31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

iFall: An Android Application for Fall Monitoring and Response

Frank Sposaro,
sposaro@cs.fsu.edu
Florida State University

Gary Tyson
tyson@cs.fsu.edu
Florida State University

Abstract— **Injuries due to falls are among the leading causes of hospitalization in elderly persons, often resulting in a rapid decline in quality of life or death. Rapid response can improve the patients outcome, but this is often lacking when the injured person lives alone and the nature of the injury complicates calling for help. This paper presents an alert system for fall detection using common commercially available electronic devices to both detect the fall and alert authorities. We use an Android-based smart phone with an integrated tri-axial accelerometer. Data from the accelerometer is evaluated with several threshold based algorithms and position data to determine a fall. The threshold is adaptive based on user provided parameters such as: height, weight, and level of activity. The algorithm adapts to unique movements that a phone experiences as opposed to similar systems which require users to mount accelerometers to their chest or trunk. If a fall is suspected a notification is raised requiring the user's response. If the user does not respond, the system alerts pre-specified social contacts with an informational message via SMS. If a contact responds the system commits an audible notification, automatically connects, and enables the speakerphone. If a social contact confirms a fall, an appropriate emergency service is alerted. Our system provides a realizable, cost effective solution to fall detection using a simple graphical interface while not overwhelming the user with uncomfortable sensors.**

I. Introduction

As age related changes in reaction time and balance reduce the capabilities of people, the likelihood of a fall leading to significant injury increases. Not only are fall related injuries the number one reason for emergency room visits, they are also the leading cause of injury-related deaths among adults 65 years of age and older [19]. Every year, more than 11 million people fall [4]. In 2005, unintentional falls accounted for an estimated 56,423 hospitalizations and 7,946 related deaths in the United States [24]. Many of these deaths are a result of a "long-lie," an extended period of time where the victim remains immobile on the ground [3]. Just the simple fear of a long-lie or falling can lead to the worsening of one's mental health, isolation, and the general degradation of his/her quality of living [21], [7].

Current systems are available that attempt to reduce the long-lie period by alerting emergency services when a fall has been detected. These systems commonly use one of three methods for classifying a fall:

1) Acoustic/vibration recognition: This is achieved by having a device, usually implanted in the floor, monitor sound and other vibrations. It listens for the vibratory signature of a human fall, which is vastly different from the signatures of walking, small objects falling, and other common activities [1], [22].
2) Image recognition: By using overhead cameras in a fixed location, one can track objects and learn movement patterns. The system adapts to the locations in which a single human enters/exits the room and remains inactive (lying/sitting on bed/chair). Common paths from entry points to inactive areas are then traced and remembered. It suspects a fall if a person becomes inactive in the middle of a common path [13], [15], [23], [16].
3) Worn Devices: These systems require the user to wear external sensors. The devices track the vector forces exerted on the user. Usually these devices are a tri-axial accelerometer or gyroscope. If a specific pattern or threshold is broken, the device alerts a wireless receiver, which then alerts emergency contacts [7], [26], [8].

The majority of fall detection systems require some application specific hardware and software design. This increases cost and limits the commercial viability to the wealthiest, or most impaired, users. Many also have significant installation and/or training times, also limiting greater adoption. Despite implementation differences, all designs have the same requirements: reliability, ease of installation/use, and restriction of false positives [7]. Falls are often sparse with months between occurrences, thus the system must always be ready and accurate. If installation costs or training time is high, users will reject the system. However, the major reason for failure is rejection by monitoring services due to a high number of false alarms [17], [20].

We propose a low priced system that is well suited to all the requirements by using existing mainstream technologies that are reliable and ubiquitous. Our approach is to use the number one fasting growing device which billions of people already own, a programmable cellular phone [25]. Using existing cell phone technology not only reduces the cost to the patient, it also exploits a greater range of communication capabilities and integrated hardware and software features. Touch screen response and voice recognition, common to smart phones, provide a reliable interface for the user. By using interfaces that are similar to applications the user frequently uses, the rare interaction with the fall detection software should be familiar. Cell phones are also more discrete than a dedicated monitor device, which will reduce rejection due to the device's poor aesthetic value and intrusiveness [7]. To limit false positives we implement several fall detection algorithms and two stages of communication.

978-1-4244-3296-7/09/$25.00 ©2009 IEEE

When a fall is detected, we first communicate with the user. If the user does not respond, we then attempt to contact members in his/her social network. If both fail or the social contact confirms a fall, the system alerts an emergency service.

II. Materials and Methods

A. Operating System

We chose to use the Android software stack produced by Google. Android is an open source framework designed for mobile devices that packages an operating system, middleware, and key applications [10]. The current prototype is deployed on the G1 by manufactured by HTC [6]. The Android SDK provides libraries needed to interface with the hardware and make/deploy an Android application [11]. Applications are written in Java and run on the Dalvik virtual machine. Android uses a SQLite database to store persistent data.

Unlike dedicated systems, our software is intended to integrate with the phone's existing applications. Our application, *iFall*, must share resources with the other apps. To make for a pleasant integration, iFall runs as inconspicuously as possible while using limited resources. We launch a background service that constantly listens to the accelerometer. Only when the algorithm described in the following section suspects a fall will the service wake up and interrupt the user. If the user responds, the previous activity is restored and iFall will sleep again. By only waking up the activity when a fall is suspected or requested by the user, we allow applications to run on top of iFall while we minimize our memory consumption and user interaction.

B. Fall Detection

Activities of Daily Living (ADL) are normal activities such as walking and standing. The forces exerted during ADL are usually different than the forces during a fall. By taking the root-sum-of-squares of the accelerometer's three axials, we are able to determine the acceleration [3]. A fall typically starts with a short free fall period. This causes the acceleration's amplitude to drop significantly below the 1G threshold [3]. This represents the period of time when the actual fall is taking place. The fall must stop and it causes a spike in the graph. The amplitude then crossing an upper threshold suggests a fall. Typically the minimum value for the upper threshold is around 3G [5]. If a person is seriously injured in a fall they usually remain on the ground for a period of time. This is characterized by the 1G flat line at the end of the graph in figure 1. All events occur within a short duration.

If the amplitude crosses the lower and upper thresholds in the set duration period a fall is suspected. However, relying strictly on this method would produce an intolerable number of false positives since certain ADL and the upper threshold can overlap [2]. We refine the algorithm by taking position into consideration. The assumption is a fall can only start from an upright position and end in a horizontal position [14]. Thus the difference in position before and after the fall is close to 90 ° [26]. A fall is only suspected if both thresholds are crossed within a duration and the position is changed. Dropping the phone is a frequent motion that resembles a suspected fall. Also a fall may occur but, be minor leaving the user unharmed. To prevent these false alarms we add one more stage to the process, recovery.

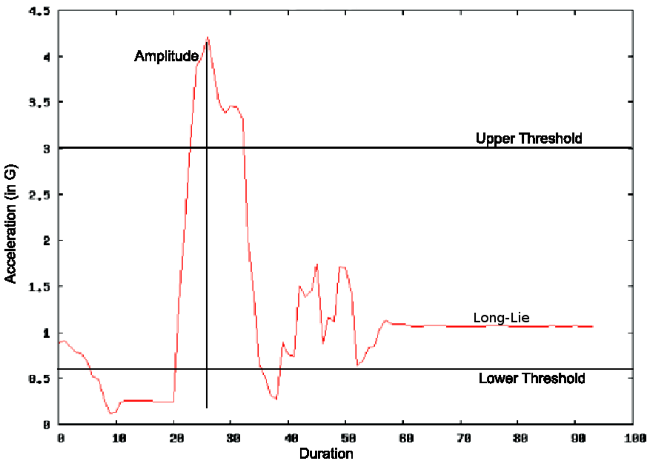

Fig. 1. A classic fall example. Denoted by amplitude crossing the lower and upper threshold, followed by a long-lie.

If a fall is suspected, we start a short timer. This timer allows a fallen user to regain an upright position or a dropped phone to be picked up. If the original position is restored within the time limit the algorithm is reset. If the timer expires and position is not restored, we assume the phone/user is lying on the ground [9]. It then emits a prompt that requires the user to respond within a short time window. A fall is confirmed if the user does not respond. This allows users to reduce the number of false positives. An alert only sends when a fall is confirmed.

C. Application Features

The iFall application is designed to be simple to use. To achieve this, we severely limit the number of buttons and options available to the user. The main screen consists of one button, light background, and large, bold lettering. The button starts and stops the fall monitor while the label displays the state. The fall monitor is implemented as a low-powered, Android service. A service allows the fall monitor to constantly run in the background. When the monitor suspects a fall, an intent is sent to an iFall activity. This wakes up the application and attempts to get the user's attention by repeatedly vibrating, flashing LEDs and screen, and playing an audio message. The app prompts the user with a simple pop-up window telling them to press an on-screen button or verbally state if they require help. A canceled alert closes iFall and the interrupted activity is restored. This gives users the opportunity to eliminate false positives [20], [8].

The iFall application has additional methods to reduce the number of false positives. We allow the amplitude's upper threshold described in the 'Fall Detection' section to be variable. The application displays a small list of configuration options when the phone's *Menu* key is pressed. One option is to adjust the sensitivity, the capacity to detect a fall [17]. So the less sensitive, the higher the upper threshold is. Dynamically raising the upper threshold allows the app or user to adjust in the case where an excessive number of falls are suspected. Given information such as age, weight, height, and level of activity are also factored into the equation [15], [26].

The other option under the application's menu is *Manage Contacts*. This allows the user to add social contacts to his/her iFall, emergency contact list. Using social contacts to confirm a fall before alerting an emergency service is another method for filtering false positives. When a fall is confirmed, every contact in the iFall emergency list is sent a SMS message [18]. This message states a fall was detected, the given time, GPS coordinates, and password. It also asks the contact to text the fallee the given password. When texted, an audio message is played on the fallee's phone, connection with the social contact is automatically made, and the phone is placed on speaker. Enabling bidirectional voice communication between the fallee and social contact greatly reduces the number of false positives [20]. The dedicated emergency services are only notified when a social contact also confirms the fall, or in the case that no social contacts text the fallee.

III. Challenges

Using smart phone technology for fall detection has numerous advantages in cost and capability of the system. However, leveraging an existing system does pose challenges that single use detectors can avoid. One advantage of using a smart phone, is that the user is more likely to carry the phone throughout the day since it is seen as indispensable in daily living, whereas users may forget to wear special micro sensors [27]. Unfortunately, it may be difficult to convince users to mount the phone to various body parts in order to improve fall detection rate [12]. Instead, the software must dynamically adjust to different methods of carrying the phone (e.g., in the purse, pants or shirt pocket, or on a belt or neck clip). This requires the software to classify acceleration parameters of general use to identify the correct parameters for the fall detection logic.

To adapt for different carrying methods, we dynamically adjust the upper threshold and starting position. If the phone is carried on more accelerated body parts, such as the arms, the level of activity is automatically risen. This causes the upper threshold to be greater [12]. Likewise, more stationary spots like the trunk will lower the threshold [3]. To account for the different orientations in which the phone may be held, such as vertically or horizontally, we dynamically adjust the starting position. If the phone is resting for an extended period of time with 1G acceleration, we designate that to be the starting position. This allows the position to be dynamically set as the user interacts with the phone throughout the day.

Figure 2 graphs the walking/running activity. Running's amplitude can break the lower and upper threshold. If the user suddenly stops, it can cause an extended period of 1G acceleration. These events together suggest a fall. However, a prompt will not be given because the phone's starting and ending positions are the same. Figure 3 graphs the sitting and standing activity. This activity changes the phone's position. However, sitting and standing's acceleration will not usually break the upper threshold. Both experiments were performed while the phone was in the user's front pant's pocket.

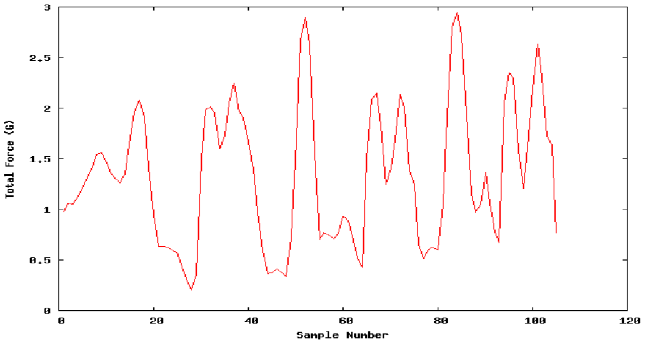

Fig. 2. Accelormeter readings while running. Crosses lower and upper threshold while posture position remains upright.

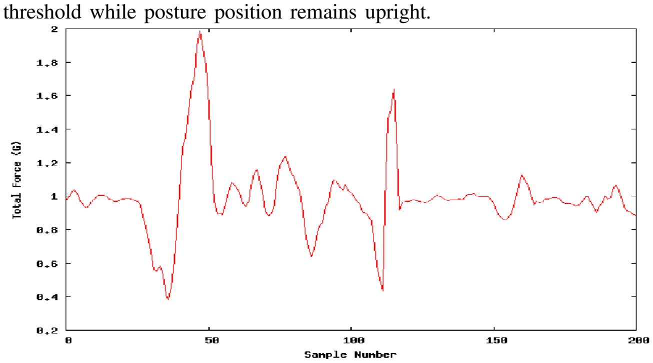

Fig. 3. Accelormeter readings while sitting and standing. Posture position changes while thresholds are not typically crossed.

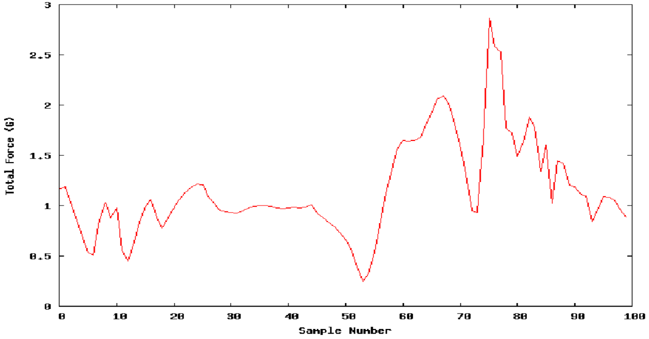

Fig. 4. Accelermeter readings while violently picking up and setting down the phone.

Certain interactions such as violently answering then ending a call (see figure 4) or dropping the phone can break the thresholds and change position. We chose not to refine our algorithm to handle these cases for two reasons. 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. The user is given the option to cancel the alert and/or adjust the upper threshold if either case becomes too frequent.

IV. Conclusion

Our system provides a viable solution to increase fall detection among people. Using existing, mass marketed technologies will limit cost making it available to the majority of the public. Implementing proven fall detection algorithms makes the system highly reliable. Reliability and reduced number of false positives mean greater adoption by emergency services. The importance of the cell phone in everyday life decreases the chances of being forgotten. Everyday interaction with the phone makes the interface more familiar to the user. A cell phone is also less intrusive than dedicated devices. The familiar interface, non-intrusiveness, and affordability leads to less rejection from users. By combining cheap hardware and open source software, we hope to provide a realistic answer to reducing the long-lie period for the elderly.

A. Future Work

Our current effort is to deploy these devices in the field. We are working with medical and geriatric professionals to create an interface that works well with the elderly. Observing interactions will help refine the user interface, gain real world power statistics , and mine sample data.

The flexibility of the Android platform along with the phone's hardware capability allows this system to be extended in numerous ways. Bluetooth support could allow iFall to gather additional data readings from micro-sensors embedded in articles of clothing [18]. Ideally, a sensor would be embedded in head or eye wear due to the fact that the head is the most reliable location for fall detection using threshold based algorithms [12]. The system could also use image support from mounted, bluetooth cameras as described in [15], [23], and [16].

The system could also use the Wi-Fi connection to log the data readings on a server. More sophisticated pattern matching algorithms can be then be executed [27]. Efforts are being made to build a database of common ADL readings [20]. This information can be exploited in attempts to classify what type of action the user is performing based on pattern matching techniques.

Ideally, this system can be extended beyond geriatrics. Vigorous activities such as running, biking, and extreme sports all increase participants' chances of falling. In addition, we would like to adapt our algorithm to detect when the user is in a vehicle. By setting a high upper threshold, along with GPS, we can then monitor if the user has been in an accident.

References

[1] 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.

[2] A.K. Bourke and G.M. Lyons. A threshold-based fall-detection algorithm using abi-axial gyroscope sensor. (30):84–90, December 2008.

[3] A.K. Bourke, J.V. O'Brien, and G.M. Lyons. Evaluation of a threshold-based tri-axial accelerometer. (26):194–199, September 2006.

[4] Garret Brown. An accelerometer based fall detector: Development, experimentation, and analysis. Technical report, July 2005. EECS/SUPERB.

[5] Jay Chen, Karric Kwong, Dennis Chang, Jerry Luk, and Ruzena Bajcsy. Wearable sensors for reliable fall detection. pages 3551–3554. Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, September 2005.

[6] HTC Corporation. G1 product specification, 2009.

[7] 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.

[8] 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.

[9] J.Y. Hwang, J.M. Kang, Y.W. Jang, and H.C. Kim. Development of novel algorithm and real-time monitoring ambulatory system using bluetooth module for fall detection in the elderly. pages 2204–2207, September 2004.

[10] Google Inc. Android. www.android.com.

[11] Google Inc. Android developers. developer.android.com.

[12] Maarit Kangas, Antti Konttila, Ilkka Winblad, and Timo Jamsa. Determination of simple thresholds for acceleroetry-based parameters for fall detection. pages 1367–1370. Proceedings of the 29th Annual International Conference of the IEEE EMBS, August 2007.

[13] 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.

[14] Suhuai Luo and Qingmao Hu. A dynamic motion pattern analysis approach to fall detection. 2004.

[15] 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.

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

[17] N Noury, A Fleury, P Rumeau, A.K. Bourke, G.ó Laighin, V Rialle, and J.E. Lundy. Fall detection - principles and methods. pages 1663–1666. Proceedings of the 29th Annual International Conference of the IEEE EMBS, August 2007.

[18] M.N. Nyan, Francis E.H. Tay, M. Manimaran, and K.H.W. Seah. Garment-based detection of falls and activities of daily living using 3-axis mems accelerometer. *Journal of Physics*, pages 1059–1067, 2006.

[19] 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.

[20] G Pérolle, P Fraisse, M Mavros, and I Etxeberria. Automatic fall detection and activity monitoring for elderly. Sponsored by the CRAFT project.

[21] EW Peterson, CC Cho, and ML Finlayson. Fear of falling and associated activity curtailment among middle aged and older adults with multiple sclerosis. (13):1168–1175, 2007. Multiple Sclerosis.

[22] 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.

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

[24] K.E. Thomas, J.A. Stevens, K. Sarmiento, and M.M. Ward. Fall-related traumatic brain injury deaths and hospitalizations among older adults - united state, 2005. *Journal of Saftey Research*, (39):269–272, May 2008.

[25] Ran Wei and Ven-Hwei Lo. Staying connected while on the move. 8(1):53–72, 2006.

[26] G Williams, K Doughty, K Cameron, and D.A. Bradley. A smart fall and activity monitor for telecare applications. volume 30, pages 1151–1154. Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1998.

[27] Tong Zhang, Jue Wang, Ping Liu, and Jing Hou. Fall detection by embedding an accelerometer in cellphone and using kfd algorithm. *IJCSNS International Journal of Computer Science and Network Security*, 6(10):277–284, October 2006.