IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

FALL DETECTION SYSTEM USING SMART SHOE AND BAND

[1][2][3][4][5] Mrs. Nethra H L, Bharath C, Jayasurya B M ,Kadhar Ibrahim B, T Guru Sai Pavan
[1](Assistant Professor, Dayananda Sagar Academy of Technology and Management)
[2][3][4][5] Dayananda Sagar Academy of Technology and Management

Abstract: Injuries due to falls are among the leading causes of hospitalization in elderly persons, often resulting in a rapid decline in quality of life or death. Rapid response can improve the patient's outcome, but this is often lacking when the injured person lives alone and the nature of the injury complicates calling for help. This project presents an alert system for fall detection using common commercially available electronic devices to both detect the fall and alert authorities. We use an Android based smart phone with an integrated tri-axial accelerometer. Data from the accelerometer is evaluated with several threshold-based algorithms and position data to determine a fall. The threshold is adaptive based on user provided parameters such as: height, weight, and level of activity. The algorithm adapts to unique movements that a phone experiences as opposed to similar systems which require users to mount accelerometers to their chest or trunk. If a fall is suspected a notification is raised requiring the user's response. If the user does not respond, the system alerts pre-specified social contacts with an informational message via SMS. If a contact responds the system commits an audible notification, automatically connects, and enables the speakerphone. If a social contact confirms a fall, an appropriate emergency service is alerted. Our system provides a realizable, cost-effective solution to fall detection using a simple graphical interface while not overwhelming the user with uncomfortable sensors.

Chapter 1 Introduction

As age related changes in reaction time and balance reduce. The capabilities of people, the likelihood of a fall leading to significant injury increases. Not only are fall related injuries the number one reason for emergency room visits, they are also the leading cause of injury-related deaths among adults 65 years of age and older . Every year, more than 11million people fall. In 2005, unintentional falls accounted for an estimated 56,423 hospitalizations and 7,946 related deaths in the United States. Many of these deaths are a result of a "long-lie," an extended period of time where the victim remains immobile on the ground. Just the simple fear of a long-lie or falling can lead to the worsening of one's mental health, isolation, and the general degradation of his/her quality of living. Current systems are available that attempt to reduce the long-lie period by alerting emergency services when a fall has been detected. These systems commonly use one of three methods for classifying a fall:

• Acoustic/vibration recognition: This is achieved by having a device, usually implanted in the floor, monitor sound and other vibrations. It listens for the vibratory signature of a human fall, which is vastly different from the signatures of walking, small objects falling and other common activities.

• Image recognition: By using overhead cameras in a fixed location, one can track objects and learn movement patterns. The system adapts to the locations in which a single human enters/exits the room and remains inactive (lying/sitting on bed/chair). Common paths from entry points to inactive areas are then traced and remembered. It suspects a fall if a person becomes inactive in the middle of a common path

• Worn Devices: These systems require the user to wear external sensors. The devices track the vector forces exerted on the user. Usually these devices are a tri-axial accelerometer or gyroscope. If a specific pattern or threshold is broken, the device alerts a wireless receiver, which then alerts emergency contact. The majority of fall detection systems enquire some application specific hardware and software design. This increase cost and limits the commercial viability to the wealthiest, or most impaired, users. Many also have significant installation and/or training times, also limiting greater adoption. Despite implementation differences, all designs have the same requirements: reliability, ease of installation/use, and restriction of false positives. Falls are often sparse with months between occurrences, thus the system must always be ready and accurate. If installation costs or training time is high, users will reject the system. However, the major reason for failure is rejection by monitoring services due to a high number of false alarms. We propose a low priced system that is well suited to all the requirements by using existing mainstream technologies that are reliable and ubiquitous.

1.2 Brief project outline: Our approach is to use the number one fasting growing device which billions of people already own, a programmable cellular phone. Using existing cell phone technology not only reduces the cost to the patient, it also exploits a greater range of communication capabilities and integrated hardware and software features. Touch screen response and voice recognition, common to smart phones, provide a reliable interface for the user. By using interfaces that are similar to applications the user frequently uses, the rare interaction with the fall detection software should be familiar. Cell phones are also more discrete than a dedicated monitor device, which will reduce rejection due to the device's poor aesthetic value and intrusiveness. To limit false positives we implement several fall detection algorithms and two stages of communication. When a fall is detected, we first communicate with the user. If the user does not respond, we attempt to contact members in his/her social network. If either fail or the social contact confirms a fall, the system alerts an emergency service.

Chapter 2 Literature Review

2.1 System Study: Alwan, Prabhu Rajendran, Steve Kell, Steve Kell "UP-Fall Detection Dataset: A Multimodal Approach" Falls, especially in elderly persons, are an important health problem worldwide. Reliable fall detection systems can mitigate negative consequences of falls. Among the important challenges and issues reported in literature is the difficulty of fair comparison between fall detection systems and machine learning techniques for detection. In this paper, we present UP-Fall Detection Dataset. The dataset comprises raw and feature sets retrieved from 17 healthy young individuals without any impairment that performed 11 activities and falls, with three attempts each. The dataset also summarizes more than 850 GB of information from wearable sensors, ambient sensors and vision devices. Two experimental use cases were shown. The aim of our dataset is to help human activity recognition and machine learning research communities to fairly compare their fall detection solutions. It also provides many experimental possibilities for the signal recognition, vision, and machine learning community.

N.Krishna Chaitanya, R.Johnathan Rao "Falling Detection System Based on Machine Learning" As falling is the most important issue that faces elderly people all over the world, this paper proposes a detection system for falling based on Machine Learning (ML). In the proposed system, a dataset of videos containing falling actions has been utilized via dividing each video into many shots that are consequently being converted into gray-level images. Then, for detecting the moving objects in videos, the foreground is firstly detected, then noise and shadow are deleted to detect the moving object. Finally, a number of features, including aspect ratio and falling angle, are extracted and a number of classifiers are being applied in order to detect the occurrence of falling. Experimental results, using 10-fold cross validation, shown that the proposed falling detection approach based on Linear Discriminant Analysis (LDA) classification algorithm has outperformed both support vector machines (SVMs) and Knearest neighbor (KNN) classification algorithms via achieving falling detection with accuracy of 96.59 %.

Mihail Popescu, Yun Li, Manty disouza, and Dannie Rantz "A Microphone Array System for Automatic Fall Detection" More than a third of elderly fall each year in the United States. It has been shown that the longer the lie on the floor, the poorer is the outcome of the medical intervention. To reduce delay of the medical intervention, we have developed an acoustic fall detection system (acoustic-FADE) that automatically detects a fall and reports it promptly to the caregiver. Acoustic-FADE consists of a circular microphone array that captures the sounds in a room. When a sound is detected, acousticFADE locates the source, enhances the signal, and classifies it as "fall" or "nonfall." The sound source is located using the steered response power with phase transform technique, which has been shown to be robust under noisy environments and resilient to reverberation effects. Signal enhancement is performed by the beamforming technique based on the estimated sound source location. Height information is used to increase the specificity. The mel-frequency cepstral coefficient features computed from the enhanced signal are utilized in the classification process. We have evaluated the performance of acoustic-FADE using simulated fall and nonfall sounds performed by three stunt actors trained to behave like elderly under different environmental conditions. Using a dataset consisting of 120 falls and 120 nonfalls, the acoustic-FADE achieves 100% sensitivity at a specificity of 97%.

2.2 Proposed Work: The proposed system consists of prototype shoe in which insole is embedded with Tag switch, LCD display and NodeMCU microcontroller. The leads of each tag switch were connected to the power supply and ground pins of the microcontroller. The setup is connected to printed circuit board (PCB) placed under the insole. The insole is removable and tag switch positions can be reconfigured as desired. Pressure and inclination of the foot are continuously monitored and captured by the NodeMCU microcontroller. The system model mentioned above describes the overview of the design. The angle of inclination in is an inclination part of the gait cycle and hence no alert gets recorded. When a fall occurs, the angle of inclination is more than the threshold level and hence, an alert gets triggered. The angles of each phase in the gait cycle are measured and stored in the tag switch , which sends information when required. The tag switch sensor detects the change in resistance based on the pressure applied and both are integrated with NodeMCU,. If we link and analyze the acceleration of movement with the position of node worn by the patient, it would be likely to identify the posture of the person. The module using information about the angle of inclination sends this data over a Wi-Fi network to the IoT device, which saves data about the patients and recipients over a cloud The android app captures latitude and longitude information where the incident has taken place and sends it to the recipient as SMS on their mobile device. The sequence of events and overall system operation. To determine the sensor positioning in the shoe insole, maximum pressure points are identified by the study of foot anatomy. It can be divided into 4 areas where heel can be plotted as 1–3 areas, mid-foot as 4–5, metatarsal as 6–10 and toe as 11–15, The maximum force transmitted onto the foot originates at the bones of the lower leg. At the ankle, this force is distributed into three forces in the form of a tripod. At the foot, these forces are transmitted to the calcaneus, first metatarsal and across the second to the fifth metatarsal. In the first prototype that was developed, six sensors were placed at high-pressure areas to determine the fall The output accuracy is almost similar in all the three cases which implies that positioning of sensors at high-pressure points is critical than the number of sensors. Hence, based on the foot anatomy and the earlier works, heel and metatarsal are identified as the pressure points where the sensors detected the fall.

2.3 Scope of the project The main objective of this project is to reduce fatalities associated with a fall by developing an accurate and reliable alert system. This helps in reducing the response time required to attend the patient. A smart-shoe has been developed with simple integration of components including the pressure sensor. Fall conditions were derived based on the analysis of human foot pressure. The IoT is a system of interrelated computing devices that are provided with unique identifiers to transfer data over a network for data collection and analysis. This technology has evolved because of convergence of multiple technologies and widely used in smart health-care application for real-time monitoring of various parameters of patient's health condition and emergency notification. In this project, the output data from the hardware component is processed using a fall detection algorithm and alert during fall condition is transmitted over IoT-based wireless network to the messenger on a mobile application. The problems were addressed after the emergence of automatic fall detection systems that detect fall and alarm gets triggered without any human intervention. Recent findings in this field have enabled the use of pressure sensors which are mounted on wearable devices to achieve accurate results.

Chapter 3 SYSTEM ANALYSIS

3.1 Software Requirements Analysis A software requirements definition is an abstract description of the services, which the system should provide, and the constraints under which the system must operate. It should only specify only the external behavior of the system and is not concerned with system design characteristics. It is a solution, in a natural language plus diagrams, of what services the system is expected to provide and the constraints under which it must operate.

3.2 Hardware Requirements Analysis Hardware Requirements Analysis is to define and analyze a complete set of functional, operational, performance, interface, quality factors, and design, criticality and test requirements. 3.3 Minimum System Requirements Hardware requirements: • Processor: ARM or QualComm Processor (32 bit). • Ram: 128MB or More. • Hard Disk: Minimum 200MB. • Android Mobile Phone. Software requirements: • Android SDK 1.5 or More. • Eclipse IDE. • Programming language JAVA and XML. • Operating System Android (Linux Kernel).

Chapter 4 System Design

4.1 Design Overview The flowchart below shows the flow of our application in cellular phone. Firstly the application will be in the sleep mode that is the accelerometer will be running in the background. The accelerometer reading is taken periodically and is compared with the threshold values which are set by the user. If the accelerometer value crosses the threshold then the fall is detected and the application is started. If the value does not cross the threshold then the fall is not detected and the application goes back to the idle state. If the fall is detected then the application starts and alerts the user. The user has to respond to the application within the stipulated time. If the user respond to the alert within the time then the application goes back to the idle state else the message is sent to the social contacts and it waits for the reply from the contact. The message contains the key word,the GPS coordinates where the fall is detected. If the contact replies to the message the application checks the replied message for the key. If the message contains key then the call is made to that contact and it also enables the speaker so that bidirectional communication take place between the victim of the fall and the social contact.

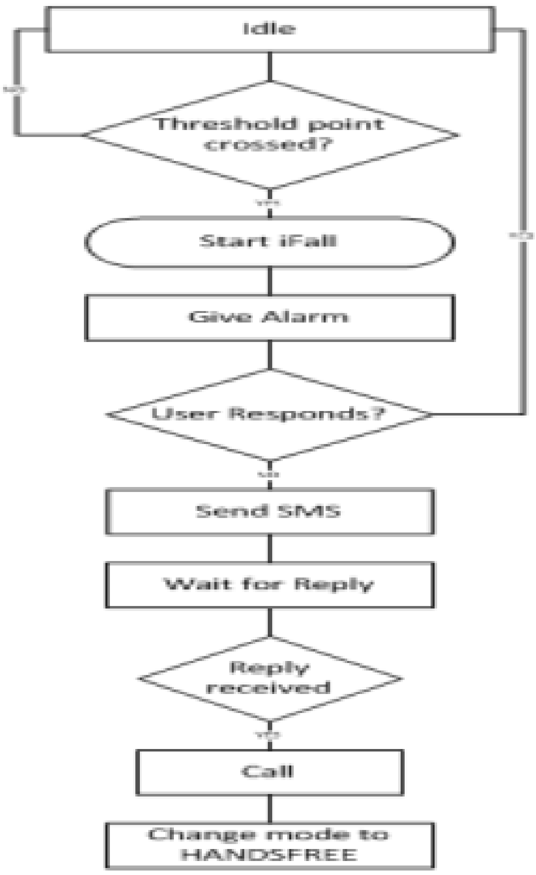

Figure 4.1: Flow chart of the application

4.2 System Architecture There are many modules in this application. Some of the modules are sensor manager module, threshold monitor module, contact manager module, SMS sending module, timer module, reply verification module and call manager module. The block diagram below shows the different modules and the relationship between them. First the accelerometer will be running in the background and when the value crosses the threshold the fall is detected. This activity comes under the sensor manager module and the threshold monitor module. In the timer module timer is started within which the user has to confirm whether the fall detected is a true alarm or a false one.

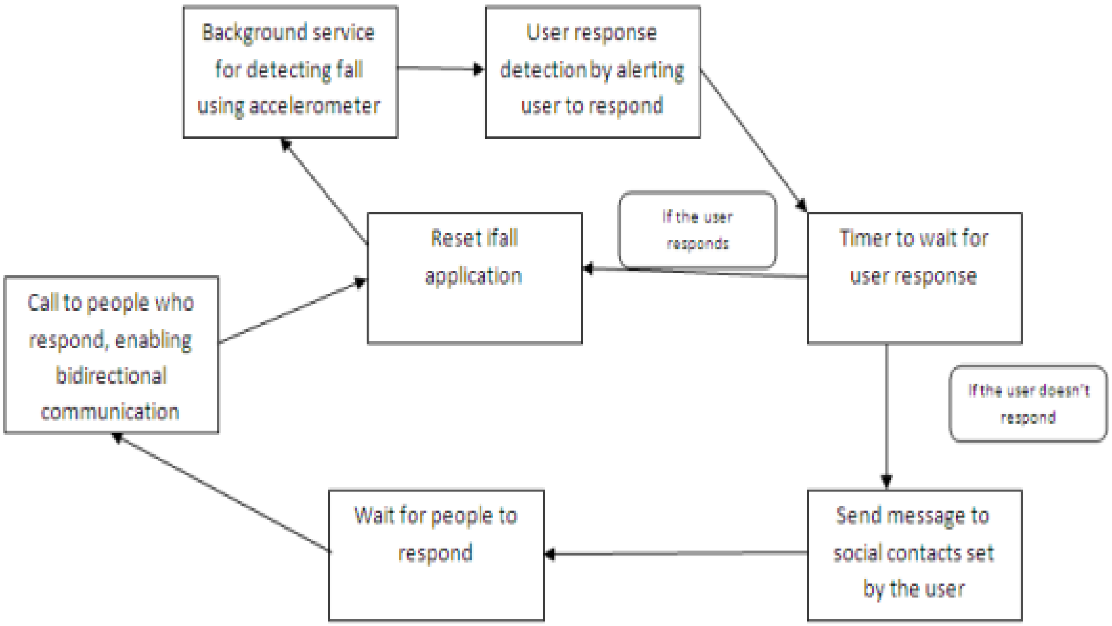

Figure 4.2: Block diagram showing different modules

In the contact manager module, the social contacts which are to be contacted in case of the fall detection is managed. In the SMS sending module the SMS is sent to the contact which contains the GPS coordinates and also the key. When the reply is received by the broadcast receiver it is verified whether the reply message contains the key or not. This is done in the SMS verification module or reply verification module. In the last module that is the call manager module a call is made to the social contact who has replied and the speaker is turned on and the communication between the contact and the user takes place.

4.3 MODULES

This project has mainly 6 modules. They are as follows: • Sensor Manager Module • Threshold Manager Module • Contact Manager Module • SMS Manager Module • Timer Manager Module • Reply verification module • Call Manager Module

4.3.1 Sensor Manager Module: A sensor is a device that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument. In this project we use accelerometer which is embedded in cellular phones. An accelerometer is a device for measuring acceleration and gravity induced reaction forces. These forces may be static, like the constant force of gravity pulling at your feet, or they could be dynamic - caused by moving or vibrating the accelerometer. An accelerometer measures the acceleration and gravity it experiences. Both are typically expressed in SI units meters/second2 (m·s-2) or popularly in terms of g-force. In this project we use accelerometer which senses the fall and then the application is started.

4.3.2 Threshold Manager Module: Threshold Manager (henceforth, TM) makes it easy to set up events and alarms (thresholds). A threshold is a test of some variable against some value, with a report when the threshold value is exceeded. So it scales to the largest networks, with little traffic overhead. Just testing if variables exceed or fall below threshold values does not do us much good unless we have a way to get the information out of the device. We probably want to be notified of drastic events by traps. We might well be content with just logging less critical threshold crossings.

4.3.3 Contact Manager Module: This module explains how to add contact to the social contact list. The user is given the option of adding more than one social contact to the list. When the fall is detected the message is sent to all the contacts one by one. There is also the options given to the user so that the user may delete the contacts and also the add contacts.

4.3.4 SMS Manager Module Text messaging, or texting, refers to the exchange of brief written messages between fixed-line phone or mobile phone and fixed or portable devices over a network. SMS/MMS Manager contains a powerful rule editor which can be used to automate message processing. This allows deploying common scenarios such as SMS voting polls, but also much more complex schemes.

4.3.5 Timer Manager Module: The Timer Manager is used to manager a set of logical timers. All Timer objects created with this Timer Manager can be centrally administered by using the suspend, resume and stop methods. All Timers execute in the same JVM as the thread that created the Timer and are transient. If the JVM fails then all Timers will be lost.

4.3.6 Reply Verification Module: When the fall is detected then the message is sent to the social contact specified by the user. The replied message is then checked for the password or the keyword. If the message contains the keyword and is sent by the same contact then the call is made or else the message is discarded.

4.3.7 Call Manager Module: Call Manager enables you to set controls and usage limits on your mobile phones. Administrators can set and update call controls at any time through an easy-to-use web interface. Mobile phone users can receive text messages or emails to alert them to approaching call limits. Call Manager lets you put limits (eg,"local calls only") on how your employees use their company mobile phones. This means you can manage your mobile phone expenditure each month.

Chapter 5 Software/tools/special libraries/Framework

5.1 Java: Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is currently one of the most popular programming languages in use, and is widely used from application software to web applications.

5.2 Apache Tomcat: Apache Tomcat (or Jakarta Tomcat or simply Tomcat) is an open source servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run. Tomcat should not be confused with the Apache web server, which is a C implementation of an HTTP web server; these two web servers are not bundled together. Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files. Tomcat started off as a servlet reference implementation by James Duncan Davidson, a software architect at Sun Microsystems. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation. The Apache Ant software build automation tool was developed as a side-effect of the creation of Tomcat as an open source project.

5.3 Android: Android is a software stack for mobile devices that includes an operating system middleware and key applications. Google Inc. purchased the initial developer of the software, Android Inc., in 2005. Android's mobile operating system is based on a modified version of the Linux kernel. Google and other members of the Open Handset Alliance collaborated on Android's development and release. The Android Open Source Project (AOSP) is tasked with the maintenance and further development of Android. The Android operating system is the world's best-selling Smartphone platform. Android has a large community of developers writing applications ("apps") that extend the functionality of the devices. There are currently over 150,000 apps available for Android. Android Market is the online app store run by Google, though apps can also be downloaded from third-party sites. Developers write primarily in the Java language, controlling the device via Google-developed Java libraries.

5.4 Eclipse: Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written mostly in Java and can be used to develop applications in Java and, by means of various plugins, other programming languages including Ada, C, C++, COBOL, Perl, PHP, Python, Ru by (including Ruby on Rails framework), Scala, Clojure, and Scheme. The IDE is often called Eclipse ADT for Ada, Eclipse CDT for C/C++, Eclipse JDT for Java, and Eclipse PDT for PHP. The initial codebase originated from Visual Age. In its default form it is meant for Java developers, consisting of the Java Development Tools (JDT). Users can extend its abilities by installing plug-ins written for the Eclipse software framework, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Released under the terms of the Eclipse Public License, Eclipse is free and open source software. It was one of the first IDEs to run under GNU Classpath and it runs without issues under IcedTea.

5.4.1 Rich Client Platform Eclipse provides the Eclipse Rich Client Platform (RCP) for developing general purpose applications. The following components constitute the rich client platform: • Equinox OSGi – a standard bundling framework • Core platform – boot Eclipse, run plug-ins • Standard Widget Toolkit (SWT) – a portable widget toolkit • JFace – viewer classes to bring model view controller programming to SWT, file buffers, text handling, text editors. • Eclipse Workbench – views, editors, perspectives, wizards

5.5 ARDUINO IDE Arduino first and foremost is an open-source computer hardware and software company. The Arduino Community refers to the project and user community that designs and utilizes microcontroller-based development boards. These development boards are known as Arduino Modules, which are open-source prototyping platforms. The simplified microcontroller board comes in a variety of development board packages. The most common programming approach is to use the Arduino IDE, which utilizes the C programming language. This gives you access to an enormous Arduino Library that is constantly growing thanks to open-source community.

Chapter 6 Conclusion

Our system provides a viable solution to fall detection in the elderly. Using existing, mass marketed technologies will limit cost making it available to the majority of the public. Implementing proven fall detection algorithms makes the system highly reliable. Reliability and reduced number of false positives means greater adoption by emergency services. The importance of the cell phone in everyday life decreases the chances of being forgotten. Everyday interaction with the phone makes the interface more familiar to the user. A cell phone is also less intrusive than dedicated devices. Here we are using Android based smart phone which is available in the market. Thus we are also integrating the advantages of Android in our application. Every Android phone has the accelerometer embedded in it, which enables us to detect the fall in all 3 axis. Android is very secure software stack for mobiles and is very reliable. Thus making our project very reliable. The familiar interface, non-intrusiveness, and affordability leads to less rejection from users. By combining cheap hardware and open source software, we hope to provide a realistic solution to the elderly fall problem.

Chapter 7 Bibliography

[1] Majd Alwan, Prabhu Jude Rajendran, Steve Kell, David Mack, Siddharth Dalal, Matt Wolfe, and Robin Felder. A smart and passive floor-vibration based fall detector for elderly.
[2] Mihail Popescu, Yun Li, Marjorie Skubic, and Marilyn Rantz. An acoustic fall detector system that uses sound height information to reduce the false alarm rate. 30th Annual International IEEE EMBS Conference, August 2014.
[3] Tracy Lee and Alex Mihailidis. An intelligent emergency response system: preliminary development and testing of automated fall detection. Journal of Telemedicine and Telecare, 11(4):194–198, 2016.
[4] Shaou-Gang Miaou, Pei-Hsu Sung, and Chia-Yuan Huang. A customized human fall detection system using omni-camera images and personal information. pages 39–41. Proceedings of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference, April 2014.
[5] Hammadi Nait-Charif and Stephen J. McKenna. Activity summarization and fall detection in a supportive home environment 2014.
[6] Caroline Rougier and Jean Meunier. Demo: Fall detection using 3d head trajectory extracted from a single camera video sequence.
[7] Google, Inc., What is Android?, http://code.google.com/android/what-isandroid. html [last accessed on Mar 1, 2009].
[8] A.Chou, J.Yang, B.Chelf, S.Hallem, and D.Engler. An Empirical Study of Android Operating Systems, In Proc. of 18th ACM Symposium on Operating Systems Principles (SOSP), P. 73–88, 2008.
[9] W.Gu, Z.Kalbarczyk, R.K.Iyer, and Z.Yang. Characterization of Linux Kernel Behavior, In Proc. of International Conference on Dependable Systems and Networks (DSN), 2007, P.4.
[10] http://code.google.com/android/ - Official Android Google Code Webpage.
[11] http://www.openhandsetalliance.com/ - Open Handset Alliance Webpage.
[12] http://www.androidwiki.com – Android Wikipedia.