Grammar-Based, Posture- and Context-Cognitive Detection for Falls with Different Activity Levels

Qiang Li
Department of Computer Science
UVA Center for Wireless Health
University of Virginia
Charlottesville, VA 22904, USA
lq7c@cs.virginia.edu

John A. Stankovic
Department of Computer Science
UVA Center for Wireless Health
University of Virginia
Charlottesville, VA 22904, USA
stankovic@cs.virginia.edu

ABSTRACT

Falls are dangerous for the aged population as they result in serious detrimental consequences. Therefore, many fall detection methods have been proposed. Most of these methods characterize falls by large accelerations and fast body orientation changes. However, certain activities like sitting down quickly, vigorous gaits, and jumping, also show these characteristics, and thus are hard to distinguish from real falls. Moreover, many falls in the elderly are slow falls which show lower activity levels. Existing work fails to detect slow falls effectively because they only identify falls with high activity levels.

In this paper, we present a grammar-based fall detection framework which not only better distinguishes fall-like activities from real falls, but also emphasizes the detection of slow falls. We utilize posture information extracted from on-body sensors and context information collected from sensors deployed in the house to reduce false positives. A fall in our framework is detected as a sequence of sensor events. We provide a context-free grammar to define these sequences so that the framework can be easily extended to detect more kinds of falls. Our case study shows that our method can distinguish various fall-like activities from real falls and can also effectively detect both fast falls and slow falls. The integration evaluation shows that our method achieves both high sensitivity and high specificity.

Keywords

Fall Detection, Body Sensor Networks, Personal Monitoring

1. INTRODUCTION

Falls are one of the most detrimental events for the aged population. About one out of three senior people 65 years or older have at least one fall per year, and falls are the leading cause of injury-related hospitalization for elderly people [19]. Detecting falls accurately can reduce the severe consequences, and thus is of great importance.

Wireless Health '11, October 10–13, 2011, San Diego, USA

However, falls are difficult to accurately detect due to three reasons. First of all, certain fall-like activities are hard to distinguish from real falls. Existing solutions mainly use accelerometers [17] to detect falls, because falls are usually characterized by larger accelerations compared to normal daily activities. However, focusing only on large accelerations can result in many false positives by detecting fall-like activities, such as sitting down quickly, or vigorous walking or jumping, as falls. Some other fall detection algorithms assume that falls are often accompanied with a prominent change of body orientation, e.g. a typical fall can be detected if one's body orientation changes from standing upright to lying prone horizontally on the floor [7]. However, these kinds of systems also cause false positives when detecting activities ending in horizontal position, such as lying onto bed quickly.

Second, falls have different characteristics according to their causes. Rubenstein et al. [27] summarized major causes of falls in elderly people and their relative frequencies. The first four causes for falls in elderly are: falls stemming from environmental hazards (31%), gait/balance disorder (17%), dizziness and vertigo (13%), and drop attacks (9%). In these categories, falls caused by environmental hazards and gait problems usually happen faster than those caused by dizziness and drop attacks. However, previous work on fall detections only focuses on fast falls, yet cannot recognize slow falls, which causes lots of false negatives.

Third, unlike activities of daily living (ADL), falls are accidents, so it is extremely difficult to collect data of real falls of the elderly. Therefore, existing work on fall detection is mainly based on falls simulated by younger people [6, 22]. However, not all types of falls can be handled by simulation, e.g. falls caused by dizziness cannot be simulated. Therefore, it is important to have a generic fall detection framework which can be adjusted and extended as more real falls are reported.

To overcome the problems of existing fall detection methods discussed above, in this paper, we present a grammar-based, posture- and context-cognitive fall detection framework that can detect falls with different activity levels. Our framework can be easily adjusted to detect falls more accurately as people gradually collect more data of real falls. Our fall detection framework makes the following contributions:

- *Our method can be easily tuned and extended.* Falls are difficult to detect accurately because there are so many different types of them. In our framework, we propose a context-free grammar to define various falls as sequences of sensor events, so that our system can be easily extended to handle new types of falls. The formal definition and generic detection framework for falls have not been studied before.

- *Our method is posture-cognitive and person-specific.* The posture and movement levels are determined in a novel way by using clustering and learning techniques, which makes our method person-specific. Training on posture and movement levels (not on falls) only requires several minutes.

- *Our method reacts to consequences of falls.* When evaluating the fall process, our method evaluates the immediate consequence of all falls — resting on the ground, combined with a period of low-level activities. If we detect the condition, we use the previous 5 seconds of data to determine if a fall actually occurred.

- *Our method is context-cognitive.* Environmental sensors attached on a bed or couch are used to eliminate false positives caused by fall-like activities also ending with a lying posture such as lying on the bed quickly.

- *Our method provides a mechanism to detect falls with different activity levels.* According to Rubenstein et al.[27], falls in elderly are caused by several different reasons such as environment-related accidents, gait and balance disorder, and dizziness and vertigo. Falls caused by environmental factors usually happen faster than falls caused by dizziness. By using a context-free grammar to define different kinds of falls, our method detects both fast and slow falls. The detection of slow falls has not been studied by previous work.

- *Our method achieves high accuracy.* In our evaluation, our method detects all 32 fast falls and 22 out of 24 slow falls from normal or fall-like activities.

2. RELATED WORK

Existing fall detection solutions mainly analyze acceleration data to detect falls. Prado et al. [9, 24] use a four-axis accelerometer, fixed to the back, at the height of the sacrum, to detect falls. They fix the sensor to the skin to minimize the influence of the displacement of the sensor. Then the system determines if there is an impact according to high frequency acceleration components. Mathie et al. [18] utilize a single, waist-mounted, tri-axial accelerometer to detect falls. In their method, if the peek acceleration exceeds a preset threshold, an abnormal event like a fall, a stumble or a collision may have happened. Lindemann et al. [17] integrate a tri-axial accelerometer into a hearing aid housing which is fixed behind the ear, and use thresholds for acceleration and velocity to differentiate falls and Activities of Daily Living (ADL). Kangas et al. [14] study the acceleration of the waist, wrist, and head for falls and ADL, and show that measurements from the waist and head are more useful for fall detection. Their results also show that the acceleration value ranges are overlapping for falls and ADL, which means simple thresholds alone are not optimal for practical fall detection. Bourke et al. [6] place two tri-axial accelerometers at the trunk and thigh, and derive four thresholds, upper and lower thresholds for both the trunk and thigh. Exceeding any of the four thresholds indicates a fall occurred. Jeong et al. [13] implement an acceleration monitoring system for convenient monitoring of activity volume and recognition of emergent situations such as falls. They use DSVM (differential signal vector magnitude) to distinguish falls from other dynamic states like running and walking. Mobile phones with accelerometers are also used to detect falls, e.g. Jiangpeng et al. [8] use an HTC G1 phone to detect falls by examining if the acceleration exceeds predefined thresholds. The problem with these methods is that some normal activities such as sitting down quickly and jumping also feature large accelerations. Therefore, only using acceleration for fall detection causes many false positives. In addition, these methods cannot detect slow falls which do not feature large accelerations.

To improve fall detection accuracy, some solutions utilize both acceleration and body orientation information to detect falls. Noury et al. [21] develop a fall detector consisting of three sensors: a tilt sensor to monitor body orientation, a piezoelectric accelerometer to monitor vertical acceleration shock, and a vibration sensor to monitor body movements. A fall is detected if the body position is standing and the acceleration exceeds the threshold. In their later work [20], they develop a sensor with two orthogonally oriented accelerometers and use this sensor to monitor the inclination and inclination speed to detect falls. Chen et al. [7] monitor the body orientation before and after an impact, and detect falls based on the change in orientation. Purwar et al. [25] use a chest worn triaxial accelerometer to record the acceleration and the tilt angle between the sensor and the vertical direction. A fall is detected if both the acceleration and the angle cross the thresholds. Leijdekkers et al. [15] present a prototype system for remote healthcare monitoring. In this system, the body orientation is analyzed after a large acceleration, and a fall is detected if the orientation is horizontal or not upright after some period of inactivity. Bourke el al. [5] develop a threshold-based fall detection algorithm using a bi-axial gyroscope sensor. They put the gyroscope at the sternum, and measure the angular velocity, angular acceleration, and change of the trunk angle to detect falls. Body orientation can help improve the fall detection accuracy, but using one single device can only monitor the orientation of the trunk, more posture information cannot be collected using this kind of method. Qiang et al. [16] propose a three-phase fall detection algorithm. Two TEMPO nodes [4], each of which features a triaxial accelerometer and a triaxial gyroscope, are attached on the chest and thigh. Using these two sensors, the activity volume, body postures, and the transition process from a dynamic state to a static posture are monitored. If the acceleration and rotational rate before a lying posture exceed thresholds, a fall is detected. This method uses preset thresholds to recognize postures and evaluate the transition process, which makes it less applicable for different people and not able to detect slower falls.

There is also work trying to detect falls before they really happen. Nyan et al. [22] attach two sensors on torso and thigh. Each sensor contains a 3-D accelerometer and a 2-D gyroscope. They show that falls can be detected with an average lead-time of 700ms before the impact occurs. But their work still uses thresholds of accelerations and rotational rates to determine whether a fall is to happen.

Some other work uses cameras or image sensors to confirm the fall detection results generated by accelerometers.

Hansen et al. [11] use a camera phone to communicate with elderly people by the emergency service when a fall is detected. Tarbar et al. develop a home care network [28]. In their system, a user wears a badge node providing user-centric event sensing functions such as detecting falls, and the appropriate location of the user is detected by measuring the Received Signal Strength Indicator (RSSI). When a fall is detected by the badge, the most nearby image sensor node is activated for further posture analysis. Though using cameras can improve the fall detection accuracy, it is not feasible when the user is in an open area. In addition, privacy is a big concern for people regarding using a camera to monitor them all the time.

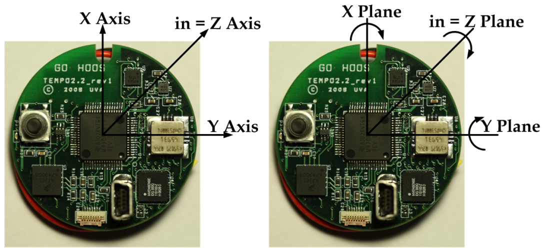
Besides solutions outlined above, complex inference techniques are also utilized to improve activity recognition accuracy. Raghu et al. [10] attach five accelerometers to a jacket, and perform activity recognition by using Hidden Markov Models to analyze acceleration data. Quwaider et al. [26] distinguish activities with different activity intensity levels, such as run, walk, and sit, by transforming collected acceleration data to the frequency domain. Low activity postures, such as sit and stand, are distinguished by using Hidden Markov Models to analyze RSSI values from multiple sensors. Pham et al. [23] compute the relative energy distribution over the body according to the acceleration data collected from right-hand top, right-hand bottom, left-hand top, left-hand bottom, and waist. Then the activity recognition is performed using a naive Bayes learner. He et al. [12] use Discrete Cosine Transform (DCT) to transform acceleration data to the frequency domain and then use Support Vector Machine (SVM) to classify different activities. Using complicated inference techniques makes the activity recognition system more reliable, but it also costs significant computational resources. Moreover, most of these methods need to learn the activity patterns before doing the classification, but activity patterns for falls are extremely difficult to obtain.

There are also commercial products able to detect falls. Many are based on a watch device that includes accelerometers using thresholds as discussed above. WellAWARE [1] markets a floor vibration sensor used to detect falls in specific locations such as near a bed. This system may consider many normal activities such as stamping as falls. Philips' Lifeline [2] uses a help button to issue medial alerts when a fall happens. However, when a really serious fall happens, people may not be able to push the button. GE's QuietCare [3] system detects falls using motion sensors: if an elderly person stays in one place like the bathroom for a long time, a fall might have happened. This system lacks fast response. While these products are being used, unfortunately, as far as we know, there is no substantial data published as to the false alarm rates for these products.

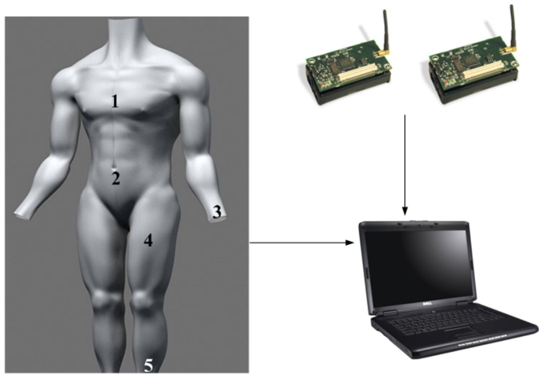
3. DATA ACQUISITION

Two kinds of sensors are used in our fall detection system: the TEMPO 3.1 nodes [4] and the widely used MICAz motes with MTS310 sensor boards. The TEMPO nodes are used to monitor the acceleration and rotational rates of different parts of the body, and the MICAz motes are used to monitor the vibration of specific furniture such as a bed to retrieve the subject's location context information.

The TEMPO 3.1 node includes a tri-axial accelerometer and a tri-axial gyroscope as shown in Figure 1(a). The



(a)



(b)

Figure 1: (a) The TEMPO 3.1 sensor node; (b) The data acquisition system setup.

MMA7261QT tri-axial accelerometer, made by Freescale Semiconductor, can monitor acceleration within a range of $\pm 10g$. The tri-axial gyroscope consists of an InvenSensce IDG-300 dual-axis gyroscope and an Analog Devices ADXRS300 Z-axis gyroscope. The IDG-300 can monitor angular velocity between $\pm 500°/s$. The ADXRS300 can monitor angular velocity between $\pm 300°/s$. The sensors are controlled by an TI MSP430F1611 microcontroller. The sampling rate is set to 120Hz, a bandwidth exceeding the characteristic response of human activities, to guarantee body movement details can be captured.

The MTS310 sensor board features a bi-axial accelerometer, which can monitor acceleration between $\pm 2g$. The sampling rate is set to 2Hz, which is sufficient for monitoring furniture vibrations.

Figure 1(b) shows the setup of our data acquisition system. In our experiments, we attach 5 TEMPO nodes to human body as shown in Figure 1(b). Using 5 nodes enables us to collect significant amounts of data to obtain very high accuracy and determine which nodes and locations are most critical. Due to space limitations we do not show these performance results, but we see that at least 3 nodes are required (chest, ankle, and wrist). The extra 2 nodes do improve performance and in the future when such nodes can be unobtrusively embedded in clothes using 5 nodes (adding on the thigh and waist) would be preferred.

The MICAz nodes are fixed to furniture including a bed and a chair. The subject is located by monitoring the vibration of the furniture.

To make sure our system works for a wide variety of situations, during the following experiments, three graduate students engaged in a battery of tests designed to simulate falls (fast fall forward/backward/leftward/rightward/ag-ainst wall

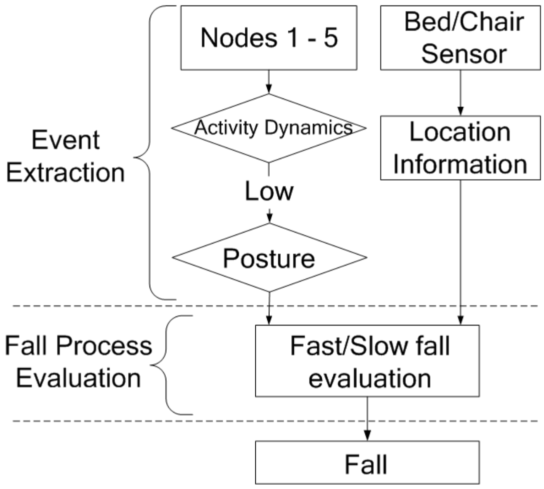


Figure 2: The main process to detect falls: posture recognition, localization, and fall process evaluation for both fast and slow falls.

and slow falls), fall-like activities (sit down fast upright, sit down fast reclined, jump into bed, stumble, jump), and normal activities (stand, sit, lie, walk, run). All fall data was taken on hard surfaces.

4. FALL DETECTION FRAMEWORK

4.1 The Fall Detection Process

In our fall detection framework, we use a context-free grammar to define various falls as sequences of sensor events. These events can be the raw readings from on-body sensors exceeding thresholds, or more high-level information, such as the posture and location of the subject, inferred from the low-level sensor readings. As shown in Figure 2, the detection process can be divided into two main steps: *event extraction* and *fall process evaluation*.

During event extraction we buffer a short period of recent sensor readings and calculate high-level features. These raw readings and derived features are used later during fall process evaluation to determine if a fall has happened. In our current system, we calculate two kinds of high-level features: the posture and location of the subject. Unlike previous work using preset inclination thresholds [29] to determine postures, no preset thresholds are used in our system when calculating the posture of the subject by using clustering and learning techniques. Therefore, our posture recognition algorithm adapts to different users automatically. The location of the subject is detected by the sensors attached on bed and couch.

The key part of the fall process evaluation is a set of rules. Each rule defines a type of fall using the context-free grammar so that it can be parsed automatically. Then we compare the rules against the readings and features collected in the previous step to determine if a fall has happened. In this step we evaluate both fast falls and slow falls according to their respective rules.

In the following of this section, we first present the context-free grammar used to define falls, then discuss the extraction of posture information from on-body sensor readings, and discuss how to retrieve context information from environmental sensors. Last we present the fall process evaluation.

$$F \rightarrow S \quad (1)$$
$$S \rightarrow E|ETS \quad (2)$$
$$E \rightarrow (E,E)|(SENSOR, FEATURE)|P|LOC \quad (3)$$
$$T \rightarrow (time, C)|(time, time)|\varepsilon \quad (4)$$
$$SENSOR \rightarrow chest|waist|wrist|thigh|ankle \quad (5)$$
$$FEATURE \rightarrow (threshold, C)|(threshold, threshold) \quad (6)$$
$$C \rightarrow < | > \quad (7)$$
$$P \rightarrow standing|sitting|lying \quad (8)$$
$$LOC \rightarrow bed|couch|other \quad (9)$$

Table 1: The production rules of the context-free grammar for defining fall processes.

Non-terminal	Meaning
F	a fall process
S	a sequence of sensor events
E	a sensor event
T	the time interval between two sensor events
$SENSOR$	which sensor the readings are collected from
$FEATURE$	the feature of the collected readings
C	a comparison with time interval or threshold
P	the subject's posture
LOC	the subject's location

Table 2: The meanings of non-terminals in the context-free grammar.

4.2 Context-Free Grammar for Defining Falls

In our framework, a fall is defined as a sequence of sensor events. This sequence can be generated using our context-free grammar shown in Table 1. The meanings of the non-terminals of the grammar are shown in Table 2. Some of the expressions in Table 1 also have special meanings as shown in Table 3.

Rule 3 of the grammar shows that the sensor events used to define falls can be divided into two categories: sensor readings exceeding *thresholds* (or within two *thresholds*) and high-level features derived from sensor readings including the posture (P) and location (LOC) of the subject. The initial *thresholds* used in specific rules are determined based on simulated falls as shown later in Sec. 4.5.1, and Sec. 4.5.2.

Expression	Meaning
(E, E)	two events happen at the same time
$(time, C)$	time interval between two events is smaller (<) or larger (>) than *time*
$(time, time)$	time interval between two events is between two values of *time*
$(threshold, C)$	sensor readings are smaller (<) or larger (>) than *threshold*
$(threshold, threshold)$	sensor readings are between two values of *threshold*

Table 3: The meanings of expressions in the context-free grammar.

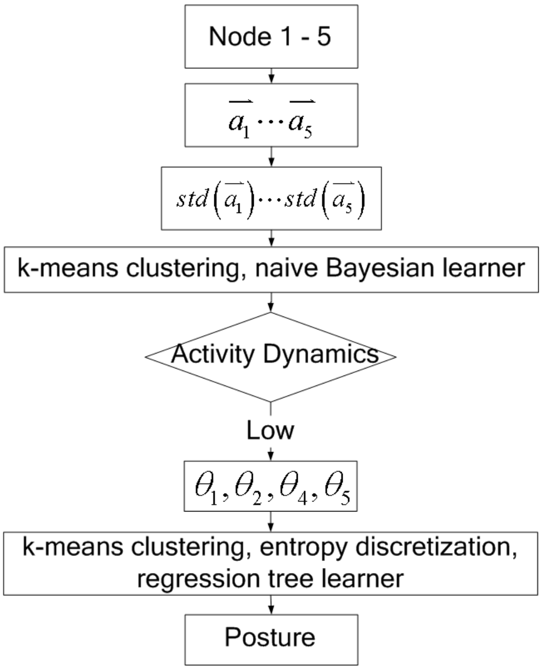


Figure 3: The process of posture recognition.

Our framework easily permits adjusting the *thresholds* if needed.

To demonstrate the usage of the proposed grammar, we use it to define several falls and fall-like activities:

Normal fast fall:
$(chest, (threshold, >))(lying, other)$

Forward slow fall:
$(thigh, (threshold, >))(2s, <)$
$(wrist, (threshold, >))(lying, other)$

Sitting down fast to couch:
$(waist, (threshold, >))(sitting, couch)$

Lying onto bed quickly:
$(chest, (threshold, >))(lying, bed)$

From these examples, we can see it is straightforward to define falls and fall-like activities using the grammar.

Our framework can be extended easily in two ways:

- *Extending the grammar.* In actual deployments, more sensors can be used to get additional information. For example, we can attach the MICAz node on a shower head to detect if the subject is showering. In this case, *showering* can be added as a new terminal in Rule 3.
- *Adding new rules.* Under a given grammar such as in Table 1, to detect a new kind of fall or fall-like activity, we only need to write a new rule using this grammar, and then the rule can be parsed automatically so that the system will be able to recognize this specific type of fall or fall-like activity.

4.3 Posture Recognition

There are many different types of falls, but most of them have the same immediate consequence — lying on the ground, combined with a period of low-level activities. Therefore, posture is an important feature to detect falls. Figure 3 shows the process of posture recognition in our framework: we first divide activities into three categories according to their dynamic levels, then we extract posture information when the subject is in low-level dynamic activities. During the process, by using the clustering and learning techniques, we do not need to use preset thresholds to detect postures.

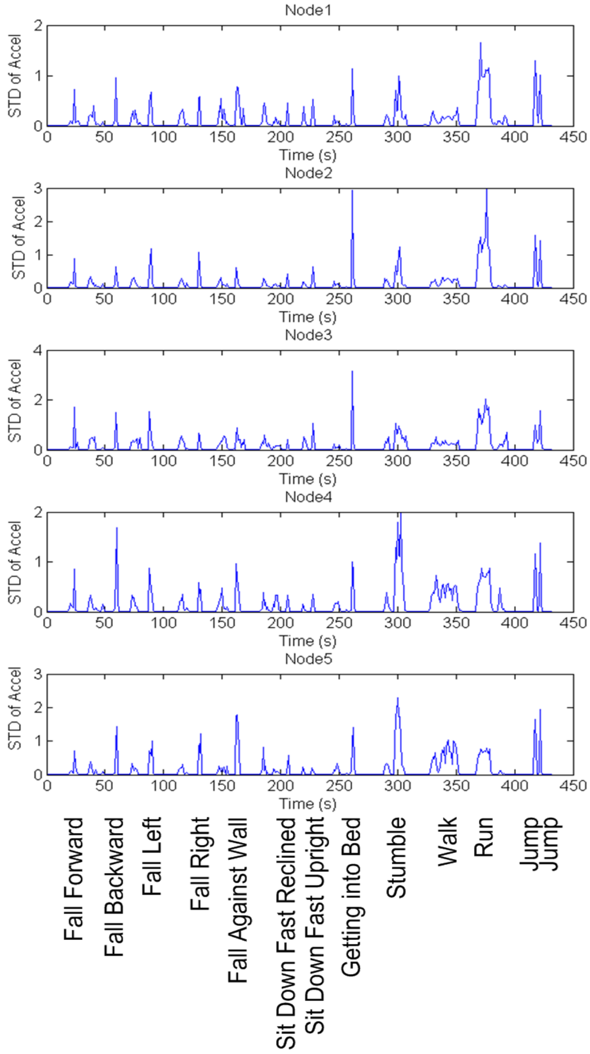


Figure 4: The standard deviation of the linear accelerations at the chest (Node 1), waist (Node 2), wrist (Node 3), thigh (Node 4), and ankle (Node 5) for various activities.

4.3.1 Activity Dynamic Level

We use the standard deviation of the acceleration readings from TEMPO nodes to determine the person's activity level. The monitoring cycle is one second, which means we calculate the standard deviation of the collected data every 120 samples.

The acceleration of each sensor can be calculated using Equation (10), where a_i is the vector magnitude linear acceleration of Node i, and a_{i_x}, a_{i_y}, a_{i_z} are the acceleration readings along the x-, y-, and z-axis.

$$a_i = \sqrt{a_{i_x}^2 + a_{i_y}^2 + a_{i_z}^2} \quad (i = 1, 2, \cdots, 5) \tag{10}$$

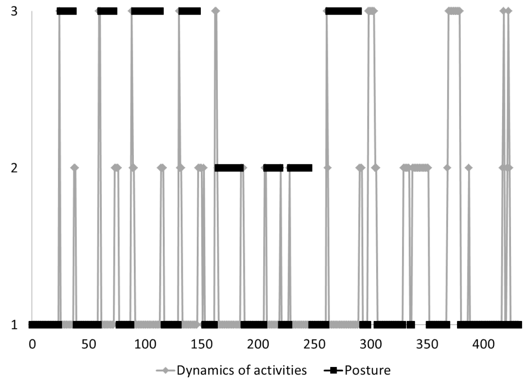


Figure 5: Gray line: clustering results of dynamic levels of different activities (1: Low; 2: Medium; 3: High); Black line: posture clustering results when activity dynamic levels are classified as low (1: standing; 2: sitting; 3: lying).

Figure 4 shows the standard deviation of the accelerations at different parts of the body when performing various activities. From Figure 4, we can see some activities like running are more dynamic than others like walking or standing still. Therefore, according to different standard deviation values, we divide human activities into three categories: high dynamic activities like running or jumping, medium dynamic activities like transactions between postures or walking, and low dynamic activities like standing, sitting, or lying.

Instead of using simple thresholds to determine activity categories, we use k-means clustering to partition activity dynamics into three categories. The gray line in Figure 5 shows the results of cluster analysis. The input data of the cluster analysis is the standard deviations of the acceleration as shown in Figure 4. From Figure 5 we can see that the results of clustering activity levels of the activities shown in Figure 4 are all correct: falling forward/backward/left/right/against wall, getting into bed, stumbling, running, and jumps are clustered as high dynamic activities (Class 3); activities including sitting down fast, walking, and the transactions between activities are clustered as medium dynamic activities (Class 2); when the person is standing, sitting, and lying, the activity dynamics are clustered as low (Class 1). Based on the clustering results, we build a naive Bayesian classifier to differentiate activity dynamic levels.

4.3.2 Extracting Posture Information

When activity dynamic levels are classified as low during one second, we recognize the postures (standing, sitting, or lying) of this one second time period to determine if there is possibly a fall.

In our experiment, we first perform an initial calibration for the stationary standing posture. In this step, we record the accelerometer readings from Nodes 1, 2, 4 and 5 as shown in Figure 1(b). Node 3 is not used because it is attached on the wrist and does not strongly relate to postures. Then we calculate the angle changes of each node based on the accelerometer readings along each axis using Equation (11).

$$\theta_i = \frac{180}{\pi} \arccos\left(\frac{\hat{a}_{i_x} a_{i_x} + \hat{a}_{i_y} a_{i_y} + \hat{a}_{i_z} a_{i_z}}{\sqrt{\hat{a}_{i_x}^2 + \hat{a}_{i_y}^2 + \hat{a}_{i_z}^2} \cdot \sqrt{a_{i_x}^2 + a_{i_y}^2 + a_{i_z}^2}}\right) \tag{11}$$

Source	Xaccel	Yaccel	Deviation
1	496	507	5
1	497	507	2
1	496	508	4
1	497	507	2
1	497	507	2
1	497	507	2
1	496	507	5
1	497	507	2
1	497	507	2
1	497	507	2
1	491	508	4
1	496	507	5
2	505	517	8
2	506	517	5
2	505	517	8
2	505	517	8
2	505	517	8
2	506	517	5
2	506	517	5
2	506	518	2
2	505	517	8
2	506	517	5
2	506	517	5
2	505	517	8

Figure 6: Collect readings from MICAz motes attached on bed and couch

In Equation (11), $i = 1, 2, 4, 5$, θ_i is the orientation change of Node i, $\hat{a}_{i_x}$, $\hat{a}_{i_y}$, $\hat{a}_{i_z}$ are the acceleration readings of Node i during the calibration phase along the x-, y-, and z-axis, and a_{i_x}, a_{i_y}, a_{i_z} are the mean values of the accelerometer readings along each axis during low dynamic activities.

The black line in Figure 5 shows the results of posture clustering. Note that posture clustering results are only available when activity dynamic levels are low. In this step, we use k-means clustering to partition postures into three categories: standing (Class 1), sitting (Class 2) and lying (Class 3). The input data for posture clustering is the vector $(\theta_1, \theta_2, \theta_4, \theta_5)$. In Figure 5, all postures are clustered correctly. Based on the results we use a regression tree learner to build the posture classifier, which is used to recognize postures.

In Sec. 4.3.1 and Sec. 4.3.2, we first determine the activity level of the subject, then extract posture information if the subject is in low activity level. The learning process for posture recognition only requires a few minutes, and during this period the subject performs normal daily activities and three postures (standing, sitting, and lying). Unlike most previous work, by using clustering and learning techniques in our method, the whole posture recognition process can be automatically adjusted across different subjects, and no predefined thresholds are used. This makes our method person-specific.

4.4 Context Information Collection

Some activities like lying onto the bed quickly are extremely difficult to distinguish from real falls only using body sensors. However, by combining context information such as the location of the subject, these fall-like activities can be distinguished from real falls easily: when the on-body sensor readings indicate a fall has happened, if the location of the subject is on the bed, then it is obviously a false alarm. Sensors in environment are becoming common in assisted living systems, e.g. GE's QuietCare[3] uses environmental sensors to learn residents' routine. In our system, we collect context information using ambient environmental MICAz motes (as shown in Figure 6).

Figure 6 shows a typical deployment of environmental sensors to retrieve location information. In this setting, one MICAz mote (with MTS310 sensor board) is attached onto the bed, the other is attached onto the couch. Every half second each MICAz mote sends x- and y-axis acceleration readings back to the sever. The right of Figure 6 shows the data collected from these sensors. If the deviation of the collected data is larger than a threshold value (we use 100 in our experiments), we know the subject is just sitting down to the couch or lying onto the bed.

In fact, many other kinds of context information can be collected from environmental sensors. For example, the assisted living system AlarmNet [30] uses X10 sensors to track which room the subject is in. By utilizing the data collected by these assisted living systems, we can have more knowledge about the subject, which can help improve fall detection accuracy.

4.5 Fall Process Evaluation

After knowing the subject is resting on the ground with low-level movements (Sec. 4.3), and not on a bed or in the couch (Sec. 4.4), our method checks the data for 5 seconds earlier to evaluate whether the process that achieved this posture is a fall. We do not use learning techniques during fall process evaluation because falls are rare accidental events and it is not feasible to train the system by letting elderly people fall. The *thresholds* used for the fall process evaluation are determined by simulated falls.

Falls have different characteristics according to their causes. Rubenstein et al. [27] summarized major causes of falls in elderly people and their relative frequencies. The first four causes for falls in elderly are: falls stemming from environmental hazards (31%), gait/balance disorder (17%), dizziness and vertigo (13%), and drop attacks (9%). In these categories, falls caused by environmental hazards and gait problems usually happen faster than those caused by dizziness and drop attacks. However, previous work on fall detections discussed in Section 2 only focuses on fast falls.

In this section, we show that both fast and slow fall processes can be defined and detected using the production rules in our framework.

4.5.1 *Fast Falls*

In the examples in Sec. 4.2, we define fast falls as

$$(chest, (threshold, >))(lying, other).$$

Though this definition is enough as a demonstration of the usage of the proposed grammar, in a real fall detection system, we need to figure out which *LOC* is more useful for detecting a fall, and what the specific value of the *threshold* should be. To solve this problem, three graduate students performed five kinds of fast falls (fall forward/backward/leftward/rightward, fall after stumbling) and typical normal daily activities (walk, sit down, lie down) two to three times.

Figure 7 shows the maximum acceleration of each node during different activities. From Figure 7, we can see that in spite of some overlap the max acceleration during a fall is most of the time much larger than during normal activities. Therefore, as shown here and in many previous works we can use preset thresholds to detect fast falls. Nodes attached on chest (Node 1), waist (Node 2), and thigh (Node 4) are more useful because their max accelerations have almost no overlap with those of normal activities. However, accelerations of nodes attached on wrist (Node 3) and ankle (Node 5) often have overlap between different activities because of impulsive movements, thus they are not suitable for detecting fast falls. In our method, a fast fall process is detected if $a_1 > 3.0g$ and $a_4 > 3.0g$. Therefore, the rule used to detect fast falls can be written as Equation (12).

$$((chest, (3.0g, >)), (thigh, (3.0g, >)))(lying, other) \quad (12)$$

4.5.2 *Slow Falls*

Besides fast falls, falls caused by dizziness and drop attacks are also common in elderly people. Drop attacks are falls associated by sudden leg weakness, but without dizziness. For elderly people, lacking exercise results in poor muscle condition, decreased strength, and loss of flexibility. Therefore, when bending, reaching, or rising from a chair or bed, elderly people are prone to falls.

These falls are not as sudden as fast falls, and the accelerations of these falls are not as large. Therefore, it is hard to distinguish them only using acceleration thresholds. Figure 8 shows two simulated slow falls: the person rose from a chair, then fell forward to the ground before he could stand steadily. During the fall process, the accelerations of both Node 1 and Node 4 are under the black line ($3.0g$) and thus not large enough to be detected as a fall according to Equation (12).

However, in this kind of fall, people usually push their hands out to cushion the body, so there are large accel-

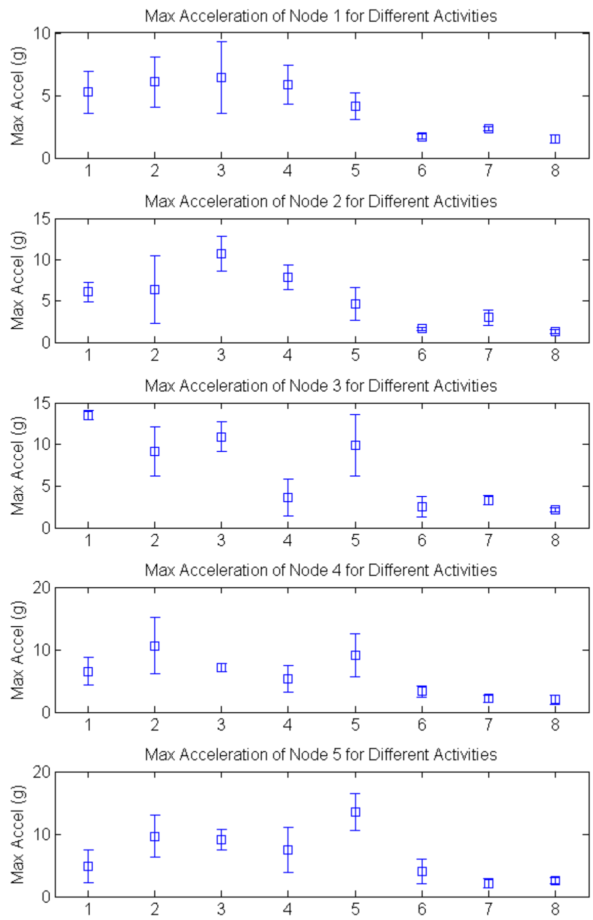


Figure 7: The max acceleration of nodes for different activities: 1) fall forward; 2) fall backward; 3) fall leftward; 4) fall rightward; 5) stumble and fall; 6) walk; 7) sit down; 8) lie down.

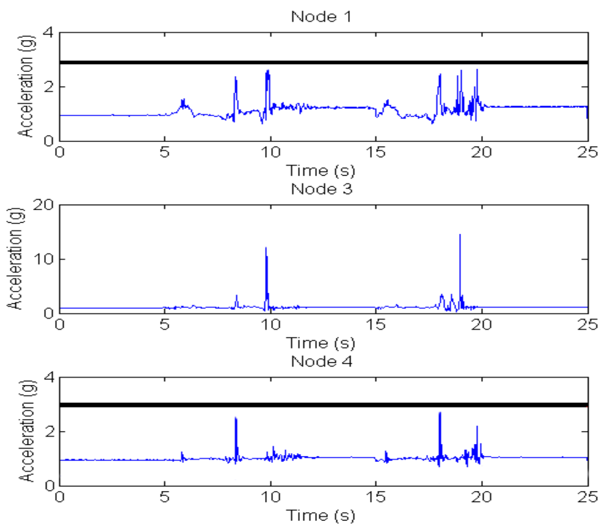


Figure 8: Accelerations of Node 1 (chest), Node 3 (wrist), and Node 4 (thigh) for a slow fall caused by dizziness.

eration readings from wrist (Node 3) as shown in Figure 8. Thus the fall process can be detected as a sequence of sensor events: sitting posture, large accelerations from wrist, and resting on the ground with low-level movements. Usually the movement of wrist is not suitable for fall detection because its impulsiveness, however, as an event of a sequence, it can be useful. Therefore, the rule used to detect this kind of slow falls can be written as Equation (13).

$$sitting(wrist, (8.0g, >))(lying, other) \tag{13}$$

5. EVALUATION

In this section, we evaluate our fall framework by studying two special cases and performing system integration tests. By discussing the first special case, we show how to use context information to distinguish fall-like activities from real falls. In the second case study, we show how our system deals with slow falls. The system integration test then shows the overall performance of our fall detection system.

5.1 Special Case Study

5.1.1 Lying onto Bed Quickly

Lying onto the bed quickly is extremely difficult to distinguish from real falls, because they all feature large accelerations and fast body orientation changes. The first five plots in Figure 9 show six seconds of acceleration from Node 1 to Node 5 when the subject lies onto the bed quickly. In Sec. 4.5.1, we use $a_1 > 3.0g$ and $a_4 > 3.0g$ as thresholds to detect fast falls. However, in Figure 9 we can see every node has max acceleration larger than $3.0g$, so most previous work only based on monitoring acceleration cannot handle this fall-like activity correctly.

However, distinguishing lying onto bed from real falls becomes straightforward if we know the location of the subject. The last plot in Figure 9 shows the accelerometer readings of the MICAz sensor attached to the bed. When the subject gets into bed, the bed sensor shows large readings (from 2s to 4s), and the activity will not be detected as a fall in our system.

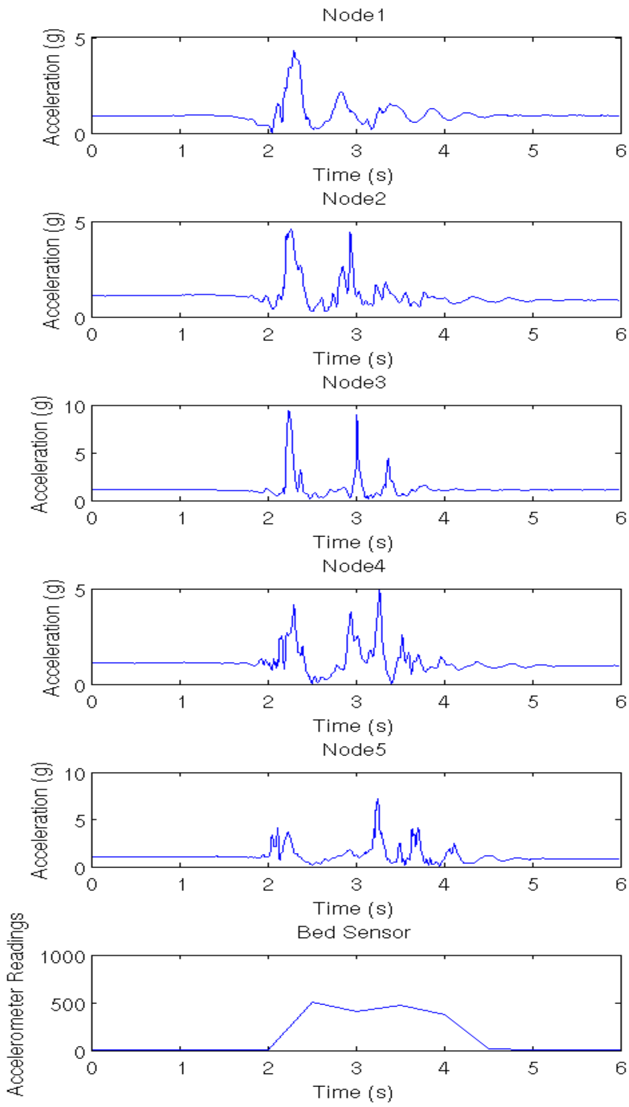


Figure 9: Accelerations of Node 1 (chest), Node 2 (waist), Node 3 (wrist), Node 4 (thigh), Node 5 (ankle), and bed sensor for lying onto bed quickly.

5.1.2 Fall against Wall

In this section, we use an example to illustrate how to detect a new type of fall — fall against a wall — using our system. In this kind of fall, the subject first touches the wall for support, then slips down to the ground, and ends with a sitting position. Figure 10 shows typical accelerometer readings from Node 1 (chest) and Node 4 (thigh) when falling against a wall. When the subject touches the wall, the acceleration of Node 1 (chest) is lager than that of Node 4 (thigh). However, when the subject slips down to the ground, the acceleration of the thigh is larger. Based on this observation, we can define a fall against a wall as

$$((chest, (threshold, >)), (thigh, (threshold, <)))$$
$$((chest, (threshold, <)), (thigh, (threshold, >)))$$
$$(sitting, other)$$

From our experiments, using $3.0g$ as the value of *threshold* can detect falls against a wall.

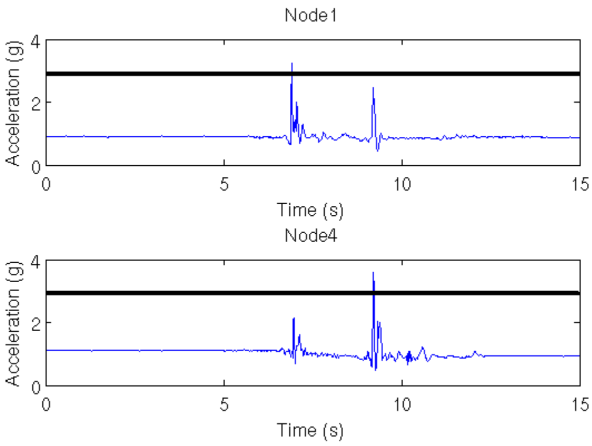


Figure 10: Accelerations of Node 1 (chest) and Node 4 (thigh) for falling against wall

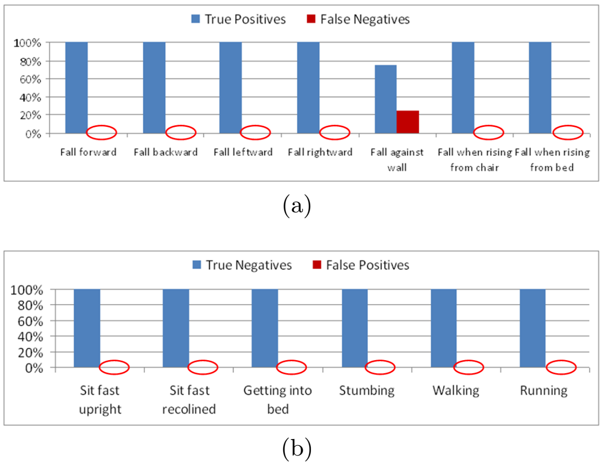


Figure 11: (a) True positive performance (sensitivity); (b) True negative performance (specificity).

5.2 System Integration Tests

To evaluate our algorithm as a complete system, three students performed a series of activities: simulated each kind of falls 8 times (falling forward/backward/left/right/against wall, falling when rising from chairs/bed), 56 falls were simulated in all; also the students performed other fall-like activities and normal activities including sitting down fast upright/reclined, getting into bed, stumbling, walking, running, and jumping. Each fall-like and normal activity was performed 2 times by each person.

Figure 11 shows the sensitivity (true positive performance, the number of detected falls divided by the total number of falls) and specificity (true negative performance, the number of detected fall-like or normal activities divided by the total number of these activities) of our fall detection framework.

Figure 11(a) shows the detection results of both fast falls and slow falls: fast falls include falling forward, backward, leftward and rightward; slow falls include falling against a wall, falling when rising from a chair or bed. All fast falls are detected correctly. Only two slow falls (against wall) out of all 24 slow falls are not detected using our method. This is because we recognized the sitting posture after the fall against a wall as a lying posture. By contrast, 20 out of all 24 slow falls (8 falls against wall, 5 falls when rising from chair, 7 falls when rising from bed) are not detected only using thresholds ($3.0g$) for the accelerations of Node 1 (chest) and Node 4 (thigh).

Figure 11(b) shows the detection results of fall-like activities (including sitting down fast ending with an upright or recolined position, getting into the bed quickly, and stumbling) and some normal daily activities (including walking and running). Our method distinguishes all these activities from falls. For fall-like activities such as sitting down and getting into the bed, the environmental sensors play a key role to eliminate the false alarms. The ending posture (either lying or sitting, definitely not standing) in the definition of falls distinguishes stumbling from real falls.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a grammar-based fall detection framework which uses both posture and context information, and can detect both fast and slow falls. By using a context-free grammar to define various falls as sequences of sensor events, our framework can be easily tuned and extended to detect new types of falls. This feature will be very useful as the system is used by the elderly and new data on falls becomes available. In our work high-level features such as posture and location information are utilized to improve detection accuracy. Slow falls, which are common in the elderly yet have not been studied by previous work, can also be effectively detected using our method. Evaluation shows our method can distinguish most falls from normal activities correctly.

In the paper, only common slow falls are discussed, other slow falls need to be handled by adding new rules in the future. In addition, irregular physiological data like low blood pressure can also be useful to detect falls, as more types of sensor are integrated to BSNs, our framework can be extended to achieve better fall detection accuracy.

7. ACKNOWLEDGMENTS

This work was supported, in part, by NSF EECS-1035303, EECS-0901686, and IIS-0931972.

8. REFERENCES

[1] http://www.wellawaresystems.com.
[2] http://www.lifelinesys.com/content/lifeline-products/classic-pendant.jsp.
[3] http://www.quietcaresystems.com/index_alt.htm.
[4] A. T. Barth, M. A. Hanson, H. C. P. Jr., and J. Lach. Tempo 3.1: A body area sensor network platform for continuous movement assessment. In *Proceedings of the 2009 6th International Workshop on Wearable and Implantable Body Sensor Networks (BSN'09)*, pages 71–76, Berkeley, CA, USA, June 2009.
[5] A. K. Bourke and G. M. Lyons. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical Engineering and Physics*, 30:84–90, Jan 2008.
[6] A. K. Bourke, J. V. O'Brien, and G. M. Lyons. Evaluation of a threshold-based tri-axial accelerometer

fall detection algorithm. *Gait and Posture*, 26:194–199, 2007.

[7] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy. Wearable sensors for reliable fall detection. In *Proceedings of the 27th Annual International Conference of the IEEE EMBS*, pages 3551–3554, Shanghai, China, Sept 2005.

[8] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan. Mobile phone-based pervasive fall detection. *Journal of Personal and Ubiquitous Computing*, 14(7):633–643, Oct. 2010.

[9] A. Díaz, M. Prado, L. M. Roa, J. Reina-Tosina, and G. Sánchez. Preliminary evaluation of a full-time falling monitor for the elderly. In *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, pages 2180–2183, San Francisco, USA, Sept 2004.

[10] R. K. Ganti, P. Jayachandran, T. F. Abdelzaher, and J. A. Stankovic. SATIRE: A software architecture for smart attire. In *Proceedings of MobiSys'06*, Uppsala, Sweden, June 2006.

[11] T. R. Hansen, J. M. Eklund, J. Sprintle, R. Bajcsy, and S. Sastry. Using smart sensors and a camera phone to detect and verify the fall of elderly persons. In *Proceedings of the 2005 3rd European Medical and Biological Engineering Conference*, Prague, Czech Republic, Nov 2005.

[12] Z. Y. He and L. W. Jin. Activity recognition from acceleration data based on discrete cosine transform and svm. In *Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 5041–5044, San Antonio, TX, USA, Oct 2009.

[13] D. U. Jeong, S. J. Kim, and W. Y. Chung. Classifiation of posture and movement using a 3-axis accelerometer. In *Proceedings of the 2007 International Conference on Convergence Information Technology*, pages 837–844, Gyeongju, Korea, 2007.

[14] M. Kangas, A. Konttila, I. Winblad, and T. Jämsä. Determination of simple thresholds for accelerometry-based parameters for fall detection. In *Proceedings of the 29th Annual International Conference of the IEEE EMBS*, pages 1367–1370, Lyon, France, Aug 2007.

[15] P. Leijdekkers, V. Gay, and E. Lawrence. Smart homecare system for health tele-monitoring. In *Proceedings of the 1st International Conference on the Digital Society*, Guadeloupe, French Caribbean, Jan 2007.

[16] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, and J. Lach. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In *Proceedings of the 2009 6th International Workshop on Wearable and Implantable Body Sensor Networks (BSN'09)*, pages 138–143, Berkeley, CA, USA, June 2009.

[17] U. Lindemann, A. Hock, M. Stuber, W. Keck, and C. Becker. Evaluation of a fall detector based on accelerometers: A pilot study. *Medical and Biological Engineering and Computing*, 43(5):548–551, Oct 2005.

[18] M. J. Mathie, J. Basilakis, and B. G. Celler. A system for monitoring posture and physical activity using accelerometers. In *Proceedings of the 23rd Annual International Conference of the IEEE EMBS*, pages 3654–3657, Istanbul, Turkey, Oct 2001.

[19] N. Noury. A smart sensor for the remote follow up of activity and falldetection of the elderly. In *Proceedings of the 2nd International IEEE EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, pages 314–317, 2002.

[20] N. Noury, P. Barralon, G. Virone, P. Boissy, M. Hamel, and P. Rumeau. A smart sensor based on rules and its evaluation in daily routines. In *Proceedings of the 25th Annual International Conference of the IEEE EMBS*, pages 3286–3289, Cancun, Mexico, Sept 2003.

[21] N. Noury, T. Hervé, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron. Monitoring behavior in home using a smart fall sensor and position sensors. In *Proceedings of the 1st International IEEE EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, pages 607–610, Lyon, France, Oct 2000.

[22] M. Nyan, F. E. Tay, and E. Murugasu. A wearable system for pre-impact fall detection. *Journal of Biomechanics*, 41(16):3475–3481, Dec. 2008.

[23] N. Pham and T. Abdelzaher. Robust dynamic human activity recognition based on relative energy allocation. In *Proceedings of the 4th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'08)*, pages 525–530, Santorini Island, Greece, June 2008.

[24] M. Prado, J. Reina-Tosina, and L. Roa. Distributed intelligent architecture for falling detection and physical activity analysis in the elderly. In *Proceedings of the 2nd Joint EMBS/BMES conference*, pages 1910–1911, Houston, USA, Oct 2002.

[25] A. Purwar, D. U. Jeong, and W. Y. Chung. Activity monitoring from real-time triaxial accelerometer data using sensor network. In *Proceedings of the 2007 International Conference on Control, Automation and Systems*, pages 2402–2406, COEX, Seoul, Korea, Oct 2007.

[26] M. Quwaider and S. Biswas. Body posture identification using hidden markov model with a wearable sensor network. In *Proceedings of the ICST 3rd International Conference on Body Area Networks (BodyNets'08)*, pages 1–8, Tempe, AZ, USA, Mar 2008.

[27] L. Z. Rubenstein and K. R. Josephson. The epidemiology of falls and syncope. *Clinics in Geriatric Medicine*, 18(2):141–158, May 2002.

[28] A. M. Tabar, A. Keshavarz, and H. Aghajan. Smart home care network using sensor fusion and distributed vision-based reasoning. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pages 145–154, Santa Barbara, California, USA, Oct 2006.

[29] A. B. Waluyo, W.-S. Yeoh, I. Pek, Y. Yong, and X. Chen. Mobisense: Mobile body sensor network for ambulatory monitoring. *ACM Transactions on Embedded Computing Systems (TECS)*, 10(1):13:1–13:30, Aug. 2010.

[30] A. D. Wood, J. A. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, and R. Stoleru. Context-aware wireless sensor networks for assisted living and residential monitoring. *IEEE Network*, 22(4):26–33, July–Aug. 2008.