FALL DETECTION FOR ELDERLY PEOPLE USING DEPTH VIDEO DATA OBTAINED BY KINECT

A Thesis

by

Amir Abbas Davari

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Electrical and Electronics Engineering

Özyeğin University
August 2013

FALL DETECTION FOR ELDERLY PEOPLE USING DEPTH VIDEO DATA OBTAINED BY KINECT

Approved by:

Professor Dr. A. Tanju Erdem, Advisor
Department of Electrical and Electronics Engineering
Özyeğin University

Professor M. Oğuz Sunay
Department of Computer Science
Özyeğin University

Professor Ali Özer Ercan
Department of Electrical and Electronics Engineering
Özyeğin University

Date Approved: 2 August 2013

To

My dear parents who are my most precious possessions

ABSTRACT

Automatic detection of unusual events such as falls is very important especially for elderly people living alone. Real-time detection of these events can reduce the health risks associated with a fall. There has been a series of ongoing researches in the field of unusual event detection using the Microsoft‘s depth sensor Kinect. It has been applied in areas like fall detection using only the depth images and features derived from skeletal data having exaggerated dimensionality. This thesis will propose a novel method for automatic detection of fall event by using depth cameras. Depth images generated by these cameras are used in estimating the skeletal data of a person. The contribution here is to use features extracted from this data to form a strong set of features which can help us achieve an increased precision at low redundancy. The achievements indicate that the calculated features which are derived from skeletal data are moderately powerful for detecting unusual events such as fall.

ÖZETÇE

Düşme gibi anormal durumlarda otomatik algılama, özellikle yaşlı ve yalnız yaşayan insanlar için çok önemlidir. Bu durumların gerçek zamanlı algılanması düşmeyle alakalı sağlık risklerini azaltabilir. Hali hazırda Microsofts depth sensor Kinect kullanılarak anormal durumların algılanması alanında bir seri araştırmalar mevcuttur. Aşırı boyutlara sahip olan iskelet verilerinden elde edilen yalnızca derinlik görüntüleri ve özellikleri kullanılarak düşme algılama gibi alanlarda uygulandı. Bu tez derinlik kameraları kullanarak düşme olaylarının otomatik algılanmasıyla ilgili yeni bir yöntem sunuyor. Bu kamaralardan elde edilen derinlik görüntüleri, kişinin vücut lekesi ve iskelet verilerin hesaplanmasında kullanılır. Buradaki katkı ise güçlü bir özellik kümesi oluşturmak için bu verilerden alınan özellikleri kullanmaktır. Düşük sayıda fazlalıkla, bu bize doğruluğu başarmamıza yardım eder. Bu başarı gösteriyor ki insan vücudundaki leke ve iskelet verilerinin ikisinden de elde edilerek hesaplanan özellikler, düşme gibi anormal durumların algılanmasında kısmen güçlüdür.

ACKNOWLEDGEMENTS

I would like to take this opportunity to dedicate this thesis to my parents who have been the source of motivation and inspiration for me through my life. I am thankful to my advisor, Prof. Dr. Tanju Erdem who helped me a lot during my M.Sc. studies like a kind father. At the end, I offer my kind regards to my fellow colleagues, Dr. Tarkan Aydin and Mr. Rangeet Mitra who always willingly helped me with their ideas and skills.

This work was supported by Türk Telekom under Grant Number 1158003.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Motivation

Elderly population has likely possibility of falling approximately 50% more than the general population [8]. Falls being 40% more likely to occur in a hospital are of the most common injuries in hospitals than in other industries and locations [9]. Doctors and care takers administration have been held responsible and liable in the lawsuits and therefore prone to overspend in areas like personnel [10]. Based on a medical observation study [6], falling in elderly people are caused by different reasons like incorrect transfer or body weight shifting (41%), trips or stumbles (21%), Hits or bumps (11%), Loss of support with an external object like falling during sitting on a wheelchair (11%), Collapse or loss of consciousness (11%) and slipping (3%).

This thesis attempts to suggest a more accurate and usable algorithm and system to prevent falling problem in elderly people that has had multiple attempts at solutions. Unfortunately, although many algorithms have been suggested in the last two decades and even reach to near 100% of detection, but still there is no optimum solution in case of usability for this problem as the false alarm rate is considerable. Furthermore solutions that use a physical alarm, would be intrusive on the surrounding patients and may even increase the likelihood of falling. Solutions like movement controlling systems via RGB cameras do not preserve the privacy of the patient. Many approaches currently and in the past require users direct input for the system to be able to function such as use of a belt-size alarm with a button on it that only sounds when pushed. The system prevents fall to an extent in some types of injuries but obviously failed to show robustness in the case of falls due to unconsciousness. Other techniques are based on generalization of a fall action but their accuracy remains a big question mark in the case of detecting most fall scenarios. As there has been no empirical evidence of a decrease in injuries resulting from falls so research is being sought out.

Therefore, these two reasons, i.e., the importance of this issue based on the medical observation statistics and the lack of a perfect algorithm/system in case of specificity and usability are the biggest motivations of this work.

1.2 Contribution

The literature lacks a thorough survey on fall detection especially using Kinect. As it is crucial for a researcher to be expert on the state of the art and the previous works, we provide such a survey comparing the most recent works on fall detection mostly using Kinect.

The existing fall detection algorithms exhibit considerable false alarm rates which makes the algorithms hard to be accepted by the users. We also propose novel features to be used for fall detection based on a recent fall-related observation and statistic as reported in medical literature [6]. The main contribution in this part is to increase fall detection accuracy and reduce false alarms in comparison with the state of the art.

The main contributions of this thesis can be summarized as follow:

1. A brief survey of the studies, research, algorithms and developments in the field of fall detection.

2. Proposing a new event named high motion and using it to distinguish fall events more easily and accurately.

3. Achieving a good confusion matrix with low dimensionality feature with inter-class confusion being equivalent to state of art by proposing a new method in

this field, i.e., intelligent generation of synthetic data.

4. Developing a generalized algorithm which works for different persons, with different ways of doing the events and in different environments by working on a general data-set.

5. Providing an extensive RGBD (i.e., RGB and depth) video database for future vision-based fall detection researchs.

1.3 Outline

Chapter 2 does a comprehensive look at previous works and researches on fall detection topic. Algorithms will be categorized and compared to each other and pros and cons of them will be discussed.

Chapter 3 describes how the depth video data was recorded using Kinect. Softwares and frameworks which were employed for this issue will be discussed. It will be continued by discussing about the features. First the main features used in the state of the art will be introduced and defined mathematically and then the proposed features by the thesis will be discussed and described. Then fall detection system in the thesis will be discussed. Classifier which was used, feature calculation and preprocessing on the calculated features are the last topics of this chapter.

In chapter 4, the experimental results of the proposed algorithm will be reported and will be discussed in details.

Finally in chapter 5, the main contribution of the thesis which was introduced in chapter 1, will be compared by the achievements of the thesis. Also the ideas of the author which were not completed for this study will be presented as the future works.

CHAPTER II

RELATED WORKS

In this chapter, a short survey on most recent works on fall detection works, systems and algorithms will be presented. We categorized fall detection systems into two main categories: Sensor-based approaches and Vision-based approaches.

2.1 Sensor-based Approaches

Briefly, sensor-based fall detection algorithms mainly rely on two main type of sensors: Wearable sensors and environmental sensors. This section will discuss and compare approaches using these two kind of sensors for fall detection.

2.1.1 Wearable Sensor-based Approaches

Wearable sensor based approaches are based on different kind of sensors which are attached to the body parts via wearing them by patients. In this approach, the location and motion of the body parts, i.e., speed and acceleration of them will be recorded and collected as the raw data for the further processing analysis for activity recognition.

2.1.1.1 Accelerometer

Accelerometer is a sensor which can measure the proper acceleration. The measured value is not necessarily the gradient of velocity. It will measure the amount of force applied to a test mass and calculate the acceleration value with respect to the second law of Newton, i.e., $a = \frac{F}{m}$ in which a is the acceleration and F is the force applied to a test mass m. Approaches that use these sensors, are usually looking for measuring the acceleration of body parts using the accelerometers attached to them, in order to monitor the body activities.

Merryn et al. [11] employed an accelerometer mounted to waist of the object person. They determine an activity as fall when a sudden negative acceleration change occurs. This means that the patient/object′s position has been changed from standing position into lying on the ground.

Bianchi et al. [12] introduced a barometer pressure sensor for measuring the altitude. As sudden negative change in acceleration can be caused by different activities like jumping down and not only by falling, height measurement using an altitude sensor can improve the fall detection performance in accelerometer based approaches.

In [13] Tamura et al. proposed an airbag worn by the person. This study is actually a kind of effort to prevent injuries caused by fall. Their system adjusts the inflation of the airbag based on the signals received from the accelerometer and gyro sensor attached to the person. This system is a kind of useful device for construction sites and places with a high risk of falling occurrences for reducing the fall related injuries. Chen et al. [14] developed a low-power wireless sensor network by taking the advantage of small sensor nodes. It performs the acceleration sampling which cause the reduction of burden on the network. In the processing step they obtained the angle of motion by taking a dot product of acceleration vector and orientation information. [10] tried a kind of signal monitoring. They monitor the amplitude of the acceleration signal measured by an accelerometer on the patients head and [11] raises a fall detection alarm if the speed and acceleration goes beyond the specific reference. Using this velocity and acceleration reference, fall would be distinguished from non-fall events.

2.1.1.2 Fusion of Accelerometery and Posture Sensors

It is logical to take advantage of body posture sensors directly or indirectly for performing a good job in fall detection. Employing a group of sensors on a belt, Luo et al. [15] represented the acceleration vectors in 3D space and by filtering the noisy components with a Gaussian filter, a three dimensional body motion model was produced that can has a direct dependency to various body posture and the output of the accelerometers.

[13] took advantage of a two-axis accelerometer with a posture sensor for detecting fall. In this article they developed a wrist-worn device which measure and record the bio-signals in addition to acceleration and reporting the injuries caused by fall for emergency reporting and help.

In [16] Ghasemzadeh et al. studied EMG (electromyogram) signals and the body acceleration in order to propose a model for body balance based on muscle signals and body acceleration. This research will have a great contribution on fall detection topic as a fall is always followed by losing balance.

2.1.1.3 Inactivity With Accelerometery

Generally a fall is considered as a dangerous activity when it causes a long time of body inactivity which can be followed by injuries. One of the advantages of accelerometer-based approaches is that the duration of activity and inactivity can simply and accurately be measured. This information can assist the fall detection systems.

[17] and [10] used motion sensors beside the wearable wireless accelerometer sensors to record the activity or inactivity durations to understand whether a fall event is occurred as a dangerous event or not.

2.1.1.4 Discussion on Wearable Devices

As all developed fall detection systems, wearable sensor based devices have specific pros and cons. The main advantage of these kinds of algorithms/devices is their cost. They are cheap and also easy to install and use. On the other hand, these devices should be worn by the person that makes them hard to be accepted for daily use. Furthermore as they are always attached to the body, the probability of being damaged or disconnected for them is higher than the other kind of sensor based approaches.

2.1.2 Environmental Sensor-based Approaches

Most of these approaches use analysis of vibration, pressure and sound produced by human activities in order to detect and distinguish fall. Zhuang et al. [30] employed a far-field microphone to monitor the noise of the environment. They created a GMM (Gaussian Mixture Model) model super vector using SVM (support vector machine) classifier. In this approach they analyze the noise in the environment and their model raise an alarm for those which are distinguished as fall related noise.

In [17], Alwan et al. proposed a totally passive system that uses the floor vibration for its analysis. Although this approach has some limitation as the type of floor material is not the same everywhere and therefore the type of vibration produced by falling will change from one place to other.

2.1.2.1 Discussion on Environmental Sensor-based Approaches

Pressure and vibration sensors are common in almost all environmental sensor based fall detection approaches. These sensors are cost effective and more acceptable by people rather than the wearable ones. But their main problem is that they will sense vibration by every object dropping or falling on the ground. Therefore the amount of false alarms is so high in these approaches.

2.2 Vision-Based Approaches

The literature related to our work can be grouped into three categories: Those that use RGB data captured by a video camera, those that use depth data captured by a Kinect sensor and those that use human skeletal data obtained from the depth data.

2.2.1 Skeletal Data as the Data Set

Among those papers that explicitly use the skeletal data obtained through the use of a Kinect sensor, [18] estimates the ground plane from the v-disparity map. This is possible through the linear increase of depth along the ground. Then it fits a line to five joints, namely head, neck, torso and mean position of their knees. The line fitted to these points forms a major axis. Their features included length and orientation of this major axis. However their scope was limited only to fall detection. They used different Kinect data and compared the results. Using 3D sensor in world coordinate, they achieved 95.8% accuracy and 92.5% recall. Data set in [18] has a disadvantage. The orientation of fall in the whole data set is static, perpendicular to Kinect camera view which can stop generalizing the classifier for falling in other orientations. Another effort [19], which uses depth data too, detects routine activities using a Maximum Entropy Markov Model (MEMM) based approach. Their features included viewing all rotation according to a man's torso (to capture body pose). It takes ten joints and represents the rotations as quarternions. Also the foot is measured with respect to the torso and the angle of line joining head and hip with the vertical. All these reflect the pose information. For motion information they select some frames for past 3 seconds and compute the temporal joint rotations. Also, they used skeletal and RGBD Histogram Of Gradient (HOG) features. First of all common observation yields and the [19] admits that activities in general are unstructured for which they use a two-stage MEMM. Furthermore it is our observation, which stems from common-sense that unusual activities are more structure-less than usual activities, which manifested in their results when they got bad results in some scenarios (with F-numbers less than 0.5). Hence we did not find it suitable to detect unusual events using Markov Model.

2.2.2 Depth Data Based Vision Approaches

Following papers used depth data. [20], used a 3-D bounding box approach. Their features were smoothed temporal gradients of height and width to depth ratio. The features were compared with a temporally varying threshold found by a search technique. Their work produced good results (100% accuracy) but their scope was limited only to fall. Also searching can be computationally tedious and no basis for the frame duration required for searching was given. [21] forms depth motion maps from three 2D perspectives from the 3-D data obtained from the Kinect. A linear SVM classifier was used to detect certain sample events like tennis serve, hand clapping etc. But there is nothing specific in their work about abnormality detection other than some cases of sport related too much motion. [22] uses skeletal data involving intra-frame joint difference for posture, inter-frame joint differences for motion and deviation from an overall pose for the global dynamics. Subsequently PCA was taken and a 128 dimensional vector was an input to a Naive Bayes classifier. Again the scope of the events that they covered was limited to fall, though the features were innovative. [23] uses just the depth data. It extracts the blob and its centroid and the ground plane via the v-disparity map. It again uses 2-D features, namely distance of blob from the ground and the velocity of the blob. A fall is signalled if these features violate a 97% confidence interval from their mean values. Unique feature of their work is that there is no sign of Bayesian spirit in their work as they used a search technique.

2.2.3 RGB Data Based Vision Approaches

[24] does something analogous to [19] using trackers in the pre-Kinect era in terms of classifiers. However, the duration of a sub-activity was modeled by a novel Coxian distribution. [25] used a criterion derived from spatiotemporal energy of the blob, which was made from RGB video data. Large motion was detected from the variation in area of the blob over time. Furthermore, an ellipse was used to approximate the

blob. The high temporal variance of orientation of the ellipse was used to detect slip and low value of the major axis as compared to minor axis was used to detect a fall. They got 91.11% accuracy for fall detection. [26] introduced a structural cost where norm of the 3-D inter-joint angles were considered. The mean and variance of this cost along with maximum and minimum value of height formed a 4-D kinematic feature. Also there were features based on 2-D blob and histogram of width to height ratios. Additionally, they used RGB data when depth was not available. They trained an SVM tree classifier. The feature dimensionality was too high and the scope was limited to detecting only fall, i.e., there is high redundancy in their features.

As our work is primarily related to unusual activity detection for elderly and child care, it is imperative to refer to [27]. It is based on multisensor fusion and detecting Normal Activities of Daily Living (NADL). Also data used in [27] is RGB and classification is based on Markovian relation between events. Although number of events are much which can be considered as an advantage of the classifier design, but number of sensors and dimension of feature are too high for NADL detection which is a disadvantage. Privacy preservation was not considered in [27] which is its other con. Another work [28], though very early outlined some challenges faced by the vision-based event detection systems (apart from axiomatically defining three different types of falls). Some of them were night-time surveillance. Other problems outlined in various other references are texture dependency (as blobs inferred from RGB data are dependent on texture) and privacy preservation [27].

[29, 30] which use depth data, evaluate the credibility of Kinect as a sensor based on some gait features. They conclude that although the performance of Kinect was comparable to a state-of-art camera, it had some minor pitfalls like blending of depth data near walls, limited field of view etc. It has also outlined a "smart fusion" of RGB and depth data to address such issues. They conclude that Kinect is a moderately accurate cost-effective sensor. Also, depth map of Kinect apart from being a useful

feature, preserves privacy.

Table 1 shows the most important vision-based fall detection approaches and works which are discussed above in order to compare their features, performance, classifier, pros and cons, events they tried to detect and their used data type.

Table 1: Comparison of important works in fall detection using vision based approaches. The first column is the reference numbers. Second column shows the activities which were trying to be detected in the references. Third column shows the detection performance achieved in the references. Classifier column shows the classier which was used. The sixth column lists the pros and cons the works. The last column shows the type of data-set which was processed.

Reference	Event	Features	Performance	Classifier	Cons and Pros	Data Type
[2]	Fall	1) Estimation of the ground plane 2) Length and orientation of the spine	1) 0.87 F-score using image coordinates 2) 0.96 F-score using world coordinates	One-class classifiers introduced by Pospescu et. al.	1) Scope limited only to fall. 2) Need for estimating ground plane 3) Need to evaluate the spinal chord.	Skeletal data.
[19]	Daily Activities	Body pose features	Precision=84.7% - Recall=83.2%	maximum-entropy Markov model (MEMM)	1) High dimensionality of features. 2) Scope limited only upto detecting usual events 3) Bad performance in some scenarios	RGBD and skeletal data.
[31]	Fall	Height and width to depth ratio of the bounding.	100% detection rate	NA	1) Scope limited only to fall. 2) Lacks Bayesian spirit.	Depth meta-Data.
[32]	Fall	1) Height 2) Fraction of frames where head drops	100% detection rate with only 7% false alarm	Naive-Bayes	1) Scope limited to fall. 2)Independence assumption. 3)High Redundancy in features for detecting just fall.	RGB data.
[24]	Daily Activities	The entire room is divided into cells. Observations of cells that a human visits during his activity.	97.5% - 100% detection rate	The hidden semi-Markov model (HSMM)	1)None of the above is an unusual activity. 2)Does not address the illumination dependence problem.	RGB data.
[33]	Slip and Fall	If dimensions of the bounding box over the foreground blob is a and b, they used $\frac{b}{a}$ and the orientation of the blob into the bounding box as their features	91.11% detection rate	Arbitrary	Choice of thresholds of 15 degrees and ratio of 0.3 not justified in a Bayesian manner.	RGB data.
[21]	Horizontal Wave, Hammer, Forward Punch, etc.	1) Activity recognition using HOG. 2) 2-D maps are analyzed from different views and motion energy is calculated. 3) Three 2-D maps (front, side and top)	80%-95% detection rate for all activities	Linear SVM	1) High dimensionality of features.	Depth Motion Map.
[22]	Horizontal Wave, Hammer, Forward Punch, etc.	1) Static Posture Intra Frame Joint Differences. 2) Motion- Inter Frame Joint Differences. 3) Overall Dynamics Deviation from a calibration pose. 4) Take PCA.	55%-85% detection rate for all activities	Naive-Bayes Nearest Neighbor	Comparatively bad results in some scenarios with high data dimensionality.	Depth Motion Map and Skeletal data.
[23]	Fall	1) Ground plane estimation from the V-disparity map. 2) Find the foreground blob from the depth images. 3) Distance of the blob centroid from the ground plane. 4) Velocity of the blob centroid in 1 sec	98.7%	97% accuracy	1)Scope limited only to fall. 2)High dimensional data for detecting only fall.	Depth Map
[27]	Daily Activities	1) RGB data plus other sensors. 2) Contact Sensor 3) Pressure Sensor.	63% - 88% detection rate	Dempster-Shafer Theory of Evidence for Fusing Sensor	1) Thesis basically based on fusing sensors. 2) Gives definitions about what is an event, sensors etc.	RGB

CHAPTER III

FEATURES AND FALL DETECTION

Selecting the appropriate set of features makes us able to do a good job in classification. in order to make feature-based classification work, we need to have some knowledge of what features make good predictors for the classes we are trying to distinguish. For example, body height distinguishes falling and walking, but doesn't distinguish falling from lying. Depending on the classification task we are facing, different sets of features may be important.

Activity recognition using classification is not an exception. To be able to distinguish falling from the other activities, we need to consider or define features which works well for fall detection.

3.1 Definition of Features Used in the Depth Video-Based Literature

In this section we summarize the features calculated from depth and skeletal data for fall detection which were used in state of the art.

3.1.1 Features Calculated From RGBD Data

In [34], they used RGBD data. They formed MHI (i.e., Motion History Image) from their dataset and then Hu moments were used to extract the features from the MHIs. It is worth mentioning that using MHI and its seven central moments, we can define the picture having the rotation, scaling and translation. Imagine $I(x, y, t)$ be an image sequence and the threshold value for generating the MHI mask be K. t to be the fixed duration and τ, the maximum time window. Each pixel intensity value in MHI video will be a function, H_t, of that pixels history of motion.

$$H_t(x, y, t) = \begin{cases} \tau, & \text{if } (I(x, y, t) - I(x, y, t-1)) > K \\ max(0, H_t(x, y, t-1) - 1), & \text{Otherwise} \end{cases} \tag{1}$$

Also in this work they added a third dimension to conventional MHI which is depth. They introduced DMHI (Depth Motion History Image) which is exactly the same as the conventional MHI, but instead of image sequence $I(x, y, t)$, they used depth $D(x, y, t)$ as follows.

$$H_t(x, y, t) = \begin{cases} \tau, & \text{if } (D(x, y, t) - D(x, y, t-1)) < -K \\ max(0, H_t(x, y, t-1) - 1), & \text{Otherwise} \end{cases} \tag{2}$$

DHMIs contain two kind of information. Forward-DHMI (fDHMI) which gives information about forward motion history, i.e., increase of depth. Backward-DHMI (bDHMI) which gives information about backward motion history, i.e., decrease of depth. According to the upper brief introduction, they used 21 features which are the 7 hu-moments for each of the 3D-MHIs

In [1] they used RGBD video as their dataset. Also as the previous paper, they consider a time threshold after falling to raise a fall alarm. The features they used are as follows:

$$\begin{cases} \text{Intra-frame features:} & \begin{cases} B_i: & \text{Bounding Box Aspect Ratio} \\ O_i: & \text{Orientation} \\ A_i: & \text{Ellipse Axis Ratio} \end{cases} \\ \text{Inter-frame feature:} & \begin{cases} M_i: & \text{Motion Speed} \end{cases} \end{cases} \tag{3}$$

The more detailed definition of the features are:

B_i :. Height of the bounding box surrounding the person divided by the mean of its both widths.

O_i : The orientation of the major axis of the ellipse fitted to the person, specified as the angle between the major axis and the ground plane.

A_i : The ratio between the lengths of the longest axis and the second longest axis of the ellipse fitted to the person.

M_i : The relative number of new motion voxels ν_i in the current frame compared to the previous frame. In other words, M_i is the ratio of the i^{th} frames new voxels and its all voxels.

By motion voxel ν_i they meant the voxels which can be changed because of the movement of the object. They defined M_i as follow:

$$M_i = |\nu_i \backslash (\nu_i \cap \nu_{i-1})| / |\nu_i| \tag{4}$$

i: Frame number

ν_i : Voxels in frame i

In Figure 1, the second picture is a bounding box fitted to the human body.

The third one is showing an ellipse fitted to the body. Generally the vertices of the bounding box and axes of the ellipse are more concerned as feature in papers.

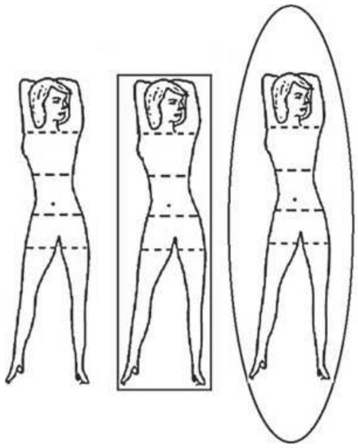

Figure 1: Bounding box and ellipse fitted to the body [1].

In [35], the research team used the height of the human 3D centroid from the ground plane. They estimated the ground plane using the V-disparity approach which is computationally better than RANSAC plane fitting approach which is the common way of ground plane estimation. They found the coefficients in the ground plane $aX + bY + cZ + d = 0$ by least squares of the 3D points estimated from the ground. And to detect the person they used the background subtraction by having the background firstly. If we say each pixel (i, j) in a current frame is I, and each one of the background is , a pixel will be considered as foreground if $|I(i, j) - B(i, j)| \geq T(i, j)$in which $T(i, j)$ is 2 times the pixel standard deviation.

Also they calculate the human body velocity as their other feature. The body velocity I was computed as the centroid displacement over a one second period. Finally the mathematical expression of the feature they used, height of the body centroid, is:

$$Features : \begin{cases} D_t = \frac{|aX_{COG}(t)+bY_{COG}(t)+cZ_{COG}(t)+d|}{\sqrt{a^2+b^2+c^2}} \\ V_t = |COG_t(x,y,z) - COG_{t-1}(x,y,z)| \end{cases} \tag{5}$$

In [2] the authors used different sensing ability of the Kinect. Sound sensor, RGB camera and depth camera were the data recording facilities they took advantage of. As we concern about depth data features, we just mention the features they used regarding the depth data. Firstly it should be mentioned that they used MS Kinect SDK as the depth recorder.

As it is shown in Figure 2 The orientation of the body major axis is calculated using the coordinates of the head, shoulder, spine, hip and knee joints obtained from MS SDK. Using the least squares algorithm to fit a straight line to the data points results in the orientation of the major axis. The angle between this major axis and the estimated ground plane was one of their features. Also they use the height of the spine from the ground plane. Fall alarm will be raised if the major axis orientation become almost parallel to the ground plane and the spine height be near the ground. According to me as a person who worked on this topic, these features cant help distinguishing the lie and fall. So they are not strong enough. Also in their result report, they say a fall is considered to be detected if the algorithm alarms within the period of falling, either for one single frame.

In [3] they used skeletal data obtained from MS Kinect SDK when the person is in the range of Kinect (4m) and RGB data in other case. As it is shown in Figure 3 they chose 8 joints on head and torso from the 20 available joints in MS SDK. Their reason is that other joints (on limbs) introduce more noise than useful information to distinguish whether a person is falling.

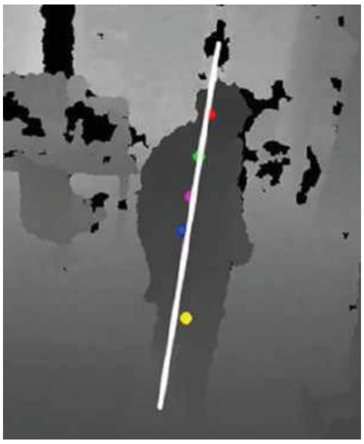

Figure 2: Line fitted to the body. Image is borrowed from [2].

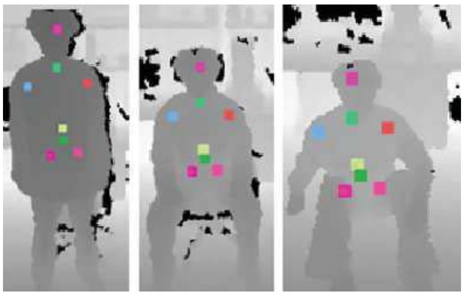

Figure 3: Joints marked on the body using MS SDK [3].

In the case of using skeleton joints, they use the structure difference cost $C(\xi)$ as follows:

$$C(\xi) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \|\theta(\xi_i, \xi_j) - \theta(\phi_i, \phi_j)\| \tag{6}$$

$$\theta(i,j) = \frac{\arcsin(\frac{i_x - j_x}{dist(i,j)})}{2\pi} \tag{7}$$

Where $\theta(\xi_i, \xi_j)$ and $\theta(\phi_i, \phi_j)$ denote the angles between two joints i and j on two skeletons.

ξ and ϕ, respectively, the geometry distance between two joints i and j is denoted as $dist(i,j)$.

When the person is out of the range of the Kinect, they use RGB as mentioned. They do background subtraction to get the human body. From that they form a bounding box and use its height as the feature. Briefly, they used the following features as their features

$$Features : \begin{cases} C(\xi) & \text{if human is in the range of Kinect (4m)} \\ H_{boundingbox} & \text{if human is outside the range of Kinect} \end{cases} \tag{8}$$

3.1.2 Features Calculated From Depth and Skeletal Data

In [36] the used dataset was depth video. The motions they tried to detect were sitting, bending, lying and standing. They used the height of the center of gravity (COG) from the floor plain and the orientation of the body spine as their features. Orientation of the human spine (angle between the Head-Torso line and a horizontal line) ϕ obtained from an appropriate extracted 3D skeleton. The Discrete Reeb Graph (DRG) is used for the skeleton extraction. Also they defined a 4 second time threshold for their detection. If COG remains lower than a threshold for 4 second, they alarm it as fall; otherwise this reduction in COG height will be interpreted as a movement rather than fall.

$$F(t) = \begin{cases} h_{COG}(t) - h_{floor}(t) \\ \phi = \angle BodySpine \end{cases} \tag{9}$$

[37] used depth image. The used a bounding cylinder instead of bounding box or ellipse. The features they used are position, speed and acceleration of the COG of the body and of the highest point and the Bounding Cylinder with its height to diameter ratio.

$$Features: \begin{cases} \text{COG}(x, y, z) \\ \text{Speed}_{COG} \\ \text{Acceleration}_{COG} \\ h_i: \text{ Position of the Highest point of the detected body} \\ \text{Speed}_h i \\ \text{Acceleration}_h i \\ \text{Bounding Cylinder height to diameter ratio} \end{cases} \tag{10}$$

In [31] they used depth image as their data captured by OpenNI. Their features are height, width and depth of the 3D bounding box they obtained from OpenNI and their first derivative.

$$\begin{cases} H = |Y_{min} - Y_{max}| \\ W = |X_{min} - X_{max}| \\ D = |Z_{min} - Z_{max}| \end{cases} \tag{11}$$

As their analysis showed the relation of fall and a combination of W and D, they decided to combine them together and define and use a new feature $WD = \sqrt{W^2 + D^2}$ instead of using W and Dseparately. So their final features which their algorithm used to detect fall event are:

$$Features : \begin{cases} H_i \\ WD_i = \sqrt{W_i^2 + D_i^2} \\ \nu_{H_i} = \frac{H_i - H_{i-1}}{t_i - t_{i-1}} \\ \nu_{WH_i} = \frac{WH_i - WH_{i-1}}{t_i - t_{i-1}} \end{cases} \tag{12}$$

This was a summary of almost all features which were calculated using RGB and/or depth video data for fall detection. Based on their results, they could reach almost 100% detection rate but their false alarm rate is still high and make their algorithm unusable in real life. This means that these features are unable to separate fall and non-fall activities completely without any confusion. Also in order to achieve an acceptable result, they used high dimensionality feature space which makes the classification process complex. Therefore we propose new features and modify the features in the state of art in order to cover these problems, i.e., high false alarm rate and high dimensionality of the feature space.

3.2 *Ideas for Fall Detection*

In this thesis, two different ideas with different feature sets were proposed and experimented for fall detection study. In the first idea, new features were introduced and in the second idea a new event, i.e., high motion, was tried to be detected as it was supposed to have high confusion with fall event.

Let the body be represented by a set of skeletal joints positions $\{\mathbf{s}_i^n(x, y, z) \in \mathbb{R}^3\}$ where i is the skeletal joint index as they are shown in figure 4 and table 2 and n denotes the time index, i.e., the frame number.

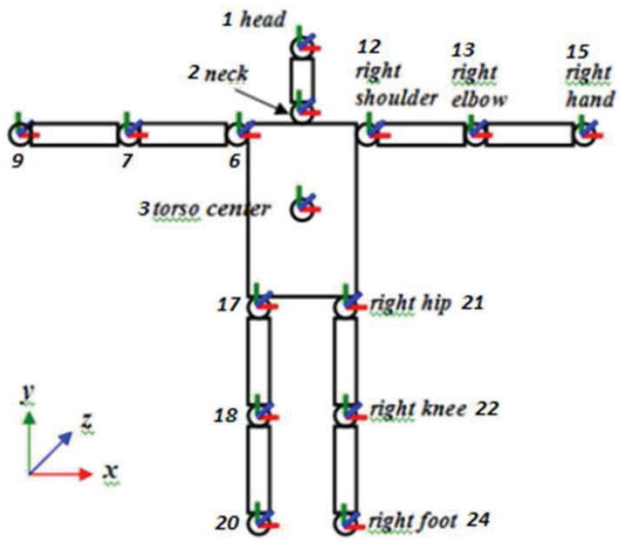

Figure 4: Skeleton joints indexes provided by OpenNI. Figure is borrowed from [4].

The features definitions proposed in this thesis are:

3.2.1 Features Used in Evaluating Idea 1

In order to evaluate the first idea, five features were chosen and calculated. Some of them are modified to be more stable and some are novel.

Table 2: Skeleton joints indexes list provided by OpenNI.

Joint Name	Joint Index
Head	1
Neck	2
Torso Center	3
Left Shoulder	6
Left Elbow	7
Left Hand	9
Right Shoulder	12
Right Elbow	13
Right Hand	15
Left Hip	17
Left Knee	18
Left Foot	20
Right Hip	21
Right Knee	22
Right Foot	24

3.2.1.1 Height

One of the indicators of body pose is the body height. Falling follows by a considerable change in body height and can be a suitable feature for fall detection. Although this feature was used in the literature before, certain modifications were applied during its calculation. Skeleton data provided by Kinect is noisy in the end points of the body, e.g., head, hands and legs. Central joints are more stable like shoulders, neck, torso, hips and knees. Average of the shoulders heights were used for body height approximation as they are provided more stable and with less noise by OpenNI; Equation 13.

$$f_1(n) = \frac{\mathbf{s}_6^n(y) + \mathbf{s}_{12}^n(y)}{2} \tag{13}$$

In which "6" and "12" are indexes of shoulders mentioned in Figure 4

3.2.1.2 Height Vertical Temporal Gradient

Vertical speed of the upper body is also a major indicator in fall definition; Equation 14.

$$f_2(n) = f_1(n) - f_1(n-1) \tag{14}$$

This feature is not novel but as more stable joints are used for its calculation, it is more powerful than the one used in the state of art.

3.2.1.3 Body Orientation

The main orientation of the body can be estimated by a line fitted on it. In this study, shoulders, torso and hip′s 3D positions were selected to find the body orientation. A line l was fitted through these 3D points and the angle θ between the line and ground plain was calculated as the third feature; Equation 15. Again this feature is a modified version of the one used in the literature.

$$f_3(n) = \theta(n) \tag{15}$$

3.2.1.4 Body Orientation Temporal Gradient

Fall is following by a big change is body orientation. Therefore the temporal gradient of θ was chosen as the fourth feature; Equation 16

$$f_4(n) = \theta(n) - \theta(n-1) = f_3(n) - f_3(n-1) \tag{16}$$

This feature is novel. The main purpose for its introduction is to remove the confusion between falling and sitting down quickly.

3.2.1.5 Distance Between Center of Mass and Body Support

The last feature is based on the main contribution of this research study, i.e., weight shifting. The main idea is to calculate the distance between the body center of mass (COM) and body supports, i.e., the feet. Body balance is kept by feet. In a well balanced situation, COM projection on the ground (COM_P) is between the feet and has the least possible distance to them. Now consider the following to possible fall instances; 1) Front/Backward fall in which the distance between (COM_P) and the line between the feet (L_P) will increase. 2) Side fall in which this distance will remains almost unchanged while the body balance burden will go to one foot. In the first instance, distance between (COM_P) and (L_P) is an important information that defines fall. In the second instance, as just the closer foot to (COM_P) should take care of body balance, distance between (COM_P) and that foot will be the crucial information to detect a fall.

In this study, torso was chosen as an approximation for COM. To write the mathematical expression of this feature, we need to introduce the following variables:

COM_P : Projection of COM on the ground plain

RF_P : Projection of right foot on the ground plain

LF_P : Projection of left foot on the ground plain

L_P : Line connecting RF_P and LF_P

P_P : If we draw two perpendicular lines to (L_P) passing from RF_P and LF_P, part of the ground plain which is bounded by these two lines is considered as (P_P)

Figure 5 shows the upper descriptions. Equation 17 describes $f_5(n)$ mathematically.

$$\begin{cases} Distance(COM_P(n), P_P(n)); \\ if\ max(|COM_P(n) - LF_P(n)|, |COM_P(n) - RF_P(n)|) \leq |L_P(n)| \\ \\ min(|COM_P(n) - LF_P(n)|, |COM_P(n) - RF_P(n)|); \\ if\ max(|COM_P(n) - LF_P(n)|, |COM_P(n) - RF_P(n)|) \geq |L_P(n)| \end{cases} \tag{17}$$

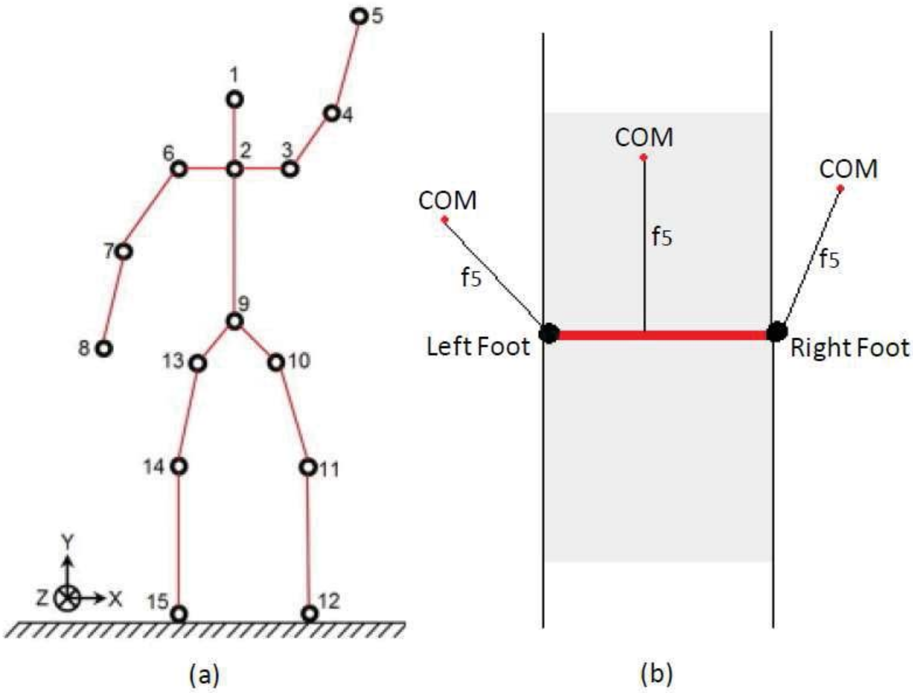

Figure 5: Skeleton joints indexes provided by Microsoft SDK. Fifth Feature Definition; (a) Frontal view of the skeleton. Figure is borrowed from [5] (b) Top view of skeleton (just feet and COM are displayed).

This novel feature is proposed for the first time based on a medical observation study [6] which says weight shifting is the most reason of falling in elderly people. As this thesis and its data set are based on falling in old people, and the main contribution is to reduce the false positives as much as possible, this feature was suggested.

3.2.2 Features Used for Evaluation of Idea 2

In this idea we tried to detect fall, high motion events and normal events. In the idea of this experiments, it was considered that every fall is a kind of high motion activity. Therefore if we try to detect high motion activities along with fall, the confusion between fall and normal will be reduced and the fall detection performance will be improved.

The feature for fall was defined as equation 18.

$$f1 = \frac{1}{4} \sum_{i \epsilon \{head, neck, leftshoulder, rightshoulder\}} \nabla_n \mathbf{s}_i^n(y) \tag{18}$$

where y denotes the vertical component of the gradient feature.

For the second event, i.e., High Motion, we used a novel feature which its definition is shown in equation 19.

$$f2 = \frac{\Sigma_{i=1}^{M} \|\nabla_n \mathbf{s}_i\|^2}{M} \tag{19}$$

We used the features $f1$ and $f2$ as discussed above, in the form shown in equation 20.

$$\mathbf{f}(n) = \begin{pmatrix} f_1(n) \\ f_2(n) \end{pmatrix} \tag{20}$$

In this section the mathematical definitions of the features were shown. In section 3.4, the practical calculation of these features on an extended data for evaluating their performances will be discussed.

3.3 Classifier Used In The Thesis

In this study we used Support Vector Machine (SVM) algorithm as the classifier. We employed a Matlab interface for SVM named LIBSVM [38] to classify our data set. The feature vectors in the experiments were fed to the classifier and using 3-fold cross validation, the results were calculated and reported.

3.4 Data Preprocessing and Feature Calculation

As it was mentioned in section 3.2, in this thesis, two different ideas with different feature sets are proposed and experimented for fall detection study. In the first one, new features are introduced and in the second one new event, i.e., high motion activity, is tried to be detected as it is supposed to have high confusion with fall event in order to improve the detection performance of fall.

Both of these ideas are using skeletal data as the raw data and the features were calculated based on the joints positions.

Skeletal data is nothing but a fitted virtual skeleton pattern on the body depth map. Therefore it is a function of camera viewing angel. In order to eliminate this dependency, skeleton should be translated from camera coordinates to world coordinates. Four points on three perpendicular lines were picked in real world as origin points and their 3D position values according to the camera were recorded. By minimum least squares error method, the translation matrix (which is a 3*3 matrix as all points are 3D) was calculated and applied to all skeleton joints 3D positions and the skeleton in the real world was created. Figure 6 shows this conversion from camera coordination to world coordination.

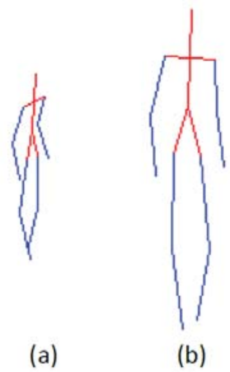

Figure 6: (a) Skeletal data in camera coordinates (b) Skeletal data translated into world coordinates.

3.4.1 Idea 1: Usefulness of New Features

Features mathematical expressions are described in the previous sections. For each frame, the mentioned five features were calculated. For smoothing and eliminating the effect of noise added during data recording, after computing the feature vector we applied a postprocessing step. Windows with one second length were pushed on the feature vectors in the steps of 0.5 second. In each window, first the feature values were sorted. The first and last 10% were removed. The average of the remaining values in the window was selected to be saved instead of all window values. In other words the following steps were done to the feature vectors:

Step1: 30 consecutive frames (which is one second as data was recorded at 30 fps) will be chosen
Step2: They will be sorted
Step3: First and last 10% will be removed
Step4: Average of the rest values will be taken
Step5: All 30 values in the window will be replaced by the average obtained in Step4
Step6: Go 15 frames forward (which is 0.5 second)
Step7: Go to Step1

After windowing the feature vector, corresponding labels for each window were saved as a label vector. The new windowed feature vector consist of 3152 window; 344 window for Fall and the rest 2808 windows for non-Fall events.

Figure 7 shows the five features calculated for a fall instance in data set.

3.4.2 Idea 2: Usefulness of Adding a New Activity Detection

In this idea we labeled and used 3 events; Fall, Normal and High Motion. When we do not find a skeleton, we attribute it as a "don't know event" and keep it out of the training and testing samples. Also some more events like "Entry", "Exit", "Lying on ground" and "Sitting" are labeled in the data set but they are labeled in the category of normal activities (non-fall activities). Table 3 shows the mentioned activities labels

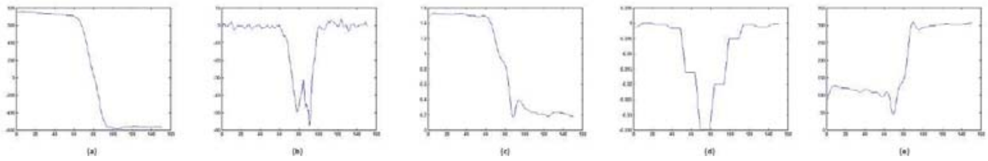

Figure 7: Features plotted v.s. time for a sample fall event, (a) Body Height: During Fall, body height is expected to be decreased (b) Body Height Vertical Temporal Derivative: Fall follows by a sudden decrease in body height (c) Body Orientation: In a Fall, angel between body and ground is expected to go from 90 to 0 (d) Body Orientation Temporal Derivative: A sudden change in body orientation is expected in Fall (e) Distance between COM and body supports, i.e., feet.

in this work.

Also we imposed some selectivity on the skeletal data even if they are available that affects the quality of features that we would be using. In this selectivity policy, we defined a criteria named "tracking" criteria. The criterion for the user to be described as being "tracked" is as follows:

Each frame and its 15 previous and 15 next frames should satisfy the following:

1) At least 60% of joints should exist in those frames, because to get an idea of the feature $f_2(n)$(which is turn is based on skeletal norm), we need to be assured that we have at least some significant and reliable part of the skeleton available to us.

2) At least one of the upper body joints (head, neck, left shoulder and right shoulder) should exist in those frames for Fall detection. This is because sometimes one of the upper body joints may not be available. Hence we reduce the probability of outage of skeletal data.

On the recordings, we had the time stamp of the events that we wanted to detect. In our skeletal data, we know the frame number of each observation $[s_i]_{i=1:N}$. N is the total number of frames we have. As we know the frame rate of the skeletal data, i.e., 30 Hz, we can draw a one to one correspondence between the skeletal data and class labels corresponding to events (which are visible from video data).

Table 3: Events and their corresponding labels in high motion detection.

Event	Label
Normal	0
Fall	1
High Motion	2

We note that $f2$ has nothing to do with Fall and $f1$ has nothing to do with detecting High Motion, because those features are in-built to detect two different events. Although not uncorrelated, they carry little information about each other. Hence for computational simplicity, we decide to use them separately. Both are in fact factor loadings [39] for the Fall and High Motion events. Hence, during training and testing our features should be ($f1$) for Fall/Not Fall classifier, ($f2$) for High Motion/Not High Motion classifier and $(f1 \;\; f2)'$ for Normal/Not Normal classifier.

To the Normal/not Normal classifier, we present the whole unfactored data set, (i.e., without omitting f1 or f2), both during training and testing. This classifier can provide a reinforcing evidence if either of the above classifiers goes wrong due to some reason. This is because we have defined it as a three-class problem and two decisions are enough to form some sort of a decision boundary. Therefore in reporting the experiments results, we will report the results with both factored and unfactored features, so that we can assess our idea of factorization which was mentioned above.

In this chapter the detailed paths and ideas of the proposed fall detection algorithms were described. In chapter 4 the performance and the final experimental results of these ideas will be reported and discussed.

CHAPTER IV

EXPERIMENTS AND RESULTS

4.1 Data Collection

In order to study fall and experiment the ideas, we should have data, either real or synthetic data. As we didn't have access to real fall data occurred by elderly people, we had to simulate this event in different positions and conditions. As we should have played this role as natural as possible, we had to watch some videos showing the event in real world. Figure 8 shows the real falling sequence in some of these videos [6].

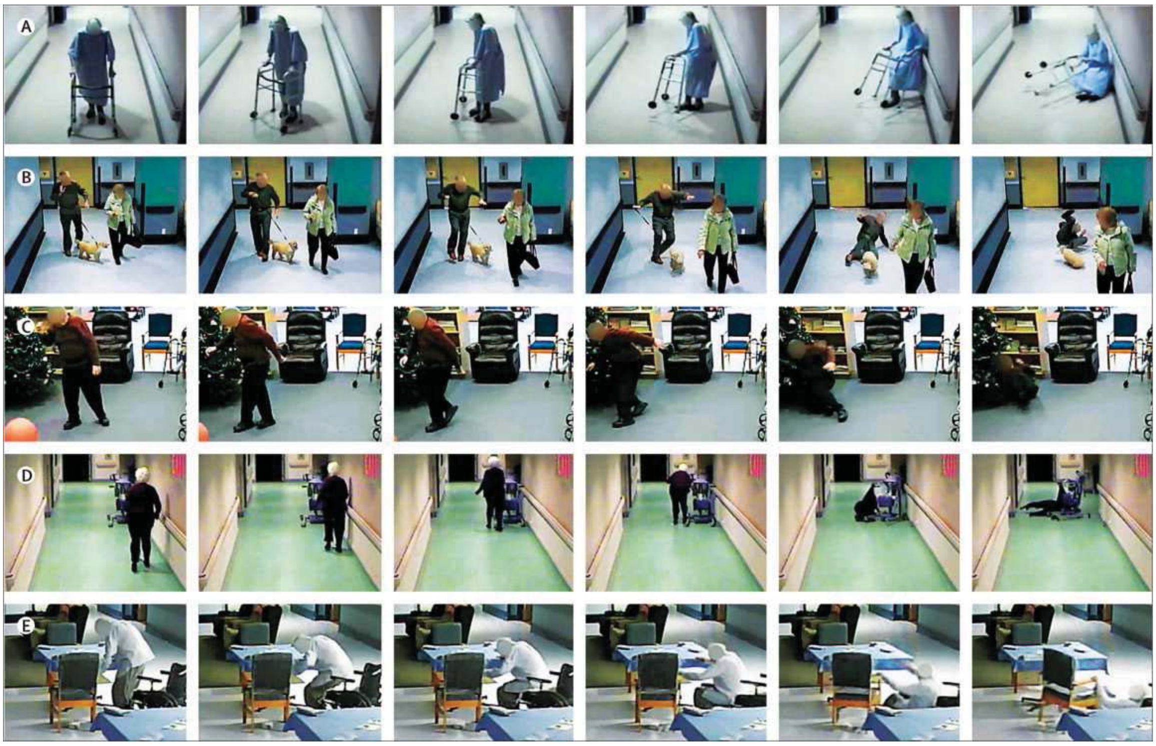

Figure 8: Falling event sequences in real videos. Images is borrowed from [6].

In the next section, the complete setup which was used for recording and collecting data will be described.

4.1.1 Data Collection Setup

4.1.1.1 Kinect

Kinect is a motion sensing input device developed by Microsoft corporation. It was firstly developed for Xbox video game console but later a version for Windows was released on February 1, 2012.

There are many wrappers and drivers released officially by the related organization like Microsoft and unofficially by hackers for Linux/Windows since 2012. Microsoft released Kinect SDK (software development kit) for Windows 7 on June 16, 2011. This SDK was meant to allow developers to write Kinecting apps in C++/CLI (Command Line Interface), C#, or Visual Basic .NET. OpenNI is an example of open source interfaces to interact with Kinect in windows and Linux.

Figure 9 shows the audio and video sensors implemented in Kinect. As it can be observed, Kinect has two microphone arrays, one simple RGB camera and a pair IR sensors.

Figure 9: Kinect's sensors. Image borrowed from [7]. Last visited on July 19. 2013.

The Kinect's various sensors output video at a frame rate of 9 Hz to 30 Hz depending on resolution. The default RGB video stream uses 8-bit VGA resolution (640 480 pixels) with a Bayer color filter, but the hardware is capable of resolutions up to 1280x1024 (at a lower frame rate) and other colour formats such as UYVY. The monochrome depth sensing video stream is in VGA resolution (640 480 pixels) with 11-bit depth, which provides 2,048 levels of sensitivity. The Kinect can also stream the view from its IR camera directly, i.e., before it has been converting into a depth map, as 640x480 video, or 1280x1024 at a lower frame rate. The Kinect sensor can maintain tracking through an extended range of approximately 0.76 m (2.320 ft). The sensor has an angular field of view of 57 horizontally and 43 vertically, while the motorized pivot is capable of tilting the sensor up to 27 either up or down. The horizontal field of the Kinect sensor at the minimum viewing distance of 0.8 m (2.6 ft) is therefore 87 cm (34 in), and the vertical field is 63 cm (25 in), resulting in a resolution of just over 1.3 mm (0.051 in) per pixel. The microphone array features four microphone capsules and operates with each channel processing 16-bit audio at a sampling rate of 16 kHz.

Figure 10 shows a sample image recorded by Kinect's RGB video camera. Figure 11 and 12 show sample depth and IR images recorded by Kinect's IR sensors.

Figure 10: RGB Image Sample.

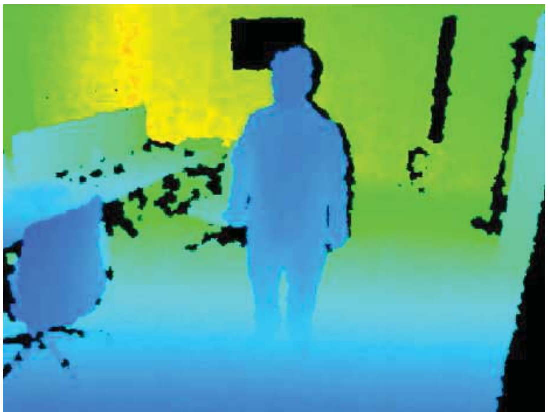

Figure 11: Depth Image Sample.

Figure 12: IR Image Sample.

4.1.1.2 OpenNI

In this research work, we mainly used OpenNI interface to communicate with Kinect and record data and process them. OpenNI or Open Natural Interaction is an industry-led, non-profit organization focused on certifying and improving interoperability of natural user interface and organic user interface for natural interaction devices, applications that use those devices and middleware that facilitates access and use of such devices. Natural Interaction Devices or Natural Interfaces are devices that capture body movements and sounds to allow for a more natural interaction of users with computers in the context of a Natural user interface. The Kinect and

Wavi X-tion are examples of such devices. The OpenNI framework provides a set of open source APIs. These APIs are intended to become a standard for applications to access natural interaction devices. The API framework itself is also sometimes referred to by the name OpenNI SDK. The APIs provide support for:

- Voice and voice command recognition
- Hand gestures
- Body Motion Tracking

Using OpenNI processing programs, a simple raw depth data recorded by Kinect can give us many information like skeletal shape data and a silhouette/blob based on the body of the person standing in front of the Kinect. Sample produced body skeleton and blob can be observed in Figure 13 and 14.

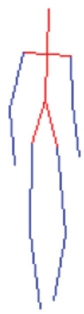

Figure 13: Skeleton Produced by OpenNI.

4.1.2 Recorded Data Statistics and Samples

By means of the frameworks mentioned in the previous sections, and having the real falling videos in mind, we started simulating different falling events and recording them. We tried to execute these activities as natural as possible and also as rich as possible in case of different activities and not only falls. The recorded data set consists of different activities like falling, sitting on the chair, sitting on the ground,

Figure 14: Silhouette/Blob produced by OpenNI.

lying on the ground and high motion activities. Totally 19 video were recorded consist of 3 hours and 47 minutes and 19 seconds with 409185 frames. Table 4 shows the recorded activities statistics in the recorded data set.

Table 4: Statistics of activities in the recorded data set.

Activity Type	Number of Occurrences (times)
High Motion	120
Sitting On the Ground	37
Sitting On Chair	48
Lying on the ground	64
Falling	176
Normal (Standing and Walking)	Rest of the time

The instances of some activities in the data set are shown in the Figure 15, 16 and 17.

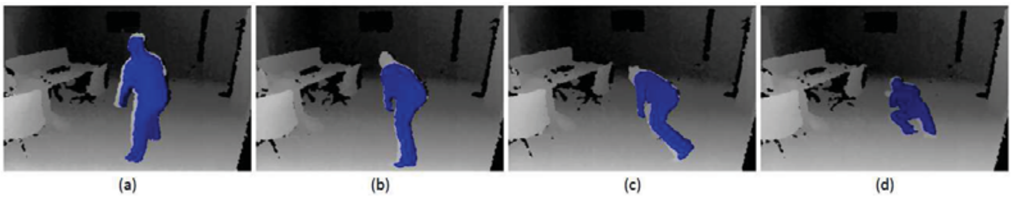

Figure 15: Fall instance sequence. Person is detected using a silhouette/blob.

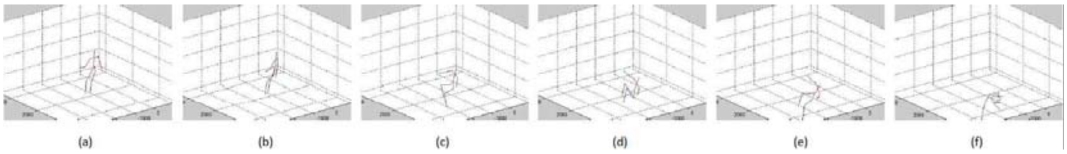

Figure 16: Fall instance sequence. Body skeleton is reconstructed using the joints locations given by OpenNI.

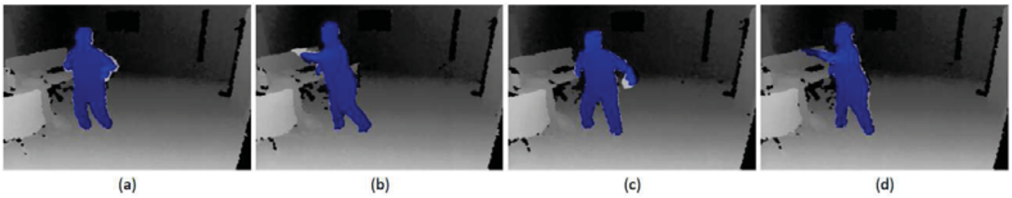

Figure 17: High motion instance sequence. Person is detected using a silhouette/blob.

4.1.3 Data Labeling

In labeling the recorded activities in the data set, we have video frames which should be categorized based on the observation of a human which makes this work difficult, time consuming and not unique for all instances. Therefore, we proposed a definition for fall to do the labeling as similar as possible for all fall instances in the data set. Fall is considered to be happened from the time the person looses control and can't recover to the balanced position till he/she completely lie on the ground. All the frames in this period would be labeled as fall. We did the same for other activities as well. The definitions are:

- High Motion: Starts from the time the person starts doing some activities with high motion (kicking, punching, etc.) and ends by the time he/she finish doing that.
- Sitting On the Ground: Starts when the object start descending from the not-sit position till fully sat on the ground
- Sitting On Chair: Starts when the object start descending from the normal

position till fully positioning on the chair.

- Lying on the ground: Starts when the object start descending from the not-lied position till fully lied on the ground.
- Falling: Starts from the time the person looses control and can't recover to the balanced position till he/she completely lie on the ground.
- Normal: All activities rather than the upper mentioned ones would be labeled as normal.

The raw recorded data set was in .oni format which contained the RGB and depth video data. The problem was that it couldn't be opened and viewed by normal video players for labeling. Therefore our group wrote a program in C++ for converting the .oni data set to two .avi file formats; one for RGB video and one for depth video data. Figure 18 shows a snapshot of this program's graphical user interface and Figure 19 shows the snapshot of the output of this program.

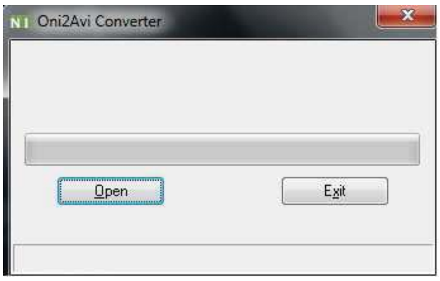

Figure 18: Snapshot of the .oni converter program's graphical user interface

In order to check whether the labels which are the ground truth for the classifier are created right, we should have checked them again. For evaluating the created labels frame by frame we wrote a program in visual C#. In this program the main target was checking the labels which were fed to the classifier as training data. So

Figure 19: Snapshot of the .oni converter program's video output

even if there is one frame mistake during labeling, the classifier will get confused between classes and will give unexpected results. The program is written so that the videos and the labels can be fed to it and one can see each event based on the labels and check if start and end of the event (according to labels) are correct or not. A sample snapshot is shown in Fig. 20.

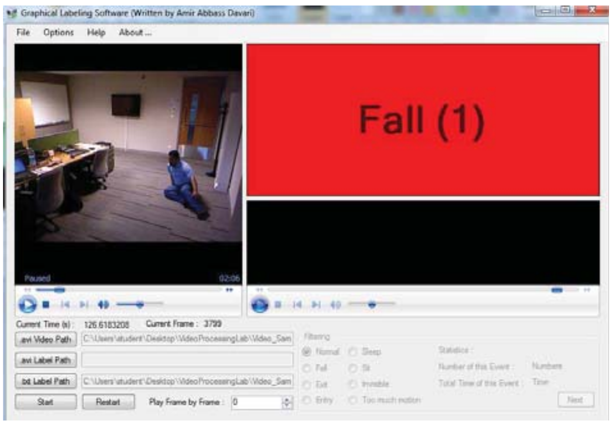

Figure 20: Snapshot of the labeling evaluation program graphical user interface

4.2 Experimental Results

As we tried two separate experiments, the results will be reported separately in the following sections.

4.2.1 Experimental Results for Idea 1

By feeding the five dimensional windowed feature vector described in previous sections as the training data and their corresponding labels to the classifier and taking advantage of 3-fold cross validation, the following results were achieved:

TP rate = 89.82%

TN rate = 99.86%

FN rate = 10.17%

FP rate = 0.14%

Which are calculated from the confusion matrix 21.

$$\begin{pmatrix} 309 & 35 \\ 4 & 2804 \end{pmatrix} \tag{21}$$

In which the first column of the confusion matrix 21 represents the fall events and the second column represents the not-fall activities. By means of these results, we can calculate $accuracy = 98.76\%$, $precision = 98.72\%$, $recall = 89.83\%$, $specificity = 99.86\%$ and $F - score = 0.94$. The given results are window-based result. In other words, during comparing the predicted values and target values, they were checked window by window.

Furthermore as an additional experiment, we fed the frame based features (not windowed features) to the classifier. we checked them based on events occurrences. For all fall events, at least one frame was detected and made alarm as fall and non of them were skipped. In addition just one false alarm was observed. Therefore for all fall occurrence, an alarm was raised by the algorithm and only one false alarm was raised.

4.2.2 Experimental Results for Idea 2

4.2.2.1 Need for Balancing the Data Set

This section shows that if there is no means of balancing the skewed data set there would be some unwanted confusion between classes. We present the following three confusion matrices for the problem. Without factoring, the individual classifiers, reported F-numbers as (0.68 (Normal/not Normal), 0.93 (Fall/not Fall), 0.34 (High Motion/not High Motion)) and with factoring F-numbers will be (0.67 (Normal/not Normal), 0.93 (Fall/not Fall), 0.34 (High Motion/not High Motion)). So we must design our tree (for this simulation), such that the upper has better performance. Hence according to the given F-numbers, the upper node is Fall/not Fall and the next node is Normal/not Normal. The detailed results obtained by the architecture shown in Figure 21 is written in section 4.2.2.

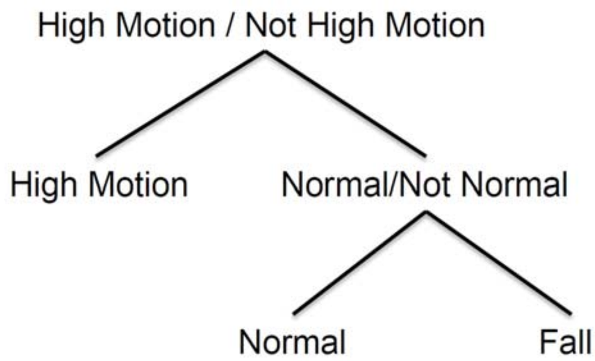

Figure 21: Architecture of the classifier before adding synthetic data.

The poor performance is obvious from confusion matrixes 22 and 23. But what we see is that there is high confusion between the Normal and Too-much-motion classes. This is due to the skewness of the too-much-motion class which can be removed by adding synthetic data, because of the need for it. Hence we decide to balance our data set by generating synthetic data from the histogram of the observations. Also we see a pattern; there is a high confusion between Normal and High Motion which is due to class skewness of the data set (in an information theoretic sense).

4.2.2.2 Choosing the Order of the Classifiers in the Classifier Tree after Adding Synthetic Data

First of all we estimated the pdf of High Motion class using Epanechnikov kernels, generate synthetic data from it and add it to our data set. The details of making and adding synthetic data to the original data set will be in described in details in chapter 4.2.2.3.

We checked each of the three classifiers' performance over simulations. For unfactored data, individual classes' F-numbers were 0.9700 for Normal/not Normal, 0.9587 for Fall/not Fall and 0.9714 for High Motion/not High Motion. For factored data, individual classes' F-numbers were 0.9705 for Normal/not Normal, 0.9507 for Fall/not Fall and 0.9713 for High Motion/not High Motion. Again it is obvious that the factorization doesn't affect the results much. And also it is obvious that the performances increased drastically after adding synthetic data. Hence according to these F-numbers (which is a good reference of the classifier performance), we decided to put High Motion classifier at the top node and Normal/not Normal classifier at the bottom. A Fall would be signalled if it is not High Motion and not Normal. This architecture which is shown in Fig. 22, will be fixed and our final classifier tree for detecting the mentioned events.

4.2.2.3 Synthetic Data Generation and Balancing the Classes

As we are dealing with skewed classes we found it necessary to generate synthetic data. The pdf of the data for High Motion (because it is the least informative class) was estimated using Epanechnikov kernels. From that estimated pdf random samples

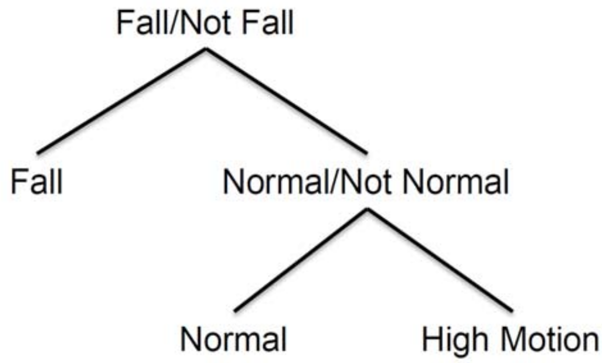

Figure 22: Architecture of the classifier after adding synthetic data.

were drawn. The synthetic data was generated as follows: with 70% probability data is drawn from the original masked data set (which contains Fall, High Motion and Normal events) and for remaining 30% of time more High Motion synthetic data, which is generated by sampling from the estimated distribution, was plugged in.

The rationale for this approach was found as follows;

The percentage of the synthetic data added to the original data set was varied from 0 to 85% and found an optimal ratio in which the synthetic data and the original data should be mixed. In this cross validation procedure we found an optimal mixing ratio to be 0.7 as it is shown in the Figure 23.

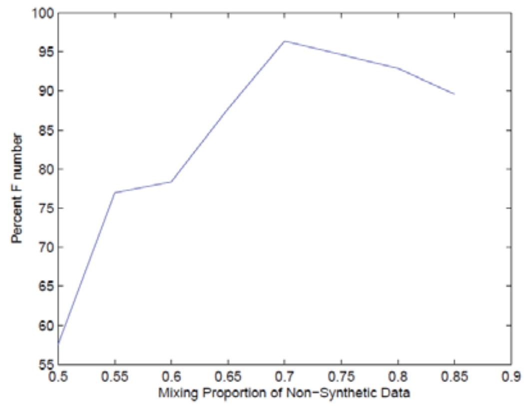

Figure 23: F-score v.s. the percentage of the synthetic data.

The results we got by the architecture shown in Figure 21 are shown in confusion matrixes 22 and 23. The following confusing matrix 22 is with factored data.

$$\begin{pmatrix} 87.42 & 0.00 & 66.39 \\ 12.57 & 100.00 & 0.27 \\ 0.00 & 0.00 & 33.33 \end{pmatrix} \tag{22}$$

F-numbers for the above confusion matrix 22 will be (0.68 (Normal/not Normal), 0.93 (Fall/not Fall), 0.50 (High Motion/not High Motion)).

If we remove factoring, we will get the following confusion matrix 23,

$$\begin{pmatrix} 87.42 & 0.00 & 66.94 \\ 12.54 & 100.00 & 2.72 \\ 0.00 & 0.00 & 33.33 \end{pmatrix} \tag{23}$$

F-numbers for the above confusion matrix 23 will be (0.68 (Normal/not Normal), 0.92 (Fall/not Fall), 0.48 (High Motion/not High Motion)).

We see that keeping/not keeping factoring does not alter the results much. The results are still not acceptable according to 23 and its F-numbers.

Using the new balanced data set (mixed up with synthetic data), we got the confusion matrix 24, averaged over three folds [40] for unfactored data.

$$\begin{pmatrix} 97.18 & 8.86 & 5.17 \\ 2.81 & 91.13 & 0.00 \\ 0.00 & 0.00 & 94.83 \end{pmatrix} \tag{24}$$

Which the F-numbers are 0.92 for Normal/not Normal, 0.94 for Fall/not Fall and 0.97 for High Motion/not High Motion.

And for factored data, confusion matrix 25 was reported by the classifier tree.

$$\begin{pmatrix} 94.60 & 0.00 & 5.08 \\ 5.39 & 100.00 & 0.0282 \\ 0.00 & 0.00 & 94.88 \end{pmatrix} \tag{25}$$

Which the F-numbers are 0.95 for Normal/not Normal, 0.97 for Fall/not Fall and 0.97 for High Motion/not High Motion.

It is obvious that factorization increased the F-number of the classes classified by the classifier tree. So this experiments prove our theory of using factorization technique. And the theory of adding synthetic data was proved by the previous reported results.

We also have a live system running in our laboratory which flags the occurring events. For developing this software, we connect Maltab engine and visual C++ together. Snapshots of the online fall detector software can be seen in figures 24 and 25. We can see in Fig. 24 that somebody has fallen down and an alarm was raised for the fall event as an online detection. Also, a high motion event detection is shown in Fig. 25.

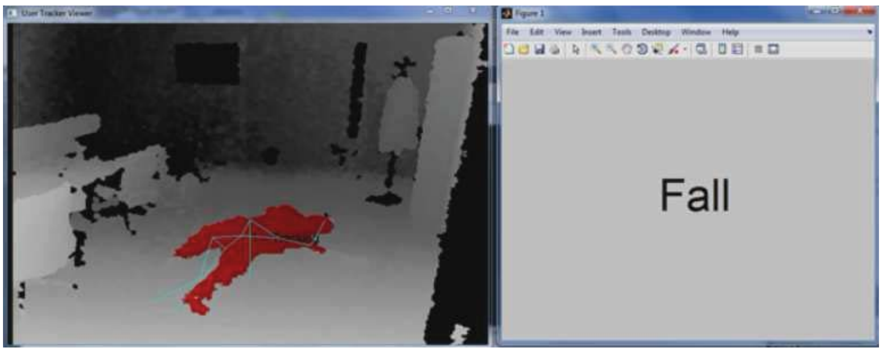

Figure 24: Online detection of fall event.

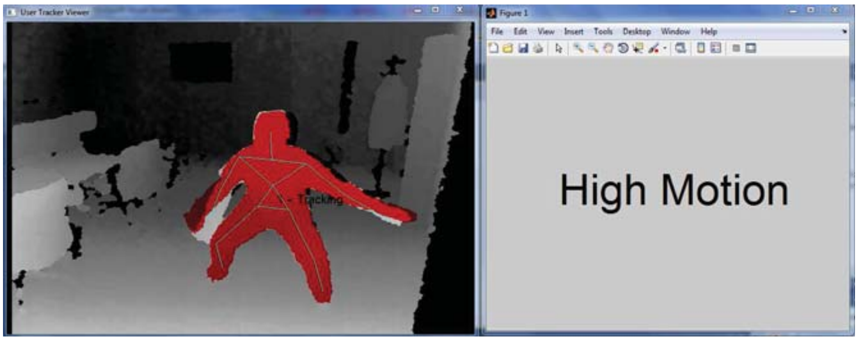

Figure 25: Online detection of a high motion instance.

Also the algorithm produced false alarms and in some cases it misses some events. Figure 26 shows a high motion event which is missed by our algorithm. Figure 27 shows a false alarm raised as a wrongly detected fall event.

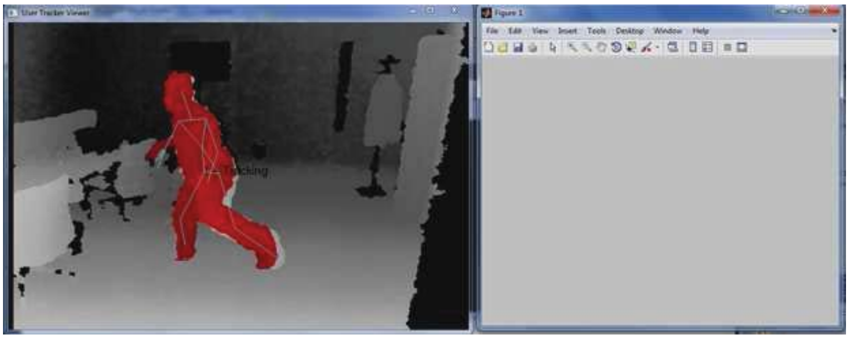

Figure 26: Instance of a high motion activity missed by our online event detection system.

4.3 Comments on the Results

Based on the achieved results for idea 1 in section 4.2.1, the proposed features have a reasonable performance. Sensitivity is 89.82% and specificity is 99.86%. Precision is high which means a Fall is hard to be misclassified and the recall is low which means false alarms are rare. The improvement in comparison to the state of the art is the false alarm rate which is decreased to less than 0.5%.

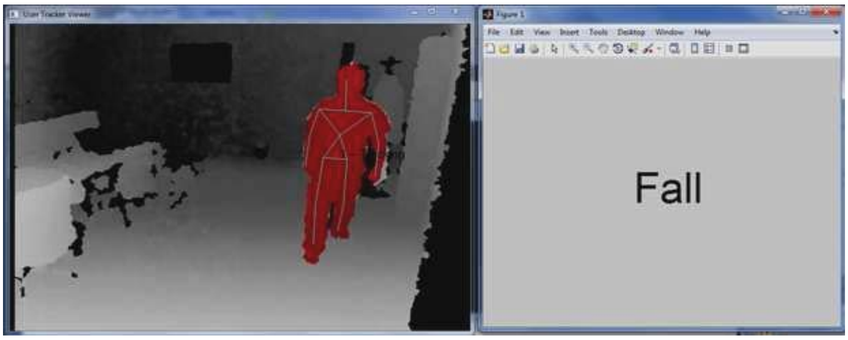

Figure 27: Instance of a false alarm raised by our online event detection system.

The results in idea 2 in section 4.2.2 shows a great achievement in fall detection accuracy, i.e., 100% although it has around 6% false alarms which is still comparable to the state of the art.

CHAPTER V

CONCLUSION

5.1 Contribution of the Thesis

Quality of life is an important fact which shouldn't be neglected. As the science power is advancing second by second, we should use it to increase our quality of life. As it was described in the introduction, falling is one of the problems old people face. This thesis tried to have a review on related researches and achievements and propose novel ideas to increase the performance of fall detection systems in order to make it a usable and acceptable technology in daily life. One of the main problems in the previous works, is that although they reach to a very high detection rate of fall events, but considerable rate of false alarms makes them unusable. The novel feature and ideas in this thesis, almost eliminates the false alarms while it has a reasonable performance in detecting most of the fall events.

Furthermore all of these achievements have obtained by means of depth video data which means not only the user doesn't need to wear sensors or markers or ..., but also their privacy will be preserved as the camera is IR sensors and not RGB video recorders. To reach this goal, an extensive depth video data-set is recorded via Kinect in this thesis that can be published for further researches on this topic.

Also this improvement in results was obtained using a low dimensionality feature space which prevent the complexity of classification. Extracting features containing high information and intelligent data preprocessing helped this work to have such an achievement.

5.2 Future Works

There are limitations in using skeleton data. It is not always reliable. Noise and limited range is its major limitations. Figure 28 and 29 shows some instances of the skeletal data unreliability.

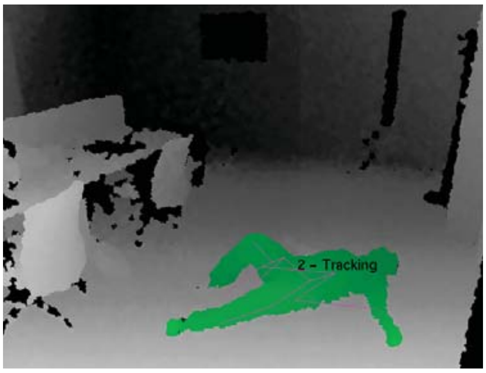

Figure 28: Unreliable skeleton of a person while lying on ground.

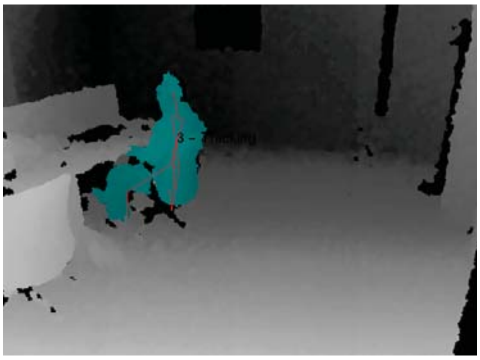

Figure 29: Unreliable skeleton of a person while sitting on a chair.

Hence we must look for alternative features which are more reliable and available for detecting various events. For this purpose, we have extracted blobs from the depth data in different scenarios like the person being in normal condition, lying, siting on the ground and sitting on the chair. Figures 30, 31 and 32 shows some examples of the extracted body blob. We will extract features from the blob and train our classifier to detect these events in which the skeletal data is unreliable.

Also there is another idea of using blob morphology classification which can assist the accuracy of the fall detection. Parts of this idea are completed and implemented and it will be applied to this algorithm in the early future.

Figure 30: Blob of a person while standing.

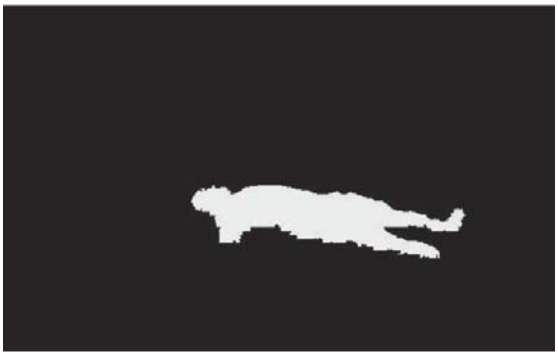

Figure 31: Blob of a person while lying down.

Figure 32: Blob of a person while sitting.

Bibliography

[1] S. Zambanini, J. Machajdik, and M. Kampel, "Detecting falls at homes using a network of low-resolution cameras," in *Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on*, pp. 1–4, IEEE, 2010.

[2] R. Planinc and M. Kampel, "Introducing the use of depth data for fall detection," *Personal and Ubiquitous Computing*, pp. 1–10, 2012.

[3] C. Zhang, Y. Tian, and E. Capezuti, "Privacy preserving automatic fall detection for elderly using rgbd cameras," in *Computers Helping People with Special Needs*, pp. 625–633, Springer, 2012.

[4] "http://netscale.cse.nd.edu/twiki/bin/view/edu/kinectrehabilitation,"

[5] "http://bodyofchristchristiancentre.com/wp-admin/non-included-angle-definition-4916.jpg,"

[6] S. N. Robinovitch, F. Feldman, Y. Yang, R. Schonnop, P. M. Leung, T. Sarraf, J. Sims-Gould, and M. Loughin, "Video capture of the circumstances of falls in elderly people residing in long-term care: an observational study," *The Lancet*, vol. 381, no. 9860, pp. 47 – 54, 2013.

[7] "http://www.xbox360-iso.com/wp-content/uploads/2012/06/d8b5kinect.jpg,"

[8] "Oldies, depressed people more likely to take a tumble," 18 June 2008.

[9] "Simple strategies can reduce falls and liability: Women and elderly fall more frequently," 2004.

[10] L. Kowalczyk, "Spending on health care rises 7 percent in hospitals, drug costs contribute to faster acceleration 12," 8 Jan. 2002.

[11] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 10, no. 1, pp. 156–167, 2006.

[12] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, and N. H. Lovell, "Barometric pressure and triaxial accelerometry-based falls event detection," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 6, pp. 619–627, 2010.

[13] T. Tamura, T. Yoshimura, M. Sekine, M. Uchida, and O. Tanaka, "A wearable airbag to prevent fall injuries," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 6, pp. 910–914, 2009.

[14] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable sensors for reliable fall detection," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pp. 3551–3554, IEEE, 2006.

[15] S. Luo and Q. Hu, "A dynamic motion pattern analysis approach to fall detection," in *Biomedical Circuits and Systems, 2004 IEEE International Workshop on*, pp. 1–5, IEEE, 2004.

[16] H. Ghasemzadeh, R. Jafari, and B. Prabhakaran, "A body sensor network with electromyogram and inertial sensors: multimodal interpretation of muscular activities," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 2, pp. 198–206, 2010.

[17] M. Alwan, P. J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, "A smart and passive floor-vibration based fall detector for elderly," in *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, vol. 1, pp. 1003–1007, IEEE, 2006.

[18] R. Planinc and M. Kampel, "Introducing the use of depth data for fall detection," *Personal and Ubiquitous Computing*, pp. 1–10, 2012.

[19] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from rgbd images," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 842–849, IEEE, 2012.

[20] G. Mastorakis and D. Makris, "Fall detection system using kinects infrared sensor," *Journal of Real-Time Image Processing*, pp. 1–12, 2012.

[21] X. Yang, C. Zhang, and Y. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in *Proceedings of the 20th ACM international conference on Multimedia*, pp. 1057–1060, ACM, 2012.

[22] X. Yang and Y. Tian, "Eigenjoints-based action recognition using nave-bayes-nearest-neighbor," Second International Workshop on Human Activity Understanding from 3D Data in conjunction with CVPR, 2012.

[23] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier, "Fall detection from depth map video sequences," in *Toward Useful Services for Elderly and People with Disabilities*, pp. 121–128, Springer, 2011.

[24] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-markov model," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 838–845, IEEE, 2005.

[25] T. Liao and C.-L. Huang, "Slip and fall events detection by analyzing the integrated spatiotemporal energy map," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 1718–1721, IEEE, 2010.

[26] Z. Zhang, W. Liu, V. Metsis, and V. Athitsos, "A viewpoint-independent statistical method for fall detection," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 3626–3630, IEEE, 2012.

[27] N. Zouba, F. Bremond, and M. Thonnat, "Multisensor fusion for monitoring elderly activities at home," in *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pp. 98–103, IEEE, 2009.

[28] X. Yu, "Approaches and principles of fall detection for elderly and patient," in *e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on*, pp. 42–47, IEEE, 2008.

[29] E. Stone and M. Skubic, "Evaluation of an inexpensive depth camera for in-home gait assessment," *Journal of Ambient Intelligence and Smart Environments*, vol. 3, no. 4, pp. 349–361, 2011.

[30] E. E. Stone and M. Skubic, "Evaluation of an inexpensive depth camera for passive in-home fall risk assessment," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference On*, pp. 71–77, IEEE, 2011.

[31] G. Mastorakis and D. Makris, "Fall detection system using kinects infrared sensor," *Journal of Real-Time Image Processing*, pp. 1–12, 2012.

[32] Z. Zhang, W. Liu, V. Metsis, and V. Athitsos, "A viewpoint-independent statistical method for fall detection," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 3626–3630, IEEE, 2012.

[33] T. Liao and C.-L. Huang, "Slip and fall events detection by analyzing the integrated spatiotemporal energy map," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 1718–1721, IEEE, 2010.

[34] R. Dubey, B. Ni, and P. Moulin, "A depth camera based fall recognition system for the elderly," in *Image Analysis and Recognition*, pp. 106–113, Springer, 2012.

[35] F. V. A.M.Srp, "Fall detection from depth map video sequences,"

[36] G. Diraco, A. Leone, and P. Siciliano, "An active vision system for fall detection and posture recognition in elderly healthcare," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010*, pp. 1536–1541, IEEE, 2010.

[37] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier, "Fall detection system for older people," in *Toward Useful Services for Elderly and People with Disabilities*, pp. 121–128, Springer, 2011.

[38] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[39] E. Alpaydin, *Introduction to machine learning*. MIT press, 2004.

[40] G. Fung and O. L. Mangasarian, "Proximal support vector machine classifiers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 77–86, ACM, 2001.

[41] J. Díaz, E. Ros, F. Pelayo, E. M. Ortigosa, and S. Mota, "Fpga-based real-time optical-flow system," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 2, pp. 274–279, 2006.

VITA

Amir Abbas Davari was born in 1989 in Tehran, Iran. He finished his high school and pre-university studies in national organization of development of exceptional talents (NODET). During high school in 2006, he could be the semi-finalist in Mathematics national Olympiad and quarter-finalist in Physics national Olympiad. In 2007 in national university entrance exam, he could achieve the rank of 128 among almost 300000 participants and be accepted in University of Tehran, the best and most prestigious university in Iran. He studied Electrical Engineering. He chose Bio-Medical Engineering as his major in B.Sc. In 2011, Amir Abbas came to Özyeğin University to continue his studies in M.Sc. electrical engineering, Computer Vision field. He is working with Professor Dr. Tanju Erdem on fall detection for elderly people using Kinect. His main research interests are image processing, signal processing, video signal processing, pattern recognition and machine learning.