PerFallD: A Pervasive Fall Detection System Using Mobile Phones

Jiangpeng Dai [*†], Xiaole Bai [†], Zhimin Yang [†], Zhaohui Shen [‡] and Dong Xuan [†]

[*] *School of Computer Sci. & Engr.*
Southeast University
Nanjing, Jiangsu, China
jpdai@seu.edu.cn

[†] *Dept. of Computer Sci. and Engr.*
The Ohio State University
Columbus, Ohio, USA
{jpdai, baixia, yangz, xuan}@cse.ohio-state.edu

[‡] *Division of Physical Therapy*
The Ohio State University
Columbus, Ohio, USA
shen.119@osu.edu

***Abstract*—Falls are a major health risk that diminish the quality of life among elderly people. With the elderly population surging, especially with aging "baby boomers", fall detection becomes increasingly important. However, existing commercial products and academic solutions struggle to achieve pervasive fall detection. In this paper, we propose utilizing *mobile phones* as a platform for pervasive fall detection system development. To our knowledge, we are the first to do so. We design a detection algorithm based on mobile phone platforms. We propose *PerFallD*, a pervasive fall detection system implemented on mobile phones. We implement a prototype system on the Android G1 phone and conduct experiments to evaluate our system. In particular, we compare PerFallD's performance with that of existing work and a commercial product. Experimental results show that PerFallD achieves strong detection performance and power efficiency.**

***Keywords*-Fall detection; Mobile phones; Accelerometer**

I. Introduction

A. Motivation

Falls are a major health hazard for elderly people [1] and also a major obstacle to their independent living [2]. The estimated fall incidence for both institutionalized and independently living people over 75 is at least 30% every year [3]. The frequency of falling is considerably higher among more dependent elderly. Researchers estimate that up to 50% of nursing home residents fall each year and more than 40% of these might fall more than once [4]. Falls not only cause physical injury such as many disabling fractures [5]; they also have dramatic psychological consequences that reduce elderly people's independence [6]. This situation deteriorates as the elderly population surges. According to a report from the U.S. Census Bureau, there will be a 210% increase of the population aged 65 and over within the next 50 years, in part due to aging "baby boomers" [7].

The considerable risks of falls and the substantial increase of the elderly population motivate both the development of commercial products and academic research on fall detection. A typical fall detection system has two major functional components: the detection component and the communication component. As their names imply, the detection component detects falls and the communication component communicates with emergency contact after fall detection. Brickhouse [8] provides a typical commercial fall detection system. The system consists of a portable sensor and a tele-assist base.

The major problem with existing commercial products and academic research is that they have deficiencies that hinder *pervasive* fall detection. Consider the aforementioned product as an example. The base must be installed somewhere indoors and the portable sensor must be attached to a belt at the waist. Once the base receives the signal from the sensor indicating a fall, it can automatically communicate with a preset emergency contact using the fixed phone. However, the maximum distance between the sensor and the base is limited. Fall detection can only be conducted within a small indoor environment and elderly people may easily forget to bring the sensor with them, as it is an extra device that they seldom use in daily life. Furthermore, these products are expensive. The aforementioned system costs $199.95 for the devices and $419.40 per year for monitoring service [8].

B. Our Contributions

In this paper, we propose utilizing *mobile phones* as the platform for pervasive fall detection system development, as they naturally combine the detection and communication components. To the best of our knowledge, we are the first to do so.

As self-contained devices, mobile phones present a mature hardware and software environment for pervasive fall detection system development. Mobile phone-based fall detection systems can function almost everywhere since mobile phones are highly portable, all necessary components are already integrated therein, and their communication services have vast coverage. One might argue that elderly people may not accept such mobile phones. However, we would like to point out that elderly people may prefer to have a single phone with self-contained fall detection functionality than to carry a separate fall detection device on their bodies. In addition, more recent data illustrate the increasing popularity of these phones. The minimum requirements for such a mobile phone platform are the presence of a simple sensor, e.g., an accelerometer. Currently, many phones, especially smartphones, contain multiple types of sensors, including accelerometers. Such phones are very popular and thoroughly accepted in society. Their popularity is likely to continuously increase in the near future due to decreasing prices. Recently, several leading telecommunication companies such as AT&T have made available affordable smartphones [9] whose features are similar to those of high-end models, in addition to cheaper service plans [10].

We summarize the contributions of this paper as follows.

– We propose utilizing *mobile phones* as the platform for pervasive fall detection system development. To our knowledge, no existing commercial products and academic work use mobile phones to integrate comprehensive fall detection and emergency communication.

– We design an algorithm for fall detection systems using mobile phones. It is an acceleration-based detection approach whose only requirement is that a mobile phone has an accelerometer.

– We design and implement a pervasive fall detection system, *PerFallD*, on the mobile phone-based platform to conduct fall detection. PerFallD has few false positives and false negatives; it is available in both indoor and outdoor environment; it is user-friendly, and it requires no extra hardware and service cost. It is also lightweight and power-efficient.

– We conduct experiments to evaluate detection accuracy. The experimental results show that our detection system achieves good performance in terms of low false negative and low false positive in fall detections. For the purpose of comparison, we implement algorithms provided in existing work and also test a typical commercial fall detection product. PerFallD outperforms existing algorithms, and achieves better balance between false negative and false positive when compared with the commercial product.

Paper Organization The rest of the paper is organized as follows. Section II presents related work. We present the system design in Section III and system implementation in Section IV. In Section V, we evaluate our system. Section VI concludes the paper.

II. Related Work

There are very few fall detection commercial products. Fall detector provided by Brickhouse [8] consists of a tele-assist base and a portable sensor. The base device needs to be installed indoor and the signal transmission distance between the sensor and the base is limited. ITT EasyLifeS [11] is one kind of cellphone that equipped with balance sensor. The manufacture claims that the phone will automatic dial SOS numbers if it is dropped. However, the device is too specific and the triggering condition is too trivial to provide pervasive and comprehensive fall detection. Betterbuys [12] provides Economical Fall Alarm Monitor that requires sensors deployed with chair pad, bed pad, floor mat, or chair seatbelt. It has more limitations to achieve pervasive detection.

Significance of fall detection also attracts academic research. Proposed fall detection techniques can be classified into three categories: acceleration based detection, databases based motion type classification, and image processing based detection.

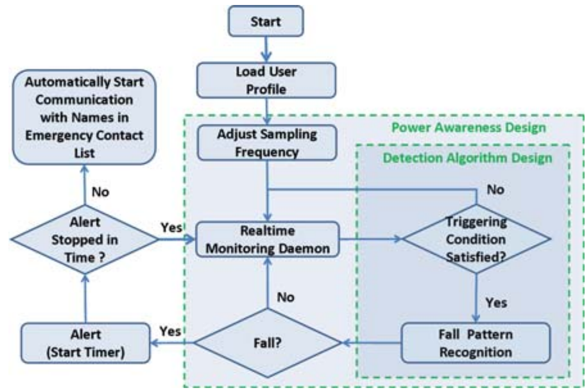

Figure 1. Working procedure of the PerFallD system. Power efficiency is explicitly considered in design of the modules illustrated within the bigger dashed box. The modules within the smaller dashed box present the algorithm design part.

When acceleration is used, the most widely used methods are based on thresholds. In [13], Nyan et al. let an accelerometer be settled into garment on the shoulder position. They use a threshold of absolute peak values of acceleration to determine fall. Kangas et al. in [14] propose four thresholds for total sum vector, dynamic sum vector, difference between the maximum and minimum acceleration values and vertical acceleration. Fall is considered detected as long as one threshold is exceeded. The above work show that the acceleration threshold-based detection works well in practice. However, the detection devices used in them are specified and not conveniently portable. The communication component that is also critical in a fall detection system is ignored in these work.

Ganti et al. in [15] and Karantonis et al. in [16] propose storing sensed user behavior data into a database for various activities, e.g., fall down, recognition thereafter. These databases built with the sensed data are very useful. They can be used to detect various normal or abnormal activities. However, it is not a trivial task to collect enough data for each individual to build up the database.

Fu et al. in [17], Sixsmith et al. in [18], Miaou et al. in [19] and Jansen et al. in [20] propose capturing images of people and then detect visual fall based on image processing techniques. Such approaches have limitations on pervasive detection, affordability and acceptability. The detection area is limited within monitoring environment, which is costly to build up. The people's privacy is compromised.

We also notice there are work that propose an integrated fall detection environment. In [21], a separated fall detector is connected to a mobile phone, which is used to query the user about his condition when the fall detector signals it. Paper [22] proposes using a network of fixed motes to provide location information of the victim after fall has

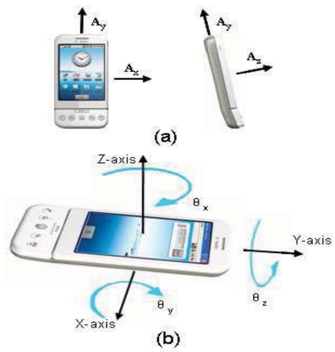

Figure 2. (a) Acceleration readings in directions of x-, y-, and z-axis that are associated with and fixed regard to the body of the mobile phone. (b) Mobile phone orientation can be decided by yaw (θ_x), pitch (θ_y) and roll (θ_z).

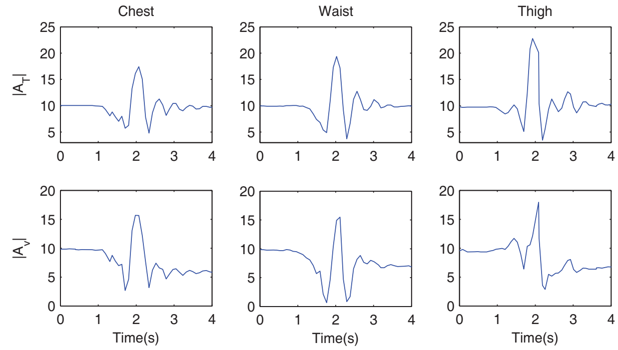

Figure 3. Examples of the amplitude of $\mathbf{A}_T$ and $\mathbf{A}_v$ that are calculated out from the readings of the integrated accelerometer in a mobile phone during a fall. We show the results when the phone is placed in different locations: in the pocket of a shirt (chest), on the belt (waist), and in a pocket of the pants (thigh).

been detected. In these pioneering work, reliability and availability of cooperation between different devices raise concerns. The system cost is also high.

III. PerFallD Design

In this section, we present the PerFallD design. We first present the system overview followed by the design of the detection algorithm. Note that the design is general—it is not constrained to a particular brand or type of mobile phone.

A. System Overview

PerFallD's workflow is illustrated in Fig. 1. Right after the program starts, a user profile will be loaded. A user dependent profile contains basic fall detection configuration such as the default sampling frequency, default detection algorithm, emergency contact list, etc. In different scenarios, users' activity patterns have varying degrees of rapidity, and it is more efficient to use different sampling frequencies in different scenarios. After the user profile is loaded, we provide users the chance to adjust the sampling frequency when interfaces that invoke sensor functions at different frequencies are provided. Then the main program, working as a background daemon, launches. If information collected in real time satisfies a certain preset condition, the pattern matching process begins to determine if a fall occurs. If no fall is detected, execution immediately returns to the daemon. If a fall is detected, the daemon service transmits a signal that triggers an alarm and starts a timer. If the user does not manually turn off the alarm within a certain time period, the system automatically calls contacts stored in the emergency contact list according to their priorities. The phone iteratively calls and texts up to five contacts.

As presented in Fig. 1, power efficiency is explicitly considered in the design of the modules illustrated within the larger dashed box. Four steps are taken to reduce power consumption: (1) the monitoring daemon runs in the background while other components of the program halt; (2) the sampling frequency can be adjusted; (3) the pattern matching process is launched only after daemon-collected data exceeds the preset threshold; and (4) hardware such as the screen is activated only when necessary.

The modules within the smaller dashed box present the algorithm design part. We will introduce the detection algorithm in the following section.

B. Algorithm Design

In this case, we present the detection algorithm designed for the mobile phones equipped with accelerometer.

Accelerometers usually provide the acceleration readings in directions of x-, y-, and z-axis. Accelerations in these directions are represented by $\mathbf{A}_x$, $\mathbf{A}_y$ and $\mathbf{A}_z$, respectively. For generality, we assume the directions of x-, y-, and z-axis decided by the posture of the phone. As illustrated in Fig. 2, the x-axis has positive direction toward the right side of the device, the y-axis has positive direction toward the top of the device and the z-axis has positive direction toward the front of the device. Vector $\mathbf{A}_T$ represents the total acceleration of the phone body. Its amplitude can be obtained by Eq. 1.

$$|\mathbf{A}_T| = \sqrt{|\mathbf{A}_x|^2 + |\mathbf{A}_y|^2 + |\mathbf{A}_z|^2}. \quad (1)$$

A mobile phone's orientation can be determined by yaw, pitch, roll values that are denoted as θ_x, θ_y and θ_z, respectively. We can further obtain the amplitude of $\mathbf{A}_v$, the acceleration at the absolute vertical direction, from Eq. 2.

$$|\mathbf{A}_v| = |\mathbf{A}_x \sin\theta_z + \mathbf{A}_y \sin\theta_y - \mathbf{A}_z \cos\theta_y \cos\theta_z|. \quad (2)$$

The fall detection algorithm is based on the values of $|\mathbf{A}_T|$ and $|\mathbf{A}_v|$. If the difference of $|\mathbf{A}_T|$ within a triggering time window win_{tt} exceeds triggering threshold Th_{tt}, the pattern recognition is triggered to check the difference between the maximum value and the minimum value of $|\mathbf{A}_T|$ within a checking time window win_{ct} following win_{tt}. If this difference is less than another threshold Th_{ct}, a fall is considered detected. A similar rule applies to $|\mathbf{A}_v|$, with corresponding time windows win_{tv}, win_{cv} and thresholds

Th_{tv}, Th_{cv}. If both the detection conditions about $|\mathbf{A}_T|$ and $|\mathbf{A}_v|$ are satisfied, a detection of fall is reported.

Fig. 3 presents the examples of $|\mathbf{A}_T|$ and $|\mathbf{A}_v|$ that are obtained from the integrated accelerometer in a mobile phone. We show the results when the phone is placed in different locations: in a pocket of a shirt (chest), on the belt (waist), and in a pocket of the pants (thigh). Thresholds can be set according to the training data obtained from extensive experiments. Based on a set of data, Fig. 4 shows the relationship between false negative and false positive for different values of Th_{tt} (one threshold is represented as one mark) when the Th_{ct} is fixed. We adjust thresholds in order to reduce false negative while simultaneously keep false positive in an acceptable range. More details will be provided in Section V.

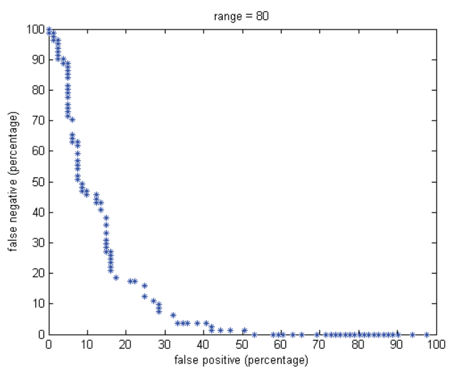

Figure 4. The relationship between false negative and false positive for different threshold Th_{tt} when the Th_{ct} is fixed to 80 (indicated as "range" in the figure). Data are from lateral falls. The phone is placed at the position of chest.

IV. Implementation

We develop the PerFallD prototype on Android G1 phone. It features an ARM-based, dual-core CPU capable of up to 4 million triangles/sec, a 98MB RAM and a 70MB of internal storage [23]. It uses a 1150mAh rechargeable lithium ion battery. It also provides an embedded accelerometer. In the following, we describe the implementation details of the PerFallD prototype.

We implement the prototype in Java, with Eclipse and Android 1.6 SDK. It consists of 7 class files, which includes 4 Activities, 1 View, 1 Service and 1 Resource. They can be divided into five major components: user interface, monitoring daemon, data processing, alert notification and system configuration. After the user starts the system, the monitoring daemon keeps running in background as a Service in Android, collecting and recording the readings of sensor. These readings are processed based on power-aware strategy and used to detect a fall. In data processing component, for simplicity, all the time windows are set to 4 seconds. When a fall is detected, the alert notification component

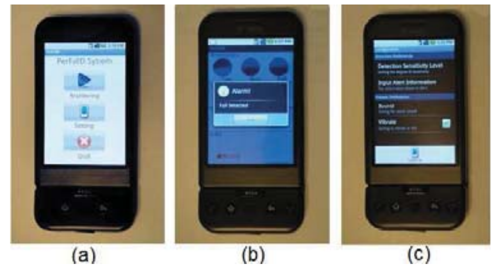

Figure 5. User interfaces in PerFallD: (a) bright, large virtual buttons on operating screen, (b) clear alert window (c) simple, non-confusing preference screen.

works to sound alarm to notify the attendant nearby and call the emergency contacts. Also, the user can change the configuration settings by invoking the preference screen.

We compile and build the system project, create and sign the .apk file in debug mode, then install it onto G1 phone by ADB tool. The size of the .apk application file is about 200KB. Ultimately, we may create the .apk file in release mode, sign it with our release private key and publish it on Android Market, making it available to users of Android-powered mobile devices for download.

We implement the user interfaces of PerFallD towards the elderly people, following the design ideas from Jitterbug[1]. The user interfaces of PerFallD have the following features. Large, lit key buttons make usage easy. Bright color screen displays everything with clarity. There is no confusing menus, making accessing all options clearly. Fig. 5 illustrates the user interfaces of PerFallD. Guided by friendly user interfaces, the operation of PerFallD is simple and straightforward. In the operating screen, three buttons are shown: *Monitoring*, *Setting* and *Quit*. The *Monitoring* button leads to the daemon. An alert window will prompt out once a fall is detected. The siren also sounds. The *Setting* button leads to the preference screen of program.

V. Evaluation

We evaluate the PerFallD prototype with experiments. In this section, we first introduce how the data are collected. Then we present PerFallD's performance and compare it with existing algorithms and a typical commercial product. We also present PerFallD's resource consumption.

A. Data Collection

We collect data of falls with different directions (forward, lateral and backward), different speeds (fast and slow) and in different environment (living room, bedroom, kitchen and outdoor garden). We also collect data of activities of daily living (ADL) including walking, jogging, standing and sitting. We separate all these collected data into two sets, one for training and the other for testing.

[1] Jitterbug [24] is a cell phone service provider that is known for providing appealing cell phone and phone service to the elderly people.

Table I
PERFORMANCE COMPARISON OF DETECTION TEST.

		FN(%)			FP(%)
		Forward Falls	Lateral Falls	Backward Falls	Other Activities
PerFallD	Chest	1.2	2.3	5.0	11.2
	Waist	2.6	3.3	2.1	8.7
	Thigh	1.0	10.0	2.2	11.0
Basic Algorithm		8.0	28.3	5.5	14.6
Fall Index		5.2	13.9	1.8	7.8
Commercial Product		0.8	1.2	29.9	21.9

* We call the ADL and movements of phone that may cause the false positive *other activities*.

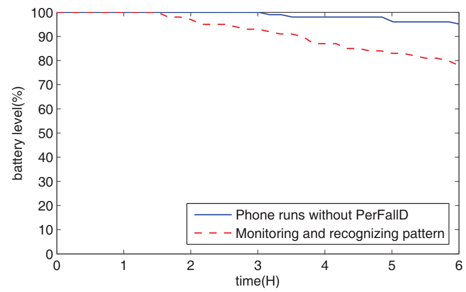

Figure 6. Power consumption: first curve presents the battery levels when the phone runs without PerFallD; the second one shows the battery levels when the monitoring daemon and the pattern recognition in PerFallD run.

We conduct the experiments with a group of real persons. Both falls and ADL are tested. Obviously, we cannot test falls with real elderly people. We recruit 15 participants who are graduate students from 20 to 30 years old, two of whom are female. Three of them are 161–170 cm tall, seven are 171–180 cm tall, and five are 181–190 cm tall. One person weighs less than 50 kg, two weigh 51–60 kg, five weigh 61–70 kg, and seven weigh 71–80 kg.

In test of fall detection, all the participants put the G1 phone in a shirt pocket, on the belt, or in one pants pocket, respectively. In each case of phone attached position, every participant falls 10 times in different directions and environment. In total, we obtain data for 450 falls that cover all falling directions and environment. We also collect ADL data for 20 minutes from each person.

B. 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 did not occur. In general, the lower the both FN and FP are, the better the performance is.

1) PerFallD Performance: We obtain the thresholds for our detection algorithm from the ROC curves generated from the training data. ROC curves show the tradeoff between FN and FP (any increase in FN will be accompanied by a decrease in FP). A typical ROC curve generated in our testing is shown in Fig. 4. With different threshold values, we can obtain different ROC curves. From each ROC curve, we select the threshold setting that achieves low FN and reasonable FP. Based on our training data, we set Th_{tt} to 120, Th_{tv} to 6, Th_{ct} to 50 and Th_{cv} to 2. These threshold settings achieve the best balance between FN and FP. Table I shows the experimental results. We find that PerFallD has different performances when the phone is placed at different positions and the *waist* is the best position to attach the phone, with the performance of average FN value being 2.67% and the FP value being 8.7%.

2) Performance Comparison: We compare the performance of PerFallD with two other existing detection algorithms and one commercial product. Table I shows the comparison of experiment results.

The basic algorithm uses the simple acceleration threshold to determine a fall. The threshold is only based on the value of $|\mathbf{A}_T|$. The detection only focuses on one big acceleration change (regarded as the impact of fall), ignoring the following acceleration changes. So it will miss some slow falls and alarm falsely in some ADL. Fall Index (FI) is proposed by Yoshida et al. in [25]. For any time i, FI can be calculated by Eq. 3.

$$FI_i = \sqrt{\sum_{k=x,y,z} \sum_{i-19}^{i} \left((A_k)_i - (A_k)_{i-1}\right)^2}. \qquad (3)$$

Since FI requires high sampling frequency and fast acceleration changes, it will miss falls that happen slowly. Its performance decreases in some specific situations.

The commercial product provided by Brickhouse [8] consists of one base and a wearable fall detector. The base needs to be connected with a phone line to communicate with emergency center. So it has to be fixed somewhere inside home. Due to the constraints of communicating range between base and fall detector, the users must be indoor to be under protection. The algorithm used in this commercial product is unknown. Experiments show that this system has high false negative (29.9%) in backward falls. Meanwhile, the false positive is also quite high (21.9%).

The results show that PerFallD outperforms existing algorithms, and achieves better balance between false negative and false positive compared with the commercial product.

C. Resource Consumption Performance

To test power consumption, we fully charge the G1 phone and then monitor the power states continuously for 6 hours in different scenarios: 1) the G1 phone runs without PerFallD; 2) the monitoring daemon of PerFallD keeps running, sensing and recording acceleration values, then calculates and recognizes fall pattern on the demand of monitoring results.

Fig. 6 presents the two curves of battery level states versus time during the time period of 6 hours. If PerFallD keeps running normally until the battery power is exhausted, it will last more than 33 hours.

Furthermore, we monitor the CPU and memory usage of G1 phone during the running of PerFallD system. The average CPU usage is 7.41%; the memory usage is about 600KB, 0.6% of total RAM capacity of G1 phone.

VI. Conclusion

In this paper, we propose utilizing mobile phones as a platform for pervasive fall detection system development, for the first time. We design the detection algorithm based on mobile phone platforms. We implement a prototype system named PerFallD on the Android G1 phone and conduct experiments to evaluate our system. Experimental results show that PerFallD achieves good detection performance and power efficiency.

PerFallD can be enhanced by integration with some extra protection devices, e.g., airbag based fall protector proposed by Charpentier [26] , to reduce fall impacts and prevent fall related injuries.

Acknowledgment

This work was supported in part by the US National Science Foundation (NSF) under grants No. CNS-0916584, CAREER Award CCF-0546668, and the Army Research Office (ARO) under grant No. AMSRD-ACC-R 50521-CI. Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] J. Stevens, *Falls among Older Adults: Moving from Research to Practice,* in Proceedings of International Conference on Aging, Disability and Independence, 2003.

[2] J. S. Fulks, F. Fallon, W. King, G. Shields, N. Beaumont and J. Ward-Lonergan, *Accidents and Falls in Later Life,* Generations Review, Vol. 12, No. 3, pp. 2-3, 2002

[3] M. C. Hornbrook, V. J. Stevens, D. J. Wingfield, J. F. Hollis, M. R. Greenlick and M. G. Ory, *Preventing Falls Among Community-dwelling Older Persons: Results from a Randomized Trial,* The Gerontologist, Vol. 34, No. 1, pp. 16-23, 1994

[4] R. Tideiksaar, *Falling in Old Age: Prevention and Management, 2nd ed,* Springer-Verlag, 1998

[5] S. Sadigh, A. Reimers, R. Andersson and L. Laflamme, *Falls and Fallrelated Injuries Among the Elderly: a Survey of Residential-care Facilities in a Swedish Municipality,* Journal of Community Health, Vol. 29, No. 2, pp. 129-140, 2004.

[6] O. P. Ryynanen, S. L. Kivela, R. Honkanen and P. Laippala, *Falls and Lying Helpless in the Elderly,* Z Gerontol, Vol. 25, No. 4, pp. 278-282, 1992.

[7] U.S. Census Bureau Population Data Public Web Site, http://www.census.gov/population/www/

[8] http://www.brickhousealert.com/personal-emergency-medical-alarm.html.

[9] G. Mies, *Six New Phones from AT&T: Have a Look,* PC World, Mar. 30, 2009, http://www.pcworld.com/article/162233/six_new_phones_from_atandt_have_a_look.html

[10] *New Smartphone Family Wireless Plan from Cincinnati Bell,* http://wirelessplansinformation.blogspot.com/2008/10/new-smartphone-family-wireless-plan.html

[11] http://www.ittmonaco.com/en/easy/lifeS.htm

[12] http://www.betterbuys-r-us.com/HomeCare Sale.htm.

[13] M. N. Nyan, F. E. Tay, M. Manimaran and K. H. Seah, *Garment-based Detection of Falls and Activities of Daily Living using 3-axis MEMS Accelerometer,* Journal of Physics: Conference Series, Vol. 34, pp. 1059-1067, 2006.

[14] M. Kangas, A. Konttila, P. Lindgren, I. Winblad and T. Jamsa, *Comparison of Low-complexity Fall Detection Algorithms for Body Attached Accelerometers,* Gait Posture. Vol. 28, No. 2, pp. 285-291, 2008.

[15] R. K. Ganti, P. Jayachandran, T. F. Abdelzaher and J. A. Stankovic, *SATIRE: a Software Architecture for Smart AtTIRE,* in Proceedings of the 4th international conference on Mobile systems, applications and services, MobiSys 2006.

[16] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell and B. G. Celler, *Implementation of a Real-time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring,* IEEE Transactions on Information Technology in Biomedicine, Vol. 10, No. 1, pp. 156-167, 2006.

[17] Z. Fu, E. Culurciello, P. Lichtsteiner and T. Delbruck, *Fall Detection using an Address-Event Temporal Contrast Vision Sensor,* in Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS 2008.

[18] A. Sixsmith and N. Johnson, *A Smart Sensor to Detect the Falls of the Elderly,* Pervasive Computing, Vol. 3, Issue 2, pp. 42-47, 2004.

[19] S. Miaou, P. Sung and C. Huang, *A Customized Human Fall Detection System Using Omni-Camera Images and Personal Information,* in Proceedings of 1st Distributed Diagnosis and Home Healthcare Conference, 2006.

[20] B. Jansen and R. Deklerck, *Context Aware Inactivity Recognition for Visual Fall Detection,* in Proceedings of Pervasive Health Conference and Workshops, 2006.

[21] T. R. Hansen, J. M. Eklund, J. Sprinkle, R. Bajcsy and S. Sastry, *Using Smart Sensors and A Camera Phone to Detect and Verify the Fall of Elderly Persons,* in Proceedings of European Medicine, Biology and Engineering Conference, EMBEC 2005.

[22] J. Chen, K. Kwong, D. Chang, J. Luk and R. Bajcsy, *Wearable Sensors for Reliable Fall Detection,* in Proceedings of 27th Annual International Conference of the Engineering in Medicine and Biology Society, IEEE-EMBS 2005.

[23] http://en.wikipedia.org/wiki/HTC_Dream.

[24] http://www.jitterbug.com.

[25] T. Yoshida, F. Mizuno, Hayasaka T, K. Tsubota, S. Wada and T. Yamaguchi, *A Wearable Computer System for a Detection and Prevention of Elderly Users From Falling, in* in Proceedings of The 12th International Conference on Biomedical Engineering, 2005.

[26] P. J. Charpentier, *A Hip Protector Based On Airbag Technology,* Bone Vol. 18, No. 1, p. S117.