Threshold-based Fall Detection on Smart Phones

Sebastian Fudickar, Alexander Lindemann and Bettina Schnor
Department of Computer Science, University of Potsdam, August-Bebel Str. 89, Potsdam, Germany

Keywords: Fall detection, Development of Assistive Technology, Sensors-based Applications.

Abstract: This paper evaluates threshold-based fall detection algorithms which use data from acceleration sensors that are part of the current smart phone technology. The evaluation was done with sampled fall records where young people simulate falls. To test the false positive rate of the algorithms, another record set with Activities of the Daily Living (ADLs) from elderlies was used. The results are very promising and show that smart phone sensors are suitable for fall detection. This will offer a new opportunity to assist elderlies in their daily living and extend their period of self-determined living.

1 INTRODUCTION

With physical disorders gaining relevance with increasing age to the elderlies' daily living, the probability of falls increases and the probability of full health-recoverage from falls decreases (especially if falls are not instantaneously detected) (I. S. Joint Commission Resources, 2008). Consequently, first-responders must be notified instantaneously if elderlies fall to increase healing chances.

Mobile devices like smart phones are equipped with accelerometers which may be suited for automatic fall detection. This is an interesting opportunity for Assistive Living technologies where the automatic fall detection can initiate an emergency call which also includes the localization information where the patient is fallen.

Prior studies have shown that the best results are achieved when the mobile device is worn at the hip (Kangas et al., 2008). The optimum should combine a high sensitivity with a high specificity. The sensitivity is a measure for the correctly detected falls:

$$Sensitivity = \frac{TruePositives}{Number\ of\ all\ falls}$$

where $TruePositives$ is the number of correctly detected falls.

The specificity measures the rate of $TrueNegatives$, i.e. the percentage of correctly classified non-fall situations:

$$Specificity = \frac{TrueNegatives}{Number\ of\ all\ non-falls}$$

A high specificity means a low number of false alarms. This is also a very important feature to make a fall detection system a suited assistive technology.

Modern smart phones are equipped with a tri-axial accelerometer sensor which collects periodically a vector with axis-specific acceleration. Typically, this is used to re-orient the screen as a user moves the device. But this sensor has also potential to be used for fall detection.

In context of the Assistive Living project KopAL (Fudickar et al., 2011), we use a mobile device called *Efficient Mobile Unit (EMU)* (Fudickar et al., 2012a) which is equipped with the ADXL345 accelerometer. This is a high-end accelerometer which is able to sample data with up to 800 Hz (for the I^2C bus) and which is capable of in-hardware preprocessing. A simulation study has shown that the EMU running the ADXL345 at 800 Hz has a very high sensitivity of 93% (Fudickar et al., 2012b).

Recently, it was indicated by Mehner et al. (Mehner et al., 2013) that the algorithm's detection rate is even accurate with sampling rates below 800 Hz. The lower sampling rates are typical for multi-purpose smart phones that are as well equipped with acceleration sensors. Smart phone operating systems such as Android aim to save energy and typically sample with low rates (of up to 100 Hz). The question was whether these lower sampling rates will allow a high sensitivity.

Since the authors evaluate their algorithm with a different fall set, it is not directly comparable to the original one from (Fudickar et al., 2012b). Hence, we extended the simulator to evaluate both algorithms

Fudickar S., Lindemann A. and Schnor B..
Threshold-based Fall Detection on Smart Phones.
DOI: 10.5220/0004795803030309
In *Proceedings of the International Conference on Health Informatics* (HEALTHINF-2014), pages 303-309
ISBN: 978-989-758-010-9

(the one optimized in (Fudickar et al., 2012b) and the one of Mehner et al. (Mehner et al., 2013)) with our recorded falls and the simulator first proposed in (Fudickar et al., 2012b), to assure a meaningful comparability.

Since the initial Activities of the Daily Living (ADLs) were recorded by young probands that executed critical ADLs in a high frequency and in an exaggerated manner, the results may be less realistic for elderlies. While recording falls of elderlies is still a challenging and critical task, the recording of ADLs of the elderly is less problematic. Therefore, ADLs of elderlies were recorded in a nursing home and enabled us to test also the specificity of both algorithms under realistic conditions for the intended user group.

The remainder of the article is structured as follows: In Section 2 related work is discussed. The basic algorithm of the threshold-based fall detection is introduced in Section 3. Section 4 presents the extension of the simulator. It follows the description of the evaluation environment. The results of the evaluation are presented in Section 6. The article ends with a conclusion.

2 RELATED WORK

Jia (Jia, 2011) proposed a threshold-based fall detection algorithm for the accelerometer ADXL345 which is able to pre-process raw acceleration data itself. This feature can be used to let the processor remain in low-power mode until a special event, in our use case a free fall, was detected. This makes the accelerometer beneficial for the use in mobile devices since it helps to save energy.

The Efficient Mobile Unit (Fudickar et al., 2012a) is a mobile device dedicated for Assisted Living scenarios which is equipped with the ADXL345. In (Fudickar et al., 2012b), the parameter of Jia's threshold-based fall detection were optimized for the EMU. For this purpose, the authors proposed a fall-detection simulator which is able to model the threshold-based fall detection. The simulator uses pre-recorded data records from fall-situations and ADLs. ADLs were generated for the simulator from recorded movements of three probands with an age between 20 and 30 years.

The optimal parameter set for the EMU was identified by running several simulations to test the complete parameter space. This resulted in an optimized algorithm with a sensitivity of 93% compared to a sensitivity of 33% of the original Jia algorithm.

The work of Mehner et al. (Mehner et al., 2013) indicated that threshold based fall-detection algo-

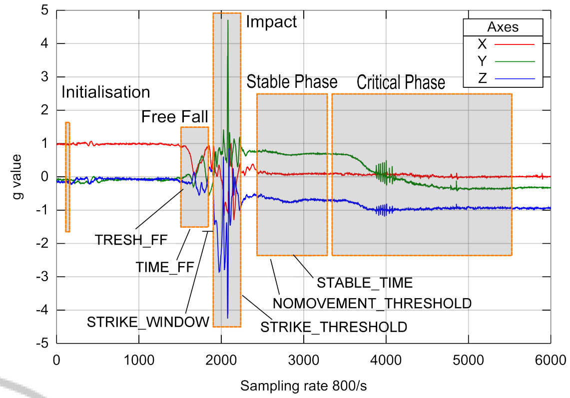

Figure 1: States of a fall shown for a frontal fall without loss of consciousness (Fudickar et al., 2012b).

rithms are as well applicable for smart phones and that the lower sampling rates (such as 50 Hz) that are supported by the smart phone's operating system are uncritical. Furthermore, they indicated that the exclusion of the free-fall detection may increase the detection accuracy by 27 % from 56% with free-fall detection to 83% without free-fall detection. Overall the proposed algorithm achieved a maximal sensitivity of 83% and a specificity of 100%.

Sannino et al. (Sannino et al., 2013) have recently proposed a fall detection algorithm that is based on supervised knowledge extraction for a windowing technique and was optimized with the an subset of the recorded fall set from (Fudickar et al., 2012b). The resulting algorithm achieved a promising sensitivity of 91% and a specificity of 92 % as the average of 25 runs for a seperate subset (testing set) of the fall set.

3 THRESHOLD BASED FALL DETECTION

For a tri-axial accelerometer, fall situations are characterized by multiple sequential events, as shown in Figure 1. The accelerometer collects a vector (x,y,z) of the axis-specific acceleration. While the following description from (Fudickar et al., 2012b) is applicable to threshold based fall detection algorithms in general, the parameter settings are described according to Jia (Jia, 2011).

Within a fall situation, the falling body experiences zero-gravity during free falls. The free fall is the initial event of each fall situation and therefore is identified first. For example, for the accelerometer ADXL345 used in our experiments, the gravity must drop below the free fall threshold THRESH_FF (as described in Table 1) for a minimal duration of

Table 1: The optimal register settings are identical for all evaluated sampling rates and fall-detection algorithms.

Parameter's Register name	Description	Optimized algorithm (Register setting)
THRESH_FF	free-fall threshold	0.75 *g* (0x0C)
STRIKE_THRESHOLD	minimal impact per axis	2 *g* (0x20)
STRIKE_WINDOW	maximal delay between free fall and impact	500 *ms* (0x19)
STABLE_TIME	minimal stable-phase duration	1 *s* (0x01)
STABLE_WINDOW	maximal stable-phase duration	3.5 *s* (0xAF)
NOMOVEMENT_TIME	duration of critical phase	5*s* (0x19)
NOMOVEMENT_THRESHOLD	maximal acceleration during stable-phase	0.4375 *g* (0x07)
TIME_FF	minimal free-fall duration	30 *ms* (0x06)

TIME_FF. Other algorithms (such as (A.K.Bourke et al., 2007)) exclude a minimal duration for free fall detection but instead detect them only via threshold value.

Once a free fall occurred, the algorithm tries to detect an impact, which is given if the acceleration values of all three axis exceed the so-called STRIKE_THRESHOLD. The maximal duration between free fall and impact recognition may not exceed the duration defined in STRIKE_WINDOW. If no impact was detected within this period, no fall was detected and the algorithm resets the accelerometer to detect further free falls.

In a fall situation the impact is followed by a stable phase. During the stable phase, all axis' acceleration values drop below the NOMOVEMENT_THRESHOLD, for a minimal duration of STABLE_TIME. A fall is recognized, if a free fall, an impact and a stable phase were detected sequentially. Therefore, the detection of the stable phase is essential for the fall detection, but does not indicate a loss of consciousness.

Instead, loss of consciousness is detected during the critical phase, which follows the stable phase. If no movements were recognized during the critical phase (we chose 5 seconds), a loss of consciousness can be assumed, caused by the previous fall.

4 SIMULATOR EXTENSION

The fall detection simulator from (Fudickar et al., 2012b) was extended by the following aspects:

Since we wanted to compare two different threshold based fall detection algorithms (one with and one without the free fall phase detection), we had to implement the one which ignores the fall's initial free fall phase. This altered algorithm was implemented according to the one presented in (Mehner et al., 2013) and is used to evaluate the fall detection algorithm for mobile devices with less complex accelerometers.

The original threshold-based fall detection algorithm classifies the following three fall types:

1. **Normal Falls:** cover falls where the proband moves again.
2. **Critical Falls:** describe falls where the proband does not move after the impact and loss of consciousness (l.o.c.) is assumed.
3. **Critical Free Falls:** are characterized by multiple free fall events before the impact, which are e.g. typical for falling down stairs. Critical free falls do not differentiate regarding the l.o.c. since they are in any case critical.

The fall detection algorithm without free fall detection classifies detected falls into *normal* or *critical* falls and can not detect the third fall type. Instead, the associated falls are detected as normal or critical falls.

The other modification was the implementation of an additional simulation stage which allows us to simulate different sampling rates (50 Hz, 100 Hz, 200 Hz, 400 Hz, 800 Hz).

5 EVALUATION ENVIRONMENT

The sensitivity and specificity of the threshold based fall detection algorithms is evaluated with the fall sets and ADLS that are described in the following subsections. All data records were sampled with the EMU (Fudickar et al., 2012a) since it is equipped with the ADXL345 accelerometer which is capable to sample data with a high sampling rate of 800 Hz.

5.1 Fall Set

The fall set was taken from (Fudickar et al., 2012b). Since, the risk of injuries in case of older probands is obviously much too high, the *fall set* was recorded by three young probands with an age between 20 and 30 years including both genders. The fall set covered frontal falls, backward falls, falls to the left and to the right, falling after standing up, falling from bending

Table 2: Detected falls with and without free fall detection for the analyzed sampling rates. Falls are only counted at the most critical fall-type. The sum is calculated from the detected falls and critical falls and critical free falls.

Sampling rate	with free fall detection				without free fall detection		
	normal falls	critical fall	critical free fall	sum falls	normal falls	critical fall	sum falls
800 Hz	29	43	6	78 (92%)	35	48	83 (99%)
400 Hz	32	41	6	79 (94%)	37	46	83 (99%)
200 Hz	29	45	4	78 (92%)	34	48	82 (98%)
100 Hz	28	44	7	79 (94%)	34	48	82 (98%)
50 Hz	28	45	4	77 (92%)	34	49	83 (99%)
Expected falls	36	36	12	84	42	42	84

down while picking up a book and falling from stairs as suggested by (Wang et al., 2008). The half of the allover 84 recorded falls were critical falls, with loss of consciousness (where the proband did not move after the falls). The other half were falls after which the proband moved again.

5.2 ADLs of Elderlies

To evaluate the specificity of the algorithms, we recorded the acceleration characteristics of ADLs of elderlies. Thereby, elderlies were equipped with a fanny pack, worn at the probands' hip in which the EMU was carried.

The recordings took place in the nursing home Florencehort, Stahnsdorf, Germany on two consecutive days in August 2013. Nine elderlies (four males and five females) with an average age of 82 years (ranging from 70 to 95 years) participated in the recordings. Among the participants, one was using a wheelchair and four were using a wheeled walking frame.

Only recordings of eight probands are used for the evaluation of the algorithm's specificity, since the recording failed in one case due to a device error. Allover, more than 41 hours of ADLs were recorded. The phases of attachment and detachment of the fanny pack were excluded (cropped) from the recordings. This results in a considered duration of allover 39 hours in which 37695360 acceleration values were recorded. The considered recording duration per device ranged from 3.37 to 5.93 hours and was in average about 5 hours. The recordings started around 10:15 a clock and therefore covered typical daily activities including eating lunch, walks and potential afternoon naps.

Figures 2 shows the recorded acceleration values (x,y,z) plotted as the length of the vector:

$$\sqrt{x^2+y^2+z^2}$$

The maximal recorded acceleration in the cropped recordings was at 8.7 g. However, such measurements over 5 g occurred rarely and its occurence is marked in the Figures by a red cross at the upper bound. Further, some missing samples exist in the recordings shown in Figure 2 c) and h) due to recording problems of the devices (which are still in prototype stage). The acceleration of the recording shown in Figure 2 d) varied less than the one of the others and thereby was probably sampled with the proband in the wheel chair.

6 RESULTS

With the extended simulator and the additional recorded ADLs of elderlies, we evaluated both threshold based fall detection algorithms (with and without free-fall detection) and the lower sampling rates to get insights on their influence on the sensitivity and specificity.

The evaluation results are discussed separately for the sensitivity and specificity in the following.

6.1 Sensitivity

Ahead of the evaluation, the threshold parameter's settings were optimized for the *fall set* for all sampling rates and both algorithms (with and without free fall detection), in respect to potential variations of the optimum. However, the optimal parameter settings as expected did not vary or change since they rather represent the fall characteristics, which did not change. Therefore, the optimal parameter settings are as shown in Table 1 identical to the ones that were identified as optimal in (Fudickar et al., 2012b). Consequently, the evaluated algorithms differed only by the inclusion of the free-fall detection step and the used sampling rate.

For the evaluated sampling rates (between 50 Hz and 800 Hz), the fall detection algorithm's overall sensitivity is minor affected by the sampling rate, as shown in Table 2. The associated confusion matrixes are shown in Table 3 and in Table 4. For the applied fall records, a sampling rate of 400 and 50 Hz resulted in a slightly higher fall-detection rate. In contrast to the sampling rate, the exclusion of the free-fall detection significantly increased the algorithm's sensitivity

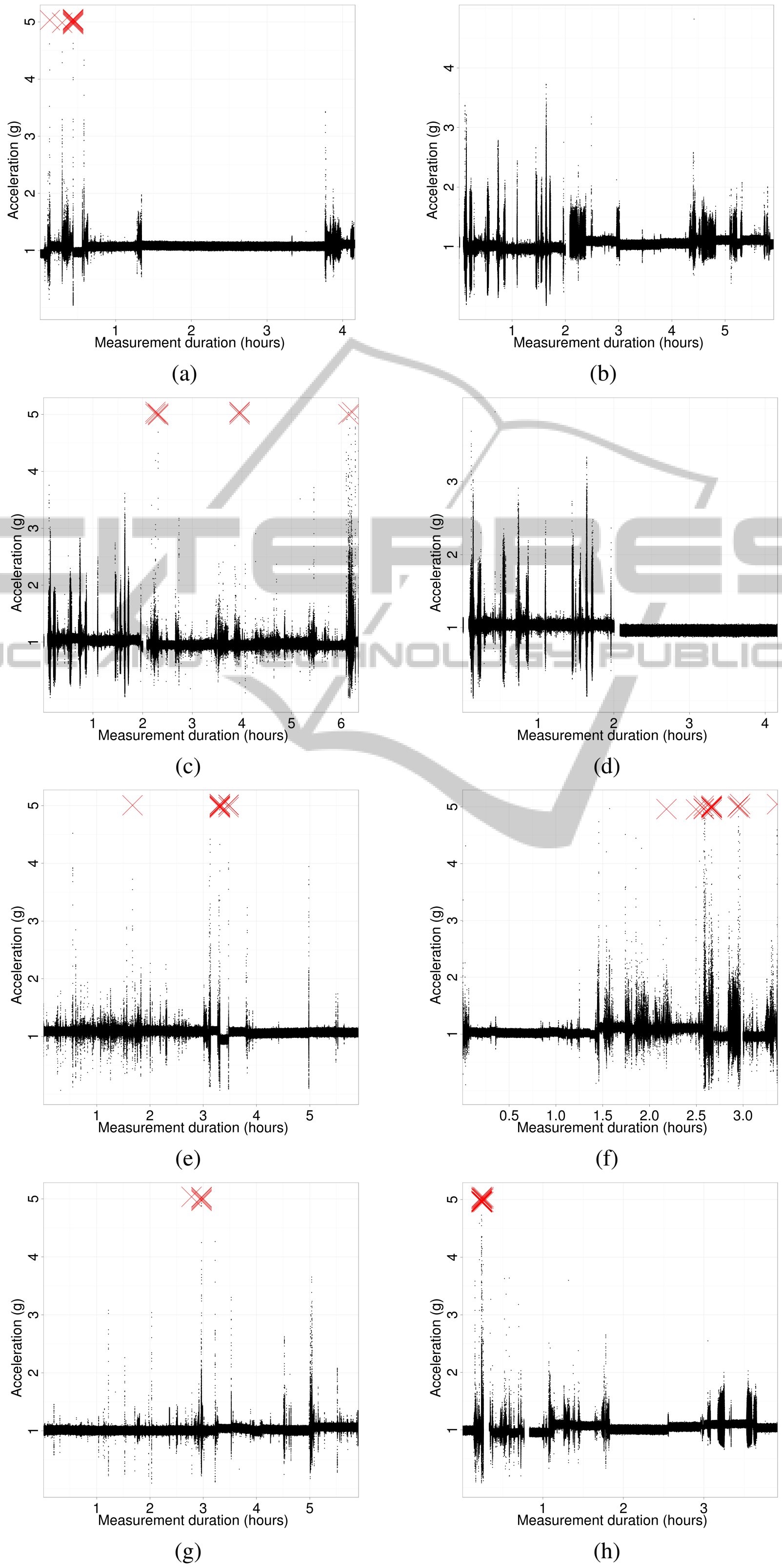

Figure 2: The acceleration in g (black) of the ADL records of elderlies (red crosses indicate the time of acceleration measure that exceeds 5 g, which occured in a) 17, in c) 8, in e) 12, in f) 20, in g) 7 and in h) 15 times).

Table 3: Confusion matrix for fall detection algorithm with freefall detection at 50 Hz.

	Detected as Falls	Detected as ADL
Falls	77	7
ADLs	0	ca. 2355960

Table 4: Confusion matrix for fall detection algorithm without freefall detection at 50 Hz.

	Detected as Falls	Detected as ADL
Falls	83	1
ADLs	0	ca. 2355960

and increased the detection rate by up to 5% to 83 falls (99%) without free-fall detection instead of 79 falls (94%) with free-fall detection.

6.2 Specificity

The specificity of both algorithms was evaluated with the records of the elderlies' ADLs. Both algorithms have optimal specificity, since no false positive fall was detected. Hence in our experiments, the free-fall detection had no influence on the specificity.

However, the specificity of the fall-type detection differs and is without free fall detection less accurate (see Table 2). While most falls are correctly detected, the type of a fall is regularly false detected among all algorithm configurations (see Table 2). Typically *normal falls* are detected as critical falls. Since in any fall-situation (even the ones without loss of consciousness) an emergency should be notified, this aspect is unproblematic, but might be further investigated.

6.3 Interpretation of the Results

The results indicate that the exclusion of the free-fall detection step enhances the sensitivity, while not affecting the algorithms specificity. Thereby, we could confirm the results of (Mehner et al., 2013) who proposed the exclusion of this processing step. Furthermore, we confirmed that the data sampling rate has minor impact on the algorithm's sensitivity and specificity in the tested range (50 – 800 Hz). The algorithm's optimal parameter settings have shown to be independent from the data sampling rates at all.

7 CONCLUSIONS

We evaluated threshold-based fall detection algorithms which can be used on smart phones. Our motivation was that multi-purpose smart phones are equipped with acceleration sensors. Smart phone operating systems such as Android aim to save energy and typically sample with low rates (typically between 20 and 100 Hz). The question was whether these lower sampling rates will allow a high sensitivity.

Therefore, we extended the simulator presented in (Fudickar et al., 2012b) to compare different sampling rates and two different fall detection variants: one with and one without detection of a free fall phase.

Our results show that

1. Fall detection with low sampling rates of at least 50 Hz can be used and have a sensitivity of 99% for our fall records.
2. The sensitivity of the fall detection algorithm variant without the detection of a free fall phase is (with up to 5 %) slightly better. From 84 falls, the algorithm with free fall phase detection recognized 77 falls where the algorithm without free fall detection recognized 83 falls.
3. The specificity of both algorithms regarding the false-positive rates is perfect for the recorded ADLs of the elderly.

The application programming interfaces for the access of accelerometers of current smart phone OSs such as Android and iOS are limited regarding the configuration of specific sampling rates and the access of interrupts. Therefore, we are looking forward to propose better extensions and will integrate the fall detection into the Kernel of the Android OS.

Android based smart phones use in some operation modes even lower sampling rates (up to 20 Hz) than the ones used in our simulation to save energy on the mobile devices. Therefore, we plan to adapt the simulator to work also for this lower sampling rates.

ACKNOWLEDGEMENTS

The nursing home Florencehort in Stahnsdorf which belongs to the Landesauschuss für innere Mission (LAFIM) supports sincerely our studies. We like to thank the LAFIM team and all of the probands for their patient support of our work.

REFERENCES

A.K.Bourke, J.V.O'Brien, and G.M.Lyons (1 July 2007). Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & Posture*, 26:194–199.

Fudickar, S., Frohberg, M., Taube, S., Mahr, P., and Schnor, B. (2012a). An energy efficient mobile device for assisted living applications. In *2012 IEEE Online Conference on Green Communications (IEEE GreenCom'12)*.

Fudickar, S., Karth, C., Mahr, P., and Schnor, B. (2012b). Fall-detection simulator for accelerometers with in-hardware preprocessing. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '12, pages 41:1–41:7, New York, NY, USA. ACM.

Fudickar, S., Schnor, B., Felber, J., Neyer, F. J., Lenz, M., and Stede, M. (2011). KopAL - An Orientation System For Patients With Dementia. In Björn Gottfried and Hamid Aghajan, editor, *Behaviour Monitoring and Interpretation - BMI*, pages 83–104. IOS Press, Amsterdam, Netherlands.

I. S. Joint Commission Resources (2008). Reducing the risk of patient harm resulting from falls. *Joint Commission Resources*.

Jia, A. D. N. (2011). AN-1023 - Fall Detection Application by Using 3-Axis Accelerometer ADXL345. Analog Devices. `http://www.analog.com/static/imported-files/application_notes/AN-1023.pdf`; accessed June 1, 2013.

Kangas, M., Konttila, A., Lindgren, P., Winblad, I., and Jamsa, T. (2008). Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait & posture*, 28:285–91.

Mehner, S., Klauck, R., and Koenig, H. (2013). Location-independent fall detection with smartphone. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '13, New York, NY, USA. ACM.

Sannino, G., Falco, I. D., and Pietro, G. D. (2013). Effective supervised knowledge extraction for an mhealth system for fall detection. In *Medicon*.

Wang, C., Chiang, C., Huang, C., and Chan, C. (2008). Development of a fall detecting system for the elderly residents. *The Second International Conference of Bioinformatics and Biomedical Engineering*, 0:1359–1362.