Location-independent Fall Detection with Smartphone

Stefan Mehner and Ronny Klauck and Hartmut Koenig
Brandenburg University of Technology Cottbus
Computer Science Department
03013 Cottbus, Germany
Email: {mehner, rklauck, koenig}@informatik.tu-cottbus.de

ABSTRACT

Due to demographic changes in developed industrial countries and a better medical care system, the number of elderly people who still live in their home environment is rapidly growing because there they feel more comfortable and independent as in a clinical environment or in a residential care home. The elderly often live alone and receive only irregular visits. Due to impaired physical skills the probability of falls significantly increases. The detection of falls is a crucial aspect in the care of elderly. Falls are often detected very late with severe consequential damages. There are existing approaches for automatic fall detection. They usually deploy special external devices. Elderly people often do not accept these devices because they expose their frailty. In this paper, we present a location-independent fall detection method implemented as a smartphone application for an inconspicuous use in nearly every situation of the daily life. The difficulty of our approach is in the low resolution range of integrated acceleration sensors and the limited energy supply of the smartphone. As solution, we apply a modular threshold-based algorithm which uses the acceleration sensor with moderate energy consumption. Its fall detection rate is in the average of current relevant research.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Health

General Terms

Experimentation, Performance

Keywords

Fall Detection, Smartphone, Accelerometer

1. MOTIVATION

In developed industrial countries the number of elderly people age of 70 and beyond who still live in their home environment is rapidly growing due to demographic changes and a better medical care system. Elderly people feel more comfortable and independent there as in a clinical environment. Often the elderly live alone and only irregularly receive visits. With age, physical ailments, such as decreased eyesight, limitations of the musculoskeletal and the cardiovascular system, and impaired physical skills occur, which significantly increases the likelihood to fall [10]. According to [9], 30% of over-65s fall at least once a year. When no one is around, the fall is not noticed by others.Sometimes the fallen people are forced to lie on the ground for several hours, waiting for help. In the worst case, the person dies before anyone can get, help or suffers severe physical and mental damages due to late rescue. If medical help fast arrives on site, the injuries quickly can be treated and consequential damages will be reduced. This benefits the patients, and relieves the healthcare system. But even if the fall causes no serious injuries, many elderly people cannot stand up alone due to their physical weakness and have to wait helpless on the ground for rescue. Sohe fear of further falls increases. Meanwhile, a range of organizations, e.g., the Red Cross and the Johanniter in Germany, offer an emergency call system. Here, the person endangered to fall is equipped with an emergency button which can be pressed when falling within a limited area. If the elderly person knocks the head, he/she is probably unconscious and not able to press the emergency button. Therefore, the automatic detection of falls has become an important research direction to independently trigger emergency calls without the help of the fallen person. There are a number of investigations, e.g., [2],[4] and [8], which use acceleration sensors to measures the acceleration of the person in the x-, y- and z-axis. These values are compared with threshold values which indicate fall situations to distinguish between normal movements and a fall. Most of the elderly people have a problem to constantly wear such a device on the body. The additional device restricts the persons in their normal life. For example, it must be repositioned when changing clothes. In addition, it can be seen that this device is used for medical purposes. Elderly people often do not accept this because it exposes their frailty. Therefore, the approach of this work is to investigate how smartphones can be deployed for fall detection because they are inconspicuous, can be used almost everywhere and more and more people, even elderly ones, wear them constantly. Moreover, an acceleration sensor is already installed in all of those phones. The original purpose is to control games and the alignment of the screen. When a fall is detected, an emergency notification can be triggered via SMS, call, or other services (twitter, e-mail, facebook). The fallen person

PETRA'13 May 29 - 31, 2013, Island of Rhodes, Greece

http://dx.doi.org/10.1145/2504335.2504346

Intrinsic	Extrinsic
- Age (> 65) - Dementia - Diseases - Motoric limitations	- Incorrect use of clothes - Drugs
Home environment	**Outside home environment**
- Slippery carpet - Stairs - Higher ground objects	- Broken roads - Places with lots of people - Inadequate lighting

Table 1: Fall risks

can be located by GPS, which allows further use cases like climbing, hiking or other activities.

Central issues of the approach are the resolution range of smartphone sensors and their energy storage. Sensors in smartphones have significantly lower resolution than the sensors of the aforementioned approaches. This makes it more difficult to distinguish between falls and normal movements. The battery life of a smartphone in normal use is about one day. If a fall detection application additionally runs, all the energy reserve may quickly be exhausted. Therefore, the algorithm should use as little energy as possible. The solution we have developed is a modular threshold-based algorithm which uses the acceleration sensor and requires little energy. To support a broad application we use Android - the most broadly used smartphone operating system - is used as implementation platform.

The remainder of the paper is structured as follows. Section 2 considers the reasons for falls and their characteristics. Thereafter, a short overview of the various approaches applied for fall detection is given. In Section 3 we discuss the constraints of using smartphones for fall detection. Section 4 describes the applied fall detection algorithm and the used parameter setting. Section 5 reports on our experimental studies on the detection capability of the algorithm and its energy consumption.

2. FALLS AND THEIR DETECTION

A fall is an unexpected event in which a person comes to lie on the floor or a lower level [13]. There are various reasons for falls. Everybody stumbles sometimes and if the body is fit the balance can be re-established by a lunge. Elderly people are less able to do so due to impaired physical skills. Therefore, a tripping over the carpet edge can already lead to a fall. Usually falls occur during daily life activities. In most cases the reason is a loss of balance, e.g., caused by dizziness. Fewer falls occur during unusual activities, such as climbing a ladder or standing on a chair. It can be assumed that people endangered to fall avoid such situations as much as possible, and therefore the percentage is so low. Age is not the only risk for a fall. There are various risk factors (see Table 2) [1]. The studies also show that 78% of the fallen people meet four or more of these risk factors. The probability of falling again increases through dwindling confidence in one's own body or by injuries from the last fall. There are different approaches and classifications for detecting falls. Noury [16] differentiates between analytical methods and machine learning, whereas Mubashir [14] distinguishes between active and passive fall detection. Analytical methods measure changes in the position of the human body after falling using active and passive methods [16]. Therefore, the various phases are investigated. Machine learning methods apply self-learning algorithms which try to optimize certain parameters based on statistical fall data. The detection accuracy depends on the training data. This, however, might not correspond to real-life situations. Taking real-life data is not appropriate, since falls are comparatively seldom [16].

Passive fall detection methods do not require that persons wear special equipment on their body. Emergency situations are detected indirectly from the environment using, for instance, cameras or mats. So in [7] a camera is used to determine a typical posture after a fall. To train the algorithm 20 positions with 256 images were required. Another approach [17] tries to recognize the distance between the head and the ground. If the average position of the head is significantly different from the original one, a fall is assumed. A detection rate of approximately 96% is reported. Another possibility for the passive detection of falls is the use of special mats, e.g, [6]. These mats are put on the ground. They have a specific number of capacitive proximity sensors per square meters. The implemented detection algorithm can identify people lying on the ground. The applied algorithm uses threshold values. Based on the contiguous area on which pressure is exerted, it can distinguish between a normal tread and a fall. The disadvantage of passive fall detection methods is the relatively high cost for cameras and pressure mats because they have to be installed in each monitored room. The detection area is limited to the equipped rooms. In addition, elderly may feel observed in their privacy by the surveillance cameras. The advantage of the passive methods is that the algorithms work well and no special devices have to be placed on the body, so that the person can move freely inside the monitored area.

Active fall detection methods use sensors which are placed on the body of the monitored person, e.g., on the chest and/or on the waist, to reproduce the movements of a person and to detect a fall situation [2], [4], [8]. These approaches usually use an acceleration sensor, which measures the acceleration in the x-, y-and z-axis. The magnitude of these vectors represents the total acceleration of the device and thus that of the person who wears it. These values are compared with threshold values which represent different phases of a fall to distinguish between normal movements and falls. Elderly people usually do not like to wear such a device on the body because it restricts them in their normal life. For example, it has to be repositioned when changing clothes and it is apparent that these devices are used for medical purposes. The acceptance rate of these devices is low. For this reason, we investigate to what extent a smartphone can be deployed for an inconspicuous and location-independent fall detection as argued in the introduction of this paper.

3. USING SMARTPHONES FOR FALL DETECTION

Smartphones have become a central communication means of our live. Everywhere it is possible to communicate and access the internet. To provide easy use and convenient features to the user, smartphones are equipped with a lot of technology. GPS sensors are used to locate the device. They are used for navigation applications. Light sensors on the front recognize whether it is light outside. They adjust the screen brightness and darken the screen when the device is held to the ear. Nowadays smartphones also contain other sensors which can be used for fall detection. All smartphones since 2009 include at least a three-axis accelerometer. Figure 1 depicts the structure of such a capacitive sensor. The

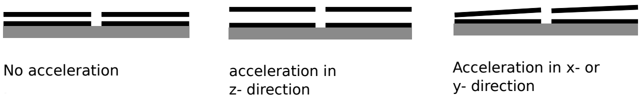

Figure 1: Structure of an accelerometer

test mass contains four electrodes. If there is acceleration in the direction of the z-axis all electrodes perform the same capacity change, since the test mass moves up or down (Figure 1 middle). An acceleration in the x- or y-axis declines the proof mass (Figure 1 right), which is registered by changing the capacities of two electrodes in a positive or negative voltage value. The gravitational acceleration of 1 g (=9.81 m/s^2) towards the ground always acts on the sensor; this allows to determine the location of the sensor in rest position. In smartphones, this sensor is positioned such that the axes are aligned as shown in Figure 2. If the device lies on a table with the display to the top, the x- and y-axes have an acceleration of 0 g, the z-axis of 1 g. Primary application of these sensors is to recognize whether the smartphone is held horizontally or vertically to adjust the screen orientation. A central issue in detecting falls with a smartphone

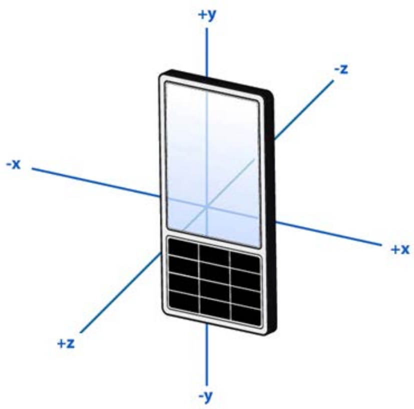

Figure 2: Alignment of the axes (source [15])

is the significantly lower resolution range. Sensors used in the above mentioned specialized approaches have a resolution range between ±6 g and ±16 g, while smartphones' ones only reach ±2 g. To automatically detecting falls it is necessary to distinguish falls from activities of daily living (ADL), such as running, sitting, climbing stairs, bending, etc., which show similarities in their movement patterns [18]. Figure 3 represents various activities of daily living which we measured with a sensor located at the waist of the test person. The black line represents the sum vector of all axes, i.e. the total acceleration. Falls have an impact on the ground which may shortly between 4 g and 6 g. The intensities of the peaks while activities like climbing stairs or running are similar to that of falls, since treading on the ground causes strong accelerations, although the maximum deflection of the activities rarely exceeds 3 g. Another difference between

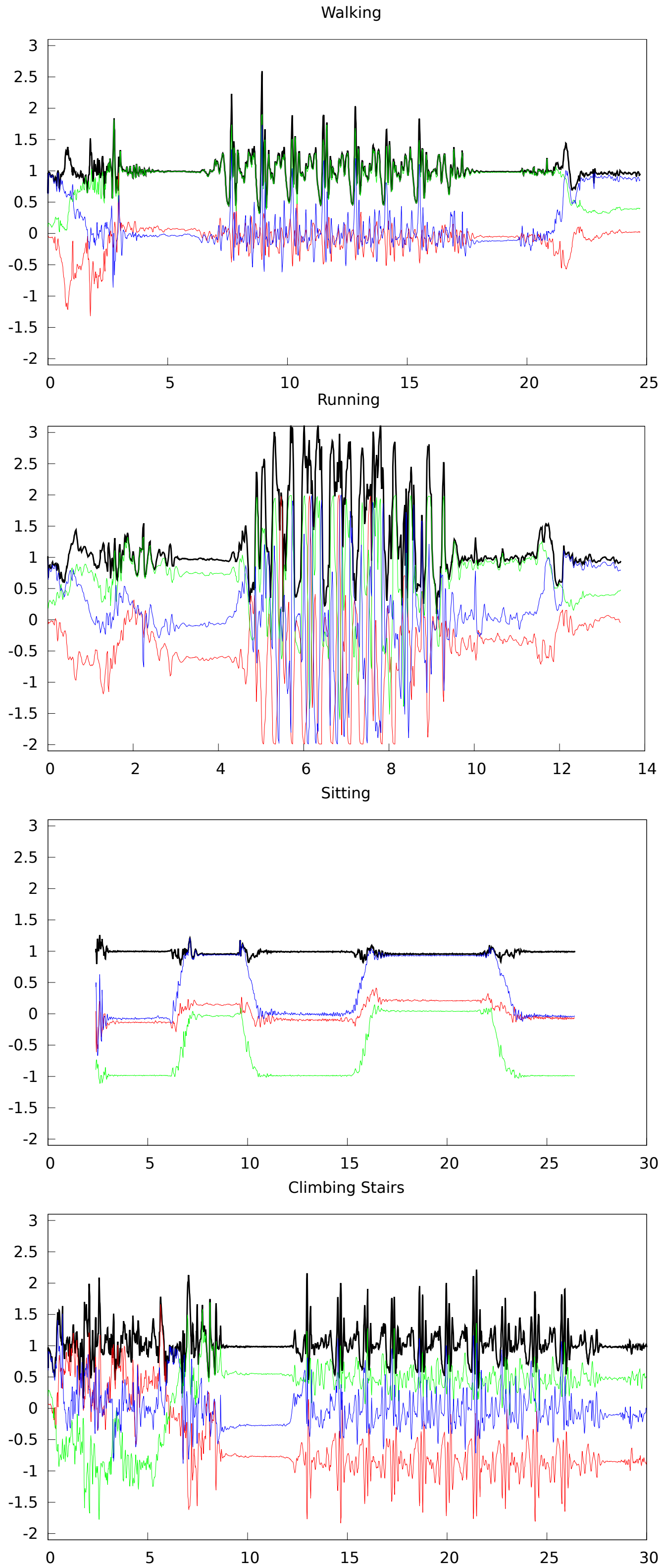

Figure 3: Activities of daily living recorded with an accelerometer at the waist

	Samsung Galaxy S	Sony Xperia Ray
Release date	06/2010	09/2011
CPU	1 GHz	1 GHz
Available RAM	329 MB	512 MB
Android	2.3.3	4.0.3
Battery capacity	1200 mAh	1500 mAh
Accelerometer	Bosch BMA023	Bosch BMA150
- Resolution	± 2 g	± 4 g
- Sampling Rate	80 Hz	90 Hz

Table 2: Specification of the smartphones

ADL and a fall is the lack of periodicity. After a single hard impact there is a rest period in which the fallen person remains on the ground. Fall detection algorithms can use this to differentiate from ADL by taking the rest phase after the high deflection of the sum vector of the individual axes into account (see Section 4.1). This also helps to avoid false positives.

In order to estimate the difference between ADL and a fall we measured various ADL movements, namely walking, running, climbing stairs and sitting. First we performed these activities with one person and three smartphones. Every activity was repeated 10 times. In the second study, we examined fall situations with three young test persons. Every person had three smartphones and performed every fall (forwards, sideways and backwards) 10 times. The devices were positioned always at the waist because a smartphone usually can be expected to be in the trouser pocket. Another important criteria is the update frequency of the sensor data. In consideration of the fact that a fall takes only 1 to 2 seconds, the frequency must be high enough to notice an emergency situation at all, and to distinguish the phases of a fall (see below). A frequency of 50 Hz should be sufficient to precisely map this small time interval. Table 2 shows the specifications of the devices we used in the tests (two devices were of the same model).

A weak point of smartphones is the battery performance. Usually it has to be recharged after one or two days. For this reason, a machine learning solution is not feasible for smartphones because the energy supply is exhausted quickly by the learning algorithm. A threshold-based approach, which compares the current sensor data with different threshold values, is less complex. Therefore, fewer computations are required which helps to save battery power. With active application the smartphone should persevere with active application one day to allow the useful deployment. Users must get accustomed to recharge the smartphone in the evening when taking their medication. Thus, it can be ensured that there is always enough power available.

4. FALL DETECTION ALGORITHM

We implemented our fall detection algorithm as an Android application. The decision to adopt Android as the development platform was made due to the widespread use of this smartphone operating system. In 2012 it had a market share of more than 50%. Note that many vendors provide smartphones running this operating system. Sometimes the hardware differs substantially in the various models. When performing the same measurements, the expected results may differ depending on the quality of the installed sensors.

When the fall detection application is installed, the user can input an emergency message and a phone number of the person to be called in an emergency case. Furthermore, the threshold parameters and timers can be adjusted. After starting the app, the sensor values are constantly measured and analyzed to detect a fall situation. Additionally, the data is stored in a file on the smartphone memory card to analyze it on the computer for further studies.

When a fall is recognized, an alert message appears on the smartphone display. Furthermore, an alert sound informs the bearer that a fall was detected. If this message was triggered by an anomalous movement, i.e. a false positive, he/she can cancel the sending of the emergency message. If the fallen person does not react in an interval of 60 seconds the emergency call is triggered.

Smartphones only have a very limited battery capacity. For this reason it is necessary to apply a relative simple fall detection algorithm. As argued in Section 3, we decided to use a threshold-based algorithm which takes the typical phases of a fall *free fall*, *impact*, *post impact* and *stability* as depicted in Figure 4, into account. Below we explain in detail how these phases are passed through during fall detection. In our approach we can only use algorithms which use existing smartphone sensors because changes in the hardware are not possible. For this purpose, we have obtained an overview of the sensors used in smartphones. All device classes contain at least one acceleration sensor. As it is suitable for fall detection, our algorithm was implemented for such a sensor.

The algorithm follows the principle of the approaches of Jia [8], Kangas [11], and Karth [12] who developed active fall detection algorithms for specialized devices. These algorithms analyze the fall phases *free fall*, *impact*, *stability*, and *orientation*. We noticed that a fallen person is not motionless on the ground immediately after an impact. For this reason, we introduced, in contrast to the aforementioned approaches, an additional phase called post impact, which is waiting a short time after an impact for a more precise fall detection (see Fig. 4). To aggregate the accelerometer values we use the sum vector (SV) of all three axis, similar to Chen [4]. This value represents the total acceleration of the device. The SV is calculated as follows:

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

4.1 Setting of Thresholds and Timers

At the beginning we took the same values for thresholds and timers as in the aforementioned approaches. Then we have adjusted them according to our experimental studies. Due to the limitation of the sensor resolution, further adjustments were necessary. Figure 5 shows how we set the thresholds. They are discussed in detail next.

Free Fall Threshold.

This threshold indicates when a free fall has been detected. In static position of the body the acceleration of the smartphone is about 1 g. During a free fall the sensor (theoretically) measures no acceleration. In reality this is not possible because of the air resistance. There is a substantial difference between the static position and the free fall. Kangas and Jia assumed values of 0.6 and 0.75 g, respectively, while Karth used 0.5625 g to reduce false positives. In our studies we achieved the best results with 0.5625 g too.

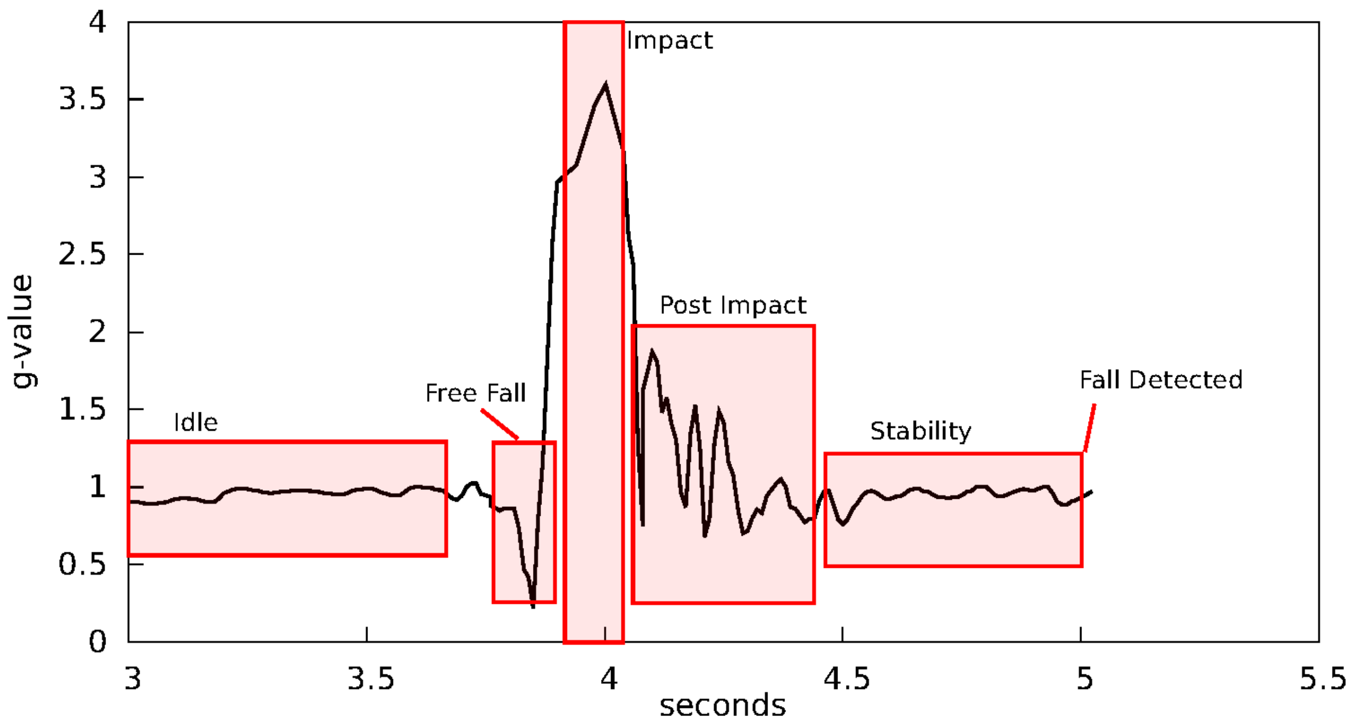

Figure 4: Phases of fall detection

Impact Threshold.
The impact lasts a very short time with a peak value for the SV. Depending on the used sensor, it is possible to set up a value of 6 g or more [3] to distinguish ADL and a fall. The resolution of a smartphone sensor, however, is not designed for this. Most smartphones only can resolve ± 2 g per axis. Our empirical studies proved a value of 2.3 g as the best one.

Motionless Threshold.
In the aforementioned approaches a value of 0.1875 g was set for this threshold. That value should not be exceeded to recognize the stable phase after the impact. We set a value to 0.4 g because it showed that with 0.1875 g even the slightest movement exceeded the threshold. An unconscious person, however, can have some movement when, for instance, the body slips on the ground. For this reason, we chose a higher tolerance.

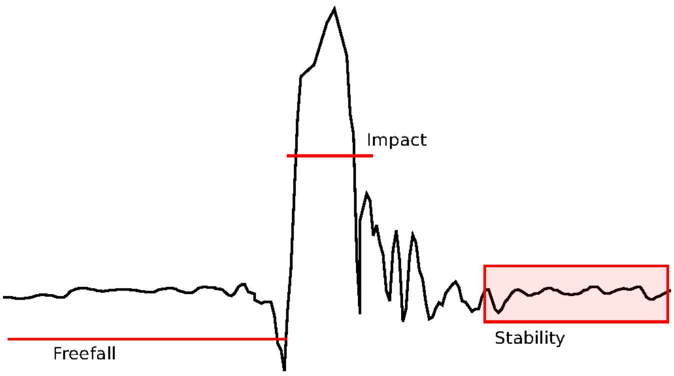

Figure 5: Thresholds

The second important aspect for initializing the algorithm is to determine the duration of the single fall phases. We set the timers as follows.

Free Fall Time.
This is the time interval in which the person is in free fall. Here we used 30 ms like Karth and Jia.

Impact Time.
The duration of the impact is differently estimated in the aforementioned approaches. Kangas chose a value of one second; Jia and Karth assumed 300 ms. Considering the fact that a fall usually takes less than a second, we also set this time interval to 300 ms.

Post Impact Time.
As argued above, we noticed that there is a short time interval between impact and stability in which the fallen person is dazed. According to our observations, this period is around 1 second long. It is important for the fall detection because the body is not suddenly motionless after the initial impact on the ground. The ground absorbs some of the acceleration, so that there are a few smaller impacts directly after the first impact. By the use of this interval, it can be assured that the stability check is not incorrectly interrupted by such subsequent impacts.

Motionless Time.
The stable phase introduced by Jia has a duration of 2 seconds. In this interval it is monitored whether the person is lying on the ground motionless or moving. According to Jia, there is a short period after any impact in which the person is still lying, even if he/she is conscious. If the person

moves during this time, probably it was not a fall. This circumstance has a high importance for the fall detection with smartphones because running produces similar peak values as the impact after a fall. In contrast to running, a fall situation has no further movements after a hard impact. In our work we set up the stable phase to 1 s. The value is short in comparison to Jia, but we additionally have introduced the *post impact* period before the stable phase. The whole interval should not be too long. Table 3 summarizes the applied threshold and timer settings.

Parameter	Threshold
Free Fall Threshold	0.56 g
Impact Threshold	2.3 g
Motionless Threshold	0.4 g
Free Fall Time	30 ms
Impact Time	300 ms
Post Impact Time	1000 ms
Motionless Time	1000 ms

Table 3: Thresholds and timers

4.2 Phases of Fall Detection

After discussing the parameter setting of our algorithm, we now discuss the different phases of the fall detection algorithm in detail (see Figure 4). The algorithm is presented as a state machine diagram in Figure 6.

Idle State.

This is the normal state where the smartphone is on and the application runs. The application waits for changes in the SV. Without any movement the SV is about 1 g. When SV falls below the *free fall threshold* $= 0.5625\,\text{g}$, a free fall is assumed.

Free Fall.

This state marks the beginning of each fall. If this event is not recognized, it is not possible to detect a fall, even if all further phases are recognized correctly. Within the interval *free fall time* $= 30\,\text{ms}$ the SV should exceed the *free fall threshold*. If this happens an impact is expected. Otherwise the fall detection is stopped and the algorithm returns to the initial state. However, it is possible to skip the free fall check (as shown in Figure 6), if there is an insufficient detection of free fall events. The algorithm then begins with checking, whether the *impact threshold* is exceeded.

Impact.

After the free fall phase the algorithm changes into the impact state in which the algorithm expects the peak sum vector of the fall. If there is no impact within the defined time window, the algorithm returns to the idle state. It is not sufficient to only look for the impact because the movement has to be distinguished from ADL situations like climbing up stairs or running that can produce similar movement patterns. This is the reason why we also analyze the stable phase.

Post Impact.

After exceeding the *impact threshold* the algorithm waits 1 s. As shown above, we introduced this interval in our algorithm because the body is still in motion for a short moment after the fall. Without this phase, a continuous motion would be detected and the stable phase tolerance zone would be exceeded. The advantage of this additional waiting time is that the body can pass into the rest position first before the stable phase looks for further motions.

Stability.

A person that has fallen does not get up immediately [8]. Instead, the person is motionless for a moment. The duration for *motionless time* is one second. Within this interval, the algorithm checks, whether the sum vector (SV) is in the tolerance zone of $1\,\text{g} \pm$ *motionless threshold* $= 0.4\,\text{g}$. This tolerance is necessary because the sensor can have a small noise as previously described and some minor shocks may occur by the person. It allows to distinguish a fall from an ADL movement, such as running. When running, the body is in continuous motion, while it steadily lies on the ground after a fall. At the end of the stability phase there is an *orientation check* that averages the last 100 sensor values before the end of the stable phase and compares it with the position before the impact.

Fall Detected.

After one second without exceeding the tolerance zone and if the *orientation check* was positive, a fall is detected. A message appears on the display of the smartphone. If the message is not cancelled within the next 60 s an emergency message is triggered to the previously specified number because we assume that the fallen person is unconscious and cannot respond.

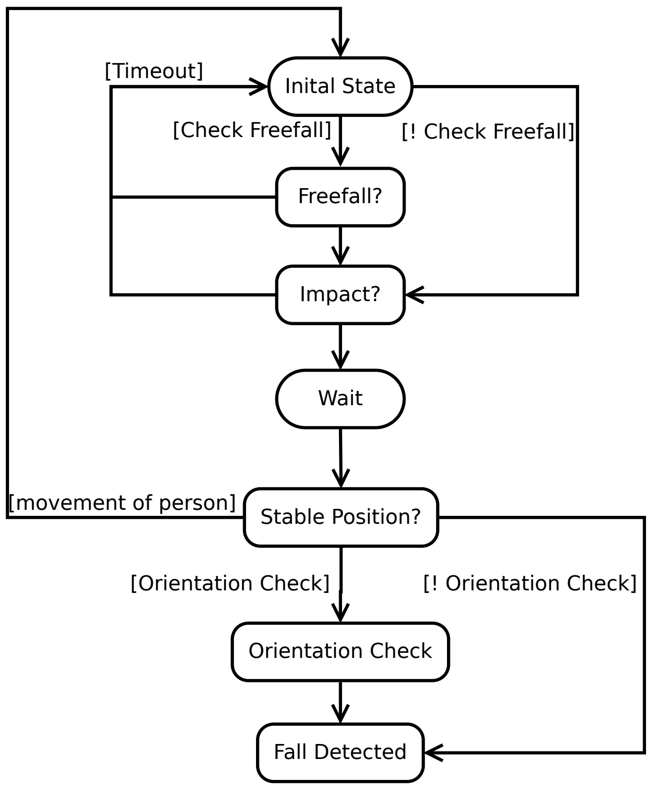

Figure 6: State Machine Diagram for the fall detection algorithm

5. EXPERIMENTS

To evaluate the practicality of smartphones for the detection of falls we run a series of experiments. The goal of the experiments was to determine the detection rate and the false positive rate. In addition, the energy consumption of the algorithm was investigated. To allow a meaningful use, the battery life should be at least one day. Another objective of our experiments was to find the optimal setting for our algorithm. It has a modular structure which permits to take different phases into account. The investigated settings are listed in Table 4.

Phase	Setting 1	Setting 2	Setting 3	Setting 4
Free Fall	X	-	X	-
Impact	X	X	X	X
Post Impact	X	X	X	X
Stability	X	X	X	X
Orientation	-	-	X	X

Table 4: Overview of the possible settings

5.1 Fall Detection Studies

For evaluating the quality of our algorithm, we determined the sensitivity and specificity. The former is a measure for the correctly detected falls of a measurement series. According to [16], it is calculated as follows:

$$Sensitivity = \frac{TruePositive}{TruePositive \ + \ FalseNegative}$$

True positives are correctly recognized falls, while false positives denote movements that were identified incorrectly as a fall. Unlike the sensitivity, the specificity measures the number of correctly identified true negatives, i.e., no fall occurred and the device also has not identified a fall. It indicates whether the algorithm generates many false alarms. The sensitivity is determined through:

$$Specifity = \frac{TrueNegative}{TrueNegative \ + \ FalsePositive}$$

In our test series, we measured falls in different directions: forward, backward, and sideway. Three people participated in these tests. They fell 10 times in each direction and for each setting. They bored 4 smartphones. In total, we performed 120 measurements per studied setting. Furthermore, the algorithm should not issue an alarm if the person did not fall. This can be checked through performing ADLs. The measurements of these activities were performed by one person with three smartphones, as described in Section 3. Thirty measurements were recorded per ADL, which should be sufficient for the evaluation of the error rate. The results of the test series for the forward fall show that the settings which skip the free fall provided the best results. Theoretically, setting 2 should provide a better detection rate than setting 4 because this setting is not as restrictive as setting 4. The problem is the imitation of a correct fall through a test person, because the body instinctively tries to intercept falls by using the arms, which affects the measurements. Therefore, it can be assumed that setting 2 and 4 will have similar results in the everyday use. With activated free fall check, only 50% of the falls in setting 1 and 42% in setting 2 were detected. Again, the issue here is the imitation of a correct fall because real-life falls happen in a very short time interval and are jerkier compared to imitated ones. Therefore, we omitted settings 1 and 2 in the following test runs because they caused false positives for ADL situations and the results of the first fall series were not sufficient for future investigations. Comparing the remaining settings in the following fall series, setting 4 scored the best results in all examined tests. The three studied fall situations have similar detection rates. Regarding the average of all falls, setting 4 correctly detected about 83% of the situations, setting 3 only 56%. To sum up, setting 4 proved to be the best, which skips the component *free fall* and checks the alignment after the fall. It reached sensitivity of 83.33% and a specificity of 100%. These results are acceptable in view of the fact that sensors in smartphones were not originally designed for this purpose.

	Setting 1	Setting 2	Setting 3	Setting 4
Walking	80%	50%	100%	100%
Running	70%	55%	100%	100%
Sit Down	100%	100%	100%	100%
Stairs	80%	60%	100%	100%
Ø ADL	**83%**	**66%**	**100%**	**100%**
Fall forward	50%	77%	42%	87%
Fall sideways	-	-	58%	86%
Fall back	-	-	69%	78%
Ø Falls	**50%**	**77%**	**56%**	**83%**

Table 5: Summary of the ADL and fall studies

5.2 Power Consumption

The power consumption of our fall application was measured with the tool Battery Monitor Widget. It presents the current charging of the smartphone battery in mAh. As the test device we used the popular Android smartphone Samsung Galaxy S (see Table 2), which is equipped with a battery capacity of 1200 mAh. For the calculation of the power consumption in a certain period, we measured the battery capacity at the beginning and at the end of each test run. To validate the results of the first measurement a second test run was performed with the same conditions. Each test run took one hour. The power consumption was evaluated for all settings (see Table 4) of daily living activities and simulated falls. In the first case, the smartphone was carried during daily activities to measure the discharging of the battery. In the second case, the smartphone was in static position at the desk 15 min and then a fall was simulated. In one hour we performed 4 simulated falls. Reference value for all measurements was the power consumption in idle mode. The smartphone always was in flight mode to eliminate other influences on the power consumption, such as active wireless connections. The display was switched off during all test runs. The results of our measurements are summarized in Table 6. It summarizes the average power consumption in percent for the various ADL settings and simulated falls.

The results show that the fall application has nearly the same power consumption in daily activities and with activated fall detection in comparison with the power consumption at rest. The various settings of the fall application have only a marginal influence on the power consumption. The reason for is the principle of how the fall detection algorithm works. As fall situations are unpredictable, the algorithm always has to wait for the free fall, and in the settings 2 and

4 for the expected impact. The average power consumption per hour of our fall application is around 8%.

	Setting 1	Setting 2	Setting 3	Setting 4
Idle	7.50%	8.04%	7.00%	7.00%
ADL	7.54%	7.67%	7.59%	7.42%
Fall	7.83%	7.25%	7.42%	7.00%

Table 6: Share of the fall detection on the power consumption (time interval 1 hour)

6. CONCLUSIONS

In this paper we have presented a fall detection algorithm for use in smartphones. The algorithm has been implemented as an app for Android smartphones. As smartphones require very energy-efficient applications, we use a threshold-based approach which requires fewer computations. Unlike other threshold-based algorithms, we additionally take the time between impact and stability into account for improving the detection precision. Crucial problems for smartphone-based fall detection are the limited resolution of the accelerometers and the power constraints of the batteries. Our algorithm takes this into account. Our tests showed that it reaches a detection rate of 83.33%, and a specificity of 100%. These results are acceptable considering the fact that the sensors originally were not designed for this purpose. A study of the energy consumption showed that the battery life is approximately 12 to 14 hours with activated fall detection application, which is sufficient to guarantee the functionality for the whole day.

The motivation for our approach has been that more and more people reach an old age, with the result that the likelihood of falls increases due to physical ailments. Perceiving early fall situations can help that aid comes quickly, so that subsequent damages are reduced. There are various systems for fall detection on the market. These are based on wearing special devices to indicate the fall. Elderly people often do not accept these devices because they restrict them in their normal life and expose their frailty. As more and more people including elderly ones wear a smartphone for a longer time of the day, a lot of these restrictions can be eliminated. Besides accelerometers, many smartphones also contain gyroscopes which can be also used to improve the detection capabilities of smartphones. Therefore, our location-independent fall detection smartphone application is also of interest for younger people to signal accidents in sport activities, such as running, skiing, hiking, biking, and others.

In a next step we plan to equip elderly people with smartphones with our fall detection application to record their motions and probable falls, to optimize our parameter settings with the *Fall Detection Simulator* of Fudickar [5].

7. REFERENCES

[1] S. Abbate et al. Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor network: A survey. pages 147–166, December 2010.

[2] A. K. Bourke et al. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait Posture*, 26(2):194–199, July 2007.

[3] A. K. Bourke et al. Evaluation of waist-mounted tri-axial accelerometer based fall-detection algorithms during scripted and continuous unscripted activities. *Journal of Biomechanics*, 43(15):3051–3057, November 2010.

[4] J. Chen et al. Wearable sensors for reliable fall detection. *Conference Proceedings - IEEE Engineering in Medicine and Biology Society*, 4:3551–3554, 2005.

[5] S. Fudickar et al. 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, New York, NY, USA, 2012. ACM.

[6] Future Shape. Sensfloor. http://www.future-shape.de/de/technologies/11/sensfloor, January 2013.

[7] Z. Z. Htike et al. A monocular view-invariant fall detection system for the elderly in assisted home environments. *Intelligent Environments*, pages 40–46, 2011.

[8] N. Jia. Fall detection application by using 3-axis accelerometer adxl345. http://www.analog.com/static/imported-files/application_notes/AN-1023.pdf, 2009.

[9] I. Joint Commission Resources. *Reducing the Risk of Falls in Your Health Care Organizaton*. Improving health care quality and safety. Joint Commission Resources, 2005.

[10] I. S. Joint Commission Resources. *Reducing the Risk of Patient Harm Resulting from Falls*. Joint Commission Resources, 2008.

[11] M. Kangas et al. Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait Posture*, 28(2):285–291, August 2008.

[12] C. Karth. Fusion of sensor data for robust fall detection in assisted living. Diploma thesis, University of Potsdam, Germany, 2012. (in German).

[13] S. Lord et al. *Falls in Older People: Risk Factors and Strategies for Prevention*. Cambridge University Press, 2007.

[14] M. Mubashir et al. A survey on fall detection: Principles and approaches. *Neurocomputing*, 2012.

[15] Nokia. Sensors. http://doc.qt.nokia.com/qtmobility/sensors-api.html, January 2013.

[16] N. Noury et al. Fall detection - principles and methods. *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 1663–1666, 2007.

[17] M. Shoaib, R. Dragon, and J. Ostermann. View-invariant fall detection for elderly in real home environment. *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium*, pages 52–57, 2010.

[18] M. Warner et al. *The Complete Guide to Alzheimer's-Proofing Your Home*. Purdue University Press, 2000.