Proposal and Experimental Evaluation of Fall Detection Solution Based on Wearable and Depth Data Fusion

Samuele Gasparrini[1], Enea Cippitelli[1], Ennio Gambi[1], Susanna Spinsante[1], Jonas Wåhslén[2], Ibrahim Orhan[2], and Thomas Lindh[2]

[1] Dipartimento di Ingegneria dell'Informazione, Universita' Politecnica delle Marche, Ancona, Italy I-60131
{s.gasparrini, e.cippitelli, e.gambi, s.spinsante}@univpm.it
[2] School of Technology and Health, KTH, Royal Institute of Technology, Stockholm, Sweden SE-100 44
{jonas.wahslen, ibrahim.orhan, thomas.lindh}@sth.kth.se

Abstract. Fall injury issues represent a serious problem for elderly in our society. These people want to live in their home as long as possible and technology can improve their security and independence. In this work we study the joint use of a camera based system and wearable devices, in the so called data fusion approach, to design a fall detection solution. The synchronization issues between the heterogeneous data provided by the devices are properly treated, and three different fall detection algorithms are implemented. Experimental results are also provided, to compare the proposed solutions.

Keywords: Depth camera, Inertial sensor, Data fusion, Synchronization, Fall detection

1 Introduction

Fall is defined by the World Health Organization as *an event which results in a person coming to rest inadvertently on the ground or floor or other lower level* [1]. This problem affects particularly the aged population and, as stated in [2], approximately 28-35% of people aged 65 and over fall each year, increasing to 32-42% for those aged over 70 years of age. These numbers are confirmed also in EU28 and EEA countries, where approximately 100000 older people die from injury due to a fall each year [3].

The direct consequences correlated to a fall could be: superficial cuts, broken or fractured bones, and abrasions or tissue damage. Also the "*long-lie*" condition, defined as involuntarily remaining on the ground for an hour or more, following a fall, represents a serious risk for the health. As stated in [4], half of elderly people who experience a "*long-lie*" die within 6 months. Taking into account all these aspects, a reliable and secure system to monitor an elderly during his daily life is strongly recommended. It must ensure an adequate robustness against false alarms, and be unobtrusive at the same time.

S. Loshkovska and S. Koceski (eds.), *ICT Innovations 2015*,
Advances in Intelligent Systems and Computing 399,
DOI: 10.1007/978-3-319-25733-4_11

In the literature, the initially proposed solutions tried to use wearable devices to solve this task. In [4], tri-axial accelerometers are placed on the trunk and the thigh of 10 volunteers that perform ADLs (Activities of Daily Living), and simulate falls. Kangas *et al.* [5] used a tri-axial accelerometer attached to belt at the waist, involving also elderly people in the ADLs subset of a test campaign. An alternative research approach uses cameras as a source of information to detect risky activity [6].

Recently, the availability of cheap depth sensors, has enabled an improvement of the robustness in camera based approaches for fall detection solutions. In particular, the Kinect sensor, i.e. the RGB-Depth device used in this publication, has been adopted in different implementations, as presented in [7] and [8].

In the last years, thanks to the growth of computational resources, the combination of the previous solutions became possible and this led to an improvement of the performance. These solutions exploit an approach defined as “data fusion”, and examples of joint use of Kinect sensor and wearable devices are visible in [9]. The synchronization issues between Kinect and wearable devices, to the best of our knowledge, is not totally covered in the literature. In view of this fact, we use the synchronization approach described in [10] to design fall detection systems that exploit heterogeneous data provided by different sensors. It is also worth noting that we started creating a database of ADLs and falls, containing visual and acceleration data, that can be exploited to compare different solutions [11].

The remaining part of this paper is organized as follows. In Section 2 the synchronization approach is presented. Section 3 describes the proposed fall detection solution. Experimental results are discussed in Sect. 4, while Sect. 5 is dedicated to concluding remarks.

2 Synchronization

The synchronization issue between a wearable inertial device and a vision based device, namely the Microsoft Kinect sensor, has been addressed in [10]. In this work, the transmission and exposure times of the frames captured by Kinect are exploited to synchronize the RGB-D sensor with two inertial measurement units (IMU) from Shimmer Research. Figure 1 shows the devices involved in the synchronization process. An ad-hoc acquisition software allows to simultaneously capture data from Kinect, connected via USB cable, and from the accelerometers, linked via Bluetooth to the same PC, running the acquisition software. The same software applies a timestamp when each packet, or frame, arrives at the PC, using the $QueryPerformanceCounter$ and $QueryPerformanceFrequency$ C++ functions. The synchronization is realized by exploiting these timestamps, taking into account the transmission times of Kinect frames and any possible delays caused by the Bluetooth protocol. Figure 2a shows, in red, the sequence of skeleton samples provided by the Kinect sensor, while the green and blue lines represent the packets sent by the accelerometers. As visible, the number of packets received from each accelerometer, is much greater than the number of frames captured by Kinect, because the sampling rate of the Shimmer is 10 ms

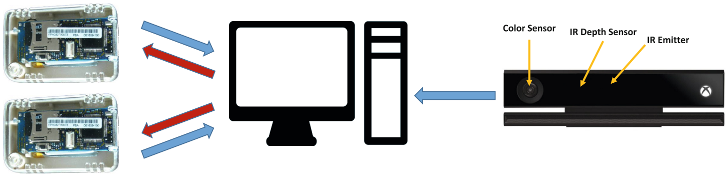


Fig. 1. The synchronization scheme involves a Microsoft Kinect v2 sensor, connected via USB cable to a PC, and two accelerometers onboard the wearable devices, linked via Bluetooth to the same PC.

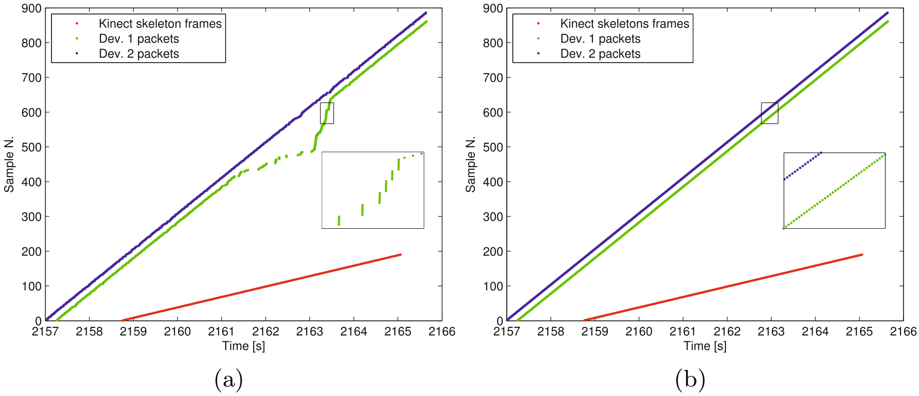
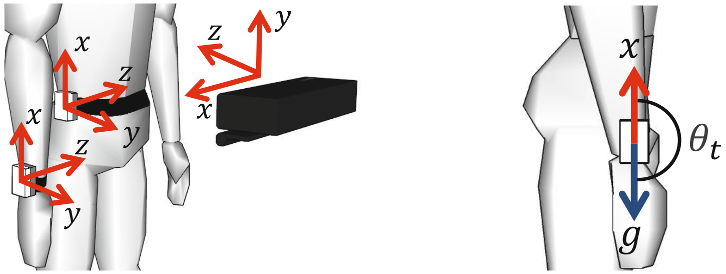


Fig. 2. (a) Raw sample time correlated to Skeleton frames and packets generated by the two accelerometers, (b) same curves after linearization.

while Kinect outputs data approximately every 33 ms. The rectangle contains a zoomed interval of the data sent by device 1. The nonlinear trend is caused by the behaviour of the Bluetooth protocol, as highlighted in [12]. Indeed, each packet arrives at the PC with a variable delay that must be corrected to enable the synchronization with the Kinect data. Using a linear regression algorithm, the wearable device curves are linearized, and the result is shown in Fig. 2b. The zoomed area shows that delays between subsequent packets have been corrected.

The aim of the synchronization process is to associate one acceleration sample to each Kinect skeleton frame. Thus, after having linearized the samples from accelerometers, the following operations have to be done:

- synchronization of skeleton and depth frames captured by Kinect, by using timestamps provided by the Microsoft SDK 2.0;
- compensation of the transmission time of the skeleton frame, which is the same as the depth frame;
- association of the closest in time acceleration sample to each skeleton frame.



(a) (b)

Fig. 3. (a) Setup of Kinect v2 and wearable sensors, (b) orientation of accelerometers angle.

3 Fall Detection

The synchronization algorithm summarized in the previous section is used to perform data fusion in a fall detection solution. It is possible to use Kinect and acceleration data to design reliable fall detection algorithms. The idea is to propose different algorithms, that can compute different parameters, and to evaluate their performances.

3.1 System Setup

The system setup includes two IMUs, mounted on the wrist and on the waist of the subject, and a Microsoft Kinect v2 sensor, placed as shown in Fig. 3a. A Shimmer device is placed to the right side of the body, constrained to the waist using a belt, since Kepski *et al.* [9] recommend to place the sensor to the trunk or lower back. Another accelerometer is placed to the right wrist, to simulate a smartwatch. The Kinect sensor monitors the test area, and it is positioned at about 1.5 meters from the floor and 2 meters from the person.

3.2 Acceleration Data Processing

The Shimmer device integrates the Freescale MMA7260QT accelerometer that provides 3-axis raw accelerations data. They are converted into gravity accelerations (X, Y, Z), taking into account possible biases. The acceleration magnitude is:

$$M_{acc} = \sqrt{X^2 + Y^2 + Z^2} \tag{1}$$

and the angle θ_t between the X axis and the g vector (Fig. 3b) is given, as defined in [13], by:

$$\theta_t = atan2\left(\sqrt{Z^2 + Y^2}, X\right) \tag{2}$$

Normally when the person is standing, with arms parallel to the body, the angle θ_t measured by both the accelerometers is equal to 180°. When the person lies down on the floor this angle should be 90°.

3.3 Algorithms 1 and 2

The first implemented solution for fall detection exploits acceleration data from the wrist IMU and skeleton information from Kinect. In particular, the following information are considered:

- variation in the skeleton joint position;
- M_{acc} of the wrist accelerometer;
- θ_t angle of the wrist accelerometer, after the extraction of the gravity acceleration component.

The first parameter, i.e. the variation of a skeleton joint position, considers the socalled $SPINE\_BASE$ joint (J_{SPB}), located at the base of the spine [14]. As visible in Fig. 3a, the y axis represents the vertical component in the reference system of the skeleton, and it can be monitored to evaluate any movements referable to a fall. In the first captured frames, the reference y value of the J_{SPB} joint is evaluated and then, if the difference between the actual value and the reference one exceeds a threshold of 50 cm, an irregular activity is detected. The second considered parameter is the magnitude of acceleration, revealed by the wrist IMU. In this case, an acceleration peak greater than 3g, as suggested by [15], has to be found within a time interval of two seconds, centered in the time instant where the irregular activity of the skeleton has been identified. The third parameter is represented by the orientation of the sensor. In order to detect a fall, the angle θ_t should be around 90° for a not negligible amount of time. In the proposed implementation, a threshold value of 90°, with a guard interval of 20°, for at least 0.5 s, has been considered. If all the parameters satisfy the chosen conditions, the action is classified as a fall.

The Algorithm 2 computes the same parameters as Algorithm 1, but it exploits data from the accelerometer placed on the waist of the subject.

3.4 Algorithm 3

The third implemented solution avoids using the orientation of the accelerometers, and exploits the following parameters:

- variation in the skeleton joint position;
- distance of the J_{SPB} joint from the floor;
- M_{acc} of the waist accelerometer.

The parameter that indicates an irregular activity is the remarkable change in the y component of the J_{SPB} joint. Then, the distance of that joint from the floor is also evaluated. The floor is modeled as a plane, which is automatically detected from the first available skeleton frame. Given the plane equation:

$$ax + by + cz + d = 0 \tag{3}$$

the constant term d is computed using the following equation:

$$d = -(ax_0 + by_0 + cz_0) \tag{4}$$

where $v_n = [a, b, c]$ is the orthogonal vector to the plane, and $P_0 = [x_0, y_0, z_0]$ is one point in the plane. In the proposed approach, the vector v_n is evaluated as the vector that models the spine, assuming that the subject is standing at the beginning of the test, that is when the plane is identified. Considering two vectors that identify two joints of the spine, namely the J_{SPB} and the $SPINE\_MID$ joint (J_{SPM}), the following equation is used to find the normal to the floor vector:

$$v_n = \frac{J_{SPM} - J_{SPB}}{||J_{SPM} - J_{SPB}||} \tag{5}$$

while the point belonging to the plane is one of the ankle joints of the subject. The distance $dist_{SPB}$ between the SPB joint and the floor is evaluated using the following equation:

$$dist_{SPB} = \frac{|v_n \cdot J_{SPB} + d|}{||v_n||} \tag{6}$$

When the distance $dist_{SPB}$ decreases below a threshold value (20 cm), the algorithm evaluates the time instant and selects a time window of 2 seconds in the M_{acc} trajectory, looking for an acceleration peak greater than 3g. If also this peak is found, the action is classified as a fall.

4 Results and Discussion

The designed algorithms have been tested in a laboratory environment, on a database composed by 11 healthy volunteers. The people involved in the test are aged between 22 and 39, with different height (1.62-1.97 m) and build. The actions that populate the dataset are separated in two main groups: ADLs and Fall. Each activity is repeated three times by each subject involved. The whole database, containing 264 different actions for a total number of 46k skeleton and 230k acceleration values, is available at [11]. The proposed algorithms are implemented in MATLAB and they have a very low complexity. The time required to process a sequence of skeleton and acceleration data goes from 2 to 6 ms, having sequences with durations in the interval 2.5-15 s. The detailed experimental protocol is provided in Table 1. Results are evaluated, as defined in [5], in terms of sensitivity, specificity and accuracy.

Table 2 shows the accuracy evaluated over the entire dataset, for the three considered algorithms. Algorithm 1 is the less invasive one because it simply relies on the accelerometer placed on the wrists and, despite it shows a specificity of 98%, it is characterized by a sensitivity of 59%, which means that a quite poor set of falls are correctly detected. Looking at Table 2, it can be noticed that the most difficult fall to detect is the *side* one, featured by an accuracy of 48% while the highest accuracy (82%) is reached by the backward fall that ends up lying, labelled as *back*. The weakness of Algorithm 1 is represented by the orientation of the accelerometer. In fact, even if the person is fallen and he/she is lying, the arm could be not parallel to the floor, thus avoiding the detection of the

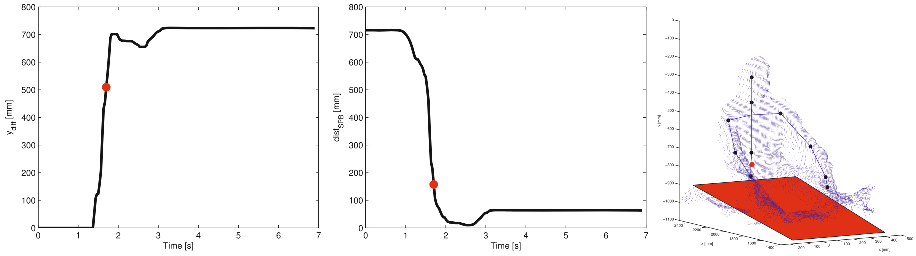
Table 1. Experimental protocol

Category	Activity	Description
ADL	*sit*	The subject sits on a chair
	grasp	The subject walks and grasps an object from the floor
	walk	The subject walks back and forth
	lay	The subject lies down on the mattress
Fall	*front*	The subject falls from the front and ends up lying
	back	The subject falls backward and ends up lying
	side	The subject falls to the side and ends up lying
	EUpSit	The subject falls backward and ends up sitting

fall. In order to have better performance, it is necessary to use the accelerometer placed on the waist, which provides a more reliable information about the orientation of the subject's body. The sensitivity and specificity of Algorithm 2 reach respectively the percentage of 79% and 100%; looking more in detail at Table 2, it gives an accuracy of 100% for each test, except the *EUpSit* fall test. In this specific case, the orientation of the accelerometer does not give values below the chosen threshold, because the torso remains perpendicular to the floor in almost all the tests. The correct detection of the *EUpSit* fall is attained using Algorithm 3 (specificity 99%, sensitivity 100% and accuracy 99%), that exploits the distance from the floor of the J_{SPB} joint instead of the accelerometers orientation. The variation of the y axis during an *EUpSit* fall is shown in Fig. 4a,

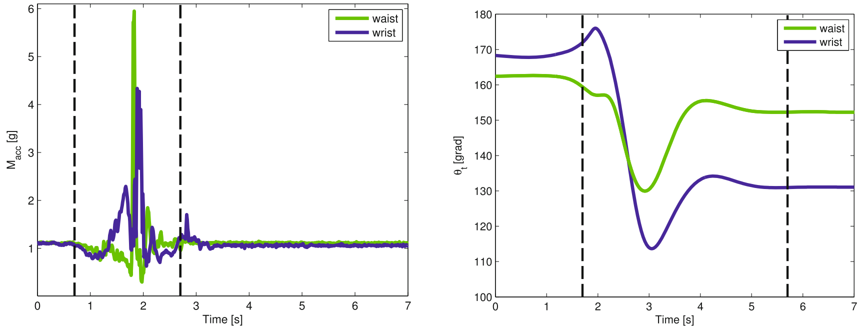
Table 2. Accuracy of the three fall detection algorithms for each activity

Category	Activity	Accuracy		
		Algorithm 1	Algorithm 2	Algorithm 3
ADL	*sit*	97%	100%	100%
	grasp	100%	100%	100%
	walk	100%	100%	100%
	lay	97%	100%	100%
Fall	*front*	54%	100%	97%
	back	82%	100%	100%
	side	48%	100%	100%
	EUpSit	52%	18%	100%
Average		79%	90%	99%



(a) (b) (c)

Fig. 4. (a) Variation of the y component of J_{SPB}, (b) its distance from the floor during an *EUpSit* fall, and (c) person's point cloud with highlighted in red J_{SPB} and detected plane.



(a) (b)

Fig. 5. (a) Acceleration and (b) orientation revealed during an *EUpSit* fall.

and the threshold that indicates an irregular activity is reached after about 1.8 seconds from the beginning of the action. The second parameter that Algorithm 3 checks is the distance between the floor and the J_{SPB} joint, revealing a value below the threshold of 20 cm, as shown in Fig. 4b. Figure 4c shows the subject's point cloud in the final phase of *EUpSit* fall, the red plane is used as a reference element to model the ground. The skeleton is superimposed to the person and J_{SPB} is highlighted by a red circle. Finally, the algorithm selects a time window of 2 seconds and searches an acceleration peak greater than 3 g in the waist accelerometer data, as depicted in Fig. 5a. Algorithm 1 and 2 fail to detect this fall because they consider accelerometer orientations that do not reveal an angle lower than the threshold of 110°, as can be noticed from Fig. 5b.

The most used features in fall detection solutions are extracted from accelerometers, gyroscope or pressure sensors, and include magnitude vectors and tilt angles [13]. Considering only these features, it is quite difficult to detect falls where the subject ends up sitting. In fact, only a few previously published works include this type of fall in the evaluated dataset. Kangas *et al.* [5] reach a *sensitivity* of 97.5% and a *specificity* of 100% considering a dataset of 6 falls and 4 ADLs. Despite the large number of performed tests, in all the considered falls the subjects end up lying, limiting the application scenarios. Pierleoni *et al.* [16], in their fall dataset, consider syncope and backward falls ending up sitting. The authors state very high performance in terms of *sensitivity* and *specificity*, with an average *accuracy* of 100%. Most of the approaches based on wearable devices try to detect the acceleration peak and evaluate the orientation of the device, to estimate the posture of the subject when he/she is on the ground. Thus, the wearable sensor must be positioned on the subject's body giving a special attention to its orientation. The solution proposed herein exploits a vision-based device and a wearable device, combining heterogeneous information by a data fusion approach. The orientation of the subject is not evaluated using the wearable device in Algorithm 3, but exploiting the information provided by the camera, which allows to identify the subject on the floor even if he/she is sitting. A drawback of the proposed solution is due to the fact that it is based on the skeleton data automatically extracted from Kinect Microsoft SDK from raw depth data. The joints estimation algorithm has been developed for gaming purposes, when the subject stands is in front of the sensor. Algorithm 3, tested on a database of 11 people performing the proposed experimental protocol, fails only in one *front* fall, as can be noticed from Table 2. In that case, Microsoft SDK is unable to estimate the skeleton when the subject is falling, and the fall is classified as an ADL because the conditions on the skeleton joints are not satisfied. This issue may be solved by integrating a barometric pressure sensor in the wearable device and using that data to evaluate the height of the waist from the floor.

5 Conclusion

This work proposed fall detection solutions exploiting skeleton data computed using the Microsoft Kinect sensor, joint acceleration data. By means of an ad-hoc synchronization algorithm, vision-based data and inertial data can be associated and used to design a simple and reliable fall detection solution. The wearable accelerometer device makes it easy to distinguish a fall from a "lying on the floor" condition because of the different acceleration vector magnitude, while the Kinect sensor is able to estimate the body orientation and the distance from the floor, enabling to identify a fall where the subject ends up sitting.

Future works will concern enriching the dataset, involving more people in the tests and considering different ADLs and falls, in order to allow a more extensive testing of the proposed algorithms.

Acknowledgments. This work was supported by STSM Grants from COST Action IC1303 *Algorithms, Architectures and Platforms for Enhanced Living Environments* (AAPELE).

References

1. World Health Organization, `http://www.who.int/violence_injury_prevention/other_injury/falls/en/`
2. World Health Organization. Ageing and Life Course Unit: WHO global report on falls prevention in older age. World Health Organization, France (2008)
3. Safety for Seniors, `www.eurosafe.eu.com/csi/eurosafe2006.nsf/wwwVwContent/l2safetyforseniors-seniornew.htm`
4. Bourke, A.K., O'Brien, J.V., Lyons G.M.: Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. Gait Posture 26, 194-199 (2007)
5. Kangas, M., Vikman, I., Wiklander, J., Lindgren, P., Nyberg, L., Jämsä, T.: Sensitivity and specificity of fall detection in people aged 40 years and over. Gait Posture 29, 571-574 (2009)
6. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Robust Video Surveillance for Fall Detection Based on Human Shape Deformation. IEEE Trans. Circuits Syst. Video Technol. 21, 611-622 (2011)
7. Gasparrini, S., Cippitelli, E., Spinsante, S., Gambi, E.: A Depth-Based Fall Detection System Using a Kinect Sensor. Sensors 14, 2756-2775 (2014)
8. Marzahl, C., Penndorf, P., Bruder, I., Staemmler, M.: Unobtrusive Fall Detection Using 3D Images of a Gaming Console: Concept and First Results. In: Wichert, R., Eberhardt, B. Ambient Assisted Living. LNCS, pp. 135-146. Springer, Heidelberg (2012)
9. Kepski, M., Bogdan, K., Austvoll, I.: Fuzzy inference-based reliable fall detection using kinect and accelerometer. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada J.M., Artificial Intelligence and Soft Computing. LNCS, vol. 7267, pp. 266-273. Springer, Heidelberg (2012)
10. Cippitelli, E., Gasparrini, S., Gambi, E., Spinsante, S., Wåhslén, J., Orhan, I., Lindh, T.: Time Synchronization and Data Fusion for RGB-Depth Cameras and Inertial Sensors in AAL Applications. To appear in: Proceedings of IEEE ICC 2015 - Workshop on ICT-enabled services and technologies for eHealth and Ambient Assisted Living, 8th June, London (2015)
11. TST Fall Detection Database, `http://www.tlc.dii.univpm.it/blog/databases4kinect`
12. Wåhslén, J., Orhan, I., Lindh, T., Eriksson, M.: A Novel Approach to Multi-Sensor Data Synchronisation Using Mobile Phones. J. Auton. Adapt. Commun. Syst. 6, 289-303 (2013)
13. Pannurat, N., Thiemjarus, S., Nantajeewarawat, E.: Automatic Fall Monitoring: A Review. Sensors 14,12900-12936 (2014)
14. JointType Enumeration, `https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx`
15. Kangas, M., Konttila, A., Lindgren, P., Winblad, I., Jämsä, T.: Comparison of low-complexity fall detection algorithms for body attached accelerometers. Gait Posture 28, 285-291 (2008)
16. Pierleoni, P., Belli, A., Palma, L., Pellegrini, M., Pernini, L., Valenti, S.: A High Reliability Wearable Device for Elderly Fall Detection. IEEE Sensors J. 15, 4544-4553 (2015)