardians or emergency services with the victim's GPS coordinates displayed on a map for timely delivery of medical help. The algorithm has been tested on human subjects, and the results are also included in this paper.**

Keywords*— *accelerometer, mobile phone, fall detection

I. INTRODUCTION

Falls contribute greatly to accidental injuries worldwide. Each year around 37.3 million victims receive medical attention [1]. Roughly 10% to 20% of falls can cause fractures [2]. The elderly are a demographic particularly vulnerable to falls. 30% percent of people with ages greater than 65 and 50% with ages above 80 fall each year [3]. Hospitalizations due to falls in older people are five times more than due to other causes [4]. The number of elderly people in the world is expected to reach 2 billion by 2050 [5]. With the rising cost of healthcare it is not possible to have separate caregivers for each individual, so in many cases in which a person experiences a fall, immediate help might not be available.

There are different approaches to detecting: The use of multiple accelerometers [6], wearable sensors [7], cameras [8] or vibration based detectors [9].

Accelerometers are very suitable for the detection of falls. Studies have also been done on accelerometer data sets for human activity recognition [10], [11]. The tri-axial accelerometer present in the mobile phone used has three axes: X, Y and Z. Each axis returns the acceleration in the direction of the axis in terms of the acceleration due to gravity ($1g = 9.8m/s^2$). The three axes are shown in fig. 1. In the usual position for our application, the positive Y-axis points vertically downward with the X and Z axes perpendicular to it. Hence (ideally) while stationary, the Y axis reads a value of 1g, and the X and Z axes read 0g.

A cellphone is a ubiquitous device that people are used to having with them all the time. Many new smartphones being produced have built in accelerometers. Thus cellphones can prove to be good devices for fall detection since no additional hardware is necessary. Moreover a cellphone serves as an ideal device to automatically notify contacts in case of a fall being detected. Some algorithms have been developed to use cellphones in this manner [12-14]. In this paper we present a

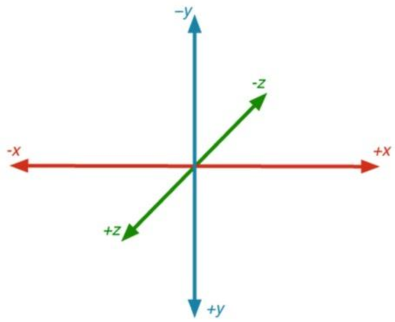

Fig. 1 Cartesian co-ordinate system of the accelerometer

computationally low cost algorithm that can detect falls in real time.

Our algorithm uses the accelerometer present in cellphones to monitor for falls, if a fall is detected the application automatically notifies predefined contacts (such as parents or emergency services) with the victim's GPS coordinates shown on a map.

II. SYSTEM ARCHITECTURE

Initially the fall detection algorithm was developed and tested on a Nintendo Wii Remote (a gaming console controller with a built in accelerometer) to allow ease of debugging, but was later ported to a Windows 8 Phone. It was tested on two models a Nokia Lumia 920 (1.5Ghz Processor, 1GB RAM) and Nokia Lumia 620 (1 GHz Processor, 512 MB RAM), which demonstrates that our algorithm can be run not only on high end consumer cellphones but also relatively less powerful ones.

The application was made using Microsoft's Windows Phone SDK .The signal processing was done in C# and the layout designed in XAML. The Math.NET open source library was used for digital filtering.

III. METHODOLOGY

The fall detection algorithm operates in a series of steps explained below:

1. Readings from the accelerometer's axes are stored at a rate of 100Hz. This data consists of a three dimensional acceleration vector ***A*** with readings from the

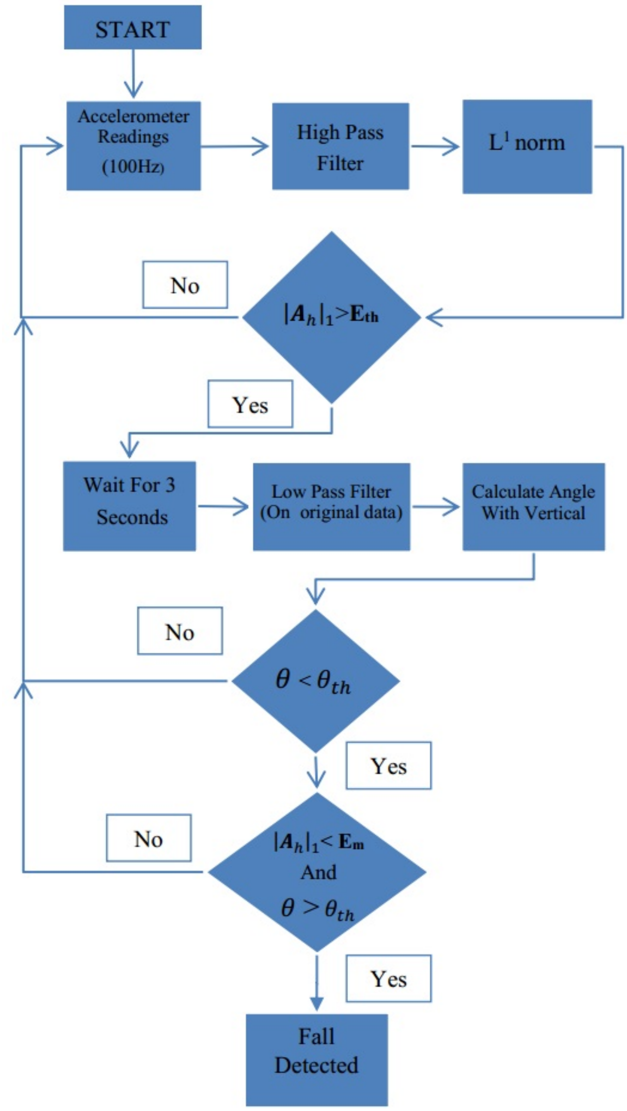

Fig. 2 Flowchart of Fall Detection Algorithm

X, Y and Z axes as its components A_x, A_y and A_z respectively.

2. Each second (i.e. for about 100 samples of data), a high pass filter is applied to the accelerometer data vector $\boldsymbol{A}$ yielding filtered values $\boldsymbol{A_h}$ and a low pass filter is also applied yielding values $\boldsymbol{A_L}$. The high pass filtered values correspond to acceleration values which are due to the movement of the cellphone (and hence the user), whereas the $\boldsymbol{A_l}$ values correspond to acceleration due to gravity. The filters are Finite Impulse Response (FIR) online, stable and causal filters implemented in the Math.NET library. FIR filters are based on Fourier series, both the high pass and low pass filters used are of the sixth order. The high pass filter has a cutoff of 2Hz, and the low pass a cutoff of 1.5 Hz.

3. The L^1 norm of $\boldsymbol{A_h}$ is calculated each second from the high pass filtered values as:

$$|\boldsymbol{A}_h|_1 = \sum_{i=0}^{99} |\boldsymbol{A}_{hx_i}| + |\boldsymbol{A}_{hy_i}| + |\boldsymbol{A}_{hz_i}| \qquad (1)$$

4. $|\boldsymbol{A}_h|_1$ directly corresponds to the amount of acceleration that a user has exerted on the accelerometer, if the value of this is above an empirically determined threshold, $\mathbf{E_{th}}$, it means the user is engaged in a high energy activity like running, jumping, exercising or has experienced a fall. The value of $\mathbf{E_{th}}$ can be adjusted in real time from within the app, as per the requirement the user to how "hard" a fall is to be detected.

5. If $|\boldsymbol{A}_h|_1$ is greater than $\mathbf{E_{th}}$ the algorithm waits for three seconds, so that if the trigger actually is associated with a fall the fall transients are over.

After three seconds, the algorithm checks the orientation of the accelerometer. The only angle of interest is the angle that the accelerometer makes with the vertical, assuming that the Y axis was initially vertical. The angle is calculated from 100 samples of $\boldsymbol{A_l}$ as:

$$\theta_i = \cos^{-1}\left(\frac{A_{lyi}^2}{\sqrt{A_{lxi}^2 + A_{lyi}^2 + A_{lzi}^2}}\right) \quad (2)$$

6. Max-wins voting is used to determine the user orientation. Each value of θ_i is compared with a certain empirically determined angle threshold, θ_{th} if θ_i is greater than the threshold, it counts as a win, and if it is less it counts as a loss. If the number of wins is greater than the number of losses, this means the user is not standing, and has fallen. In this case the algorithm proceeds to the next stage.

7. In practice, there are many falls in which the user might not be hurt, and the fall might be minor so that the user gets up and can walk again, in this case the alarm does not need to be triggered by the algorithm, so in order to prevent a false positive fall from registering in this case the algorithm again checks the value of $|\boldsymbol{A}_h|_1$ and of θ_i for five seconds, if in those five seconds the value of $|\boldsymbol{A}_h|_1$ exceeds a certain empirically determined threshold $\mathbf{E_m}$ and the value of θ_i indicates that the user is standing the alarm is cancelled and a fall is not detected.

8. At this stage we are almost certain that a fall has occurred however there is still one false positive that remains to be accounted for, and that is if a user accidentally drops his phone. To account for this we added a five second ‘grace period’ for the user in which the phone first beeps and shows a pop up notifying the user that an alarm has been sounded, the user can then cancel the alarm if he wishes, if the alarm is not cancelled within a pre-specified time, the phone will not only sound an on-spot alarm but also send the persons GPS coordinates with a message to predefined people.

IV. RESULTS

Figures. 3 to 7 show the values of raw data as well as calculated parameters for a typical fall. The annotations in Fig.3 show the activities (Sit, walk, fall), Fig. 2 to Fig.7 are the results of the algorithm steps applied to the data in Fig. 1. The accelerometer (phone) is attached to the subject’s waist using

The algorithm was tested with three test subjects wearing the cellphone at the waist (clipped to the belt) and falling in a variety of positions, the results are summarized in Table-I.

As can be seen the algorithm has a good detection rate

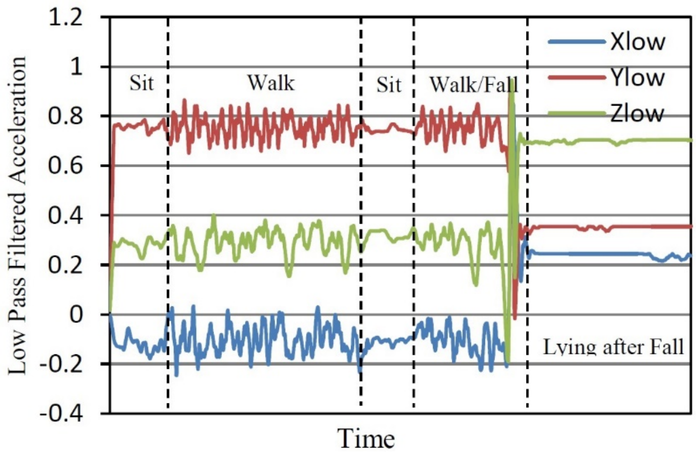

Fig. 3 Raw Acceleration Values

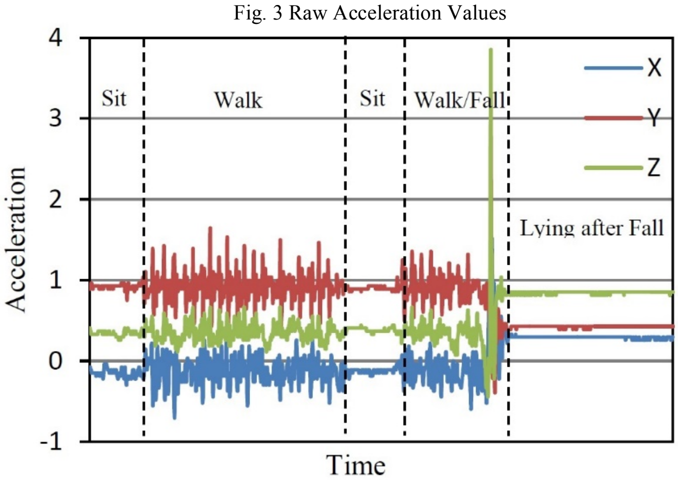

Fig. 4 Low Pass Filtered Values

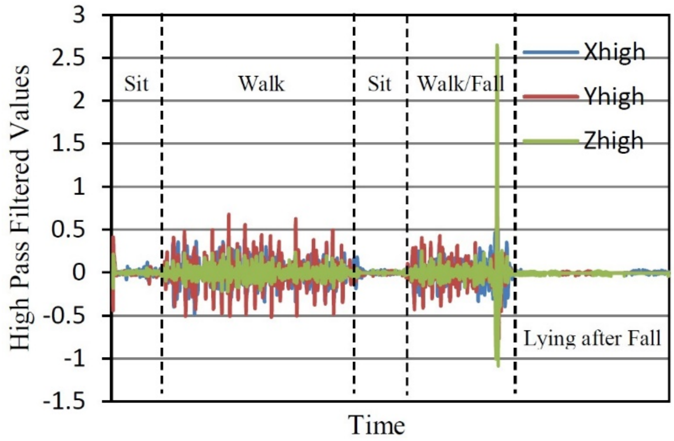

Fig. 5 High Pass Filtered Values

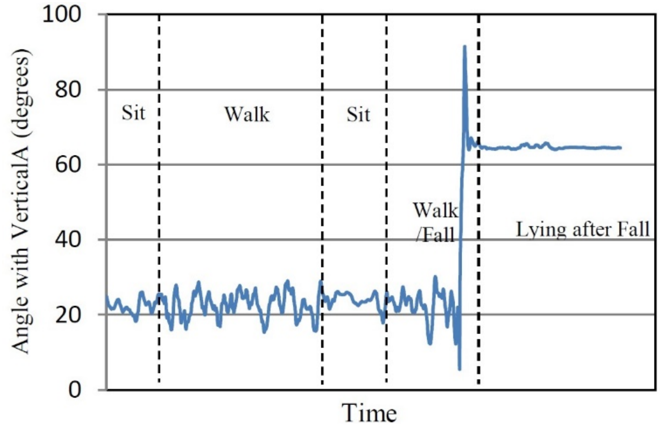

Fig. 6 Angle With Vertical

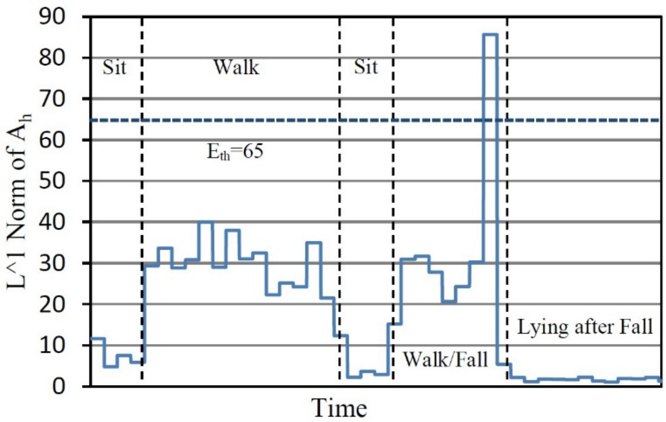

Fig. 7 L1 Norm Values

for actual falls, and has low false negatives except in the case of jumping onto a lying position in a bed or dropping the phone. This is expected since this scenario is extremely similar to a fall and it can be expected that the algorithm will not handle this case well.

However even in this case the fail-safe of the cellphone beeping first and allowing the user to cancel the alarm worked, and the subject was able to cancel the alarm in each case of a false positive.

TABLE-I EXPERIMENTAL RESULTS

FALL TESTS	NUMBER OF TESTS	CORRECT	INCORRECT	ACCURACY (%)
Fall forward/backward	20	20	0	100
Fall sideways	20	19	1	95
TESTS FOR FALSE POSITIVES	**NUMBER OF TESTS**	**CORRECT**	**INCORRECT**	**ACCURACY (%)**
Stumbling and getting up	20	18	2	85.5
Jumping up and down	20	20	0	100
Dropping Phone	20	3	17	15
Sitting Down Suddenly	20	17	3	85
Jumping and laying on bed	20	2	18	10

CONCLUSION:

The strategy to use a cellphone as a portable fall detector and for health related data collection (such as the EEG data collector implemented above) shows great potential, it can not only provide portability and efficiency but also significantly reduce the costs related with healthcare, in the future, even

More sensors (body temperature, blood glucose, blood--pressure etc.) can be integrated within such a system to provide a complete tele-health experience.

ACKNOWLEDGMENT

We are thankful for the contribution of the National ICT R&D Fund of Pakistan who provided financial support for this initiative, and are grateful to the participants who volunteered to test the fall detection algorithm.

REFERENCES

[1] World Health Organization, "Ageing, and Life Course Unit", *WHO global report on falls prevention in older age*, 2008.

[2] B.H. Alexander, F.P. Rivara, and M.E. Wolf, "The cost and frequency of hospitalization for fall-related injuries in older adults", American Journal of Public Health, 82 no.7, 1992, pp. 1020-1023.

[3] J. O'Loughlin *et al.*, "Incidence of and risk factors for falls and injurious falls among the community-dwelling elderly", American journal of epidemiology, 1993, 137: pp. 342-54.

[4] C. Cryer, "Reducing unintentional injuries in older people in England: goals and objectives for the period 1998-2010", Report to the Department of Health, London, 1998.

[5] United Nations, "World population ageing: 1950–2050." New York, Department of Economic and Social Affairs, 2002.

[6] A. Jeon *et al.*, "Development of the Algorithm for Detecting Falls during Daily Activity using 2 Tri-Axial Accelerometers", Proc. World Academy of Science, Engineering and Technology, no.61, 2012.

[7] T. Degen *et al.*, "Speedy: a fall detector in a wrist watch", In Wearable Computers, 2003. Proc. 7th IEEE International Symposium, 2005, pp. 184-187.

[8] Z. Fu *et al.*, "An address-event fall detector for assisted living applications." IEEE Trans. Biomedical Circuits and Systems, June, 2008, pp. 88-96.

[9] M. Alwan *et al.* "A smart and passive floor-vibration based fall detector for elderly", Information and Communication Technologies, 2nd, vol. 1, 2006, pp. 1003-1007.

[10] J. Parkka *et al.*, "Activity classification using realistic data from wearable sensors", Information Technology in Biomedicine, IEEE Trans. Jan., 2006, pp. 119-128.

[11] L. Bao and S. S. Intille. "Activity recognition from user-annotated acceleration data", In Pervasive Computing, Springer Berlin Heidelberg, 2004, pp. 1-17.

[12] J. Dai *et al.*, "PerFallD: A pervasive fall detection system using mobile phones", In Pervasive Computing and Communications Workshops (PERCOM Workshops), 8th IEEE Int. Conference, 2010, pp. 292-297.

[13] F. Sposaro and G. Tyson. "iFall: an Android application for fall monitoring and response", In Engineering in Medicine and Biology Society, Annual Int. Conference of IEEE, 2009, pp. 6119-6122.

[14] T. Zhang *et al.*, "Fall detection by embedding an accelerometer in cellphone and using KFD algorithm", International Journal of Computer Science and Network Security 6, no.10, 2006, pp. 277-284.

[15] Q. Li *et al.*, "Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information", In Wearable and Implantable Body Sensor Networks, 6th Int. Workshop, 2009, pp. 138-143.