Technical University of Crete

Department of Production Engineering and Management

“Humans’ fall detection through visual cues”

Master Thesis

Konstantinos Makantasis

Supervisor: Assistant Professor Anastasios Doulamis

Chania 2011

To my family.

Preface

This master thesis was prepared in Production Engineering and Management department of Technical University of Crete and it needed seven months to be completed, from February 2011 till September 2011.

At this point, I have to thank my supervisor professor Anastasios Doulamis for his useful and valuable advices and Meropi Manataki for psychological support.

September 2011

Konstantinos Makantasis

Table of Contents

Abstract

Population in developed countries is ageing. The quality of life for elderly is associated with their ability to live independently and with dignity without having the need to be attached to any person whose help would they need for their daily life and social behavior. On the other hand, according to medical records, traumas resulting from falls in the third age have been reported as the second most common cause of death for the Elderly. For this reason, a major research effort has been conducted in the recent years for automatically detecting persons' falls especially for the Elderly. Such identification is a prime research issue in computer vision society due to the complexity of the problem as far as the visual content is concerned.

In this master thesis, a fast real time computer vision algorithm able to detect humans' falls in complex dynamically changing visual conditions, was implemented. The algorithm exploits single cameras of low cost while it requires minimal computational cost and memory requirements. Due to its affordability it can be straightforwardly implemented in large scale clinical institutes/home environments.

Chapter 1

Introduction

Life expectancy in developed countries is increasing and population is ageing. This mostly caused by the development of medical science and public health during the 20th century. Through medical science and public health intervention many diseases and injuries are preventable and manageable while other diseases are considered extinct.

However, the quality of life is, partially, depended on medical science and public health. Especially, for elderly, it is associated with their ability to live independently and with dignity, without having the need to be attached to their children, grand-children or any other person in order to live a normal life and fulfill daily living, physical and social activities.

Falls are the leading cause of injury-related visits to emergency departments and the primary etiology of accidental deaths in persons over the age of 65 years. The mortality rate for falls increases dramatically with age in both sexes and in all racial and ethnic groups. So, falls are one of the most important problems that hinder these people's ability to have such an independent life, making necessary the presence and monitoring of their daily activities by care-givers.

For this reason, a major research effort has been conducted in the recent years for automatically detecting persons' falls especially for the elderly. The most common way for detecting persons' falls is through the use of specialized devices, such as accelerometers, floor vibration, combination of accelerometers with barometric pressure, wearable equipment, gyroscope sensors, or combination/fusion of them.

A more research challenging alternative is the use of visual cameras. In computer vision society, such identification is a prime research issue due to the complexity of the problem as far as the visual content is concerned. For instance, the algorithm should ideally,

a) detect falls in real time (or at least just in time), i.e., without losing the resolution accuracy for the fall detection,
b) be robust to background changes and illumination changes,
c) be robust when more than one person are present in the scene,
d) identify falls occurring in any position with respect to the camera and
e) be tolerant to camera changes (active cameras).

1.1 Thesis definition

As mentioned above, the most common way for detecting persons' falls is through the use of specialized devices. However, such approaches are device dependent and present a series of drawbacks; they prevent the elderly or cognitive disable persons from being normally function as all we do in our lives since they impose them to wear specialized devices. To overcome aforementioned drawbacks, intelligent vision-based systems have been used, by which, person no longer needs to wear anything and all detection work can be done by cameras and computers. Actually, vision-based systems can give semantic information of a person's actions and at the same time they preserve its privacy.

So, the main purpose of this work was the implementation of a fast real time computer vision algorithm able to detect humans' falls in complex dynamically changing visual conditions. This algorithm has to exploit single cameras of low cost while it will require minimal computational cost and memory requirements.

The completion of this work followed the stages below:

1. Collect and study publications related to machine vision in combination with humans' fall detection problem, in order to examine and understand available techniques and the way they can be applied on this problem.
2. Select appropriate tools and define the framework under which the algorithm will be developed.
3. Gradual development of the algorithm and continuous monitoring its results, in order to fulfill the requirements of the problem.
4. Evaluate the algorithm's results regarding robustness and real time operation.

1.2 Thesis organization

The rest of this master thesis is organized as follows. Chapter II describes the problem of humans' fall and its effects on human health. Chapter III reviews related work. Chapter IV describes the way this problem is approached by this work. Chapter V presents the tools and the framework under which the algorithm was developed. Chapter VI evaluates the algorithm and its results regarding robustness and real time operation. And, finally, chapter VII concludes this work.

Chapter 2

Humans' Fall Problem

The elderly constitute the majority of world population. According to demographic and epidemiological data, the number of people over 65 years old is increasing six times faster than the rest population on earth. The quality of life of elderly is depended on ensuring these people's ability to have an independent life with proper functionality and autonomy.

Emergency department visits related to falls are more common in children less than five years of age and adults 65 years of age and older. Compared with children, elderly persons who fall are ten times more likely to be hospitalized and eight times more likely to die as the result of a fall(J.W. Runge 1993). So, falls can be regarded as an important factor in the health limitations of this age group. High morbidity can be caused by physical and psychological consequences of fall, loss of independence.

Elderly patients who have fallen should undergo a thorough evaluation. Determining and treating the underlying cause of a fall can return patients to baseline function and reduce the risk of recurrent falls. These measures can have a substantial impact on the morbidity and mortality of falls.

2.1 Fall statistics

The first thing someone needs to examine when tries to understand the elderly fall problem is why and where a fall event occurs. Regarding the first question, why elderly fall, the root cause for many fall incidents are believed to be the effects of drugs and the difficulty surrounding medication compliance. Poor health of elderly is another common reason of fall incidents. Elderly people are not as well able to save themselves from a fall as younger people. They are less stable with slower reflexes and are therefore less effective in preventing falls. Moreover, in older people, a fall may be a nonspecific presenting sign of many acute illnesses, such as pneumonia, urinary tract infection or myocardial infarction, or it may be the sign of acute exacerbation of a chronic disease(D.W. Rabin 1995). However, we can't reject that many fall incidents caused by accident. The diagram 1(a) summarizes the most common reasons of fall incidents.

Answering the second question, where a fall event occurs, helps in preventing fall incidents. A recent study, which conducted by Yale University, about elderly fall problem, shows that 55% of all fall incidents take place inside the home and more than three-quarters take place either inside or in close proximity to the home, where a medical alert system can be of immediate assistance.

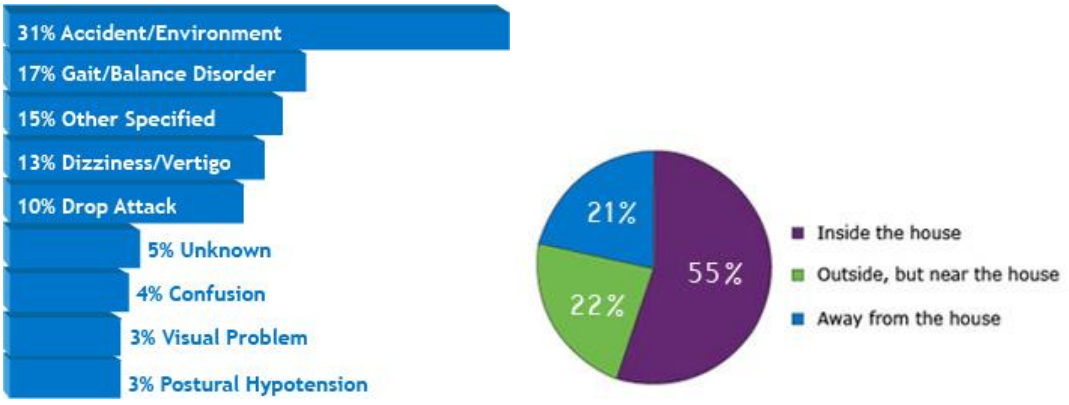


Diagram 1: (a) the most common reasons of a fall incident. (b) places where a fall event occurs

By examining the risks of elderly falls, it's easy for someone to understand the importance of this problem. About one third of the elder population over the age of 65 falls every year. The risk of falls increases proportionately with age and at 80 years, over half of seniors fall annually. Statistical data show that elderly who experienced a fall incident are two to three times more likely to fall again, while about half (53%) of the older adults who are discharged for fall-related hip fractures will experience another fall within six months. Falls account for 25% of all hospital admissions, and 40% of all nursing home admissions 40% of those admitted do not return to independent living, while 25% die within a year. Although, many falls do not result in injuries, a large percentage of non-injured fallers (47%) cannot get up without assistance. This is another serious problem, the elderly who fall and are unable to get up on their own, spend a period of time immobile, which often affects their health outcome. Muscle cell breakdown starts to occur within 30-60 minutes of compression due to falling. Dehydration, pressure sores, hypothermia, and pneumonia are other

complications that may result. The diagram 2, below, summarizes the risks of elderly falls.

Diagram 2: risks of elderly falls

2.2 Fall consequences

As mentioned before, falls are the leading cause of injury-related visits to emergency departments and high morbidity caused by its physical and psychological consequences. Trauma is the fifth leading cause of death in persons more than 65 years old, and falls are responsible for 70% of accidental deaths in persons 75 years of age and older.

Major injuries that can be caused by a fall are laceration with suture, dislocations, sprains and fractures. Fractures and especially hip fractures, which occur at 1.4% of falls, have major impacts on older people's life and the mortality rate during one year follow up a hip fracture is 12%.

Elderly persons who survive a fall experience significant morbidity. Hospital stays are almost twice as long in elderly patients who are hospitalized after a fall than in elderly patients who are admitted for another reason. Compared with elderly persons who do not fall, those who fall experience greater functional decline in activities of daily living and in physical and social activities and they are at greater risk for subsequent institutionalization.

Injuries and fractures are related with the development of fall post syndrome. This syndrome includes a network of physical and psychological factors, among which the fear of another fall and the loss of self confidence. The psychological impact of a fall or near fall often results in a fear of falling and increasing self-restriction of activities. The fear of future falls and subsequent institutionalization often leads to dependence and increasing immobility, followed by functional deficits and a greater risk of falling.

Besides the physical and psychological consequences of falls there are the economical consequences due to the socio-medical attention they generate. In 1996, more than 250.000 older Americans suffered fractured hips, more than 90% of these are associated with falls, at a cost in excess of $10 billion. This incidence is expected to quintuple by the middle of 21th century.

Chapter 3

State of the Art

The humans' fall problem, especially for the elderly, and its social and economical consequences were described in Chapter 2. Because of these consequences the addressing of this problem is concerned as national health priority.

In the recent years, worldwide scientific community has conducted a major research effort in order to address such an important problem. The solution of this problem focuses on automatically detecting persons' falls in order to avoid its unpleasant consequences. A lot of solutions have been proposed and their approach to the problem depends on the specialization and research interests of each research group.

The most important approaches to the humans' fall problem can be divided into two main categories. The first category includes solutions which approach the problem through wearable devices and sensors, like accelerometers, while the second one includes those solutions which approach the problem through signal (image and/or sound) processing with no need of any wearable device.

The rest of this Chapter presents some of the most indicative works that have been conducted on this topic with their advantages and drawbacks.

3.1 Solutions through Sensors

The technological advances in wireless sensor networks enables the development of various applications, which intent to provide appropriate and friendly services based on the recognition of human activities. Such services can concern in detecting humans' falls, in order to prevent and/or reduce the severity of injury in the elderly.

The most common sensors which are employed in human activity recognition are accelerometers and inertial sensors (Shuangquan Wang, Yang et al. 2005; M.N. Nyan, Francis et al. 2008; Thinh M. Le and .Pan 2009; Federico Bianchi, Redmond et al. 2010), which are used to measure the speed and orientation changes of human bodies, atmospheric air pressure sensors (Federico Bianchi, Redmond et al. 2010), by which there is a notion of altitude, and floor vibration sensors (Yaniv Zigel, Litvak et al. 2009), by which there is no need for the human to wear anything.

Most of the approaches above make the main assumption that the patient will be able to wear the device. By relying on this assumption they use the sensors to capture motion data and a processing unit to process these data, in order to detect irrational activities, like a fall. All these approaches propose a system which in its general form can be modeled as shown in the picture below.

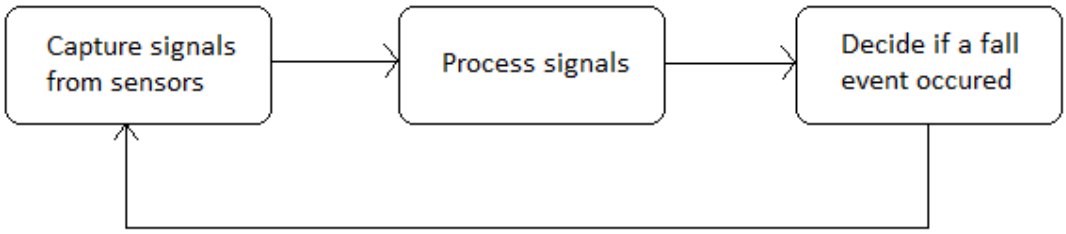


Figure 1: General form of the system, which is proposed by device dependent approaches

3.1.1 Wearable Sensors

The most common sensors that used in fall detection problem are accelerometers. Some indicative works which try to address this problem through the usage of accelerometers are described below.

The system that proposed by the authors of (Thinh M. Le and .Pan 2009) is trying to detect a fall event through acceleration breaching and posture analysis. The operation of this system is divided into two functional blocks. The first one includes a set of sensors (accelerometers) and the second one includes a processing unit consisting of a home based PC and a microcontroller.

Movements and postures of a person are captured by several miniature sensors which mainly sense the acceleration at a particular location in 3D space and transmit the data to microcontroller. Data are furthered preprocessed by the microcontroller before being sent to a home based PC. Finally, the home based PC integrates all the data gathered from different sensors and decide if a fall event occurred.

A few custom wearable sensors (accelerometers) are mounted on different part of a human body to form a body-area network. The sensor board is designed and developed for easy mounting on human body using an elastic belt. The user is equipped with two sensor modules. The first one is placed on the trunk while the other

on the thigh bone. The two sensor modules operate independently and their orientation is determined based on some observations, assuming the three basic postures of standing, sitting and lying.

The two key terminologies used in this fall detection algorithm are acceleration breaching and posture analysis. The algorithm starts from detecting the sudden acceleration changes within a time window. This alerts the system about any abnormal acceleration. After that, user's body posture will be analyzed and determined, it is assumed that user's posture after a fall is perpendicular to the up-right (standing) position. If the body posture suggests that a fall event occurred, system will alert the next-of-kin, care giver, or hospital staff so that further action can be taken immediately.

This system is able to detect acceleration threshold breaching along each axis and recognize the breaching signal from a false alarm. By combining user's body posture it able to confirm if a fall occurred.

Another system which provides rapid detection of falls in order to minimize the negative effects of a fall and limit the detrimental impact of the "long lie" scenario was proposed by (Federico Bianchi, Redmond et al. 2010). Several studies have analyzed accelerometer data to detect falls; however, most suffer from an unacceptably large false positive rate, since there is no notion of altitude associated with the accelerometer signals. It is expected that the altitude-related information acquired by an atmospheric air pressure sensor might provide some utility in improving more traditional accelerometer base falls detection systems.

So, this study evaluates a falls detection algorithm based on signals from tri-axial accelerometer and atmospheric air pressure sensors incorporated into a wireless wearable device that was worn on the subject's belt, aligned with the right anterior iliac crest of the pelvis and measures the barometric pressure and the accelerations relative to the trunk along the vertical, mediolateral and anteroposterior directions. By atmospheric air pressure sensors there is a notion of altitude which is combined with the accelerometer signals.

The fall detection algorithm proposed in this study was obtained by augmenting the algorithm proposed by (Thinh M. Le and .Pan 2009) to consider the barometric

pressure measurement as a surrogate estimation of the change in altitude associated with a fall. It is assumed that when a fall occurs, the altitude of the device placed on the subject's waist level changes. So, this algorithm considers an extreme impact that estimates the degree of movement intensity and can be obtained from the signal magnitude vector which is derived by the barometric accelerometer, the postural orientation of the wearer and changes in altitude associated with a fall.

In order to investigate and evaluate the implemented fall detection algorithm three different experimental protocols were conducted. The first protocol comprises indoor simulated movements and falls; the second protocol comprises indoor and outdoor simulated falls; and third protocol comprises indoor and outdoor simulation of normal activities of daily living. Movements and falls were simulated by a young pool of subjects. The experimental result shown that this algorithm is able to correct a lot of false positives compared to the algorithm proposed by (Thinh M. Le and .Pan 2009), especially when recognizing falls with recovery and falls with attempt to break the fall.

The authors of (M.N. Nyan, Francis et al. 2008) developed a wearable system, by exploiting the correlation of the movement of body segments, in order to prevent or reduce the severity of injury in the elderly by detecting the fall in its descending phase before the impact. So, the key point on this approach is lead-time before the impact, that falls can be distinguished from activities of daily living. This approach is using inertial sensors to detect faint falls in its incipience and is based on the characteristics of angular movements of the thigh and torso segments in falls and activities of daily living.

The hardware setup that was used by this system includes a thigh sensor, a waist sensor set and a data processing unit. During the process flow of pre-impact detection algorithm, acceleration samples are transformed into two dimensional degrees of body orientation, measuring how many degrees these body segments deviate from the vertical axis. If this deviation from the vertical axis intersects the predefined threshold levels, then a fall is confirmed.

Experimental results shown that this system is able to detect a fall in its incipience by exploiting correlated movement of thigh and torso, but the most important thing is that it achieved a lead-time of 700ms, which is the longest lead-time obtained so far.

All the studies described above, approach the problem of humans' fall through the usage of wearable devices. The experimental results from the usage of such devices are promising as they can detect fall events with low miss ratio and distinguish fall events from normal everyday activities, like walking, crawling, running, lying. Another advantage of these approaches is that they respect the personal life of the patient, as sensors can't provide any information about the way the patient lives. Apart from the advantages, there are two significant disadvantages. Firstly, they require that the patient will be able and willing to wear such a device, something that is not sure especially for the elderly or cognitive disable persons. And, secondly, these devices must be worn all day long, which is impossible for the total of activities of daily living.

3.1.2 Non-Wearable Sensors

Another method for automatic fall detection is presented in (Yaniv Zigel, Litvak et al. 2009). The authors developed a solution which is based on the combination of floor vibrations and sound detection during a fall. By this approach the subject doesn't require to wear anything.

This study uses the combination of vibration and sound sensors, because they can supply the information about the way the fall vibrates the floor and how it sounds. The main hypothesis is that someone can accurately identify human falls and discriminate them from other events using a sound and floor vibration detection in conjunction with advanced signal processing techniques. The goal was to develop a method, which consists of an algorithm for event detection and classification. This algorithm, which is based on pattern recognition techniques, enables to distinguish between a simulated human fall event and other events such as fall of an object on the floor.

The proposed algorithm contains the training phase and the testing phase of data analysis. Both phases use vibration and sound signals as inputs. In order to trigger the classification algorithm, there must be a significant event in the vibration signal. Once an event is detected in the vibration signal, the sound signal is analyzed. In the training phase of the algorithm two different classes for two different types of events were estimated. These classes are the "fall" class and the "other event" class.

After the training phase the testing phase was following. Simulated falls using a human mimicking doll were performed and the values of the vibration and sound event signals are used as input for the estimated classifier. The calculation of the maximum likelihood was performed, and the classifier returned the classification result whether the event is “fall” (positive) or it is “other event” (negative).

Experimental results showed the significant drawbacks of this system. This system can’t succeed without audio information which is prone to be contaminated by other noises. This was shown by trials which were performed in different days, and the noise level of the system that was different in those days influenced its performance. Besides this, the distance from the sensors can influence the robustness of the system, as events occur close to the sensors generate false positive events while events occur away from the sensors, for distance bigger than 5 meters, can’t be detected. Finally, another limitation is that it may not be sensitive to low-impact real human falls.

3.2 Solutions through Visual Cameras

In the previous section studies that approach the humans’ falls detection problem through sensors were described. Such approaches are device dependent and present a series of drawbacks. For this reason, a more challenging research alternative approach has been developed that tries to address this problem through the use of visual cameras.

Vision-based systems present several advantages as they are less intrusive because installed on building (not worn by users), they are able to detect multiple events simultaneously and the recorded video can be used for post verification and analysis.

The workflow of an algorithm, which tries to address this problem through visual cameras, contains three basic stages. On the first stage background/foreground subtraction must be implemented, in order to detect and track the human, on the second stage motion and/or posture analysis is carried out and information about daily

activities is obtained. And finally, on the third stage the algorithm by using this information, is trying to detect abnormal events like falls.

A vision-based system which approaches the fall problem is described in (G. Diraco, Leone et al. 2010). This study presents a framework for the monitoring of ageing people, including fall detection capabilities by using a self-calibrated vision-based system without any user intervention. For this purpose a Time-Of-Flight 3D camera is employed, which provides a depth map, a gray-level image and the 3D coordinates of the points cloud representing the acquired scene.

Concerning background subtraction, the Mixture of Gaussians (MoGs) method was implemented using raw depth information since it is not sensitive to illumination changes or shadows making moving blobs extraction more easily. Once background has been subtracted and people silhouette has been detected in the 3D world its center of mass is being computed and tracked.

A fall event is detected by threshold the distance *h* of the 3D centroid from the floor plane, which defined by the horizontal planes in the scene that are filtered according to appropriate constraints. In general a fall event is characterized by:

1. a distance of the centroid from the floor plane (the feature *h*) lower than a prefixed value,
2. an unchangeable situation for at least 4 seconds.

Although this proposed method is simple and it performs a low cost algorithm it presents a series of drawbacks that affects the robustness of the system. Firstly, it doesn't take into account the orientation of motion of the moving blob, and secondly the measures that are provided by the camera could be affected by reflectivity objects properties and aliasing effects when the camera-target distance overcomes the non-ambiguity range.

Finally it's worth noting that this work proposes a preliminary method on posture recognition which may help in humans' fall problem. This method proposed according to a 3D hybrid approach in which model-based and learning-based techniques are used in conjunction in order to estimate location and orientation of the different body articulations. A Discrete Reeb Graph is used for body skeleton extraction and the Geodesic distance is used since it provides a better segmentation of

the human body. Although this preliminary method needs more upcoming work to be done in order to be applicable in real problems, it's an innovative approach to posture recognition through visual cues.

Another work that approaches the fall detection problem is (Homa Foroughi, Rezvanian et al. 2008). This work proposes a method for monitoring human posture-based events in a home environment with focus on detecting a fall.

After background subtraction, in order to analyze the motion occurring in a given window of time, the feature extraction process for the foreground object is following. The feature extraction process is based on the changes in the human shape, since these changes can provide the information needed to discriminate if the detected motion is normal or abnormal.

The extracted features are the combination of Approximated ellipse, horizontal and vertical projection histograms and temporal changes of human head pose. The moving person is approximated by an ellipse, which is defined by its center, its orientation and its length. This representation of a person is rich enough to support recognition of different events, while it is coarse enough to enable a wide range of body poses and clothing to be tracked. Experimental results showed that the approximated ellipse is completely distinctive for different postures. The shape of a 2D binary silhouette can be represented by its projection histogram. Horizontal and vertical projection histograms can provide information about the relationship between width and height of the moving object. Finally, the temporal changes of head position are estimated. Head pose is a potentially powerful and intuitive pointing cue if it can be obtained accurately and noninvasively. To localize the head of the person, the top-most detected point of the silhouette is marked and the movement features are directly calculated from the curvature extrema in these trajectories and are invariant to several changes in viewpoint. By using appropriate threshold for the vertical displacement of head position the system can distinguish different behaviors.

After features extraction, motion information that is described by the features is applied in multi-class support vector machine in order to determine if a fall event was occurred. Figure 2 shows the flow of the proposed system.

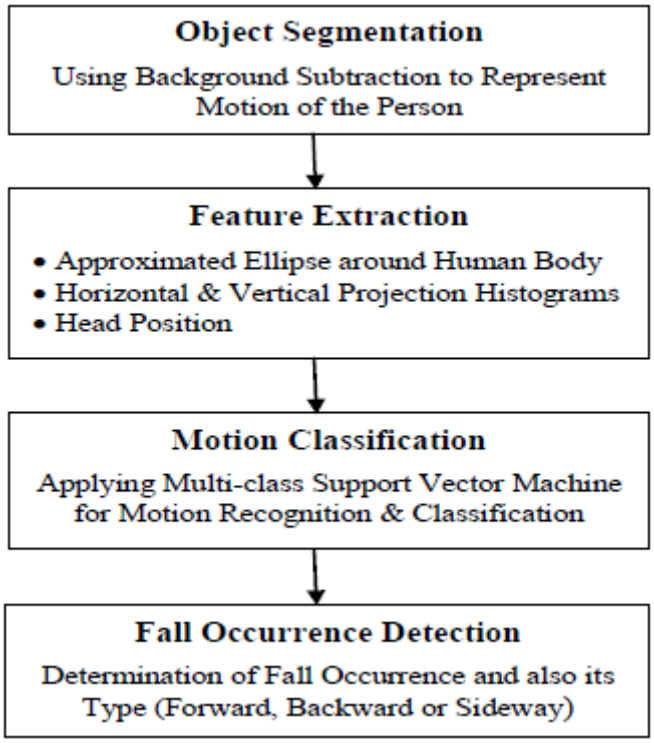


Figure 2: Proposed System Flow

Although this system is able to detect type of fall incident and not just a fall event, and its recognition rate is 88%, it presents several drawbacks. Firstly, it assumes that the whole body of the subject will be in the field of view of the camera, in order to be able to recognize human postures without ambiguities. And secondly, it assumes that the top-most detected point of the silhouette is the head. Both assumptions above can't be applied in the real world thus efficiency and robustness of this system, under real circumstances, are unknown.

The study of (Huimin Qian, Mao et al. 2008) mainly deals with detecting the activity of falling down taking place in home environment. This method considers six kinds of indoor activities generally appeared in daily life, walk, jogging, sitting down, squatting, falling down and immobility, among which falling down is taken as an abnormal activity.

In the proposed activity recognition and classification system, two phases of processes are performed. First, the human blobs are detected using a background subtraction method based on non-parameter estimation technique. Then, the features of the

detected human blobs are extracted and used as input into a combined classifier to discriminate the activity of falling down.

Feature extraction process is based on two minimum bounded boxes. From anatomy, it can be found that each part of the human body occupies an almost fixed percentage in length relative to body height. So, two minimum bounding boxes are defined. The first one contains the whole human body, while the second one contains the lower part of human body with fixed height 5 times smaller than the total human body height. For the first bounding box two measures are defined, the first one is the height of human body and the second one is the aspect ratio of height versus width of human body. For the second bounding box the width increment during a time window is defined. These measures are used as input into the classifier.

In the beginning the algorithm classifies the activities between the mobility and immobility classes and then classifies the activities of the immobility class in order to discriminate a fall event.

Experimental results showed that the incomplete human blob from the background subtraction doesn't influence the performance of the system owing to the feature extraction from the two minimum bounding boxes of the human blobs. However, in the case that most of the human blobs cannot be detected for a long time, the incorrect feature vector will affect the identification rate. In addition, the angle between the activity and the camera mounted on the side wall of the room cannot be too big and finally, if the camera's field of view doesn't contain the whole human body, the assumption derived from anatomy is not applicable.

The authors of (Chia-Wen Lin, Ling et al. 2005) propose a vision based compressed-domain fall detection scheme for intelligent home surveillance applications. The proposed scheme can detect and track moving objects from incoming compressed video in the compressed domain, without the need of decoding the incoming video into pixel values. In addition to the motion information, a DC+2AC image is used to perform change detection and/or background subtraction to refine the object mask. After detecting the moving objects, compressed domain features of each object are then extracted for identifying and locating fall incident.

To identify and locate a fall down event, three features are used. Firstly, the duration of the events is calculated, as a fall down event usually occurs in a short time period, secondly, the person's centroid changes are estimated and finally the change rate of vertical projection histogram is used, as this is a useful feature for detecting a fall down event, because a standing object will have different vertical projection histogram values than a falling down one.

The main drawback of this system is that the person's centroid estimation can be significantly affected as a person falls, because the detecting region shrinks.

Another approach to humans' fall detection problem is presented by the authors of (Zhengming Fu, Culurciello et al. 2008). This study describes a fall detector using an asynchronous temporal contrast vision sensor. The detector takes multiple side views of a scene in order to detect accidental activities and raise alarms.

A temporal contrast vision sensor extracts changing pixels from the background and reports temporal contrast, which is equivalent to image reflectance change when lighting is constant. A temporal contrast vision sensor can extract motion information because, in normal lighting conditions, the intensity of a significant number of pixels changes as an object moves in the scene. When the change of light intensity in a pixel passes a threshold, an event is triggered.

In order to estimate if a fall event occurs in the scene, firstly, centroids are computed as moving averages of a series of events as centroids are an effective way to estimate object motion in space. Then, based on the assumption that a faster motion generates more events during a time period, the event rate is used for characterizing the motion in the scene. In order to characterize a motion as a fall the system checks the number of events per second and vertical centroid velocity of the moving object.

The main advantages of this method is that an asynchronous temporal contrast vision sensor features very high read-out speed and reports pixel changes with a latency on the order of millisecond. Beside this, the sensor pushes information to the receiver once the predefined condition is satisfied, which leads to a lightweight computation algorithm that permits to compute an instantaneous motion vector and report fall events.

On the other hand, a temporal contrast vision sensor depends on image reflectance changes and requires constant lighting conditions to operate properly. In addition, in this work, centroids are estimated and then the vertical centroid velocity is used for a fall alert. However, as a person falls the detecting region shrinks and this significantly affects the position of the centroid and therefore the performance of the system. Finally, possible erroneous estimation of moving objects may confuse the fall detection sensor deteriorating its performance.

The paper of (Doulamis 2010) presents a visual fall detection system, which is able to detect person falls by taking into consideration only camera information. The system is able to perform tracking of the person using advanced image processing computer based algorithms in complex and dynamic background situations.

The innovation of this work is that motion is detected only on pre-determined points of interest to reduce the noise. Each pixel is self-timed and responds to relative changes in luminosity and motion field estimation approximates the motion information of the scene. Based on motion information foreground object is estimated. It is expected that in case of significant motion information, the foreground object is accurately estimated (e.g., from the motion cues) rather in cases of no motion presence. For this reason, the accurate motion vectors are incorporated with the foreground detection algorithm to stress the reliability of the foreground detection. The reliability of the foreground is also estimated through the application of shape and time constraints. The aspect ratio of a human object's height and width represents the shape constraints, while the time constraints requires that the detected foreground mask should not be significantly changes through time. After accurate foreground estimation, a fall event can be detected by monitoring the time derivative of the height of the foreground object.

3.3 Summary

With the population growing older and increasing number of people living alone, supportive home environments able to automatically monitor human activities are

likely to widespread due to their promising ability of helping elderly people. A lot of researches focus on the detection of fall accident for the elderly, and current solutions to detect falls can be categorized into two main categories. The first category includes wearable sensors approaches. These autonomous sensors are attached on the human body and have gyroscopes or accelerometers embedded in them and detect velocity or accelerate exceed a specific threshold, vertical posture toward lying posture, and also absence of movement after fall. However, their efficiency relies on the person's ability and willingness to wear them. The second category includes more research challenging approaches based on computer vision systems. These approaches try to extract some considerable features from video sequences of movement patterns to detect falls.

Chapter 4

Our approach

In this thesis, a real-time computer vision–based system capable of automatically detecting falls of elderly persons in rooms, using a single camera of low cost and thus low resolution is proposed. This system is automatically reconfigured to any background changes which cover not only lighting modifications but also texture alterations, and thus it is able to identify humans' falls in complex, dynamic in terms of background visual content, and unexpected environments, like the ones encountered in real-world clinical and/or home conditions. The primary goal of this work was the implementation of a robust and efficient algorithm with low computational cost, which can be used in large scale implementations and in real world conditions.

Implementation process followed three main steps:

i. Background modeling and subtraction
ii. Foreground features extraction
iii. Fall detection algorithm implementation

4.1 Background Modeling

In general background subtraction techniques are not so simple in computational cost and memory and they fail for a large scale implementation. For this reason, in this work, the intensity of motion vectors along with their directions is exploited to identify humans' movements. However, motion vectors are still very sensitive to luminosity changes and color/camera parameter variations. For instance, a different focus of the camera, which is a continuous usual process of the current camera sensors, may result in estimating of large intensity values of motion vectors, though no motion information is encountered in the captured image frames. For this reason, initially we apply the methodology of (Jianbo Shi and Tomasi 1994) so as to detect corners, edges and other salient points on video frames and these points to be considered as "good" locations for estimating motion vectors.

Before the description of background modeling method, let's see what a good feature is and how the method of (Jianbo Shi and Tomasi 1994) is working. A good feature is an image point which can be tracked in subsequent frames. In practice, this point should be unique, or nearly unique, and should be parameterizable in such way that it can be compared to other points in another image. These points are characterized by their derivatives. A point to which a strong derivative is associated may be on an edge of some kind, but it could look like all of the other points along the same edge. But, if strong derivatives are observed in two orthogonal directions then there is a strong indication that this point is more likely to be unique. These trackable points are called corners and contain enough information to be picked out from one frame to another.

The most commonly used definition of a corner was provided by (Chris Harris and Stephens 1988). This definition relies on the matrix of the second-order derivatives of the image intensities. According to this definition corners are places in the image where the autocorrelation matrix of the second derivatives has two large eigenvalues. This autocorrelation matrix is defined as follows:

$$A(x) = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2(k) & I_x I_y(k) \\ I_x I_y(k) & I_y^2(k) \end{bmatrix}$$

where I_x and I_y are the respective derivatives (of pixel intensity) in the x and y direction at point k and $w(x,y)$ is a weighting term that can be uniform but is often used to create a circular window or Gaussian weighting. This definition has the advantage that eigenvalues of the autocorrelation matrix are invariant to rotation, which is important because objects that are tracked might rotate as well as move.

Based on the work of (Chris Harris and Stephens 1988) Shi and Tomasi (Jianbo Shi and Tomasi 1994) found that good corners resulted as long as the smaller of the two eigenvalues of the autocorrelation matrix was greater than a minimum threshold.

As mentioned before, in this work, the intensity of motion vectors along with their directions is exploited to identify humans' movements and subtract the background. There are two methods for identifying motion vectors in a scene. The first one associates some kind of velocity with each pixel in the frame or, equivalently, some displacement that represents the distance a pixel has moved between the previous

frame and the current frame and is usually referred to as *dense optical flow*. The second relies on some means of specifying beforehand the subset of points that are to be tracked, such as corners. For many practical applications, the computational cost of sparse tracking is so much less than dense tracking that the latter is relegated to only academic interest.

In this work the "pyramidal" Lucas-Kanade algorithm was used to identify motion vectors. The Lucas-Kanade algorithm (Bruce D. Lucas and Kanade 1981) can be applied in a sparse context because it relies only on local information that is derived from some small window surrounding each of the points of interest. One significant disadvantage of using small local windows in Lucas-Kanade is that large motions can move points outside of the local window and thus become impossible for the algorithm to find. This problem led to development of the "pyramidal" Lucas-Kanade algorithm, which tracks starting from highest level of an image pyramid (lowest detail) and working down to lower levels (finer detail). Tracking over image pyramids allows large motions to be caught by local windows.

In order to identify motion vectors, the Lucas-Kanade algorithm uses small windows around pixels of interest and rests on three assumptions below.

1. *Brightness constancy*. A pixel from the image of an object in the scene does not change in appearance as it (possibly) moves from frame to frame. For grayscale images, this means that the brightness of a pixel does not change as it is tracked from frame to frame.
2. *Temporal persistence or "small movements".* The image motion of a surface patch changes slowly in time. In practice, this means the temporal increments are fast enough relative to the scale of motion in the image that the object does not move much from frame to frame.
3. *Spatial coherence*. Neighboring points in a scene belong to the same surface, have similar motion, and project to nearby points on the image plane.

By using small windows the algorithm can't catch large motions; while a large window too oft en breaks the coherent motion assumption. To circumvent this problem, a variation of the main algorithm can be applied according which the algorithm track first over larger spatial scales using an image pyramid and then refine the initial motion velocity assumptions by working its way down the levels of the

image pyramid until it arrives at the raw image pixels. The "pyramidal" Lucas-Kanade algorithm is a coarse-to-fine optical flow estimation and described by the figure below.

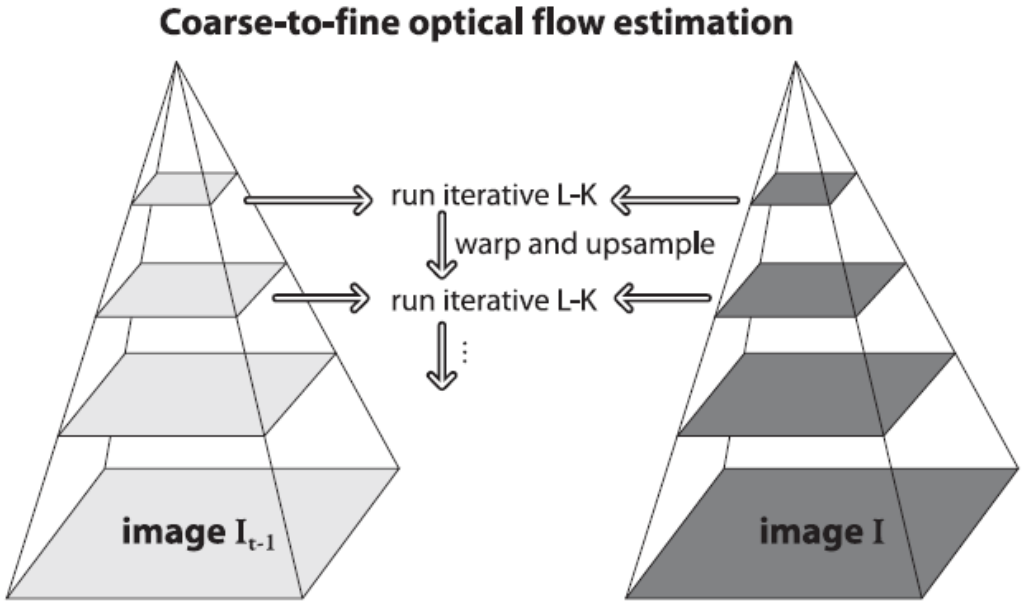


Figure 3: Pyramid Lucas-Kanade optical flow

Motion vectors estimation is followed by the creation of a binary mask in order to indicate areas of high motion information and good features to track. The initially detected feature points are spatially sampled by retaining the local maximum feature points, within a neighbor region. And, finally, morphological operators (erosion, dilation) are applied to clarify the results from noise.

Motion information is used as background updating stimuli. Based on the assumption that high motion activity areas belong to foreground region, regions which are spatially far away from the motion activity segments are selected to be background areas. More specifically, the background is updated at every frame instance. Initially the intensity of motion activity is estimated. If this intensity is greater than a threshold value and the motion is outside the foreground area, the background values should be updated, since a foreground object was appeared and covered parts of the background. Otherwise there is no important variation in the scene which imposes that there is no need for background updating.

When background updating should take place, the detected regions are filtered so that the ones that are far away from the estimated moving regions to be selected as new background while the ones that are close to the foreground objects to be considered as ambiguous regions. These regions are defined by a rectangular that includes the left-right, upper-down extreme boundary motion vectors locations. The workflow for background modeling and updating is shown in the figure below.

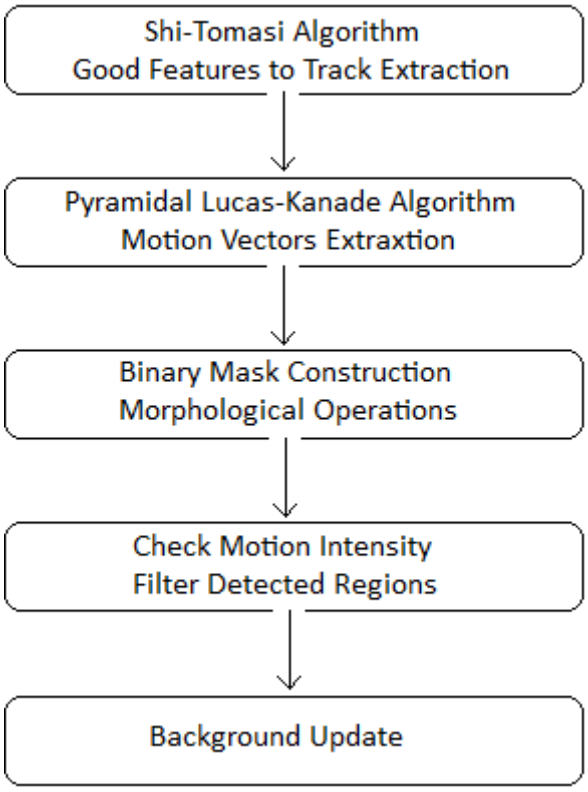


Figure 4: the workflow for background modeling and updating

4.2 Foreground Features Extraction

After background estimation and foreground extraction, foreground feature extraction is following. Foreground features are used as input into fall detection algorithm in order to establish if a fall event occurs. The features that are used in this works are:

i. The top-most detected point of the foreground.
ii. The width the foreground object.
iii. The real height of the foreground object in 3D space.

The first feature is used to establish the vertical motion velocity of the foreground object; the second one is used to check the width-height ratio of the foreground object. There is the assumption that this ratio is bigger in value when a fall event occurs than the same ratio with the foreground object in standing position. The third one is used to check the changes of foreground object's height during activities of daily living.

In order to establish the top-most point and the width of the foreground object, a bounding box, which includes the foreground, is created. The top-most point of the foreground corresponds to the top-most point of the bounding box, while the width can be calculated using the right-most and left-most point of the same box.

The real height of the foreground object in 3D space is a little more difficult process. By using a single camera stereo vision mathematics can't be applied. However, by using a calibrated camera and inverse perspective mapping, the real height estimation can be established.

Let's see the process from the beginning. Firstly, the camera has to be calibrated so as to remove lens distortions and manufacturing defects. Camera calibration is important for relating camera measurements with measurements in the real, three-dimensional world. This is important because scenes are not only three-dimensional; they are also physical spaces with physical units. Hence, the relation between the camera's natural units (pixels) and the units of the physical world (e.g., meters) is a critical component in any attempt to find the dimensions of an object in a three dimensional scene.

The process of camera calibration gives both a model of the camera's geometry and *distortion* model of the lens. These two informational models define the *intrinsic parameters* of the camera. In order to understand how a camera can be calibrated, firstly camera models and the causes of lens distortion have to be described.

The simplest model of a camera is the pinhole camera. By using a pinhole camera, the image on the image plane (also called the projective plane) is always in focus, and the size of the image relative to the distant object is given by a single parameter of the camera: its *focal length*. This is shown in figure below, where f is the focal length of the camera, Z is the distance from the camera to the object, X is the length of the object, and x is the object's image on the imaging plane.

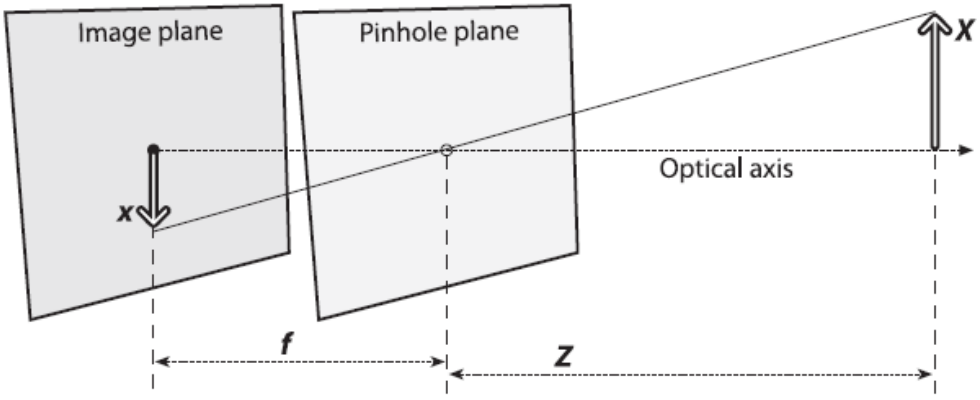


Figure 5: pinhole camera model

By using similar triangles *x* can be calculated by:

$$-x = f\frac{X}{Z}$$

The point in the pinhole, the center of the pinhole plane, is interpreted as the *center of projection* and the point at the intersection of the image plane and the optical axis is referred to as the *principal point*. If someone uses absolute values for the measures above can the negative sign of *x* on the imaging plane is gone.

Pinhole camera is an ideal model of camera. For cameras used in real world applications, due to manufacturing defects, the principle point is not equivalent to the center of the imager, because the center of the chip is usually not on the optical axis. Thus two new parameters, c_x and c_y, have to be introduced to model a possible displacement (away from the optic axis) of the center of coordinates on the projection screen. The result is that a relatively simple model in which a point *Q* in the physical world, whose coordinates are (*X, Y, Z*), is projected onto the screen at some pixel location given by (x_{screen}, y_{screen}) in accordance with the following equations:

$$x_{screen} = f_x\left(\frac{X}{Z}\right) + c_x\,, \qquad y_{screen} = f_y\left(\frac{Y}{Z}\right) + c_y \qquad (1)$$

Two different focal lengths have been introduced and the reason for this is that the individual pixels on a typical low-cost imager are rectangular rather than square. The focal length f_x (for example) is actually the product of the physical focal length of the

lens and the size s_x of the individual imager elements (this should make sense because f_x has units of pixels per millimeter while *F* has units of millimeters, which means that f_x is in the required units of pixels). Of course, similar statements hold for f_y and s_y.

The relation that maps the points *Q* in the physical world with coordinates (*X, Y, Z*) to the points on the projection screen with coordinates (*x, y*) is called a projective transform. When working with such transforms, it is convenient to use what are known as *homogeneous coordinates*. The homogeneous coordinates associated with a point in a projective space of dimension *n* are typically expressed as an (*n* + 1) dimensional vector (e.g., *x, y, z* becomes *x, y, z, w*), with the additional restriction that any two points whose values are proportional are equivalent. The image plane has two dimensions, so points on that plane will be represented as three dimensional vectors $q = (q_1, q_2, q_3)$. This allows us to arrange the parameters that define our camera into a single 3-by-3 matrix, which we will call the *camera intrinsic matrix*. The approach to camera intrinsics is derived by (J. Heikkila and Silven 1997). The projection of the points in the physical world into the camera is now summarized by the following simple form, which is equivalent with equations (1):

$$q = MQ, \quad where\ q = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

By the procedure above intrinsic parameters of the camera can be defined. The other issue that must be figured out is that of lens distortion. In theory, it is possible to define a lens that will introduce no distortions. In practice, however, no lens is perfect. The two main lens distortions are *radial distortions* that arise as a result of the shape of the lens and *tangential distortions* that arise from the assembly process of the camera as a whole.

Let's start with radial distortions. With some lenses, rays farther from the center of the lens are bent more than those closer in, that's the reason why real cameras often noticeably distort the location of pixels near the edges of the imager. The distortion at the (optical) center of the imager is zero and increases as we move toward the periphery. In practice, this distortion is small and can be characterized by the first few terms of a Taylor series expansion around *r=0*. For cheap web cameras, the first two

such terms are enough to characterize the distortion. This way the radial location of a point on the imager will be rescaled according to the following equations:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

Here, (*x, y*) is the original location (on the imager) of the distorted point and ($x_{corrected}$, $y_{corrected}$) is the new location as a result of the correction. The k_3 parameter is the third term of Taylor series and is used for huge radial distortions, like these caused by fish-eye cameras.

The second-largest common distortion is tangential distortion. This distortion is due to manufacturing defects resulting from the lens not being exactly parallel to the imaging plane. Tangential distortion is minimally characterized by two additional parameters, p_1 and p_2, such that:

$$x_{corrected} = x + [2p_1 y + p_2(r^2 + 2x^2)]$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 x]$$

Here parameters, p_1 and $p_2$corresponds to pixel error. So, total there are five distortion coefficients, that can be bundled into one distortion vector; this is just a 5-by-1 matrix containing k_1, k_2, p_1, p_2 and k_3.

In order to compute *intrinsic matrix* and *distortion vector* the method of calibration is to target the camera on a known structure that has many individual and identifiable points (i.e. a chessboard). By viewing this structure from a variety of angles, it is possible to then compute the (relative) location and orientation of the camera at the time of each image as well as the intrinsic and distortion parameters of the camera.

After camera calibration process, the camera that was used in this application can be though as an "ideal" camera. In order to compute the real height of objects through equation (1), the distance *Z* between the camera and the object has to be known. If perspective transformation will be performed then the pixel-distance relationship will be linear and distance *Z* will be easily calculated. In general, the perspective transformation relates two different images that are alternative projections of the same three-dimensional object onto two different *projective planes.* With perspective

transformation the representation of the 3D scene objects being captured are translated into the 2D image plane. This is necessary when the goal of the application is to infer measures in the Euclidean space or in metric units.

In this work the algorithm of (A. Bevilacqua, Gherardi et al. 2008) was used. This algorithm requires a chessboard grid to operate properly and its purpose is to find the inverse projective transformation that maps a set of 3-dimensional points of the real world object (belonging to intersections of the chessboard pattern plane) to the corresponding set of 2-dimensional points of a virtual grid, representing the orthographic view from above of the flat chessboard grid. In our work it is very important as by the orthographic view from above of the floor plane linearity between pixel-distance measures is achieved.

This algorithm starts by detecting the corners of the chessboard grid and then identifies the principal angles of each extracted corner, with respect to the chessboard grid. By principal angles local corner orientation are computed and the extracted set of point coordinates in the *image reference frame* together with their respective PA angles have now to be matched with the corresponding points of a "virtual" 2D grid in the *object reference frame.* By using the virtual grid and the real chessboard grid the perspective projection that can be formalized according (2) can be applied.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (2)$$

Where $H = [h_{ij}]$ represents the homography matrix. By knowing the homography matrix the inverse perspective transformation can be performed for any new captured image.

By perspective transformation all parameters of (1) are known so the real dimensions on the foreground object can be computed. At this point all foreground object features have been extracted and are ready to be used by the fall detection algorithm. So in the following the fall detection algorithm is going to be described.

4.3 Fall Detection Algorithm

As described in the previous section, the fall detection algorithm uses three foreground features as inputs to detect a fall event. These features are:

i. Vertical velocity of foreground object's top-most point.
ii. The ratio of foreground object's width and height (pixel values).
iii. The real height of the foreground object.

Firstly, the algorithm uses the ratio of foreground object's width and height. When a human is standing this ratio is small. If the human changes his posture to sitting or laying which in most cases is the result posture after a fall event, this ratio increases. So, for every captured frame this ratio is checked and compared to a predefined threshold in order to estimate if the posture of the foreground object may be a result of a fall incident.

If the above ratio suggests that a fall event might be occurred, the algorithm uses the other two features to detect a fall event. The vertical velocity of foreground object's top-most point is calculated. This velocity is relative to the vertical shifting of the same point during a sequence of frames. This vertical shifting is compared with a threshold relative to the real height of the foreground object. It has to be mentioned that this vertical shifting is measured in cm and is comparable to the real height of the foreground object. By adopting a threshold relative to the real height of the foreground object and measuring the shifting in cm, the performance of the system is not affected by cases where the foreground object is far away or close to the camera.

The fall detection system overview is shown in Figure 6. Firstly, the camera captures a frame, then background modeling and subtraction is following in order to detect the foreground object. After the foreground detection the foreground features are extracted and used as input into the fall detection algorithm. The algorithm checks the width-height ratio of the foreground object and if this ratio suggests that no fall event occurred, the system is processing the next frame. Otherwise if the ratio suggests a human posture that might be the result of a fall event, the vertical shifting of the foreground object's top-most point is calculated. If the vertical shifting of the foreground object's top-most point is bigger than a threshold then the system detects a fall incident, otherwise the next captured frame is processing.

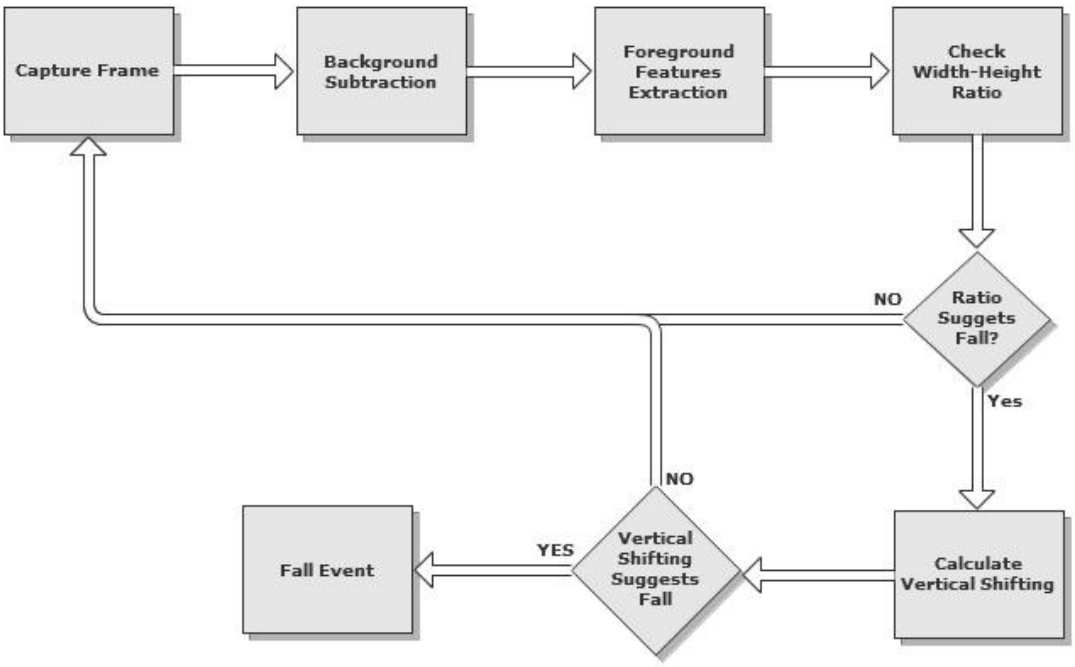


Figure 6: Fall Detection System Overview

Chapter 5

Application Development and Evaluation

The previous chapter describes the way this work approaches the problem of humans' fall detection. The present chapter focuses on presenting the tools (hardware and software) that are used for the development of the application as well as the evaluation of the system.

5.1 Tools

The application was developed on a laptop PC with 4GB Ram and a dual-core Intel processor at 2.1GHz. The camera that was used was a simple Logitech USB webcam with 640x480 pixels resolution.

Although the fall detection application is cross-platform it was developed under Ubuntu 10.04 operating system. The code was written in C by using Intel's OpenCV library.

Before the description of development and evaluation for each part of the system let's see what OpenCV is. OpenCV is an open-source, cross-platform computer vision library which is written in optimized C and C++. OpenCV grew out of an Intel Research initiative to advance CPU-intensive applications. The main goal of OpenCV is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly, this main goal can be divided into three specific goals (Gary Bradski and Kaebler 2008):

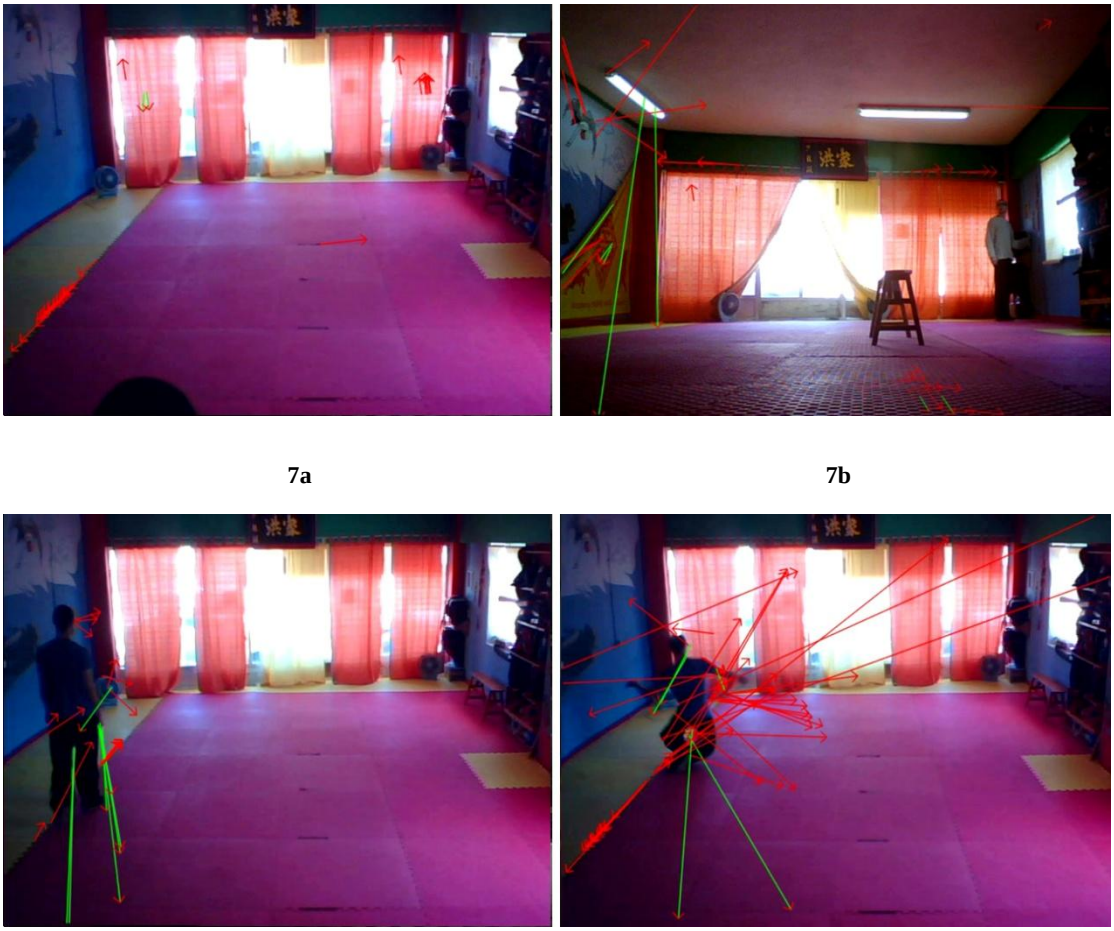
i. Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
ii. Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
iii. Advance vision-based commercial applications by making portable, performance optimized code available for free, with a license that did not require commercial applications to be open or free themselves.

5.2 Background Modeling

The algorithm for background modeling was analytically described in chapter 4. In this chapter evaluation of this algorithm will be done. The main goal of this study is to develop an effective and robust fall detection algorithm that can be applied in real life complicated conditions where the background is dynamically changes. Changes in the background can be caused by changes in lighting conditions or appearance of moving objects, other than humans, in the scene.

This algorithm exploits motion information in order to successfully subtract the background and extract the foreground. Before we examine this algorithm let's see the way motion information is represented by motion vectors. Motion vectors in the scene are represented by red and green arrows and the length of the arrows is associated with the intensity of the motion.

Figure 7a shows small changes in the background. In this example curtains are moving. The length of motion vectors shows that this movement is very small. The next figure (7b) shows what is happening when lighting conditions are changing. In this example the lights are turned on and motion vectors are created close to the lights area, although no real motion is appeared. This motion vectors appearance is caused by the appearance of new shadows in the scene. Finally, figure 7c and 7d show the motion vectors when a human is walking (normal everyday activity) and when a human is falling. In the first case the number of motion vectors is relative small as well as their length, while in the second one the number of motion vectors is increased and the length of these vectors is much bigger than the first case.



7a **7b**

7c **7d**

Figure 7: Motion Vectors

After we know how motion vectors represent motion information in the scene, it's time to examine the background modeling algorithm. It has to be mentioned that this algorithm is responsible for background subtraction as well as foreground extraction.

The background modeling algorithm was tested twice in a martial arts school and once in a demo room in Trikala Municipality. When the application starts the background modeling algorithm assumes that the background is equal to the first captured frame. When subsequent frames are captured the algorithm exploits motion information in order to adapt the background by adding or removing image blocks that are respectively away or close to the image block where high motion activity is appeared.

8a

8b

8c

Figure 8: Background adaptation during time

Figure 8 presents background adaptation process. Images on the left show the captured frame while images on the right show background (black area) and foreground (white area). As anyone can see in figure 8a background in not yet

adapted as it doesn't contain all background areas. A few frames later (figure 8b) a large area of the background has been adapted and only a small area of the background has been signed as foreground. Finally in figure 8c background has been completely adapted and foreground object has been successfully extracted. At this point it has to be mentioned that foreground object's extracted features can be used effectively as inputs into the fall detection algorithm only if background is completely adapted.

In the example above the field of view of the camera includes the whole body of human and main motion activity was constrained in a small image block. The next example (figure 9) shows how the algorithm corresponds when the foreground object is close to the camera and the image block with high motion activity is much bigger.

Figure 9: Background modeling when foreground object is close to the camera

In figure 8 background areas outside the foreground object is successfully detected but inside the foreground area there are small black background areas. Although this background modeling and foreground extraction can be though as a successive one as all needed foreground features can be extracted and effectively used as inputs into the fall detection algorithm.

In the previous examples the algorithm was tested with static background. Although this application has to be robust and effective when is applied in real conditions where background is dynamically changes. Dynamic background changes can be caused by two reasons:

i. lighting conditions changes and
ii. moving objects appearance in the scene

The following examples show how the algorithm corresponds to dynamically changing background. In figure 10 the lights are turned on and lighting conditions are changed. Figure 10a shows the background and foreground areas the moment when the lights are turned on, while figure 10b shows background adaptation after a few frames. In figure 11 curtains of the room are opened. This causes lighting conditions changes and as curtains move, moving objects appeared in the scene. Figure 11a shows the background and foreground areas the moment when curtains are opened, while figure 11b show background adaptation to new lighting and spatial conditions.

10a

10b

Figure 10: Background adaptation to new lighting conditions

When the lights are turned on high motion activity was appeared in the image area that includes the lights. Due to this high motion activity, in figure 10a, foreground

areas are created close to the lights. After a few frames and while the “real” foreground object is moving in the scene, so high motion activity area is moving away from the lights area, the background is successfully adapted(figure 10b).

In the next example, exactly for the same reasons, when curtains are moved, they are sighed as foreground object (figure 11a). After a few frames the background has been completely adapted to new lighting and spatial conditions (figure 11b)

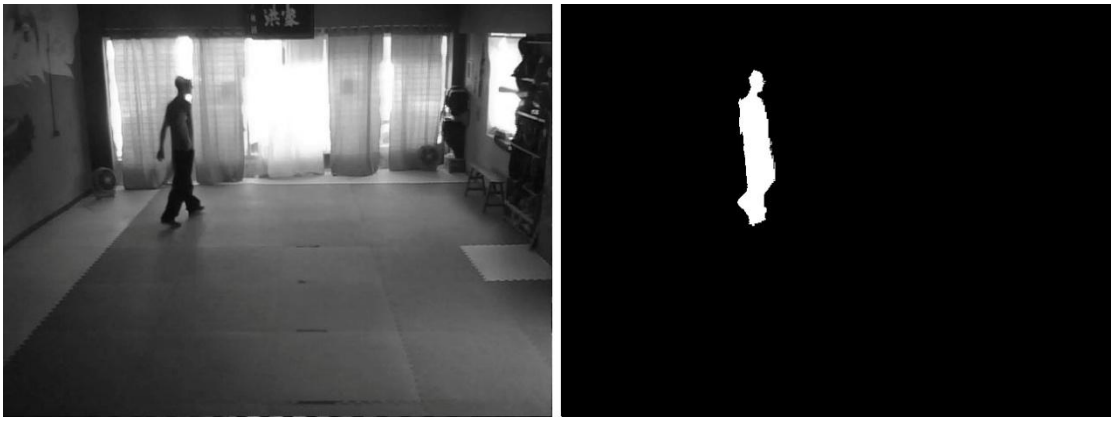
11a

11b

Figure 11: Background adaptation to new lighting and spatial conditions

The next example shows how the algorithm corresponds to continuous changes in the background where small object movements are appeared in the scene due to physical phenomena, like air blowing.

12a



12b

Figure 12: Background adaptation when the background is changing continuously

This explains the appearance of white areas in the right image of figure 12a. A few frames later human has walked in the scene and high motion activity areas have been changed. In this figure the slight movement of curtains creates very low intensity motion compared to the motion of human, so the area of curtain is not signed as foreground and the algorithm successfully extracts the foreground object.

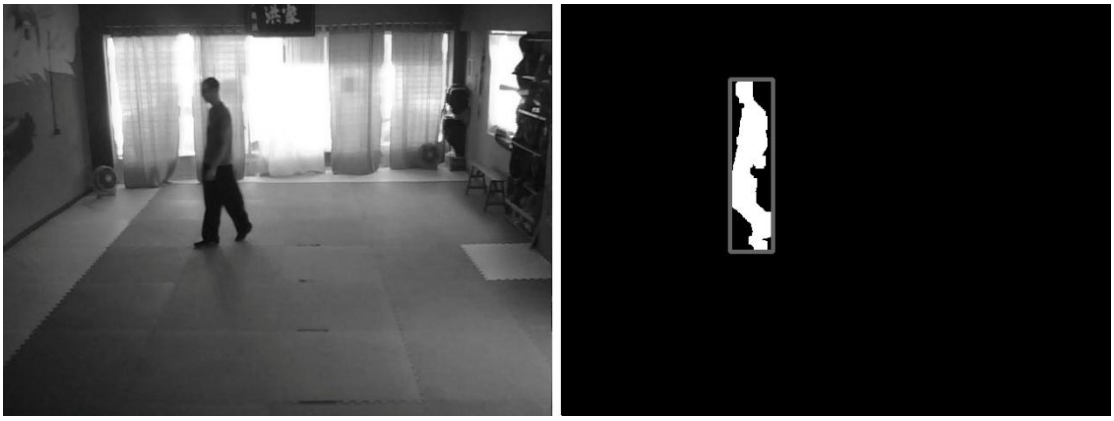
5.3 Foreground Features Extraction

After successful background subtraction and foreground extraction the next step through the development of this fall detection application is to extract foreground features that will be used as input into the fall detection algorithm.

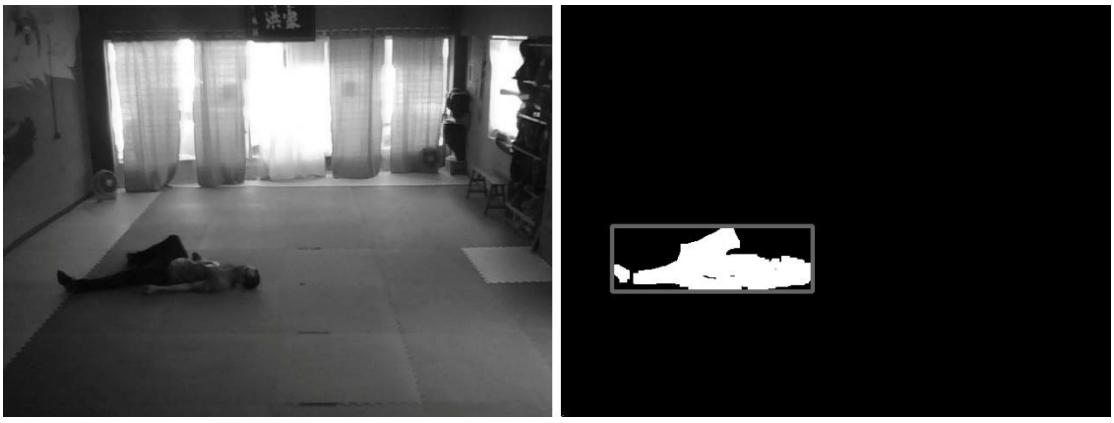
As described in chapter 4, three features are extracted from the foreground object. These features are:

i. top-most point of foreground object,
ii. height and width of foreground object and
iii. real height of foreground object.

Before features extraction, in order to refine foreground area by eliminating foreground pixels caused by noise, the system performs morphological transformations using erosion and dilation. After morphological transformations a bounding box is created around the foreground object by using the extreme (left-most, right-most, top-most and bottom-most) foreground pixels.



13a



13b

Figure 13: Foreground object after morphological transformations and bounding box

The top-most-point of foreground object as well as the height of the human blob and the aspect ratio of the height against the width will be extracted from this bounding box. Figure 13 shows two examples of foreground object after morphological transformations as well as the bounding box. In figure 13a the human is in standing position while figure 13b shows the human after a fall event. As anyone can see the aspect ratio of the width against the height of the foreground object, which is used as input into the fall detection algorithm to estimate if a fall incident occurred, is much bigger in figure 13b than in figure 13a.

For the extraction of the real height of foreground object, things are a little bit more complicated. As described in chapter 4 the extraction of this feature requires a calibrated camera and inverse perspective mapping transformation. Camera calibration as well as inverse perspective mapping transformation is a manual process, based on the recognition of "known" objects like a chessboard.

During camera calibration process and inverse perspective mapping transformation experimental errors is a common issue. In order to minimize these errors camera calibration and inverse perspective mapping transformation routines were run multiple times and average value of each item of intrinsics, distortion and homography matrices was used to form the final matrices. The final intrinsics, distortion and homography matrices are shown below.

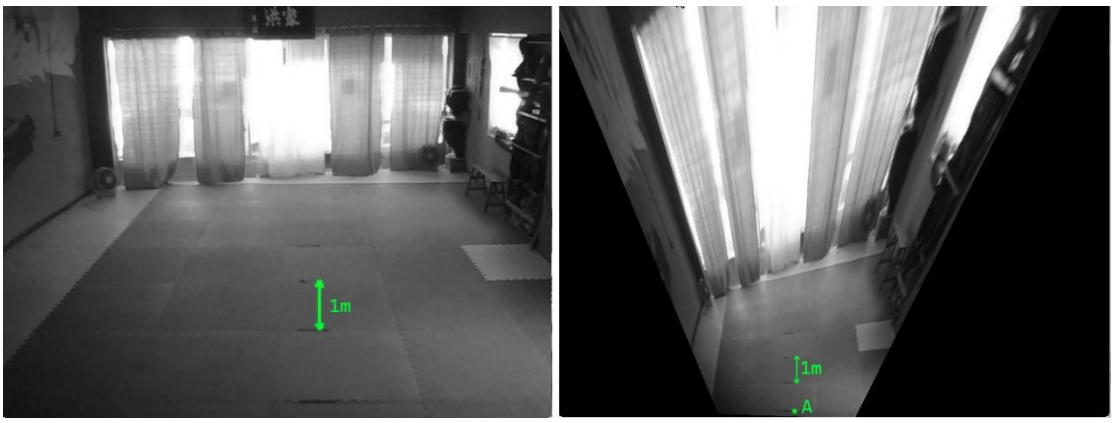
$$Intrinsics = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 672.12 & 0 & 309.74 \\ 0 & 662.48 & 207.14 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Distortion = \begin{bmatrix} k_1 \\ k_2 \\ p_1 \\ p_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} 0.36 \\ -3.54 \\ -3.32e-03 \\ -0.043 \\ 14.66 \end{bmatrix}$$

$$Homography = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} 30.72 & -12.67 & 336.69 \\ 4.96 & 3.29 & 309.19 \\ 0.012 & -0.038 & 1 \end{bmatrix}$$

After camera calibration and inverse perspective mapping transformation the relationship between foreground object's distance from the camera and difference of foreground object's bottom-most pixel and frame's bottom-most pixel becomes linear. Examples of inverse perspective mapping transformation images are shown below.

14a



14b

Figure 14: Inverse perspective mapping transformation image

The left images show the captured frame and right images show inverse perspective transformation images. The distance between point A, in figure 14a – right image, and camera is 190cm. In the same image every pixel is associated with 1.27 cm. By this knowledge the system is able to calculate the distance between the foreground object and camera for every captured frame. The distance between point A, in figure 14b – right image, and camera is 337cm. In the same image every pixel is associated with 3.33 cm. Finally a comparison between distance representation for captured frame and inverse perspective mapping transformation image is performed..

The knowledge of distance between foreground object and camera and the height in pixels of foreground object permits the calculation of foreground object's real height in centimeters by using the form (1) of chapter 4. Because of the motion of foreground object errors in the calculation of its height can be occurred. For this reason in the beginning the height of foreground object is initialized to 175cm (average height for adult males). For every subsequent captured frame the height is being updated by the form below:

$$H_i = 0.8H_{i-1} + 0.2EH_i$$

Where H_i is the real height of foreground object at frame i and $H_0 = 175cm$ and EH_i is the estimation of height at frame i. By using this form for every captured frame, H_i converges to the real height of foreground object. It has to be mentioned that real height is being updated only if EH_i is bigger than $H_{i-1} - 10cm$ and smaller than $H_{i-1} + 10cm$. This constraint doesn't permit the height to be affected when a fall event occurs and height is significantly decreases. Examples of foreground object's real height estimation are shown below.

Figure 15: Distance and height estimation

In the first example the distance between foreground object and camera was estimated to 537cm and the height of foreground object to 189cm. In the second one distance and height estimations are 219cm and 119cm respectively. The real height of foreground object in the first case is 193cm, so there is 2% error in system's approximation while in the second case the real height is 117cm and system's approximation error is 1.7%. Thus both approximations are quite good as these errors are too small to affect the performance of fall detection algorithm.

5.4 Fall Detection Algorithm

During the experimentation process, which took place in a martial arts school, one person simulated falls, in every direction according to the camera, and normal every day activities, like lying on the floor. This way we were able to evaluate the performance of the system by testing false positive and false negative rates. During experimentation process we created a video to test the algorithm with different variable definitions. We choose to test the algorithm with a video and not real time, because this way we are able to understand how every variable affects its performance.

The system used a simple Logitech web-cam that is working at 25fps with resolution 640x480 pixels and it was placed 230cm above the floor plane. During experimentation process the background was changing dynamically, there is a continuous movement of curtains caused by little air blows and lighting conditions changed by turning on the lights.

As mentioned before, simulated falls were made in every direction according to the camera. This includes falls to the right, to the left, with forward motion and backward motion in regard to the camera position. In the sampled video there are nineteen falls. Fall examples are shown in the pictures below.

16a

16b

Figure 16: Examples of simulated falls

Figure 16 shows examples of simulated falls in different directions according to the camera position. Figure 16a shows falls with right and left motion while figure 16b shows falls with forward and backward motion.

Beside simulated falls, normal activities were simulated during the experiment, included leaning forward to tie the shoelace, lying down on the floor and sitting down on the floor. These normal activities may look like a fall, due to vertical shifting of the top-most point of the foreground object, but they are not a real fall, so they used to check false negative rate and consequently the performance of the system. During the experiment nine normal activities that may look like a fall were simulated. Examples of normal activities are shown below.

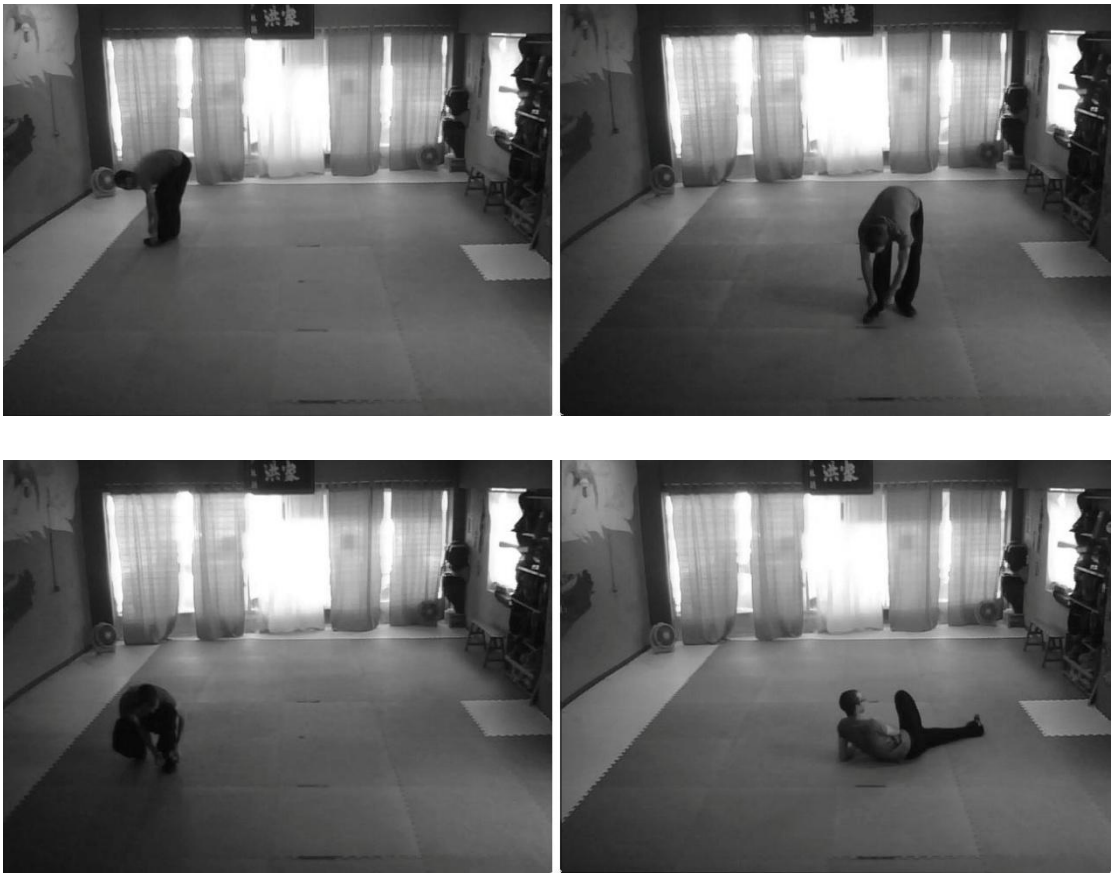


Figure 17: Normal everyday activities

All these activities look like a real fall due to the occurrence of rapid vertical motion of the moving object. So, the variable, which represents the velocity of the vertical motion is critical to distinguish a real fall from another activity and consequently to achieve high system performance.

The algorithm, in order to estimate if a fall incident occurred, checks the constraint below:

$$Asp = \frac{w}{h} > T$$

Where Asp is the aspect ratio of the height against the width of the extracted foreground object and T is a predefined threshold.

If this constraint is qualified, then the algorithm checks the vertical shifting of the top-most point of the foreground object during a sequence of frames. If this vertical

shifting is bigger than a threshold, which is relative to the real height of the foreground object then a fall event is detected.

The following diagrams show how the values of the variables affect the performance of the system.

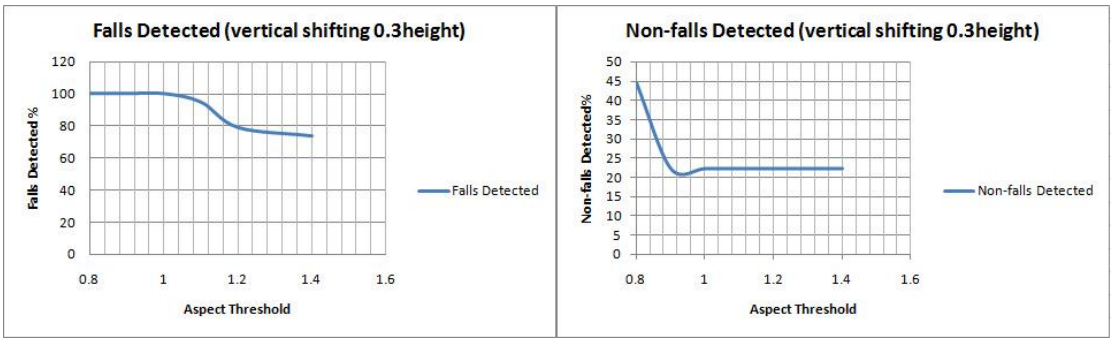


Figure 18: False negative and false positive rates in regard to aspect ratio variable

In the above diagrams, the value of vertical shifting is kept constant to 0.3 of the real height of foreground pixel, while the value of aspect ratio threshold of the height against the width was changing. It is clear that as the value of aspect is increasing the number of successful fall detections as well as the number of non fall detections is decreasing.

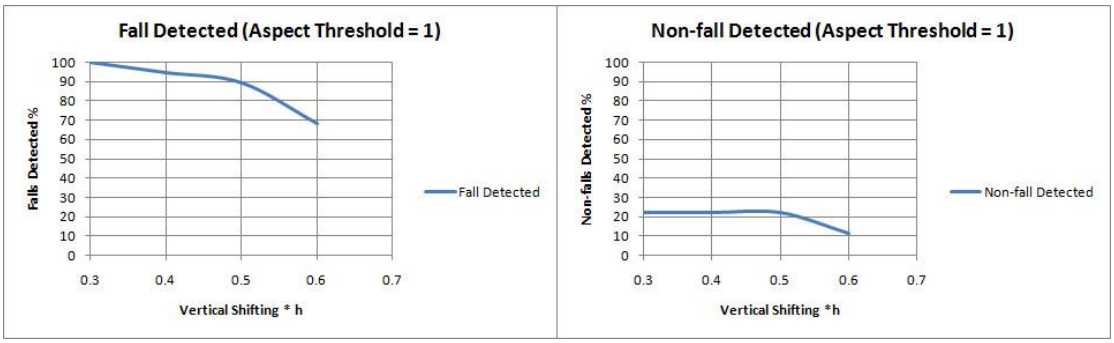


Figure 19: False negative and false positive rates in regard to vertical shifting

In this example, the value of aspect ratio threshold is kept constant to one; this means that width of foreground object is at least equal with height and the value of vertical

shifting was changing. It is clear that as the value of vertical shifting is increasing the number of successful fall detections as well as the number of non fall detections is decreasing.

By examining the above two examples it's obvious that performance is depended on the number of fall detections and the number of non fall detections, however increasing the number of fall detections and decreasing the number of non fall detections are two competitive goals. The tables below summarize the performance of our system.

Performance with Vertical Shifting = 0.3H and changing Aspect ratio threshold				
Aspect Threshold	**Falls Detected**	**Falls Detected %**	**Non-falls Detected**	**Non-fall Detected %**
0.8	19/19	100	4/9	44.4
0.9	19/19	100	2/9	22.2
1.0	19/19	100	2/9	22.2
1.1	18/19	94.7	2/9	22.2
1.2	15/19	78.9	2/9	22.2
1.4	14/19	73.7	2/9	22.2

Performance with Vertical Aspect ratio threshold = 1 and changing Vertical Shifting				
Vertical Shifting	**Falls Detected**	**Falls Detected %**	**Non-falls Detected**	**Non-fall Detected %**
0.3H	19/19	100	2/9	22.2
0.4H	18/19	94.7	2/9	22.2
0.5H	17/19	89.5	2/9	22.2
0.6H	13/19	68.4	1/9	11.1

Chapter 6

Conclusions

As people in developed countries are ageing, the humans' fall problem becomes one of the most important problems in human society with physical, psychological and economical consequences. This problem especially is concerning on elderly and patients with mental cognitive problems like mild dementia or epilepsy. So, our main goal in this master thesis was the development of a fast real-time computer vision algorithm, capable to detect humans' fall incidents. This algorithm has to operate properly in complex and dynamically changing conditions and have minimal computational cost and minimal memory requirements, in order to be suitable in large scale implementations like implementations in clinical/hospital or home environments.

Our system was tested in a martial arts school and its experimental results are very promising. This algorithm is capable to detect over 95% of fall incidents in complex and dynamically changing visual conditions, while it presents low false positive rate. Besides the contribution to humans' fall problem, this algorithm contributes to computation of significant measures of a scene like real height of the foreground object and distance between foreground object and camera position, by using a single and simple camera, like a webcam. By using these measures much more information can be revealed which will be useful in different kind of applications, like developing an intelligent car braking system.

This algorithm makes the assumption that only one person is present in the scene. So, primary priority in to-do list is the evolution of this algorithm in a way that it will operate properly with more than one person in the scene and even in crowded conditions. Beside this, another drawback of this algorithm is that background adaptation to new visually conditions, takes about ten frames and if a fall incident will be occurred during these frames it will not be detected. So, further work has to be done in order to increase background adaptation rate.

Finally, although detection rate of the algorithm is higher than 95%, when someone is dealing with a problem like the one of humans' fall detection, where human health is the primary priority, it is easily to understand that a lot of work has to be done to further increase this rate.

Bibliography

A. Bevilacqua, A. Gherardi, et al. (2008). Automatic Perspective Camera Calibration Based on an Incomplete Set of Chessboard Markers. Sixth Indian Conference on Computer Vision, Graphics & Image Processing, 2008. ICVGIP '08. . Bhubaneswar 126 - 133.

Bruce D. Lucas and T. Kanade (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. 1981 DARPA Image Understanding Workshop.

Chia-Wen Lin, Z.-H. Ling, et al. (2005). Compressed-domain fall incident detection for intelligent home surveillance. IEEE International Symposium on Circuits and Systems, 2005. ISCAS 2005. **4:** 3781-3784.

Chris Harris and M. Stephens (1988). A Combined Corner and Edge Detection. Proceedings of The Fourth Alvey Vision Conference 147-151.

D.W. Rabin (1995). "Falls and gait disorders." The Merck manual of geriatrics: 65-78.

Doulamis, N. (2010). Iterative Motion Estimation Constrained by Time and Shape for Detecting Persons' Falls. Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments. New York, USA.

Federico Bianchi, S. J. Redmond, et al. (2010). "Barometric Pressure and Triaxial Accelerometry-Based Falls Event Detection." IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING **18**(6): 619 - 627.

G. Diraco, A. Leone, et al. (2010). An Active Vision System for Fall Detection and Posture Recognition in Elderly Healthcare. Design, Automation & Test in Europe Conference & Exhibition (DATE). Dresden 1536 - 1541

Gary Bradski and A. Kaebler (2008). Learning OpenCV, O'Reilly.

Homa Foroughi, A. Rezvanian, et al. (2008). Robust Fall Detection Using Human Shape and Multi-class Support Vector Machine. Sixth Indian Conference on Computer Vision, Graphics & Image Processing**:** 413-420.

Huimin Qian, Y. Mao, et al. (2008). Home Environment Fall Detection System Based On a Cascaded Multi-SVM Classifier. 10th Intl. Conf. on Control, Automation, Robotics and Vision. Hanoi, Vietnam.

J. Heikkila and O. Silven (1997). A four-step camera calibration procedure with implicit image correction. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Juan , Puerto Rico 1106 - 1112.

J.W. Runge (1993). "The cost of injury." Emerg Med Clin North Am(11): 241-253.

Jianbo Shi and C. Tomasi (1994). Good Features to Track. IEEE Conference on Computer Vision and Pattern Recognition. Seattle.

M.N. Nyan, Francis, et al. (2008). "A wearable system for pre-impact fall detection." Journal of Biomechanics **41**: 3475–3481.

Shuangquan Wang, J. Yang, et al. (2005). Human activity recognition with user-free accelerometers in the sensor networks. International Conference on Neural Networks and Brain, 2005. ICNN&B '05. Beijing 1212 - 1217.

Thinh M. Le and R. .Pan (2009). Accelerometer-based sensor network for fall detection. Biomedical Circuits and Systems Conference, 2009. BioCAS 2009. IEEE Beijing 265 - 268

Yaniv Zigel, D. Litvak, et al. (2009). "A Method for Automatic Fall Detection of Elderly People Using Floor Vibrations and Sound—Proof of Concept on Human Mimicking Doll Falls." IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING **56**(12): 2858-2867.

Zhengming Fu, E. Culurciello, et al. (2008). Fall detection using an address-event temporal contrast vision sensor. IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008. . Seattle, WA 424-427.