Video - based Fall Detection in Elderly's Houses

Saleh Alaliyat

Master's Thesis
Master of Science in Media Technology
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2008

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Video - based Fall Detection in Elderly's Houses

Saleh Alaliyat

2008/06/30

Abstract

Automatic detection of a falling person based on video analysis is an important problem with application in safety areas including supportive home environments and elderly's houses. The use of computer vision systems offers a good solution to analyze people behavior and detect unusual events. Compared to classical methods that use sensors-based solutions, the video-based methods improve the performance with lower costs and give more functionality.

In this project we have developed a method to detect a human falling event in elderly's houses. The proposed method is video-based and its mainly includes a network web camera (axis 207w) [1] , image processing, audio analysis and recognition algorithms. Only one camera is used and novel classification features are extracted from the camera images.

Our method is based on a combination of several features extracted from the foreground segmentation images. The extracted features are: aspect ratio after X-Y projections, height of the center of mass, motion quantity, motion history image (MHI), orientation, speed, major axis and minor axis. In addition, the audio track of the video is also used to distinguish a fall event from other events. The extracted features from audio track are the variation and zero-crossing rate. Our method combines all these features to get better decision in real time.

The system is implanted in MATLAB, and gives good results on the experimental datasets. The experimental datasets was made individually and the results are presented at the end of the report.

Keywords: Video analysis, video surveillance, audio signal analysis, segmentation, object tracking, feature extraction, event detection, fall detection, pattern recognition.

[1] http://www.axis.com/products/cam_207/index.htm last visited 17.05.2008

Sammendrag

Automatisk oppdagellse av en fallende person som blir basert påvideoanalyse er et viktig problem med anvendelse i sikkerhetsområder samt støttende hjemmiljøer og gamlehjem. Bruken av datavisjonssystemer byr en god løsning til åanalysere menneskeppfrsel og til åoppdage noen uvanlige hendelser. Sammenliknet med klassisk metoder som bruker sensorløsninger, video-basert metodene forbedret prestasjonen med lave kostnader og gir mer funkjsonalitet.

I dette prosjektet utviklet vi en metode til åobservere mennesket som faller i et gamlehjem. Den foreslåtte metoden er video-basert og det inkluderer hovedsakelig et nettverkswebbkamera (axis 207w) [1], avbildeprosess, lyd-analyse og anerkjennelsesalgoritmer. Bare et kamera er brukt og et ferskt klassifikasjonskjennetegn er trukket ut fra kamerabilder, den tilknyttete anerkjennelsen algoritme for fallende hendelsesoppdagelse er implementert.

Vår metode er basert påen kombinasjon av mange trekte ut (ekstakt) kjennetegn fra forgrunnene segmenteringene avbilder. De trekte ut (ekstrakt) kjennetegnene er : aspektforhold etter X - Y projeksjoner, høyde av midtpunktet av massen, bevegelse kvantitet, bevegelse historieavbilde, orientering, hastighet, major og minor akse. I tillegg blir audiosporet av videoen ogsåbrukt til åskille mellom fallbegivenhet fra andre begivenheter. De ekstrakt funksjoner fra lydsporet er det variasjon og null-krysset rate. Vår metode kombinerer alt for ågi bedre avgjørelse i riktig tid.

Systemet er implantert i MATLAB, og gir gode resultater påden eksperimentale datasets. Den eksperimetale datasett er laget individuelt og resultatet er til stede ved slutten av denne raport.

Nøkkelord: Videoanalyse, videoovervåkning, lydsignal analyse, segmentering, objektsporing, kjennetegnsuttrekking(ekstrakting), hendelsesoppdagellse, mønsteranerkjennelse.

[1] http://www.axis.com/products/cam_207/index.htm last visited 17.05.2008

Preface

This Master's thesis was carried out at Gjøvik University College, Department of Computer Science and Media Technology, in the period from January 2008 to July 2008, under the supervision of Dr. Tech. Faouzi Alaya Cheikh. I would like to thank Dr. Faouzi Alaya Cheikh for his guidance, advices, expertise, motivation, encouragement, and for following me up during the thesis work. Also I would like to thank my fellow students for their encouragement and for helping me with MATLAB related problems.

I would also like to thank my father, my mother, my brothers and sisters for their encouragement during these last two years.

Saleh Alaliyat, 2008/06/30

List of Figures

List of Tables

Contents

1 Introduction

1.1 Topics covered by this thesis

The main goal of this thesis is to detect person falling events in elderly's houses based on video and give an alarm when a fall is detected in the real-time. New method is proposed to detect falling events by combining low level features extracted from the video and audio track to classify the fall event.

1.2 Background

The background of this project is the analysis of video stream from camera (video processing) and extracting features from foreground objects to analyse the human behaviour. This includes video motion detection, image segmentation, motion-based tracking, object classification, and features extractions for behavioural analysis.

1.3 Problem description

In response to the problem of growing population of seniors, we need to think of developing new technologies to ensure the safety of elderly people. According to health centers, they are starting to face a problem of lacking employees to take care of the seniors, and this is due to the fast growing number of seniors. New video surveillance technologies can help seniors to live independent by providing them a secure environment and improving their quality of life. The use of computer vision systems offers solutions to analyze people's behavior and detect unusual or abnormal events; e.g. a person running, fighting or falling down.

Falling down is the greatest danger facing old people living alone. The majority of injury-related hospitalizations for seniors are results of falls [1]. And the situation will be much worst if the person can not call for help. Detection of moving objects in video streams is known to be an important and challenging research problem. The moving object in our case is a person moving in a room (indoor surveillance). The person is monitored by a webcam with microphone. The video stream from the camera will be analyzed to distinguish the moving object (person) from the background. Then the system will extract the information (important features for behavioral analyses, falling event in this case) of that moving object, and check if it's in a falling down event or a normal motion. The system will also extract classification features from the audio track to discriminate the falling down event from other events. Then the system will combine all the extracted information from the video sequence and audio track to detect the fall event and to confirm that it is not false detection. The system will be real-time system and fully automatic not to intrude in the private life of the monitored person.
The system flowchart is shown in Figure 1. We first extract the foreground (moving objects), then analyze the extracted objects (features extraction) to see if the condition of fall accident is met.

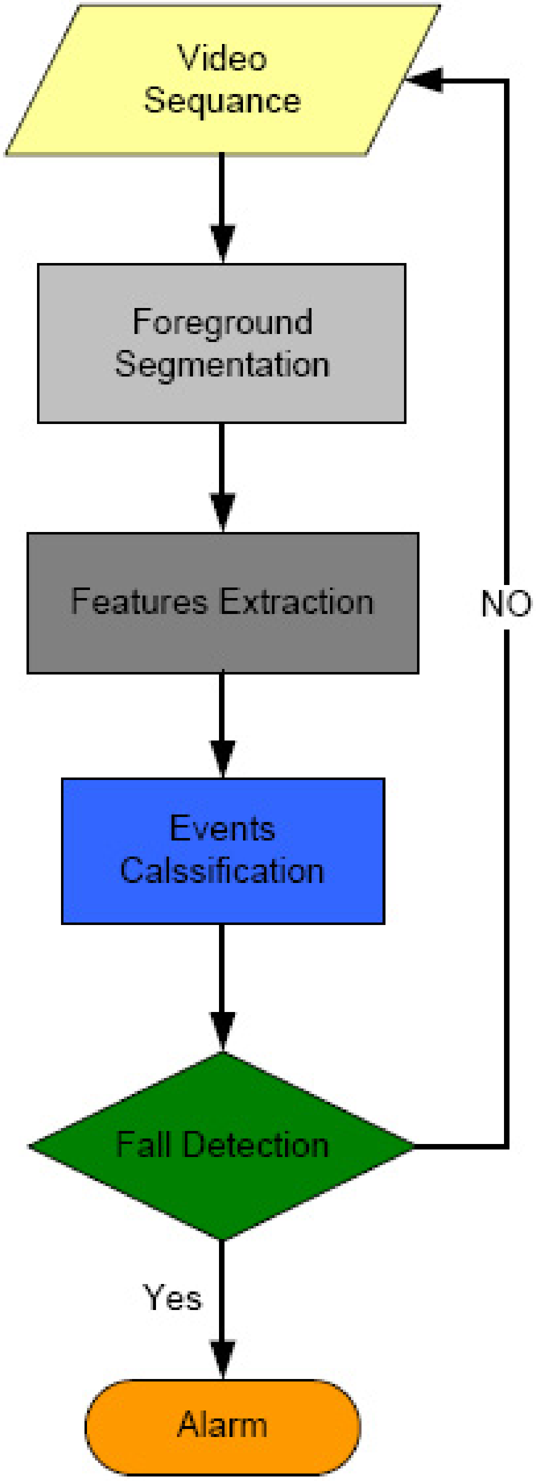


Figure 1: System overview

1.4 Motivation and benefits

Nowadays we are facing the problem of growing population of seniors particularly in the western countries including Norway. if we look at the statistics from the Public Health Agency of Canada (PHAC) [2] as an example, it gives a clear idea about the problem in the western countries. The majority of seniors resides in private house and spend a lot of time alone. Almost 62% of injury-related hospitalizations for seniors are the result of falls [1]. One of the greatest dangers for old people living alone is falling down. And the gravity of the situation can increase if the person can not call for help. Most of current technologies that are used to detect falls in use some wearable sensors like accelerometers [3] or infrared (IR) sensor [4] or help buttons. The problem of wearable sensors is that older people often forget to wear them. To overcome these problems, we decided to use a computer vision system to analysis the human behavior.

1.5 Stakeholders

This project will be interesting to the researchers in video-based event detection field, the elderly's houses, companies producing and developing video surveillance systems, and the individuals interested.

1.6 Research questions

The objective is to detect a person falling event in elderly house based on video analysis and give alarm. we propose these research questions:

Q1: How to utilize the existing detection techniques to detect a person falling event to achieve as few false or missed detections as possible?
Q2: How to distinguish between falling events from sitting or sleeping events?
Q3: Based on the existing techniques: how to develop automatic ways to use feedback to improve the detection systems (automatic learning method).
Q4: How to use the audio information from the microphone of the webcam to improve the detection system?

1.7 Choice of methods

The Methodology that has been used to answer the research questions and carry out the work in this project is summarized below:

1.7.1 Literature review

Recently lots of research work has been done on event detection in video surveillance field. There are many detection techniques proposed these days; we have been read most of them; a comparative study of these techniques has been done also, based on this knowledge we designed a method to detect a person falling event with few false or missed detections.

1.7.2 Design the detection method

The fall detection system has three main parts: segmentation, features extractions and events classification.

We worked on these parts in steps. First we designed the segmentation algorithm to get clear foreground image. Then we extract some features of the moving objects. We did study every feature individually and how it will contribute to the fall detection classifier. Some features were dropped out. we extract also some features from the audio track. After that we designed a method to exploit these features to classify the events and detect falls. We used K-NN algorithm as a main part of our classifier.

1.7.3 Implementation

The algorithms of the proposed fall detection system are implemented in MATLAB.

1.7.4 Setup the experiment

The place of the experiment was in the masterlab A220 room, A-building room no. A128 and in the corridor of the first floor in A-building. The network Axis 207w camera is used to capture the training data and the test data. We used hama CS-4711[1] microphone to record the audio waves. We took 24 short video sequences to train K-NN classifier. The training video sequences representing normal activities (walking, sitting down, standing up, and crouching down). In the final experiment we took nine video sequences that have different activities to test the system.

During the development of the system, we tested each algorithm individually and did reversal review iterations to improve the results.

1.7.5 Analysis of the results

During the implementation of each algorithm we were testing it individually and analyzing its results and then doing reversal review iterations to improve the results. The analysis methods were done by studying the results differences between different activities and its distributions.

The K-NN classifier was tested on a video sequence that was recorded for this purpose. The results were analyzed by synchronizing the output from the K-NN classifier and the observations that have taken manually from still images.

In the final experiment, we have applied some testing videos to the system with different activities (fall and others). We analyzed the results as:

- How many falls detected with our system (True Positive).
- How many falls not detected with our system (False Negative).
- How many lures detected as a falls in our system (False Positive).
- How many lured not detected as falls in our system (True Negative).

1.8 Project overview

Chapter 2 - Related work

This chapter is a review of the state of the art related to the topics covered by this thesis. The topics are foreground segmentation, tracking of moving objects, features extraction, and fall event detection system.

Chapter 3 - implementation

In this chapter, we will present the video and audio analysis algorithms. We will explain the low-level features that we extract from both video and audio, and how we extract them. we will explain the classification method.

[1] http://www.hama.co.uk

Chapter 4 - Experimental Setup and results

We will present the most suitable combinations of low-level features to classify falls. We will also present the results of testing K-NN classifier and the results of final experiment.

Chapter 5 - Conclusion and future work

gives the conclusion of the project and point out areas of future work.

Chapter 6 - Legal and ethical considerations

Presents the legal and ethical considerations in the project

2 Related work

In the last few years the interest in video surveillance has increased a lot and lots of research work has been done smart video surveillance [5], and video surveillance systems are applied almost every where. Detection of moving objects in video streams is still one of the important research problems, many techniques have been developed to detect moving objects and get foreground image that has only the interesting objects to do further processing for several types of applications in the security and the safety systems. This chapter is a review of the state of the art related to the topics covered by this thesis.

2.1 Foreground segmentation

To detect moving objects in a scene for applications such as surveillance, it involves comparing an observed image (current frame) with an estimate of the image if it contained no moving objects (background model). The areas of the image plane where there is a significant difference between the observed and estimated images indicate the location of the objects of interest.

2.1.1 Background estimation

There are many techniques to estimate the background or background model [6], such as using the first frame of the sequence that doesn't have moving objects as an initial background estimate and update it when no moving objects are in the scene. But the problem with this simple technique that it can easily fail in some cases e.g. initialization with moving objects, quick illumination changes and relocation of background objects. Non-changing segments of the image are considered as being part of the background, whereas the foreground consists of the changing segments - including moving and new objects. another straightforward way of acquiring a reference image would be using the previous "history" by obtaining a background model based on the statistical representation of the previous N frames, for example a pixel-wise average image [7] or a pixel-wise median image [8]. After estimating the background model, the segmentation can easily be obtained from an efficient thresholded subtraction operation. The threshold could be proportional to the standard deviation. These simple approaches, although efficient, but may not perform well in real-world systems, non-controlled environments that have complex background. Median is computationally expensive because of the sorting operation. Changes in illumination conditions and dynamic behavior in the background may cause unacceptable rates of false positives and the system may consider the whole image as foreground particularly in quick illumination changes. Quick illumination changes can occur for instance when the lights are turned on or off, when sunlight comes through a window etc. In indoor video surveillance, we mostly have stationary background and the motion is mainly caused by interesting objects and its salient motion (motion from a typical surveillance target e.g. person) [9], unlike outdoor surveillance where motion may be caused by both interesting and uninteresting motion. To achieve robust background modelling, techniques that can adapt to dynamic behavior are needed. The performance should not be sensitive to lighting effects. It should also be capable of dealing with move-

ment through cluttered areas, objects overlapping in the visual field, shadows, lighting changes, effects of moving objects in the scene, slow moving objects and objects being introduced or removed from the scene [7]. In the case when the static objects move in the scene (changes in the background geometry), generally adaptive background subtraction techniques will detect false positive (considering the place was occupied by the static object that start to move as a moving object) for a short time which will affect the tracking process and make it difficult or impossible in some cases [10].
Damien et al. [11], in his project (Real-time People counting system using a single video camera) that he did at HIG last year; proposed a method of background estimation by combining adaptive background generation with three-frame differencing algorithm. The background estimation proposed method computes and uses luminance components to estimate background model, assuming that luminance component is less sensitive to sensor noise and changes of lighting conditions. Computation of the luminance components from RGB color space was done by the following equation:

$$Y_t(x,y) = 0.2989 * \mathbf{R_t}(x,y) + 0.5870 * \mathbf{G_t}(x,y) + 0.1140 * \mathbf{B_t}(x,y)\,. \tag{2.1}$$

Y_t is luminance value and the R_t, G_t, and B_t are color components. Then a binary motion mask M_t , $t > 1$ is defined by thresholding the two difference frames between each three consecutive frames. Motion estimation is computed as follows:

$$M_t(x,y) = \begin{cases} 1 & \text{if } |\mathbf{Y_t}(x,y) - \mathbf{Y_{t-1}}(x,y)| \geq \mu_{t-1} + \sigma_{t-1} \bigwedge |\mathbf{Y_t}(x,y) - \mathbf{Y_{t-2}}(x,y)| \geq \mu_{t-2} + \sigma_{t-2}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

where μ_{t-1}, μ_{t-2} and σ_{t-1}, σ_{t-2} represent the means and standard deviations of the pixel-wise absolute differences between the pairs of frames $(\mathbf{Y_t}, \mathbf{Y_{t-1}})$ and $(\mathbf{Y_t}, \mathbf{Y_{t-2}})$. M_t highlights only the different edges of moving objects. Next step is computing regions of interest (ROI) mask of the binary motion image. ROIs are created by finding the bounding boxes (the smallest rectangle which completely contains the region). The new background B_t is computed as follows:

$$B_t(x,y) = \begin{cases} \alpha \cdot \mathbf{B_{t-1}}(x,y) + (1-\alpha) \cdot \mathbf{I_t}(x,y) & \text{if } \mathbf{ROI_t}(x,y) = 0, \\ \mathbf{B_{t-1}}(x,y) & \text{otherwise.} \end{cases} \tag{2.3}$$

where $\alpha \in [0,1]$ is the learning rate and controls the background adaptation speed. The variable α determines the update sensitivity to the variations. An automatic way to estimate α is used, given by the following equation [12]:

$$\begin{aligned} \alpha &= \frac{\text{Number of all moving pixels}}{\text{Total Frame area in pixels}} && (2.4) \\ &= \frac{\sum \mathbf{ROI_t}}{Area(\mathbf{I_t})} && (2.5) \\ &= mean(\mathbf{ROI_t})\,. && (2.6) \end{aligned}$$

Figure 2 (c) shows the background model estimated using the updating method based on the ROI mask.

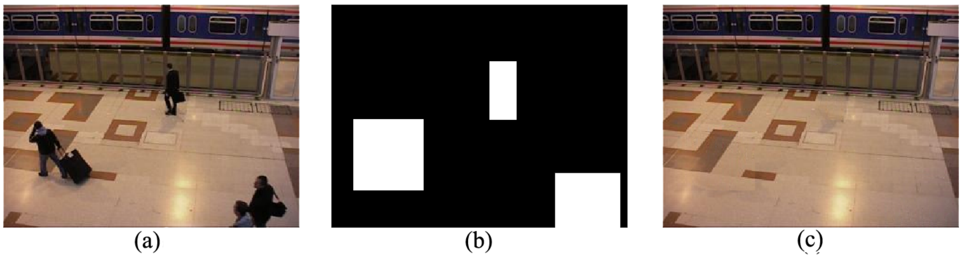


Figure 2: (a) current image; (b) ROI mask of the binary motion mask; (c) Background model updated with the ROI mask [11].

Figure 3 (d) shows the background model estimated using Damien's method [11] fail in the case that the object doesn't move for a while, like in our case when the person falls down, after number of frames, the background model mixed some part of the foreground objects with background estimate when the updating method is based on the ROI mask.

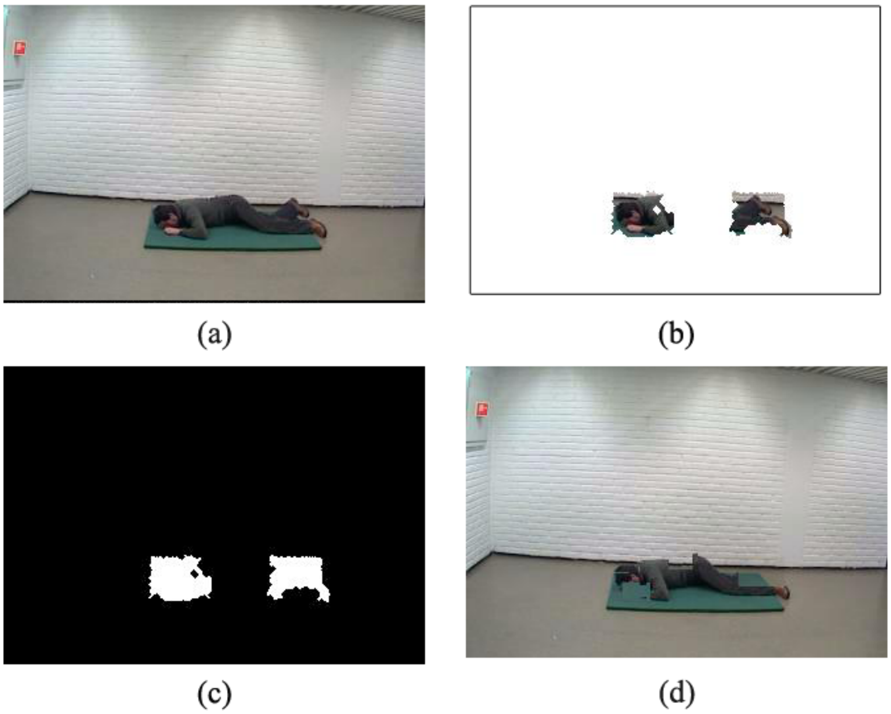


Figure 3: (a) current image (frame #84); (b) Foreground image; (c) Binary image,(b) Background model updated with the ROI mask.

It is also important to stress that background estimation operation is often required to perform as fast as possible, since it is usually the first step in the video analysis processing chain, and complex modelling methods are difficult to apply in real-time systems.

2.1.2 Detection of moving objects

In indoor video surveillance systems, typically stationary cameras are used to monitor the activities in the sites. In this case, the detection of moving regions/ objects can be achieved by comparing each new frame with a representation of the scene background (background model); this process is called background subtraction. Background subtraction forms the first stage in any automated visual surveillance system. The results from background subtraction process will be used for further processing, such as tracking the moving objects and understanding events. Probably due to its simplicity, the most common approach for discriminating a moving object from the background is background subtraction. Detection of moving regions depends on how good background estimation was. Most of the techniques used to detect moving objects, update the background model dramaticaly. The simplest technique is by computing the absolute difference between the pixels in current frame and the reference background [13]. in [13], a pixel is marked as foreground if

$$|\mathbf{I}_t - \mathbf{B}_t| > \tau \tag{2.7}$$

Where $\mathbf{I}_t$ is a current frame, $\mathbf{B}_t$ is a background model, and τ is a predefined thresholded after that closing holes and discarding of small objects is done, and the background reference is updated as

$$\mathbf{B}_{t+1} = \alpha\mathbf{I}_t + (\mathbf{1} - \alpha)\mathbf{B}_t \tag{2.8}$$

Where α must be small to prevent artifical tails forming behind moving objects. The background model in this technique needs a correction as in the case of appearance of static new objects, see Figure 4 (a). the background subtraction will leave hole. Or when sudden illumination changes, the simple correction is to update the background as $\mathbf{B}_{t+1} = \mathbf{I}_t$. when a pixel is detected as foreground pixel for more than m of last $\boldsymbol{M}$ frames [13].

Detection of moving objects by simple frame differencing of consecutive frames will not detect the entire objects as in Figure 4 (b). And if the object stops for a while it will not be detected [10].

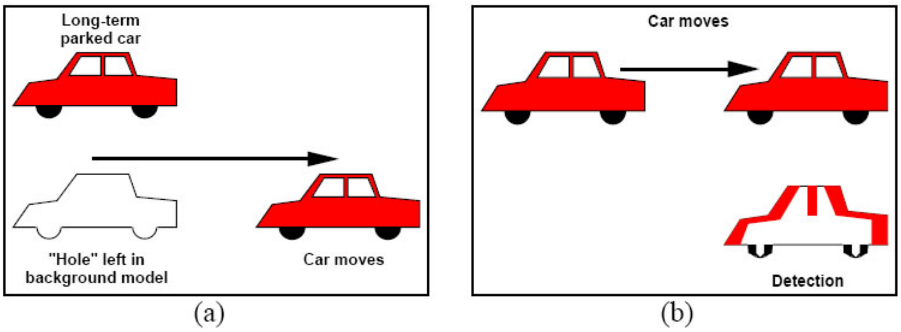


Figure 4: problems with standard MOD algorithms. (a) Background subtraction leaves holes when stationary objects move. (b) Frame differencing does not detect the entire object [10].

Pfinder [14] uses a simple scheme for real-time tracking of the human body, background pixels are modeled by a single value and updated by

$$\mathbf{B}_t = (\mathbf{1} - \alpha)\mathbf{B}_{t-1} + \alpha\mathbf{I}_t \tag{2.9}$$

and foreground pixels are modeled by mean and covariance which are updated recursively. This technique requires an empty scene to start.
In $\mathbf{W}^4$: Who? When? Where? What? A Real Time System for Detecting and Tracking People paper [15], a pixel belongs to the foreground if

$$|\mathbf{M} - \mathbf{I}_t| > \mathbf{D} \quad \text{or} \quad |\mathbf{N} - I_t| > \mathbf{D} \tag{2.10}$$

where the parameters **M**, **N** and **D** represent the minimum, maximum, and largest interframe absolute difference observable in the background scene for each pixel. These parameters are intially estimated from the first few seconds of video and updated periodically for those parts of the scene not containing foreground objects.

Javed et. al [16] proposed a three level algorithm using statistical method to deal with the problems of quick illumination changes, relocation of the background object and initialization with moving objects. The three different levels are divided into pixel level, region level and frame level. Gradients of image are less sensitive to illumination changes than only color based background systems and this system can deal with the global illumination changes at the third level (frame level).

Other techniques use K Gaussians distribution [17, 18] to detect moving objects. Ren et al. [17] proposed a spatial distribution of Gaussians (SDG) model to deal with moving object detection having motion compensation which is only approximately extracted. Stauffer et al. [18] proposed a method by modeling each pixel as a mixture of Gaussians and use on-line approximation to update the model, this system can deal with lighting changes, slow moving objects, and introducing or removing objects from the scene. Huwer et al. [19] proposed a method of combining a temporal difference method with an adaptive background model subtraction scheme to deal with lighting changes. The main problem with these techniques is that they cannot adapt to quick image variations such as a light turning on or off.

Motion based methods for detecting moving objects have also been proposed [20, 9]. Wildes [9] proposed a measure of motion saliency using spatiotemporal filtering. But his method didn't work for slow moving objects. Wixson [20] presented a method to detect motion by accumulating directionally-consistent flow. They use optical flow to compute the motion, but this method is time consuming and have trails left by objects. In [21], the authors used optical flow based on Lucas Kanade for motion estimation [22]. Tian et al. [23] proposed a method to detect moving objects by combining temporal difference imaging and a temporal filtered motion field. Their method assumes that the object moves in constant direction, so if it stops or moves in zigzag, the system will loose it, but this method can handle quick image variations; e.g., a light being turned on or off.

In most cases, the detection moving objects follow by some morphological operations to clean up noise and give better foreground.

2.1.3 Shadow removal

There are some situations where background modelling and differencing methods perform poorly. For instance when there are quick illumination changes, relocation of background objects, initialization with moving objects and shadows in the scene. Objects cast shadows that might also be classified as foreground due to the illumination change in the shadow region. Shadows change consequently the color properties in RGB color space; shadows make the color darker and this causes a big variation in the three RGB channels that leads to detection of shadows as foreground. Detected shadows as moving objects will cause problems in post-processing operations, it will increase the area and in some cases shadow is detected as new moving object, and this will make the tracking module fail, event classifier will fail as well. So, we need to remove shadows after the segmentation in order to ensure a reliable tracking process and event detection.

In Javed et al [16], the detection system is divided into three levels, pixels, region (gradients) and frame level. Gradients of image are less sensitive to illumination changes than color based detection systems, and this will decrease the effect of the shadow. In [24, 25], they used (YUV) color space in the detection algorithm. Sundaraj [24], defined shadows as regions in the image that differ in Y but (U,V) stays unchanged, and based on that he eliminate the shadow pixels from the foreground image.

Chen et al [26], they extract features in the RGB color space. Two feature variable: chromaticity and brightness distortion, are used to classify the foreground and background [27], the brightness distortion used in detecting shadows. Figure 5 (a) shows the effect of shadow on the foreground image, and (b) shows the foreground image after shadow suppression.

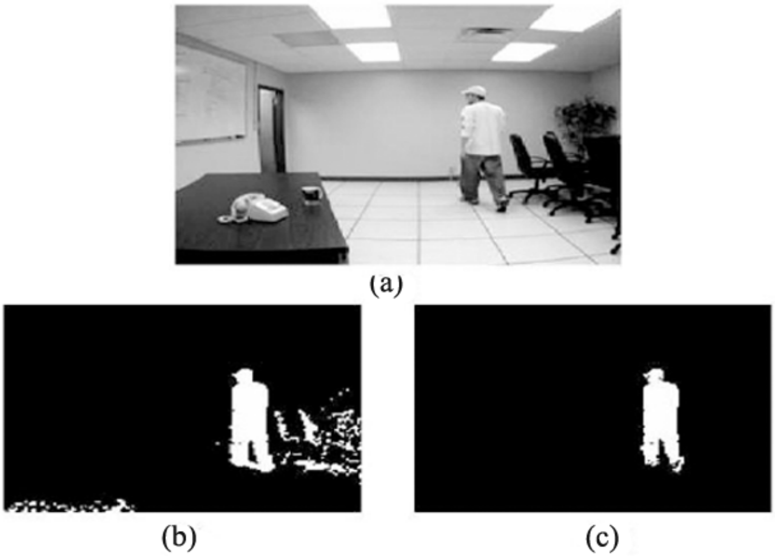


Figure 5: (a) original image, (b) silhouette without shadow suppression and (c) silhouette with shadow suppression [26].

Damien et el. [11] used the Hue - Saturation - Value (HSV) color space to explicitly separate chromaticity and luminosity. A shadow and non-shadow points differ principally in the luminance axis V.

2.2 Tracking of moving objects

Tracking of the moving objects in a video sequence is the process of finding the same object in different frames. To trace the objects, we need to use the information of the object trajectories, positions, sizes, color distribution, shape, speed, direction, ... etc. Variables based on the information are first computed (features extracted) from the foreground images, and then the tracking results are decided based on variable values. The objects are represented by their position coordinates (center of mass).
The main goals of the object tracking step are to [15]:

- Determine when a new object enters the scene and track that object, and add it to the list of objects to be tracked.
- Compute the correspondence between the foreground regions in current frame and the objects currently being tracked by the system.
- Employ tracking algorithms to analyze what the objects are doing and where are they in the scene?
- Improve the segmentation by connecting the blobs that belong to the same object (region merging) or split the blob that belong to many moving objects (region splitting).

There are different techniques used to detect the moving objects in video surveillance. Wang et al. [28] presented a object tracking rule-based algorithm using the information of the object trajectories, sizes, grayscale distribution, and texture. He assumes that the object acceleration rate is constant in a few adjacent frames. Bunyak et al. [29] presented a multi-hypothesis method for tracking of salient moving objects. He apply filtering and pruning at different levels of processing to eliminate spurious objects and trajectories from the tracker (Figure 6). This method utilizes the features extracted from the foreground, Kalman filter and color similarity to handle occlusions.

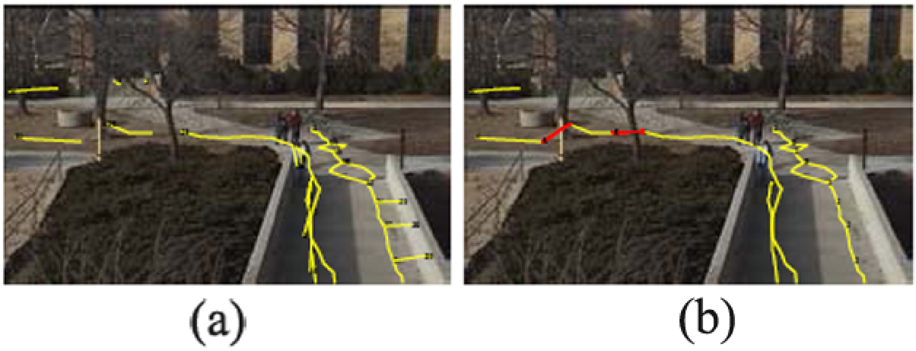


Figure 6: Tracking for walk-in sequence. (a) before pruning, and (b) after pruning and occlusion handling [29].

Wan et al. [30] used the Kalman filter to predict the motion parameters in the tracking module, then a tracking matrix is built to determine whether the objects occur occlusion or not. Other techniques using a maximum likelihood classification scheme [5] or using a continusly adaptive mean shift algorithm (CAMSHIFT) to track of moving objects [31].

The tracking stage is very important for any event detection system and failing in tracking will probably make the event classifier fail as well. The existing tracking of moving objects techniques are working for some cases but in other cases they don't give robust results.

2.3 Features extraction

After detection of moving objects and tracking them, we should extract some information about the moving objects which are important features for event detection and behavioral analysis. We extract the features from the foreground image after making some improvements on it to use them in the event classifier. The improvements are done by tracking algorithms, merging and splitting regions algorithms.

The foreground regions are labeled to distinguish between the different moving objects. Each moving object can be described by a set of features. Some of these features are depending only on the current foreground frame; such as centroids, heights, bounding boxes, areas, and color histograms, but other features that we can extract depend on a sequence of frames; such as direction of movements, speed, motion, ... etc.

We could classify the information of low level features that describe objects between instantaneous information of the spatial features of objects [32, 33, 34], such as width, height and temporal information about changes in the object's size and changes in motion features; such as direction of movement and speed.

The challenge is to find the relative features that can describe well the activities that take place in the scene and have different values for different activities to allow the event classification in the next stage. The classifier will take the features vector (scalar values) as an input and classify what the object doing in the scene depending on the features vector values. The output of the classifier should be meaningful and correlated with the human activities such as running, walking, falling down, sitting down, ... etc.

2.4 Fall event detection

In the last few years the interest in video surveillance increased a lot and lots of research has been done about smart video surveillance [5], and video surveillance systems are applied almost every where and mainly in the event detection and activity analysis field. Lots of research has been done to detect the falls using the wearable sensors [3], but the problem in this way is that the older people often forget to wear them. Infrared sensors have also been used in event detection and can also be used to detect falls [35]. Sixsmith and Johnson [35] developed intelligent monitoring system to detect falls based on a low-cost array of infrared detectors. They used in their project SIMBAD (Smart Inactivity Monitor using Array-Based Detectors) IRISYS (InfraRed Integrated Systems) thermal-imaging sensors, the sensor is wall mounted. The IRISYS sensor's low element-count infrared array technology can reliably locate and track a thermal target in the sensor's field of view, providing size, location, and velocity information.

They classify falls with a neural network using 2D vertical velocity of the person. But, 2D

vertical velocity could not be sufficient to discriminate a real falls from a person sitting down abruptly.
A simple method consists to analyze the bounding box representing the person in a single image [36]. In [36]; the authors proposed a method to automatically detect a fall by using audio and video. The video analysis consists of three steps:

1. moving region detection in video.
2. calculation of wavelet coefficients of a parameter related with aspect ration of the bounding box of the moving regions.
3. HMM (Hidden Markov Models) [37] based classification using wavelet domain data.

The audio track of the video is used to distinguish a person simply sitting on a floor from a person stumbling and falling. The analysis of audio is based on wavelet domain decomposition of the signals. The audio analysis algorithm consists of three steps:

1. computation of the wavelet signal.
2. feature extraction of the wavelet signal.
3. HMM based classification using wavelet domain features.

Actually a typical stumble and fall produces high amplitude sounds whereas the ordinary actions of bending or sitting down has no distinguishable sound from the background. so using the analysis of audio will make the system robust and decreases the false positives. This method can only be done if the camera is placed sideways, and can fail because of occluding objects. To avoid this problem, the camera can be placed higher in the room to not suffer of occluding objects and to have a larger field of view.
In this case, depending on the relative position of the person, the field of view of the camera, a bounding box will not be sufficient alone to discriminate a fall from a person sitting down. To overcome this problem, some researchers [38, 39] have mounted the camera on the ceiling. Lee and Mihailidis [38, 39] detect a fall by analyzing the shape and the 2D velocity of the person, and define inactivity zones like the bed. Nait- Charif and McKenna [38, 39] track the person using an ellipse, and analyze the resulting trajectory to detect inactivity outside the normal zones of inactivity like chairs or sofas. In thier proposed method for automatic summarization of human activity and detection of unusual inactivity in a supportive home environment, they combined activity zones with body's pose and motion information, this will provide a useful cue for fall detection. In addition, a human-readable description of activity in terms of semantic regions provides a useful summary of behavior.
In [40]; the authors proposed a method to detect and record various posture-based events of interest in a typical elderly monitoring application in a home surveillance scenario. The proposed method has four steps:

1. Obtaining the segmentation of moving objects by using adaptive background subtraction approach developed by Stauffar and Grimson [18]. They remove the adaptive characteristic to prevent the eventual inclusion of a static person as the background.
2. Feature extraction process for foreground object. They used horizontal and vertical projection histograms of sending posture with current foreground bounding box as feature set.

3. Posture classification using K-Nearest Neighbor (K-NN) algorithm and Evidence Accumulation technique.
4. Using the falling speed to infer real falling events.

In [41]; the authors proposed a vision-based system to detect unusual shape cues of people in the view of a monitoring camera, indicating possible behaviors of falling, fainting, slipping, or tripping. The system consists of two main parts:

a) a vision component to reliably detect and track each moving person as well as to extract his/her shape feature into observation sequence.

b) an event-inference module to parse the observation sequences in order to determine whether a fall down event is taking place.

In [42]; the authors proposed a method for recognizing falls from video sensors. They used the video sensors to collect information about daily activities of the elderly residents, then extracts important information to perform automated functional assessment and detect abnormal events, such as people falling on the floor. The privacy of residents is ensured through the extraction of silhouettes, a binary map that indicates the position only. They proposed a technique for extracting silhouettes based on statistically modeling a static background and then segmenting humans based on color information. Once the human is segmented and shadows removed, features need to be extracted from the silhouette in order to use hidden Markov models (HMM) for temporal pattern recognition. The activities recognized with HMM are falling, walking, and kneeling.

In [43]; the authors proposed a new method to detect a fall event, their method based on the motion history image and some changes in the shape of the person. They supposed that the motion is large when a fall occurs, and they used the motion history image to extract the motion. When a large motion is detected, they analyze the human shape of the person in the video sequence to check if the person on the ground. They used the background subtraction method to segment the person in the video sequence and they approximated the blob by an ellipse. The system has three steps:

1. Motion quantification.
2. Analysis of the human shape.
3. Lack of the motion after fall.

The system is developed to work in real-time, and it gives good results. But they used manual thresholds and the system deals with one moving object only.

Some methods developed to detect a fall using the vertical and horizontal 3D velocities of the head of the person extracted from a monocular camera video sequence [38]. 3D information is really helpful to analyze the actions of a person in a room. In [44], the authors proposed a detection system uses a MapCam (omni-camera), and they considered the information system in their algorithm.

In our proposed system to detect the falls in elderly's houses, the fall detection is based on a combination of many low level features that we extract from the foreground regions, the features are: motion history image (MHI), direction of movements, aspect ratio, height of the center of mass, orientation, major and minor axis, motion quantity and speed. The classifier takes these features and will identify if a real fall happened.

Our event detection is using K-NN classifier as part of the classifications procedure. The audio information from the microphone of the webcam or a separated microphone will be analyzed to decrease the false positives. The features that we extract from the audio track are variance and zero-crossing rate. The classifier combines the audio track features with other low level features that we extract from video sequence to detect falls.

3 Implementation

In this chapter we will focus on details of the implementation of the work in this thesis.

3.1 Overview of the proposed method

This section will give a general idea about the proposed devices we could use to setup our experiment, and we will summarize the algorithms that we used in the proposed method. The details of the algorithms will be discussed in the following sections.

3.1.1 Fall detection system overview

The proposed system has three main devices as can be seen in Figure 7:

- The network camera to monitor the room, the camera has built-in microphone also [1].
- The server that takes the video and audio from the camera as an input, analyze them and decides if a fall happened.
- The alarm device, the alarm device can be a normal alarm device invoked by the system if a real fall happens. It can be expanded to other devices that the controller can use, such as: PDA devices or Mobile devices.

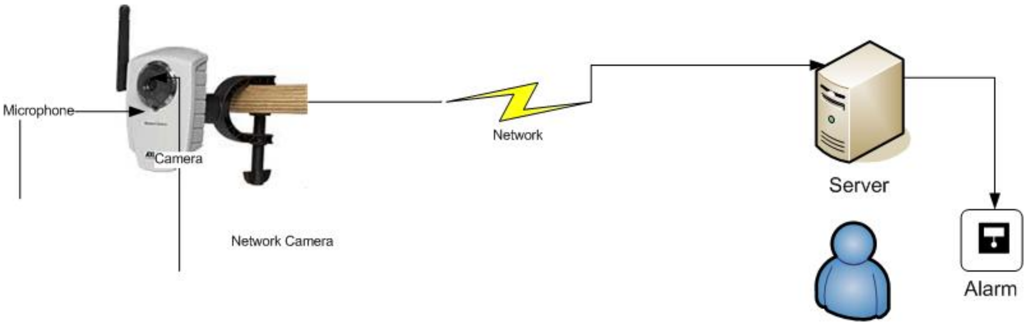


Figure 7: System devices

3.1.2 Proposed method overview

A schema for the whole system is illustrated in Figure 8. The proposed system has three main algorithms; the first algorithm is dealing with the images sequence analysis. It has five sub-algorithms. They are foreground segmentation, shadow removal, morphological operations, tracking algorithm and features extraction. The second algorithm is dealing with the audio track, the purpose of this algorithm is to extract features from audio track to classify a fall from other normal activities. The third algorism is the classifier. The classifier takes the outputs from the previous algorithms as an input, and give the output in two classes, a fall is taken place or other activities are taken place in the scene. Figure 8 shows all the algorithms that the system has, and the connections between them.

[1] http://www.axis.com/products/cam_207/index.htm last visited 17.05.2008

Our purpose is to implement a system has the capability to answer the project research questions. The algorithms are implemented in MATLAB, some of the MATLAB code was adopted from the previous project [11]. The parts of this system will be discussed in the next sections.

3.2 Video analysis

The aim of the video analysis is to extract some features from the image sequences that describe the moving objects in the scene. The features will be used in the activity analyses in the following stages to classify the events and detect fall events.

3.2.1 Segmentation

In the segmentation algorithm, we extract the foreground image that has only moving objects. The inputs to the segmentation are the current frame and the background image that has no moving objects as shown in Figure 9. The idea is any pixel can be part of the foreground if its value is different enough from its corresponding value in the background reference. The inputs are in the RGB color space, then an absolute difference between the current frame and the background reference is done for each color channel (R, G and B).

$$\mathbf{D}_t^c(x,y) = |\mathbf{I}_t^c(x,y) - \mathbf{B}_t^c(x,y)|, \forall c \in \{r,g,b\}. \tag{3.1}$$

$\mathbf{I}_t$is the current frame, **B** is the background and *c* is a color channel. Then a binary frame is extracted from the **D** by applying this equation:

$$Br_t(x,y) = \begin{cases} 1 & \text{if } (\mathbf{D}_t^r(x,y) > \tau^r) \vee (\mathbf{D}_t^g(x,y) > \tau^g) \vee (\mathbf{D}_t^b(x,y) > \tau^b) \\ 0 & \text{otherwise.} \end{cases}, \tag{3.2}$$

τ^r, τ^g and τ^b are the thresholds for each channel calculated from the $\mathbf{D}_t$ by determining the median and the median absolute deviation . $MED^c = med(D_t^c)$, $MAD^c = med\,|(D_t^c - MED^c)|$

Then we can find the threshold by applying the equation 3.3:

$$\tau^c = MED^c + 3 \cdot a \cdot MAD^c\,, \tag{3.3}$$

Where *a* is equal 1.4826 and it's the normalization factor for a Gaussian distributions.

By only using Background subtraction in the RGB color space; the binary image may have the shadow as moving objects. The shadows make the color darker and this make a big variation in the RGB values, so the background subtraction will probably consider a shadow as part of the moving objects. Therefore we have another step to remove the shadow and improve the segmentation process. We used the shadow removal algorithm that was proposed by Damien et al. [11]. The shadow removal process works in Hue-Saturation-Value (HSV) color space to separate the chromaticity and luminosity.

The Shadow removal process takes the current frame (RGB), background and the output from the first segmentation step (binary image) as shown in the Figure 9. The first step is to convert RGB to HSV color space, then find a shadow mask for each point only belonging to the moving objects in the foreground as [11]:

$$SM_t(x,y) = \begin{cases} 1 & \text{if } (\alpha \leq \frac{\mathbf{I}_t^v(x,y)}{\mathbf{B}_t^v(x,y)} \leq \beta) \wedge (|\mathbf{I}_t^s(x,y) - \mathbf{B}_t^s(x,y)| \leq \tau_s) \wedge (\mathbf{D}_t^h(x,y) \leq \tau_h\ ) \\ 0 & otherwise. \end{cases}, \tag{3.4}$$

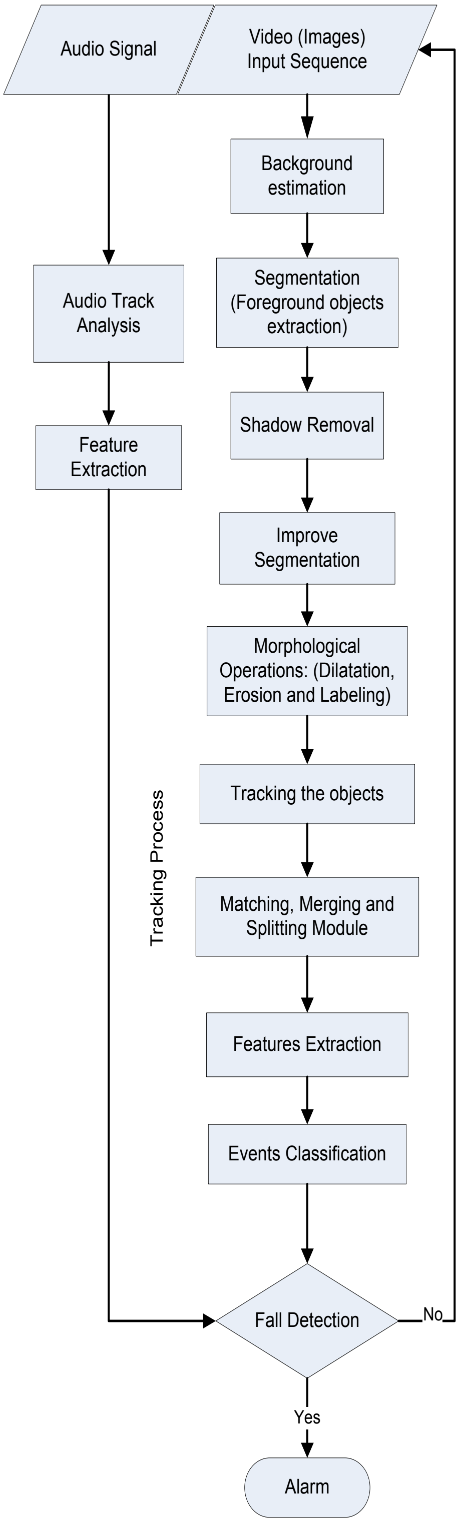


Figure 8: Proposed algorithm scheme

Where D_t^h represents the *angular* difference between the hue channel of the current image I_t^h and the background B_t^h and is defined as follows:

$$D_t^h(x,y) = \min\left(|\mathbf{I_t^h}(x,y) - \mathbf{B_t^h}(x,y)|, 360 - |\mathbf{I_t^h}(x,y) - \mathbf{B_t^h}(x,y)|\right) . \quad (3.5)$$

The thresholds ($\alpha, \beta \in [0,1]$) are depending on the light source intensity and the background darkness. These thresholds are necessary to evaluate the effect of shadow in the luminance channel. The shadow and non shadow points should have enough distance in the luminance dimension V.

To improve the segmentation, morphological operations are used to clean up the noise, fill holes and remove small components. We use dilatation to expand the foreground and erosion to expand the background, then we remove the small regions that are coming from noise and label the connected regions. Figure 9 shows the segmentation algorithm proposed.

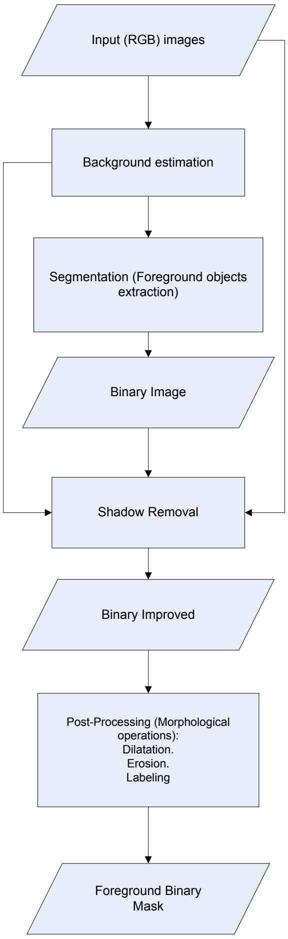


Figure 9: Segmentation scheme

3.2.2 Tracking

Tracking objects is the process of finding the same object in a sequence of frames. In this project; we used the same tracking algorithm that was implemented by Damien et al [11]. The tracking algorithm is using the blob's centroids and size features and it's based on the motion model proposed by Wan [30] that uses Kalman filter for prediction.

Motion model

The time between two consecutive frames is short in video sequence, so we assume that moving objects change slowly. The object parameters are modeled by discrete-time kinematic model. Kalman filters are employed to maintain and predict the state of the object. The kinetic model of an object is described as:

$$x_t = x_{t-1} + \Delta_t v_{t-1}, \tag{3.6}$$

Where Δ_t is the interval between two continuous frames. It is a linear system, therefore a Kalman filter can be used as follows:

$$X_t = A \cdot X_{t-1} + W_{t-1}, \tag{3.7}$$

$$Y_t = C \cdot X_t + V_t, \tag{3.8}$$

Where X_t and X_{t-1} are the state vectors at time k and $k-1$, W is the state noise, and assumed to have Gaussian distribution. Its mean is zero, the covariance matrix is Q which is set as: $Q = 0.01 * I$, where I is 8X8 unit matrix. V is the measurement noise which can be estimated directly from data, its mean is zero and covariance matrix is R. Y_t is the observation vector at time t, in our experiment they are set as:

$$X_t = \begin{bmatrix} x(t) \\ y(t) \\ a(t) \\ v_x(t) \\ v_y(t) \\ v_a(t) \end{bmatrix}, \qquad Y_t = \begin{bmatrix} x(t) \\ y(t) \\ a(t) \end{bmatrix},$$

Where the elements of X_t represent the coordinates of centroid, area and their corresponding change velocities. Then we can get the matrix A and C from previous equation:

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The Kalman filter has two distinct phases: Prediction and Updating [11]. The prediction phase uses the state estimate from the previous time-step to produce an estimate of the state at the current time-step. In the updating phase, measurement information at the current time-step is used to refine this prediction to arrive at a new more accurate state estimate for the next time-step.

The notation $\hat{X}_{n|m}$ represents the estimate of X at time n given observations up to, and icluding time m.

In our algorithm, we have the prediction of position and size for the object with an estimate of the error. This prediction is used to build the tracking matrix [30].

Building tracking matrix

For each frame in the frame sequance, a tracking matrix is built. This matrix links the predicted previous object's positions and area to the new ones. The rows of the tracking matrix represent all objects in the current frame, and columns all estimated objects from the previous frame. So, the tracking matrix is a $n \times m$ matrix noted M with n and m indicate, respectively, the number of objects in the current frame and in the previous frame.

Every elements y_{ij} of the tracking matrix is the Euclidean distance between the i-th measurement and the estimated position predicted from the j-th previous object.

$$y_{ij} = \sqrt{(Y_t^i - \hat{Y}^j_{t|t-1})^T \cdot (Y_t^i - \hat{Y}^j_{t|t-1})}\,, \tag{3.9}$$

Where Y_t^i is the i-th measurement in frame t and $\hat{Y}^j_{t|t-1}$ is the j-th estimated object in frame t from frame $t-1$.

$$Y_t^i = (x_i(t), y_i(t), A_i(t))\,. \tag{3.10}$$

The tracking matrix is then passed to the matching, merging and splitting (MMS) module.

Matching, merging and splitting module

The aim of MMS module is to mach the objects in the current frame with the same objects in the previous frames and solve the problems of merging and splitting the objects. The first step consists scanning the tracking matrix along rows and built an another matrix (called flag matrix). If there is just one non-zero element in the i-th row, then a `splitting or matching` flag is stored. Whereas, if there are more than one non-zero elements in the i-th row, a `merging` flag is stored. Note that if there are only zero elements in the i-th row, then the i-th object is considered to be a new one. The second step consists of scanning this new flag matrix along the columns. If there is one `splitting or matching` flag in the j-th column (at the i-th row), then we are sure that it is exactly a `matching` flag (between the j-th previous object and the i-th current object). Note that if there are only zero elements in the j-th column, then the j-th previous object has disappeared from the scene. Otherwise, the flag is totally ambiguous (`splitting, merging or matching`) and thus needs more analysis. The third step is to find the best distance (Euclidean distance) of all possible combinations between objects; minimize the error of decision to resolve the ambiguous cases. As a result of the MMS module, labels are attributed to every current objects considering their connection with the previous ones.

3.2.3 Features extraction

At this stage; when we have a binary image that has only the moving objects. We want to extract some features to have good description of the moving objects position. In indoor video surveillance we could classify the main activities in the scene to (walking, standing, sitting, squatting, running and lying down). First we study each of these activities and we described them in a simple way to find the differences between them to extract some features that could classify the fall event efficiently. we summarize these activities as:

- Walking: walking is a human behavior or activity that he/she use to move from one place to another. "Walking is a physical activity which enables humans to get from place a to place b". The human use his/her legs to move by using regular legs movements. So walking is one of human's activities that has some distinguishable properities from other human activities as for example; standing, sitting down or running. In the walking activity; the person moves his/her legs and may be his arms also in a regular way. So in walking activity, there is some motion, may be regular motion and not so fast motion as in running activity and the height of the person is almost steady.
- Standing up: Standing is a human behavior. In the standing situation; the person will have almost no movements, his/her legs must be in stable situation without any movements (the body will have no motion).
- Sitting down: in the sitting case; we should distinguish between two activities. The first activity when the person is going to sit down, the second one when the person will be in the sitting down position. In the process to sit down, the person usually moves his body from standing up or walking situations to sitting down situation. The person is moving his/her legs in a way to have a sitting down situation (put the legs in 90 degree angle at the knees joint). In the sitting down situation, the person sits on the seat with no motions, the person will be in sitting down posture.
- Squatting: in the squatting situation, the person will sit in a crouching position with knees bent and the buttocks on or near the heels, and his/her back will be sloping. There is no motion in the squatting situation, but the person can not be in the squatting situation for a long time.
- Lying down: lying down is the human behavior where the human body will be flat and lying on the floor. The head, the body and the legs must touch the floor. The lying down situation may happen after a fall down event or may be a normal human activity to rest or to sleep. The fall down can happen for the person if he/she lost his/her balance or if he/she face some obstacles in the way or for other reasons. When the person is in the lying down situation, he/she will have no motion.

After some analysis of the features that can be used to describe the fall event and classify it from other activities, we extract a group of features that could be divided in two groups. Some of the features depend on the still images; such as: aspect ratio, height of the center of mass, height of the bounding box, orientation, major axis and minor axis. And the others depend on the sequence of frames such as: motion history image (MHI), direction of the motion, motion quantity and speed. In rest of this section, we will explain the extracted features and how we build the features vector to be used in the event classification in the next step. Figure 10 summarizes the features that our algorithm extract.

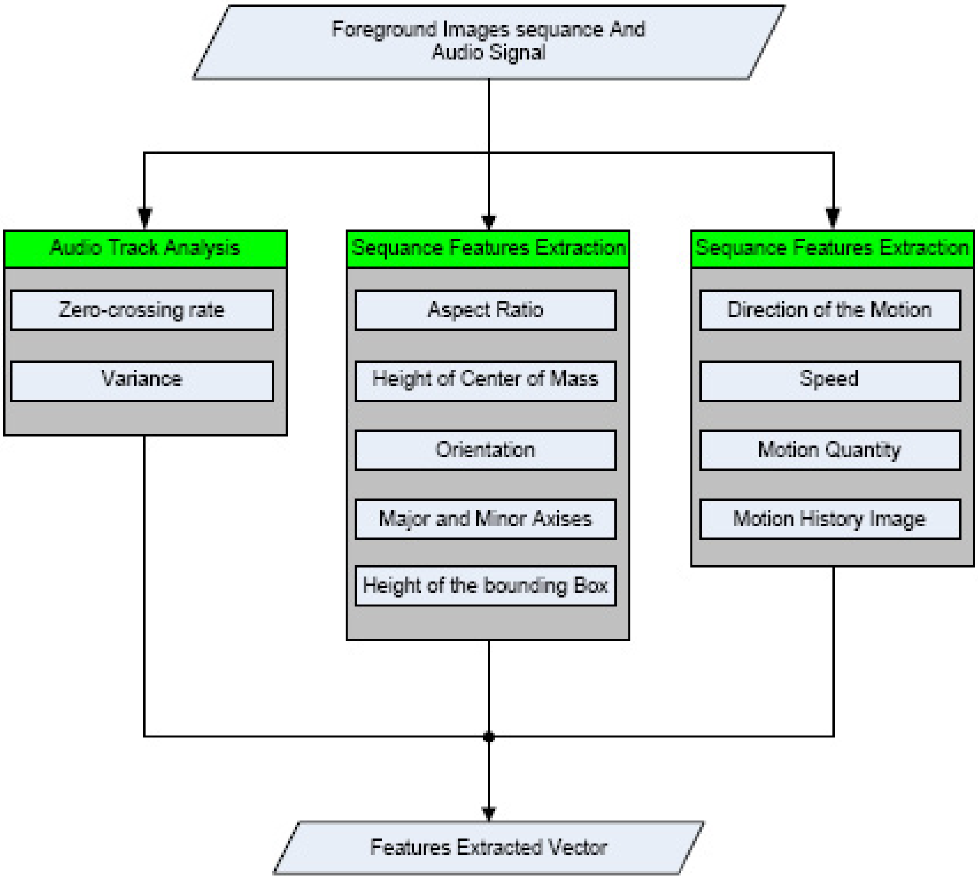


Figure 10: Features extraction algorithm

The descriptions of the features in Figure 10, and how they are implemented and extracted is presented in the following.

Aspect ratio

We compute the aspect ratio by finding the bounding box "the smallest rectangle containing the blob (moving object)". Then the aspect ratio of the moving object is defined as:

$$\mathrm{AspectRatio}(n) = \frac{H(n)}{W(n)} \tag{3.11}$$

Where, $H(n)$ and $W(n)$ are the height and the width of the minimum bounding box of the object at frame n.

But to determine the bounding rectangle by just finding the furthest foreground pixels will maybe give an error because of noise, including detection errors and shadows. These errors could make the size and shape of the bounding box change drastically. Thus we compute the aspect ratio by estimating the foreground object by projecting the fore-

ground pixels onto the x and y axes, that is, calculating the number of changed pixels row wise and column wise.
For X-projection: $\Psi(x)$ is the vertical projection against x-axis, $\tilde{x} = arc(max\Psi(x))$. The right and left borders of the minimum bounding rectangle are determined as:

$$w_r = max\{x : x > \tilde{x} \quad and \quad \Psi(x) > T\}. \tag{3.12}$$

$$w_l = min\{x : x < \tilde{x} \quad and \quad \Psi(x) < T\}. \tag{3.13}$$

where T is a chosen threshold proportional to the maximum number of pixels in the columns of the foreground region. The top and bottom of the minimum bounding rectangle h_t and h_b can be obtained likewise using horizontal projection.

The aspect ratio can thus be calculated as:

$$AspectRatio = \frac{h_t - h_b}{w_r - w_l} \tag{3.14}$$

Motion quantity

In the beginning; we evaluate the motion using three consecutive luminance images. Then we find the motion quantity by computing the number of changing pixels.

Speed

Speed is the rate of motion, or equivalently the rate of change in position, often expressed as distance d traveled per unit of time t.
Distance is a numerical description of how far apart objects are at any given moment in time. We can find the distance between two points of the xy-plane using the distance formula. The distance between $(x1, y1)$ and $(x2, y2)$ is given by:

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2} = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \tag{3.15}$$

Where $(x1, y1)$ and $(x2, y2)$ are the centers of mass of the blob in a sequence of frames.
Speed is a scalar quantity with dimensions pixels /second and its only computed for the center of mass for each blob.

Height of the center of mass

The idea is to calculate the distance (height) between the center of mass of a person and the floor. we calculate this value by finding the center of mass and the bottom edge of the miminum bounding box, then calculate the vertical distance between the center of mass and that edge.

Motion history image (MHI)

The purpose from extracting this feature is to detect when a large motion of a person happens. This is based on the fact that when a fall occurs; the motion will be large. The motion history image is an image where the pixel intensity represents the recency of motion in an image sequence. Therefore is gives the most recent movements of a person during an action [45]. The MHI is computed by equation 3.16:

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if D(x,y,t) } =1, \\ max(0, H_\tau(x, y, t-1)) & \text{otherwise.} \end{cases} \tag{3.16}$$

where H_τ is the MHI, τ is a fixed duration and taking values between 1 and a maximum number of frames in a sequance of frames.
The results are scalor values, we quantify the motion of the person based on the MHI values by this equation [43]:

$$CM = \frac{\sum_{Pixels(x,y)\in blob} H_\tau(x,y,t)}{Numberofpixels \in blob} \tag{3.17}$$

Orientation

The orientation gives the overall direction of the shape. It is the angle in degrees ranging from (-90 to 90 degrees) between the x-axis and the major axis of the ellipse that has the same second-moments as the region.

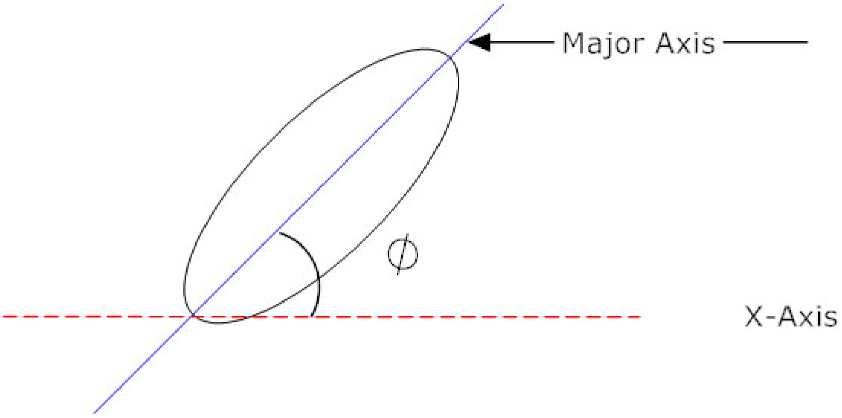


Figure 11: θ : Orientation: it is the angle between the major axis of the ellipse fitting the object and the X-axis.

Moments are another way to describe the shape by using its statistical properties. The statistical moments are: mean value μ, Variance σ^2 , and a statistical property called *skew* to describe how symmetric the function is.

For discrete one-dimensional function , we can find the moments about some arbitrary point, usually about zero or about the mean. The *n*-th moment about zero denoted as m_n.

$$m_n = \frac{\sum_{x=1}^{N} x^n f(x)}{\sum_{x=1}^{N} f(x)} \tag{3.18}$$

The zeroeth moment, m_0 is equal 1, the mean μ is the first moment about zero.

$$\mu = m_1 \tag{3.19}$$

The *n*-th moment about mean denoted as μ_n and called the *n*-th central moment. We can compute μ_n as follows.

$$\mu_n = \frac{\sum_{x=1}^{N}(x-\mu)^n f(x)}{\sum_{x=1}^{N} f(x)} \tag{3.20}$$

The zeroeth central moment μ_0 is 1. The first central moment μ_1 is 0, The second central moment μ_2 is the variance.

$$\sigma^2 = \mu_2 \tag{3.21}$$

The third central moment μ_3 is the *skew*:

$$skew = \mu_3 \tag{3.22}$$

For discrete two-dimensional functions, the *ij*th moment about zero can be computed by equation 3.23.

$$m_{ij} = \frac{\sum_{x=1}^{N}\sum_{y=1}^{N}(x)^i(y)^j f(x,y)}{\sum_{x=1}^{N}\sum_{y=1}^{N} f(x)} \tag{3.23}$$

The $m_{00} = 1$, $m_{10} =$ the *x* component of μ_x of the mean; and $m_{01} =$ the *y* component of μ_y of the mean.

The central moments defined as:

$$\mu_{ij} = \frac{\sum_{x=1}^{N}\sum_{y=1}^{N}(x-\mu_x)^i(y-\mu_y)^j f(x,y)}{\sum_{x=1}^{N}\sum_{y=1}^{N} f(x)} \tag{3.24}$$

from the equation above we can see, $\mu_{10} = \mu_{01} = 0$.
we can use these moments to provide useful descriptors of shape. In the binary image, the pixels outside the shape have value 0 and the pixels inside the shape have value 1. The moments μ_{20} and μ_{02} are thus the variance of *x* and *y* respectively. The moment μ_{11} is the covaiance between *x* and *y*.
We can use the covariance to determine the orientation of the shape. The angle between the major axis of the shape (person) and the horizontal axis *x* gives the orientation of the ellipse that has the same second-moments as the region, can be computed with the central moments of second order:

$$\theta = \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}\right) \tag{3.25}$$

Major and minor axis

Major axis definition: The major axis of an ellipse is its longest diameter, a line that runs through the centre and both foci, its ends being at the widest points of the shape.

Major axis definition: The minor axis of an ellipse is its shortest diameter, and its runs through the centre as shown in the figure down.

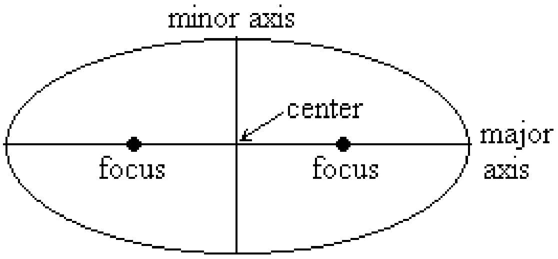


Figure 12: ellipse, major axis and minor axis

We can calculate the major axis length and the minor axis length of the ellipse that has the same second central moments as the region by the following steps. find the covariance matrix C:

$$C = \begin{pmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{pmatrix} \tag{3.26}$$

then find the eigenvalues *Imin* and *Imax* of the covariance matrix [46],

$$Imin = \frac{\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{2} \tag{3.27}$$

$$Imax = \frac{\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{2} \tag{3.28}$$

then the major axis is [47]:

$$major\ axis\ length = 2 \times (4/\pi)^{1/4} \left[\frac{(Imax)^3}{Imin}\right]^{1/8} \tag{3.29}$$

and the minor axis is:

$$minor\ axis\ length = 2 \times (4/\pi)^{1/4} \left[\frac{(Imin)^3}{Imax}\right]^{1/8} \tag{3.30}$$

Direction of the motion

The blobs belong to the objects represented by their centers of mass, so the direction of motion for each blob is measured by the direction of their centers of mass. The direction is the information contained in the relative position of current center of mass with respect to the center of mass in the previous frame.

Height of the bounding box

This feature measures the height of bounding box "the smallest rectangle containing the blob (moving object)".

3.3 Audio analysis

The information from the audio track can also be used to analysis the human behavior. Usually a typical stumble and fall produces high amplitude sounds, where as the ordinary actions of bending or sitting down has no distinguishable sound from normal (background). There are different methods to analysis the audio signal to extract the features and use them in the fall detection approach such as analyze the audio signal based on wavelet domain feature extraction or analyze it based on Fourier domain feature extraction. Wavelet domain feature extraction produces more robust results than Fourier domain [48], in the wavelet domain feature extraction, the wavelet coefficients of a fall sound are different from normal, bending or sitting down sound.

3.3.1 Wavelet signal

The audio analysis is based on wavelet domain signal decomposition. A wavelet transform of a signal provides better result than the time domain signal because wavelets capture sudden changes in the signal and separates them from stationary parts of the signal. The wavelet transform of the audio signal is calculated by passing it through a series of filters. The samples are passed through a low pass filter, and the signal is also decomposed simultaneously using a high-pass filter. The outputs giving the detail coefficients (from the high-pass filter) and approximation coefficients (from the low-pass filter) as seen in figure 13 bellow. The outputs are then subsample by 2. We use the detail coefficients that are calculated by using high-pass filter followed by decimation for the further analysis on the wavelet coefficients corresponding to the audio signal.

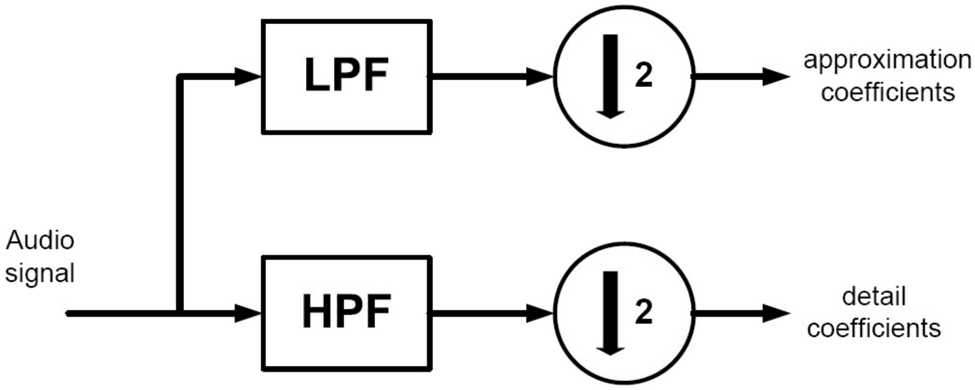


Figure 13: Wavelet Transform,Wavelet Coefficients corresponding to the audio signal followed by Decimation.

The advantage of using wavelet coefficients (detail coefficients) is that the wavelet signals can easily reveal the aperiodic characteristics which is intrinsic to the falling down case. The slow variations in the original audio signal lead to zero mean wavelet coefficients because wavelet coefficients are high-pass filtered signal. The second advantage from using wavelet coefficients is that it's easier to set thresholds in the wavelet domain since they are biger changes.

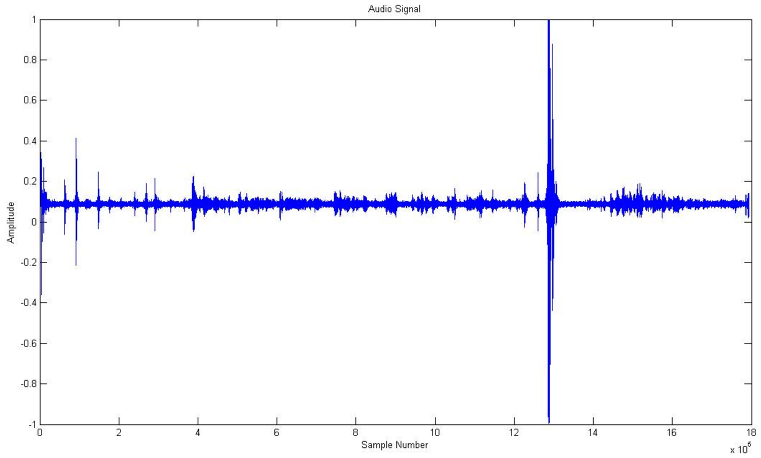


Figure 14: Audio signal corresponding to a fall which takes place at around number 13 × 10^5, different actions take place before and after a fall as talking and walking, talking and sitting down.

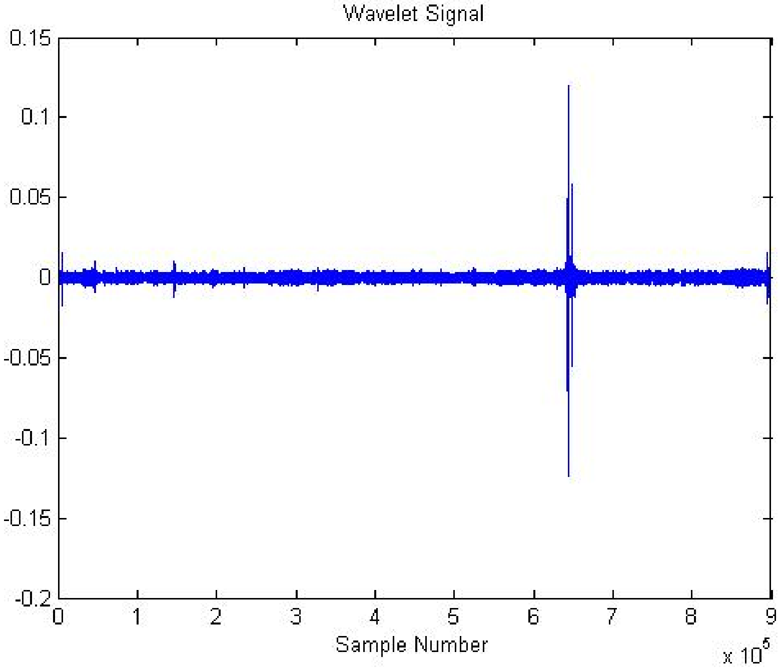


Figure 15: The wavelet signal corresponding to the audio signal in Figure 14.

3.3.2 Analysis of wavelet signals

After extracting the audio signal power and transform it to wavelet domain we extract two features to discriminate a fall event from other events and background. We are interested only in a fall event classification that usually has a big difference from other normal activities (speech, music and silence). We studies different features (signal power, zero-crossing rate, signal to noise ratio, and variance) and we found that zero-crossing rate (ZCR) and variance features gives an obvious difference between a fall and other events.

Zero-crossing rate (ZCR)

Zero-crossing rate is a much used in a wide range of audio applications such as speech recognition and audio classification [49]. The definition of zero-crossing rate is the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative or back. The zero-crossing rate feature is extracted from the wavelet signal corresponding to the audio track data in fixed length short-time windows. In our implementation, the sampling frequency is 44.1 KHz, and we take 500 sample windows to find zero-crossing rate in each sample window. The zero-crossing rate is defined in each sample window as:

$$ZCR(i) = \frac{1}{N}\sum_{n=0}^{N-1} |sgn\,[s(n)] - sgn\,[s(n-1)]| \tag{3.31}$$

Where $s(n)$ is the wavelet signal, N is the sample window length, and i is the sample window number.

ZCR tells how often the dominant frequency in the signal across the zero level. A high dominant frequency will give more crossing than a low frequency. As we mentioned in the beginning of this section, a typical stumble and fall produces high amplitude sounds as shown in figure 14. When a person stumbles and falls, ZCR values decreases as shown in figure 16(b).

Talking is constructed of voiced parts and pauses in between. A pitch can be found in the voice periods. Voiced parts will therefore give small ZCR values and unvoiced parts or background (music) give larger values.

Variance

The variance measures of how spread out a distribution is. In other words, it will measure the variability. The variance is computed as the average squared deviation of each value from its mean. When a person stumbles and falls, variance will increase rapidly. We determine the variance in each sample window as:

$$VAR(i) = \frac{1}{N}\sum_{n=0}^{N-1} (s(n) - \mu)\,.^2 \tag{3.32}$$

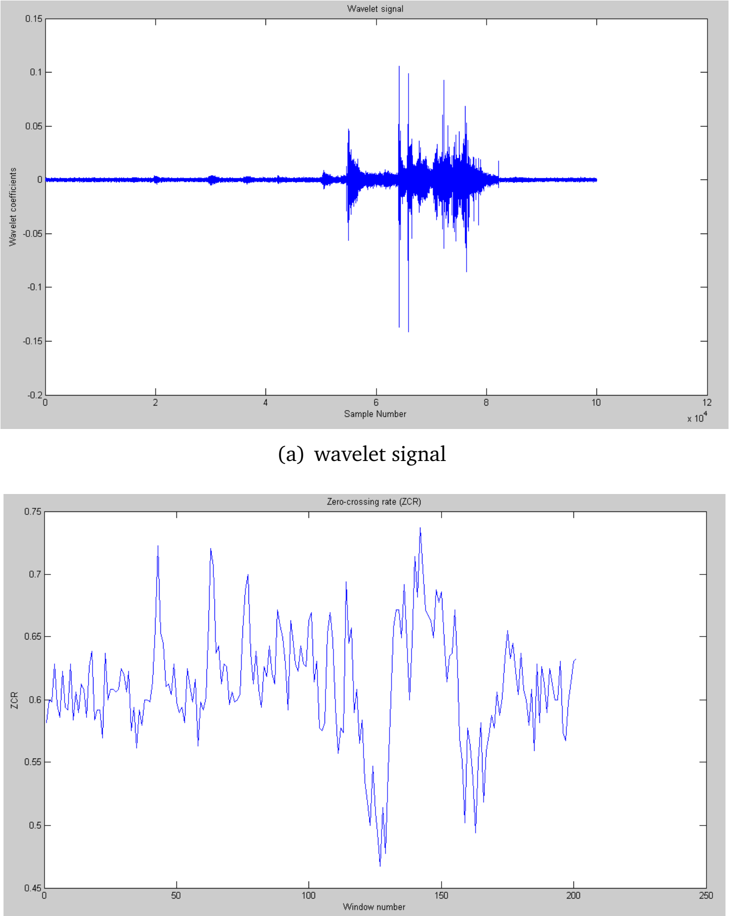
Where $s(n)$ is the wavelet signal, N is the sample window length, μ is the mean of the sample window values, and i is the sample window number. Figure 17(b) shows the variety of variance signal between falling and other activities.

As we said, variance values increases whereas ZCR values decreases when a person falls down. To utilize these two features in an optimized way to discriminate falls from other activities taken place in the scene; we define a feature parameter VZ in each window as follow:

$$VZ(i) = \frac{VAR(i)}{ZCR(i)} \tag{3.33}$$

Where i is the sample window number.

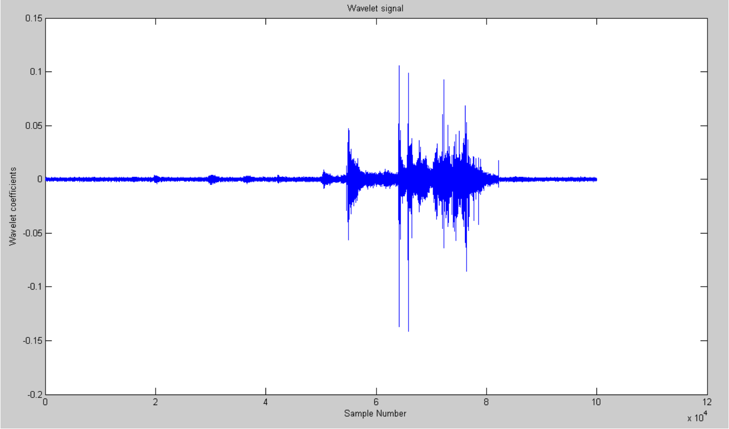
We will explain how we determine the threshold for VR to discriminate the falls, and



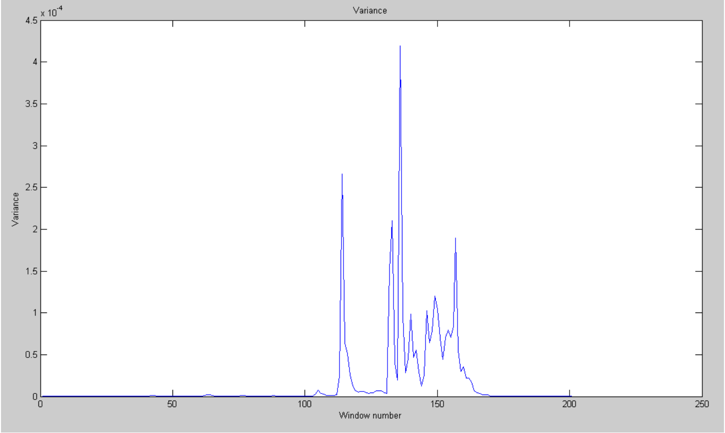
(a) wavelet signal

(b) zero-crossing rate

Figure 16: Zero-crossing rate of a falling and talking signal (fall gives less ZCR values).



(a) wavelet signal



(b) variance

Figure 17: Variance of a falling and talking signal (fall gives larger variance values).

how to utilize this audio feature in the fall event classifier that based on video and audio features in details in the next chapter.

3.4 Events calssification

The aim of event classification algorithm is to detect the real falls and distinguish between fall and other normal activities in the scene. As shown in Figure 18, the classifier takes the features vector from the video analysis and VZ parameter from audio track analysis as inputs and give alarm when a fall happens as an output. The classifier divides the features vector in sub-vectors and then combines the results from all of them to give the final decision. The proposed classifier has five decisions. The first decision will check if a large motion happens (MHI) and if the direction of motion is to down direction (the person position is in the transformation stage between standing and lying down). The second decision will check if a person in lying down position. The system check this by using K-NN algorithm, the inputs of K-NN are aspect ratio, height of the center of mass, height of bounding box, orientation, major axis and minor axis. The third decision is coming from the audio track analysis; the classifier checks if the VZ parameter passes the threshold. The fourth decision will combine the previous decisions and decide if its a possible fall. If the fourth decision gives that a possible fall happens, we do the check to find if the person stay on the ground without movement for a defined period. The fifth decision takes its inputs from K-NN algorithm and combination from motion quantity and speed features. The fifth decision gives the final result. We assume that after a fall, the person will be on the ground (lying position) with a little motion, and we used this assumption to robust the system and decrease the false positives.

Figure 18, shows our classifier steps and the relation between the decisions.

3.4.1 K-Nearest Neighbor classifier (K-NN)

K-nearest neighbor is a supervised learning algorithm where the result of new instance query is classified based on majority of K-nearest neighbor category. The purpose of K-NN algorithm is to classify a new entry based on training samples that already exists in the memory. Given a new query, we find K nearest neighbors (training points) closest to this query point. The classification is using majority vote among the classification of K points. K nearest neighbor algorithm is simple. It works based on minimum distance from the query instance to the training samples to determine the K nearest neighbors. After we find K-nearest neighbors; we take simple majority of these K nearest neighbors to the prediction of the query instance as illustrated in Fig. 19. Euclidean distance is commonly used as the metric to measure neighbor-hood. For the case of K=1; we will obtain the nearest neighbor classifier which is simple assign the input feature vector to the same class as that of the nearest training vector. The Euclidean distance between feature vectors $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$ is given by the following equation:

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.34}$$

The K-NN algorithm is very simple and effective, but we should take in our consideration that the Euclidean distance gives undesirable outcome in some cases where several feature sets with very big differences in values are used as a combined input to a K-NN classifier, the K-NN classifier will be biased by the larger values and this leads to poor per-

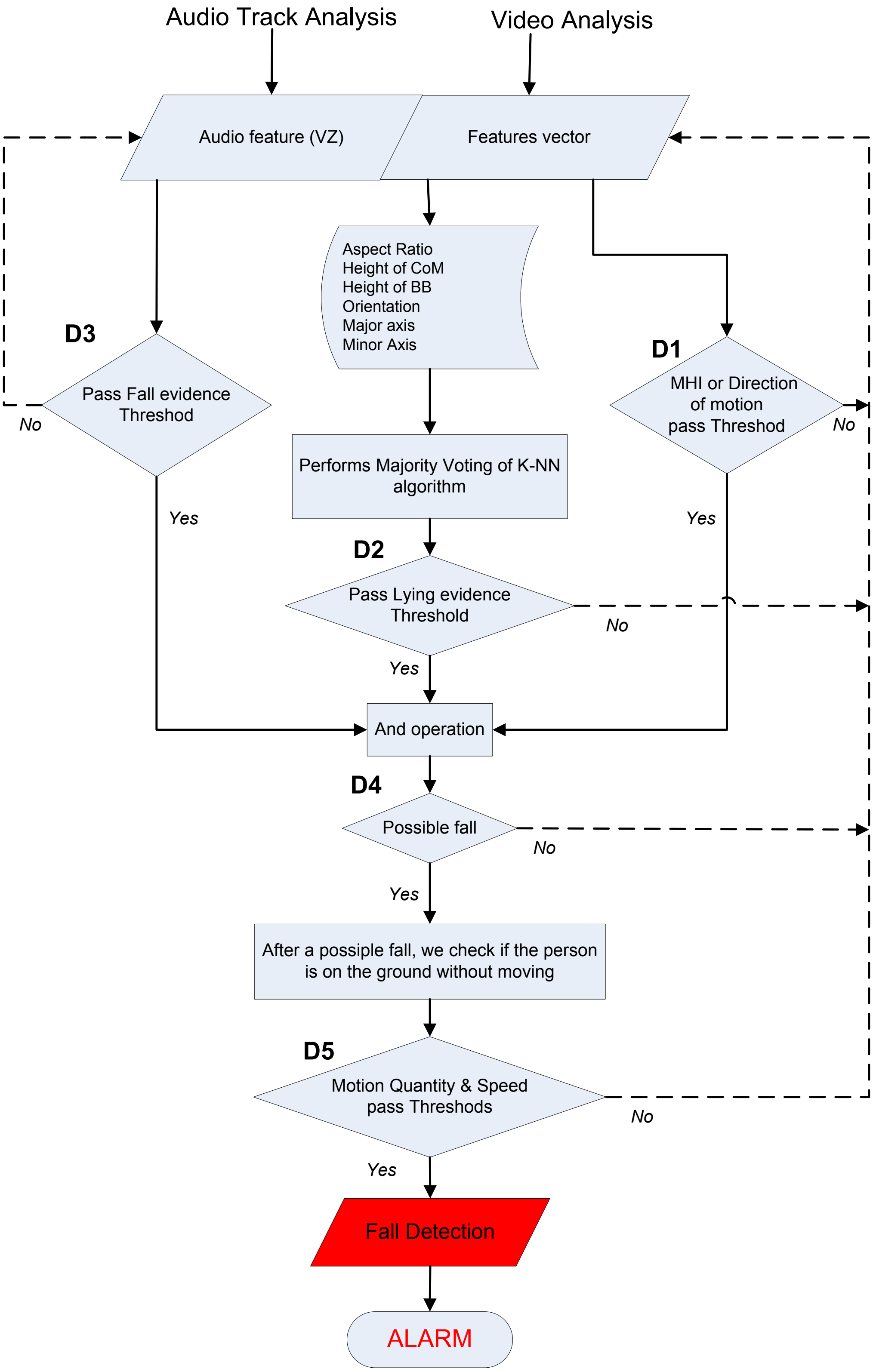

Audio Track Analysis
Video Analysis
Audio feature (VZ)
Features vector
Aspect Ratio
Height of CoM
Height of BB
Orientation
Major axis
Minor Axis
D3
Pass Fall evidence Threshod
No
Yes
D1
MHI or Direction of motion pass Threshod
No
Yes
Performs Majority Voting of K-NN algorithm
D2
Pass Lying evidence Threshold
No
Yes
And operation
D4
Possible fall
No
Yes
After a possiple fall, we check if the person is on the ground without moving
D5
Motion Quantity & Speed pass Threshods
No
Yes
Fall Detection
ALARM

Figure 18: Event classification algorithm

formance. We avoid this problem by normalizing the feature sets. Figure 19 illustrates how the K-NN work.

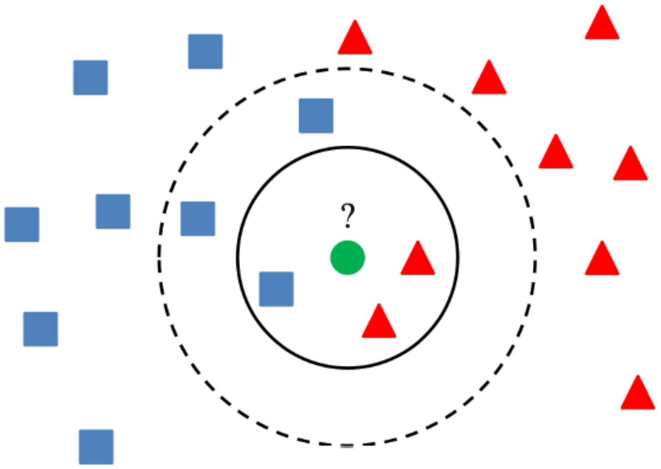


Figure 19: K-nearest neighbor algorithm illustration. The green circle is the sample which is to be classified, blue squares and red triangles illustrates samples from two different classes in the training set. If K is 3, then the 3 nearest neighbors (illustrated by the black circle with the solid line) will decide which class the analyzed sample will be assigned. If K is 5, the 5 nearest neighbors will be considered (illustrated by the black circle with the dotted line).

4 Experimental setup and results

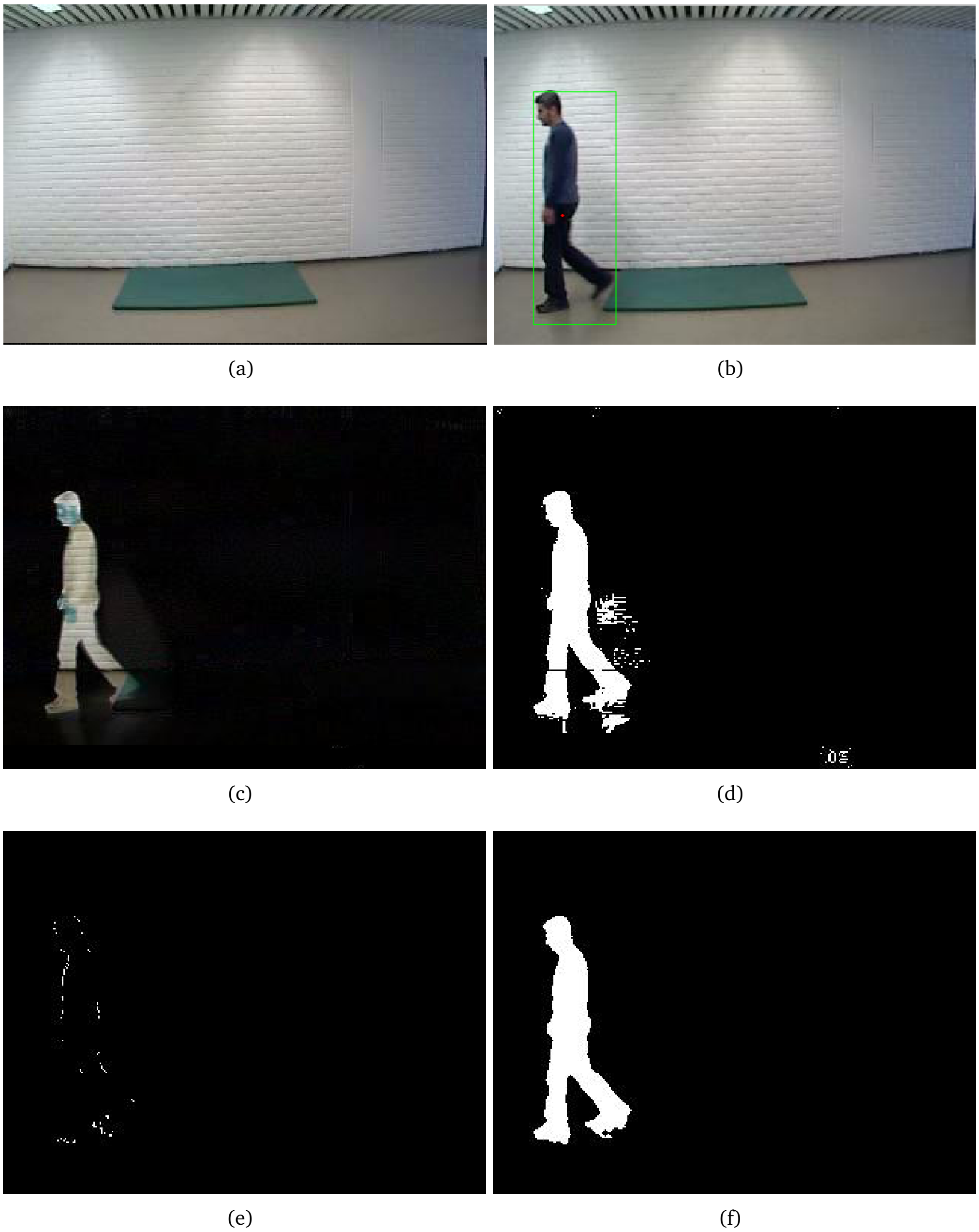
In this chapter; we will discuss testing the algorithms. The algorithms are implemented in MATLAB, and the testing data was taken in the school. Each algorithm is tested individually first and then we combine all the algorithms together and test whole the proposed system.

4.1 Video analysis

In this section we will discuss the experimental setup that used to test the video analysis algorithm parts and the testing results. First we will discuss the testing of the segmentation algorithm, and then we will discuss the testing of all the low level features that we have extracted from the foreground images and how we utilize each feature to discriminate a fall from other activities.

4.1.1 Segmentation

Segmentation is the first step in the fall event detection. The segmentation algorithm is explained in the previous chapter and is implemented in MATLAB. Firstly; we test the segmentation algorithm on two test movies that we performed in A220 room and A-building first floor corridor. We use shadow removal algorithm and morphological operations to improve the segmentation output. Figure 20 shows the segmentation process results for the test movie for a person walking in the corridor. In (c); we see the absolute frame difference between the current frame [frame # 101] (b) and the background reference frame(a), (d) shows the binary mask stbefore the shadow removal and morphological operations, (e) is the shadow mask, and (f) is the improved mask binary.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 20: Segmentation algorithm testing (a) Background reference; (b) current image; (c) Background subtraction; (d) Binary Image; (e) shadow mask; (f) Binary Improved.

4.1.2 Features extraction

For the purpose of fall detection and classifying human activities in the scene; we need to extract some information (low-level features) from the foreground binary mask that has only the moving objects. In the video sequence we extract two types of features; the first type is obtained by analyzing the shape (moving object mask or blob) of the moving object and this is done by analyzing the current frame only. The low-level features extracted are: aspect ratio, orientation, height of center of mass, height of bounding box, major axis and minor axis.

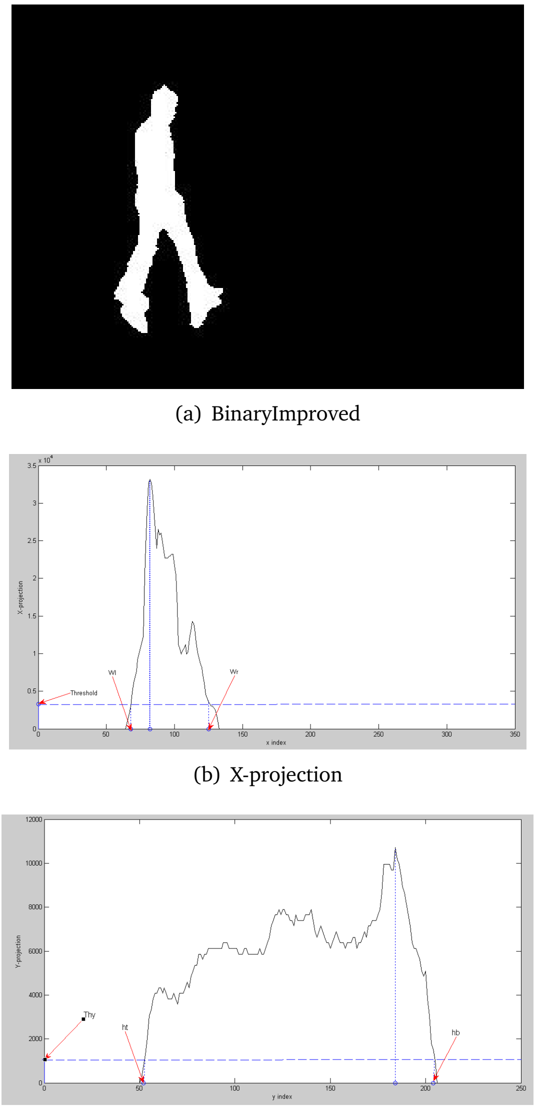
Aspect ratio is computed by projecting the pixels belong to the moving object on X and Y axises as shown in the Figure 21. Then we find the right and left borders of the minimum bounding rectangle by determining a threshold proportional to the maximum number of pixels in the columns of the foreground region after vertical projection as in equations 3.12, 3.13. And we find the top and bottom of the minimum bounding rectangle by using horizontal projection and determining threshold proportional to the maximum number of pixels in the columns as in the first step. Figure 21 shows in (a) foreground binary mask of moving object, (b) vertical projection and right, left borders, (c) horizontal projection and top, bottom borders.

The aspect ratio is a good low-level feature that we can use to discriminate lying down position from other positions. When the person standing or walking; the aspect ratio (height/width) is big and always larger than one, but in lying position, width is bigger than height, so the aspect ratio is small and less than one. In Figure 22 (a); we can see the aspect ratio values for the person standing, walking, falling down (lying down) and walking. We can clearly see the big differences in the aspect ratios of the lying down position and other positions.

Orientation is the second low-level extracted feature. We used the *regionprops* built in MATLAB function to get the orientation. *regionprops* function gives many statistics about the blobs. It is using second-moments statistics to find the ellipse that has the region (blob) pixels. Computing the orientation by using moments is explained in details in section 3.2.3. Orientation is the angle between the x-axis and the major axis of the ellipse. It's measured in degrees and ranges from -90 to 90. The orientation gives the overall direction of the object's shape. When the person is standing; the angle of the major axis will be around 90 angle degree with the x-axis, but when he is lying down; the angle will be very small. Figure 22 (b) shows the orientation of a person walking, standing and lying down. We can see in the figure that the orientation has values between 80 and 90 when the person walking or standing (note: we took the absolute value), and has values less than 10 degrees when he is lying down position. The orientation gives very good result to discriminate a lying down situation from other situations. Also from the *regionprops* function we get major and minor axises values. Major and minor axises are calculated also by using statistical moments to find the ellipse.

Another low level feature we extracted; is the height of the center of mass, we compute the height by calculating the distance between the center of mass and the bottom edge of the minimum bounding box of the object. Height of center of mass gives us information about how much the center of mass of the person is far from the floor. Figure 22 (c) shows how the height of center of mass has different values between walking and lying down situations. When a person is in lying down position; his center of mass should be near to the floor. We extract another low level feature to give more meaningful for the height of center of mass feature; we compute the height of smallest bounding box containing the blob.

The second type of extracted features depends on a sequence of frames. The objects are represented by their centers of mass. The first feature is the vertical direction of the motion, when the person falls down; his mask will move towards the floor, so his vertical motion will be large, but when he is waking or standing; his vertical motion will be almost constant if he is moving with constant speed away from the camera, or

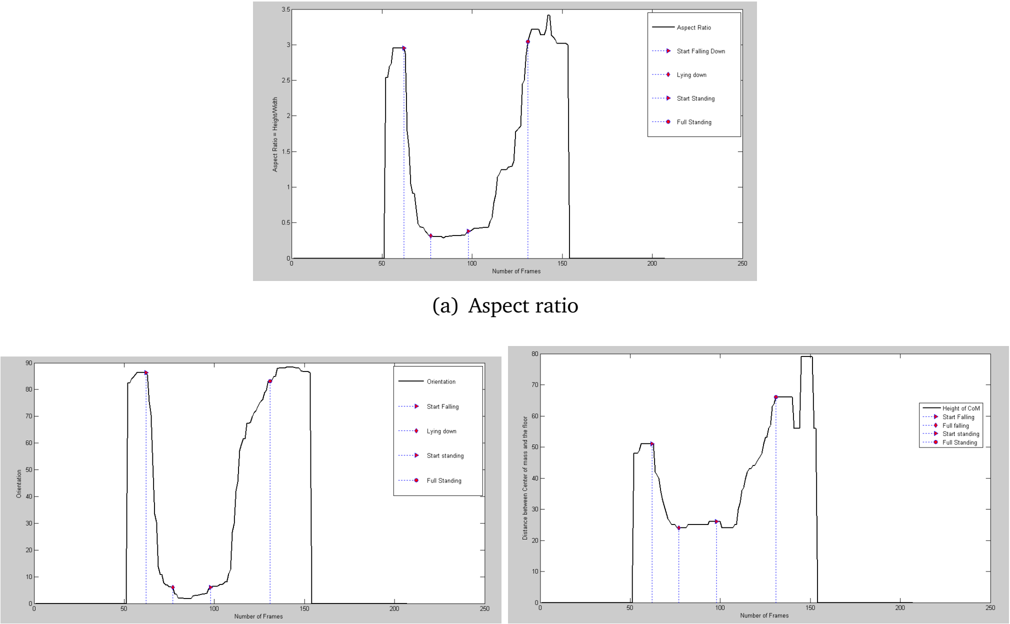


(a) BinaryImproved

(b) X-projection

(c) Y-projection

Figure 21: X - Y projections



(a) Aspect ratio

(b) Orientation

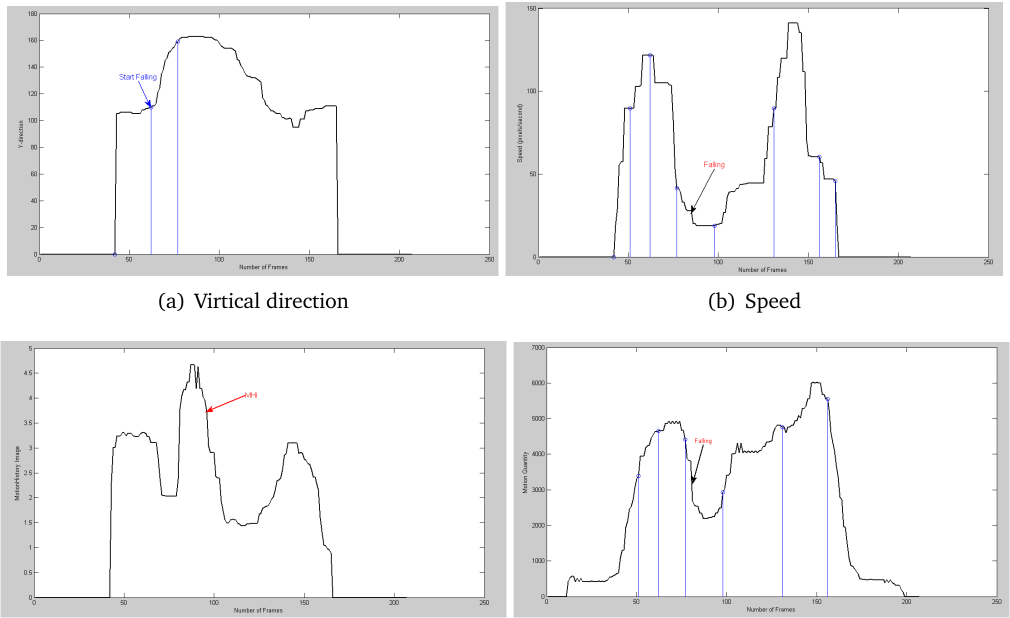
(c) Height of CoM

Figure 22: Features analysis: aspect ratio, orientation and height of center of mass

will change little bit if he is moving toward the camera or in opposite direction. When the person start to fall down; differences will be distinguishable in the vertical motion values in the direction toward the floor, see Figure 23 (a). When the person is in the lying position he will have no motion or little motion. We compute the vertical motion for the center of mass that represents the object and is measured in pixels. This feature is important to check when the person starts falling or sitting down to start further testing to resolve the ambiguous person situation.

The next feature we extracted from the sequence of frames is speed of objects. We compute the speed by computing the distance (in pixel) between the centers of mass of blobs in consecutive frames and divide it by the time. Figure 23 (b) shows the speed of moving object when he was walking then lying down, standing and walking again. The speed when the object is lying down is very small and we are interested in this. This feature gives us obvious idea if the person is not moving and when we have a fall, we need to check the person situation for a period after the fall to be sure if he is in lying position.

The motion quantity give a good imagination of what is a person doing. We assume that the person will have large motion when is starting to fall or to sit down. From the motion we extracte two features as we explained in the previous chapter. The first feature is called Motion History Image (MHI). The purpose of MHI is to detect when a large motion of a person happens. We quantify the motion of the person based on MHI by dividing the sum of blob's pixels in MHI by number of blob's pixels as in equation 3.17. Figure 23 (c) shows how is MHI quantity is large when the person start falling. The second feature we extracted from the motion is motion quantity. Motion quantity is the sum of moving pixels in three consecutive frames (we consider the pixel as moving pixel if it moves in the three consecutive frames) that belong to the object. When the person falling will be in the lying position, he will have no motion or little movements, so the motion quantity will be very small for a period when he is in lying position. But if he moves again like standing up and walk, the motion quantity will get higher values and we will not consider it a fall event. Figure 23(d) shows the motion quantity for a person walking, falling down, lying down for short period and then continue walking.



(a) Virtical direction

(b) Speed

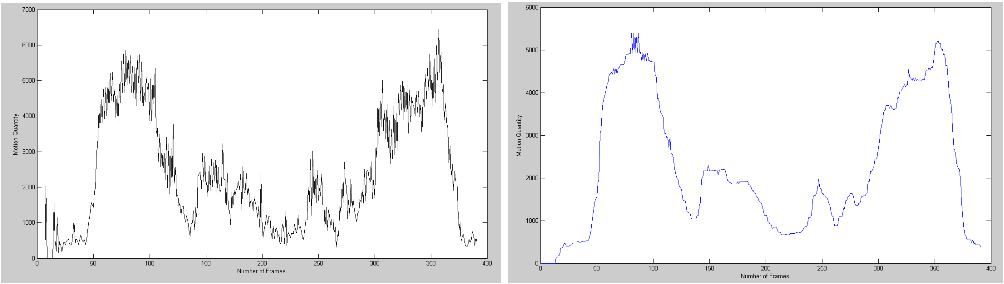
(c) Motion history image

(d) Motion quantity

Figure 23: Features analysis: virtical direction, speed, MHI and motion quantity

Smoothing:

We have variation in the blobs centroids and areas resulting from the problems in segmentation sometimes and the shadow effects, and this will affect all the features. So we use median filter to smooth the features signals to have more reliable values. Fig. 24 (a) shows a motion quantity feature values before smoothing; we can see the noise effect in the signal, and Fig. 24 (b) shows the signal after applying median filter with window equal 13. We applied same median filter for all the extracted features from video sequence.



(a) Motion Quantity before filtering

(b) Motion Quantity before filtering

Figure 24: Smothing

Thresholds

The features we have extracted are divided into two groups; the first group has the features that are extracted from the information of shape depending only on the current binary mask. These features are aspect ratio, height of center of mass, height of bounding box, orientation, major and miner axises. These features are grouped together in one features vector. The K-NN classifier will take this vector as an input and decide in which position the person is by depending on the K-NN training data that has. We will discuss the K-NN classifier in details in the classifier section in this chapter. The second group has the features that are extracted from a number of consecutive frames. These features are motion history image, direction of motion, motion quantity and speed. These features are given more valuable data during the fall event. In the purpose to find the best thresholds to discriminate falls from other activities, we have used two recorded video sequences (same video sequence that we used in segmentation testing) to analyze these features and define the thresholds to generalize them.

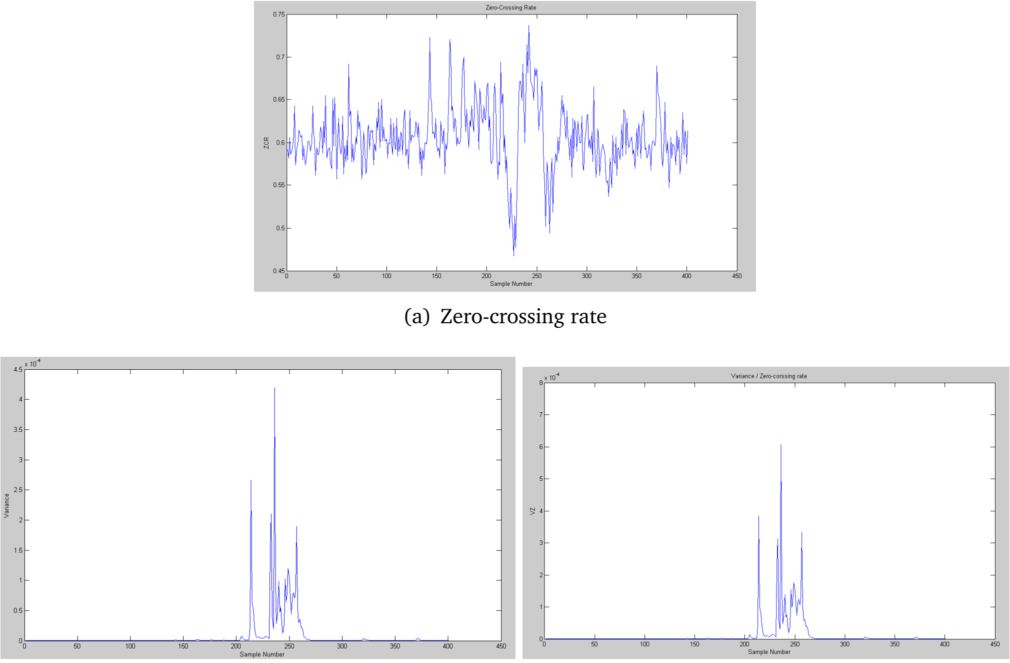
The motion history image quantity is scaled to a percentage of motion between 00%, no motion and 100%, full motion. From our testing data, we have observed that the duration of the fall is extremely short, typically less than a second. So we compute the motion history image by accumulation the motion during 5 frames (frame rate = 14 frame/second). Our detection system considers a motion as a possible fall if the coefficient CM (equation 3.17) is higher than 65%.

When the person is walking in front of the camera, his direction is changeable in the horizontal direction and almost steady in the vertical direction. Whereas when the person is sitting or falling down, he moves vastly in the vertical direction down toward the ground. To discriminate the falling case we have developed an automatic method to estimate the threshold. The threshold is estimated from the previous 30 values, we take the last previous 30 values, and then we compute the summation of the mean and the variance and take it as a threshold to discriminate the fall from other activities.

The speed and motion quantity features are used to check if the person is on the ground movements or with only little movements after a possible fall. We assume that the person will be on the ground with little movements. The speed is measured in pixels/second. From the testing results we found 40 pixels/second is a good threshold to discriminate when the person stays in one place, lying or sitting down with little movements, from walking or other moving activities. For the motion quantity threshold, we consider the blob as unmoving if it has motion quantity less than 15%.

4.2 Audio track analysis

Audio signals that we get from the camera microphone or from the independent microphone can also be used to discriminate between falling down event and other events. Usually when a fall happens, high amplitude of sound will be produced, where as the ordinary actions e.g. walking, sitting down and bending have no distinguishable sound from the background. Background sound can be music from CD player, TV,... etc.
In the purpose to discriminate falls from other events in the scene by analyzing the audio track, we have recorded 19 audio files that have falls and other activities such as walking, talking, sitting, and including background i.e. TV, movie, and music. Some of these files are recorded in the school (room A220 and A-building first floor corridor) together with the testing videos. The rest is recorded in my apartment. We used hama CS-471[1] microphone to record the audio waves. The recorded audio files are originally sampled in 44.1 KHz, 16 bits, stereo. Our algorithm is taking one channel (mono) for analyzing the audio track.
After getting the wavelet signals corresponding to audio track data, we calculated the variance over Zero-crossing rate (VZ) for 500-sample windows.
Figure 25(a) shows the ZCR corresponding to the audio track data for one of the testing audio files that has walking, talking and falling events, (b) shows the corresponding variance, and (c) shows the VZ. The parameter VZ takes non-negative values.



(a) Zero-crossing rate

(b) Variance

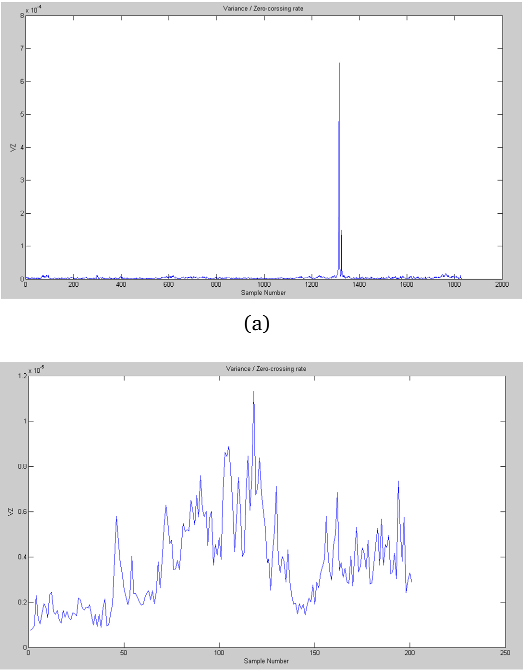
(c) Variance / Zero-crossing rate

Figure 25: Zero-crossing rate, variance and their ratio

Talking has a varying variance and ZCR characteristic depending on the utterance. When vowels are uttered, variance increases while ZCR decreases and this gives larger VZ values. Figure 26 shows the variation of VZ values versus sample numbers for different

[1] http://www.hama.co.uk

cases (walking, talking, sitting and falling down).



(a)

(b)

Figure 26: The ratio of variance over number of zero crossing, VZ; (a) falling(1311), (b) walking (500 - 541), walking and talking (542 - 602), talking and sitting (603 - 662), and talking (663-700). Note that VZ values for (b) are an order of magnitude less than (a)

Thresholds

To find the best threshold to discriminate between fall and other events, we analyzed all the 19 recorded audio files that have fall events and other different activates. We summarize the audio files in table 1. We used different background in the recording to include most of the possible situations; the backgrounds were music, talk, walking, music, songs, TV, movies and silence.

Table 1: Audio data testing

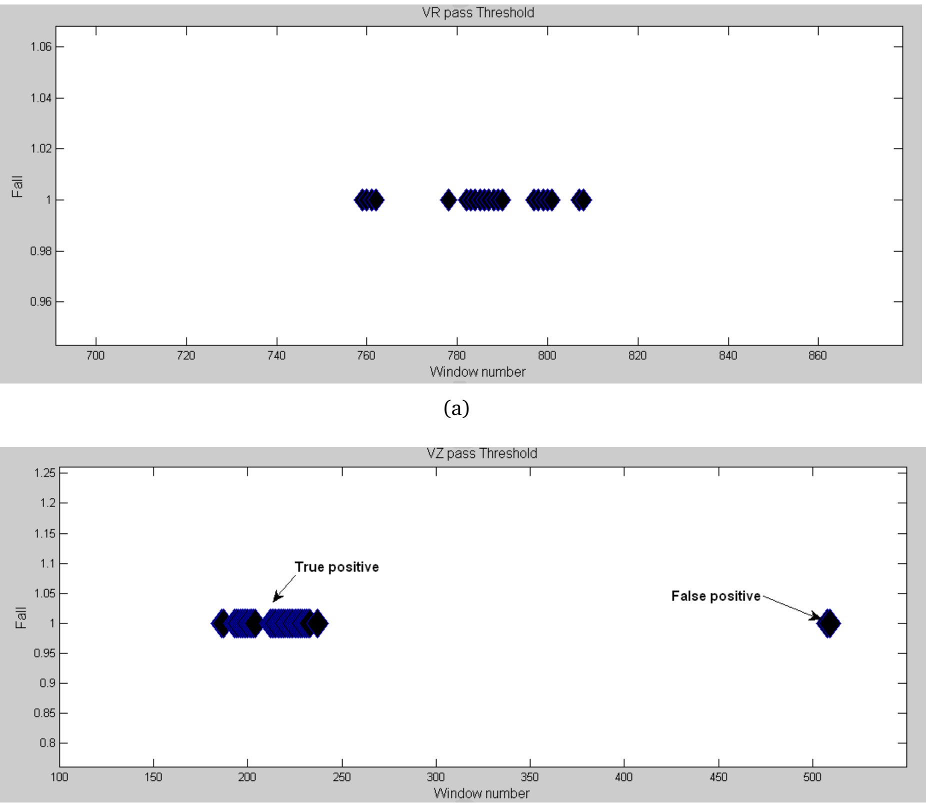
Audio Content	**No. of files**
TV + Walking + Falling	2
TV + + Walking + Talking + Falling	3
Music + Walking + Falling	2
Music + Walking + Talking + Falling	1
Movie + Walking + Falling	2
Walking + Falling	6
Talking + Walking + Falling	3

From the observing in taking the data set in audio and video, we observe that the fall down event gives obvious peak in VR signal and this peak is distinguishable from the

rest. After doing a lot of testing to find a best threshold to detect a fall event in different cases; we proposed a method to find the best threshold and update it recently when other activities are taken place.
The threshold is estimated from the specified training samples of VR values. In the beginning the system will take the set as the first hundred consecutive VR values corresponding to 100 consecutive sample windows, the threshold is updated every 500 sample windows if there is no falls.
If VR pass the estimated threshold, we consider it as a fall

$$Fall(i) = \begin{cases} true & \text{if } V(i) > TH \\ false & \text{otherwise.} \end{cases}, \tag{4.1}$$

In our experiment, we define TH as a proportional to the maximum VR in the training set (100 consecutive VR values without any fall). In our testing audio files, we found $TH = 5max(VR)$ gives good result.
We test the system on all the training audio files, it detects always the falls, an example shown in figure 27(a), but in some test audio files that have loudly talking gives sometimes false positive falls (detect a non-fall as a fall) as shown in figure 27(b), but we pass this problem by combining the audio and video results to decrease the false positives.



(a)

(b)

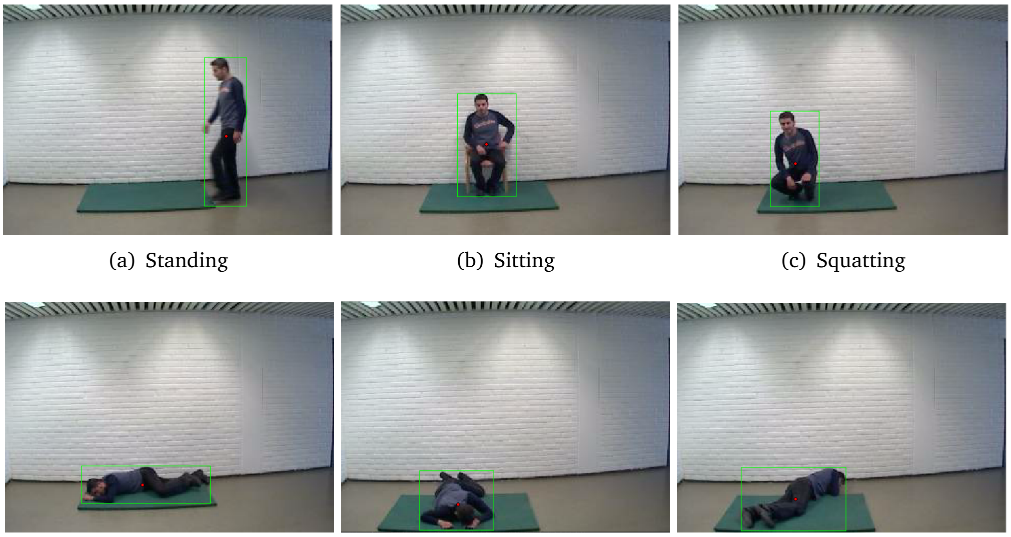
Figure 27: VZ passing the threshold, (a) true positive fall at window (759), (b) true positive fall at window (186), and false positive fall at window (508).

4.3 Classifier

After getting the features vector from the video sequence analysis and the VZ parameter from the audio signal, we have developed a method to utilize these features to classify a fall event from other events that are taken place in the scene. Our classifier has five decisions as shown in figure 18, and include K-NN classifier as one part of it. In the beginning we have trained the K-NN classifier. Then we have tested all the classifier parts (decisions) individually and later on we have tested the all classifier parts together on one testing video that we have recorded it at school (the corridor in A-building). In the rest of the section we will explain K-NN training process and discuss the testing results of the testing video.

4.3.1 Training K-NN

In the purpose of training K-NN algorithm we have recorded 24 video sequences; the training video sequences have variance activities. They have mainly six positions (standing, sitting, squatting, lying side by the camera, lying toward the camera, and lying by making 45 degree angle with the camera). Figure 28 shows the main postures that our training video sequences have. In Table (2) we summarize the video sequences that we have used to take frames to define the standing and walking positions. In Table (3) we summarize the video sequences that we have used to define the lying down positions. The lying down positions are divided into three positions. The lying down positions are lying side by the camera as shown in figure 28(d), lying toward the camera as shown in figure 28(e), and lying by making 45 degree angle with the camera as in figure 28(f). In Table (4) we summarize the video sequences that we have used to take frames to define the sitting, squatting and kneeing positions.


(a) Standing (b) Sitting (c) Squatting

(d) Side lying (e) lying toward the camera (f) lying in 45 degrees

Figure 28: KNN postures

As we have mentioned in the implementation chapter, K-NN algorithm takes six features as an input and depending on the training data will classify the person position in one of three groups. The inputs are aspect ratio, orientation, height of center of mass, height of bounding box, major axis and minor axis. The output is standing, sitting or lying down. We have selected 673 input features vector from the training video sequences to define these three postures. In the purpose of defining the standing posture we have selected features vectors from *Walk1*, *Walk2*, *Walk3*, and *Walk6* video sequences. We have selected the specified vectors after the segmentation process, and after observing the foreground images. The selected vectors are corresponding to foreground images that have optimized segmentation without any noise. For defining the falling down posture we have selected features vectors from *Falling1* and *Falling2* video sequences. And For defining the sitting down posture we have selected features vectors from *Sitting1* and *Sitting2* video sequences. We have used the same methodology in the selecting process in defining lying down and sitting postures same as standing posture. We have normalized all the feature sets to avoid the biasing by the larger values.

Figure 29 shows the distribution of the training data, the lying down is obviously discriminated from sitting and standing postures. The sitting down centroid is near more to standing centroid than lying down centroid, this is because the orientation is almost the same in both situations. We are interesting in discriminating the lying down posture from other postures, so this will give more opportunities to our classifier.

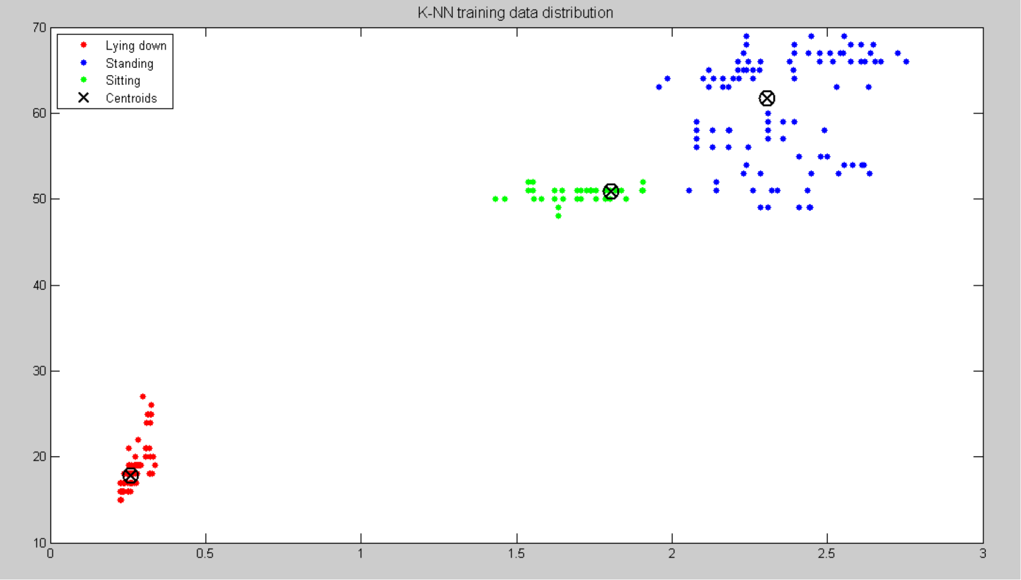


Figure 29: K-NN training data distribution

Table 2: K-NN standing training videos

No.	Video file name	Duration	Number of frames	Place	Description
1	Walk1	9 seconds	135 frames	A-building's corridor	walking in front of the camera
2	Walk2	12 seconds	165 frames	A-building's corridor	walking in front of the camera
3	Walk3	9 seconds	137 frames	A-building's corridor	walking in front of the camera
4	Walk4	7 seconds	104 frames	A-building's corridor	walking in front of the camera
5	Walk5	7 seconds	94 frames	A-building's corridor	walking in front of the camera
6	Walk6	7 seconds	102 frames	A-building's corridor	walking in front of the camera
7	Walk7	13 seconds	172 frames	room no. A128	walking in front of the camera
8	Walk8	11 seconds	163 frames	room no. A128	walking in front of the camera
9	Walk9	11 seconds	165 frames	room no. A128	walking in front of the camera

Table 3: K-NN lying down training videos

No.	**Video file name**	**Duration**	**Number of frames**	**Place**	**Description**
1	Falling1	27 seconds	391 frames	A-building's corridor	walking in front of the camera, falling down and staying in lying side position for some moments
2	Falling2	24 seconds	344 frames	A-building's corridor	walking, falling down and staying in lying side position for some moments
3	Falling3	23 seconds	332 frames	A-building's corridor	walking , falling down and staying in lying toward the camera position for some moments
4	Falling4	18 seconds	282 frames	A-building's corridor	walking, falling down and lying in 45 degree angle toward the camera for some moments
5	Falling5	34 seconds	480 frames	room no. A128	walking, falling down and staying in lying side position for some moments
6	Falling6	21 seconds	297 frames	room no. A128	walking, falling down and staying in lying side position for some moments
7	Falling7	25 seconds	383 frames	room no. A128	walking in front of the camera

Table 4: K-NN sitting down, squatting and kneeing training videos

No.	Video file name	Duration	Number of frames	Place	Description
1	Sitting1	16 seconds	237 frames	A-building's corridor	walking and sitting on a chair for some moments
2	Sitting2	25 seconds	359 frames	A-building's corridor	walking and sitting on a chair for some moments
3	Sitting3	21 seconds	328 frames	A-building's corridor	walking and sitting on a chair for some moments
4	Sitting4	30 seconds	451 frames	room no. A128	walking and sitting on a chair for some moments
5	Sitting5	28 seconds	431 frames	room no. A128	walking and sitting on a chair for some moments
6	Squatting1	16 seconds	246 frames	A-building's corridor	walking and squatting for some moments
7	Squatting2	29 seconds	135 frames	A-building's corridor	walking and squatting for some moments
8	Kneeing1	15 seconds	195 frames	A-building's corridor	walking and kneeing for some moments

4.3.2 Testing K-NN

We have tested the K-NN classifier on the test video sequence that was taken for this purpose at the school (the corridor in A-building). The test video sequence has one person moving, first he enter the scene, walking, falling down, lying down for some moments, and then standing to exit the scene. Table (5) summarize the activities that are taken place in the test video sequence.

Table 5: K-NN testing video sequence description

Frame number	Activity description
43	The person start to enter the scene
52	The person has full appearance
62	The person start falling down
77	The person is in lying down shape
98	The person start to stand up
131	The person is in standing shape and start walking
154	The person start to exit the scene
166	The person is disappearing

We have tested K-NN algorithm on our testing data. Figure 30 shows the output of K-NN classifier. The K-NN gives output that the person has a standing posture when the person walking (from frame # 52 to # 65 and from frame # 129 to # 153), when the person is in falling down activity and when he is standing up again, the K-NN output changes from standing posture to sitting and the opposite in standing up case. In the testing video sequence, the person is in lying down position from frame # 77 to frame # 98, the K-NN gives that the person is in lying down position from frame #69 to frame #118 and this include the region that we are interesting in it. During the fall and standing up processes, the person shape is moving between the three postures, so the K-NN classifier will give the closest one according to the training data set. To find the most suitable number of neighbors to consider, a classification was performed on K = 3, 5 and 7. The results give the same for lying down posture, we define K = 5 for all the rest experiments.

All the features in the features vectors we have used as a training data for K-NN algorithm are normalized to their minimum and maximum values. We stored the minimum and maximum values for each feature. In the testing process, we normalize the input feature vector to these values.

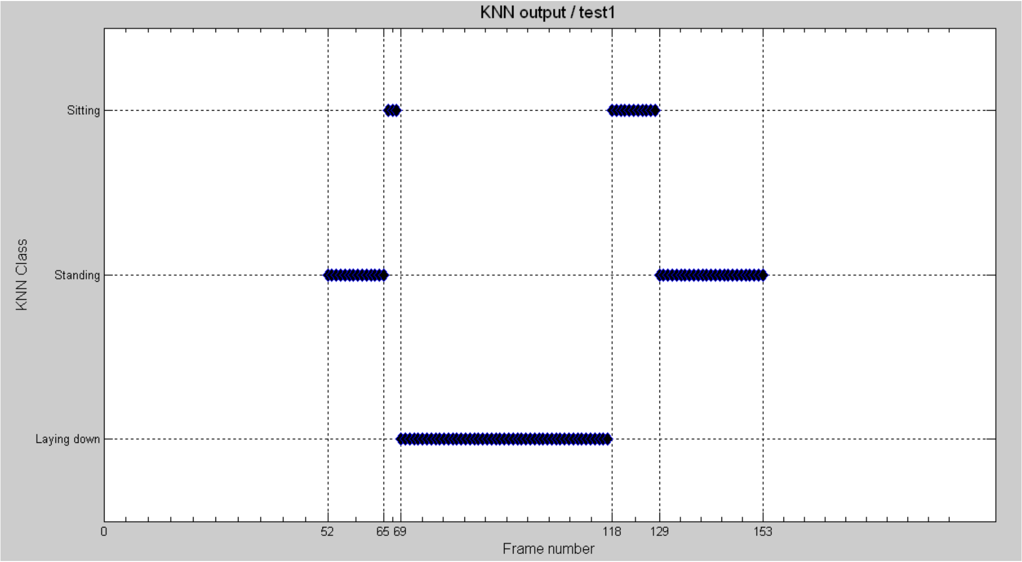


Figure 30: K-NN test output

4.4 Final experiment

In the final experiment we have used nine video sequences including their corresponding audio waves. We have recorded the audio in separately files during the experiment. We have synchronized the recorded data manually. The testing data are summarized in Table (6) with the description of their content. We have tested the recorded data offline, our system is reading the video sequences and the audio waves separately, and then combine the extracted features from both video and audio to make its decision. The extracted data from video and audio are synchronized and combined.

In all the experiment data set, there is only one moving object exists in the scene. The experiment data was taken only from the author thesis at the school. The data set has different activities, five clips have falling down events, two clips have sitting down, and two clips have only walking. The falling down event duration average is around one second. After the falling process, the person was lying down on the ground for a period from 2 seconds as in Video-1, to more than 10 seconds as in Video-3 before standing up again and continues walking.

The experiment was performed on an Intel Pentium M 2.00GHz processor laptop with 1.5Gbytes RAM. And we have used the K-NN algorithm that we have trained it on lying, standing, and sitting postures as we discussed in the K-NN training section. The classifier works by making consecutive decisions as seen in Figure 18. The final output is taken from the last decision (D5). D4 checks if a possible fall happens. If a possible fall happens, D5 will check if the person is on the ground without movements (little motion and speed) for a while. In our experiment we have defined the checking interval after a possible fall as 3 seconds. If the system detects a fall, it will give an alarm.
The thresholds that have been used for the final experiment were the same as the thresholds that have selected during the video and audio analysis processes. For K-NN algorithm we defined K = 5.

Table 6: Final experiment data set

Video file name	**Duration (seconds)**	**Number of frames**	**Place**	**Content description**
Video-1	14	207	A-building corridor	walking in front of the camera, falling down and staying in lying down position for some moments.
Video-2	27	391	A-building corridor	walking, falling down and staying in lying down position for some moments.
Video-3	24	344	A-building corridor	walking in front of the camera, talking, falling down and staying in lying down position for some moments.
Video-4	23	332	A-building corridor	walking , falling down and staying in lying toward the camera position for some moments.
Video-5	18	282	A-building corridor	walking, falling down and lying in 45 degree angle toward the camera for some moments.
Video-6	9	135	A-building corridor	walking in front of the camera and talking.
Video-7	12	165	A-building corridor	walking in front of the camera and talking.
Video-8	16	237	A-building corridor	walking, talking and sitting down on a chair for some moments.
Video-9	25	359	A-building corridor	walking, talking and sitting down on a chair for some moments.

The outputs of the testing set are summarized in Table (7). The experiment data set has 9 clips. Five clips have five falls, each clip has one fall. The data set also has 2 clips that have only walking activity. Moreover, there are two clips of a person sitting down. In five of the clips, the person was talking during walking and sitting down events.
The system has detected 4 falls (Video-2, Video-3, Video-4, and Video-5). The detection results of the tested data set with description of the content are summarized in Table (7).

Table 7: Detection results for the data set

Video file name	Video content	Falling is detected
Video-1	Walking + falling	No
Video-2	Walking + falling	Yes at frame # 139
Video-3	Walking + falling + talking	Yes at frame # 129
Video-4	Walking + falling	Yes at frame # 182
Video-5	Walking + falling	Yes at frame # 135
Video-6	Walking + talking	No
Video-7	Walking + talking	No
Video-8	Walking + sitting + talking	No
Video-9	Walking + sitting + talking	No

The output is given at each frame; the system is checking the features vector values from images sequence and its corresponding audio track features at each frame. The image analysis results are combined with audio analysis results in the time domain at each frame. The output is simply has two possible values (Fall or Not fall).
Figure 31 shows the system output of Video-2 clip. The system detects the fall at frame # 139 (at tenth second).

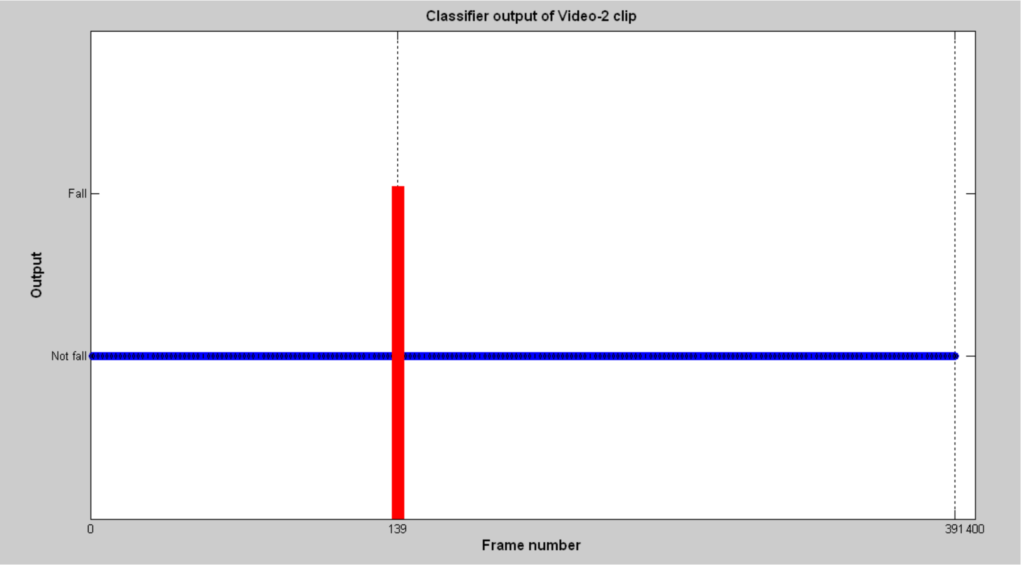
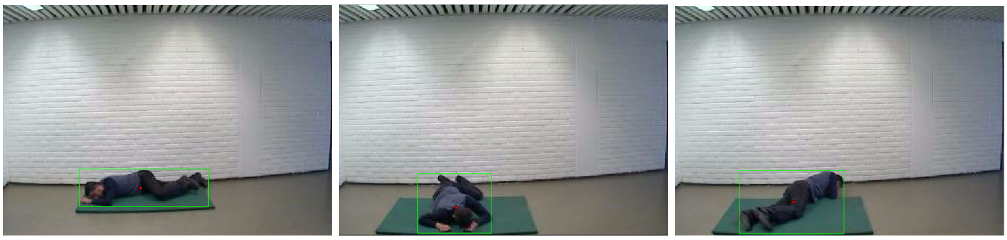


Figure 31: System output of testing Video-2 clip

4.4.1 Discussion and evaluation

Our data set for the final experiment is composed of video sequences have different activities as summarized in Table (6). The data set has five clips to represent different possibilities of falling down in front of the camera. Figure 32 shows the person postures after falling down. The data set has three clips that the person is falling down and lying side in front of the camera, one clip that the person is falling down and lying toward the camera, and one clip that the person is falling down and lying in 45 degree angle with the camera.



(a) side lying (b) lying toward the camera (c) lying in 45 degrees

Figure 32: Lying postures

Table (8) itemizes the final experiment results. There is five falls in our data set. The system has detected 4 of them (True Positive = 4, and False Negative = 1). In our data set, there is also 4 clips that have no falls, they have only walking and sitting down activities. The system has detected Not fall in all of them (False Positive = 0, and True Negative = 4).

Table 8: Recognition results (interval = 3 seconds)

	Detected	**Not detected**
FALLS	True Positive: 4	False Negative: 1
LURES	False Positive: 0	True Negative: 4

Analysis process of the data set and final experiment results has been done. The fall event is short, its average one second. In our data set, after each fall, the person stays on the ground for a while, and then he stands up and continues walking. The period that he is staying on the ground its ranges from 2 seconds (Video-1) to more than 10 seconds (Video-2). When the person stays on the ground, the motion and speed are small (pass the defined threshold). But when he starts to stand up again, the motion and speed will be larger. We have performed the experiment again on smaller interval (2 seconds) to check if the person is lying down with little movements. Table (9) summarizes the output when the interval is only 2 seconds.

Table 9: Detection results for the data set taken from D4 (see Figure 18)

Video file name	**Video content**	**Falling is detected**
Video-1	Walking + falling	Yes at frame # 69
Video-2	Walking + falling	Yes at frame # 85
Video-3	Walking + falling + talking	Yes at frame # 84
Video-4	Walking + falling	Yes at frame # 126
Video-5	Walking + falling	Yes at frame # 74
Video-6	Walking + talking	No
Video-7	Walking + talking	No
Video-8	Walking + sitting + talking	No
Video-9	Walking + sitting + talking	No

The second experiment detects all the falls, gives 0 False Negative as seen in and Table (10). But decreasing the interval will increase the risk of increasing the False Positives.

We should take in our consideration in the analysis that the data set are recorded from the same person, and after each fall, the person stays for few seconds on the ground with little movements and then stands up again and continues his activity.

Table 10: Recognition results (interval = 2 seconds)

	Detected	**Not detected**
FALLS	True Positive: 5	False Negative: 0
LURES	False Positive: 0	True Negative: 4

In this thesis work, we assume when the person is on the ground after a possible fall, he will have a little movements. This can be argued in some cases, for example in the injury cases where the person could move rapidly because of the pain. This can be solved by defining more than one model to deal with several cases and to define several safety levels (see the future works of this thesis).

The experiment was performed offline. Table (11) summarizes the processing time for each clip in our data set. From this, we can see clearly that the system can be applied in a real-time.

Table 11: Experiment files processing time

Video file name	**Duration (seconds)**	**Processing time (seconds**
Video-1	14	14.7846
Video-2	27	26.9922
Video-3	24	22.8427
Video-4	23	21.5909
Video-5	18	10.7556
Video-6	9	9.3145
Video-7	12	10.3366
Video-8	16	17.0589
Video-9	25	23.6261

5 Conclusion and future work

In this project we have developed a method to detect automatically fall incidents in elderly's houses. Our method is based on combining audio and video features to decide if a fall happens in a video sequence. The system includes foreground segmentation, low-level features extraction, and the events classification tasks. The conclusions in this section provides answers to our initial research questions.

For the purpose of getting clear foreground image that have only the moving objects without noise; we implemented several improvements on the segmentation algorithm to improve the foreground binary mask (see section 3.2.1). Firstly, we pass the binary image that we get from the background subtraction to a shadow removal algorithm to remove the shadow effects. Secondly, we do morphological operation on the output of the shadow removal algorithm. Finally, the tracking algorithm improves the foreground binary mask by matching, merging and splitting techniques. The segmentation algorithm gives clear foreground binary masks in our experiments.

We extract two groups of low-level features from the images sequence and two features from the audio track.
The first group has six low-level features (aspect ratio, height of center of mass, height of bounding box, orientation, major axis and minor axis). These features are extracted only from the information of the current foreground frame. The objective of extracting this group of features is to discriminate the lying down posture from other postures (standing, sitting, and squatting). We have used K-NN classifier to classify these postures. The experimental results proved that we could clearly discriminate a lying down posture from standing or sitting postures by combining the previous features.

The second group has four level features (motion history image, direction of motion, motion quantity, and speed). These features are extracted from consecutive number of frames. The motion history image and direction of motion are used to classify when a person starts to fall and during the falling event. Motion quantity and speed are used to check if the person stays on the ground for a while after a possible fall.

From the audio track, we extract two features from the corresponding wavelet coefficients of the audio signal. We extract the variance and zero-crossing rate. Our experiments proved that the audio has valuable information to discriminate a fall from other ordinary actions as video. The audio information is essential to distinguish a falling person from a person normally sitting down or sitting on the floor.

A novel classifier is developed to combine all the low-level features from video and audio to classify the events in the scene and to detect a fall if it's happen. Our system considers a fall as real fall if the person stays on the ground without or with little movements after a possible fall for a few seconds. The classifier includes the K-NN algorithm and contains five decisions. Depending on the classifier output; the system will give an alarm or notification if a real fall happens.

The experimental results demonstrate that the system can work robustly in indoor environments. The system can be easily deployed in elderly's houses or homes to take care of the people who need help.
The processing time of the experimental data proves the capability of the system to work in real-time.

The system is implemented to detect fall incidents in elderly's houses but it has some flexibility to fit other applications. The algorithms in the system can be used for other video surveillance applications or in other event detection fields.

5.1 Future work

Experiments in this project have proven that including the audio in the system increase the robustness of the proposed fall detection system. We have extracted the variance and zero-crossing rate features from the audio track, further analysis and extracting more features from the audio track could increase the robustness in our system. A speech recognition algorithm could also be used to identify calls for "help!".

Do further studies about the senior's life, what are the possible falls, and do more analysis of their life to include most of the possible cases in the fall detection system. Define several models to deal with several possible fall down cases and define several safety levels could be one of the future tasks to further the work in this thesis.

The system is tested on the data we have recorded in two places in the school. The tested data has only one moving person. One of the future works on this project is to take more data in different places and perform the experiment in larger scale. Our system has already tracking algorithm that can handle with occlusions, but our tested data has only one person, so testing the algorithm on data that has more than one person in the scene and including occlusions will be a future task of our project. Further testing on the segmentation algorithm and improve the shadow removal algorithm could also be one of future tasks of our project.

Single camera limits the viewing angle of the scene and more importantly the resident. Using multiple cameras in a single scene will increase the robustness. Using the 3D information to check if the head is near the floor for instance could be one of suggestions for the future works.

Improve the system to include personal information such as shape and health information could be one of the future works. Define normal inactivity zones in the scene could decrease the false positives and give more robustness for the system. The inactivity zones could be bed, chair or other typical furniture.

Discrimination between the human moving objects and non-human objects as pets or other devices is one of the future works. The elderly people probably they using some tools to help them in moving such as crutch and walker, it could be one of the future tasks to improve the system to include this situation.

6 Legal and ethical considerations

The video surveillance systems are useful for monitoring suspicious circumstances or vulnerabilities. There are many benefits for using the video surveillance systems to increase the safety and the security. In our case, using video surveillance to detect fall events in elderly's houses will increase the safety level, the quality of live and care, and decrease the risk.

The use of this system can be in elderly's houses, children's rooms and sick person's rooms. And as mentioned before, the purpose is to increase the safety level and help the residents. The privacy of the residents is ensured through many steps: we extract some features from the binary foreground images that have blobs represent the persons, then we used these features to analysis the activates. The system is run automatically and the outputs are numbers to explain and distinguish activities.

The Norwegian Personal Data Act [50] mentions that the video surveillance is part of the processing of personal data (section 37), and the personal data law will apply for it. The personal data "may only be disclosed to a person other than the controller if the subject of the recording consents thereto or if there is statutory provision for such disclosure" (section 39) [50]. And if the surveillance system is running in a place which is regularly visited by a group of people, notification that the place is under surveillance shall be drawn clearly (section 40).

The purpose of the project is to use the technology to facility the resident's life while respecting the privacy and confidentially interests of all relevant stakeholders. And the intrusion into living spaces of elderly's or other people needing help, may be justified by the benefits from this system [51].

The experiments are carried out in the school, the movies are taken only from the thesis author to make analysis and train the system.

Bibliography

[1] Report on seniors's falls in canada. *Public Health Agency of Canada, Division of Aging and Seniors*, 2005.

[2] Canada's aging population. *Public Health Agency of Canada, Division of Aging and Seniors*, 2002.

[3] S. Brownsell and M. Hawley. Automatic fall detectors and the fear of falling. *Journal of telemedicine and telecare*, 10(5):262, 2004.

[4] A. Sixsmith and N. Johnson. A smart sensor to detect the falls of the elderly. *Pervasive Computing, IEEE*, 3(2):42–47, April-June 2004.

[5] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *Signal Processing Magazine, IEEE*, 22(2):38–51, March 2005.

[6] M. Piccardi. Background subtraction techniques: a review. *International Conference on Systems, Man and Cybernetics, IEEE*, 4:3099–3104 vol.4, Oct. 2004.

[7] L. F. Teixeira, J. S. Cardoso, and L. Corte-Real. Object segmentation using background modeling and cascaded change detection. *Journal of Multimedia*, 2(5):55–65, SEPTEMBER 2007.

[8] M. Nixon and A. Aguado. *Feature Extraction and Image Processing*. Newnes, 2002.

[9] R. Wildes. A measure of motion salience for surveillance applications. *In Proceedings of the International Conference on Image Processing (ICIP 98)*, pages 183–187 vol.3, 4-7 Oct 1998.

[10] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2000.

[11] D. Lefloch, F. Alaya-Cheikh, Jon Y. Hardeberg, P. Gouton, and R. Picot-Clemente. Real-time people counting system using a single video camera. *In Proceedings of the SPIE on Real-Time Image Processing Volume 6811, pp. 681109-681109-12*, March 2008.

[12] A. Leone and C. Distante. Shadow detection for moving objects based on texture analysis. *Pattern Recognition*, 40:1222–1233, April 2007.

[13] J. Heikkila and I. Silven. A real-time system for monitoring of cyclists and pedestrians. *Image and Vision Computing*, 22(7):563–570, July 1 2004.

[14] A. Pentland, T. Darrell, A. Azarbayejani, and C. Wren. Pfinder: Real-time tracking of the human body. In *International Conference on Automatic Face and Gesture Recognition*, pages 51–56, 1996.

[15] I. Haritaoglu, D. Harwood, and L. Davis. W4: Who? when? where? what? A real time system for detecting and tracking people. In *IEEE International Conference on Automatic Face and Gesture Recognition*, April 14 1998.

[16] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *IEEE Workshop on Motion and Video Computing*, pages 22–27, 2002.

[17] Y. Ren, C. Chua, and Y. Ho. Motion detection with non-stationary background. *In Proceedings. 11th International Conference on Image Analysis*, pages 78–83, Sep 2001.

[18] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999*, 2:–252 Vol. 2, 1999.

[19] S. Huwer and H. Niemann. Adaptive change detection for real-time surveillance applications. In *Workshop on Visual Surveillance*, pages xx–yy, 2000.

[20] L. Wixson. Detecting salient motion by accumulating directionally-consistent flow. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):774–780, 2000.

[21] S. Lim and A. El Gamal. Optical flow estimation using high frame rate sequences. *In Proceedings International Conference on Image Processing*, 2:925–928 vol.2, Oct 2001.

[22] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. DARPA Image Understanding Workshop*, pages 121–130, 1981.

[23] Y. L. Tian and A. Hampapur. Robust salient motion detection with complex background for real-time video surveillance. In *IEEE Workshop on Motion and Video Computing*, pages II: 30–35, 2005.

[24] K. Sundaraj and V. Retnasamy. Fast background subtraction for real time monitoring. In *ACST'07: Proceedings of the third conference on IASTED International Conference*, pages 382–387, Anaheim, CA, USA, 2007. ACTA Press.

[25] O. Schreer, I. Feldmann, U. Golz, and P. Kauff. Fast and robust shadow detection in videoconference applications. *Video/Image Processing and Multimedia Communications 4th EURASIP-IEEE Region 8 International Symposium on VIPromCom*, pages 371–375, 2002.

[26] X. Chen, Z. H. He, D. Anderson, J. Keller, and M. Skubic. Adaptive silouette extraction and human tracking in complex and dynamic environments. In *International Conference on Image Processing*, pages 561–564, 2006.

[27] T. Horprasert, D. Harwood, and L. S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *FRAME-RATE: Frame-rate Applications, Methods and Experiences with Regularly Available Technology and Equipment*, 1999.

[28] Y. Wang, J. F. Doherty, and R. Van Dyck. Moving object tracking in video. In *AIPR '00: Proceedings of the 29th Applied Imagery Pattern Recognition Workshop*, page 95, Washington, DC, USA, 2000. IEEE Computer Society.

[29] F. Bunyak, I. Ersoy, and S.R. Subramanya. A multi-hypothesis approach for salient object tracking in visual surveillance. *In IEEE International Conference on Image Processing, (ICIP 2005).*, 2:II–446–9, Sept. 2005.

[30] Q. Wan and Y. Wang. Multiple moving objects tracking under complex scenes. *The Sixth World Congress on Intelligent Control and Automation,(WCICA 2006).*, 2:9871–9875, June 2006.

[31] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2nd Quarter 1998.

[32] F. Porikli and T. Haga. Event detection by eigenvector decomposition using object and frame features. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04), Volume 7*, page 114, Washington, DC, USA, 2004. IEEE Computer Society.

[33] A. Y. Johnson and A. F. Bobick. A multi-view method for gait recognition using static body parameters. In *AVBPA '01: Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 301–311, London, UK, 2001. Springer-Verlag.

[34] Y.Ma, M. Bazakos, Z. Wang, and W. Au. 3D scene modeling for activity detection. In *Proceedings of Perspectives in Conceptual Modeling, ER Workshops AOIS, BP-UML, CoMoGIS, eCOMO, and QoIS, Klagenfurt, Austria, October 24-28, 2005*, volume 3770 of *Lecture Notes in Computer Science*, pages 300–309. Springer, 2005.

[35] A. Sixsmith and N. Johnson. A smart sensor to detect the falls of the elderly. *In IEEE, Pervasive Computing,*, 3(2):42–47, April-June 2004.

[36] B. U. Toreyin, Y. Dedeoglu, and A. E. Cetin. HMM based falling person detection using both audio and video. In *Computer Vision in Human-Computer Interaction*, page 211, 2005.

[37] N.P. Cuntoor, B. Yegnanarayana, and R. Chellappa. Interpretation of state sequences in hmm for activity representation. *Proceedings in IEEE International Conference on Acoustics, Speech, and Signal Processing,(ICASSP '05).*, 2:709–712, March 18-23, 2005.

[38] T. Lee and A. Mihailidis. An intelligent emergency response system: preliminary development and testing of automated fall detection. *Journal of telemedicine and telecare*, 11(4):194–198, June, 2005.

[39] H. N. Charif and S. J. McKenna. Activity summarisation and fall detection in a supportive home environment. In *ICPR (4)*, pages 323–326, 2004.

[40] A. Nasution and S. Emmanuel. Intelligent video surveillance for monitoring elderly in home environments. *IEEE 9th Workshop on Multimedia Signal Processing, (MMSP 2007).*, pages 203–206, 1-3 Oct. 2007.

[41] J. Tao, M. Turjo, and Yap-Peng Tan. Quickest change detection for health-care video surveillance. *Proceedings. 2006 IEEE International Symposium on Circuits and Systems, (ISCAS 2006).*, pages 4 pp.–, May 2006.

[42] D. Anderson, J. M. Keller, M. Skubic, X. Chen, and Z. He. Recognizing falls from silhouettes. *Conf Proc IEEE Eng Med Biol Soc*, 1, 2006.

[43] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Fall detection from human shape and motion history using video surveillance. In *AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, pages 875–880, Washington, DC, USA, 2007. IEEE Computer Society.

[44] S. Miaou, P. Sung, and C. Huang. A customized human fall detection system using omni- camera images and personal information. In *1st Distributed Diagnosis and Home Healthcare (D2H2) conference*, April 2-4, 2006.

[45] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *Transactions on Pattern Analysis and Machine Intelligence, IEEE*, 23(3):257–267, Mar 2001.

[46] W. K. Pratt. *Digital Image Processing: PIKS Inside*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[47] A. K. Jain. *Fundamentals of digital image processing*. Englewood Cliffs, Prentice Hall, New Jersey NJ, USA, 1989.

[48] F. Jabloun, A. E. Cetin, and E. Erzin. Teager energy based feature parameters for speech recognition in car noise. *Signal Processing Letters, IEEE,*, 6(10):259–261, Oct 1999.

[49] E. Gustavsen. Classifying motion picture audio. *Gjøvik University College*, July 2007.

[50] Personal data act, act of 14 april 2000 no. 31 relating to the processing of personal data, chapter vii, sections 36-41. (last visited May 16 2008).

[51] J. Ashok, J. Alex, B. David, W. Howard, A. Mary, and F. Charles. Ethical considerations in the conduct of electronic surveillance research. *journal of law, medicine and ethics, race and ethnicity*, 34:611–619, Fall. 2006.