WAVELET TRANSFORM BASED FALL DETECTION USING WEARABLE ACCELEROMETERS

by

Gökhan Remzi Yavuz

B.S., Computer Engineering, Boğaziçi University, 2008

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2010

WAVELET TRANSFORM BASED FALL DETECTION USING WEARABLE ACCELEROMETERS

APPROVED BY:

Prof. Cem Ersoy (Thesis Supervisor)	....
Prof. Lale Akarun	....
Dr. Hülya Yalçın	....

DATE OF APPROVAL: 17.09.2010

DEDICATION

This thesis is dedicated to my family.

ACKNOWLEDGEMENTS

First of all I would like to thank to my thesis supervisor Prof. Cem Ersoy. I am grateful to him not only for his guidance and efforts in the writing of this thesis but also because he presented me the opportunity to pursue an academic career. I am glad that I had the opportunity to work with, to discuss with and to have fun with him.

I would also like to thank to Prof. Lale Akarun for the idea that formed the basis of this thesis and for the efforts she put into making this work a better thesis.

I also thank to Dr. Hülya Yalçın for her belief in me. Without her constant efforts to make me work, this thesis would have taken a lot more time.

I thank to Hande Alemdar, Mustafa Ozan Özen, Yunus Emre Kara, Özlem Durmaz İncel, Bilgin Koşucu for their constant support and collaboration. I lost count of how many times I made you all fall down. I also thank to Tim van Kasteren for his patient efforts in developing my scientific skills, he made this thesis a better reading material.

I would like to thank to Kutluhan İncekara for being the perfect home mate. I of course thank to Rabun Koşar as he helped me on any topic I needed help dealing with. Rabun, you will be surely missed!

I thank to my family for their support and love even when I made them angry. I am very lucky to have you as my family.

Finally, Deniz, thank you for giving me the best days of my life.

This research is supported by the Scientific and Technological Council of Turkey (TUBITAK) under the grant number 108E207, by the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610, and by the Boğaziçi University Research Fund under the grant agreement number “5146”.

ABSTRACT

WAVELET TRANSFORM BASED FALL DETECTION USING WEARABLE ACCELEROMETERS

Falls are identified as a major health risk, especially for the elderly people and are considered a major obstacle to independent living. Quick medical response is desired in case of a fall event. However, the fall may leave the elderly person in such a state that the elderly may be unable to call for help on his/her own. Automatic and fast detection of falls would decrease the health risks associated with the falls and would make independent living safer for the elderly people. In this thesis, we propose an automatic fall detection system that uses a wearable accelerometer and incorporates wavelet transform as a feature extraction method. We conducted experiments to investigate the performance of the system under the effect of several factors including fall properties, selection of wavelet transform parameters and sensor platform types. Results indicate that our proposed approach is robust with high fall detection performance.

The fall detection mechanism was realized using the wearable sensors that were part of an indoor monitoring environment, namely WeCare. WeCare not only provided the necessary sensing capabilities for the fall detection but it also made available several communication and notification methods. Using these methods, we were able to notify caregivers in case of fall detection. In this thesis, we also describe the WeCare system and the integration of our fall detection study into it.

ÖZET

GİYİLEBİLİR İVMEÖLÇER KULLANILARAK DALGACIK DÖNÜŞÜMÜ TABANLI DÜŞME SEZME

Düşme, özellikle yaşlılar için, önemli bir risk ve yaşlıların bağımsız yaşamı önünde bir engel olarak belirlenmiştir. Bir düşme durumunda hızlı müdahale gerekmektedir, fakat düşmeden kaynaklanan sebeplerle düşen kişi kendi başına yardım çağıramayacak durumda olabilir. Düşme durumlarının hızlı ve otomatik bir şekilde algılanması, düşme kaynaklı sağlık risklerini azaltacağı gibi yaşlılar için bağımsız yaşantıyı daha güvenli hale getirecek ve bağımsız yaşam önündeki bu engeli kaldıracaktır. Bu tezde otomatik düşme sezme amacı ile giyilebilir ivmeölçer kullanan dalgacık dönüşümü tabanlı bir yöntem önermekteyiz. Ayrıca çeşitli etmenlerin düşme sezme yönteminin başarımına olan etkisini incelemek amacı ile çok sayıda deney yapılmış olup, bunların sonuçları da bu tezde verilmektedir. Sonuçlar önerilen yöntemin pek çok farklı etmenin etkisi altında yüksek başarım gösterdiğine işaret etmektedir.

Bahsedilen düşme sezme yöntemi, WeCare adı verilen bir sağlık gözetimi ortamının parçası olan giyilebilir ivmeölçerler kullanılarak gerçeklenmiştir. WeCare düşme sezme için gereken algılama yeteneklerini sunmakla kalmayıp, çeşitli iletişim ve uyarı yöntemlerini de kullanıma açmaktadır. Bu yöntemleri kullanarak düşme sezilmesi durumunda bakıcılara ve ilgili kişilere uyarı gönderilerek bu kişilerin hızla durumdan haberdar edilmesi sağlanmaktadır. Bu tezin bir parçası olarak WeCare ortamının yapısı ve düşme sezme yönteminin bu ortama eklenmesi de anlatılacaktır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

A_i	Approximation coefficient at scale i of DWT
$\vec{a_t}$	Acceleration vector at time t
d	Euclidian distance between two vectors
D_i	Detail coefficient at scale i of DWT
$F(Y)$	Set of coefficients from the DWT of an arbitrary signal Y
$\hat{f}$	Fourier Transform of a function f
f_s	Scaling of function f by s
f_τ	Translation of function f by τ
$FD(t)$	Fall decision for time t
FP	Number of false positive fall labels
g	High pass decomposition filter
GT	Number of falls in a dataset
h	Low pass decomposition filter
l	Length of the decomposition filters
N	Length of the discrete signal
$MP(X)$	Mapping function
$P(\vec{a_t})$	Probability of observing acceleration $\vec{a_t}$
$P(C_i\|)$	Probability of action class i
$P(C_i\| \vec{a_t})$	Probability of action class i given acceleration $\vec{a_t}$
TP	Number of true positive fall labels
v	Template vector calculated from a training set
w	Length of the calculation window
X	A sequence of uniformly sampled acceleration vectors
α	Sampling Interval
γ	Continuous wavelet transform of a function
θ	Predefined Threshold
ϕ	Scaling function
ψ	Mother Wavelet

AC	Acceleration Cross Product
CWT	Continuous Wavelet Transform
DWT	Discrete Wavelet Transform
FoV	Field of View
GPS	Global Positioning System
GUI	Graphical User Interface
HMM	Hidden Markov Model
kNN	k-Nearest Neighbor
PC	Personal Computer
PCA	Principal Component Analysis
RFID	Radio Frequency Identification
RSS	Root Sum Square
SMS	Short Message Service
SVM	Support Vector Machine
WHO	World Health Organization
WSN	Wireless Sensor Network

1. INTRODUCTION

Accidental falls are risky events, especially for the elderly people living independently. Studies show that, more than one third of the adult population over the age of 65 falls at least once a year in the USA [2]. Up to 30 percent of these falls result in medium to severe injuries that can lead to the death of the older adults [3]. Quick medical response is desired in such situations [4], but the injuries may cause the older adults to be immobile to the extent that they can not even be able to reach a phone to call for help. One proposed solution to this problem is to use emergency buttons installed throughout the house or on the elderly people themselves so that they can press them in case of a fall related injury. However, if the elderly person becomes unconscious, he/she may not be able to press the button to call for help. Hence, it is important to develop an automatic fall detection system that requires no human intervention.

World Health Organization (WHO) defines a fall as an event which results in a person coming to rest inadvertently on the ground or floor or other lower level [5]. Since these events involve motion and change of pose, observing certain characteristics may provide us with necessary information to detect falls. Many types of sensors can be used to observe motion and pose of the older adult and determine if a fall has occurred or not. Current work on automatic fall detection methods can be classified into three main classes in terms of the sensors they use: video based methods, acoustics based methods and wearable sensor based methods [6]. Video based methods use images provided by cameras installed in the environment and they analyze changes in designated features to detect falls (e.g. orientation and aspect ratio of a bounding ellipse [7]). Acoustics based methods try to detect falls by detecting vibrations caused by the impact to the ground. For instance, Zigel, et al. proposes a method that uses a vibration sensor and a microphone to detect vibrations and noise generated by the impact [8].

For automatic fall detection, methods based on wearable sensors are more attractive since video based methods raise privacy concerns and acoustics based methods are very susceptible to ambient noise. Moreover, video based and acoustic based methods can only operate in environments that are wired with them, while wearable sensor based methods will be able to operate as long as the person wears the sensors.

The most common wearable sensor used for fall detection is the accelerometer which can be used to detect acceleration characteristics of movement of the person as well as estimating the pose of the person in order to detect falls. However, there are challenges in automatic fall detection using wearable accelerometers. First of all, the person may forget to wear the sensor or charge its battery. However, this challenge may be addressed by a specially designed user interface that is capable of reminding the person to wear or charge the sensor. Secondly, the acceleration signal is continuous in time and value, which makes the task of locating a fall in time more challenging. The third challenge is that the acceleration signal provided by the accelerometer is noisy, due to possible hardware imperfections and possible independent movements of the sensor.

Besides the challenges due to the nature of accelerometers, there are also challenges in terms of analyzing the signal since there is a significant ambiguity in the acceleration signal. Many non-fall actions, such as jumping and quickly sitting down, can cause amplitudes similar to that of falls. Therefore, computations solely based on the raw acceleration signal directly do not provide significant information in terms of detecting falls. On the other hand, falls are relatively quick events, in which a free-fall period due to gravity, and an impact afterwards are observed, resulting in an acceleration signal that has significant local variations that can be utilized to detect falls.

In order to extract local variations, a preprocessing of the accelerometer signal is required. Discrete Wavelet Transform (DWT) provides us a flexible method for time-frequency analysis of signals, which can be used to extract the local variations in the acceleration signal as well as dealing with the challenges mentioned earlier. The behavior of DWT is governed by the selection of the mother wavelet, ψ, which is the central parameter of the transform. Since, mother wavelet defines the time and frequency resolutions of the transform, selecting a proper wavelet is crucial to achieve the desired task.

In this thesis, we are going to employ DWT as a feature extraction method for the fall detection task and investigate the fall detection performance with a variety of mother wavelets. Since it is not practical, if at all possible, to take an analytical approach to mother wavelet selection for fall detection, we are going to use an empirical approach and evaluate the performance of the feature extraction method and mother wavelet selection via experiments.

The rest of the thesis is organized as follows: In Chapter 2, we give an overview of related work; in Chapter 3, we describe Wireless Enhanced Healthcare (WeCare) system and the integration of fall detection module into this environment; in Chapter 4, we summarize our earlier studies on fall detection, describe the discrete wavelet transformation and its use as a feature extraction method; in Chapter 5, we describe the experimental setup and give the results of the experiments. Finally, our concluding remarks will be presented in Chapter 6. The notation used throughout this thesis is presented in Appendix A.

2. FALL DETECTION METHODS

Research on automatic fall detection show that many different types of sensors can be employed for fall detection purposes. As stated earlier, automatic fall detection methods can be categorized into three classes according to the sensors being used in the process: Video based, Acoustic based, Wearable sensor based [6]. Since the information provided by each sensor differs for different sensor types, fall detection algorithms employed for each sensor differ as well.

2.1. Video Based Fall Detection Methods

Video based fall detection methods utilize video streams provided by one or more cameras that are installed in the environment monitored for fall events. These methods generally employ a background subtraction step in order to detect the presence of a human in the field of view (FoV) of the camera. This step is generally followed by a noise removal step, in which undesired noise is removed from the image. After these steps a feature extraction method is applied to the video stream. During the feature extraction process, a single frame or multiple frames from the video stream may be used to evaluate certain features, such as vertical and horizontal speeds of the human image which can be used in fall detection [9]. These features are then used in the final evaluation step to detect falls.

In [10], Cucchiara, et al. propose an automatic fall detection system that may be configured to utilize multiple cameras for fall detection. Their main rationale behind using multiple cameras is that; a single camera cannot cover the whole living space of a home environment due to occlusions and existence of multiple rooms. Therefore, they utilize multiple cameras and employ a tracking and hand-over mechanism that enables the system to track people over multiple cameras. On each camera stream, background subtraction is applied to detect the human in the FoV and then projection histograms of the human blob in the image is calculated on each axis. Finally these histograms are used as the observations for a Hidden Markov Model (HMM) classifier.

A similar approach was employed in [11], in which a Layered HMM was employed for classification and the feature used for classification was the orientation of the bounding rectangle with respect to the vertical axis.

In [12], another fall detection method is proposed which utilizes two perpendicular cameras. Similarly background subtraction is applied to each frame and human presence is detected in the FoV. When a human is detected in the image, Principal Component Analysis (PCA) is applied to the image of the person in order to determine the orientation of the human body and the variation along the axis or orientation. The outcome of the PCA in both cameras are then used by a probabilistic classifier which takes advantage of the axis information gathered from multiple frames to calculate the probability of a fall event occurring in those frames.

In [13], a single camera fall detection system is proposed which again starts with a background subtraction. After the background subtraction step the image is transformed into a black and white image to obtain the silhouette of the human in FoV of the camera. A bounding box is then fitted to the silhouette of the human. In the feature extraction step, the aspect ratio, that is the ratio of width of the bounding box to the height of the bounding box, is calculated. The aspect ratio is used to determine the posture of the human in the image using a k-Nearest Neighbor (kNN) classifier. The posture information is then used to detect unexpected transitions from standing posture to a lying posture which are assumed to correspond to a fall.

In [14], Zhang, et al. propose two similar methods in which they also find the human object via background subtraction. Using the human object in the image they calculate the row centroid, row coordinate of the vertical middle point, of the human for each frame. The first method they propose applies a threshold to the row centroid. If the row centroid is lower than the predefined threshold a fall is detected. The second method they propose utilizes multiple frames and calculates additional features such as the speed of decrease of the row centroid, the total decrease in the row centroid. Multiple thresholds are then applied to these features and a fall is detected in case each of these features exceed the corresponding thresholds.

In one of our earlier studies, we have proposed a video based fall detection system which also started with a background subtraction step [7]. After this step, a silhouette of the person in the image is extracted to which an ellipse was fitted. Figure 2.1 shows a sample frame from one of the experimental videos, the outcome of background subtraction and the ellipse fitting, respectively. Using the ellipse, we calculated several features including the aspect ratio of the ellipse, the orientation of the ellipse and the change of rate of these two. These features are then used by a probabilistic classifier which was trained earlier using the same features.

Figure 2.1. Background Subtraction and Ellipse Fitting

Although most of the studies on video based fall detection approaches report high fall detection performance, there is a major criticism against these methods based on privacy of the elderly. Producing a live video stream of the living environment of the elderly person creates the risk that the stream may be accessed externally to violate the privacy of the elderly. Moreover, daily living involves actions and places for which privacy is of top priority, such as bedrooms and bathrooms. Although there are studies which employ indirect image features such as silhouettes [13], or use other imaging modalities such as thermal imaging [15], privacy is still a major concern for video based fall detection methods.

2.2. Acoustics Based Fall Detection Methods

Acoustics based methods detect fall based on vibrations that are caused by the fall event, such as the noise or the floor vibrations caused by the impact to the floor.

The main advantage of these methods is that they do not require the user to wear anything and they do not pose a significant threat to the privacy of the user.

In [8], Zigel et al. propose a method that uses a vibration sensor and a microphone to detect vibrations and noise generated by the impact. Their method consists of two phases: Training and Testing. In the training phase, the fall detection system is supplied with audio and vibration signals recorded during multiple fall and non-fall events. The system extract several features for each event, such as the length of the event, the amount energy contained in the event and the amount of energy contained in certain frequencies for the signals. These features are then used to train a Bayesian classifier which separated events into fall or non-fall classes. In the test phase, the system uses the Bayesian classifier trained in the training phase for fall detection.

Another approach to acoustics based fall detection is to use active acoustics systems. In [16], such a system based on ultrasonic transmitters and receivers is described for use in nursing homes. The system uses an ultrasonic transmitter attached to the wheelchair of the elderly, together with the fixed receivers in known locations this is used to localize the elderly. Nurses are notified when an elderly approaches a risky zone, such as the entrance of the toilet. Furthermore, these risky zones are equipped with roof mounted ultrasonic radars which can be used to assess the posture of the elderly and detect fall events. In this aspect, the system resembles a video based method, while maintaining privacy.

A quick review of the literature indicates that acoustics based methods receive less attention than video or wearable sensor based methods owing to the fact that acoustics based methods are more susceptible to external noise. Instead of using only acoustics for fall detection, there are studies on hybrid methods that combine acoustics based methods with video based methods [17,18] or with wearable sensor based methods [19].

2.3. Wearable Sensor Based Methods

Another approach to fall detection is using wearable sensors to monitor the subjects in order to detect falls. Mainly, kinematic sensors such as tilt sensors, gyroscopes and accelerometers are used in these methods as they can be used to estimate the posture or the motion of the human body. Although there are also studies that utilize other types of sensors, such as [20] which monitors autonomous nervous system responses or [21] which uses an air-pressure sensor as well as an accelerometer, the main focus is on the use of kinematic sensors.

An accelerometer based method was proposed in [22] which used an acceleration sensor board with radio communication capabilities. They used the accelerometers to estimate the pose of the human body for regular intervals. Using these pose estimates, they distinguished transitions to and from the lying posture and calculated the speed of these transitions. The falls are detected when the transitions to the standing posture exceed a certain value. However, if the posture quickly changes from lying posture to standing posture again, then the detected event is assumed to be a recovery from a fall.

In [23], an accelerometer sensor board is used to detect impacts to the ground by applying a threshold to the magnitude of the acceleration. When the acceleration magnitude exceeds the mentioned threshold a possible fall is detected. This technique is also employed in [24]. However, they state that using only this threshold based technique fails to distinguish falls from all normal activities as some normal activities may result in acceleration magnitudes greater then the threshold value. Therefore they employ pose estimation in the fall detection as well. They detect a fall if they observe an impact during a transition from a standing posture to a lying posture. The same algorithm is used in [25] as well in a secure fall detection device prototype. A similar algorithm was proposed in [26], which utilized a sensor that detected impacts and any pose change before and after the impact. If the sensor detected an impact another device worn by the user applies a frequency analysis method to the acceleration signal sent by the sensor in order to confirm the impact.

In [27], a smart home system is proposed for tele-monitoring of the elderly. A fall detection module was included in this smart home system which uses a waist-worn accelerometer which estimates the pose of the elderly person with regular intervals as well as monitoring the activity level of the elderly. A fall is detected when the person is in a lying posture after a period of inactivity. Their assumption is that the falling person will end up in a lying posture and will not be able to move.

A threshold based method was used in [28] to detect falls. They determined three thresholds for three different features calculated from the acceleration signal. The features and the corresponding threshold values were determined empirically by analyzing simulated falls of a young volunteer. The first feature they used was the acceleration of the body in the horizontal plane. The second feature was the velocity of the body at the impact time which is calculated using the integral of the acceleration signal. And the third feature was the acceleration magnitude in the three dimensional (3D) space. A fall is detected if the first and second features exceed their thresholds simultaneously or if the third feature exceeds its threshold.

In [29], Chao, et al. propose a fall detection method based on the acceleration cross product (AC) feature. AC is the cross product of the current acceleration vector with a reference acceleration vector that was recorded when the person was standing still. Due to the nature of the cross product operation, when a person is standing still, AC would assume a zero value whereas this value would be one in case of a lying posture. Moreover, AC showed great fluctuations during the impact to the ground. The fall is detected when AC exceeds a predefined threshold. As a further improvement, they also necessitate a lying pose to be detected after the threshold is exceeded.

Using only a single wearable sensor limits the pose detection capabilities of the fall detection system. In [30], a health monitoring system is described that utilizes multiple sensors worn on the body at different locations. Using these multiple sensors the authors can calculate the angles and orientations of joints and multiple parts of the body. Although the basic motivation behind their work is to monitor patients after hip operations, the system can detect falls as a side effect. A very similar system is defined in [31] in which multiple sensors are used to estimate the pose of the human body. However, fall detection is carried out by applying thresholds to two of the accelerometers. While both of these systems have their merits and may improve the fall detection performance with respect to single sensor systems, they suffer from added obstructions to the user.

In [32], a three-axis accelerometer is used to detect near falls instead of falls. Authors recorded signals while subjects performed several actions on a controlled environment and these actions were divided into five second gait segments. Then, acceleration vector magnitudes and acceleration vector areas are calculated which in turn are used to determine if the gait was a near-fall or not. Detection of near-falls enable fall risks being detected beforehand and may be used for pre-impact fall detection. Detecting the falls before the impact occurs may enable injury preventive measures to be taken, such as inflating an airbag to mitigate impact forces [33].

In [34], four characteristics were identified for falls: Weightlessness (Free-fall), Impact, Motionlessness and Posture Change. Free-fall is the time just before the impact to the ground during which the movement of the body is governed mainly by the gravity. Since the accelerometers measure the gravity of the Earth while they are held still, we expect a zero or low acceleration reading during the free-fall followed by the high acceleration impulse caused by the impact to the ground. Motionlessness is the interval during which the falling person does not move due to injuries or unconsciousness caused by the impact. The author propose to analyze the acceleration signal in order to detect these characteristics as well as the posture changes. Similarly, in [35], free-fall and impact characteristics of the falls are utilized, they apply two threshold and determine whether a free-fall is followed immediately by an impact. Yet another similar approach was taken in [36], which extended the idea of impact to include multiple impact as in the case of rolling downstairs.

Three algorithms that use some of these characteristics were compared in a study of Kangas, et al. They used total sum vector and dynamic sum vector together with vertical acceleration and velocity in order to detect fall characteristics [37,38]. In their first fall detection algorithm only impact and posture change is detected, for the second algorithm they included detection of free-fall and for the last algorithm they included a threshold condition on the velocity. Their results indicated that the simplest algorithm, that only monitored impacts and posture changes performed the best.

Another approach to the fall detection is calculating the energy expenditure in the muscles of the person as they give direct indications about the action being carried out. However, since this cannot be observed directly, it has to be estimated by other means. Accelerometers are good candidates for this estimation. In [39], Zhang propose an accelerometer based fall detection method based on this energy expenditure estimation. He first observes if the acceleration magnitude exceeds a predefined threshold, if the threshold is exceeded, the energy expenditure is calculated for a fixed time window. If the signal results in a high energy expenditure value, the pose of the body is also checked to see if a change from standing posture to a lying posture has occurred. If all the conditions are satisfied, a fall is detected.

2.3.1. Mobile Phone Based Fall Detection Methods

Wireless sensor network (WSN) nodes have been a natural choice for the wearable sensor based fall detection studies with their small structure, wireless communication capabilities and relatively long battery lives. However, the use of WSN nodes required an infrastructure, i.e. another WSN node connected to a gateway computer, in order to detect falls and relay emergency notifications. There was a need for better communication capabilities in order to achieve pervasive fall detection and cellphones became an option for such a duty with the introduction of Bluetooth. In one of the initial studies on cellphone assisted fall detection, a wearable accelerometer device with Global Positioning System (GPS) and bluetooth capabilities was used to detect falls and localize the person [40]. The fall notification together with the location of the user was then relayed to the cellphone, which in turn notified emergency personnel. A similar use of cellphones was studied in [41], in which the authors used a two layered decision making method based on Support Vector Machines (SVM) and kNN, however in this study, the cellphone is not only used to relay fall notifications to the external parties but also to receive feedback about the fall detection decisions.

With the integration of accelerometers on smartphones, it has become possible to develop accelerometer based fall detector applications that can run on the smartphones such as [42,43]. Smartphones possess high computational powers that can be compared to the personal computers of the recent past. Together with their communication capabilities and GPS support they form a promising platform for pervasive fall detection. Utilizing smartphones for fall detection has the added advantage that the person is not required to carry any extra devices for fall detection. For example, iFall [44] is a fall detection application designed for the Android platform [45] produced by Google. Since the application has to share smartphone resources with other applications, the authors used a thresholding method that requires minimal computation power. They defined two threshold values based on other studies and they detected suspected falls when the acceleration value falls below the first threshold and rises above the second threshold in a predefined duration. The application detects a fall if the suspected fall is followed by a motionless state.

3. WeCare: HEALTHCARE MONITORING SYSTEM

WeCare, is a wireless sensor technology based application which is designed to provide a multi modal sensing environment for the homes of the elderly people [46]. Due to the modularity of the WeCare application, it is possible to develop application modules for WeCare that utilizes the sensory capabilities of WeCare and implement particular functionalities. The fall detection study in this thesis aims to implement such a *fall detection module* for WeCare. Therefore, before moving onto the fall detection method discussion, a brief description of the WeCare application, its components and its capabilities will be given in this chapter. The basic functionalities of the system can be summarized as follows:

- Providing a multi modal sensing environment, e.g. temperature sensors, humidity sensors, video cameras
- Identifying the presence and the location of the residents in home
- Identifying predefined emergency situations
- Identifying user-defined emergency situations
- Providing a simple graphical user interface (GUI) through which systems configurations can be made
- Providing a web based interface for remote access and monitoring.
- Providing warning mechanisms for the emergency situations such as SMS, e-mail

3.1. WeCare System Architecture

In order to provide these functionalities and to extend them to include support for application modules and plug-and-play sensors the WeCare architecture proposed in [46] was modified. Figure 3.1 shows the modified architecture.

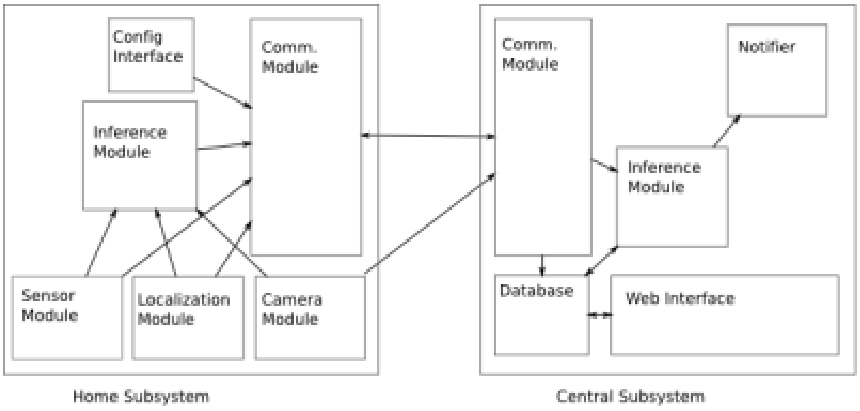


Figure 3.1. WeCare System Architecture

3.1.1. Home Subsystem

The main purpose of this subsystem is to implement ambient intelligence in the target home in which the elderly person lives. This subsystem is responsible from data gathering using sensors and cameras, inference from the gathered data and delivering the data to the Central subsystem. This subsystem consists of:

- Sensor Module: The sensors that are installed in the home and the related data gathering software for these sensors.
- Localization Module: The sensors and software that provides the localization functionality.
- Camera Module: The video cameras that are installed in home and the related software to acquire images from these cameras.
- Configuration Interface: The interface that enables users to modify the home subsystem, for instance add a new sensor to the Sensor Module.
- Inference Module: The collection of software that uses data provided by Sensor and Camera Modules to make inferences.
- Communication Module: The client software that communicates with the Central Subsystem. This module relays the data gathered from the house as well as the

inferences made from the data to the Central Subsystem.

3.1.2. Central Subsystem

This subsystem is mainly responsible for data representation, i.e. it provides an interface through which the data gathered from the house and the inferences made about the data can be accessed. Past data can also be accessed as it is stored in a database. Furthermore, this subsystem has the ability to notify people about emergency situations described as alarms. This subsystem consists of the following modules:

- Communication Module: This is the server software that communicates with the Home Subsystem and makes the data received from the Home Subsystem available to the other modules.
- Database: This module stores the information about the Home Subsystem configuration that is necessary for data representation as well as the data gathered from the sensor module in the Home Subsystem and related inferences.
- Web Interface: This module provides the web based remote access interface to the WeCare system.
- Notifier: This module relays the emergency notifications to the related people. For instance, this module may send messages through Short Messagin Service (SMS) or via e-mail in case of emergency.
- Inference Module: This is the collection of software that uses data provided by one or more Home Subsystems in order to make inferences.

3.2. Testbed and WeCare Prototype

In order to test the proposed architecture, a testbed was setup in a $55m^2$ laboratory. The testbed was designed to be similar to a home with a living room, a bedroom and a kitchen. The testbed home is decorated using furniture that can be found in an ordinary home; coaches, carpets, dining table, chairs and a bed (Figure 3.2). Therefore, we can realize almost any scenario that can occur at the home of the elderly person. The prototype of the WeCare was developed on this testbed. Several sensors

Figure 3.2. WeCare testbed home

were deployed in the testbed environment as well as multiple cameras. A desktop computer hosted the software for the home subsystem while another hosted software for the central subsystem.

The prototype software was mainly developed using .NET technologies, with the exception of the configuration software which was developed using Java. The communication between modules were realized using TCP/IP communications protocol. The web interface of the system was built using Silverlight technology due to its support for animated graphics. Figure 3.3 shows the home screen of the WeCare prototype. As seen on the figure the system has four types of entities: House(Rooms), people, alarms and cameras. In the home screen recent alarm events are displayed as well as the people that are in the house and the people that are defined in the system but not present in the house. Entities such as rooms, people and cameras are defined using the configuration interface on the Home Subsystem. Sensor assignments for each of these entities are also carried out using the configuration interface. Once the rooms are defined, sensor and camera assignments are carried out, these become available in the web interface as well. Figure 3.4 shows the Living Room page in the WeCare prototype. At the top of the page, values for related sensors are displayed, below that any recent alarms are displayed as well as the people in the room, in this case there are neither

Table 3.1. Hardware Used in WeCare Prototype

Testbed Room	**Hardware**	**Function**
Living Room	1 Imote2.NET with ITS400 [47]	Ambient Sensing
	1 SenseNode [48]	Acoustics sensing
	1 AXIS 207W Network Camera [49]	Video stream
	1 AXIS 1031W Network Camera [50]	Video stream
	1 UDEA RWID-R12 RFID Antenna [51]	Localization
Bedroom	1 Imote2.NET with ITS400	Ambient Sensing
	1 SenseNode	Acoustics sensing
	1 AXIS 1031W Network Camera	Video stream
	1 UDEA RWID-R12 RFID Antenna	Localization
Kitchen	1 Imote2.NET with ITS400	Ambient Sensing
	1 SenseNode	Acoustics sensing
Elderly Person	1 Imote2.NET with ITS400	Acceleration
	1 Active RFID Tag	Localization
Home Computer	1 Imote2.NET with ITS400	Data Sink

alarms nor people in the Living Room. At the bottom of the page the live stream from the associated camera can be seen. Figure 3.5 shows the page for the Kitchen, in which an alarm can be seen as well as the sensor values. The alarm is triggered with the fall of the *GrandPa*, it displays the time of the event, its duration and the person related to the event. Alarms are defined using the web interface unlike the rooms, people and cameras which were defined using the configuration interface. Through the web interface, one can define the conditions that will trigger the alarm as well as the actions that should be taken in case the alarm is triggered. There are two ways to define the alarm conditions: Custom and Predefined. In the custom alarm definition, trigger conditions can be determined for each sensor separately and multiple conditions can be combined using logic notation. In the predefined alarm definition method rooms and people defined in the system are listed for the user to choose, when the user chooses one of these applicable predefined alarm conditions are listed. Figure 3.6 shows the predefined alarm definition page with only *Bedroom* selected. There is only one predefined condition that can be selected; “Fire in the room”. Applicable actions

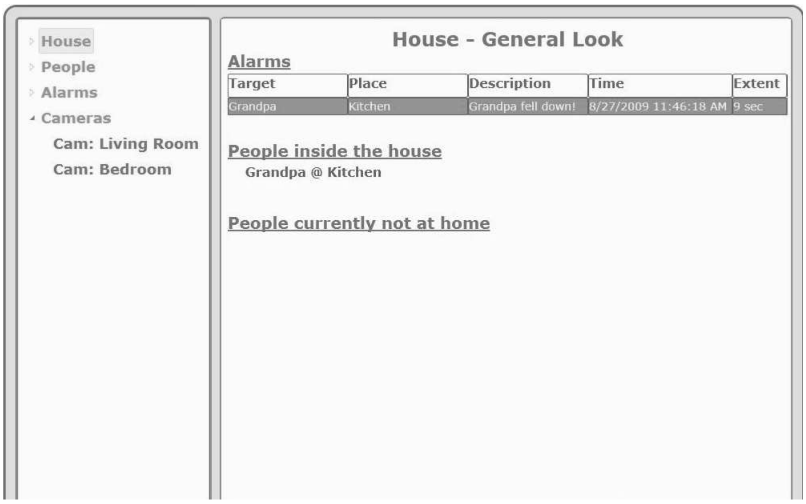


Figure 3.3. Home Screen of the Web Interface

are also listed in the bottom part of the page, in this instance, we want the system to send a text message via SMS. When an alarm is triggered, in addition to the actions specified in the alarm description, the alarm is displayed in the web interface until the user gives feedback about the alarm. Figure 3.7 shows the feedback mechanism for the alarms. In this page, the last 10 alarms are listed as well as any alarms that have not been checked. The user has the ability to give positive feedback stating that the alarm was correct and the necessary actions were taken, or negative feedback stating that the alarm was false. The false alarm feedback may then be used to improve the performance of the module that generated the false alarm.

In this thesis, a fall detection mechanism using a wearable accelerometer is designed for use with the WeCare system. The mechanism is going to be added to the WeCare prototype as an optional predefined alarm for the accelerometers associated with people. Using the regular alarm definition mechanism detailed earlier, users of the system will be able to activate the fall detection alarm mechanism and choose the actions to be taken in case of a fall detection. Since the fall detection mechanism will be embedded into the WeCare system, it will be able to benefit from all of its features such as providing a location-aware video stream when a fall is detected.

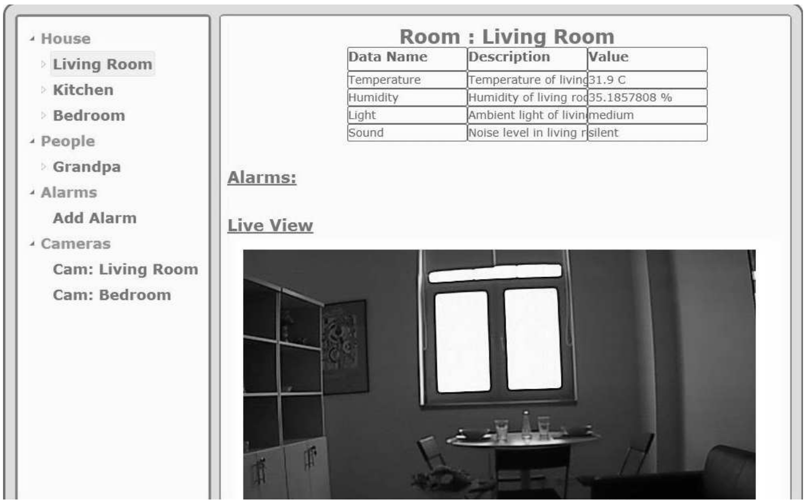


Figure 3.4. Living Room in the WeCare Prototype Interface

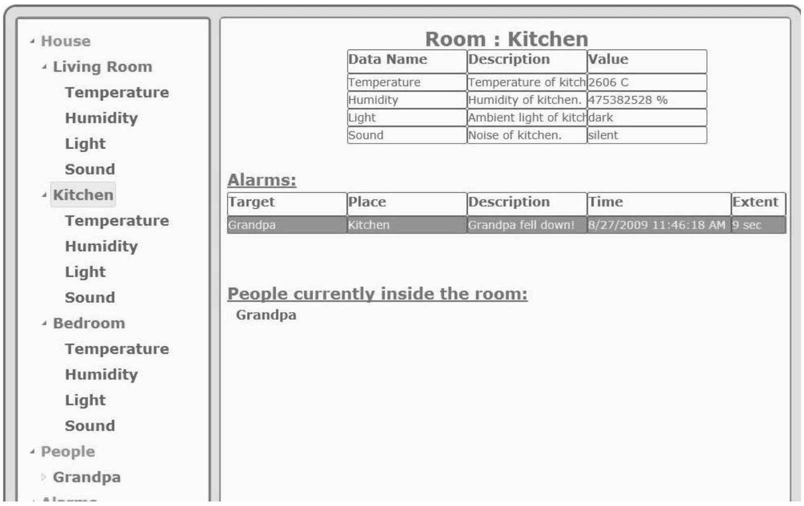


Figure 3.5. Alarm in Kitchen as displayed on the WeCare Prototype Interface

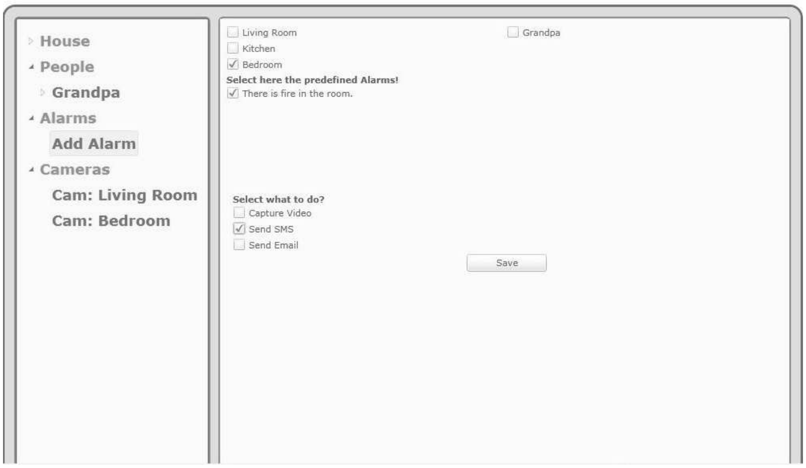


Figure 3.6. Alarm Definition via WeCare Prototype Interface

Time	Place	Extent	Actions	
6/26/2009 4:50:11 PM	Kitchen	12 sec	EMAIL	Checked.
6/26/2009 5:28:20 PM	Kitchen	10 sec	EMAIL	False Alarm.
6/26/2009 5:35:02 PM	Kitchen	10 sec	EMAIL	Checked.
6/26/2009 5:53:27 PM	Kitchen	10 sec	EMAIL	Checked.
6/26/2009 6:01:35 PM	Kitchen	9 sec	EMAIL	False Alarm.
6/26/2009 6:01:50 PM	Kitchen	2 sec	EMAIL	Checked.
8/27/2009 11:21:15 AM	Kitchen	11 sec	EMAIL	False Alarm.
8/27/2009 11:46:18 AM	Kitchen	9 sec	EMAIL	OK FALSE

Figure 3.7. Alarm Feedback via WeCare Web Interface

4. FALL DETECTION METHODS AND DISCRETE WAVELET TRANSFORM

Before our discussion on DWT and its use as a feature extraction method, it may be beneficial to give the structure of fall detection process we used in our initial studies and the findings of these studies, as they will give better intuition on the subject and clarify the need for better detection algorithms.

4.1. Fall Detection Process

In our studies we used WSN nodes with integrated accelerometers as our wearable sensors. The users were able to wear these sensors on their chests using improvised vests or on their waists or ankles using adjustable belts. The sensors provided a three dimensional discrete signal for acceleration on three perpendicular axes which were then transferred via wireless link to another WSN node which acted as a data sink and was connected to a Personal Computer (PC). Although, in our studies this PC was mainly used to record the data for later processing, it was considered to be running the fall detection algorithm in an actual implementation. In that case, this PC will also be able to provide notification services for the caregivers, as in the WeCare environment. It should be noted that, running the fall detection algorithm on the sensor platforms may be possible or even desirable due to battery and communication constraints. The decision about which system component should run the fall detection algorithm should be made after considering these constraints as well as the soft real time requirements and other factors that may effect the wireless communication link, such as existence of components that use the same wireless link.

4.2. Earlier Fall Detection Methods

In our initial studies, we implemented several fall detection algorithms and conducted experiments in order to compare the performances of these algorithms. In this section, we briefly describe each of these algorithms and then present then results of these preliminary experiments.

4.2.1. Euclidian Distance Based Fall Detection Method

Euclidian distance based fall detection method uses a training dataset which consists of three dimensional discrete accelaration signals recorded during falls and non-fall actions. From this training set, a three dimensional template vector is calculated, by taking the average of the acceleration samples associated with the falls. Let v be the template vector calculated in this manner and $\vec{a}_t$ be the acceleration vector for time t. Then we calculate the euclidian distance between v and $\vec{a}_t$ as:

$$d = \sqrt{(v_x - Acc\_x_t)^2 + (v_y - Acc\_y_t)^2 + (v_z - Acc\_z_t)^2} \tag{4.1}$$

A fall is detected if d is below a predefined value, which was 0.7 in our initial studies.

4.2.2. Naive Bayes Based Fall Detection Method

Naive Bayes based fall detection method utilized a Naive Bayes classifier that was trained for two classes, normal and fall, from a training dataset similar to that of euclidian distance based fall detection method. Let C_1 be the class of normal actions and C_2 be the class of falls. Naive Bayes classifier makes the assumption that each class produces acceleration samples that are normally distributed with mean μ_i and variance σ_i for C_i. Using the training set, these parameters are calculated for both classes, which enable us to calculate the probability of observing a specific acceleration vector, $\vec{a}$, during a specific action, i.e. $P(\vec{a}\,|C_i)$. From the training set, the probability of observing a specific class, i.e. $P(C_i)$, is calculated as the ratio of the number of samples in that class to the number of all samples. Using these values, probability of each class given an acceleration sample, i.e. $P(C_i|\,\vec{a})$, can be calculated as:

$$P(C_i|\,\vec{a}) = \frac{P(\vec{a}\,|C_i) \cdot P(C_i)}{P(\vec{a})} \tag{4.2}$$

$$P(\vec{a}) = \sum_k P(\vec{a}\,|C_k) \cdot P(C_k) \tag{4.3}$$

A fall is detected if probability of fall for an acceleration sample is greater than the probability of normal action, i.e. $P(C_2|\,\vec{a}) > P(C_1|\,\vec{a})$.

4.2.3. Fall Detection by Infrequent Pattern Discovery

This method is based on infrequent pattern discovery technique which is a data mining tool that is described in [52]. Basically, this method analyzes a training dataset in order to learn frequent patterns and then, uses this knowledge to detect infrequent patterns in the test dataset or the actual signal. For this purpose a training dataset consisting of three dimensional acceleration signals recorded during only non-fall actions is used. The algorithm calculates the euclidian distances between all subsequent acceleration samples. These distances are recorded as well as the number of occurances for each observed distance. The same process of calculating distances between subsequent samples is repeated for the test dataset, or the actual signal, and when we observe a distance that has not been recorded during training we detect a fall.

Since acceleration signals are noisy, using exact matches for distances observed in training and test sets may have a deteriorating effect on the performance of fall detection, as it would not consider two values, with only a neglicable difference, the same. To counter this problem, we also introduced an error margin to the distance comparison step. In the modified algorithm, each distance observed in the test set is compared to the distances observed in the training set. If a distance that is not close to any of the distances in the training set, with the predefined error margin, is observed, a fall is detected. In our experiments, we determined the error margin to be 0.005.

4.2.4. Fall Detection by Thresholding

The final method we investigated was a thresholding based fall detection method. In this method a threshold value is determined and a fall is detected when the observed value exceeds this threshold. We first used the acceleration magnitude as the observed value and determined the threshold as the value that detected all fall event while keeping false alarms at the minimum. Since the acceleration signal is ambiguous, it was possible to observe similar acceleration magnitudes from non-fall actions. To counter this, we applied the threshold only to a single axis which was determined to be the major axis.

4.3. Initial Results

In order to evaluate and compare performances of these methods, we conducted experiments in the Ambient Assisted Living Laboratory. Three male and two female subjects were asked to repeat a predefined motion scenario while wearing two sensors, one on their chests and one on their waists. Acceleration signals from these sensors were transmitted to a PC over the wireless link and recorded in the computer for later processing. Scenarios were repeated 10 to 20 times by each subject summing up to a total of 81 repetitions. Figure 4.1 displays the acceleration signal gathered from the sensor worn on the chest during one of these repetitions. Twenty per cent of these repetitions were separated as the training set and the rest was used as the test set.

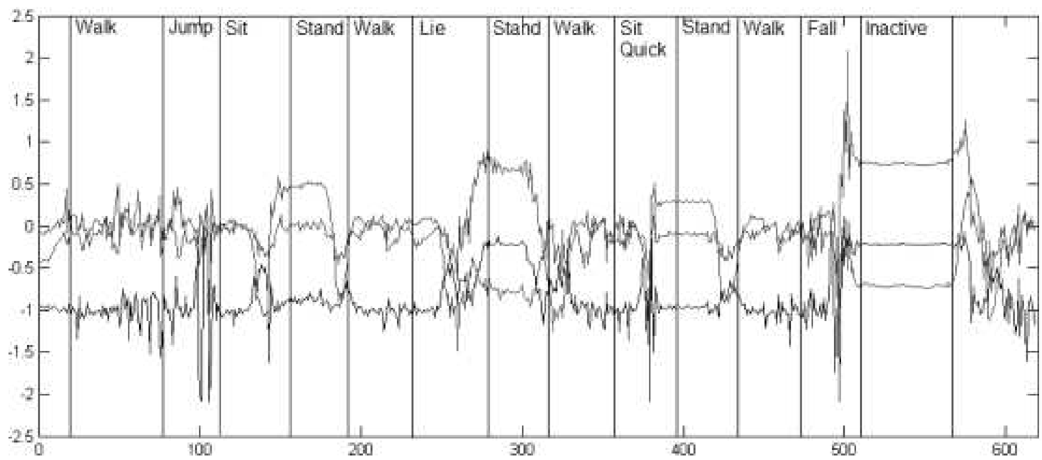


Figure 4.1. Acceleration Signal of the Chest Sensor During a Scenario

When the algorithms were run on the datasets, we acquired better performance on the dataset of chest-worn sensor, therefore we only focused on the preformance of chest-worn sensors. Recall and precision was used as the performance metrics. Euclidian distance based fall detection method provided 77 per cent recall and 29 per cent precision. Naive Bayes classifier based fall detection method yielded 98 per cent recall and 20 per cent precision. Using the infrequent pattern discovery technique we acquired 92 per cent recall and 16 per cent precision, while the modified version of this technique yielded 100 per cent recall and 18 per cent precision. In fall detection using thresholding we acquired 100 per cent recall and 21 per cent precision. The best performance was avhieved by the modified version of the thresholding algorithm, in which the threshold was applied to the major axis, with 100 per cent recall and 34 per cent precision [7].

These results, also summarized in Table 4.1, show that we could achieve high detection rates for each method, up to 100 per cent for thresholding and infrequent pattern discovery based methods. However, all of the implemented fall detection methods suffered from many false alarms they generated. The best performing method was the modified thresholding algorithm and only 34 per cent of the alarms generated by that algorithm corresponded to actual fall events. These poor performance values indicated that we needed better methods for fall detection. We, then, focused on discrete wavelet transform as it addressed shortcomings of these initial methods, such as discarding temporal relations, and not suffer from the ambiguities in the acceleration signal.

Table 4.1. Recall and Precision for Different Fall Detection Methods

Method	**Recall (%)**	**Precision (%)**
Euclidian Distance Based	77	29
Naive Bayes Based	98	20
Infrequent Pattern Based	92	16
Infrequent Pattern Based (Modified)	100	18
Thresholding	100	21
Thresholding (Modified)	100	34

4.4. Discrete Wavelet Transform

4.4.1. Issues Related to Continuous Wavelet Transform

Discrete wavelet transform is a tool that has been developed to address some issues related to continuous wavelet transform (CWT). Therefore, it is crucial to give an overview of CWT and its shortcomings that resulted in the development of discrete wavelet transform before explaining discrete wavelet transform and our use of DWT as a feature extraction method.

Wavelet transform of a signal is a scale-time analysis of the given signal, where each scale corresponds to a frequency band in the Fourier domain [1, 53, 54]. CWT takes a function, $f(t)$, that is defined in time domain and decomposes it into a two dimensional function $\gamma(s, \tau)$, which is defined over scaling parameter s and translation parameter τ, using dilations and translations of a function called mother wavelet, ψ. Let $\psi_{s,\tau}$ be the dilation of the mother wavelet ψ by s and translation of it by τ such that:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \tag{4.4}$$

Then the CWT of a function, $f(t), f : \Re \rightarrow \Re$ is defined as:

$$\gamma(s,\tau) = \int f(t)\psi_{s,\tau}(t)dt \tag{4.5}$$

In order to illustrate the transformation process, we give an example mother wavelet in Figure 4.2 with two of its dilations and translations. Using this mother wavelet, CWT takes the function given in Figure 4.3(a), which was chosen arbitrarily, and transforms it into a two dimensional function of scale s, and translation τ. In Figure 4.3(b), scales one to 640 of the wavelet transformation are displayed. The bright parts indicate scales where the energy is concentrated at a given time.

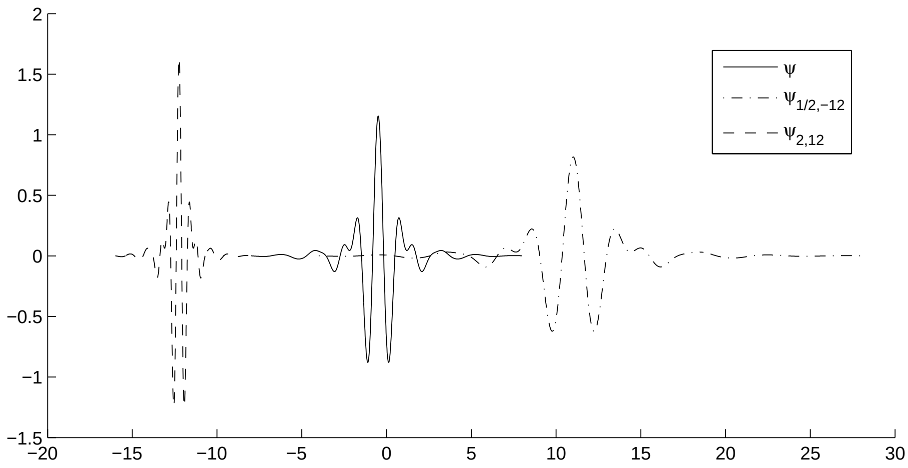


Figure 4.2. Translations And Dilations of a Mother Wavelet

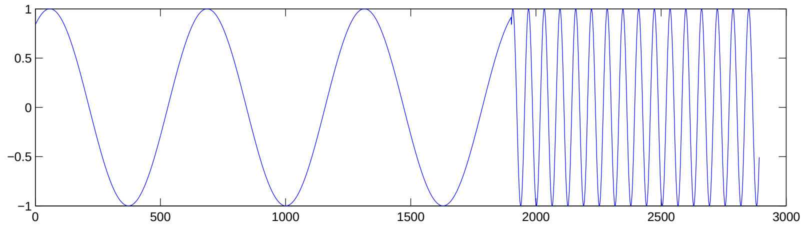
Any arbitrary function with zero average (Equation 4.6) can be chosen as the mother wavelet, ψ, however it is generally desired that the mother wavelet satisfies the admissibility condition given in Equation 4.7. If the mother wavelet satisfies the admissibility condition, the original signal can be reconstructed from the wavelet transform of that signal.

$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \tag{4.6}$$

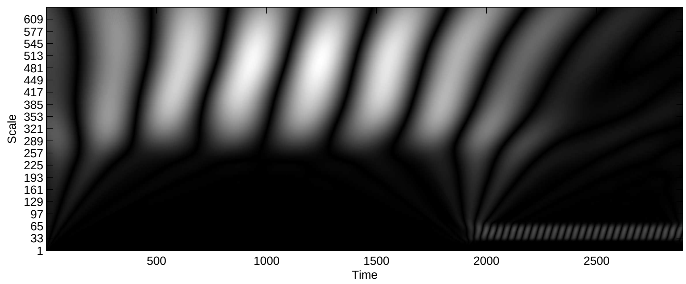
$$\int \frac{|\hat{\psi}(w)|^2}{|w|} dw < +\infty \tag{4.7}$$

The *admissibility condition* implies that $|\hat{\psi}(0)|^2 = 0$, which means the mother wavelet should have a spectrum that resembles a band-pass filter spectrum in the Fourier domain, i.e. the frequency spectrum of the mother wavelet should be a finite interval excluding zero in the Fourier domain [53].

As mentioned earlier, there are three issues that make CWT impractical, if not impossible, in real life applications. These can be stated as:



(a) An Arbitrary Function



(b) Its Continuous Wavelet Transform

Figure 4.3. An Arbitrary Function and its CWT

- Redundancy: Using continuous scalings and translations of the mother wavelet in decomposition causes redundancy in the transformation, as it cannot be guaranteed that the frequency components of each scale will be distinct.
- Infinity: CWT requires infinite translations and scalings of the given mother wavelet.
- Lack of Analytical Solution: It may be impossible to take the integral given in Equation 4.5 because of one of the two reasons. First, the integral may not have an analytical solution for some functions. Second, we may not have the function itself, but instead only a discrete set of samples of it.

In order to reduce redundancy, we may use discrete wavelets instead of continuous wavelets. This will reduce the redundancy in the resulting transform as the function will no longer be transformed into a two dimensional continuous function but instead it will be transformed into a discrete two dimensional function. The new wavelet generating function in this case would be:

$$\psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}}\psi\left(\frac{t - k\tau_0 s_0^j}{s_0^j}\right) \tag{4.8}$$

where $j, k \in Z$ and $s_0 > 1$ is a fixed dilation step. τ_0 is the translation factor related to s_0. Now that we have discrete wavelets, we have eliminated some redundancy in the transform. However, we still cannot guarantee that there is no redundancy in the transform. This can only be achieved if the discrete wavelets we use in the transform are orthonormal, meaning:

$$\int \psi_{j,k}(t)\psi^*_{m,n}(t)dt = \begin{cases} 1, & if(j,k) = (m,n) \\ 0, & otherwise \end{cases} \tag{4.9}$$

This property can be satisfied with a proper selection of the mother wavelet, ψ, and eliminates the redundancies that exist in the wavelet transform. However, we still have infinite number of wavelets that we need to generate from the mother wavelet. Although the number of translations that we can use is limited by the length of the signal we are analyzing, there is an unlimited number of scales.

Recall that Equation 4.7 implied that the wavelets acted like bandpass filters. It is known that compressing a signal in time domain is equivalent to stretching and shifting up the signal in the frequency domain. As a result, dilating the wavelet by a factor of two would squeeze the frequency spectrum of the wavelet by a factor of two and shift all frequency components down by a factor of two. This property enables us to cover the finite frequency spectrum of the original signal with the spectra of dilated wavelets. The wavelets can be arranged to "touch" each other in order to get a good coverage, as seen in Figure 4.4, and can be considered as a band-pass filter bank. Covering the entire spectrum of the original signal using only the dilations of wavelet

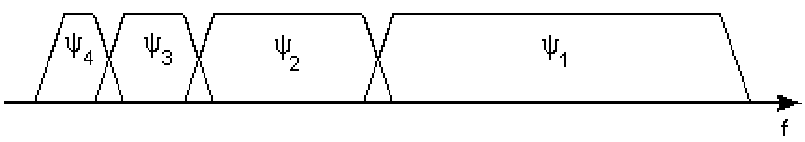


Figure 4.4. Wavelet filterbank in Fourier Domain [1]

function still requires infinite number of dilations. Instead of using only wavelets, we can use another function, spectrum of which covers the low frequency components of the signal so that we can use only a finite number of wavelets. Since we want that function to cover the spectrum that is covered by the infinite number of dilations of the mother wavelet, it can be defined via its desired spectrum as:

$$|\hat{\phi}(w)|^2 = \int_1^{\infty} |\psi(sw)|^2 \frac{ds}{s} \tag{4.10}$$

ϕ is called the scaling function and its dilations can be used to cover the low frequency spectrum of the signal, while covering the high frequency spectrum of the signal by the spectra of dilations of the mother wavelet. Utilizing ϕ we can limit the wavelets we have to use to a finite number. Figure 4.5 shows the spectrum of the scaling function covering the spectra of infinitely many wavelet functions. Using ϕ and ψ we can

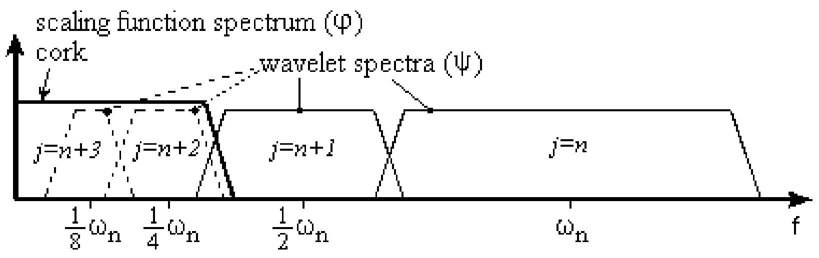


Figure 4.5. Wavelet filterbank in Fourier Domain with Scaling Spectrum [1]

calculate the low frequency and high frequency components of the signal at scale j as $A_j =< f(t), \phi_{j,k}(t) >$ and $D_j =< f(t), \psi_{j,k}(t) >$. A_j and D_j are called approximation coefficients and detail coefficients for scale j, respectively.

Since the spectrum of ϕ is designed to cover spectra of infinitely many wavelets, we can add a wavelet spectrum to the spectrum of ϕ or remove a wavelet spectrum from it. Removing a wavelet spectrum from the spectrum of the scaling function is equivalent to dilating the scaling function by a factor of two. One important result of this fact is that, scaling function at a scale has all the information to construct the scaling function for the next scale. This is called the *twin scale relationship* and can be expressed as:

$$\phi(2^{j+1}t) = \sum_k h_j(k)\phi(2^j t - k) \tag{4.11}$$

Similarly:

$$\psi(2^{j+1}t) = \sum_k g_j(k)\phi(2^j t - k) \tag{4.12}$$

When we use 4.11 and 4.12 and insert suitably scaled and translated versions of ϕ and ψ into the inner products that calculate the coefficients, the inner products become:

$$A_{j+1}(p) = \sum_k h(k-2p)A_j(k) \tag{4.13}$$

$$D_{j+1}(p) = \sum_k g(k-2p)A_j(k) \tag{4.14}$$

These equations indicate that the coefficients for a scale can be computed by a weighted sum of the approximation coefficients of the previous scale. Since approximation coefficients are the lower frequency components, h can be seen as a *low-pass filter* and similarly g can be seen as a *high-pass filter*. The indices of g and h have a coefficient of two in terms of p, resulting in the fact that only half of the points in A_j are used in the given computation. It should be noted here that, proper selection of ψ, therefore ϕ, may yield finite sequences as h and g. In the light of these, we can rewrite Equations 4.13 and 4.14 as:

$$A_{j+1} = (A_j * h) \downarrow 2 \tag{4.15}$$

$$D_{j+1} = (A_j * g) \downarrow 2 \tag{4.16}$$

where $(x * y)$ is the convolution of x and y and $x \downarrow 2$ is the downsampling of x by a factor of two, i.e. discarding every other sample. These two equations form the basis of the algorithm known as the fast wavelet transform which is also employed in this study.

4.4.2. Discrete Wavelet Transform as a Feature Extraction Method

Discrete wavelet transform begins with the choice of the mother wavelet, ψ, and the generation of the corresponding scaling function, ϕ, using Relation 4.9. Using ψ and ϕ, we then generate the filters h and g via Relations 4.11 and 4.12. Recall from the previous subsection that any function that satisfies the *admissibility* condition can be chosen as a mother wavelet. However, as there are many well-studied and well-known proper wavelet families, we are going to limit ourselves to the investigation of these existing mother wavelets. Assuming that we have a mother wavelet and the related filters h and g, our feature extraction method can be formally expressed as:

$$A_0 = MP(X) \tag{4.17}$$

$$\left. \begin{aligned} A_i &= (A_{i-1} * h) \downarrow 2 \\ D_i &= (A_{i-1} * g) \downarrow 2 \end{aligned} \right|_{1 \leq i \leq J} \tag{4.18}$$

If we define $F(Y)$ to be our set of coefficients resulting from the DWT of Y such that:

$$F(Y) = \{A_J, D_i \mid 1 \leq i \leq J\} \tag{4.19}$$

then $F(MP(X))$ is the set of extracted features.

Wavelet families may contain multiple related mother wavelets, that are distinguished from each other by a parameter. For instance, the family of Daubechies wavelets contain 20 mother wavelets and each wavelet is distinguished by the number of vanishing moments, which also effects the length of the resulting filters. Some wavelet families on the other hand have only a single mother wavelet, which can be sampled in different intervals and lengths to obtain different filters from the family, e.g. Meyer family [53]. Different wavelets have different concentrations in time the axis and the frequency axis, therefore their resolutions in the frequency axis vary as well as their capability of localization in the time axis.

4.4.3. Computational Complexity of DWT Based Fall Detection

First, assume that we want to calculate the DWT of a discrete signal of length N. Also assume that we are using analysis filters, h and g, of length l in the transformation. Then, the convolutions in Equation 4.18 can be computed with approximately $N \cdot \frac{l}{2}$ multiplications each, so the first level of the transform requires approximately $N \cdot l$ multiplications. Since we downsample the approximation coefficients at each level, every level requires half the number of multiplications as their previous level. Therefore the total number of multiplications required to compute the discrete wavelet transform for infinite number of levels becomes:

$$N \cdot l + \frac{N \cdot l}{2} + \frac{N \cdot l}{4} + \frac{N \cdot l}{8} + \cdots = N \cdot l \cdot \left(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots\right) = 2 \cdot N \cdot l \quad (4.20)$$

which shows that the computational complexity of DWT is linear in the length of the signal. However, the signal may quickly become very long in a fall detection system, as the fall detection task requires constant monitoring, One approach that can be taken to address this issue is to use a windowing scheme. By using a window of length w, we discard all the acceleration samples except the most recent w samples. This effectively limits the signal length to w and the required number of multiplications becomes $2 \cdot w \cdot l$.

Since fall detection is a real-time problem, we need to carry out DWT calculations frequently. The frequency of DWT calculations need not be higher than the sampling frequency as acceleration signal does not change between samplings. Therefore, with a sampling interval of α, we need to carry out DWT calculations $\frac{1}{\alpha}$ times a second, resulting in $2 \cdot \frac{1}{\alpha} \cdot w \cdot l$ multiplications a second.

It was mentioned earlier that the decision about which system component should run the algorithm may depend on the complexity of the algorithm. To exemplify, we may consider a fall detection system with a window size of 20, a filter size of 15 and a sampling rate of 10 Hz. For such a system, we would require $2 \cdot 20 \cdot 15 \cdot 10$ or 6000 multiplications per second, which is a relatively small number even for the wireless sensor nodes. Considering the heavy burden of radio communication on the battery life of the wireless sensors, carrying out the processing on the wireless sensor seems to be more viable. Although, the extra load caused by the constant processing could be considered an energy issue, we do not consider it to be of great importance, as the elderly may be asked to charge the sensor daily.

4.4.4. Fall Detection System Architecture

Before moving on to the experiments, we are going to describe a fall detection system based on DWT in order to form a complete picture. Figure 4.6 displays a schematic representation of a fall detection system together with the intended users of the system. The elderly wears a battery operated wireless sensor platform that

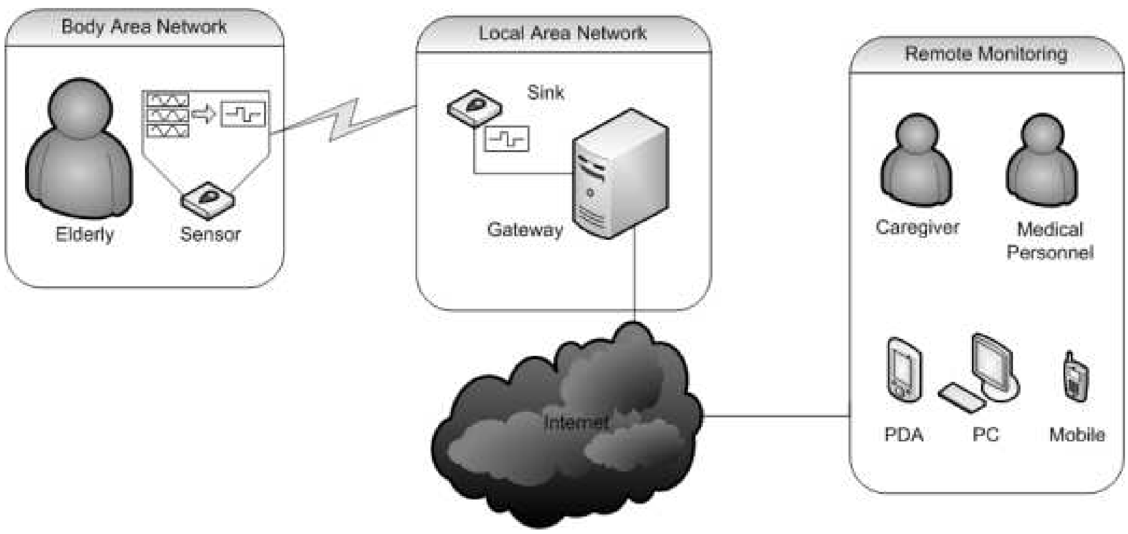


Figure 4.6. Fall Detection System Architecture

is either integrated to his/her clothes or is designed to be worn as an accesory. The sensor platform has an integrated three-axial accelerometer which is sampled at regular intervals. Then the dimensions of the acceleration sample is reduced from three to one, either by discarding two axes or by calculating a mapping, and the resulting sample is inserted into a buffer with a fixed length. The buffer holds the most recent acceleration samples in chronological order. After each sample insertion to the buffer, DWT is applied to the signal in the buffer on the sensor platform. The DWT coefficient are then passed onto the classifier which detects fall based on these coefficients. An example classifier may be the basic thresholding method, as will be the case in the Experimental Results chapter, which checks if the detail coefficients at the first level exceed a pre-defined threshold. If the classifier detects a fall, sensor platform transmits a message to a sink node via wireless link. The sink node is a WSN node that is connected to PC. It notifies the PC via the wired connection between them. Finally, the PC notifies the caregivers by sending e-mails and short messages via SMS and takes any other necessary actions.

Figure 4.7 shows an example of the acceleration signal, the curve at the top, and detail coefficient of the first six levels of its DWT. First thing to note on this graph is that, the Time axis is the sample index. Since detail coefficients at each level are of different sizes, they were scaled in this graph to match the time indices of the original signal. In the figure, a fall occurs around 500 and it can be noticed on the original signal with a sudden acceleration impulse. It can seen that, the amplitudes in the first and second level coefficients can be used to effectively distinguish the fall from other non-fall actions. This observation led to our decision of using thresholding in our experiments. However, different classifiers may be used instead of thresholding, as a future direction we are going to include new classifiers in our studies.

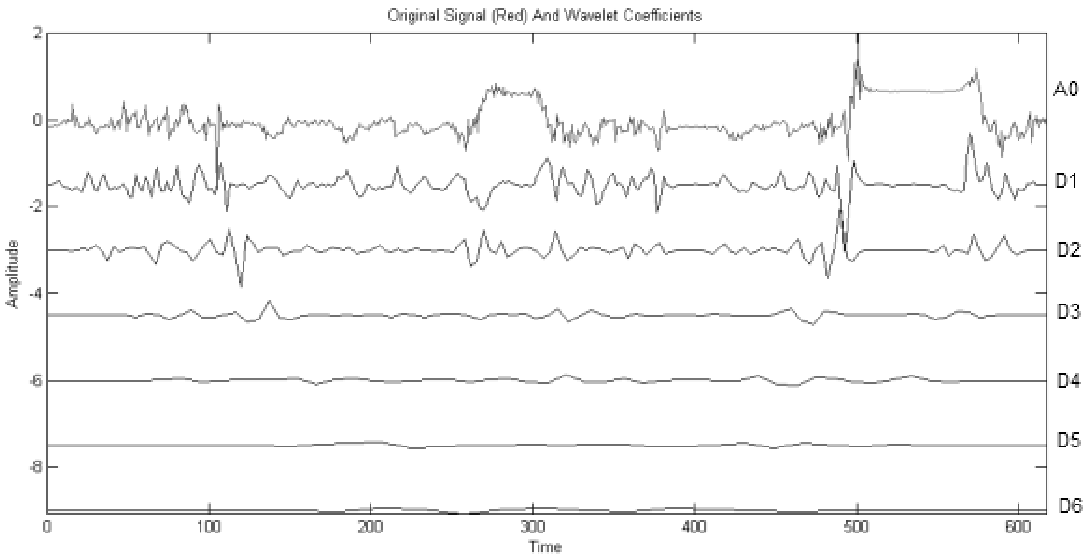


Figure 4.7. An Acceleration Signal and Detail Coefficients of its DWT

5. EXPERIMENTAL RESULTS

As discussed earlier, the selection of the mother wavelet governs the frequency components that reside in each scale as well as the resolution of localization in the time domain. Therefore, the appropriate selection of a mother wavelet may have a significant impact on the performance of the wavelet transform based fall detectors. Since it is hard, if not impossible, to come up with a mathematical model for falls, taking an analytical approach to wavelet selection does not seem plausible. Hence, we are going to use an experimental approach to the problem of finding a suitable mother wavelet for the task of fall detection.

Our experiments consist of four sections: *data collection*, *feature extraction*, *fall detection* and *performance evaluation*. In the *data collection* part, voluntary subjects are asked to repeat a predefined motion scenario which includes normal daily actions, such as walking and sitting down, normal actions that can be confused with falls, such as jumping and lying down as well as falls. An automated data collection program was developed for this study, which instructs the users which action he/she needs to perform according to the scenario given in Table 5.1. During each repetition, acceleration data is collected from an accelerometer that the subject is wearing and recorded on a remote computer together with the label of action the subject is performing. The reason that we use an offline processing scheme is its flexibility and practicality. Moreover, this scheme allows us to observe the effect of the parameters on the same physical phenomenon.

In the *feature extraction* part, we apply DWT to the data we have recorded during the *data collection* part. Then, the extracted features are passed to a classifier which detects falls using the feature set. Although our goal was to evaluate the performance of DWT as a feature extraction method, we need a classifier, because evaluating the performance of a feature extraction method requires us to investigate the separation of features related with falls from that of non-fall actions. The classifier used in our experiments was a thresholding method, which makes the fall decision for time t, $FD(t)$

Table 5.1. The Motion Scenario Used in Experiments

Order	**Action**
1	Walking
2	Jumping
3	Sitting Down
4	Getting Up
5	Walking
6	Lying Down
7	Getting Up
8	Walking
9	Quickly Sitting Down
10	Getting Up
11	Walking
12	Falling Down
13	Lying Still

as:

$$FD(t) = \begin{cases} 1, & if\ D_1 \left\lfloor \frac{t}{2} \right\rfloor\ >\ \theta \\ 0, & otherwise \end{cases} \tag{5.1}$$

where θ is a predefined threshold and $\lfloor\ \rfloor$ is the floor function.

The decisions made by the classifier are then used to compute performance metrics, $Recall$, $Precision$ and $F - value$ that are defined in A.

5.1. Two-Phase Experiment Design

We designed our experiments according to a two-phase experiment design scheme [55]. In a two-phase experiment, the factors that may affect the outcome of the process, which is to be investigated, are identified. Then, in the first phase of the experiment, all the factors that can be investigated are included. Some factors may not be investigated due to several reasons, for example in the fall detection case, we cannot risk demanding a person over 70 to fall, therefore we cannot investigate the effect of old age. In the first phase, only a small number of possible values are investigated for each factor and these values are generally selected to be one high value and one low value. Then in the second phase of the experiment, the factors that affected the outcome the most are investigated further with more levels.

After our initial studies, we identified several factors that may affect the performance of a fall detector system that we can investigate:

- Sensor Hardware is the device that is used to collect data. It may have an effect on the performance as different sensors may have different accuracy and noise characteristics.
- Sensor Location is the part of body on which the sensor will be worn. It may affect the acceleration the sensor will be subjected to. For instance, a sensor worn on the chest may have a different acceleration signal than a sensor worn on the ankle during a fall.
- Sampling Rate is the rate with which data is gathered. It affects the ability of the accelerometer to capture the necessary information for fall detection.
- Wavelet Family together with *Wavelet Parameter* determines the time-frequency resolution of DWT, hence it affects the features extracted from the acceleration signal.
- Fall Direction is the direction towards which a person falls. It changes the characteristics of the signal produced by the accelerometer and may have an effect on the performance of a fall detection system.
- State Before Fall is the motion state of the person before the fall. The person may be moving or standing still before falling and this may have an effect on the acceleration signal.

5.1.1. Phase One of the Experiments

Up to now, we have determined the factors that may affect the outcome of a fall detector that uses DWT as a feature extraction method. The investigated values related with these factors are listed in Table 5.2 and Table 5.3.

Table 5.2. The Factors and Their Values for the First Phase of the Experiments

Factor	**Value**
Sensor Hardware	Imote2.NET, SunSPOT
Sampling Rate	100Hz,10Hz
Sensor Location	Chest,Waist, Ankle
Fall Direction	Frontal, to the Right
State Before Fall	Walking, Standing
Wavelet Family	Shown in Table 5.3
Wavelet Parameter	Shown in Table 5.3

Table 5.3. The Wavelet Families and Related Parameters For The First Phase

Family	**Parameter**
Daubechies	4,8,12
Morlet	12,24,36,48
Symlet	12,24,36
Meyer	4,5,6
Coiflet	1,3,5
Gaussian	12,24,36,48
Mexican Hat	12,24,36,48
Biorthogonal	(2.)2,4,6,8,(3.)1,3,5,7
Reverse Biorthogonal	(2.)2,4,6,8,(3.)1,3,5,7

We used an Imote2.NET sensor node with an ITS400 sensor board attached to it [47] and a SunSPOT sensor node [56] in the first phase of our experiments. Figure 5.1 shows the two sensors and the selected axis conventions associated with each. These sensors were worn by the subject at the same time and data was collected from both of them simultaneously. Our subject was a 25 year-old healthy male. The data was collected at the rate of 100 Hz for all repetitions and then a copy of the data was downsampled by a factor of 10 in order to get the 10 Hz copy of the same signal. As for the *sensor location*, *fall direction* and *state before fall* 10 repetitions were recorded for each of the combinations of these factors, these 10 repetitions are called a set.

Hence, we have 6 sets for each sensor and a total of 24 sets including the downsampled copies of each set.

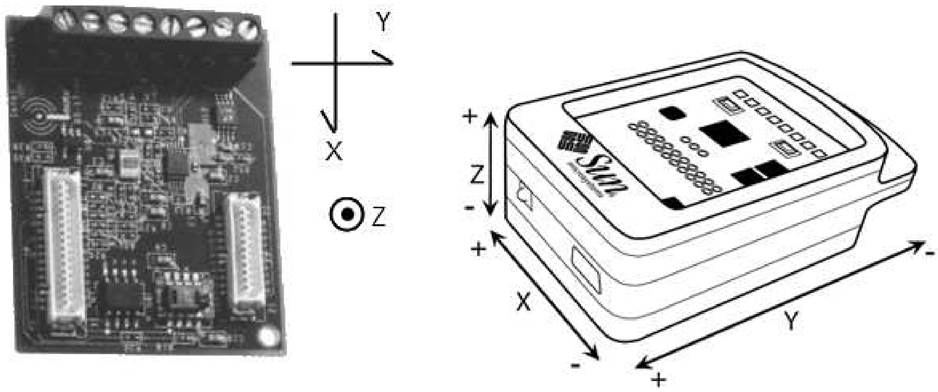


Figure 5.1. Sensors Used in Experiments and Selected Axis Conventions

Before presenting our results, it should be noted that using a classifier also may introduce one or more factors that affect the performance of fall detection. In our case, this factor is the selection of the predefined threshold. In order to eliminate the effect of this selection, we first determine a threshold interval for each set. This interval is defined such that the minimum value in the interval is the maximum threshold value that detects all falls in the set and the maximum value in the interval is the minimum threshold that produces no false alarms. Then this interval is discretized into 10 points. The value that gives the best performance when used as a threshold among these 10 points is chosen as the actual threshold. The results are presented using performance metrics related to this threshold in the remainder of the section.

<u>5.1.1.1. Effect of Sensor Hardware.</u> In order to observe the effect of the sensor hardware on fall detection performance, average values of performance metrics were calculated over all datasets for each sensor hardware was needed. For this purpose, fall detection performance metrics were calculated using all wavelet family and wavelet parameter couples over all data sets. Then the average of these performance metrics was calculated in order to get the average performance for each sensor hardware.

Average values of performance metrics calculated over all sets for each sensor hardware suggest that *sensor hardware* has a significant effect on the performance of a fall detection system. It can be seen on Table 5.4 that the use of Imote2.NET yields similar *Recall* performance to that of SunSPOT, but it has significantly better *Precision* and it increases the performance metric $F-Value$ by 17 per cent. Moreover,

Table 5.4. Average Values of Performance Metrics per Sensor Type

Sensor Type	**Recall(%)**	**Precision(%)**	**F-Value(%)**
Imote2	75	49.5	59.6
SunSPOT	75.5	30	43

as seen in Figure 5.2, the average performance of each wavelet family is higher when we use the Imote2.NET sensor, except for the *Biorthogonal*(2) family. As evidenced here, the sensor hardware may have a significant effect on the performance of a fall detector system. The most likely reason for this effect is the difference in the noise levels of the sensors. In fact, our observations have pointed out that the data received from the SunSPOT sensor is very noisy and therefore we have calibrated the sensor and used a mean-filter in order to eliminate the noise to some degree. The results shown here were acquired using this de-noised signal.

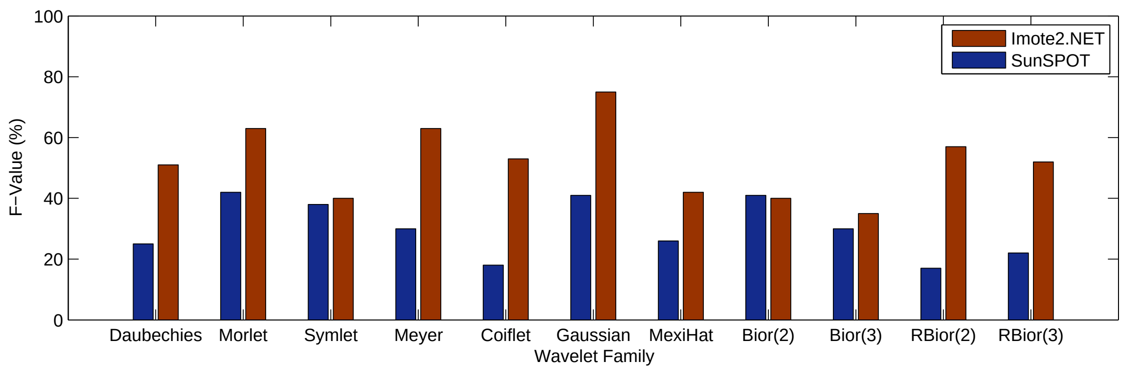


Figure 5.2. Best Average Performance of Each Wavelet Family

5.1.1.2. Effect of Sampling Rate. Our intuition on the effect of sampling rate on the performance of fall detection suggest that we would observe better performance using the 100 Hz signal than using the downsampled copy of the same signal. This was supported by the results of the experiment on the average case with using 100 Hz signal yielding an $F - Value$ of 64 per cent and using 10 Hz signal yielding an $F - Value$ of 51 per cent. However, when we investigated the best performance per wavelet family, we saw that for some wavelet families, the performance of using signals with different sampling rates were very similar. Even in one case, the Gaussian wavelet family, using the 10 Hz signal outperformed the one using 100 Hz signal as if the low frequency signal is a filtered version of the high frequency one that had less noise. These results, which are displayed in Figure 5.3, suggest that the sampling rate has a significant effect on the performance of a fall detection system.

Table 5.5. Average F-Value(%) per Sampling Rate

Sampling Rate	**F-Value(%)**
10Hz	51
100Hz	64

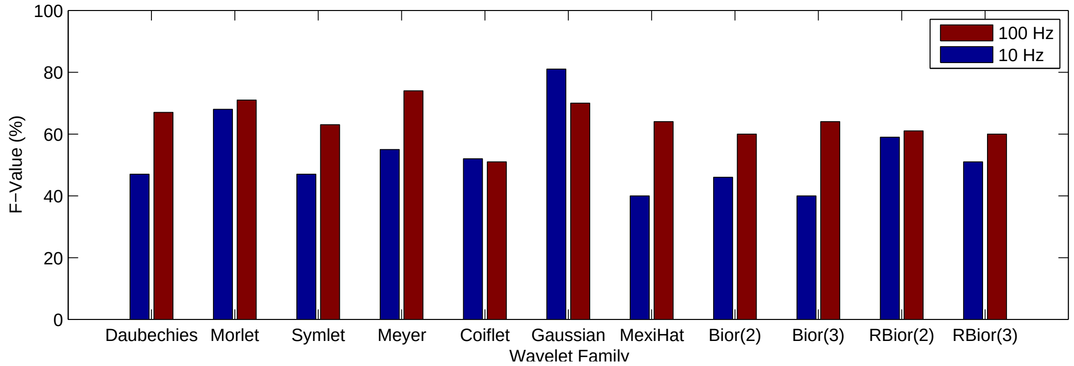
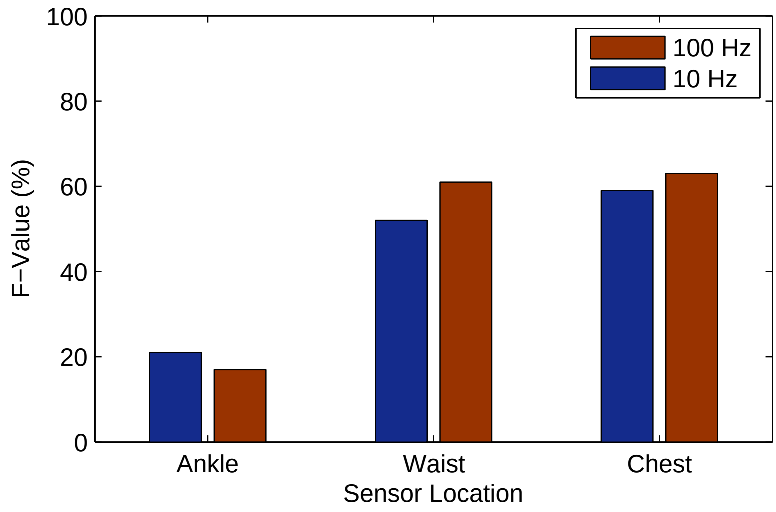
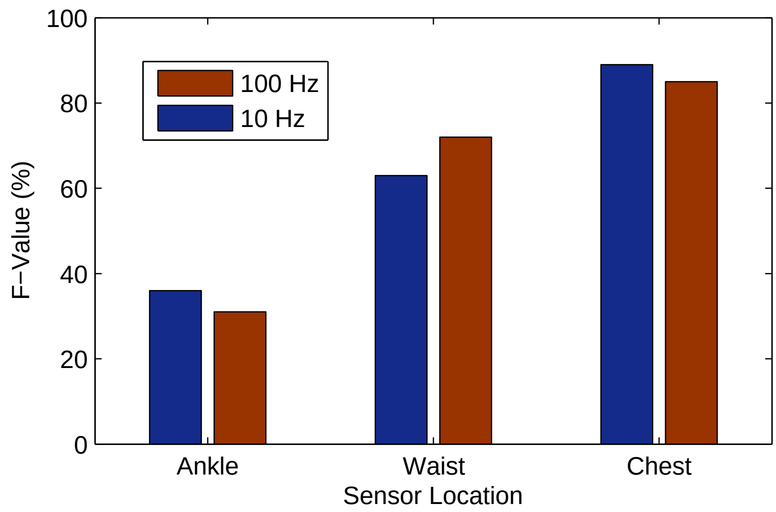


Figure 5.3. Best Average Performance of Each Wavelet Family



(a) Average Performance vs. Sensor Location



(b) Best Performance vs. Sensor Location

Figure 5.4. F-Value vs. Sensor Location

5.1.1.3. Effect of Sensor Location. Results displayed in Figure 5.4(a) show that wearing the sensor on the waist or the chest performs significantly better than wearing the sensor on the ankle on the average case. It can also be seen from Figure 5.4(a) that, while the sensor worn on the chest performs better than the sensor worn on the waist, there is only a slight difference. However, when we investigated the best performance provided by each sensor location, the difference became more apparent. Figure 5.4(b) shows the best results acquired for each sensor location, and it can be seen from the figure that using a chest-worn sensor outperforms using an ankle-worn or waist-worn sensor.

5.1.1.4. Effect of Fall Direction. Our experimental results suggest that the direction of the fall affect the information content of the signals provided by separate axes of the accelerometer worn by the subject. This implies that the axis that provides the most valuable information in detecting a fall may change according to the direction of the fall. Hence, a fall detection system that utilizes only a single axis in decision making may perform poorly for certain types of fall. This can be overcome by incorporating signals from multiple axes into the decision making process. The most basic method for including multiple acceleration axes is to calculate the magnitude of the acceleration as the Root Sum Square (RSS) of signals from each axis. As seen in Table 5.6, when the

Table 5.6. Average F-Value(%) for Different Fall Directions

Acceleration Axis Used	**Frontal**	**To The Right**
Y	60.93%	92.4%
Z	92.4%	66.7%
RSS	86.8%	84.2%

fall direction changes, the axis that contain the most significant information changes as well. Moreover, it can also be seen that using the RSS yields fall direction invariant performance, although its peak performance is lower than that of individual axes.

5.1.1.5. Effect of State Before Fall. Experimental results presented in Table 5.7 show that the motion state of the person just before the fall has an effect on the fall detection performance, such that if the subject is walking before falling, e.g. falling due to tripping, the fall is more likely to be detected.

Table 5.7. F-Value(%) for Different States Before Fall

	Walking	**Standing**
Average Case	67%	58%
Best Case	85%	67%

5.1.2. Phase Two of the Experiments

In the previous section, we have given the first phase of our experiments and their results. For the second phase of our experiments, we decided to investigate the relatively more significant subset of the factors we investigated in the first phase, namely the *sampling rate* as well as the wavelet families and parameters, Table 5.8 shows the values related to these factors. It can be seen from Table 5.9 that some families have the same parameter values for the second phase of the experiments as for the first phase of the experiments, because wavelet parameters can only assume a limited number of values for those families. We excluded the other factors from the second phase of the experiments due to the following reasons:

- Sensor Hardware: We have seen that using a specific hardware, Imote2.NET, provided significantly better results than using SunSPOT hardware. Since the hardware to be used is a system parameter that can be decided upon by the system designer, we chose to use the Imote2.NET hardware for the remaining experiments.
- Sensor Location: It was evidenced by the experiments that a chest-worn sensor provided the signal that was most suitable for fall detection rather than waist-worn or ankle-worn sensors. Similar to the sensor hardware decision, we chose to use only chest-worn sensors for the second phase.
- Fall Direction: The first phase experiments indicate that although the fall direction has an affect on the axis that contain the most significant information, this can be mitigated by using a data fusion scheme or RSS.
- State Before Fall: Our results indicate that we have a better performance in detecting falls of people that were moving before falling. Therefore using only falls from a standing still state constitutes a lower-bound for the performance of a fall detection system.

For the second phase experiments, five new sets were recorded with the participation of five new subjects. Three of the subjects were female while the rest of them were males. The average age of the subjects were 26.6 and all the subjects were healthy volunteers.

Table 5.8. The Factors and Their Values for the Second Phase

Factor	**Value**
Sampling Rate	10Hz,20Hz,25Hz,33Hz,50Hz,100Hz
Wavelet Family	Shown on 5.9
Wavelet Parameter	Shown on 5.9

Table 5.9. The Wavelet Families and Related Parameters For The Second Phase

Family	**Parameter**
Daubechies	1:12
Morlet	$\{2 \cdot i \mid i = 1 : 240\}$
Symlet	12,24
Meyer	4,5,6
Coiflet	1,3,5
Gaussian	$\{2 \cdot i \mid i = 1 : 240\}$
Mexican Hat	$\{2 \cdot i \mid i = 1 : 240\}$
Biorthogonal	(2.)2,4,6,8,(3.)1,3,5,7
Reverse Biorthogonal	(2.)2,4,6,8,(3.)1,3,5,7

Motion scenario described in Table 5.1 was used again as well as the performance metrics described in the previous section.

5.1.2.1. Sampling Rate. In the previous section, we have investigated the effect of the sampling rate on the performance of a fall detection system. We have seen that, although it had an effect on the outcome, the effect varied for different wavelet families. In order to make sure that the selection of wavelet parameters related with each wavelet family was not favoring a specific sampling rate, we repeated the same experiments with an extended set of wavelet parameters that are given in Table 5.9. We also generated 20Hz, 25Hz, 33Hz and 50Hz copies of the 100Hz signal and calculated the best $F-Value$ we achieve for each wavelet family and parameter selection. Figure 5.5 shows the $F-Value$ vs. $WaveletParameter$ plot for the Gaussian wavelet family. It can be seen from the figure that certain parameter values favor certain sampling rates.

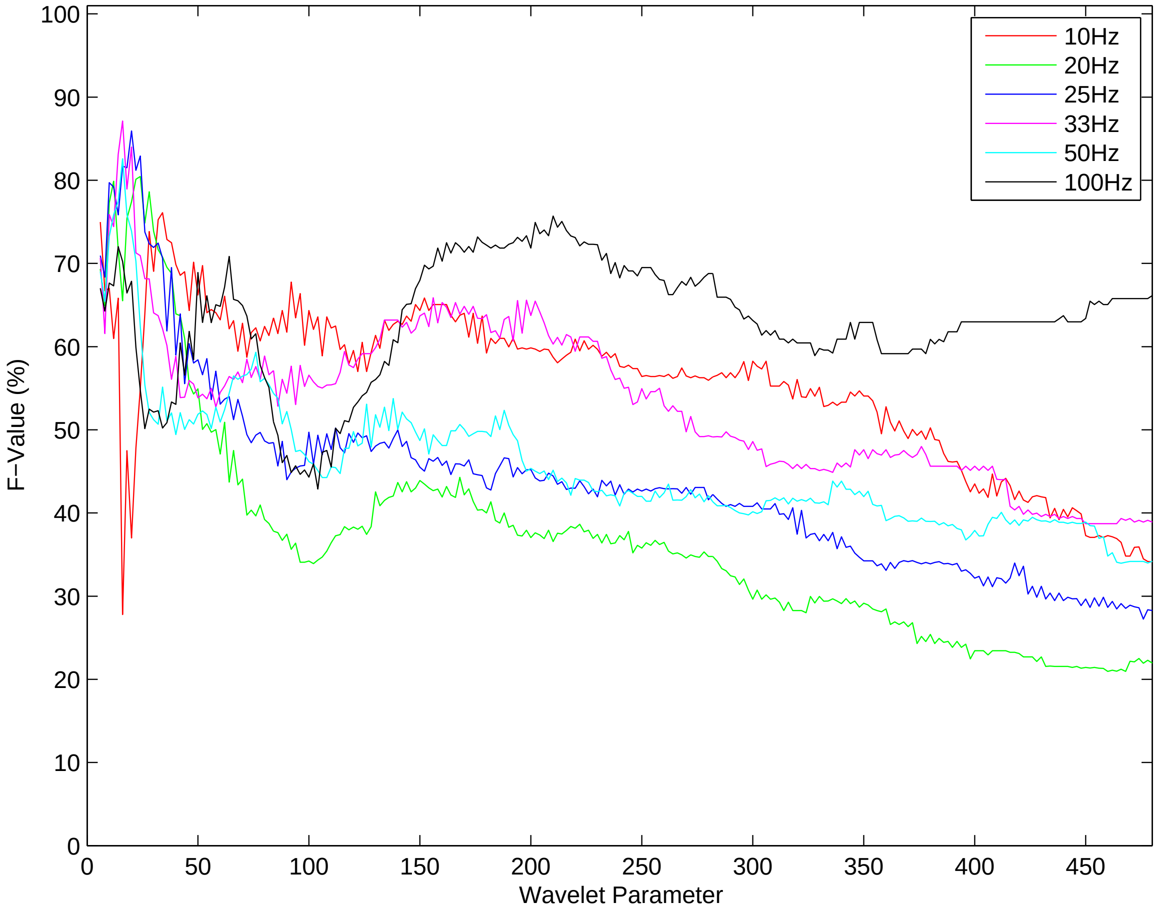


Figure 5.5. F-Value for Gaussian Wavelet Family Parameters

We are also interested in the best performance we can achieve for each sampling rate. In Figure 5.6, we see that there is an improvement in performance as we go from 10 Hz to 33 Hz then a decrease again until 100 Hz. The increase in performance until 33 Hz can be explained by the Nyquist criteria. In [57], Purwar, et al. state that most of the energy related to a fall was contained within $0 - 15$ Hz frequency band. Therefore, we need a minimum sampling rate of 30 Hz in order to fulfill the Nyquist criteria. Below this rate, we may not fully represent the fall event therefore we may lose information that is useful in the detection of falls. The decrease in the performance from 33 Hz to 100 Hz is counter-intuitive and unexpected. This decrease may be caused by the selection of wavelet parameters, i.e. we may have selected parameters that favor smaller sampling rates, or it may be caused because of the noise in the 100 Hz signal.

On the other hand, the results indicate that we obtain a similar performance using 10 Hz signal 100 Hz signals. In wireless sensor applications where the battery life is considered to be a constraint and storage capabilities of the sensor are limited, the natural choice for the sampling rate becomes 10 Hz considering that the $F-Value$ of 78 per cent is acceptable.

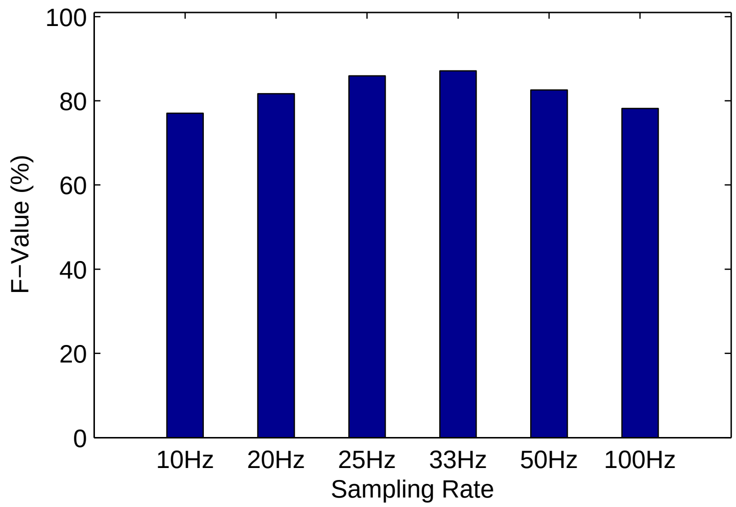


Figure 5.6. Best Performance Among All Wavelet Families for Each Sampling Rate

5.1.2.2. Wavelet Family and Parameter Selection. Until now, we have investigated and discussed the effects of the factors listed in Table 5.3 except for the wavelet family and the wavelet parameter. Since the choice of mother wavelet governs the behavior of the DWT, we expect this selection to have a significant effect on the performance of the fall detection system. Indeed, we have seen earlier in Figure 5.2 and Figure 5.3 that the selection of wavelet family can have a significant effect on the performance of the system. Furthermore, we have also seen in Figure 5.5 that the wavelet parameter selection which defines the mother wavelet that will be used together with the wavelet family, can have an effect on the performance as well. Then, we would like to distinguish the mother wavelet that is best suited for the fall detection task. Therefore, we would like to distinguish the wavelet family that gives the best performance on average.

Table 5.10 shows the average performance for each family and according to these results Gaussian family is the best suited family for the fall detection task. The wavelet parameter that yielded this performance metric was 14 for the Gaussian wavelet family. In order to understand why Gaussian wavelet family performed better than the other wavelet families, we investigated the wavelets and the acceleration signals in the Fourier domain. We saw that the fall actions had more energy in high frequencies compared to other actions and Gaussian wavelet family, particularly the mother wavelet defined by wavelet parameter of 14, was able to distinguish that frequency band better than other wavelets.

Table 5.10. Average Performace of each Wavelet Family

Wavelet Family	**Recall(%)**	**Precision(%)**	**F-Value(%)**
Daubhecies	73	34	47
Morlet	60	77	68
Symlet	70	36	47
Meyer	38	100	55
Coiflet	73	40	52
Gaussian	70	98	81
Mexican Hat	75	38	40
Biorthogonal(2)	70	34	46
Biorthogonal(3)	80	26	40
R.Biorthogonal(2)	58	61	59
R.Biorthogonal(3)	68	41	51

6. CONCLUSION

Falls constitute a major health risk in the lives of the elderly people, and therefore fall events are considered a major obstacle to independent living of the elderly people. With the aging population, the need for remote monitoring systems and fall detection methods rises. In this thesis, we surveyed the research on automatic fall detection using several types of sensors, namely video sensors, acoustics sensors and wearable sensors. Results of our survey suggest that wearable sensor based methods, especially wearable accelerometer based methods in particular, received much more attention since they offer more robust fall detection capabilities while maintaining the privacy of the users.

We described WeCare testbed which was built in a 55 m^2 laboratory. WeCare system is a remote health monitoring system with multi-modal sensing capabilities such as video, RFID and ambient sensing. The system has the ability to detect events based on user defined alarms and notify the caregivers about the detected events. Since the WeCare environment has the ability to localize the elderly person in the home, it can deliver accurate video streams of the elderly person or the event location via its web interface. Moreover, the SMS capabilities of the system makes rapid notification of caregivers possible, which may prove to be vital in some situations such as a fall. We also described how a fall detection mechanism was implemented and integrated to the WeCare prototype.

We described a wavelet transform based fall detection method using wearable accelerometers. The fall detection method employed DWT as a feature extraction method. Coefficients of DWT of the acceleration signal are used to detect rapid local variations that can be used to detect falls. We conducted multiple experiments in order to determine the wavelet family that is most suitable for fall detection. In our experiments, we investigated the effects of several factors including: Sampling Rate, Fall Direction, Sensor Location, Sensor Hardware and State Before Fall. Our results indicate that the proposed fall detection algorithm exhibits robust performance and the wavelet family that is most suitable for fall detection is Gaussian Wavelet.

As for the future work, we plan to investigate the characteristics of falls in more detail in order to gain a better insight on fall detection. Besides that we also plan to investigate the possibility of constructing a custom mother wavelet that is specialized in detecting falls. Moreover, we plan to extend our research by incorporating several classifiers in the experimental architecture, such as probabilistic models. Finally, we plan to experiment with more fall types and actions similar to falls, such as falling to a sitting position or recovering from a trip.

APPENDIX A: NOTATION USED IN THE THESIS

The accelerometer senses the acceleration the device is being subjected to. The readings of the accelerometer can be expressed as:

$$\vec{a}_t = (Acc\_x_t, Acc\_y_t, Acc\_z_t) \tag{A.1}$$

where Acc_x,Acc_y,Acc_z are the components of acceleration vector along x, y, z axes, respectively. A sequence of uniformly sampled acceleration vectors is then:

$$X = \{\vec{a}_t \,| t \in t_1 : t_n\} \tag{A.2}$$

where $t_i - t_{i-1} = \alpha$ and $\frac{1}{\alpha}$ is the sampling rate. Since DWT is a single dimensional transformation we need a mapping from $\Re^3$ to $\Re$ so that $MP(X) \in \Re$. An example to such a function is given in Equation A.3, in which $e_1 = (1, 0, 0)$.

$$MP_1(X) = e_1 \cdot X \tag{A.3}$$

Consider a function $f(x) : \Re \to \Re$, and let:

$$f_\tau(x) = f(x - \tau) \tag{A.4}$$

$$f_s(x) = f(\frac{x}{s}) \tag{A.5}$$

Then $f_\tau(x)$ is the *translation* of $f(x)$ by τ and $f_s(x)$ is the *scaling* or *dilation* of $f(x)$ by s. Fourier transform of a function $f(x)$ is defined as:

$$\hat{f}(w) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi xw} dx \tag{A.6}$$

where $i = \sqrt{-1}$. The Fourier transform is a frequency analysis of the given function. If $f(x)$ is a function of time, then $\hat{f}(w)$ is a function of frequency with hertz as its unit. The parameter of the Fourier transform is named w in this text to signify that it is defined in the frequency domain. Convolution, $(f * g)$ of two discrete signals, f and g is defined as:

$$(f * g)[n] = \sum_{m} f[m]g[n-m] \tag{A.7}$$

Similarly, the inner product, $< f, g >$ of two discrete signals, f and g is defined as:

$$< f, g >= \sum_{k} f[k]g^{*}[k] \tag{A.8}$$

where g^{*} is the complex conjugation of g such that $(a - ib)^{*} = (a + ib)$.

Finally, three performance metrics are used in this thesis to evaluate fall detection performances, namely $Recall$, $Precision$ and $F - value$. These metrics are defined as:

$$Recall = \frac{TP}{GT} * 100 \tag{A.9}$$

$$Precision = \frac{TP}{TP + FP} * 100 \tag{A.10}$$

$$F - value = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{A.11}$$

where TP is the number of falls that are correctly identified by the fall detector, i.e. true positives, GT is the actual number of falls in the processed data, i.e. ground truth, FP is the number of false alarms generated by the system, i.e. false positives.

REFERENCES

1. Valens, C., "A Really Friendly Guide To Wavelets", 2010, `http://polyvalens.pagesperso-orange.fr/clemens/wavelets/wavelets.html`.

2. Axer, H., M. Axer, H. Sauer, O. W. Witte, and G. Hagemann, "Falls and gait disorders in geriatric neurology", *Clinical Neurology and Neurosurgery*, Vol. 112, pp. 265–274, 2010.

3. CDC, "Falls Among Older Adults: An Overview", `http://www.cdc.gov/HomeandRecreationalSafety/Falls/adultfalls.html`.

4. Gurley, R., N. Lum, M. Sande, B. Lo, and M. Katz, "Persons found in their homes helpless or dead", *New England Journal of Medicine*, Vol. 334, No. 26, pp. 1710–1716, 1996.

5. WHO, "Falls", `http://www.who.int/ violence_injury_prevention/ other_injury/falls/en/index.html`.

6. Luo, S. and Q. Hu, "A Dynamic Motion Pattern Analysis Approach to Fall Detection", *IEEE International Workshop on Biomedical Circuits and Systems*, December 2004.

7. Özgür Alemdar, H., M. O. Özen, Y. E. Kara, G. R. Yavuz, Özlem Durmaz İncel, L. Akarun, and C. Ersoy, "A Robust Multimodal Fall Detection Method for Ambient Assisted Living Applications", *IEEE 18th Signal Processing and Communications Applications Conference, SIU*, Diyarbakir, April 2010.

8. Zigel, Y., D. Litvak, and I. Gannot, "A Method for Automatic Fall Detection of Elderly People Using Floor Vibrations and SoundProof of Concept on Human Mimicking Doll Falls", *IEEE Transactions on Biomedical Engineering*, Vol. 56, pp. 2858–2867, 2009.

9. Wu, G., "Distinguishing fall activities from normal activities by velocity characteristics", *Journal of Biomechanics*, Vol. 33, No. 11, pp. 1497 – 1500, 2000.

10. Cucchiara, R., A. Prati, and R. Vezzani, "A multi-camera vision system for fall detection and alarm generation.", *Expert Systems*, Vol. 24, No. 5, pp. 334 – 345, 2007.

11. Thome, N., S. Miguet, and S. Ambellouis, "A Real-Time, Multiview Fall Detection System: A LHMM-Based Approach", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, pp. 1522 – 1532, 2008.

12. Hazelhoff, L., J. Han, and P. de With, "Video-Based Fall Detection in the Home Using Principal Component Analysis", *Advanced Concepts for Intelligent Vision Systems*, Vol. 5259 of *Lecture Notes in Computer Science*, pp. 298–309, Springer Berlin / Heidelberg, 2008.

13. Liu, C.-L., C.-H. Lee, and P.-M. Lin, "A fall detection system using k-nearest neighbor classifier", *Expert Systems with Applications*, Vol. 37, No. 10, pp. 7174 – 7181, 2010.

14. Zhang, Z., E. Becker, R. Arora, and V. Athitsos, "Experiments with computer vision methods for fall detection", *PETRA '10: Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments*, pp. 1–4, New York, Samos, Greece 2010.

15. Kido, S., T. Miyasaka, T. Tanaka, T. Shimizu, and T. Saga, "Fall detection in toilet rooms using thermal imaging sensors", *IEEE/SICE International Symposium on System Integration*, pp. 83 –88, Tokyo, January 2009.

16. Hori, T. and Y. Nishida, "Ultrasonic sensors for the elderly and caregivers in a nursing home", *Seventh International Conference on Enterprise Information Systems*, pp. 110–115, Miami, May 2005.

17. Kolovou, X. and I. Maglogiannis, "Video-surveillance and context aware system for activity recognition", *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments*, Samos, Greece, June 2010.

18. Töreyin, B. U., Y. Dedeolu, and A. E. Çetin, "HMM Based Falling Person Detection Using Both Audio and Video", *Computer Vision in Human-Computer Interaction*, Vol. 3766 of *Lecture Notes in Computer Science*, pp. 211–220, Springer Berlin / Heidelberg, 2005.

19. Doukas, C. and I. Maglogiannis, "Advanced patient or elder fall detection based on movement and sound data", *Second International Conference on Pervasive Computing Technologies for Healthcare*, Tampere, February 2008.

20. Nocua, R., N. Noury, C. Gehin, A. Dittmar, and E. McAdams, "Evaluation of the autonomic nervous system for fall detection", *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3225 –3228, Minneapolis, September 2009.

21. Salomon, R., M. Luder, and G. Bieber, "iFall - A new embedded system for the detection of unexpected falls", *8th IEEE International Conference on Pervasive Computing and Communications*, pp. 286 – 291, Mannheim, Germany, April 2010.

22. Noury, N., P. Barralon, G. Virone, P. Boissy, M. Hamel, and P. Rumeau, "A smart sensor based on rules and its evaluation in daily routines", *Proceedings of the 25th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society*, Vol. 4, pp. 3286 – 3289, Cancun, September 2003.

23. Benocci, M., C. Tacconi, E. Farella, L. Benini, L. Chiari, and L. Vanzago, "Accelerometer-based fall detection using optimized ZigBee data streaming", *Microelectronics Journal*, Vol. In Press, 2010.

24. Chen, J., K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable sensors for reliable fall detection", *Annual International Conference of the IEEE Engineering*

in Medicine and Biology - Proceedings, Vol. 7, pp. 3551 – 3554, Shanghai, China, September 2005.

25. Marinkovic, S., R. Puppo, R. Pan, and E. Popovici, "Implementation and testing of a secure fall detection system for Body Area Networks", *27th International Conference on Microelectronics (MIEL) Proceedings*, pp. 315 –318, Nis, May 2010.

26. Estudillo-Valderrama, L. R.-T. J. N.-H. D., M.A.; Roa, "Design and Implementation of a Distributed Fall Detection SystemPersonal Server", *IEEE Transactions on Information Technology in Biomedicine*, Vol. 13, pp. 874–881, 2009.

27. Leijdekkers, P., V. Gay, and E. Lawrence, "Smart Homecare System for Health Tele-monitoring", *First International Conference on the Digital Society*, Guadeloupe, January 2007.

28. Lindemann, U., A. Hock, M. Stuber, W. Keck, and C. Becker, "Evaluation of a fall detector based on accelerometers: a pilot study", *Medical & Biological Engineering & Computing*, Vol. 43, No. 5, pp. 548–551, 2005.

29. Chao, P.-K., H.-L. Chan, F.-T. Tang, Y.-C. Chen, and M.-K. Wong, "A comparison of automatic fall detection by the cross-product and magnitude of tri-axial acceleration", *Physiological Measurement*, Vol. 30, No. 10, p. 1027, 2009.

30. Iso-ketola, P., T. Karinsalo, and J. Vanhala, "HipGuard: A Wearable Measurement System for Patients Recovering from a Hip Operation", *Second International Conference on Pervasive Computing Technologies for Healthcare*, Tampere, February 2008.

31. Lai, C.-F., Y.-M. Huang, J. H. Park, and H.-C. Chao, "Adaptive Body Posture Analysis for Elderly-Falling Detection with Multisensors", *Intelligent Systems, IEEE*, Vol. 25, No. 2, pp. 20 –30, 2010.

32. Weiss, A., I. Shimkin, N. Giladi, and J. Hausdorff, "Automated detection of

near falls: algorithm development and preliminary results", *BMC Research Notes*, Vol. 3, No. 1, p. 62, 2010.

33. Nyan, M., F. E. Tay, and E. Murugasu, "A wearable system for pre-impact fall detection.", *Journal of Biomechanics*, Vol. 41, No. 16, pp. 3475 – 3481, 2008.

34. Jia, N., "Fall Detection Application by Using 3-Axis Accelerometer ADXL345", `http://www.analog.com/static/imported-files/application_notes/AN-1023.pdf`.

35. Jantaraprim, P., P. Phukpattaranont, C. Limsakul, and B. Wongkittisuksa, "Improving the accuracy of a fall detection algorithm using free fall characteristics", *International Conference on Electrical Engineering/Electronics Computer Telecommunications and Information Technology*, Chiang Mai, May 2010.

36. Tong, L., W. Chen, Q. Song, and Y. Ge, "A research on automatic human fall detection method based on wearable inertial force information acquisition system", *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics*, Guilin, December 2009.

37. Kangas, M., A. Konttila, P. Lindgren, I. Winblad, and T. Jms, "Comparison of low-complexity fall detection algorithms for body attached accelerometers.", *Gait & Posture*, Vol. 28, No. 2, pp. 285 – 291, 2008.

38. Kangas, M., I. Vikman, J. Wiklander, P. Lindgren, L. Nyberg, and T. Jms, "Sensitivity and specificity of fall detection in people aged 40 years and over.", *Gait & Posture*, Vol. 29, No. 4, pp. 571 – 574, 2009.

39. Zhang, P., *Wireless hazardous fall detection and location tracking for medical urgency and residence care applications*, Master's thesis, Ecole Polytechnique, Montreal (Canada), Canada, 2009.

40. Hansen, T. R., J. M. Eklund, J. Sprinkle, R. Bajcsy, and S. Sastry, "Using smart

sensors and a camera phone to detect and verify the fall of elderly persons", *European Medicine, Biology and Engineering Conference*, Prague, November 2005.

41. Zhang, T., J. Wang, P. Liu, and J. Hou, "Fall Detection by Embedding an Accelerometer in Cellphone and Using KFD Algorithm", *IJCSNS International Journal of Computer Science and Network Security*, Vol. 6, pp. 277–284, 2006.

42. Dai, J., X. Bai, Z. Yang, Z. Shen, and D. Xuan, "Mobile phone-based pervasive fall detection", *Personal and Ubiquitous Computing*, Vol. 14, No. 7, pp. 633–643, 2010.

43. Ketabdar, H. and T. Polzehl, "Fall and emergency detection with mobile phones", *Assets '09: Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*, pp. 241–242, Pittsburgh, October 2009.

44. Sposaro, T. G., F., "iFall: an Android application for fall monitoring and response.", *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, Vol. 2009, pp. 6119–6122, 2009.

45. Google, "Android", `http://www.android.com/`.

46. Özgür Alemdar, H., *RFID Enhanced Wireless Sensor Networks for Healthcare Monitoring*, Master's thesis, Boğaziçi University, 2009.

47. Memsic, "Imote2.Net Wireless Sensor Node", 2010, `http://www.memsic.com/`.

48. Genetlab, "Sensenode", 2010, `http://www.genetlab.com/`.

49. AXIS, "207W Wireless Network Camera", 2010, `http://www.axis.com/products/cam_207w/index.htm`.

50. AXIS, "M1031 Wireless Network Camera", 2010, `http://www.axis.com/products/cam_m1031w/index.htm`.

51. UDEA, "RFID Reader Antenna", 2010, `http://www.udea.com.tr/`.

52. Zhang, C., C. Wang, D. Li, X. Zhou, and C. Gao, "Unspecific Event Detection in Wireless Sensor Networks", *International Conference on Communication Software and Networks*, pp. 243 –246, Macau, February 2009.

53. Stphane Mallat, S. G., *A Wavelet Tour of Signal Processing*, Academic Press, 1998.

54. Daubechies, I., "The Wavelet Transform, Time-Frequency Localization and Signal Analysis", *IEEE Transactions on Information Theory*, Vol. 36, pp. 961–989, 1990.

55. Jain, R., *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, Wiley- Interscience, New York, 1991.

56. Sun, "SunSPOT", 2010, `http://sunspotworld.com/`.

57. Purwar, A., D. U. Jeong, and W. Y. Chung, "Activity Monitoring from Real-Time Triaxial Accelerometer Data Using Sensor Network", *International Conference on Control, Automation and Systems*, Seoul, October 2007.