Fall Detection Algorithm Based on Thresholds and Residual Events

Fily M. Grisales-Franco[1(✉)], Francisco Vargas[2], Álvaro Ángel Orozco[3], Mauricio A. Alvarez[3], and German Castellanos-Dominguez[1]

[1] Signal Processing and Recognition Group, Universidad Nacional de Colombia, Manizales, Colombia
{fmgrisalesl,cgcastellanosd}@unal.edu.co
[2] SISTEMIC, Faculty of Engineering, Universidad de Antioquia UdeA, Medellín, Colombia
jesus.vargas@udea.edu.co
[3] Grupo de Investigacion En Automatica, Universidad Tecnologica de Pereira, Pereira, Colombia
{aaog,malvarez}@utp.edu.co

Abstract. Falling is a risk factor of vital importance in elderly adults, hence, the ability to detect falls automatically is necessary to minimize the risk of injury. In this work, we develop a fall detection algorithm based in inertial sensors due its scope of activity, portability, and low cost. This algorithm detects the fall across thresholds and residual events after that occurs, for this it filters the acceleration data through three filtering methodologies and by means of the amount of acceleration difference falls from Activities of Daily Living (ADLs). The algorithm is tested in a human activity and fall dataset, showing improves respect to performance compared with algorithms detailed in the literature.

1 Introduction

The elderly people are facing risk factors as are the falls that can lead to suffer minor, serious and even fatal injuries. It is estimated that each year about one-third of adults over 65 years old suffer falls, and the likelihood of falling increases substantially with advancing age [9]. When a fall occurs, it is possible to minimize the risk of injury depending largely on the response, rescue and timely care. Therefore, an appropriate system for falls detection on real time of elderly people is a problem of interest, which has been approached from different fields based on video [3], acoustic [12], inertial sensors [4] or mobile phone technology [7].

Existing fall detection approaches can be explained and categorized into three different classes, which are: camera, environment sensor, and wearable device based approaches [5]. Fall detection through cameras (vision) or environment sensors (audio, vibration) requires expensive equipment that limits the scope of activity for the person being monitored and can compromise his privacy because these sensors commonly are in indoors. In contrast, fall detection through wearable devices has been increasing because the scope of activity is relatively unrestricted, the device may be easily attached to the body, and its cost is low; this

© Springer International Publishing Switzerland 2015
A. Pardo and J. Kittler (Eds.): CIARP 2015, LNCS 9423, pp. 575–583, 2015.
DOI: 10.1007/978-3-319-25751-8_69

approach is based on embedded inertial sensors to detect the body movement and they are divided into two classes: the first class only analyzes the acceleration to detect falls (accelerometers), and the second class uses the acceleration and the body orientation (accelerometers and gyroscopes) [7].

A key factor to fall detect is the ability that possess the methodologies to acquire, manage, process and get useful information from the inertial sensor raw data, hence, the methodologies must be able to discover accurately features that differentiate a fall from the ADLs. But the raw data generated by the sensors are affected for several sources: some related to intentional movement of the body as human and gravitational acceleration (low frequency signals) and others that may add noise as external vibrations and mechanical resonance (high frequency signals), that should be attenuated by adequate filtering techniques [2].

The fall detection paradigm can be interpreted as a binary classification problem between falls and ADLs, some works implement complex inference techniques as in [9] that use hidden Markov models to analyze acceleration data, but they are inappropriate for falls detection because they spend excessive amounts of computational resources and a fast response is essential. Therefore, most solutions with wearable devices use threshold-based algorithms for detection of falls events because the processing capacity is lower [4].

In this work, we present a fall detection algorithm based on thresholds and residual events after the fall occurs, through an accelerometer worn on the human body. The algorithm uses three filtering methodologies to attenuate the sources that affect the data (Median filter, High pass Finite Impulse Response (FIR) filter, and Soft thresholding), it also uses one feature that measures the amount of acceleration to differentiate falls from ADLs (Signal Magnitude Area). To evaluate the proposed algorithm we implement two algorithms detailed in the literature and we test them using The MobilFall Dataset available online [11]. The results are presented like the capacity to detect or not detect a fall in terms of sensitivity, specificity and hit rate, showing that the proposed algorithm improvement the fall detection regarding algorithms of the literature.

2 Methods

Generality the output of the accelerometer has three signals $\boldsymbol{A} = (\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{z}(t))$ that represent the tri-axial acceleration $\boldsymbol{x}$,$\boldsymbol{y}$,$\boldsymbol{z}$ due to the motion and gravity.

Preprocessing: In the preprocessing step is used a median filter that refers to the replacement of a point $\boldsymbol{A_i}$ by the median values of the signal in a segment $\boldsymbol{M_e}\{\boldsymbol{A}_{(i,j)}\}$ this filter eliminates most of the signal generated by noise, keeping the low frequency components as are the body motion and the gravitational acceleration. Also is used a high pass filter that eliminates the low frequency corresponding to acceleration due to gravity, removing the offset from the signal to give a dynamic acceleration.

To generate a signal and to do more representative the fall with respect to ADLs considering the residual movement, is formulate the Soft Thresholding [6], that is an optimization problem of the form:

$$\underset{\boldsymbol{B}}{\operatorname{argmin}}\{||\boldsymbol{A}-\widehat{\boldsymbol{A}}||_2^2+\lambda||\widehat{\boldsymbol{A}}||_1\}$$

where $\widehat{\boldsymbol{A}}=(\widehat{\boldsymbol{x}}(t),\widehat{\boldsymbol{y}}(t),\widehat{\boldsymbol{z}}(t))$ are the estimation to be determined from the observe $\boldsymbol{A}$ signals. The regularization term $\lambda||\widehat{\boldsymbol{A}}||_1$ is chosen to promote sparsity of the solution $\widehat{\boldsymbol{A}}$. The Soft Thresholding allows depending on the regularization parameter λ selected to ensure that the noise variance is reduced to a specified fraction of its original value.

Feature Extraction: The algorithms for fall detection use the acceleration and/or orientation data to detect sequential stages to determine a fall, if the sequence is met the fall is confirmed. The sequence of stages may include: start of fallen, velocity, change of orientation and posture monitoring. To such detection the accelerometer signals are characterized through the Signal Magnitude Area (SMA) [13]:

$$SMA=\sum_i|x_i|+\sum_i|y_i|+\sum_i|z_i|,$$

where x_i, y_i, z_i are the ith sample of the $\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}$ axis respectively. This feature is independent of the orientation of the device and corresponds to the amount of acceleration that an user has exerted on the accelerometer.

Fall Detection Algorithm: The fall detection algorithm operates in a series of steps represented in the Algorithm 1. First over an observation window in the accelerometer data three preprocessing methods (a median filter, a high pass FIR filter and finally the Soft Thresholding) are applied in order to make more representative the fall. After the Signal Magnitude Area (SMA) is calculated to get the accumulated acceleration and to evaluate if the accumulated acceleration exceeds an acceleration threshold empirically determined, meaning the user is engaged in a high energy activity like running, jumping or a possible fallen. So, if this upper acceleration threshold is exceeded the window is moved a determinate time (sliding windows) and over it is applied again the three preprocessing methods and is calculated the Signal Magnitude Area (SMA), finally if this SMA value no exceeds a lower acceleration threshold empirically determined, it means the user is in a low energy activity and the fall is detected.

3 Experiments

Usually for falls detection through portable devices are used the methodology presented in Fig. 1, our work emphasizes on the stage of data preprocessing and fall detection.

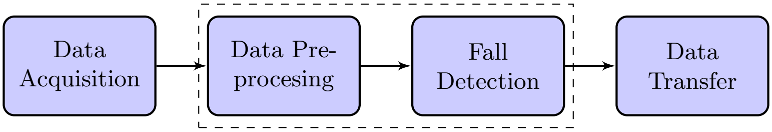


Fig. 1. Framework for fall detection.

The data set used called The MobilFall Dataset (second version) [11] was recorded on the Biomedical Informatics & eHealth Laboratory, at the Department of Informatics Engineering of the Technological Educational Institute of Crete. Data from the accelerometer and gyroscope sensor of a smartphone were recorded with sampling period $7.6ms$ at $87Hz$ mean sampling frequency for the accelerometer and $0.3ms$ at $200Hz$ mean sampling rate for the gyroscope. The MobilFall Dataset contains data from 24 volunteers: seventeen males (age: $22-47$ years, height: $1.69-1.89m$, weight: $64-103kg$) and seven females (age: $22-36$ years, height: $1.60-1.72m$, weight: $50-90kg$). Nine participants performed falls and ADLs, while fifteen performed only the falls, discriminated as shown in the table Table 1.

To build the fall detection algorithm we use the following parameters based on the basic trade-off between detecting all falls and avoiding false positives: The windows length is $l = 2$ seconds, the first acceleration threshold is $11m/s^2$, the second acceleration threshold is a value close to zero, and the wait time is equivalent to $n = 1$ second. In the preprocessing step we use the follow set of values to ensure good filtering as they do in [1]: for the median filter we determine

---

Algorithm 1. Fall Detection

Input acceleration: $\boldsymbol{A} \in \mathbb{R}^{3\times t}$, Windows length: l and Thresholds: $\zeta_u, \zeta_l \in \mathbb{R}^+$.
Initialize the windows $\boldsymbol{A}_{(i,j)}$ where $l = |i - j|$.
while Fall is not detected **do**
  Apply the Median Filter, the High Pass Filter, and the Soft Thresholding.
  Calculate $\boldsymbol{A}_{SMA} = SMA\{\boldsymbol{A}_{(i,j)}\}$
  if $\boldsymbol{A}_{SMA} > \zeta_u$ **then**
    wait n samples and set $\boldsymbol{A}_{(i+n,j+n)}$
    Apply the Median Filter, the High Pass Filter, and the Soft Thresholding.
    Calculate $\boldsymbol{A}_{SMA} = SMA\{\boldsymbol{A}_{(i+n,j+n)}\}$
    if $\boldsymbol{A}_{SMA} < \zeta_l$ **then**
      Fall is detected
    else
      Fall is not detected
    end if
  else
    Fall is not detected
  end if
  Increment the windows.
end while

Table 1. The MobiFall Dataset.

Type		Trials	Time	Description
FOL	Forward-lying	3	$10s$	Fall Forward from standing, use of hands to dampen fall
FKL	Front-knees-lying	3	$10s$	Fall forward from standing, first impact on knees
SDL	Back-sitting-chair	3	$10s$	Fall backward while trying to sit on a chair
BSC	Sideward-lying	3	$10s$	Fall sidewards from standing, bending legs
STD	Standing	1	$5m$	Standing with subtle movements
WAL	Walking	1	$5m$	Normal walking
JOG	Jogging	3	$30s$	Jogging
JUM	Jumping	3	$30s$	Continuous jumping
STU	Stairs up	6	$10s$	Stairs up (10 stairs)
STN	Stairs down	6	$10s$	Stairs down (10 stairs)
SCH	Sit chair	6	$6s$	Sitting on a chair
CSI	Car-step in	6	$6s$	Step in a car
CSO	Car-step out	6	$6s$	Step out a car

Table 1 Shows the Falls (FOL, FKL, SDL, BSC) and the ADLs (STD, WAL, JOG, JUM, STU, STN, SCH, CSI, CSO) recorded in the MobiFall Dataset.

a filter of order 13, in the high pass FIR filter we use a order of 35, with stop frequency of $0.5Hz$. And to select the regularization parameter λ in the Soft Thresholding we set a series of values and select the value that gives better results in the methodology respect to the Hit Rate, Sensitivity and Specificity, as shows the Fig. 2, that value corresponds to $\lambda = 0.1$.

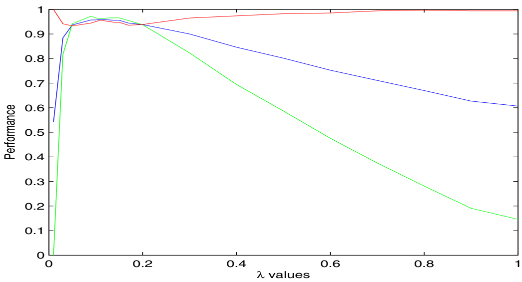
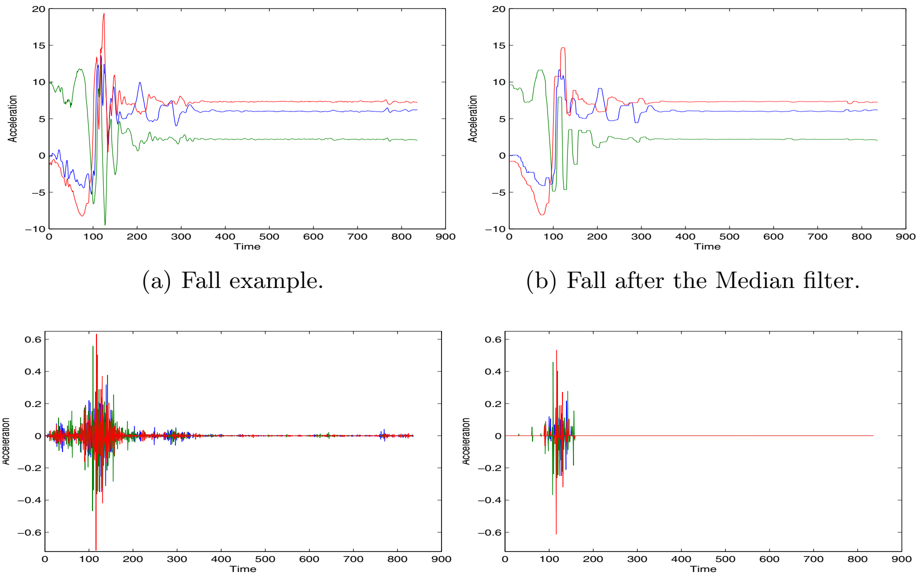


Fig. 2. Performance against λ, respect to Hit Rate, Sensitivity, Specificity.

The Fig. 3 shows an example of how the preprocessing step with the three filter methods affects the accelerometer signals to generate a representation more suitable of a fall taking into account the residual events. The Fig. 3(a) shows a fall of type FOL. The Fig. 3(b) shows the fall signal after of apply the median filter. The Fig. 3(c) shows the fall signal filtered Fig. 3(b) after of apply the high pass FIR filter. And the Fig. 3(d) shows the fall signal filtered Fig. 3(c) after of apply the Soft Thresholding.



(a) Fall example.

(b) Fall after the Median filter.

(c) Fall after the HP-FIR filter.

(d) Fall after the Soft Thresholding.

Fig. 3. Example of the preprocessing step in a fall signal. x axis, y axis, z axis.

In order to compare the proposed algorithm we implement two algorithms detailed in the literature:

Panagiotis Algorithm: This algorithm is reported by [10], and detects the fall when the magnitude of the acceleration overcomes two thresholds. If the magnitude of the acceleration exceeds the first threshold referred as upper, then the algorithm waits a predetermined time interval and evaluates if the magnitude of acceleration exceeds the second threshold referred as lower, thus, if the two thresholds are overcome the fall is detected.

Suleman Algorithm: This algorithm is reported by [8], and detects the fall evaluating two thresholds. If the norm $L1$ of the acceleration calculated every second exceeds a first threshold of acceleration, the algorithm waits a predetermined time and checks the orientation through the tilt angle, and if a second threshold of angle is overcome by the tilt angle, this means that the user is not standing and has fallen.

To evaluate properly the fall detection algorithm the results are presented in terms of sensitivity, specificity and hit rate as shown in the table 2, the results are represented like the capacity of detect or no a fall, so, the sensitivity represents the capacity to detect falls, the specificity represents the capacity of only detect falls and ignore non fall events, and the hit rate represents the proportion of true results among the population.

Table 2. Results of the four fall detection algorithms.

		Panagiotis Alg [10]		Suleman Alg [8]		Proposed Alg	
Activity	Type	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
Fall	FOL	0,9722		0,9306		0,9583	
	FKL	0,9306		0,9861		0,9722	
	SDL	0,9444		0,9722		0,9722	
	BSC	0,9722		0,9167		0,9583	
ADL	STD		1,0000		1,0000		1,0000
	WAL		0,1111		1,0000		1,0000
	JOG		0,7037		0,7037		1,0000
	JUM		1,0000		0,3333		1,0000
	STU		0,6481		1,0000		1,0000
	STN		0,5556		1,0000		0,9074
	SCH		1,0000		1,0000		0,9444
	CSI		1,0000		0,6296		0,8889
	CSO		1,0000		1,0000		0,9444
Total		0,9549	0,8275	0,9514	0,8655	**0.9653**	**0,9503**
Hit Rate		0,8857		0,9048		**0,9571**	

4 Discussion and Conclusion Remarks

In the present work, we implement a fall detection algorithm based on thresholds to detect sequential stages in inertial sensor data, the MobilFall Dataset (second version) was used in order to evaluate the fall detection algorithm comparing it against two algorithms detailed in the literature. The result presents in the table 2 show the strengths and weaknesses of each one of the algorithms with respect to the different types of fall and ADLs.

As seen in the Fig. 3 for a fall detection algorithm in the step of preprocessing an adequate filtering techniques allows detect patterns corresponding to a fall and give acceleration signals more suitable to the classification step to decide between a fall and an ADL. Therefore the median filter Fig. 3(b) reduces the noise in the accelerometer signal and provide to the High Pass FIR filter a cleaner signal. The High Pass FIR filter Fig. 3(c) is used to remove the offset from the signals and obtain a dynamic acceleration. The Soft Thresholding Fig. 3(d) promotes sparsity to the acceleration signal representing the low energy activities as zero, necessary in the algorithm to decide whether there has been a fall, since in the moment of detect the fall is important to consider the residual events, like they are: the user direction, the energy activity, and the normal acceleration values.

In the table 2 we see that the Panagiotis algorithm [10] presents better capacity of detect falls that of ignore non fall events, higher sensitivity than specificity. It works better for falls of type FOL, BSC than for FKL and SDL; on the other hand the algorithm works well for ADLs as STD, SCH, CSI, CSO. Commonly these activities do not exceed the upper threshold and if exceed it, they do not exceed the lower threshold. Unlike the algorithm does not perform well with WAL, JOG, STU and STN that are higher energy activities that they can exceed the upper and lower thresholds. We see also that the Suleman algorithm [8] has less sensitivity and higher specificity with respect to Panagiotis algorithm, this algorithm present more sensitivity for falls of type FKL and SDL than for FOL and BSC falls; at the ADLs the algorithm presents more specificity for STD, WAL, STU, STN, SCH, CSO than for JOG, JUM, CSI; as the algorithm works with the subject direction these three ADLs present more intensity that affects the device direction.

The results in the table 2 shows with respect to the Hit Rate, the total of sensitivity and specificity that the proposed algorithm performs better than the Panagiotis and Suleman algorithms that are detailed in the literature. For falls we see that our algorithm in terms of sensitivity compared to the Suleman algorithm improves in FOL, BSC and keeps in SDL and in terms of specificity it is equal in STD, WAL, and STU because these activities do not exceed the first threshold in the two algorithms. For ADLs the proposed algorithm has more specificity in JOG, JUM, and CSI because it does not take into account the direction (in these ADLs the Suleman algorithm presents the lowest values of specificity), our algorithm has less specificity in STN, SCH, and CSO; probably It detects them as a fall, because these activities present a moment of high energy where the first threshold could be overcome, followed by a moment of low energy where the second threshold could be not overcome.

As future work, it is important to test the proposed algorithm in others data sets, and find automatically the optimal regularization parameter λ in the Soft Thresholding.

Acknowledgments. This work was supported by the project *"Plataforma tecnológica para los servicios de teleasistencia, emergencias médicas, seguimiento y monitoreo permanente a los pacientes y apoyo a los programas de promoción y prevención"*, code "Ruta-N: 512C-2013"; the research project 16882 funded by Universidad Nacional de Colombia and Universidad de Caldas; and the the research projects 111065740687, 111056934461 funded by COLCIENCIAS.

References

1. de Alba, S.G., Clifford, M., Reyna, R., Palomera, E., Gonzalez, R.: Human fall detection using 3-axis accelerometer (2005)
2. Figo, D., Diniz, P.C., Ferreira, D.R., Cardoso, J.M.P.: Preprocessing techniques for context recognition from accelerometer data. Personal Ubiquitous Comput. **14**(7), 645–662 (2010)
3. Ke, S.R., Thuc, H.L.U., Lee, Y.J., Hwang, J.N., Yoo, J.H., Choi, K.H.: A review on video-based human activity recognition. Computers **2**(2), 88–131 (2013)
4. Li, Q., Stankovic, J., Hanson, M., Barth, A., Lach, J., Zhou, G.: Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In: Sixth International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2009, pp. 138–143 (2009)

5. Mubashir, M., Shao, L., Seed, L.: A survey on fall detection: Principles and approaches. Neurocomputing **100**, 144–152 (2013). special issue: Behaviours in video
6. Chen, P.-Y., Selesnick, I.W.: Overlapping group shrinkage/thresholding and denoising. In: ASEE 2014 Zone I Conference, September 2012
7. Hwang, S.-Y., Ryu, M.-H., Yang, S.Y., Lee, N.B.: Fall detection with three-axis accelerometer and magnetometer in a smartphone. In: Computer Science and Technology, pp. 65–70 (2012)
8. Kazi, S.B., Sikander, S., Yousafzai, S.: Fall detection using single tri-axial accelerometer. In: ASEE 2014 Zone I Conference, April 2014
9. Tong, L., Song, Q., Ge, Y., Liu, M.: Hmm-based human fall detection and prediction method using tri-axial accelerometer. Sensors Journal, IEEE **13**(5), 1849–1856 (2013)
10. Torrent, J., Kostopoulos, P., Nunes, T., Salvi, K., Deriaz, M.: Increased fall detection accuracy in an accelerometer-based algorithm considering residual movement. In: ICPRAM 2015, 4th International Conference on Pattern Recognition Applications and Methods, January 2015
11. Vavoulas, G., Pediaditis, M., Spanakis, E., Tsiknakis, M.: The mobifall dataset: an initial evaluation of fall detection algorithms using smartphones. In: 2013 IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE), pp. 1–4, November 2013
12. Yazar, A.: Multi-sensor based ambient assisted living (2013)
13. He, Y., Li, Y., Chuan, Y.: Falling-incident detection and alarm by smartphone with multimedia messaging service (mms). Computer Science & Communications, ETSN **1**(1), 1–5 (2012)