Expert System for Wearable Fall Detector

Gabriele Rescio, Alessandro Leone and Pietro Siciliano

Abstract Falling down can cause moderate to severe injuries, increasing the risk of death among elderly. For this reason there is a substantial growth of Ambient Assisted Living technologies, including smart environments, in order to support elderly and fragile people in potentially dangerous situations. The paper describes an expert system based on a wireless wearable low-cost accelerometer able to automatically detect falls, generalizing the detection of critical events in several practical conditions. The algorithmic scheme appears invariant to age, weight, height of people and relative positioning area (even in the upper part of the waist), resulting compliant with many commercial wearable devices. Experimental results show high generalization properties and better performances than well-known threshold-based approaches.

1 Introduction

The main reason for the development of the presented system is to allow partially self-sufficient people to live safely in their own homes as long as possible. The problem of falls in the elderly has become a health care priority due to the related high social and economic costs [9]. Many solutions have been proposed in detection and prevention of falls and some excellent reviews have been presented [9, 10, 13]. The availability of miniaturized low-cost MEMS accelerometers on the market and the new reliable wireless communication technologies allow the realization of affordable wearable systems useful for daily activities monitoring [4]. However, this kind of technology

G. Rescio (✉) · A. Leone · P. Siciliano
CNR – Institute for Microelectronics and Microsystems, Via Monteroni, 73100 Lecce, Italy
e-mail: gabriele.rescio@le.imm.cnr.it

A. Leone
e-mail: alessandro.leone@le.imm.cnr.it

P. Siciliano
e-mail: pietro.siciliano@le.imm.cnr.it

S. Longhi et al. (eds.), *Ambient Assisted Living*, DOI: 10.1007/978-3-319-01119-6_10,
© Springer International Publishing Switzerland 2014

presents some drawbacks: they are prone to be forgotten, worn in a wrong body position or accidentally damaged. Regarding fall detection, an accelerometer-based solution presents some advantages respect to vision or acoustic sensors: just one sensor needs to be used, re-design of the environments is not required and ethical issues (such as privacy) are always satisfied. On the other hand, camera-based or acoustic-based approaches are non-invasive [6] since contactless solutions. In this paper a fall detector through a wearable tri-axial MEMS accelerometer is presented. The proposed solution overcomes the limitation of well-known threshold-based approaches [1] in which several parameters need to be manually estimated according to the specific features of the end-user. In particular, a machine learning scheme has been used and high generalization capabilities in the fall detection discrimination process have been recovered. The expert system uses robust features extracted taking into account important constraints and/or requirements of mobile solutions (workload). The extracted features are (quasi-) invariant both to specific characteristics of the mounting setup (device on chest, on waist, on abdomen) and specific characteristics of the end users in terms of age, weight, height and gender.

2 Framework Overview and Materials

The main computational steps of the software architecture are data acquisition, noise filtering, data pre-processing, system calibration, feature extraction and classification. Each step is described in the following sections. The capability of the system to detects a fall is evaluated by using a state-of-the art supervised classifier, according to the procedure described in [5]. The algorithmic framework has been developed by using the wearable device [12] composed by commercial discrete circuits.

The system integrates a ST LIS3LV02DL tri-axial MEMS accelerometer with digital output, an FPGA for computing functionalities and a Zigbee module for wireless communication up to 30 m, suitable for indoor contexts. The power consumption is about 190 mW in streaming mode and 9 mW in idle. The wearable device can operate in streaming mode (raw data are sent via ZigBee to an external computing platform for data analysis with a 10 Hz frequency) or in standalone mode. In the last release of the device a threshold-based fall detector has been integrated on the on-board FPGA and the implementation has been used as comparison item for the evaluation of the proposed scheme. Raw data are in hexadecimal 16-bit and represent the acceleration values with full scale in the range ±2 g for higher sensitivity. The accelerometer is DC coupled (it responds up 0 Hz) and it measures both static and dynamic acceleration along the 3 axes, describing the 3D spatial relative position of the person who wears it. Generally speaking, the accelerometer measures the projection of the gravity vector on each sensing axis. Assuming a particular axis, the component of the acceleration (amplitude A, Eq. 1) is defined according to the value of the sin of the angle α between the considered axis and the horizontal plane, which is perpendicular to the Earth's gravity component (g):

$$A = g\sin(\alpha) \tag{1}$$

In this way, if the accelerometer relative orientation is known, the resulting data can be used to determinate angle of the user posture respect to the vertical direction.

3 Self-Calibration System

The acceleration data on three axes (A_x, A_y, A_z) were read out from the device worn by a user during the data collection. Data were stored and converted from hexadecimal to decimal format to be compliant the implemented software. A normalization step has been used in order to represent acceleration data in the range ± 2 g. The samples coming from the device are filtered out by a low pass 8-order, 8 Hz cut-off FIR (Finite Impulse Response) filter to reduce the noise due to electronic components, environment and human tremor. In order to handle correctly pre-processed data, a calibration procedure has been accomplished by recovering the initial conditions after the device mounting. During the step, the correct placing of hardware is verified by checking if two acceleration axes are orthogonal to Earth's gravity g (Fig. 1): the acceleration values measured on the two orthogonal components must be close to zero. The calibration procedure is achieved by the following steps:

1. The user wears the device in a still standing position for 10 s, as shown in Fig. 1.
2. The calibration routine calculates the average of the acceleration (A_{x0}, A_{y0}, A_{z0}) on each axis over this period.
3. If (A_{x0}, A_{y0}, A_{z0}) are close to those expected (-1, 0, 0 according to Eq. 1), the calibration routine finished and (A_{x0}, A_{y0}, A_{z0}) is the reference in the features extraction.

The calibration procedure is concluded if the initial measured values do not differ more than 30 % from the expected ones and the values A_{x0}, A_{y0} and A_{z0} will be recorded and used as references in the feature extraction phase. Otherwise, a routine to compensate the sensed misplacement is enabled and the angles displacements of the sensor axes (αA_{x0}, αA_{y0} and αA_{z0}) are calculated using the following trigonometric Eq. 2:

$$\alpha A_{x0} = arctan\left(\frac{A_{x0}}{\sqrt{\left(A_{y0}\right)^2 + (A_{z0})^2}}\right), \quad \alpha A_{x0} = arctan\left(\frac{A_{y0}}{\sqrt{(A_{x0})^2 + (A_{z0})^2}}\right),$$
$$\alpha A_{x0} = arctan\left(\frac{A_{z0}}{\sqrt{(A_{x0})^2 + \left(A_{y0}\right)^2}}\right) \tag{2}$$

These values are stored and will be used for the correction of the misplacement during the feature extraction phase.

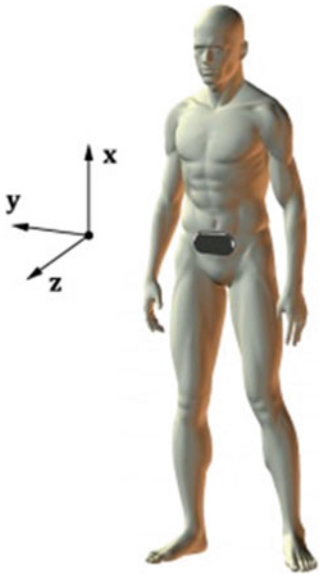

Fig. 1 Mounting position and calibration

4 Robust Feature Extraction

Robust features are extracted in the time domain by considering both quick and relevant acceleration changing along each axis (due to the fall) in a 5 s sliding window and the posture changing registered after the shock. The aim is to produce robust features, taking all the information able to discriminate falls from other events. It is also important that such features have a low dependence on both the position of the sensor (whether it is placed on the waist, on the chest or on the abdomen) and the human body characteristics of the user. For the features extraction process, both critical and post fall phases are of interest and corresponding two kind of features are extracted. In the former, the shock is measured due to the impact toward the floor plane and a dynamic acceleration changing is registered. In the latter (the body is already lying) the static acceleration value records a great change due to the new position of the individual with respect to the calibration phase. The posture changing with respect to the initial condition (A_{x0}, A_{y0}, A_{z0}) can be realized through the static information of acceleration which varies with the inclination of the accelerometer sensing axis. Hence the difference between the value of the 3D-static acceleration after the fall and the one stored in the calibration phase will result in an offset, called Changing Position Offset (CPO), which is proportional to the user displacement. In this way, a study of posture was not made: only the relative varying posture analysis was considered, causing a computational cost reduction and mounting setup invariance.

If the device is not worn correctly, the routine to compensate the device misplacement is enabled (see Self-calibration system section). In this case, for the CPO

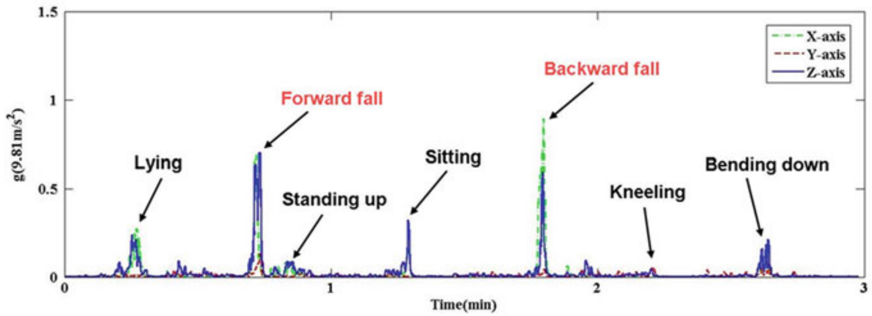

Fig. 2 Extracted features for simulated falls and ADLs (the device is on the waist)

calculation, the axes angles of the sensor in the initial condition ($\alpha \mathrm{A}_{x0}$, $\alpha \mathrm{A}_{y0}$ and $\alpha \mathrm{A}_{z0}$) need to be taken into account. The CPO value for x-axis is obtained as:

$$CPO = \left|\sin\left(\alpha\overline{A_x} - \alpha A_{x0}\right)\right|, \tag{3}$$

where $\alpha\bar{\mathrm{A}}_x$ is the X-axis angle of the sensor during the post-fall phase; it is calculated considering the static averaged acceleration $\bar{\mathrm{A}}$ and the same trigonometric equation used in (2):

$$\alpha\overline{A_x} = arctan\left(\frac{\overline{A_x}}{\sqrt{\left(\overline{A_y}\right)^2 + \left(\overline{A_z}\right)^2}}\right), \tag{4}$$

The same procedure must be done for the two other axes. Thanks to this routine the problems of device positioning are strongly reduced at the expense of an increase in the computational costs.

The feature vector is made up by three parameters, one for each acceleration component. It makes sense to consider the acceleration signal on each axis singularly, because a fall event leads to a change in the value of the static acceleration in at least two of the three acceleration axes (due to the orientation change of sensing axes). In Figs. 2 and 3 the features of a sequence of falls and daily events, when the device is worn on the waist and on the chest, are shown. So it is evident that the features obtained discriminate the falls from typical Activities of Daily Living (ADLs) also when the device is placed in other area of the torso.

5 Supervised Classification and Experimental Results

Once features are extracted, the fall events are detected by a One Class Support Vector Machine (OC-SVM) which is less computationally intensive than other algorithms like neural networks [7]. SVM is a robust classification tool (in presence of outliers

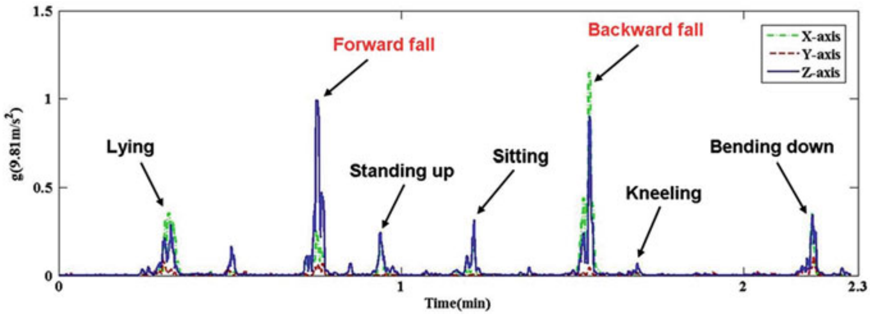

Fig. 3 Extracted features for simulated falls and ADLs (the device is on the chest)

too) with a good generalization ability. OC-SVM divides all samples into objective field and non-objective field and then non linearly maps those sample into high dimensional features space with some efficient functions, called kernel function. The commonly types of kernel found in literature have been tested. They were analyzed using Matlab tool STPRtool [2], changing the values of key parameters as described in [11]. For the extracted features, the optimal kernel are Gaussian Radial Basis Function (GRBF) [14] and polynomial, as better detailed in the following.

The OC-SVM classifier has been trained by using about 40 falls and 50 ADLs belonging to a large dataset in which more than 250 falls and 200 ADLs were performed compliant the specifics proposed in [8]. The remaining 210 falls and 150 ADLs have been used for testing. To validate the implemented algorithmic framework, the previously described dataset has been considered. The performances of the system have been evaluated considering two normally used metrics, sensitivity and specificity [8], respectively. The algorithm is tested when the device is placed on waist, abdomen or chest. GBRF and polynomial kernel functions give the best results in terms of performance, even the polynomial kernel shows a lower computational cost (relative number of support vectors and relative execution time are considered). In particular, GRBF (with $\sigma = 3$ and $C = 2$) and the Polynomial (with $P = 3$ and $C = 2.8$) provide similar values of specificity and sensitivity but the last one works faster and its number of vectors is slightly lower. Misclassifications are for falls presenting slow dynamics or falls with partial recovery. The implemented SVM improves in the specificity and sensitivity respect to the threshold-based approaches detailed in [1, 3], as reported in Table 1. As already discuss, the two fall detection systems have in common hardware, benchmark dataset and training/test sets. Of course, the computational cost of threshold-based is lower than the implemented expert systems, but it suffers in detection rate due to high false positive and true negative. Since the number of features is compact and the computational cost of the extracted features is low, the overall system workload is compatible with an integration in embedded low-power solutions (DSP, FPGA, microcontroller).

Table 1 Comparison of the proposed OC-SVM method and threshold-based algorithm detailed in [3]

	Sensitivity (%)	Specificity (%)	Relative execution time
OC-SVM (Polynomial $C = 2.8$, $P = 3$)	97.7	94.8	1x
OC-SVM (GRBF $C = 32$, $\gamma = 3$)	97.4	95.2	1.5x
Threshold-based	89.5	85.7	0.3x

6 Conclusions

The proposed supervised scheme overcomes the limitation of well-known threshold-based approaches in which a heuristic choice of the parameters is accomplished. A specific study on postures was not made in order to make a low computational power system. A new inferred information has been computed (Changing Position Offset) in order to intrinsically acquire relevant posture changing with possible affection on fall events. The calibration step guarantees the generalization of the approach in terms of invariance to physics characteristic of the end-users, during the fall detection process. High performances in controlled conditions (simulated events) in terms of sensitivity and specificity were obtained using only the 20 % of dataset for training purposes. Performance metrics of different kernels in One Class SVM have been compared and the best results are obtained with polynomial function and Gaussian Radial Basis Function, even the polynomial kernel presents a limited workload. Future works are addressed both to validate the solution in real conditions, to test the methodology with a large set of different MEMS accelerometers and to port the framework on embedded mobile solutions.

References

1. Bagalà, F., Becker, C., Cappello, A., Chiari, L., Aminian, K., Hausdorff, J.M., Zijlstra, W., Klenk, J.: Evaluation of accelerometer-based fall detection algorithms on real-world falls. PLoS One **7**, e37062 (2012)
2. Franc, J., Hlavac, V.: Statistical Pattern Recognition Toolbox for Matlab. http://cmp.felk.cvut.cz/cmp/software/stprtool/ (2004)
3. Grassi, M., Lombardi, A., Rescio, G., Malcovati, P., Malfatti, M., Gonzo, L., Leone, A., Diraco, G., Distante, C., Siciliano, P., Libal, V., Huang, J., Potamianos, G.: A hardware-software framework for high-reliability people fall detection. In: Proceeings of IEEE Sensors 2008, pp. 1328–1331 (2008)
4. Karantonis, D., Narayanan, M.R., Mathie, M., Lovell, N.H., Celler, B.G.: Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE Trans. Inf. Technol. Biomed. **10**(1), 156–167 (2006)
5. Liu, S.H., Cheng, W.C.: Fall detection with the support vector machine during scripted and continuous unscripted activities. Sensor **12**, 12301–12316 (2012)
6. Leone, A., Dirago, G., Siciliano, P.: Detecting falls with 3D range camera in ambient assisted living applications: a preliminary study. Med. Eng. Phys. **33**, 770–781 (2011)

7. Manevitz, L.M., Yousef, M.: One-class SVMs for document classification. J. Mach. Learn. Res. **2**(1), 139–154 (2001)
8. Noury, N., Fleury, A., Rumeau, P., Bourke, A.K., Laighin, G.O., Rialle, V., Lundy, J.E.: Fall detection–principles and methods. In: Proceedings of the 29th IEEE EMBS, pp. 1663–1666 (2007)
9. Noury, N., Rumeau, P., Bourke, A.K., Laighin, G.O., Lundy, J.E.: A proposal for the classification and evaluation of fall detectors. IRBM **29**(6), 340–349 (2008)
10. Sadigh, S., Reimers, A., Andersson, R., Laflamme, L.: Falls and fall-related injuries among the elderly: a survey of residential-care facilities in a Swedish municipality. J. Community Health **29**, 129–140 (2004)
11. Sangeetha, R., Kalpana, B.: Performance evaluation of kernels in multiclass support vector machines. Int. J. Soft Comput. Eng. **1**(5), 2231–2307 (2011)
12. Siciliano, P., Leone, A., Diraco, G., Distante, C., Malfatti, M., Gonzo, L., Grassi, M., Lombardi, A., Rescio, G., Malcovati, P.: A networked multisensor system for ambient assisted living application. In: 3rd International, Workshop on Advances in Sensors and Interfaces, IWASI 2009, pp. 139–143 (2009)
13. Yu, X.: Approaches and principles of fall detection for elderly and patient. In: Proceedings of the 10th IEEE HealthCom, pp. 42–47 (2008)
14. Zhang, T., Wang, J., Xu, L., Liu, P.: Fall Detection by Wearable Sensor and One-Class SVM Algorithm. Lecture Notes in Control and Information Science, vol. 345, pp. 858–863 (2006)