A combined smartphone and smartwatch fall detection system

Thomas Vilarinho and Babak Farshchian
SINTEF ICT
SINTEF
Trondheim, Norway
Emails: {thomas.vilarinho,babak.farshchian}@sintef.no

Daniel Gloppestad Bajer, Ole Halvor Dahl, Iver Egge,
Sondre Steinsland Hegdal, Andreas Lønes,
Johan N. Slettevold and Sam Mathias Weggersen
Norwegian University of Science and Technology (NTNU)
Department of Computer and Information Science
Trondheim, Norway
Emails: {danielgb,olehda,iwasperu,sondresh,
andrlone,johannsl,sammw}@stud.ntnu.no

***Abstract*—Falls are behind many elderly hospitalizations and can lead to injuries that greatly debilitate old patients. Much of the deployed fall detection systems rely on the user wearing a personal emergency response device, being conscious and at home. The limitations of the existing systems regarding usability and efficiency have yield an overarching research question on whether systems based on new and advanced consumer mobile devices can be used as ubiquitous automatic fall detectors for seniors.**

This paper specifically looks into the accuracy of a fall detection system based on an off-the-shelf smartwatch and smartphone. We have implemented a system which combines threshold based and pattern recognition techniques in both devices, with the intent of having the watch to contribute to the specificity of the fall detection strategy.

We tested the accuracy of the system through a series of simulated falls and activities of daily living, resulting on the correct identification of 63% of the falls and 78% of the activities and outperforming two other baseline fall detection applications (iFall and Fade). The sensors and algorithm on the watch were able to provide a marginal contribution to the system's accuracy. Indications from the tests suggest that it should be possible to improve the system accuracy by adjusting the used thresholds and fuzzyfying them. Moreover, it is expected that the open source nature of this work and it's results boost such threshold tuning and serve as a better basis for researchers to benchmark their work.

Keywords*—*Ubiquitous computing, Wearable Computers, Mobile Applications, Health and Pattern Recognition

I. Introduction

Around one-third of elderly fall at least once a year [1]. Elders living alone risk having their fall unnoticed and not being able to stand up by themselves and ask for help. [2] indicates that 47% of elderly fallers were unable to stand-up without assistance in at least one fall. On the other hand, a long waiting time before assistance increases the chances of hospitalization and death [3]. Therefore, an early detection and assistance in case of a fall is important for increasing elderly life expectancy.

Much of the existing fall detection systems rely on some sort of push-button device (in the shape of a pendant or wristband) which the user is supposed to activate after a fall and which will connect to a fall alarm center through an in-house communication system. Such systems will not trigger the alarm in case the one falling is unconscious or is outside of his home. Those limitations together with the dropping cost of accelerometer hardware have triggered an enormous growth on the research of automatic fall detection systems based on accelerometers [4].

A recent systematic review on automatic fall detection systems using body-worn sensors [4] gathered 96 publications from 1998 to 2012. From within that list, around 15% have used sensor placements around the thigh, equivalent to a pants front pocket placement, and 8% have tried the wrist. The majority of the studies (66%) targeted the area of the waist/trunk.

While a trunk/waist placement of accelerometers is known to provide data that can better model a fall, sensors placements on the trunk tend to generate discomfort to elders and suffer from a high risk of not being used in practice [5]. Indeed, [6] shows that many users of the push button based system have fallen and the alarm was not triggered because they were not wearing the system. In order to increase the usage of fall detectors it is important to comply with known user requirements. Requirements from elderly users drawn on [7], [8] show that fall detection devices must not stigmatize them nor disturb their daily life. Trunk/waist based commercial sensors would need first to become cheap and seamlessly integrated (such as into clothing) for sustaining such requirements. Therefore, there is a need to look further into the usage of other sensors and placements.

The above mentioned elderly users requirements motivated us to build a system using devices in which the user is likely to incorporate on his daily usage without introducing any burden or generating any sort of stigmatization. We decided to use a system composed of a standard off-the-shelf smartphone (to be carried in the user's pocket, as many users would do) and a smartwatch. Both are common consumer devices that would not trigger negative stigmatization.

II. Related Research

When it comes to the usage of smartphones and/or watches for fall detection, there has been much more research on algorithms and implementations on phones than watches or wrist-placed sensors. As presented in [9] there are more than 50 different scientific publications developing smartphone-based fall detection systems. On the other hand, when it comes to watches and wrist based sensors, this number is reduced to 8 [4]. From those 8, only [10] and [11] used a system based on a device with a real watch form factor, while others mainly attached sensors (which were not necessarily usable outside of the testing context) to the wrist region.

From within the articles testing wrist placement, both [12] and [5] performed fall tests using both wrist and other placements of sensors together. [5] looked into the sensitivity of a few algorithms for different placements for forward, backward and lateral falls. It did not look into their performance in activities of daily living (ADLs) and the percentage of false positives nor on how the sensors could complement each other.

[12] examined 5 different sensor placements, including the wrist and the thigh. However it mainly observed the data from the different placements as to describe different falls in terms of different sequence of patterns captured by the different sensors in a conjunct. As a consequence of that, there was limited insight on how much a wrist sensor could complement a thigh sensor. However, it was able to indicate that the wrist can help identifying sitting situations and slow falls. On top of that, together with [13] and [14], it has shown that merging the sensing data from different placements has led to an increased sensitivity of detection of falls.

That being said, this is the first, to our knowledge, published work specifically looking at an automatic fall detection system composed exclusively of two sensors located as one on the thigh region (in the front pocket) and one on the wrist.

As being the first study to test such set-up, it contributes to answering the following research questions:

1) Which level of fall detection can we achieve using a mobile phone together with a smart watch?
2) What are the effects of adding the sensor reading from the watch to a phone based fall detection system? Can it help reducing false positives?

In the next section we describe the implementation of the system illustrating the choices of algorithms used, how do they fit the system architecture and the hardware used on the development and tests. In the Evaluation section, we describe the different test cases, the accuracy of the system per test case, and how it compares with existing systems. Then, we conclude with the analysis of the test results and suggestions of future work for further expanding the findings of this research.

III. The fall detection system

The system was designed so that the phone would be the core as it is the most powerful device in terms of computing capabilities and battery. Being that said, the phone is responsible for retrieving its own sensors readings and the ones from the watch, and for analyzing the data as to decide whether there is a fall or not. As it will be later detailed in the Fall Algorithm subsection, such analysis is based on three different approaches combined (two based on the smartphone sensors data and one on the smartwatch).

A. Hardware and baseline software

The hardware used in this work consisted on the Android Wear smartwatch LG G Watch R[1] and the Android smartphone Samsung Galaxy S3[2]. Both devices have a 9-axis motion sensor combining a 3-axis gyroscope, 3-axis accelerometer and 3-axis compass and can communicate to each other via Bluetooth low energy (BLE).

The adoption of Android by the fall detection researchers is very positive as it allows other researches to more easily get hold of fall detection systems and experiment with them. However, as discussed in [9] almost none of the apps described on other fall detection research have been released to the public, and, as far as our knowledge goes, none of them have been released as open source and published in a publicly accessible repository. This hinders the possibilities for others to compare and verify the findings. Therefore, we decided to release the source code, its documentation and a more thoughtful description of the tests as open source in github: https://github.com/SINTEF-SIT/project_gravity

B. Fall Algorithm

The main goal of this research is to build a fall detection system which takes advantage of the motion sensors both on a smartphone and a smartwatch, assess its accuracy and the benefit of integrating the watch as part of the system. Since there were no published open source Android fall detection applications, we had first to choose and implement a fall algorithm both on the smartphone and on the watch.

Fall detection strategies may differ in terms of the types of movement they try to identify (as part of the identification of a fall) or the method used to detect them: fixed thresholds, acceleration patterns, fuzzy logic and artificial intelligence (AI) methods. A fall can be modeled as staged event consisting of essentially 5 phases [15] (where the three intermediary phases are highlighted and illustrated through the acceleration amplitude of one of our test cases in Figure 1):

- **Pre-fall phase**: an initial activity of daily living of which the subject is engaged in
- **Free-fall phase**: the free fall movement towards the ground derived from the loose of balance
- **Impact phase**: the impact representing the moment in which the subject touches the surface
- **Post-impact phase**: the instant after the impact when the person lies inactive on the floor due to the shock of the fall
- **Recovery phase**: the subjects effort to stand-up or recover from the fall

For this research, we have implemented both a threshold based and a pattern recognition algorithm in both the watch

[1] http://www.lg.com/global/gwatch/index.html#specification
[2] http://www.samsung.com/global/galaxys3/

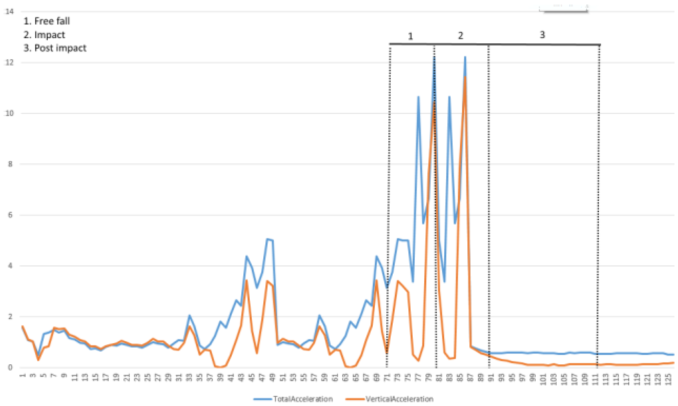

Fig. 1. Fall phases represented on acceleration graphs

and the phone. Threshold based algorithms use one or more formulas against a threshold that can represent a phase of a fall (usually both free-fall and impact phases). It is the simplest solution for detecting a fall and it can be used together with some form of pattern recognition in order to be more reliable. Threshold based is the most commonly used method for fall detection [4] and it has been previously implemented and well described in other research articles targeting Android platforms [9], [16]. We followed the lead of those articles and narrowed down our selection to 3 different threshold formulas:

1) Total vectorial acceleration: This is the same formula used to calculate the Euclidean vector distance, but used for calculating the total acceleration at one point in time regardless of its direction. Due to its simplicity, it is the most used formula for detecting falls [17], but it is rarely used alone. It uses the acceleration values in *x*-, *y*- and *z*- axis.

$$| A_T | = \sqrt{| A_x |^2 + | A_y |^2 + | A_z |^2} \quad (1)$$

2) Fall Index: It is based on the sums of the difference in acceleration in all directions between adjacent data points until the *j*-th previous sample. As the formula relies on the use of adjacent samples, it requires a high sampling frequency for being efficient.

$$FI_i = \sqrt{\sum_{k=x,y,z} \sum_{i-j}^{i} ((A_k)_i - (A_k)_{i-1})^2} \quad (2)$$

3) Absolute vertical acceleration: This formula calculates the total acceleration in vertical direction. Considering that a fall contains a large vertical acceleration there should not be much difference between this value and the value of the total acceleration in a fall.

$$| A_v | = | A_x sin\Theta_z + A_y sin\Theta_y - A_z cos\Theta_y \cdot cos\Theta_z | \quad (3)$$

On the other hand, pattern recognition algorithms use a series of the movement readings patterns and compare with databases or knowledge gathered from training sets. Pattern recognition is often implemented to detect fall stages and differentiate fall events from ADLs. The sensor data collected after a suspicious acceleration threshold can be used to distinguish if a person is lying still after high acceleration (which characterizes a post-fall phase) or if there is continuous movement (which can indicate that he is engaged in other ADL). Therefore, it is able to provide an extra insight in the fall detection on top of a threshold approach. However, pattern recognition algorithms use more memory and are heavier to compute for the device.

The final fall detection strategy implemented on the system has been to have the phone as the main device for the fall detection as its placement, the thigh, is more favorable for identifying falls. The strategy consisted on using three different approaches for detecting the fall in this presented order:

1) Phone Acceleration Threshold (PAT)
2) Phone Pattern Recognition (PPR)
3) Watch Threshold and Pattern Recognition (WTPR)

PAT is based on the monitoring of the acceleration threshold on the phone and is the only approach running continuously as it is the lightest one in terms of processing power and it does not require exchanging data between the devices. The monitoring is based on a sampling of 25 Hz, where the readings are used in formula 1 and 3 and compared with theirs respective thresholds A_{Tt} and A_{vt}. If both accelerations exceed their set thresholds, the algorithm will compare the two acceleration values by calculating the quotient of the division of A_{Tt} over A_{vt} and check it against a third threshold: A_{rt}. This is to find out how much of the total acceleration is vertical. Such comparison helps to eliminate false positives due to ADLs involving non-vertical acceleration such as walking or running.

If the three acceleration thresholds are exceeded, the PPR will be executed. The PPR uses the phone sensed data to identify the free-fall, impact and post-impact phases and it was based on an initial collection of training sensing data and fall patterns described in the literature. The PPR first finds the sample with highest total acceleration (A_{TH}) within a 2 seconds dataset of sensor readings. Then, it checks whether there has been a sudden increase in acceleration prior to it and a sudden decrease after it. The sudden increase is detected when the ratio between A_{TH} and the lowest total acceleration(A_{TL}), in a 0,4 second prior to A_{TH}, is over the free-fall threshold T_{ff}.The sudden decrease is calculated on the same fashion as the increase, but considering the lowest total acceleration(A_{TL}) within 0,4 seconds after A_{TH}, and by comparing it with a specific impact ratio threshold T_i. If both ratios are over their thresholds, the phases are detected and the algorithm analyzes the average acceleration pattern from the impact for 0,8 seconds. If the acceleration samples during that period are under the post-impact threshold T_{pi}, it indicates that the person is lying fairly still, corresponding to the post-fall phase. The identification of those fall stages in their sequence triggers the algorithm to detect the event as a fall.

If a fall is confirmed by the PPR, the watch data is pulled so that the watch threshold and pattern recognition algorithm is executed. Due to a limitation on the watch for providing reliable orientation data, we based its threshold detections on the Fall Index formula. We use the formula to identify if the difference of acceleration on the impact phase is above a threshold F_{Ti} and if the difference of acceleration during the post-impact phase is under the threshold F_{Tpi}. The watch data is also used to detect a short resting pattern from the post-impact phase. This is done by checking that for all the samples within 0,8 seconds after the impact, the difference of acceleration to the power of two of each pair is lower than the watch's post-impact threshold T_{pi}.

IV. Evaluation

Due to the prototype nature of our system, we decided to use a set of different intentional falls types and ADLs in order to assess the accuracy of the system. [17] shows that more than 70 different types of ADLs and more than 40 different types of falls have been tested across different studies. In order to define our set of fall and ADL patterns to be tested, we used the cases from [5] as a basis, since it is a widely cited publication and it includes the most used test cases reported in the review study [17]. Based on that, our test case consisted the 12 fall patterns and 7 ADL patterns below.

1) Fall Test Cases:

1) **Normal fall**: faint fall forward with knee bent
2) **Step down from platform**: Step down off a platform and fall forward in the process of stepping down
3) **Self tripping**: Walking tripping on owns foot
4) **Falling backwards**: Faint fall backwards with a round back and knee bent
5) **Falling backwards, failing to sit down**: Backward sitting-on-empty on the floor, no arm use, no step back
6) **Falling backwards against wall**: Backward fall at the base of a wall
7) **Falling left**: Faint fall left with knee bent
8) **Falling backwards/left**: Fall backwards and turning to the left side
9) **Falling sideways, left landing on wall**: Side fall to the left landing at the base of a wall
10) **Falling backwards/right**: Fall backwards and turning to the right side
11) **Falling sideways, right landing on wall**: Side fall to the right landing at the base of a wall
12) **Falling from a sitting position**: Falling of a chair, sitting on edge and slipping of

2) ADL Test Cases:

1) **Walking** (at least 10 meters)
2) **Turning around**: Quickly turn (spin) 360 degrees
3) **Sitting down slowly**
4) **Sitting down quickly**
5) **Tying shoes**: Crouching (going on the knees) and tying shoes
6) **Stairs**: Going down and up stairs (at least 10 steps each way)
7) **Jogging** (at least 10 meters)

The tests were performed by three different subjects (S1, S2 and S3) with the set of age/weight/height as per Table I using the watch around their left wrist and the phone on their pants left/front pocket. S1 was instructed to fall as naturally as possible in a 55mm soft mattress placed on top of a hard 35mm martial arts mattress. The following subjects reviewed the recordings of S1's tests and used those as guidelines for falling and performing the ADLs. Both S2 and S3 falls were performed in a different surface: a stack of three 10mm Airex fitline mattresses [3].

TABLE I. Subjects Characteristics

	S1	S2	S3
Age	22	26	32
Weight	75 kg	80 kg	63 kg
Height	185 cm	185 cm	170 cm

No additional instructions were given to any subject regarding the movements of the arms as there are no published guidelines on the engagement of the subjects' arms during fall tests using wrist sensors. However, in order to reduce the bias, we compared the recordings of the test cases performed using the watch with recordings of the same test cases without the watch (which were previously performed to gather training sensor data for the phone algorithm) as to check that the hands movements were not unnatural during the tests with the watch.

The thresholds described in the Table II were used in the system in order to assess its accuracy. Those values were based on the analysis of a training dataset generated when testing the system correctness and the values used by articles referred in this work.

TABLE II. Thresholds values used during the tests

PAT	PPR	WTPR
$A_{Tt} = 11m/s^2$	$T_{ff} = 1.5$	$F_{Ti} = 30m/s^2$
$A_{vt} = 9m/s^2$	$T_i = 1.5$	$F_{Tpi} = 20m/s^2$
$A_{rt} = 0.5$	$T_{pi} = 6m/s^2$	$T_{pi} = 170(m/s^2)^2$

It was difficult to find a suitable benchmark to compare with our results properly. As described earlier, different studies have used distinct test cases and, on top of that, they described the accuracy of theirs systems in different ways. Most studies we have read just describe the total sensitivity and specificity instead of providing those numbers per test case. The difference on test cases together with the lack of details on the results, make it impossible to establish a fair comparison without some bias. In order to also avoid any implementation error bias, we tried to find benchmarks which could be used as off-the-shelf software products.

For that reason, we chose to benchmark our tests against iFall [18] and Fade[4]. iFall is the only publicly available Android fall application we found whose fall strategy is documented, and it has been already tuned and used as benchmark by other researchers [9]. Fade does not have its fall detection strategy published, but it was the one who performed the best in a small battery of tests among the five top ranked applications on Google's Playstore.

Fade's sensibility can be switched between High, Medium and Low: where High is recommended to vulnerable people such as elderly, Medium is recommended to others engaged in regular ADLs and Low is recommended to those in moving vehicles. For our tests, we have used the High setting as the context of our fall detection would be detecting elderly falls. iFall has many thresholds which can be configured. We first tried the default settings, but the phone would trigger the

[3]https://www.my-airex.com/en/products/detail/2/fitline-180

[4]http://fade.iter.es/

TABLE III. EVENT DETECTION PERFORMANCE PER APPLICATION AND PER TEST CASE

Test Case	IFall	Fade	Our System
Fall 1	100%	56%	56%
Fall 2	67%	33%	89%
Fall 3	67%	67%	89%
Fall 4	100%	56%	89%
Fall 5	0%	56%	44%
Fall 6	33%	22%	56%
Fall 7	0%	11%	33%
Fall 8	0%	44%	56%
Fall 9	33%	11%	67%
Fall 10	0%	44%	78%
Fall 11	33%	33%	44%
Fall 12	67%	44%	56%
ADL 1	100%	78%	89%
ADL 2	100%	89%	89%
ADL 3	100%	100%	78%
ADL 4	67%	89%	89%
ADL 5	100%	100%	67%
ADL 6	100%	44%	100%
ADL 7	100%	67%	33%
Sensitivity	42%	40%	63%
Specificity	95%	81%	78%
Accuracy	61%	55%	68%

alarm at any movement. When we set the lower and the upper thresholds to 1G and 3.5G respectively (based on [9] and [18]) the app was much less sensitive. With those settings, although it had little sensitivity, it managed to trigger a couple of falls of S1. However, when S2 and S3 tested iFall, the phone did not trigger a single alarm in any test case. Despite double checking iFall settings and reinstalling the app, we could not restore its sensitivity or understand why it did not detect any falls.

The comparison of the test results between different applications can be found on Table III. It shows the accuracy of falls and ADL detection per application for each test case and the results in terms of sensitivity, specificity and accuracy (ratio of correct detection based on both falls and ADLs) for all applications. It takes in consideration the tests of all the subjects for all the apps, except for iFall where we just considered the tests of S1. If we had considered the tests of all subjects for iFall, its sensitivity would have dropped to 16%.

Our system implementation was such that PAT was the main algorithm to detect falls. Then, PPR and WTPR would come in only if PAT had detected a fall. PPR and WTPR would be used with the intention of ruling out ADLs. In order to save battery resources, WTPR would also just be consulted once PPR also confirmed the fall, however during the tests we decided to always consult PPR and WTPR in order to clearly outline the effects of each one of those algorithms in the results. This meant that we could distinguish between the cases where: all algorithms agreed on the fall, PAT did not detect the fall (and the other algorithms were not consulted), PPR did not detect the fall (but WTPR did), WTPR did not detect the fall (but PPR did) and both WTPR and PPR did not detect the fall (although this last case did not happen).

Table IV summarizes the distribution of our test cases based on what has been described above. The "'ALL FALL"' column matches the cases where all algorithms detected the test as a fall (a true positive for the fall cases and a false positive for the ADL cases). The columns with an algorithm name corresponds to the cases where that algorithm did not detect a fall out of the event (meaning that it falsified a true fall or detected an ADL).

TABLE IV. TESTS PER APPLICATION

Test Case	ALL FALL	PAT	PPR	WTPR
Fall 1	56%	22%	22%	0%
Fall 2	78%	0%	22%	0%
Fall 3	78%	11%	11%	0%
Fall 4	89%	0%	11%	0%
Fall 5	56%	44%	0%	0%
Fall 6	44%	11%	44%	0%
Fall 7	44%	33%	22%	0%
Fall 8	67%	33%	0%	0%
Fall 9	78%	22%	0%	0%
Fall 10	67%	22%	11%	0%
Fall 11	44%	33%	22%	0%
Fall 12	56%	33%	11%	0%
ADL 1	11%	78%	11%	0%
ADL 2	11%	56%	11%	22%
ADL 3	22%	78%	0%	0%
ADL 4	11%	78%	11%	0%
ADL 5	33%	67%	0%	0%
ADL 6	0%	78%	11%	11%
ADL 7	67%	0%	33%	0%

The test cases were recorded both in video and as accelerometer data. They were published at the project's public Github repository in order to easily enable others to reproduce them, test different thresholds and strategies or use them as benchmark.

V. CONCLUSION AND FUTURE WORK

The tests done so far have been useful for providing an initial assessment of such a system and helping to identify possible points of improvements. When it comes to the accuracy of the system as whole, the performance was below the ones reported in other research works (such as [16] who managed to reach an accuracy of around 90%), but above the systems we managed to get hold of and test. This indicates that it should be possible to improve the system's accuracy while it also highlights the importance of releasing the software and detailed testing information for future usage by other researchers.

The results of the fall cases show that, with the current thresholds, the PAT if used alone would have been able to detect 77,75% of the falls. Since our strategy is such that the PAT serves as a trigger for the other algorithms, it's thresholds must be relaxed in order to allow further increase of the sensitivity of the system. At the same time, the pattern recognition strategy needs to be tuned. As it is, it helped in detecting 40% of the ADLs which the PAT would have considered as falls, being 12% detected by WTPR and 28% by PPR. However that came at the cost of nullifying the correct detection of 19% of the falls detected by the PAT, where all those were mistakenly identified as ADLs by the PPR. Those numbers indicate that the watch, in the system as it is, could be able to marginally increase the specificity without affecting the sensitivity. However, the system would still have a high rate of false positives and the current contribution would be based on both the PAT and the used thresholds, as the PAT functions as the trigger to the other algorithms. Contrary to [12], the watch helped in the identification of a turning ADL instead of the sitting ones, but that is likely due to the fact that they have used posture recognition and we did little pattern recognition with the watch data.

Our strategy has been to use different algorithms in conjunction and many thresholds making it difficult to find the most suitable combination of thresholds. For that reason, we plan to shift the current boolean logic used to converge the outputs of the different algorithms towards a fuzzy logic approach. By translating the acceleration related thresholds into fuzzy values, we should be able to translate the outputs of the algorithms into the actual confidence of their detection and therefore reach more robust thresholds. The usage of fuzzy logic has been effective in other research works [19], [20] merging sensed data for fall detection and it should be even more appropriate to our case as we deal with even more candidate thresholds.

In total, we performed 171 tests with each system. Although this number was enough to initially assess the accuracy of the system and the possibilities of improvements around the different thresholds, it definitively needs to be expanded. Indeed additional tests are planned in order to help us investigate further the different algorithms performance on a per test case basis and to come up with the fuzzy thresholds. Most of the published articles we have read (such as [9], [16]) used around 500 or more test samples to build and test theirs thresholds and strategies. Besides conducting more tests ourselves, we hope that the open publication of the data and the videos illustrating the fall cases from this work can boost the development of a more representative database of falls on smartphones and smartwatches.

Still on the testing, it would be interesting to study the algorithms performance for other typical phone placements such as in a jacket, purse, back pocket or even when the user is talking on the phone or interacting with its keypad. [21] has started this work by finding thresholds based on the total vectorial acceleration resulting on around 80% accuracy.

The future work related to test cases should also include test subjects which are more representative and ideally real falls. Both the usage of young subjects [22] and simulated fall [23] introduce some bias in the results as they lack precision on representing real elder falls. However, capturing real-world falls is very difficult as it needs a large number of subjects running the system for a long time in order to collect enough data [23]. Such real-world trials are not feasible with the accuracy of the system at the moment, but when the accuracy is improved it would be realistic to consider a wide real world trial due to the large adoption of Android devices and the ease of distributing the app globally through Appstores.

ACKNOWLEDGMENTS

The work behind this study was possible thanks to the support from the EU-funded FP7 project OPTET under grant agreement no.317631 and the ADAPT project funded by the Research Council of Norway under the program FRIMEDBIO (grant 230435).

REFERENCES

[1] World Health Organization. Ageing and Life Course Unit, *WHO global report on falls prevention in older age*. World Health Organization, 2008.

[2] M. E. Tinetti, W.-L. Liu, and E. B. Claus, "Predictors and prognosis of inability to get up after falls among elderly persons," *Jama*, vol. 269, no. 1, pp. 65–70, 1993.

[3] R. J. Gurley, N. Lum, M. Sande, B. Lo, and M. H. Katz, "Persons found in their homes helpless or dead," *New England Journal of Medicine*, vol. 334, no. 26, pp. 1710–1716, 1996.

[4] L. Schwickert, C. Becker, U. Lindemann, C. Maréchal, a. Bourke, L. Chiari, J. L. Helbostad, W. Zijlstra, K. Aminian, C. Todd, S. Bandinelli, and J. Klenk, "Fall detection with body-worn sensors: a systematic review." *Zeitschrift für Gerontologie und Geriatrie*, vol. 46, no. 8, pp. 706–19, Dec. 2013.

[5] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers." *Gait & posture*, vol. 28, no. 2, pp. 285–91, Aug. 2008.

[6] B. Heinbüchner, M. Hautzinger, C. Becker, and K. Pfeiffer, "Satisfaction and use of personal emergency response systems," *Zeitschrift fur Gerontologie und Geriatrie*, vol. 43, no. 2, pp. 219–223, 2010.

[7] R. Steele, A. Lo, C. Secombe, and Y. K. Wong, "Elderly persons' perception and acceptance of using wireless sensor networks to assist healthcare." *International journal of medical informatics*, vol. 78, no. 12, pp. 788–801, Dec. 2009.

[8] N. Holliday, "Fall detectors what do users want ?" Health Design & Technology Institute,Coventry University Technology Park, Tech. Rep., 2012.

[9] R. Luque, E. Casilari, M.-J. Morón, and G. Redondo, "Comparison and Characterization of Android-Based Fall Detection Systems," *Sensors*, vol. 14, no. 10, pp. 18 543–18 574, Jan. 2014.

[10] T. Degen, H. Jaeckel, M. Rufer, and S. Wyss, "SPEEDY: a fall detector in a wrist watch," *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.*, pp. 184–187.

[11] T.-H. Tan, M. Gochoo, C.-S. Chang, C.-T. Wu, and J. Y. Chiang, "Fall detection for elderly persons using android-based platform," *Energy*, vol. 2, no. 2, p. 2, 2013.

[12] Q. Li and J. A. Stankovic, "Grammar-based, posture- and context-cognitive detection for falls with different activity levels," in *Proceedings of the 2nd Conference on Wireless Health - WH '11*, 2011.

[13] O. Aziz and S. N. Robinovitch, "An analysis of the accuracy of wearable sensors for classifying the causes of falls in humans," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 19, no. 6, pp. 670–676, 2011.

[14] H. Gjoreski, M. Luštrek, and M. Gams, "Accelerometer placement for posture recognition and fall detection," *Proceedings - 2011 7th International Conference on Intelligent Environments, IE 2011*, pp. 47–54, 2011.

[15] N. Noury, P. Rumeau, a. K. Bourke, G. ÓLaighin, and J. E. Lundy, "A proposal for the classification and evaluation of fall detectors," *Irbm*, vol. 29, no. 6, pp. 340–349, 2008.

[16] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan, "PerFallD: A pervasive fall detection system using mobile phones," *Proc. of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops*, vol. 10, pp. 292–297, 2010.

[17] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "Automatic fall monitoring: a review." *Sensors*, vol. 14, no. 7, pp. 12 900–36, Jan. 2014.

[18] F. Sposaro and G. Tyson, "iFall: An android application for fall monitoring and response," *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*, pp. 6119–6122, 2009.

[19] C. Dinh and M. Struck, "A new real-time fall detection approach using fuzzy logic and a neural network," *Proceedings of the 6th International Workshop on Wearable, Micro, and Nano Technologies for Personalized Health*, pp. 57–60, 2009.

[20] P. Boissy, S. Choquette, M. Hamel, and N. Noury, "User-based motion sensing and fuzzy logic for automated fall detection in older adults." *Telemedicine journal and e-health*, vol. 13, no. 6, pp. 683–693, 2007.

[21] V. Viet and D.-j. Choi, "Fall Detection with Smart Phone Sensor," *Proceedings of the 3rd International Conference on Internet (ICONI)*, pp. 15–19, Dec. 2011.

[22] J. Klenk, C. Becker, F. Lieken, S. Nicolai, W. Maetzler, W. Alt, W. Zijlstra, J. Hausdorff, R. Van Lummel, L. Chiari *et al.*, "Comparison of acceleration signals of simulated and real-world backward falls," *Medical engineering & physics*, vol. 33, no. 3, pp. 368–373, 2011.

[23] F. Bagalà, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, and J. Klenk, "Evaluation of accelerometer-based fall detection algorithms on real-world falls." *PloS one*, vol. 7, no. 5, p. e37062, Jan. 2012.