MOBILE ALERT: COMBINING HUMAN MOTION DETECTION AND VOICE ANALYSIS

Telma Cristóvão de Oliveira

Instituto Superior Técnico
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
telma.oliveira@tecnico.ulisboa.pt

ABSTRACT

In developed countries societies are getting older. Due to better health care and better overall conditions the average life time of an individual tends to increase. The fraction of elderly population also tends to increase. These societies need to adapt to this reality by providing elder citizens ways to assure a certain quality of life and that can extend their autonomy. Taking into account the huge growth of the smartphones’ market, chances are that most of the people who already have a mobile phone will have a smartphone in the future if they don’t own one already. This project aims to take advantage of what is already present in everyday life like a smartphone, and exploit the multiple built-in sensors to try to create a mobile monitoring system. This study aims to assess the possibility of having a monitoring system that detects both fall and screams in a distress situation for an elderly user. This study also aims to put into perspective how would such system work.

Index Terms— monitoring system, elderly, scream detection, fall detection

1. INTRODUCTION

In developed countries’ societies, due to several factors such as better healthcare services or better general life conditions, the average life expectancy of individuals tends to increase. This means that there is a global tendency for this kind of societies to become aged. Therefore, the fraction of the total population under these conditions will grow in the future.

According to Eurostat in 2011, the average number of individuals living in the European Union with ages over 65 years old represented 17.5% of the total population, and individuals over 50 years old represented roughly 50% of the total population — see Figure 1. Also, according to Eurostat projections, these numbers tend to slightly increase in the next 5 decades.

When comparing European statistics to similar numbers compiled by the USA government in 2010, one can see that only 32% of the USA’s total population is above 50 years old and 13% is over 65. USA and European numbers on the 65+ population are similar, but when comparing to Japan we can see that the Japanese society is aging even faster. In Japan, in 2010, the group of people over 65 years old already represented 23% of the total population, with an enormous tendency to grow, according to the Japanese government projections. This shows that aging is a very concerning subject, and it has deep economic and social repercussions [1] [2] [3].

The degradation of human physical characteristics with ageing, such as balance or motor skills, can lead to cases of falls among older people. When a person lives alone, some of these falls, depending on the victim’s overall condition and physical fragilities, may lead to episodes where the person is injured and cannot get up to ask for help, or may lose consciousness without anyone knowing what is going on.

Let us assume that despite living alone these individuals have family, friends or neighbours who worry about their wellbeing and can be trusted to ask for help when needed. We’ll call them caretakers. In such scenarios as those described above, the caretaker might never be warned in time to assist the person in need and in more tragic situations help won’t be there within a reasonable period of time.

Monitoring falls is a way to keep caretakers aware of what’s going on with their elders and in case of emergency, a more prompt assistance action can be taken.

This project aims to study the possibility of a monitoring system solution that can help caretakers understand if their elders are in a risky situation due to a fall event. The proposed system relies on a smartphone device, since these gadgets are not too expensive and are expanding their market. Through fall detection, complemented with a scream detection method, the proposed solution’s main goal is to understand if it is possible to correctly identify a fall event, supporting that assessment with a voice analysis process applied on an audio sample collected through the microphone of the mobile device. If a scream is detected then the caretaker should be warned, i.e., a distress call should be performed.

There are two major goals that are proposed by this work. One of them is to study a suitable method that can provide a correct classification of recorded sound as human voice by
the means of a scream. That sound is recorded from a mobile device inside a pocket. The second one is to implement an adaptive fall algorithm resilient to changes of movement.

Voice analysis has been subject of deep study for a long time under multiple circumstances. This work, performs the classification of sound captured from a mobile device inside a pocket as human voice and recognizes pitch to identify human voice in a very noisy and variable environment without any control of the noise sources.

From another perspective, this work also contributes with a proposed fall detection algorithm, based on a two threshold algorithm, to increase the system resilience to false positives while changing the type of locomotion. The algorithm adjusts its thresholds to more suitable values for walking and running. The adjustments are made in an adaptive way without user interface for pre-defined profiles. The consulted literature – see [4], [5], [6], [7], [8] – studied fall detection algorithms minding solely one kind of motion at a time, i.e. check if an individual falls while walking or while running, for example. What about an individual that is walking, afterwards decides to run and then falls? Literature algorithms studied weren’t tested for motion shift resilience. The adaptive changes in thresholds minding natural variations in motion is the novelty this work brings to present literature.

2. STATE OF THE ART

Nowadays there are multiple services and solutions to monitor the elderly population, assessing their wellbeing and sending alerts in emergency situations to their caretakers. These solutions serve multiple purposes and monitor different aspects, all related with health.

Even though Europe is getting older faster than USA, nowadays USA provides more solutions, particularly as services, offered by multiple companies providing monitoring systems. Several services were studied, like Life Station, Bay Alarm Medical, ADT Medical and Phillips Lifeline [9]. All of them offer a device connected to the phone line and an electronic wristband that communicates with that device wirelessly. These gadgets typically have an emergency button, a microphone and a loud speaker embedded. In dangerous situations the user can push the emergency button existing in either one of the devices and make a phone call to the call centre associated with the service provider. Once the emergency case is stated, the call centre operator evaluates the event, and calls either the national emergency service or a family member or caretaker. These systems should present a low complexity user interface, making it easy to use and, in some cases, the monitoring range goes up to a 300 meters radius. This kind of service is paid having annual fees and extra expenses like device activation fees, which may vary from service to service.

ADT Medical and Bay Alarm Medical differentiate from the rest because they provide passive monitoring also. The user may require that carbon monoxide monitoring and fire/smoke monitoring to be performed and connected to the service provider monitoring centre. ADT Medical also provides the installation of burglar alarms to be connected to their monitoring centre. ADT also offers an integrated service that contains burglar alarm, fire alarm, carbon monoxide, temperature and flood monitoring – see Figure 5. In this case there is a sensor network that requires a more complex set up. There is a main device similar to a regular burglar alarm that is capable of correctly identify the problem. If a sensor triggers, that information will be sent by Wi-Fi to ADT monitoring centres where the information is processed and an assistant will get the information. This assistant will contact the person whose house is being monitored. The assistant will contact the police, fire department or other emergency personnel they will be dispatched to the person’s home.

Video monitoring is another common solution when to maintain the safety of an elder person. This kind of monitoring helps to evaluate the amount of activity produced by the person being monitored. A very good example of this type of monitoring system is FamilyLink [10]. This product, for example, consists of a laptop that through a built-in web camera captures video images and has software that automatically understands if there was a change in scene, identifying movement if a change is detected. If no activity is monitored an alert text message or email will be sent to the caretaker. The alerts may be configured according to the elder’s routine. If there is no movement in the kitchen at 4 a.m. no alert will be sent, for example. This product offers an easy interface to keep the elder included in social networks and online communication with relatives and friends, which helps checking if the elder is well. However, the FamilyLink laptop needs to be connected to the internet in order to issue alert messages, it won’t identify situations of conscience loss and may raise privacy issues.

GPS tracking devices can help monitoring the elder person’s position allowing the caretaker to be informed of the elder person’s whereabouts at all instants. Some of these devices can be programmed with a delimited zone, called safe zone, and when the person being monitored gets out of this area the caretaker receives an alarm to inform about the event. These alarms can be received by text message or e-mail. Despite the usefulness of these gadgets the state of unconsciousness is difficult to monitor outdoors as they cannot monitor a person’s activity inside a building due to technology limitation.

Smartphones, even low cost solutions, have several built-in sensors such as gyroscope, compass, accelerometer, proximity sensor, light sensor and barometer in some cases. It makes sense that one can make use of this hardware that is already present in that kind of devices to develop a monitoring system.
The development of a fall detection system should be a feasible goal and in fact there are already some solutions available for free at Google Play, the Android app market. Solutions for other operating systems such as Windows Phone or iOS were considered in the performed survey. No application regarding fall detection was found for Windows Phone and only one was found for iOS. Ten Android applications were collected and researched upon. The featured applications were tested in order to determine what kind of features were present in each. Manual Thresholds Tuning is a setting that allows the user to edit the default values of the thresholds required to detect a fall. This is a very relevant feature that can improve the fall detection accuracy. Location ID allows the application to access device’s location. Voice Analysis is a feature that enables the capture of audio clips through the device’s microphone. SMS Sent to Caretaker allows the application so send an SMS to predefined contacts set as emergency numbers when a fall occurs. Along with the SMS sent to the caretaker, some applications also gather the GPS coordinates of the place where the fall took place. One can see that none of the studied applications requires Internet Connection to function properly and all of them are available free of charge.

Besides the study of the features available in the applications, all of them were tested regarding the accuracy of decision. They were tested in three different situations: while jumping, running and falling. Through the results we can conclude that none of the applications are able to distinguish between any of these three types of movements. Most of them do not accurately detect a fall. Clearly there is an issue with the monitoring process. Under this scope this work aims to provide an improvement on the monitoring process by having an adaptive method resilient to movement changes. Other research performed included the analysis of already existing Fall Detection Techniques such as the Low-Complexity Fall Detection Algorithm and already existing Voice Detection Techniques such as Short-Time Energy-Based Voice Activity Detection, Feature extraction and classification and Pitch Detection.

3. PROPOSED SOLUTION

As stated before, this project’s goal, is to implement an Android application that can correctly detect a fall event and uses the audio input from the microphone to help validate the detected fall. The application will take advantage of some of the sensors available on the device and through the analysis of the input values provided by those sensors, a fall will be detected if it occurs. The application will use mainly the accelerometer readings, as the algorithm that will be used for fall detection only needs this kind of input. Regarding audio input, the device’s microphone will be used. The device is placed within a pocket waist/thigh high. The desired behaviour is that when a fall is detected an alarm message will be sent to a designated phone number.

Also a prompt window will appear after the fall is detected so that in case of a false positive or a not so serious fall the user is able to cancel the alarm message before sending it. After a fall, if a scream is detected, it is expected the fall validation to be faster. The timer will be ignored and the message will be sent right away. To implement this application it is necessary to define three essential steps: how to detect a fall, how to assess if an audio input is a scream and how to decrease the number of false positives.

3.1. Fall Detection

The algorithms present in the available solutions are either too stiff or too sensitive regarding fall detection. This means that most likely their thresholds are hardcoded and are not flexible. Three of the applications studied allowed the manual setting of thresholds, therefore a test subject can freely edit these thresholds in order to find the most suitable ones. This solution is not intuitive to the end user. Hardcoded thresholds are not suitable for accurate fall detection in most cases. Taking this into account this work proposes an adaptive algorithm meant for fall detection that requires less interaction from the user and self-adjusts to the type of movement that is happening at the moment. In order to accomplish these goals, the mobile application starts by requesting the accelerometer readings to the device operating system. This is an ongoing action from the start. After obtaining a certain number of samples the threshold calculation should be performed. As stated before, the threshold setting methodology is not very clear in the consulted literature, therefore the rule to determine threshold values will come from observation. From the sampled vector created in order to store the readings, the higher value will be set as the maximum threshold and the lower value of the sampled vector will be set as the minimum threshold value.

If the system detects a reading from where both those thresholds are surpassed one of two things might be happening: either the subject changed the way he/she was moving or a fall actually occurred. If the subject simply changed the type of movement, the system should perceive that change in order to lessen false positives. The change in the type of movement, as for example, from walking to running, can be identified through the periodicity of the movement. For instance, running won’t generate single thresholds exceeding peaks. The peaks would exceed once again the thresholds within a certain period of time. If an analysis is performed regarding this point of view it is expected that the error resilience of the system would increase. Figure 1 helps to understand the proposed process.
3.2. Scream Detection Relying on Pitch Analysis

In the context of this work audio samples were taken from mobile phones inside a pocket waist high. This represents a huge difficulty when aiming to perform scream detection. It is important to understand what a scream is in this context. As stated before, this scream might occur while the fall occurs. This scream is not a stressful scream like it could be under a more dangerous situation. It is most likely a sound slightly louder and higher pitched than the average speaking.

Since neither the VAD nor the feature extraction studied in the context of this work proved to be suitable approaches for this scenario the analysis of the pitch was another option available. Despite the extremely noisy samples, one can see that for the sentence used in Figure 2 it is possible to extract the pitch information.

The android application in its final form was envisioned to identify if a scream occurred after a fall had happened. Since the fall and the scream usually happen simultaneously, a valid fall detection couldn’t be used as a trigger for the voice analysis to start. To solve this issue periodic audio recordings should be performed. The latest periodic recording should be kept in storage. Since no information is provided at the start of the application regarding the user’s pitch or the conditions of the surrounding environment, it is necessary to proceed with periodic calibrations. The calibrations should access what is the pitch information from the saved recording. Like the fall detection proposed algorithm, from the pitch analysis on the saved recording the higher pitch value from the sample should be set as a threshold. When a scream occurs the pitch captured during the recording will be compared with the pitch value set as threshold. If it surpasses this threshold then the scream would be detected.

4. IMPLEMENTATION

Figure 4 represents the proposed system integration overview. The application reads the data from the accelerometer. This data is used by the fall detection module to extract the information on the acceleration over the user’s body. This acceleration readings are tri-axial. The acceleration information will be analysed in order to set thresholds and later to test the actual acceleration values against these thresholds. Simultaneously the application will also read the data stream captured by the device’s microphone, will encode it and save it to the local storage. When a fall is detected by the fall detector, the scream detector will access the local storage, load the input data and check if a scream was performed. If so, then a possible risky situation is in order and an emergency contact should be warned. There will be no scream detection actions performed if the fall detector doesn’t identify a fall.
4.1. Scream Detection Approaches

4.1.1 Voice Activity Detection

From the previous evidence it was concluded that to recognize speech under this conditions was not possible through the power spectrum information. A captured sample of a scream raised the possibility to develop a voice activity detection algorithm to identify only a scream. To achieve this, the use of a short-time energy based system was considered to analyse if human speech was present or not in the audio signal. Despite regular speech could not be identified through this method it was expected that a scream would, due to the fact that in most of the samples collected different subjects tended to use vowel like sounds to express a scream. These sounds are, for example, “ah!”. This approach did not work out well due to the fact that most of the samples collected had also noise overlapping the scream making the distinction very difficult to perform.

4.1.2 MFCC and GMM Approach

Another approach that was taken into account was to implement a classifier and build a machine learning algorithm. This algorithm should be trained from a data base of screams and afterwards it should be able to perform autonomous decisions. This algorithm would make use of MFCC for feature extraction and then an Expectation Maximization algorithm for likelihood weights. This wasn’t adequate as the captured segments were too noisy. Note that the noise in the samples is not added noise. It is real noise and therefore hardly treatable under the conditions stated since MFCC is little robust in these conditions. In addition to the noise, one must consider that the distance from the sound source (the user vocal system) to the microphone on the mobile device inside the pocket adds an extra attenuation to the sound captured.

4.1.3 Pitch Analysis Approach

Taking into account what was previously explained in subsection 3.4, periodic recordings should be performed every 5 seconds. The chuck of audio captured would be saved to the device’s storage. This file should be overwritten for storage management purposes at each capture. The first recording when the application starts should be analysed to find the maximum pitch value of that audio file to be set as threshold. If no fall occurs meanwhile this calibration process should happen every 30 minutes. If a fall is detected the pitch analysis of the audio file that holds the information of the 5 seconds prior to the fall detection should be triggered. In this case, a new audio file would be created to carry on with the periodic recordings while the older one would be kept until the analysis of the signal was finished. This constant recording is necessary since the scream usually occurs while falling, not after. It is also necessary to keep these small samples of prior timespans since no reasonable trigger exists to identify the exact moment where it is more likely for a scream to occur. To analyse the audio signal the techniques described in the subsection 2.3.4 should be applied. Since the pitch is something that will vary from person to person based on age and gender, the choice of the window to apply the short-time autocorrelation is a sensitive parameter since, as stated before, for high pitch speakers the window should be 5ms to 20ms and for low pitch speakers the windows should be about 20 – 50ms. In this case the window chosen would be of 25ms in order to provide a reasonably ranged solution. Having a 25ms window implies that from a 5 second audio file is possible to retrieve 200 samples, for each sample the autocorrelation would be applied and 200 pitch values would be extracted. From the 200 values, the maximum pitch value should be retrieved and compared to the previously set threshold. If it exceeds the threshold value then it is possible a scream happened. After the fall occurred and the audio signal’s been analyse, a resting period of 10 minutes should take place and after that a new calibration period should start.
4.2. Fall Detection Implementation

The fall detector reads data from the accelerometer and in the performed implementation obtains 20 samples that are kept in a vector. The lower peak and higher peak of the acceleration analysis are retrieved from that vector. Also the mean value of this vector is obtained. If the thresholds are not set yet these values will respectively be the lower threshold value and the upper threshold value and we will go back to the first task. If the thresholds were already set, the amplitude of the acceleration should be analysed. If both thresholds are surpassed we need to check if it is a periodic event. If so, thresholds should be updated. If not, it is likely that a fall had happened.

To determine the periodicity of the movement captured in order to assess if the thresholds should be updated or not, a vector with 20 samples is collected and the mean value of that vector is obtained. If the newly calculated mean value is twice as big as the previously calculated value, related to the first sample capture, then new thresholds are set. This will enable a change of movement from walking to running without a false positive for a fall.

On the other hand, the thresholds will also be updated if the newly calculated mean is half of the previously calculated mean. This will allow the user to change from running to walking adjusting the thresholds to be ready to detect a fall in a body with less acceleration at the present time.

While running the maximum and minimum values of the thresholds are usually higher when comparing to those obtained while walking. If the user changes the pace from running to walking and these thresholds are not reconfigured, then to detect a fall will be very difficult since the acceleration of the user has lowered significantly and possibly one of the thresholds will not be surpassed when a fall occurs.

The threshold update is very relevant to improve the accuracy of the decision when a fall happens.

5. EXPERIMENTAL RESULTS

After adding the fall detection implementation for an adaptive algorithm – see Figure 25 for the correspondent code – fall detection tests were performed.

Testing procedure was meant to test the following aspects: behaviour from walking to running, from running to walking, walking and then falling, running and then falling. Each behaviour had 10 recurrences.

In the scenario where the subject is running and then falls, the application never detected the fall occurred. This was due to threshold values. The maximum threshold value is usually quite high and the lower threshold value is quite low while running. This means the impacted caused by the fall wasn’t able to cause the thresholds to be surpassed.

In the scenario where the subject is walking and then falls, the application was able to correctly detect 8 out of 10 falls. This translates in a success rate of 80% versus a fail rate of 20%.

In the scenario where the subject is walking and the starts to run, the application was able to correctly assume that wasn’t a fall 6 out of 10 trials. This means for the performed test the success rate was 60% and the fail rate was 40%. Also the thresholds weren’t always updated correctly and that was one of the main factors that caused the application to misbehave.

At last, the scenario where the subject was running and then started to walk no fall was detected because no thresholds were supposed to be surpassed. The thresholds did not update correctly as they should have lowered to more suitable values regarding the new type of movement.

6. CONCLUSIONS

One of the biggest difficulties faced during this project was the amount of variables present under its scope. Regarding the fall detection, the non-adaptive algorithm used is quite robust for applications of this kind. When adding the adaptive behaviour the robustness should increase. It was observed that most of the times the application can correctly distinguish a movement change. Nonetheless the error rate is still quite high. Also it was observed that the thresholds didn’t update themselves when necessary and most classification errors derived from there.

On the other hand the scream detection was far from simple in its nature due to the condition of the audio samples taken. The scenarios where the monitoring system makes more sense to act are the outdoor ones. In this kind of scenarios the noise environment is quite unpredictable. Despite some calm environments are likely to occur, environments with multiple speech sources or extremely noisy are also probable, like riding the bus or going to a
market fair. These factors adding extra noise to the audio signal to be captured by a device inside a pocket makes the analysis quite difficult to perform.

Other factors like the kind of clothes the person is wearing are also relevant to consider when thinking about how those affect the obstruction for the sound capture. The conditions set in this work are not favourable to this kind of audio processing. This work culminated with the adaptation of the application named SenseAck which is now ready for adaptive recognition of a fall.

7. FUTURE WORK

7.1. Where to Improve

Despite the implementation of a full functional application was not successfully performed, it is possible to do it. Regarding the scream detector it is not that trivial to manipulate an audio stream using Android OS. If you manipulate an audio stream in real time, i.e., recorded and analysed at the same time is quite simple but to read that stream from a device’s storage unit presents more difficulties. It is necessary to specify a fixed sampling rate and format for the recording part in order to generate uniform audio samples. To be able to read those audio samples it is needed a decoder in the application. This decoder will be responsible for decoding those samples into raw data. After this raw data is available the manipulation should be easier. In order to implement the proposed algorithm this is the first step. After the implementation it is required testing, which presents another issue. To test the scream detector it is required a data base containing suitable audio samples, recorded by a subject having the mobile device inside a pocket waist high while falling. This is also not easily obtained because if the subject falls on purpose then the scream is less genuine. During this study some audio samples were collected and a small data base was created. They are usable for a first approach on testing the proposed scream detector but not that useful when it comes to improve the robustness. Another aspect that requires improvement is the resilience to false positives on the fall detection algorithm. Having an adaptive algorithm is something useful in far more areas than just this scenario. Some other features can be considered to improve the presented fall detection algorithm is the assessment of the angle between the body of the subject and the vertical in order to understand to which direction the subject fell: sideways, backward, forward. From a technical point of view no studies were performed during this work in order to understand the impact of this system in the smartphone power consumption. Since the accelerometer and the mic are constantly on demand when thinking about a final product is relevant to perform an analysis from this perspective.

7.2. Where to Improve

To successful implement the scream algorithm and correctly integrate it with the fall detection algorithm would be the major next step. From this point, having the system correctly working, it would be interesting to create an adequate interface and study what would be the acceptance of this system among a target group of users. The system developed under the scope of this work aimed to lessen the active interaction between the user and the technology. It was envisioned to be an almost autonomous system that wouldn’t demand the user to be highly technically educated. One next step in the continuation of this work would pass through this point. Ultimately, under different usage conditions, such as using the smartphone on a small bag around the waist guaranteeing to provide less obstruction and noise addition to the sound capture it would probably be possible to understand if a distress situation was occurring by recognizing key words like “help”. It would be interesting to provide an interface that required no other interaction besides speech to perform actions like calling someone through voice commands. This would be particularly useful in situations where the user could not move but didn’t lose conscience. The application scenarios of such system would be vast.

8. REFERENCES

Information Technology (ECTI-CON 2010), Chiang Mai, 2010.
