Biomed Tech 2013; 58 (Suppl. 1) © 2013 by Walter de Gruyter · Berlin · Boston. DOI 10.1515/bmt-2013-4196

FALL DETECTOR WITH RANDOM CLASSIFIER OPTIMIZATION

M. Mend[1], W.H. Kullmann[1]

[1]Institute of Medical Engineering, University of Applied Sciences Würzburg-Schweinfurt, Germany

Matthias.Mend@fhws.de

Abstract: *Fall accidents are one of the main causes of injuries and accidental deaths for elder people. Automated fall detectors can be helpful for detecting the accident and sending an emergency call. For this purpose, a wearable fall detector device was developed, which served to collect a training set and to test certain classifiers. The current approach uses a random search algorithm to optimize a given model for discrimination of fall events from activities of daily living by measuring tri-axial accelerometer data. In the current state the device is able to recognize significant falls and can trigger an emergency call over mobile networks.*

Keywords: *Fall classification, random search, tri-axial accelerometer, fall detector*

Introduction

Fall accidents of elder people often lead to complicated fractures and are reasons for death [1]. The resulting economic effects on modern health systems are enormous. Automated fall detection systems may allow a quick detection of fall accidents and enable sending of an emergency call. Despite the presence of acceleration measurement based fall detectors with high sensitivities and specificities [2], there are still challenges to increase the performance and the robustness of those systems [3].
The current approach uses a random search to optimize a given linear classifier for discrimination of fall events from activities of daily living (ADL).

Methods

For data acquisition and proof-of-concept purposes, a wearable measurement device was developed. It consists of an evaluation board for 32-bit Atmel microcontrollers (UC3-A3 Xplained), a nine-degree-of-freedom inertial sensing platform (ATAVRSBIN2), a Bluetooth module, a GSM module, and a 450 mAh accu. The 12 x 7.5 x 3 cm^3 box is side-mounted on a belt. In the recording mode the device measures tri-axial accelerometer data with a sampling frequency of 100 Hz and transmits them to a computer wirelessly. In the evaluation mode fall detection algorithms can be tested, where the device is able to send emergency calls over a mobile phone network.
For developing the classification software of the fall detector a versatile simulation environment was created. The software contains additional tools for the examination of human computer interface architectures and biosignal processing algorithms. The program is written in the Microsoft Visual C# .NET 4 environment.

The simulator provides a block for random model generation and block parameter mutation. On one hand the experimenter can define static blocks, on the other hand a superior algorithm is able to modify the content. A screenshot of the model editor is shown in Fig. 1.

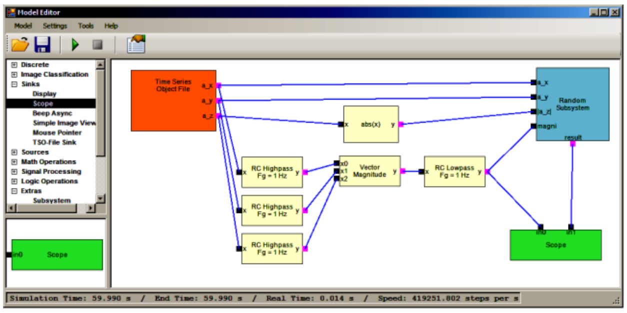

Figure 1: Model editor for the selection of static blocks, parameter adjustment and signal visualization

The software is used to find a classification rule, which should be able to detect the fall sequences within a training set of imitated falls and ADL epochs. 5 healthy male subjects with a maximum age of 31 years participated on several recording sessions. They were instructed to perform ADL movements and to fall on a mat, triggered by an acoustic signal. Each of the actions lasted 10 s with adequate recovery time between the trials. The movement types are specified in the result section. In total, a training set of 397 trials is available.
Pre-defined signal features are the components a_x, a_y and a_z of the acceleration vector and its band-pass filtered magnitude m. For symmetry reasons the absolute value of the lateral component (a_z) was used. The task for the optimization algorithm was to find a weight vector $\vec{w}$, the bias parameter b, and an array of delay times d_1 to d_8 for the linear classifier

$$y(s) = \vec{w} \cdot \begin{pmatrix} a_x(s-d_1) \\ a_x(s-d_2) \\ a_y(s-d_3) \\ a_y(s-d_4) \\ |a_z(s-d_5)| \\ |a_z(s-d_6)| \\ m(s-d_7) \\ m(s-d_8) \end{pmatrix} + b \quad , \qquad (1)$$

where the sign of the classification result y decides on the class affiliation (fall or ADL). For each sampling time, which is denoted by s, a classification output can be calculated during the simulation. A linear rule is used to avoid overfitting. Each acceleration component exists twice in order to consider signal changes in the time domain. The fitness value of a single rule is equal to the percentage of correct classified trials. If a fall has been detected, the algorithm continues with the next trial immediately. The training loop aborts, if the progress remains static. Subsequently the bias parameter is tuned manually in order to balance between the false positive and false negative classified trials. Fig. 2 visualizes the basic algorithmic principle.

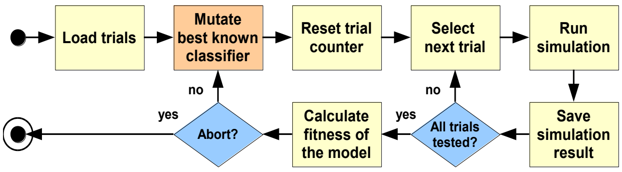

Figure 2: Block diagram for algorithm's basic principle

Results

After a couple of thousands of iterations, the sensitivity settled to a value of 93.2 %, and the specificity was 97.0 %. Tab. 1 and Tab. 2 contain the types of performed ADL movements and fall scenarios combined with the number of correct and incorrect classified trials after training.

Table 1: Activities of daily living with the number of true negative (TN) and false positive (FP) classified trials

Activities of daily living	TN	FP
Usual walking and jogging	56	0
Sitting down and standing up from chairs	28	0
Descending from a chair (to simulate stairs)	30	0
Lying down (different end states)	22	2
Turning around (in lying state)	26	4
Cycling (on an exercise bike)	10	0
Doing push-ups	14	2
Jumping (in different directions)	30	0
Doing squats and other further ADL tasks	41	0

Table 2: Fall scenarios with the number of true positive (TP) and false negative (FN) classified trials

Fall scenarios	TP	FN
Falling (in different directions)	42	0
Falling on the knees (different end states)	18	0
Falling from a chair (in different directions)	12	0
Vertical slipping (different end states)	43	2
Falling backward (sitting end state)	8	1
Slowly vertical slipping (leaning at a wall)	0	6

For checking the generalizability, firstly the classifier is applied on the samples between the trials. The expected outcome, to detect no falls, is confirmed for the most epochs. Furthermore, the discrimination rule is installed on the detector device for real time evaluation. Fig. 3 shows the classification result as a function of recording time. The output value exceeds the zero line, if a fall event occurs. It can be summarized, that the accuracy is comparable to the outcomes of the training, overall.

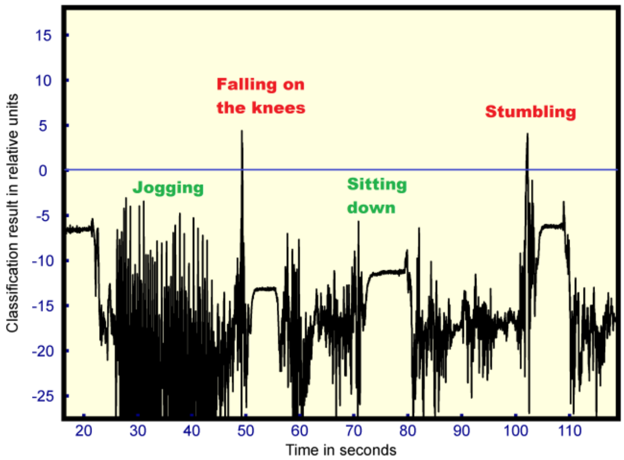

Figure 3: Classification result while performing two ADL tasks and simulating two fall scenarios

Discussion

Performing random searches for optimization is principally feasible. Fall events can be separated from the activities of daily living with high accuracy. However, scenarios, where a person faints slowly, cannot be identified by the detector. The measurement of physiologic parameters and additional features, for example the barometric pressure, may be necessary to cover falls where the accelerations keep on lower values [4].

It is difficult to indicate performance measures and to compare them with other approaches, because of the non-homogeneously composited data set. But so far, there is neither a common definition of a fall nor a standardized evaluation protocol [3].

In current state the fall detector device is able to recognize significant falls, which can trigger an emergency call over mobile networks. Long term evaluations are objectives of future work.

Acknowledgement

Financial support from the Bavarian State Ministry of Sciences, Research and the Arts, Munich, is gratefully acknowledged.

Bibliography

[1] N. Noury, A. Fleury et. al.: Fall detection – Principles and Methods, Proc. 29th Ann. Int. Conf. IEEE EMBS, pp. 1663-1666, Lyon, France, 2007.

[2] G. Wu, S. Xue: Portable Preimpact Fall Detector With Inertial Sensors, IEEE Trans. Neur. Sys. Rehab. Eng., vol. 16, no. 2, pp. 178-183, 2008.

[3] M. Mubashir, L. Shao et. al.: A survey on fall detection: Principles and approaches, Neurocomputing, vol. 100, pp. 144-152, 2013.

[4] F. Bianchi, S.J. Redmond et. al.: Barometric Pressure and Triaxial Accelerometry-Based Falls Event Detection, IEEE Trans. Neur. Sys. Rehab. Eng., vol. 18, no. 6, pp. 619-627, 2010.