Pers Ubiquit Comput
DOI 10.1007/s00779-010-0292-x

ORIGINAL ARTICLE

Mobile phone-based pervasive fall detection

Jiangpeng Dai · Xiaole Bai · Zhimin Yang · Zhaohui Shen · Dong Xuan

Received: 21 January 2010 / Accepted: 2 March 2010

Abstract Falls are a major health risk that diminishes the quality of life among the elderly people. The importance of fall detection increases as the elderly population surges, especially with aging "baby boomers". However, existing commercial products and academic solutions all fall short of pervasive fall detection. In this paper, we propose utilizing *mobile phones* as a platform for developing pervasive fall detection system. To our knowledge, we are the first to do so. We propose *PerFallD*, a pervasive fall detection system tailored for mobile phones. We design two different detection algorithms based on the mobile phone platforms for scenarios with and without simple accessories. We implement a prototype system on the Android G1 phone and conduct extensive experiments to evaluate our system. In particular, we compare PerFallD's performance with that of existing work and a commercial product. The experimental results show that PerFallD achieves superior detection performance and power efficiency.

Keywords Pervasive fall detection · Mobile phones · Context information · Accelerometer · Magnetic field sensor · Accessory

J. Dai (✉)
Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing 210096, China
e-mail: jpdai@seu.edu.cn; jpdai@cse.ohio-state.edu

J. Dai · X. Bai · Z. Yang · D. Xuan
Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210, USA
e-mail: baixia@cse.ohio-state.edu

Z. Yang
e-mail: yangz@cse.ohio-state.edu

D. Xuan
e-mail: xuan@cse.ohio-state.edu

Z. Shen
Division of Physical Therapy, The Ohio State University, Columbus, OH 43210, USA
e-mail: shen.119@osu.edu

1 Introduction

1.1 Motivation

Falls are a major health hazard for elderly people [1] and also a major obstacle to their independent living [2, 3]. The estimated fall incidence every year for both institutionalized and independent living people over 75 is at least 30% [4, 5]. 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 them might fall more than once [6]. Falls not only cause physical injuries such as disabling fractures [7], but also have dramatic psychological consequences that reduce elderly people's independence [8, 9]. 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" [10].

The considerable risk of falls and the substantial increase of the elderly population stimulate both commercial product development 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 contacts after a fall is detected. Figure 1 depicts a typical commercial fall detection system, which is provided by Brickhouse [11]. The system consists of a portable sensor (the detection component on the left side of the figure) and a tele-assist base (the communication component on the right side of the figure).

The major problem with existing commercial products is that they cannot provide *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 around the waist. Once the base receives the signal from the sensor indicating a fall, it can automatically communicate with a preset emergency contact using a 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 need to bring the sensor everywhere. They may easily forget to do so, 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 [11].

Besides the commercial products, existing academic research also has deficiencies that hinder pervasive fall detection. More detailed information on the state of the art is provided in Sect. 2.

1.2 Our contributions

In this paper, we propose utilizing *mobile phones* as the platform for developing pervasive fall detection system, 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 developing pervasive fall detection system. Mobile phone-based fall detection system 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 argue 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 requirement for such a mobile phone platform is 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 popular and thoroughly accepted in society. As shown in Fig. 2, over 120 million smartphones were sold in 2008 [12], and their popularity is projected to continuously increase in the near future due to decreasing price. Recently, several leading telecommunication companies, e.g., AT&T, have made available affordable smartphones [13, 14], whose features are similar to those of high-end models, in addition to cheaper service plans [15].

We summarize our contributions in this paper as follows.

- We propose utilizing *mobile phones* as the platform for developing pervasive fall detection system. To our best knowledge, no existing commercial products and academic work use mobile phones to integrate comprehensive fall detection and emergency communication.
- We design two algorithms for fall detection systems using mobile phones. The first is an acceleration-based detection approach. The second algorithm is designed for the case when a certain accessory can be used to further capture human behavior information. This algorithm is based on shape context and Hausdorff distance.

Fig. 1 A typical commercial fall detection system provided by Brickhouse [11] consists of a portable sensor (the detection component shown in *left*) and a tele-assist base (the communication component shown in *right*)

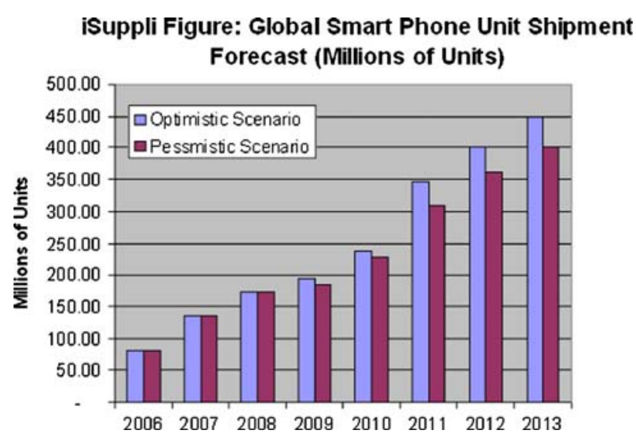

Fig. 2 From shipment data, smartphones are widely accepted and increasing in popularity in the near future

- We design and implement a pervasive fall detection system, *PerFallD*, on the mobile phone-based platform to conduct fall detection with or without other small accessories. PerFallD has few false negatives and false positives. It is available in both indoor and outdoor environment. Besides being user-friendly, it requires no extra hardware and service cost. It is also lightweight and power-efficient.
- We conduct extensive experiments, with both a mannequin and real persons, to evaluate detection accuracy. The experimental results show our detection system achieves superior performance in terms of low false negative and low false positive in fall detections with or without accessory. 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 2 presents the related work. We introduce the system design in Sect. 3 and system implementation in Sect. 4. In Sect. 5, we evaluate our system with extensive experiments. Section 6 concludes the paper.

2 Related work

There are very few fall detection commercial products. Fall detector provided by Brickhouse [11] consists of a tele-assist base and a portable sensor. The base device needs to be installed indoors and the signal transmission distance between the sensor and the base is limited. ITT EasyLifeS [16] 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. Philips Lifeline [17] provides medical alert service requiring the elderly people to push a certain button to report a fall. It cannot handle certain critical cases, e.g., the elderly people fall down with a sudden faint. Betterbuys [18] 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.

When acceleration is used, the most widely used methods are based on thresholds. In [19], Nyan et al. make an accelerometer be settled into garment on the shoulder position. They use a threshold of absolute peak values of acceleration to determine falls. Kangas et al. in [20] propose four thresholds for total sum vector, dynamic sum vector, vertical acceleration and difference between the maximum and minimum acceleration values. A fall is considered detected as long as one threshold is exceeded. In [21] and [22], information of body orientation and posture are used to complement acceleration-based detection. All aforementioned 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. *Mover* [23] is proposed by Fraunhofer recently as a kind of activity monitor and fall detector for Android system. It uses the accelerometers on phones to collect the accelerations and sum them to measure human activity level. It may also detect user falls; however, the algorithm is still being tested as the developer announced. Now, *Mover* is more like a simple interesting APP on Android.

Ganti et al. in [24] and Karantonis et al. in [25] 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 [26], Sixsmith et al. in [27], Miaou et al. in [28] and Jansen et al. in [29] propose capturing images of people and then detecting visual falls based on image-processing techniques. Such approaches have limitations on pervasive detection, affordability and acceptability. The detection area is limited within the monitoring environment, which is costly to build up. The people's privacy is also compromised.

We also notice there are work that propose an integrated fall detection environment. In [30], 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. In [31], the mobile phone acts as a relay between the sensors and a remote server that runs the fall detection algorithm. Paper [32] proposes using a network of fixed motes to provide location information of the victim after a fall has been detected. Lieberman-Aiden invents *ishoe* [33], an electronic shoe insole, to help users manage their balance before a fall occurs. It uses sensors to track users' balance patterns and transmits data to computers for interpretation. In these pioneering work, reliability and availability of cooperation between different devices raise concerns. The system cost is also high.

3 PerFallD design

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

3.1 System overview

PerFallD's workflow is illustrated in Fig. 3. 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 as 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. 3, 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 exceed 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 algorithms in the following sections.

3.2 Algorithm design without accessory

In this section, we present an acceleration-based detection algorithm designed for mobile phones.

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 are determined by the posture of the phone. As illustrated in Fig. 4, 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.

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

A mobile phone's orientation sensor can determine its yaw, pitch and 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 in the absolute vertical direction, from Eq. 2.

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

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

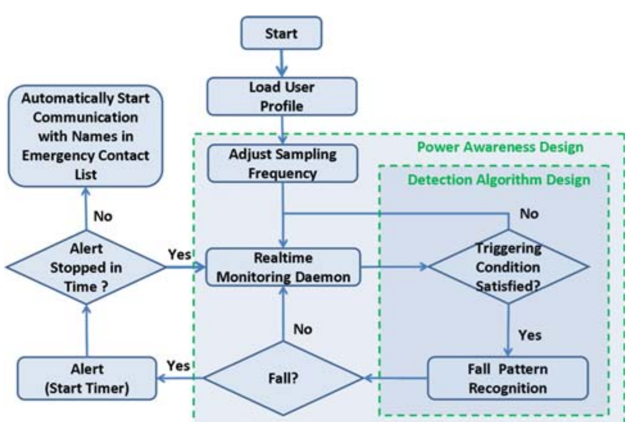

Fig. 3 Working procedure of the PerFallD system. Power efficiency is explicitly considered in design of the modules illustrated within the *bigger dashed box*. The modules within the *smaller dashed box* present the algorithm design part

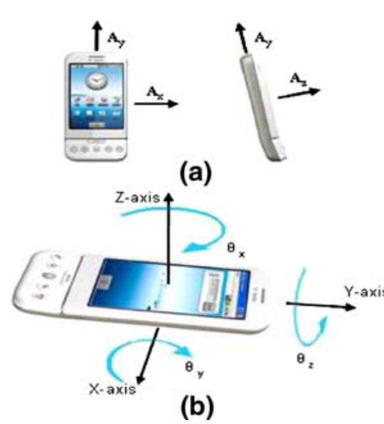

Fig. 4 **a** Acceleration readings in directions of x-, y-, and z-axis that are associated with and fixed regard to the body of the mobile phone. **b** Mobile phone orientation can be determined by yaw (θ_x), pitch (θ_y) and roll (θ_z)

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.

Figure 5 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. 6 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 Sect. 5.

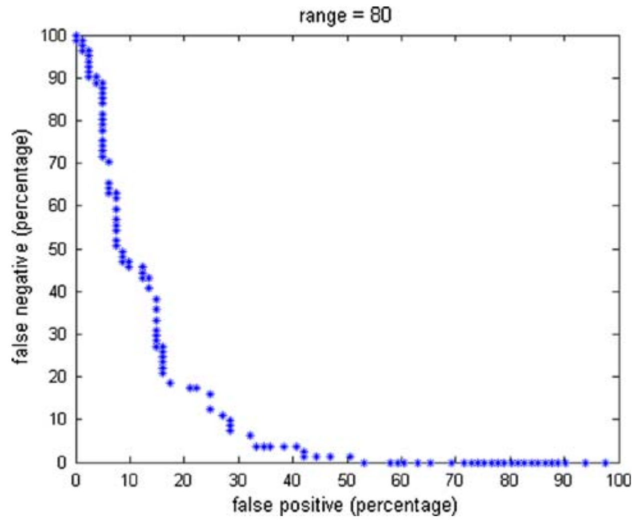

Fig. 6 The relationship between false negative and false positive for different threshold Th_{tt} when the Th_{ct} is fixed to 80 (indicated as "range" in the Figure). Data are from lateral falls. The phone is placed at the position of chest

3.3 Algorithm design with a magnetic accessory

The performance of detection using accelerations alone can be improved by introducing some simple accessories. In biomechanics, some researchers focus on the mechanics of human falling [34]. Some of them have made attempts to define some common step mechanics during falling from the viewpoint of biomechanics [35]. The mobile phones together with simple accessories can effectively capture these common steps to help detect a fall.

In this section, we introduce a mobile phone-based detection method with some simple accessories. In particular, we focus on a mobile phone with a magnetic field sensor and a magnetic accessory. The reading value from the sensor represents the strength of magnetic field (MF) around the phone. We then can use this value to infer the relative position between the phone and the magnetic accessory.

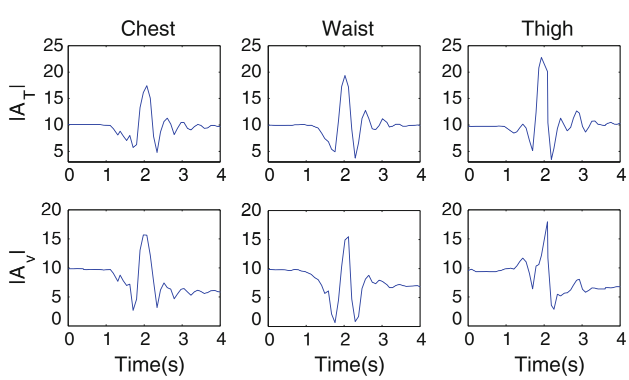

Fig. 5 Examples of the amplitude of $\mathbf{A}_T$ and $\mathbf{A}_v$ that are calculated out from the readings of the integrated accelerometer in a mobile phone during a fall. We show the results when the phone is placed in different locations: in the pocket of a shirt (chest), on the belt (waist), and in a pocket of the pants (thigh)

Research in biomechanics has found that the movements of legs have salient features in falls [34]. To catch such features, we put a magnetic accessory slightly above the knee of left leg and the mobile phone in the right pocket of the pants.

In Fig. 7, the two figures in the left column show typical readings of magnetic sensor for falls. The top-right figure shows the readings at the start of walking. The bottom-right figure shows the readings in arbitrary postures that are not falls. We find that the reading curves for falls have some unique patterns. As illustrated in Fig. 7, before a fall happens, the MF values are on a relatively stable high level. At the beginning of a fall, the MF value decreases slowly, but suddenly goes higher, making a salient convex shape, and then decreases fast. After it reaches the minimum, it turns back to a higher value that is much lower than the values before falling, and then keeps relatively stable afterwards. This interesting observation can be explained from the view of biomechanics. Before and after a fall, the human postures are significantly different. The distance between two legs when people are standing is often smaller than that when people are lying on the floor after a fall. Furthermore, during a fall, the legs bend and cross slightly, which causes knee movement. This slightly increases the MF value before it decreases further.

Our algorithm aims to capture the aforementioned unique characteristics effectively and to distinguish the readings that indicate falls from others. We present some preliminaries before introducing the detection algorithm.

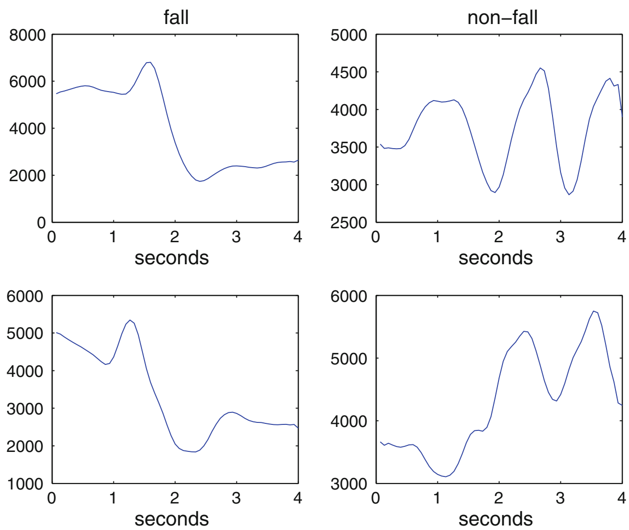

Fig. 7 The two Figures in the *left* column show the typical readings of magnetic sensor for falls. The *top-right* Figure shows the readings at the start of walking. The *bottom-right* Figure shows the readings in arbitrary postures that are not falls

Successive decrements Assume the readings show a monotonous decrease at $[t_1, t_2]$ followed by another monotonous increase at $[t_2, t_3]$ and then a monotonous decrease at $[t_3, t_4]$. We call these two decrements occurred at time periods $[t_1, t_2]$ and $[t_3, t_4]$ successive.

Shape context The concept of shape context is proposed by Belongie et al. at [36]. For a point p_i on a shape, considering the set of vectors originating from this point p_i to all other sample points on the shape, we compute a coarse histogram h_i of the relative coordinates of the remaining $n - 1$ points by Eq. 3.

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in bin(k)\}. \quad (3)$$

This histogram is defined to be the shape context of p_i. We use bins that are uniform in log-polar space, making the descriptor more sensitive to the positions of nearby sample points than to those of points farther away. Interested readers may refer to [36] for more details.

Hausdorff distance Hausdorff distance is a classical correspondence-based measure on how far two subsets of a metric space are from each other. As pointed out in some recent work, such as [37], Hausdorff distance can be used for shape comparison. If X and Y are two non-empty subsets of a metric space (M, d), we define their Hausdorff distance $d_H(X, Y)$ by Eq. 4.

$$\max\{\sup_{x\in X}\inf_{y\in Y} d(x,y), \sup_{y\in Y}\inf_{x\in X} d(x,y)\}. \quad (4)$$

Our algorithm has two steps. In the first step, the sum of every two successive decrements in sensor reading curve will be checked to see whether it exceeds a threshold Th_{tm}. If the check at the first step returns a true, the operations in the second step will be triggered at a certain short time after the end point of the second decrement. In the second step, the shape contexts of two end points in successive decrements of the current sensor reading curve will be calculated and combined. Then we compute Hausdorff distance between this new generated shape context and the known shape context from the sensor readings during fall, for comparison. If the Hausdorff distance is smaller than a threshold Th_{hd}, the current reading curve is labeled as fall, otherwise as non-fall. In the following, we use an example to further illustrate the algorithm.

We zoom in the top-left figure in Fig. 7 and then get Fig. 8. We first check whether there exist two successive decrements and their total decrement exceeds the preset threshold. In Fig. 8, AB and CD are two successive decrements and it is found that their total is larger than some threshold. Then the second step of the detection algorithm launches at 1.6 s after point D occurs. We take B as the center of one log-polar coordinate system. In such a coordinate system, the largest distance is divided into k_d distance ranges uniformly according to the log values denoted as $D_1^r, D_2^r, \ldots, D_{k_d}^r$, and center angle 2π is divided into k_a angle ranges uniformly denoted as $A_1^r, A_2^r, \ldots, A_{k_a}^r$. Figure 8 illustrates an example of such log-polar coordinate system at its bottom-left corner. Each point p excluding B in the time window can be represented by distance ρ and angle θ as shown in Fig. 8, falling into a certain distance range and angle range. Then we can obtain a $k_d \times k_a$ matrix M_B wherein the value at position (i, j) denotes the number of points that fall into

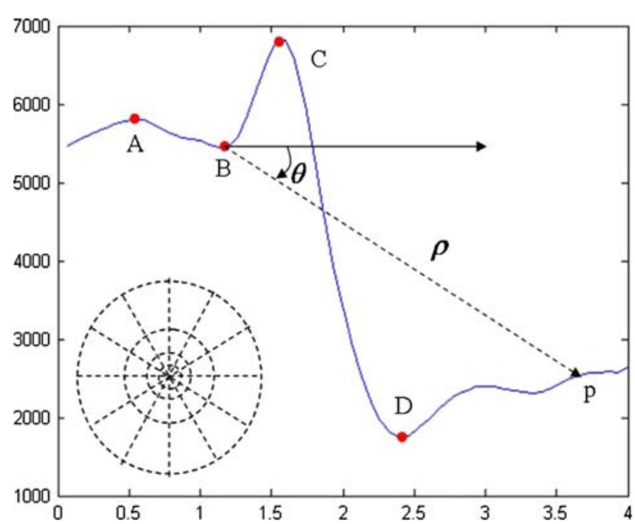

Fig. 8 *Y* axis represents the magnetic sensor readings and *X* axis represents seconds. AB and CD are two successive decrements with B and D the end points. An example of a log-polar coordinate system is shown at the corner

the distance range D_i^r and angle range A_j^r. We take D as the center of another log-polar coordinate system that is the same as mentioned earlier. We can obtain another matrix M_D. These two matrices represent the shape contexts of two end points in the successive decrements. We combine M_B and M_D together into a $2k_d \times k_a$ matrix M. This M is illustrated as an image with gray color level shown as the top-left image in Fig. 9. A darker color indicates a higher element value in the matrix. Figure 9 shows images of different combined shape context matrices obtained from corresponding figures in Fig. 7. The matrices of falls are similar, while the matrices of non-fall have great differences. That similarity can be captured by Hausdorff distance. If Hausdorff distance between one M and the Ms known from falls is small enough, the M is considered to indicate a fall.

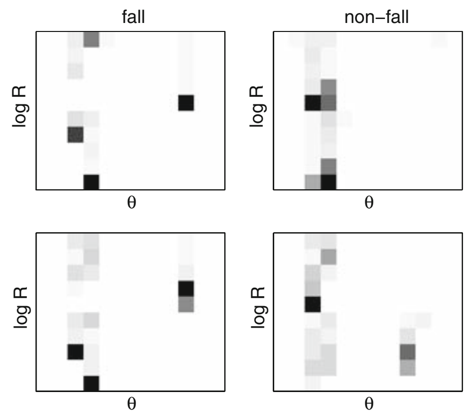

Fig. 9 Image representation of combined shape context matrices (2 × 5 distance ranges and 12 angle ranges) corresponding to Figures in Fig. 7. A *darker color* indicates a higher value in the matrix

4 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 [38]. It uses a 1150 mAh rechargeable lithium ion battery. Besides camera, G1 phone provides three sensors: an accelerometer, an orientation sensor and a digital magnetic field sensor. 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 user starts the system, the monitoring daemon keeps running in background as a Service in Android, collecting and recording the readings of sensors. 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 s. When a fall is detected, the alert notification component works to sound alarm to notify the attendant nearby and call the emergency contacts. Also, 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 216 KB. 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 toward 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. Figure 10 illustrates the user interfaces of PerFallD. Guided by friendly user interfaces, the operation 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.

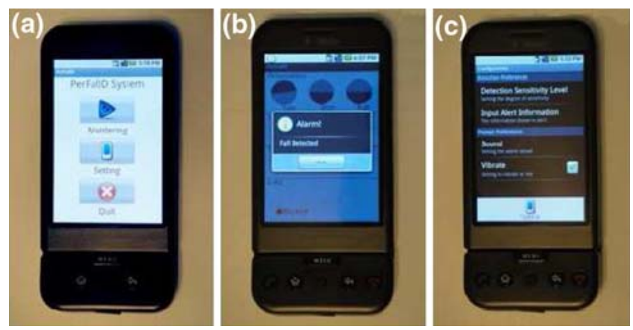

Fig. 10 User interfaces in PerFallD: (**a**) bright, large virtual buttons on operating screen (**b**) clear alert window (**c**) simple, non-confusing preference screen

In fall detection method with an accessory, we utilize several magnets and a bandage that are originally for healthcare use. These magnets' surface Gauss rating is 350 Gauss (35 *mT*), measured by a hand-held digital gaussmeter–Model 410 [41]. According to the WHO standard,

[1] Jitterbug [39] is a cell phone service provider that is known for providing appealing cell phone and phone service to the elderly people [40].

they are harmless to general public who do not use implanted metallic devices [42].

5 Evaluation

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

5.1 Data collection

We collect extensive falls data. We study 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.

We first collect falls data using a mannequin equipped with PerFallD system. The mannequin is forced to fall in different directions and in different environment. The PerFallD system is attached to the mannequin at one of the three positions for fall detection without accessories: chest, waist, and thigh. Figure 11a illustrates the situation in which the phone is on the mannequin's waist. Since the relative position of two legs on the mannequin does not change during a fall, we did not conduct the experiments for fall detection with magnetic accessories. We cannot collect the ADL data from the static mannequin, either. Using the mannequin, we collect data of 600 falls in total. They cover all the falling direction, environment and the position of phone attachment (as mentioned before).

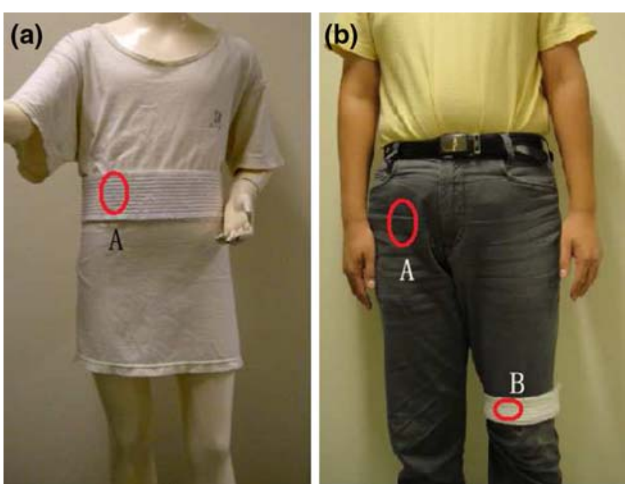

Fig. 11 **a** The G1 phone is attached to the waist of the mannequin (position denoted by *circle A*). **b** The participant puts the G1 phone in his pocket of the pants (position denoted by *circle A*) and binds the magnetic accessory on the leg (position denoted by *circle B*)

We further 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 women. 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.

To test detection without accessories, all the participants put the G1 phone in a shirt pocket, on the belt, or in one pants pocket, respectively. To test detection with accessories, they put the phone in one pocket of the pants while attaching the magnet accessory to the other leg. Figure 11b illustrates this. In each case of phone attached position, every participant falls 10 times in different directions and environment. In total, we obtain data for 600 falls that cover all falling directions and environment. We also collect ADL data for 30 min from each person.

5.2 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.

5.2.1 PerFallD performance

We obtain the thresholds for our detection algorithms from the ROC curves which are 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. 6. 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. The threshold settings that are reported in the following achieve the best balance between FN and FP based on our training data. In detection test without accessories using the mannequin, we set the Th_{tt} to 150, Th_{tv} to 4, Th_{ct} to 70 and Th_{cv} to 2. While in the tests with real persons, we set Th_{tt} to 120, Th_{tv} to 6, Th_{ct} to 50, Th_{cv} to 2; and we set Th_{tm} to 1970 and Th_{hd} to 1 in detection with a magnetic accessory. Tables 1 and 2 show the experimental results of PerFallD with the mannequin and real persons, respectively.

In detection without accessories, PerFallD has different performances when the phone is placed at different

Table 1 Performance comparison of detection test with the mannequin

	FN(%)		
	Forward falls	Lateral falls	Backward falls
PerFallD without accessory			
Chest	1.1	2.5	5.5
Waist	2.5	3.4	2.4
Thigh	1.3	9.5	2.6
Basic algorithm	8.2	27.9	5.8
Fall index	4.9	14.1	1.9
Commercial product	0	0	28.3

No false positive is caused, since the ADL data are not available and the phone is bound firmly in the test with the mannequin

positions. We discover that the *waist* is the best position to attach the phone. In this case, the average of FN values in the experiments with the mannequin is 2.77%; in the experiments with real persons, the average FN value is 2.67%, while the FP value is 8.7%.

In detection with a magnetic accessory, we find that it can achieve better accuracy. In the experiments with real persons, the average of FN values is 2.13% and the FP value is 7.7%. Meanwhile, the experimental results show that it performs well in detecting lateral and backward falls because the patterns of leg movements in such falls are apparent. Our experiments also show that fall detection method with a magnetic accessory works well especially to those slow falls, which are common to elderly people.

5.2.2 Performance comparison

We compare the performance of PerFallD with two other existing detection algorithms and one commercial product. Table 1 shows the results of experiments conducted with the mannequin, while Table 2 illustrates the experiments on real persons.

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 a 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 [43]. For any time i, FI can be calculated by Eq. 5.

$$FI_i = \sqrt{\sum_{k=x,y,z} \sum_{i-19}^{i} \left((A_k)_i - (A_k)_{i-1}\right)^2}. \tag{5}$$

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

Figure 12 shows the commercial product of Brickhouse together with a G1 phone. This commercial product consists of one base and a wearable fall detector [11]. 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 with the mannequin and real persons both show that this system has high false negative (28.3 and 29.9%) in backward falls. Meanwhile, the false positive of detection in experiment with real persons is also quite high (21.9%).

The results show that no matter testing with the mannequin or real persons, PerFallD outperforms existing algorithms, and achieves better balance between false negative and false positive compared with the commercial product.

5.2.3 Environment impact

We test PerFallD in different locations (in living room, bedroom, kitchen and outdoor garden). In all these

Table 2 Performance comparison of detection test with real persons

	FN(%)			FP(%)
	Forward falls	Lateral falls	Backward falls	Other activities
PerFallD without accessory				
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
PerFallD with accessory	3.1	2.2	1.1	7.7
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*

Fig. 12 The commercial product of fall detection (white) compared with Android G1 Phone

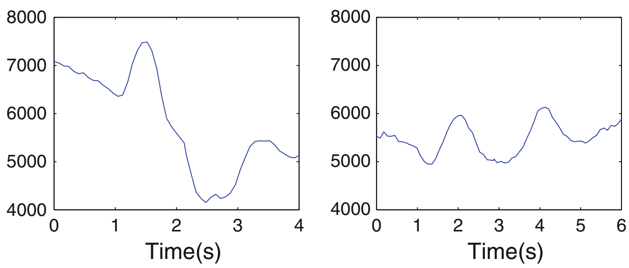

Fig. 13 (*left*) The magnetic sensor readings during a fall in kitchen are similar to those in other places. (*right*) The sensor readings when passing by an oven in kitchen are clearly different from those in a fall

locations, PerFallD works well. Especially, the detection with the magnetic accessory is not interrupted badly, even in the kitchen with many ferric stuffs. Figure 13 shows some typical readings of magnetic sensor in kitchen.

5.3 Resource consumption performance

To test power consumption, we fully charge the G1 phone and then monitor the power states continuously for 6 h in different scenarios: (1) the G1 phone runs without PerFallD; (2) the monitoring daemon in PerFallD keeps running, sensing and recording sensor reading values, without pattern recognition; (3) PerFallD not only monitors, but also calculates and recognizes fall pattern on the demand of monitoring results. Figure 14 presents the three curves of battery level states versus time during the time period of 6 h. If PerFallD keeps running normally until the battery power is exhausted, it will last about 33.5 h.

Furthermore, we monitor the CPU and memory usage of G1 phone during the running of monitoring daemon and after the fall pattern recognition is invoked. When only the monitoring daemon is running, the average CPU usage is 6.68%; the memory usage is about 350KB, 0.35% of the

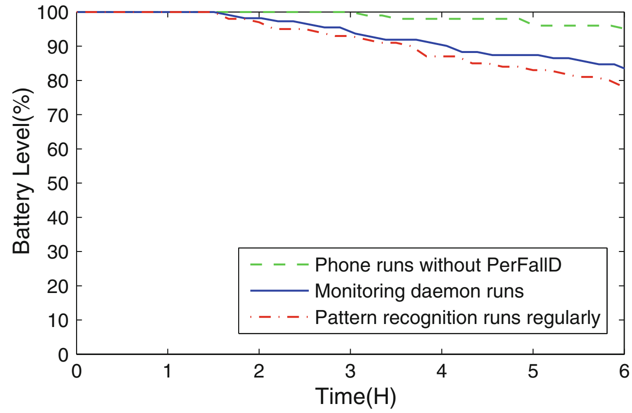

Fig. 14 Power consumption: first curve presents the battery levels when the phone runs without PerFallD; the second one shows the battery levels when the monitoring daemon in PerFallD runs; the third one presents the battery levels when the pattern recognition in PerFallD runs regularly

total RAM capacity of G1 phone. During the process of fall pattern recognition, the CPU usage is 8.15%; the usage of memory increases by about 300KB.

6 Conclusion

In this paper, we propose utilizing mobile phones as a platform for developing pervasive fall detection system, for the first time. We design different detection algorithms based on mobile phone platforms for scenarios with and without simple accessories. We implement a prototype system named PerFallD on the Android G1 phone and conduct extensive experiments to evaluate our system. Experimental results show that PerFallD achieves good detection performance and power efficiency.

In future, we may further enhance the current system by integrating extra protection devices, e.g., airbag-based fall protector proposed by Charpentier [44] and Fukaya [45], to reduce fall impacts and prevent fall-related injuries.

Acknowledgments 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-R50521-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. Stevens J (2003) Falls among older adults: moving from research to practice. In: Proceedings of international conference on aging, disability and independence, 2003
2. Fulks JS, Fallon F, King W, Shields G, Beaumont N, Ward-Lonergan J (2002) Accidents and falls in later life. Generations Rev 12(3):2–3

3. Pynoos J, Sabata D (2003) Home environmental modification: preventing falls at home. In: Proceedings of international conference on aging, disability and independence, 2003
4. Duthie E (1989) Falls. Med Clin North Am 73:1321–1335
5. Hornbrook MC, Stevens VJ, Wingfield DJ, Hollis JF, Greenlick MR, Ory MG (1994) Preventing falls among community-dwelling older persons: results from a randomized trial. Gerontologist 34(1):16–23
6. Tideiksaar R (1998) Falling in old age: prevention and management, 2nd edn. Springer, Berlin
7. Sadigh S, Reimers A, Andersson R, Laflamme L (2004) Falls and fallrelated injuries among the elderly: a survey of residential-care facilities in a Swedish municipality. J Community Health 29(2):129–140
8. Ryynanen OP, Kivela SL, Honkanen R, Laippala P (1992) Falls and lying helpless in the elderly. Z Gerontol 25(4):278–282
9. Kiel DP (1991) Falls. R I Med J 74(2):75–79
10. U.S. Census Bureau (2009) Population data public web site. http://www.census.gov/population/www. Accessed 29 Sept 2009
11. http://www.brickhousealert.com/personal-emergency-medical-alarm.htm. Accessed 2 October 2009
12. Resende P (2008) Smartphone competition heats up with many choices. NewsFactor Network Sept. 2008. http://www.newsfactor.com/story.xhtml?story_id=6167. Accessed 2 October 2009
13. Boran M (2009) How the mobile social web will make 2009 a smart phone odyssey. http://www.independent.ie/business/technology/how-the-mobile-social-web-will-make-2009-a-smart-phone-odyssey-1609944.html. Accessed 2 October 2009
14. Mies G (2009) Six new phones from AT&T: have a look. PC World, 30 Mar 2009. http://www.pcworld.com/article/162233/six_new_phones_from_atandt_have_a_look.htm. Accessed 2 October 2009
15. New smartphone family wireless plan from Cincinnati Bell. http://www.wirelessplansinformation.blogspot.com/2008/10/new-smartphone-family-wireless-plan.html. Accessed 26 Sept 2009
16. http://www.ittmonaco.com/en/easy/lifeS.ht. Accessed 20 Dec 2009
17. http://www.lifelinesys.com/index.jsp?campaign=10. Accessed 20 Dec 2009
18. http://www.betterbuys-r-us.com/HomeCareSale.ht. Accessed 20 Dec 2009
19. Nyan MN, Tay FE, Manimaran M, Seah KH (2006) Garment-based detection of falls and activities of daily living using 3-axis MEMS accelerometer. J Phys Conf Ser 34:1059–1067
20. Kangas M, Konttila A, Lindgren P, Winblad I, Jamsa T (2008) Comparison of low-complexity fall detection algorithms for body attached accelerometers. Gait Posture 28(2):285–291
21. Brown G (2005) An accelerometer based fall detector: development, experimentation, and analysis. SUPERB internship report, University of California, Berkeley
22. Hwang JY, Kang JM, Jang YW, Kim HC (2004) Development of novel algorithm and real-time monitoring ambulatory system using Bluetooth module for fall detection in the elderly. In: Proceedings of 26th annual international conference of the engineering in medicine and biology society, IEEE-EMBS 2004
23. http://mover.projects.fraunhofer.pt/. Accessed 3 March 2010
24. Ganti RK, Jayachandran P, Abdelzaher TF, Stankovic JA (2006) SATIRE: a software architecture for smart AtTIRE. In: Proceedings of the 4th international conference on mobile systems, applications and services, MobiSys, 2006
25. Karantonis DM, Narayanan MR, Mathie M, Lovell NH, Celler BG (2006) Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE Trans Inf Technol Biomed 10(1):156–167
26. Fu Z, Culurciello E, Lichtsteiner P, Delbruck T (2008) Fall detection using an address-event temporal contrast vision sensor. In: Proceedings of IEEE international symposium on circuits and systems, ISCAS 2008
27. Sixsmith A, Johnson N (2004) A smart sensor to detect the falls of the elderly. Pervasive Comput 3(2):42–47
28. Miaou S, Sung P, Huang C (2006) A customized human fall detection system using omni-camera images and personal information. In: Proceedings of 1st distributed diagnosis and home healthcare conference, 2006
29. Jansen B, Deklerck R (2006) Context aware inactivity recognition for visual fall detection. In: Proceedings of pervasive health conference and workshops, 2006
30. Hansen TR, Eklund JM, Sprinkle J, Bajcsy R, Sastry S (2005) 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
31. Zhang T, Wang J, Liu P, Hou J (2006) Fall detection by embedding an accelerometer in cellphone and using KFD algorithm. Int J Comput Sci Netw Secur 6(10):277–284
32. Chen J, Kwong K, Chang D, Luk J, Bajcsy R (2005) Wearable sensors for reliable fall detection. In: Proceedings of 27th annual international conference of the engineering in medicine and biology society, IEEE-EMBS 2005
33. http://www.cnn.com/2010/TECH/03/03/ishoe.mit.award/index.html. Accessed 3 March 2010
34. Kumar S (1999) Biomechanics in ergonomics. CRC Press, Boca Raton
35. Sacher A (1996) The application of forensic biomechanics to the resolution of unwitnessed falling accidents. J Forensic Sci 41(5):776–781
36. Belongie S, Malik J and Puzicha J (2002) Shape matching and object recognition using shape context. IEEE Trans Pattern Anal Mach Intell 24(24):509–522
37. Ding L, Belkin M (2008) Component based shape retrieval using differential profiles. In: Proceedings of ACM international conference on multimedia information retrieval
38. http://en.wikipedia.org/wiki/HTC_Dream. Accessed 26 Sept 2009
39. http://www.jitterbug.com. Accessed 30 Sept 2009
40. http://www.shopaservice.com/articles/jitterbug-cell-phones. Accessed 30 Sept 2009
41. http://www.lakeshore.com/mag/ga/gm410po.htm. Accessed 23 July 2009
42. World Health Organization (2009) Electromagnetic fields and public health: static electric and magnetic fields. http://www.who.int/mediacentre/factsheets/fs299/en/print.html. Accessed 26 July 2009
43. Yoshida T, Mizuno F, Hayasaka T, Tsubota K, Wada S, Yamaguchi T (2005) A wearable computer system for a detection and prevention of elderly users from falling. In: Proceedings of the 12th international conference on biomedical engineering, 2005
44. Charpentier PJ (1996) A hip protector based on airbag technology. Bone 18(1):S117
45. Fukaya K (2002) Fall detection sensor for fall protection airbag. In: Proceedings of the 41st SICE annual conference