International Journal of
Applied Mathematics,
Electronics and Computers
ISSN: 2147-8228 http://ijamec.atscience.org
Advanced Technology and Science
Original Research Paper

Fall Detection Using Smartphone-Based Application

Kerim Yildirim*[1], Gokhan Ucar[2] , Tahir Keskin[3], Adnan Kavak[4]

Accepted 09th August 2016

Abstract: Among the elderly persons and disabled citizens injuries caused by falls can be dangerous even leading to death. Fast response can improve the people's outcome but without knowing the accident nobody can help them. Falls are more dangerous when accidents happen while the people are alone. So it is very important to inform acquaintance and caregivers at the situation like this. Our purpose here is to inform caregivers by using a device an android-based smart phone which is available everywhere or can be get easily. The method is depends on the tri-axial accelerometer integrated with the phone. The android application -evaluated with threshold based algorithm- captures data from the accelerometer when the fall occurs. Then informs the caregivers which is defined before in the settings. The algorithm first asks the user and waits for a dedicated time if he/she is all right. If he/she is not responding the application will send an e-mail, make a phone call to inform the caregivers etc…

Keywords: *Fall detection, smart phone, android, accelerometer, threshold algorithm.*

1. Introduction

Falls can be very dangerous for elderly people and disabled citizens. Also it can affect their independent life. The estimated fall incidence is at least 30% every year for people over 75 who lives alone or not [1]. The falling frequency is considerably higher among more dependent person. According to the researchers' estimate nursing home residents' fall up to 50% and 40% of them might fall more than once each year [2]. When people get ages their reflexes are also weakens leading to involuntary accidents and this can cause significant injury increases. The number one reason for elderly people to go to the hospital is injuries caused by falls. Additionally the situation can be worse and this can cause of injury-related deaths among adults 65 years of age and older due to the falls [3].

Fuzzy logic approach, rather than a certain or binary logic, uses a logic and decision mechanism which does not have certain boundaries like human logic. With this concept coined, one of its most common implementation was in fuzzy logic-based control mechanisms. Fuzzy logic control systems do not require complete model knowledge as in the other known control systems like proportional integral. For this purpose, many design methods have been derived. Making use of medicine expert's knowledge and experience uncertain sensual data fuzzy systems are being developed currently [3-6].

As the quality of life increases day by day the population gets older and more products for fall detection are developed for elderly people to make their life safer. It also means that academic researches for fall detection are increasing. Fall detection methods can be classified into three main categories due to the sensors they use [4, 13-15]:

i. Acoustic/vibration analysis: In this method, some sensors are placed in the floor where the person live, to track sound and/or other vibrations. The sensor types are very different then the other methods. The purpose here is to detect the vibratory signature of a human fall. The signatures of fall is very different than walking, small objects falling, and other common activities [5-6].

ii. Image analysis: This method depends on cameras. Cameras are generally placed on walls and fixed. By doing so the person's activities and movements patterns are tracked. If the person remains inactive for a long time in the middle of the common path the system suspects a fall [7-10].

iii.Wearable Equipment: These are external wearable sensors and must be put on by the user. Generally the method is depends on triaxial accelerometer or gyroscope devices. The vector forces exerted by the user are tracked by the devices. If a predefined pattern or threshold is broken the device sends an alarm to initiate a inform process [4, 11-16].

Fall detection systems require both hardware and software design whether they are both on the same device or not. These requirement makes the cost increase and makes difficult to have it for everyone who needs the equipment. Also the need to the long training time is limiting greater adoption. People are looking for reliability, ease of installation/use and restriction of false alarms. Even though falls not occur every day, the system must be ready when it is needed. If all these need's cost is high, users don't won't to use it. But the main reason of not using this kind of stuff is neither high training time nor installation time; the major reason for failure is rejection by monitoring services due to a high number of false alarms [4].

2. Related Work

There are many studies made by wearable sensors on fall detection. In [13], Dai, et al. proposes a pervasive fall detection system, PerFallD, using a platform which is android based mobile phones. The design consists of two algorithms for fall detection systems using mobile phones. First one uses acceleration based method for detecting falls and second one uses a certain accessory to capture human behavior information. This algorithm is based on shape context and Hausdorff distance.

[1] Graduate School of Natural and Applied Sciences
Computer Engineering, Kocaeli University Kocaeli/TURKEY
** Corresponding Author: Email: kerim_yildirim@yahoo.com*
 # This paper has been presented at the International Conference on Advanced Technology&Sciences (ICAT'14) held in Antalya (Turkey), August 12-15, 2014.

In [4], Sporaso, et al. proposes a method using wearable devices, android based mobile phone, for detecting falls. They try to reduce rejection due to the device's poor aesthetic value and intrusiveness. They implement several fall detection algorithms to limit false positives.

Their algorithm [4] first communicates with the user, when a fall is detected, to ensure of the fall. If the user doesn't respond, then attempt to inform the members of his/her social network. If the fall is confirmed either way the system alerts an emergency service.

In [14], Yavuz, et al. an application which claimed to yield better detection result. In order to better distinguish falls from non-fall actions, they use wavelet decomposition as a feature extraction method. If a fall detected, the application can inform predefined caregivers about the event and the location of the event. Also they added a panic button to prevent true negative situations.

In [15], Cao, et al. proposes E-FallD. They uses threshold to detect falls, which is adaptive based on user's message such as: height, weight, sex and level of age E-FallD uses the threshold algorithm for fall detection. It provides personalized service in order to better distinguish falls from non-fall actions. When a fall is detected, the system can inform caregivers by sending text message. And if it is a false-positive situation, the user can cancel it by pressing a button.

In [16], Tacconi, et al. also proposes a fall detection system using a Smartphone application able to analyze in real-time. They use tri-axial accelerometer sensor of the smartphone to catch the falls. And they designed *elderly-oriented* user interface to make it easier to use.

As you can see there are several similar systems in this area. These systems are all designed using a platform which is based on Android smart phone.

We also choose The Smartphone Platform in our design. Because it is an open platform which allows the full programmability of all the relevant software components: sensor management, power management, data storage management, connectivity management (Internet connection, SMS, etc.).

Our design is to detect falls in real time for elderly and disabled persons. Data are captured from phone's tri-axial accelerometer sensor. The designed fall detection system, based on threshold, has a user friendly interface.

3. Fall Detector

3.1. Fall Behavior

One can do activities like walking, standing, jumping etc. in a daily life. However the circumstances of these activities are different than a fall.

A fall must start with a short free fall period and this causes the acceleration's amplitude to drop significantly below the threshold. Also this represents the period of time when the actual fall is occurred. When the fall ends it causes a spike in the graph. We can assume a fall when the amplitude crossing the upper threshold value. If someone is seriously injured due to the fall they usually remain still on the ground for a while. This can be seen by the flat line at the graph. You can see a typical fall in Fig. 1.

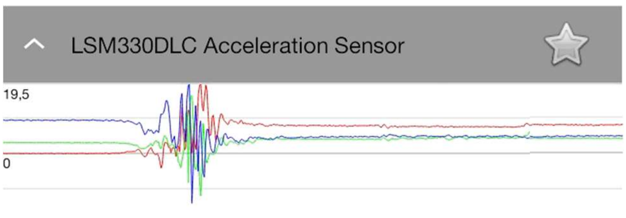


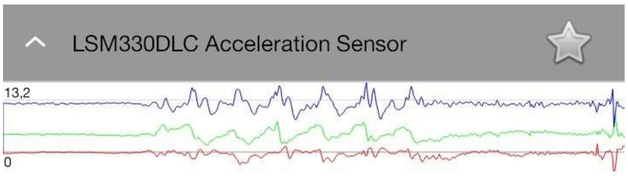
Figure 1. Graph of a typical fall

As you can see from the Figures below, walking, standing, jumping, sitting and climbing stairs activities' pattern are different. To show these differences we captured the acceleration sensor's values.

Walking pattern can be seen from Fig. 2. While walking, acceleration sensor values changes are not rapid. There are regular and very similar changes in values. Also the changes are not crossing upper or lower threshold values.

Figure 2. Graph of a typical walking

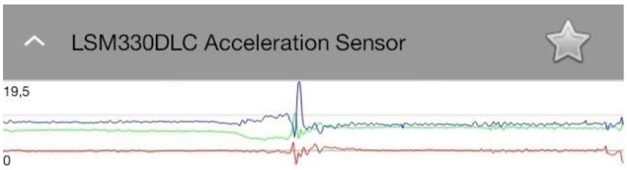
There is a significant change in acceleration values while sitting. The curve is sharp and stabilizes afterwards. But we can eliminate this situation easily because the change is not greater then



threshold (due to the calculation), leaving fast and sharp sitting.

Figure 3. Graph of a typical sitting

Jumping is a little tricky and easily can be seen as a fall. As you can see from Fig. 4 the graph is alike with Fig. 1. Jumping causes big spike and big drop and this can be seen at the graph below. So we can conclude that jumping's behavior is similar with a fall -not



for slow motion jumps- and can be easily seen as a fall. Our algorithm also gets this 7 false detection as falls out of 25.

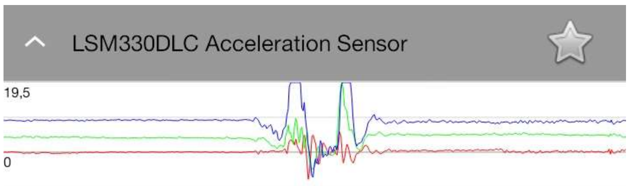


Figure 4. Graph of a typical jumping

While sitting one have to get down slowly and this will cause a spike at the graph. As you can see from Fig. 3, sitting behavior is different from a fall and can be easily distinguished. In our tests we have no FN or FP.

Walking and climbing stairs outputs are similar. There are no sharp and rapid changes in the graph. The calculation which is made of using the accelerometer's three axial values will not pass threshold value. So our test results have no FP or FN.

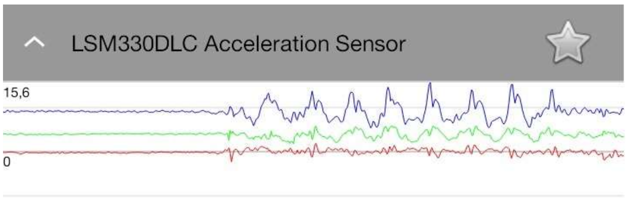


Figure 5. Graph of a typical climbing stairs

3.2. Detecting a Fall

A fall can be detected if the amplitude crosses the lower and upper thresholds in the set duration period of time. On the other hand to detect a fall just using the threshold values can lead a number of false positives, because some activities can imitate similar results as fall like jumping, falling the phone etc. Also a fall can occur leaving the user unharmed. To prevent these false alarms we added an extra screen asking if the user is ok.

A short timer of 10 seconds is started if a fall is suspected. This timer allows the user to let the application know he/she is ok, or a dropped phone to be picked up before it calls caregivers or emergency. If the user pushes the OK button the application restarts. If not application calls caregivers and sends a message “I’ve fallen and I can’t get up”. If the user chooses to call emergency, applications starts the call. By the way if the user chooses “Do Nothing”, application does nothing.

There are several algorithms which have been defined in literature using Sum Vector (SV). Algorithms’ performances are defined in terms of impact detection capability and identification of the false positive events. These algorithms make a single or double threshold or detect the subject speed before the fall occurs.

Because it requires low computational power, we have decided to use a simple single-threshold algorithm to implement fall detection. Also in our opinion it is more suitable method for real-time applications.

To test our algorithm’s performance we did some measurements listed in Table I. Also to make comparisons we did same measurements with other algorithms, i-Residence Fall Detection [17] and T3LAB Fall Detector [18]. We choose these algorithms, because the way to detect a fall is similar to our method and also it is easy to get these programs and available on the web.

4. System Design and Implementation

4.1. Hardware

The system is designed to run on any Android device that has accelerometer, and has ability to make calls and send SMS. Yet we will give hardware specifications of the phone we used for test; Samsung GT-I8190 known as Galaxy SIII Mini. The CPU of the phone is Dual Core running at 1 GHz. Its dimensions are 121.6 mm x 63 mm x 9.9 mm and weighs 120 grams. The touch screen has a 480 x 800 resolution. It has 1 GB RAM and 8 GB ROM memory. The phone has many sensors however the most important is that it has a tri-axis accelerator for getting acceleration data.

4.2. Software

We design our application on the Android 2.2 Platform, it could also work on higher versions of Android Platform. Android is an open source framework designed for mobile devices and it has a powerful Software Development Kit (SDK) based on Java Framework. The Android SDK provides libraries needed to interface with the hardware and deploy an Android application. And Android applications are written in Java and run on the Dalvik virtual machine. Of course, we need to install Eclipse + ADT Plugin on the computers. It is available in http://developer.android.com/sdk.

4.3. Equation

This paper mainly concerned about the designing and realization of the fall detection system, so the algorithm we choose is threshold algorithm.

Our algorithm is based on three axis accelerations. Accelerations in X-axis, Y-axis and Z-axis are represented by A_x, A_y and A_z. And the resultant acceleration A_{sum} can be obtained by equation (1):

$$A_{sum} = \sqrt{A_x^{\,2} + A_y^{\,2} + A_z^{\,2}} \quad (1)$$

4.4. Screen Capture of The Application

The screen shots of the application and short description of the screen is given below:

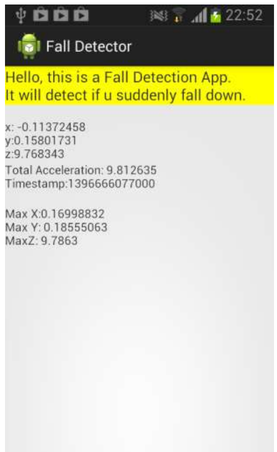


Figure 6. This is the start screen; and the instant values of X, Y, Z coordinates received from the accelerometer sensor is shown.

After starting the application it continuously listens to the accelerometer sensor, when the application detects a fall it goes to Fig. 7.

Figure 7. At this screen; when fall is detected this screen pops up and the user is asked whether or not he/she is ok.

If the user selects “Yes”, the application will go back to Fig. 7. If the user selects “No”, the application will go to Fig. 9. If the user gives no response in 10 seconds, the application will go to Fig. 8.

Figure 8. At this screen; if the user -in Fig. 7- selects “NO” or gives no response in 10 seconds, application asks the user what to do.

If the user gives no response in 10 seconds the application will call the emergency and send an SMS to the caregiver.
If the user selects “Call + Text Caregiver”, the application will call the caregiver as shown in Fig. 9 and send an SMS to caregiver as shown in Fig. 10.
If the user selects “Call Emergency”, the application will call emergency as shown in Fig. 11. and send SMS to caregiver as shown in Fig. 10.
If the user gives “NO” response in 10 seconds, the application will go to Fig. 6.

Figure 9. The screen shot of the call to inform the caregiver.

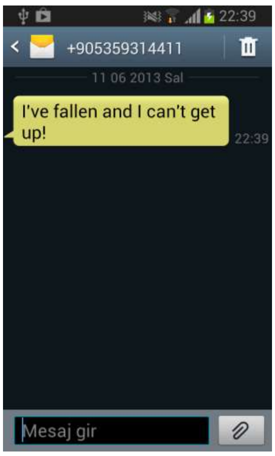


Figure 10. The screen shot of the SMS sent to the caregiver.

Figure 11. The screen shot of Emergency Call.

5. Detection Performance

We measure the detection performance in terms of false negative (FN) and false positive (FP). False negative happens when a fall occurs but the device misses it. False positive happens when the device alarms a fall but it does not occur. In general, the lower the both FN and FP are, the better the performance is. The experiment is made by five people each one done it five times and the result is shown at Table 1.

Table 1. The result of the applications for fall detection

	Our Algorithm		**i-Residence**		**T3LAB**	
	FN	**FP**	**FN**	**FP**	**FN**	**FP**
FALL	9/25	0	0	0	10/25	0
WALKING	0	0	0	0	0	0
SITTING	0	0	4/25	0	0	0
JUMPING	0	7/25	23/25	0	0	0
LYING	0	0	0	0	0	0
CLIMBING STAIRS	0	0	21/25	0	0	0

Table Footnote

The results show that our algorithm has missed 9 falls out of 25 falls, better then T3LAB, but i-Residence has no FN.
All three algorithms have no FN or FP for walking and lying.
While testing of algorithms for sitting and climbing stairs our algorithm and T3LAB have same performance (no FN or FP) and better than i-Residence. The performance of i-Residence is 4/25 FN for sitting and 21/25 FN for climbing stairs.
By looking at jumping test we can say that T3LAB has better results. Our algorithm has 7 FP out of 25.
Algorithms can’t easily distinguish jumping from fall events, because the characteristics of both event are very similar, this can be seen from Fig. 1 and Fig. 4.
By looking at the results we can agree that i-Residence is the best algorithm for detecting the “Fall” events. On the other hand, just by looking at the “Fall” events, we cannot say that i-Residence algorithm is the best detector for all type of

events. This can be seen from other values at Table I.
So for total success, our algorithm is slightly the best detector among three algorithms and more reliable then i-Residence according to our experiments.

6. Conclusion

Some events can be seen as a fall by the system such as jumping, falling the phone, violently answering then ending a call etc. can break the thresholds and change position. We chose not to refine our algorithm to handle these cases for two reasons as in [4]. First, eliminating these cases may result in decreased accuracy of detecting actual falls. Second, we are assuming that the user wants an alert if the phone was accidentally dropped and the user is given the option to cancel the alert.
In future, we are planning to add some machine learning algorithms into our method and extend the test subjects quantity with different age groups.

References

[1] Duthie E (1989*) Falls.* Med Clin North Am 73:1321–1335

[2] Tideiksaar R (1998) *Falling in old age: prevention and management*, 2nd edn. Springer, Berlin

[3] Department of Health and Human Services. *Fatalities and injuries from falls among older adults - united states*, 1993-2003 and 2001-2005. pages 1221–1224, November 2006. Morbidity and Mortality Weekly Report.

[4] Frank Sposaro, Gary Tyson (2009*), iFall: An Android Application for Fall Monitoring and Response*, 31st Annual International Conference of the IEEE EMBS Minneapolis, Minnesota, USA, September 2-6, 2009

[5] Majd Alwan, Prabhu Jude Rajendran, Steve Kell, David Mack, Siddharth Dalal, Matt Wolfe, and Robin Felder. *A smart and passive floor-vibration based fall detector for elderly.*

[6] Mihail Popescu, Yun Li, Marjorie Skubic, and Marilyn Rantz. *An acoustic fall detector system that uses sound height information to reduce the false alarm rate.* 30th Annual International IEEE EMBS Conference, August 2008.

[7] Tracy Lee and Alex Mihailidis. An intelligent emergency response system: preliminary development and testing of automated fall detection. Journal of Telemedicine and Telecare, 11(4):194–198, 2005.

[8] Shaou-Gang Miaou, Pei-Hsu Sung, and Chia-Yuan Huang. *A customized human fall detection system using omni-camera images and personal information.* pages 39–41. Proceedings of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference, April 2006.

[9] Hammadi Nait-Charif and Stephen J. McKenna. Activity summarization and fall detection in a supportive home environment. 2004.

[10] Caroline Rougier and Jean Meunier. Demo*: Fall detection using 3d head trajectory extracted from a single camera video sequence*.

[11] K Doughty, R Lewis, and A McIntosh. *The design of a practical and reliable fall detector for community and institutional telecare*. Journal of Telemedicine and Telecare, 6(1):150–154, 2000.

[12] Thomas Riisgaard Hansen, J. Mikael Eklund, Jonthan Sprinkle, Ruzena Bajcsy, and Shankar Sastry. *Using smart sensors and a camera phone to detect and verify the fall of elderly persons*. European Medicine, Biology and Engineering Conference (EMBEC 2005), November 2005.

[13] Jiangpeng Dai, Xiaole Bai, Zhimin Yang, Zhaohui Shen, Dong Xuan (2010), *Mobile phone-based pervasive fall detection*, Pers Ubiquit Comput, DOI 10.1007/s00779-010-0292-x, Springer-Verlag London Limited 2010.

[14] Yavuz, G., Kocak, M., Ergun, G., Alemdar, H. O., Yalcin, H., Incel, O. D., & Ersoy, C. (2010, November). *A smartphone based fall detector with online location support*. In International Workshop on Sensing for App Phones; Zurich, Switzerland (pp. 31-35).

[15] Yabo Cao, Yujiu Yang, WenHuang Liu (2012), E-FallD : *A Fall Detection System Using Android-Based Smartphone*, 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012).

[16] Carlo Tacconi, Sabato Mellone, Lorenzo Chiari,(2011*) Smartphone-Based Applications for Investigating Falls and Mobility*, 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops.

[17] https://play.google.com/store/apps/details?id= at.spantec.securemotion.falldetector.

[18] https://play.google.com/store/apps/details?id=com.ec.fall detection.

