Intelligent Computer Vision Processing Techniques for Fall Detection in Enclosed Environments

by

Adel Rhuma

A doctoral thesis submitted in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy (PhD)

February 2014

Advanced Signal Processing Group (ASPG),
School of Electronic, Electrical and Systems Engineering,
Loughborough University, Loughborough,
Leicestershire, UK, LE11 3TU

To my Mother and Father

Abstract

Detecting unusual movement (falls) for elderly people in enclosed environments is receiving increasing attention and is likely to have massive potential social and economic impact.

In this thesis, new intelligent computer vision processing based techniques are proposed to detect falls in indoor environments for senior citizens living independently, such as in intelligent homes. Different types of features extracted from video-camera recordings are exploited together with both background subtraction analysis and machine learning techniques.

Initially, an improved background subtraction method is used to extract the region of a person in the recording of a room environment. A selective updating technique is introduced for adapting the change of the background model to ensure that the human body region will not be absorbed into the background model when it is static for prolonged periods of time.

Since two-dimensional features can generate false alarms and are not invariant to different directions, more robust three-dimensional features are next extracted from a three-dimensional person representation formed from video-camera measurements of multiple calibrated video-cameras. The extracted three-dimensional features are applied to construct a single Gaussian model using the maximum likelihood

technique. This can be used to distinguish falls from non-fall activity by comparing the model output with a single preset threshold.

In the final works, new fall detection schemes which use only one uncalibrated video-camera are tested in a real elderly person's home environment. These approaches are based on two-dimensional features which describe different human body posture. The extracted features are applied to construct a supervised method for posture classification for abnormal posture detection. Certain rules which are set according to the characteristics of fall activities are lastly used to build a robust fall detection model.

Contents

Statement of originality

The contributions of this thesis are mainly focused on using different features and algorithms to achieve an efficient fall detection system which has potential to be used in real applications for remote health care of elderly people. These methods have been tested in the Intelligent Audio/Video Experimental Laboratory within the Advanced Signal Processing Group (ASPG) at Loughborough University and within real indoor environments where elderly people habit. The novelty of the contributions is supported by papers presented at international conferences and presented in international journals as follows:

In Chapter 3, a new method is introduced to reliably extract humans from a video-camera sequence even when the humans are static for long periods of time. The proposed method addresses a common problem in background subtraction techniques whereby humans that are static are mistaken for new additions to the background scene and are consequently absorbed into the background model. The proposed method is therefore to use head tracking to identify where the human is within the current frame and therefore all pixels which are associated with the human are not updated in terms of their background model. The results have been published in:

I. Colman, A. Rhuma, M. Yu and J. Chambers, "A robust technique for person-background segmentation in video sequences based on the

codebook method of background subtraction and head tracking", Sensor Signal Processing for Defence (SSPD), Imperial College London, UK, pp.1-5, 2010.

Chapter 5 proposes a simple and robust fall detection scheme based on three-dimensional features and a single Gaussian model.

Two video-cameras are initially calibrated by Tsai's video-camera calibration method and a three-dimensional person is then constructed from the background subtraction results of the two calibrated video-cameras. Three-dimensional feature vectors, (including the centroid position and the orientation value) corresponding to fall activities are extracted to build a model for distinguishing fall activities and non-fall activities using a single threshold in the Intelligent Audio/Video Experimental Laboratory within the Advanced Signal Processing Group (ASPG) at Loughborough University. These works were presented in:

A. Rhuma, M. Yu and J. Chambers, "Fall detection system in enclosed environments based on single Gaussian model", Journal of Measurement Science and Instrumentation, vol.3, no.2, pp.123-128, 2012.

Chapter 6 proposes an effective fall detection method for a real home application, based on a supervised learning technique. The codebook method background subtraction is used to extract the human body postures and the same post-processing technique is applied to solve the background subtraction errors caused by environmental changes in a real home environment. A new combination of features is adopted which can describe postures in more detail and used to construct the corresponding supervised classifier. The results of the constructed classifiers, together with some rules determined from the fall characteristics, are used to distinguish fall activities from non-fall activities. This re-

search work is supported by the following publications:

A. Rhuma, M. Yu, and J. Chambers,"Posture recognition based fall detection system", Lecture Notes on Software Engineering, vol. 1, no. 4, pp. 350-355, 2013.

M. Yu, A. Rhuma, S. Naqvi, W. Liang, J. Chambers, "Posture recognition based fall detection system for monitoring an elderly person in a smart home environment", IEEE Transactions on Information Technology in Biomedicine, Volume 16, Issue 6, pages: 1274-1286, 2012.

Acknowledgements

Many people have contributed to the research in this thesis, and to the writing process. Without their help, this thesis could not have been realised.

I am very highly grateful to my dear Professor. Jonathan A. Chambers for giving me a great opportunity to work in his research group under his supervision, for his advice and his unlimited cooperation during the period of this research work. Furthermore, he is a very kind, understanding and patient professor. He has always tried to understand what my problem is and tried to understand what I have tried to communicate. Most importantly, I will consider him as my greatest advisor forever; I gained a lot from him while working under his supervision.

Thanks also go to Dr. Paul Lepper, who made valueable comments on my research work during the early stages.

I would like to thank my close friends in my research group who were really supportive and working hard behind me all the time; and a very special thanks to Dr. Miao Yu, and Ian Colman, of the Advanced Signal Processing Group, Loughborough University for their help and advice whenever needed.

I gratefully acknowledge the financial support I received from the Ministry of Higher Education, Ministry of Defence and from the Biruni Remote Sensing Centre, Tripoli, Libya for funding my research at Lough-

borough University, a very special thanks goes to Dr. Ali Ghenniwa.

I would also like to express my gratitude to Dr. Mohamed El Haram, Civil Engineering Department, University of Dundee and Dr. Ayodeji Akiwowoa, Civil and Building Engineering, Dr. Ronald Mainawairagu, Chemical Engineering Department, Loughborough University for their valuable help and advice at different stages of the research.

I am very grateful to research associates and all PhD students within the Advanced Signal Processing Group (ASPG), Loughborough University: Special thanks go to, Dr. Gaojia Chen, Dr. Anastasia Panoui, Miss. Lan, Dr. Yanfeng Liang, Dr. Ata Ur-Rehman, Mr. Abdusalam Alhutmani and others for their support of my research by helping to simulate the elderly activities.

Last, but not least, I wish to thank my family in United Kingdom and Libya, Brother, Sisters, my wife, Haya, Abdulmajed and Sraa for assisting me in achieving my aims I will never forget their constant encouragement.

List of Acronyms

ABM	Adaptive Block Matching
AMF	Approximate Median Filter
ASPG	Advanced Signal Processing Group
AVI	Audio Video Interleaved
BAN	Body Area Network
BDT	Binary Decision Tree
BGS	BackGround Subtraction
BMI	Body Mass Index
BSS	Blind Source Separation
CB	CodeBook
CCD	Center to Center Distance
CERT	Center for Eldercare and Rehabilitation Technology
2D	Two dimensional
3D	Three dimensional
DAGSVM	Directed Acyclic Graphic Support Vector Machine

DBTC	Distance Between Two Classes
FADE	Fall Detection
FHMM	Fuzzy Hidden Markov Model
FNR	False Negative Rate
GPU	Graphic Processing Unit
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HOF	Histogram of Optic Flows
HHMM	Hierarchical Hidden Markov Model
KKT	Karush-Kuhn-Tucker
k-nn	k-nearest neighbor
LHMM	Layered Hidden Markov Model
MCSVM	Multi-Class Support Vector Machine
MFCC	Mel Frequency Cepstral Coefficients
MoG	Mixture of Gaussians
MHI	Motion History Image
ML	Maximum Likelihood
MLPNN	Multi-Layer Perceptron Neural Network
MNRL	Maximum Negative Run-Length
MNRL	Maximum Negative Run-Length

MMSE	Minimum Mean Squared Error
NBC	Naive Bayes Classifier
PCA	Principle Component Analysis
ROC	Receiver Operating Characteristic
SGM	Single Gaussian Model
SiCEX	Silver Industry Conference and Exhibition
SIS	Sequential Importance Sampling
SVM	Support Vector Machine
TOF	Time-of-Flight
TPR	True Positive Rate
VVD	Vertical Volume Distribution

List of Symbols

Some frequently used notations are as follows:

θ	Tilt angle
K	Gaussian distribution
fps	Frames per second
k	1^{st} order radial lens distortion coefficient
f	Focal length of the pinhole video-camera
s_x	Scale factor
R	Rotation matrix
$\|.\|$	Euclidean norm
T	Translation vector
P	The object point in a scene
k	The coefficient of the radial lens distortion
$//$	Parallel line
Δ_t	Centroid position of the voxel person
θ_t	Orientation angle

$V'_{t,j}$	Voxel person
$\mathrm{Tr}(\cdot)$	Trace of matrix
$\mathbf{d}(\cdot)$	Differential operation
$\dot{d}_x$	The corresponding size of a pixel
$int(.)$	Represents the integer part of the result
$eigenvec_t$	Eigenvector at time t
$f(x, y)$	Binary image
$I_t(x, y)$	Intensity of a pixel
$B_t(x, y)$	Background model
D	Deviation threshold
$colordist(.,.)$	Chromatic difference

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Importance of fall detection

The importance of detecting unusual movement (falls) for elderly people is receiving more and more attention and is likely to have massive potential social and economic impact [1], [2] and [3]. Falls are an extremely common and critical health problem for elderly people in security and safety application areas including supportive home environments. Moreover, falls are the main cause of admission and extended period of stay in hospitals or nursing homes for long term treatment [4], [5], [6], and [7].

Nowadays, due to the development of the healthcare industry in modern society, human life expectancy has grown and there continues to be related trends in the population of older people across the globe [8].

According to the US Bureau of Statistics, as reported by the Guardian newspaper [9] in the United Kingdom, within 10 years, old people will outnumber children for the first time. Figure 1.1 [10] shows that over the next 30 years the number of people over 65 years of age across the world is expected to almost double, from 506 million in 2008, to 1.3 billion, a leap from 7% of the world's population to 14%. Already, the number of people in the world aged 65 and over is increasing at an

average of 870,000 each month.

Among these elderly people, a large percentage of them live alone at home independently according to the research reported in [11].

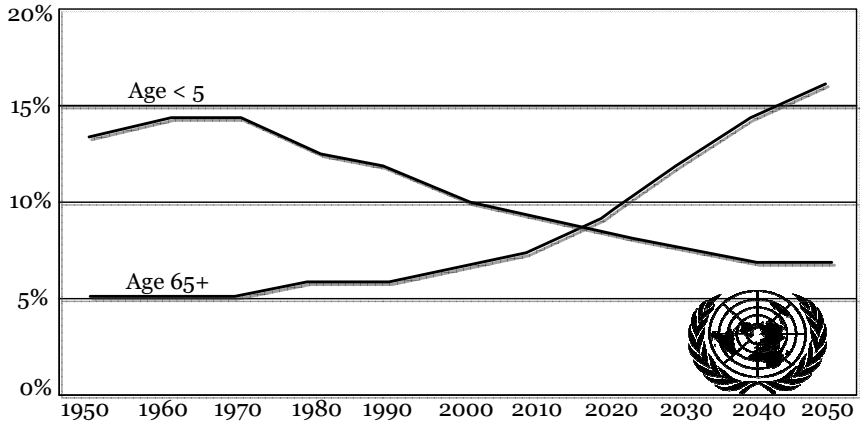


Figure 1.1. Elderly people as a percentage of global population compared with young children [10].

Therefore, caring for the elderly living alone is very important and poses a big challenge in the community and the world at large.

There are many issues related to the care of elderly living alone at home. Perhaps, one of the most important factors is detecting the occurrence of unusual movement such as falls. According to [12] and [13], falls occur commonly in the elderly community and can lead to serious damage, such as broken bones, connective and soft tissue damage, and may even cause death; as such the problem is responsible for considerable medical costs, morbidity and mortality among the elderly population. Besides, as reported by Age UK [14], in the United Kingdom, it has been shown that about 250,000 people aged 65 and over in England are treated in hospitals as a result of a fall every year.

As reported in [15], in the United States, falls happen among the elderly with a median age of 79 and commonly result in fractures (primarily

hip and femur), estimated at 155,000 to 200,000 each year. Cost estimates range from a current annual amount of 750,000,000 to one billion US dollars, and a similar situation exists in other developed countries. Unlike in the case of monitoring young children, for example in a nursery, it is unrealistic to assign nurses to take care of elderly person in their homes on a 24/7 basis. So, instead of human resources, new technologies are required to detect unusual activity (falls) when they occur, as part of the area of healthcare based on assisted living, with the target of reducing the tremendous costs incurred by falls in a home environment.

The governments in many developed countries have increased investment to push development of fall detection technology.

As reported in [16], a conference held in Singapore, Silver Industry Conference and Exhibition (SiCEX), promoted concepts, products and technologies related to healthcare for the elderly with focus on fall detection.

In the United States of America (USA), many research institutes, which include interdisciplinary groups of faculty, staff, and students, are being built to investigate, develop, and evaluate technology to serve the needs of older adults and others with physical and cognitive challenges. One of these is the Center for Eldercare and Rehabilitation Technology (CERT), University of Missouri, Columbia, where one important project is passive fall detection and gait analysis for fall risk assessment, investigating a non-intrusive method to detect falls in a home environment [17] and [18].

1.2 Falls and typical scenarios

In this section, the definition of a fall and its typical scenarios is firstly summarised before introducing different fall detection techniques.

According to [19], a fall is defined as "unintentionally coming to the ground or some lower level other than as a consequence of sustaining a violent blow, loss of consciousness, sudden onset of paralysis as in stroke".

Some other researchers have used a broader definition to include those falls which occur as a result of dizziness as in [20]. Based on the particular definition, falls can be divided into different scenarios according to different criteria [4]- [7] and [20]:

a) According to the orientation:

1. Frontal fall: A person falls towards his/her frontal direction, mostly with his/her face impacting with the floor.

2. Backward fall: A person falls towards his/her backward direction, mostly with the back of their head impacting with the floor.

3. Side fall: A person falls towards his/her side direction.

b) According to the amplitude:

1. Fast fall: A person falls fast, the amplitude of the body movement is large, the duration is short (1-2s).

2. Slow fall: A person falls slowly, the amplitude of the body movement is comparatively small and the duration is comparatively long.

c) According to the transition of postures:

1. Fall from standing: A person falls from an initial standing posture. This type of fall occurs when an elderly person walks or stands still due to slipping or unconsciousness. Both the head and center of gravity move towards one direction and their heights reduce (normally to the plane of the ground). Typically, this type of fall belongs to the category of fast fall with large movement amplitude.

2. Fall from sitting: A person falls from an initial sitting posture, this type of fall occurs when an elderly person slips from a chair due to his/her unconsciousness. Similarly, the head and center of gravity move towards one direction with a reduced height; however, compared with fall from standing, this type of fall has smaller movement amplitude.

3. Fall from lying: A person falls from an initial lying posture. This type of falls means that an elderly person rolls to the floor from the bed during sleep. The person is initially on the bed when a fall happens and the body reduces its height from the bed to the floor plane, with the final body position being near the bed. This type of fall usually happens when an elderly person sleeps and his/her body rolls out of the bed while the person remains unconscious.

4. Fall from other postures: A person falls from an initial bending/crouching or other posture. This type of fall happens for example when an old person ties his/her shoe lace or does other activities and suddenly becomes unconscious.

Compared with non-fall activities, a fall is an unconscious activity or an activity happening beyond an old person's control (such as he/she slips and falls).

Typically, fall activities end with a lying posture on the floor, as presented in Figure 1.2.

Figure 1.2. An old person falls ending with a lying posture on the floor.

In the next section, the general scheme of a fall detection system is presented.

1.3 Overview of a fall detection system

A general complete fall detection system is proposed in Figure 1.3. Initially, vision signals are acquired from different types of sensors (including wearable and non-wearable device sensors such as accelerometers, video-cameras respectively) and the acquired signals are then processed with the corresponding information extracted. This information is then fed into a fall detection system to detect falls with the aid of certain algorithms (typically analytical algorithms or machine learning algorithms). When the fall activity of an elderly person is detected, an alarm signal is generated and this signal will be either sent to his/her

family members, or some caregiver suppliers (including a hospital, or monitoring center for elderly people) by modern communication techniques (such as the wired or wireless communication network depicted in Figure 1.3.

After receiving the alarm signal, the care staff will swiftly come to assist the elderly person at the right time. It is important that such systems minimise the number of false alarms and accurately report fall events.

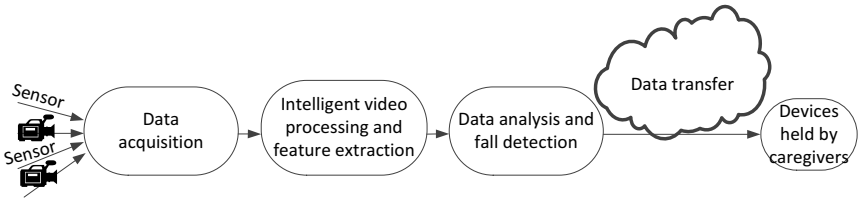


Figure 1.3. The diagrammatic representation of a complete fall detection system.

Some examples of the different sensors and extracted information used to detect falls are presented in Figure 1.4. There are non-vision based sensors (wearable device or ambience device) or intelligent vision based sensors (video-cameras) to capture certain types of signals, and corresponding information (posture information, motion information, and body shape change information) from the acquired signals for fall detection purpose. This thesis focuses on fall detection in indoor environments by applying intelligent vision based processing techniques with effective vision methods proposed for detecting fall activities by using one or multiple video-cameras with processing which can be performed on a normal personal computer. As such, this type of system can be used to cover a wide area and in different places. The main disadvantage of these techniques is the accuracy will be affected by the lighting condition problem, changing the background model (moving

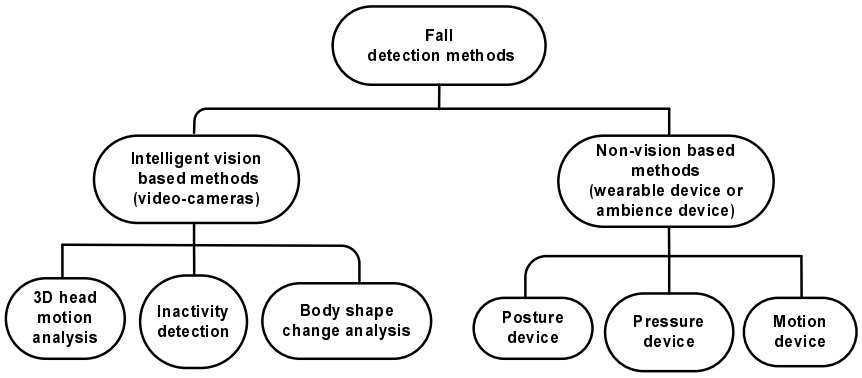


Figure 1.4. Different categorisation of fall detection methods [1] and [16].

furniture for example) and foreground object when it has been static for period of time. Moreover, they are prone to generate false alarms even though the context information is used and they have a late alarm because they detect a fall only when the person lies on the ground for a while. However, this approach has the major advantage that it avoids the requirement for wearing a sensor or recharging the batteries which becomes difficult for elderly people, particularly those suffering from conditions such as comatose. Therefore, the target of this thesis is to improve the robustness of video based techniques through more intelligent processing.

1.4 Aims and objectives

The aims of this thesis are to:

- Exploit state-of-the-art intelligent vision based methods in the development of fall detection systems for potential application in healthcare based assisted living for elderly in indoor environments.

- Perform extensive study of the three modern background subtraction techniques which have been employed and improved to extract moving objects from sequences of images or video-camera in a room environment and determine the most suitable approach for the fall detection system in indoor environments.

- Provide a mathematical and technical overview of the video-camera calibration method based on Tsai's model.

- Extract three-dimensional features to characterise the position and pose of a human target.

- Employ support vector machine (SVM) based classifiers to perform robust fall detection. A new scheme is also introduced based on appropriate rules to minimise false alarms in the detection of falls.

- Evaluate methods on extensive datasets measured in the Advanced Signal Processing Audio/Video Laboratory and real room environments.

At the end of the study the objectives are to have:

- Demonstrated the feasibility of fall detection with the proposed system using datasets with volunteers who attempt to simulate the movements of an elderly person in a real elderly home environment.

- Published the research findings in leading international conferences and journals.

1.5 Overview of the Thesis

This section provides brief description of the research work presented in this thesis.

Chapter 2, reviews related state-of-the-art fall detection schemes by using either non-vision techniques or intelligent vision based techniques, which provide the background for later work within the thesis.

In Chapter 3, three background subtraction techniques are compared including the approximate median filter (AMF), mixture of Gaussians (MoG), and codebook (CB) methods. They are used to develop a robust video processing technique that reliably detects and extracts the presence of a person in the recording of a room environment based on background subtraction techniques and head tracking based method. Extracting moving objects from sequences of images or video stream is an interesting problem and remains a challenging topic in machine vision based method. The aim of the chapter is to develop a robust video-camera processing technique that reliably detects and extracts the presence of a person in the recording of a room environment based on background subtraction techniques and tracking algorithm to ensure that the human body will never be lost into the background model, even when static for an indefinite period of time.

Chapter 4, provides a technical overview of video-camera calibration based on Tsai's model. By applying the Tsai's video-camera calibration using a set of correspondent points (three-dimensional points and corresponding two-dimensional image points), both the external and internal parameters of a video-camera can be estimated. These estimated parameters can be further used to obtain particular three-dimensional types of information, such as the approximated three-dimensional hu-

man body shape region, which is presented in detail in the next following Chapter 5.

In Chapter 5, a fall detection scheme is proposed based on three-dimensional features initially using two video-cameras calibrated by the popular Tsai's video-camera calibration method. A three-dimensional person is then constructed from the obtained codebook background subtraction results from two calibrated video-cameras. A five-dimensional feature vector is obtained (including the three-dimensional centroid position of the voxel person, vertical position, horizontal variation of the centroid and three-dimensional orientation angle) corresponding to fall activities are extracted to build a model for distinguishing fall and non-fall activities. A single Gaussian model is used in this chapter for building the classifier. The experiments were performed in a simulated laboratory environment and one person was invited to participate in the experiment by simulating different fall activities and non-fall activities, which were used to build the training dataset for model construction and test dataset for performance evaluation using receiving operating characteristic (ROC) analysis.

In Chapter 6, an effective fall detection method for a real enclosed home environment is presented. In particular, only a single camera and PC reduce considerably the complexity of the scheme in Chapter 5. This is based on a supervised learning technique.

The codebook background subtraction method is reused to extract the postures and certain advanced post-processing techniques are applied to reduce the background subtraction noise cased by indoor environmental changes in a real home environment. Some features (projection histogram and ellipse features) which can describe postures in detail are

extracted and used to construct the corresponding supervised directed acyclic graphic support vector machine (DAGSVM). The classification results for the DAGSVM, with some rules determined from the fall characteristics, are used to distinguish fall activities and non-fall activities. The experiments are performed in a real enclosed home environment, for the supervised learning based fall detection methods. Fifteen people were invited to simulate different postures, which are then used to construct a dataset used for training the DAGSVM. A series of simulated fall and non-fall activities by different people were recorded for testing purpose.

Finally, in the last Chapter 7, this thesis concludes by summarising its contributions and suggestions are given for future possible research directions.

Chapter 2

RELATED LITERATURE REVIEW

2.1 Introduction

This chapter reviews some previous literature related to fall detection methods in enclosed environments. The detection of fall of elderly people is an interesting scientific problem which one could approach using various methods. Although the concept of a fall may seem to be common sense, it is difficult to describe it precisely, and thus to specify by means of detection. It could be described as the rapid change situation from the upright/sitting position to the reclining or almost lengthened position, but it is not a controlled movement, such as lying down on the sofa, for example [21]. Distinct methods have been developed for fall detection to monitor an elderly person in enclosed environments. These methods have been modified and extended and can generally be divided into two categories. That is, non-vision based methods, and intelligent vision based methods for fall detection as shown in Figure 2.1 [1] and [16] repeated from Chapter 1 for convenience. For intelligent vision based methods, video-camera sequences are captured by digital video-camera recording and intelligent vision techniques are applied to

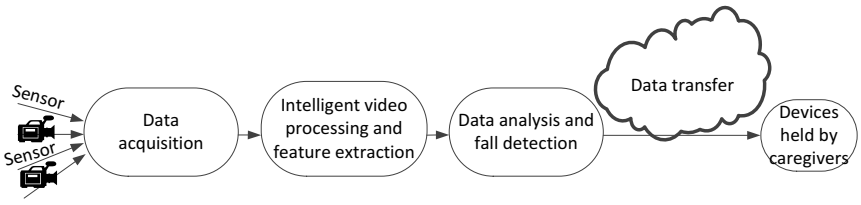


Figure 2.1. The hierarchy of approaches of fall detection methods [1] and [16].

analyse these captured video signals. For non-vision based techniques, different types of sensors (such as accelerometers and gyroscopes) are applied and non-video signals are fed into the fall detection system for evaluation. The details of these two categories are presented in the next two sections.

2.2 Non-vision based methods for fall detection

There are many non-vision based methods for fall detection in the global market today. For these methods, different sensors are used to capture the sound, vibration and human body movement information and such information is applied to determine when a fall happens [22].

The most popular used sensors in this non-vision category are accelerometer and gyroscopic sensors. For accelerometer sensors, they are typically based on small integrated circuits consisting of two surface micro machined capacitive sensing devices and a signal conditioning unit contained in a single integrated circuit package. This type of sensor is typically used to measure acceleration, tilt angle θ, the direction of the acceleration of a body along the X, Y and Z axes due to movement and acceleration due to gravity [23]. These sensors generate a signal that acts as input data to a computer system or an embedded system which

then analyses the data to detect falls. Figure 2.2 shows at a high level the sensor and the principle of using accelerometers for fall detection.

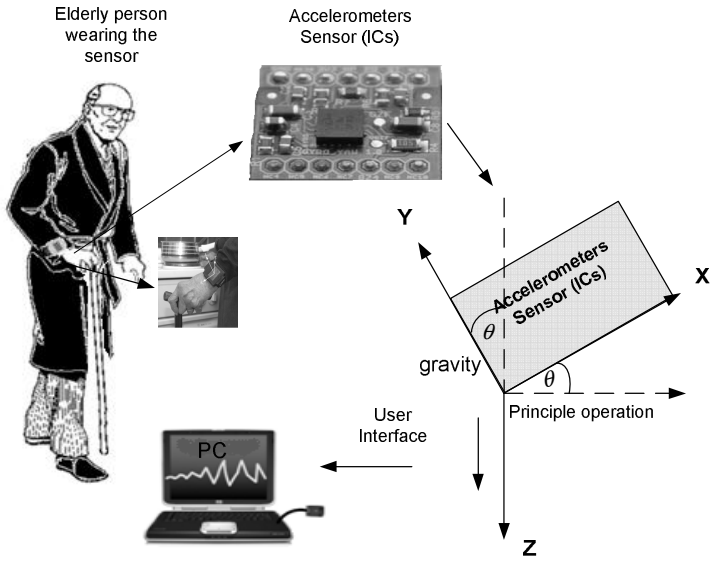


Figure 2.2. Block diagram of basic principle of accelerometer sensor and fall detection system [23].

Veltink et al. [24] were the first to utilise a single axis acceleration sensor to distinguish dynamic and static activities in 1996.

Acceleration sensors were placed over the chest and at the feet to observe the changes. Since then, several studies of using accelerometer and gyroscope sensors have been published in the last 10 years [25]. The simple and basic approach is to use the sensors (accelerometer and or gyroscope) based on a threshold value as a fall detection process denoting a fall when the acceleration is maximum.

Acceleration thresholds for fall detection have been studied using triaxial accelerometric measurements at the waist, wrist, and head for different fall events (forward, backward and lateral). A study has shown that the measurements from the waist and head have potential to distinguish between falls and activities of daily living [26]. Besides, triaxial

accelerometers worn on the chest were used in [27] and [28]. They detected certain falls with 98.9% accuracy by applying a simple threshold to the acceleration.

Kangas et al. [26] designed a single three axis acceleration sensor to attach to the subject's body in different positions: head, waist and wrist to sense fall accidents. The dynamic and static acceleration components measured from these acceleration sensors were compared with proper thresholds to determine a fall. The results showed that a simple threshold based algorithm was appropriate for certain falls, and optimum sensing effect could be achieved at the head and waist.

The other sensor which can be used for fall detection is a gyroscope Figure 2.3, which measures orientation and consists of a spinning wheel whose axis is free to take any orientation.

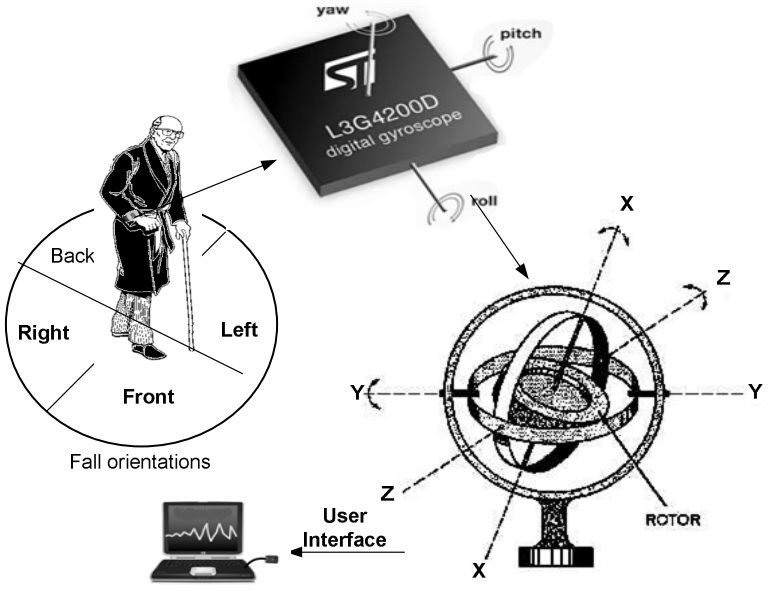


Figure 2.3. Block diagram of basic principle of digital gyroscope sensor within fall detection system [29].

It measures the orientation along one axis or multiple axes. By equip-

ping an object with the gyroscope to measure the orientation along three axes, it is possible to determine the orientation of the object and the changes in orientation, from which the angular velocity can then be computed [29].

Bourke and Lyons [30] used a biaxial gyroscope worn on the chest to measure angular velocity data based on thresholds and optical motion capture to distinguish between falls and non-fall activities. They showed that they could detect falls with 100% accuracy.

Sometimes, accelerometers and gyroscope can be used in a combined way in order to obtain a better result.

Tong et al. [31] used a combination of accelerometers and a gyroscope to detect acceleration and orientation of the subject for detecting falls. The data from sensors were processed locally and a call sent wirelessly to the main server to flag a fall. The system was attached to the chest or back of the person, which had been determined as the best option after considerable research on positioning of kinematic sensors. In the case of a fall, the accelerometer sensed acceleration greater than a set threshold value and the gyroscope determined the orientation of the subject; if there was a fall, an emergency call was sent to the care people. Moreover, using both an accelerometer and two gyroscopes could be used to detect the forward falls. The study showed that one could be able to detect successfully all 60 falls and differentiate between falls and activities of daily living with no false positives.

Nyan et al. [32] used a three-dimensional accelerometer and 2D gyroscope based on a body area network (BAN). The experimental results showed that falls could be detected with an average time of 700ms before the impact fall detection occurred, with no false alarms, and a fall

detection rate of 95.2% can be achieved.

In addition, other researchers have used acoustic and vibration sensors to detect falls. Li et al. [33] developed an acoustic fall detection system, which automatically detected a fall and reported it to the care giver. The study used an 8-microphone circular array which provided a better three-dimensional estimation of the sound location. Although promising results were obtained in their experiment, whether this algorithm was workable on more realistic datasets, such as falls in presence of noises, needed more testing.

In Mihail et al. [34], an acoustic FAll DEtection system (FADE) that will automatically signal a fall to the monitoring care giver was designed. A linear array of an electric microphone [35] condenser acoustic sensors was applied to obtain the audio signal; mel frequency cepstral coefficients (MFCC) features were extracted and the k-th nearest neighbor method was applied to determine a fall and non-fall activity. The sound was considered a false alarm if it came from a source located at a height higher than two feet so that the false alarm rate could be reduced. Their method seemed to be successful on a limited dataset, more experiments were needed however to determine whether their method would be successful in real scenarios.

Alwan et al. [36] proposed a design for a floor vibration-based fall detection system that was completely passive and unobtrusive to the resident.

The system used a special piezoelectric sensor coupled to the floor surface by means of mass and spring arrangement. Successful differentiation between the vibration patterns of a human fall from other activities of daily living and from the falls of other objects was achieved. Lab-

oratory tests were conducted using anthropomorphic dummies. The results showed 100% fall detection rate with minimum potential for false alarms. The drawback of this approach was the limited range of the vibration sensor; i.e. only six meters. Moreover, the vibrations couldn't be detected on all kinds of floor materials. The piezoelectric sensor only captured the signal produced by the floor vibration; it was robust to the background noises.

Although non-vision based methods show a potentially wide application in the fall detection field; however, several problems exist. They are intrusive (accelerometer and gyroscope) and easily affected by noises in the home environment (acoustic and vibration based methods). In order to overcome these problems, intelligent vision based fall detection techniques are next considered.

2.3 Intelligent vision based fall detection

In the last 10 years, there have been many advances in intelligent vision and camera-video and advanced image processing techniques that use real time movement of the subject. This is the most commonly used solution for fall detection at present to solve a particular problem such as elderly fall detection, which opens up a new branch of methods for fall detection.

Compared with non-vision based methods, intelligent vision based methods have the following advantages:

1. They are non-intrusive, an elderly person need not wear some special equipment such as an accelerometer;

2. They are not easily affected by noise in the home environment

(suffered by floor vibration and acoustic sensors based methods).

Based on the analysis of algorithms for fall detection, the intelligent vision based methods are divided into two categories; background subtraction based analytical and machine learning methods.

2.3.1 Background subtraction based analytical methods

In background subtraction analytical methods, certain types of video-camera features are extracted and these features are analysed empirically to determine whether falls happen or not.

Tao et al. [37] used the aspect ratio of the human body as a basis for fall detection. They apply a single Gaussian like model for background subtraction to extract the human body region and use the aspect ratio measurement of the human body and hypothesis testing for fall detection. Although simple, the aspect ratio feature can not efficiently distinguish the different poses from different view-points (only limited postures viewed from limited view-points are considered); secondly, this method only considers the transition from the standing posture to fall posture, which is incomplete in the real environment; besides, when there was occlusion in the scene, this method is likely to fail.

Wang et al. [38] used a calibrated omnidirectional vision sensor for abnormal event and fall detection. Firstly, they use a camshift algorithm for tracking the object in the scene; the tracked trajectory information was used to estimate the 'activity zone' and 'non-activity zone' in order to detect an abnormal event (an object stays in the 'activity zone' for a long time); besides, the object's physical height (which is known beforehand) and the one calculated from the calibrated video-camera was compared to determine whether a person falls or not. For the ap-

plication of fall detection only, this method was inconvenient due to a person's height information being needed beforehand; besides, when there was occlusion, such as a person passing by a table, this will fail.

Rougier [39] developed an approach to detect a fall using monocular three-dimensional head tracking in real time. The tracking component first located the head, and estimated it. The fall detection component computed the vertical and horizontal velocity of the head and then used two appropriate thresholds to distinguish falling from walking. They used manual thresholds and the system dealt with one moving object only; however, head tracking can not always guarantee 100% correct result and the false alarm rate for this system is high (fast nodding will be taken as falls).

Rougier et al. [40] also proposed a new method to detect a fall event, the method was based on the motion history image (MHI) and some changes in the shape of the person. They assumed that the motion was large when a fall occurred, and they used the motion history image to extract the motion. When a large motion was detected, they analysed the human shape of the person in the video-camera sequence to check if the person was on the ground. A background subtraction technique was used to segment the person in the video-camera sequence. Although this method considers falls from different view-points, as for other threshold based methods, its performance is affected by occlusion.

Toreyin et al. [41] suggested a method for fall detection by making use of an hidden Markov model (HMM) using both audio and video-camera. For the vision part of the approach, the aspect ratio of the bounding box of the moving region detected with a standard camera was passed

to the motion interpretation module. More precisely, its wavelet transform was used as input feature for the HMM. Using conjointly video-camera and audio cues seemed to be well founded; however, only three state models for fall and walk activities were built, hence it was not comprehensive.

Miaou et al. [42] and [43] used a detection system consisting of an omni-dimensional video-camera and a computer server. This has the advantage of capturing the whole 360 ° simultaneously in a single shot. This way the problem of conventional video-cameras having blind spots was removed. In this approach, a clean background was first obtained. After that, the foreground of interest was obtained by subtracting the background model from the current image. After removing noises from the picture, a rectangle enclosing the object was created. The height to width ratio of this rectangle was used to detect falls. The threshold value in this system was customizable depending on personal physique. The experimental results showed a detection rate of 78% without personal information that increased to 90% with personal information. The drawbacks of this system were that the monitored individual needed to give his personal information, such as height, body mass index (BMI). This increased the infrastructure required to implement this system.

The work in [44] proposed a system consisting of a fixed video-camera and a personal computer. The first step in this approach was segmenting the foreground. It was achieved by subtracting the background from the current frame. The next step was the feature extraction process. The features extracted in this approach were the horizontal and vertical projection histograms of segmented foreground and the angle between the last standing posture with the current foreground bound-

ing box. As a final step the falling speed was used to infer the real falling events. A recognition rate of about 90% was achieved in the experiments. The occlusion problem, which is obstruction of the individual by some other object, (due to a dynamic background) exists in this approach. The possible blind spots were also a major concern due to the use of a single video-camera.

Lee [45], designed an intelligent emergency response system to detect falls in the home environment. It used image based sensors. A pilot study was conducted using 21 subjects to evaluate the efficacy and performance of the fall detection component of the system. Trials were conducted in a mock-up bedroom setting, with bed, a chair and other typical bedroom furnishings. A small digital video-camera was installed in the ceiling at a height of approximately 2.6m. The video-camera covered an area of approximately $5.0m \times 3.8m$. The subjects were asked to assume a series of postures, namely walking/standing, sitting/lying down in an inactive zone, stooping, lying down in a 'stretched' position, and lying down in a 'tucked' position. These five scenarios were repeated three times by each subject in a random order. These test positions totalled 315 tasks with 126 fall simulated tasks and 189 non-fall simulated tasks. The system detected a fall on 77% of occasions and missed a fall on 23%. False alarms occurred on only 5% of occasions. The results encouraged the potential use of a vision-based system to provide safety and security in the homes of the elderly.

Nait-Charif and McKenna [46] proposed a method for automatically extracting motion trajectory and providing human readable summary of activity and detection of unusual inactivity. Tracking was performed with an omni-camera by a particle filter on the ellipse parameters de-

scribing human posture. A fall was detected as a deviation to usual activity. A total of 97 sequences were acquired at 30 Hz with a resolution of 480 × 360 pixels (46755 frames, 26 minutes). Acquisition was over two days of changeable weather. The scene contained multiple light sources (windows and indoor lighting) and no attempt was made to control the extent of lighting changes and cast shadows.

Anderson et al. [47] used a system consisting of a single video-camera and a computer. In this approach, video-camera was recorded at 3fps and then the silhouette of the individual was segmented from the background. This was done by first statistically modeling the background and then segmenting the human, based on colour information. The brightness feature was also used for the detection and removal of shadows. After the silhouette was obtained from the frames of the video-camera, a feature was extracted to determine a falling activity. The feature extracted was the width to height ratio of the bounding box of the silhouette. Experiments showed encouraging results, although no exact figure was stated. This approach had certain shortcomings, such as using only a single feature to make a decision and a single video-camera which limited the viewing angle of the individual, and it had a limited range of experimentation.

Thome and Miguet [48] used an hierarchical hidden Markov model (HHMM) based algorithm to detect a fall. The single feature of an HHMM was the orientation of a body's blob. The state level of an HHMM was the postures of the body. The other two levels of the HHMM represent behaviour pattern and global motion pattern respectively. The complicated structure of HHMM limits its real time application. Only two types of activities (walk and fall) were analysed, more

types of activities should be evaluated.

Cucchiara et al. [49] proposed a system consisting of a normal online workstation and a fixed calibrated camera in an indoor environment. In this approach, first of all the background was identified. Foreground segmentation was then achieved by subtracting the background from the current frame. Shadows and ghost pixels were also processed to achieve better foreground segmentation. Projection histograms were used to estimate the posture of the person. If the person was detected to be in a lying position and was static for too long, an alarm was generated indicating detection of a fall or system failure. Experimental results showed up to 90% success rate. This system couldn't differentiate between standing and crouching.

Hsu [50] used deformable triangulation of body shape to classify the postures of a person (one class in fall). Firstly, the extracted human shape was triangulated; two types of features from the triangulation result, skeleton and centroid contexts, were extracted for posture recognition. The behaviors (walk and fall, which were represented in the video sequence) were then classified by a set of symbols generated by the posture classification of the behavior's key frames. However, for this method, it was impossible to be real time because of the complexity of the posture classification step (the extraction and matching of skeleton and centroid contexts are time consuming); besides, there exists a problem of how to segment a long video properly to obtain video sequences containing one behavior in the real application.

Williams [51] developed a fall detection algorithm for a smart sensor network, which consisted of a number of low resolution video-cameras. The video-cameras are calibrated so that this system can not only de-

tect a fall event, but also locate the place of a fall (it was mentioned that localisation errors of less than 0.6 meters can be achieved). Due to the system being based on low power hardware a complicated algorithm could not be executed in this system, they proposed to use a threshold on the aspect ratio (the length between width and height) to detect a fall.

Lin et al. [52] and [53] developed a fall detection algorithm based on two-dimensional shape of human extracted from compressed domain. Global motion parameters were estimated for object extraction to distinguish between local object motions and camera motions, and also to obtain a rough object mask. After detecting the moving objects, compressed domain features (centroid and projection histogram) of each object were then extracted for identifying and locating a fall incident. The result showed the method correctly detected fall incidents in real time. But it can not overcome the problems of occlusion and falling in different directions commonly existing in the current fall detection systems.

Yu et al. [54] proposed a new fall detection method based on the three-dimensional head velocities in both the horizontal and vertical directions. Motion history image and codebook background subtraction were combined to determine whether large movement occurred within the scene. Based on the magnitude of the movement information, particle filters with different state models were used to track the head. The head tracking procedure was performed in two video streams taken by two separate video-cameras and three-dimensional head position was calculated based on the tracking results. Finally, the three-dimensional horizontal and vertical velocities of the head were used to detect the

occurrence of a fall. This study detected a fall event when it was happening so the alarm signals could be sent immediately. However, only the head's position information is used and the false alarm rate is high (fast nodding will be mistaken as a fall).

Shoaib et al. [55] proposed a novel context based human fall detection mechanism in a real home environment. The image was divided into small blocks and a training phase is needed to estimate the normal head and floor blocks, while each floor block corresponds to a Gaussian distribution. To detect a fall, head and feet locations and the vertical distance of the object head from the head mean location (the mean of the positions of the head blocks) were checked. If the distance is large, then a fall is detected. The core of this system was the location of the head's position and it will fail when the head is invisible in the image.

Auvinet et al. [56] employed a network of multiple video-cameras to reconstruct three-dimensional bodies of the subjects. Their system then detected falls by analysing the volume distribution along the vertical axis. An indicator based on the ratio between the sum of vertical volumn distribution (VVD) values from the first 40cm with respect to the whole volume were computed to compare with a threshold, if it was larger than that threshold, then a fall is detected. By using multiple cameras, their system can effectively deal with the occlusion problem and this fall detection system is robust to detect falls in different directions. The only inconvenience of this fall detection system was that video-camera calibration was needed to obtain the three-dimensional volume information needed to detect a fall.

Leone et al. [57] proposed a similar approach that reconstructed three-dimensional bodies of the subjects. In their work, the mixture of Gaus-

sian background subtraction method is used initially with a depth image to obtain the foreground region. The three-dimensional position of the centroid of the foreground region is obtained by a special time-of-flight (TOF) camera, which is self calibrated initially by automatic floor detection. The distance from the three-dimensional centroid position and the floor plane was compared with a threshold to detect a fall. In this method, the need for a special sensor limited its popularity.

Zambanini et al. [58] proposed to detect falls by analysing bounding box aspect ratio, orientation, axis ratio and the speed of motion from the three-dimensional visual hull. In their method, these three-dimensional features were extracted from the reconstructed three-dimensional visual hull, which is obtained from the background subtraction results of multiple calibrated video-cameras. A fuzzy logic system based on these three-dimensional features was then applied to determine the confidence of a fall. This method is simple and effective; however, the video-cameras need to be calibrated and the membership functions and fuzzy rules were all set empirically.

In [59], a model based threshold method was proposal using three-dimensional features. Calibrated video-cameras were used to reconstruct the three-dimensional shape of a person. Fall events were detected by analysing the volume distribution along the vertical axis, and an alarm was triggered when the major part of this distribution was abnormally near the floor over a predefined period of time, which implied that a person had fallen on the floor. The experimental results showed good performance of this system (achieving 99.7% fall detection rate or better with four cameras or more) and a graphic processing unit (GPU) was applied for efficient computation.

For obtaining a good fall detection performance, proper thresholds and different features are need to be chosen for the fall detection systems; however, in the real scenario, sometimes it is difficult to choose the proper thresholds for different persons to be monitored. In order to solve this problem, machine learning algorithms can be applied.

2.3.2 Machine learning algorithms

Machine learning algorithms, as proposed in [60], have been used in a wide range of areas and many researchers have applied these machine learning algorithms for fall detection application. For machine learning algorithms, different types of video features are extracted from video signals, and these features are used to train the classifiers by supervised methods to classify different types of postures or activities for fall detection, to distinguish normal activities and fall activities when they occur.

2.3.2.1 Supervised learning algorithms

For the supervised learning based fall detection systems, some video features are extracted from postures or short video sequences are then used to construct a particular supervised classifier for distinguishing different postures or activities to detect falls.

Posture recognition based fall detection methods are proposed in [61], [62] and [63]. In [61] and [63], projection histogram features were extracted from the segmented human body region from the background subtraction method and different types of supervised classifiers such as a neural fuzzy network in [61] and posture probabilistic template in [63] were constructed from the projection histogram features for classifying

different postures. If the detected posture changed from 'stand' to 'lie' in a short time [61] or the 'lie' posture stays for a long time [63], fall activities are reported. A posture recognition rate of 97.8% was achieved in [61] and four sequences were shown to illustrate falls can be successfully detected by the posture recognition results combined with the rule set in this work (a fall is confirmed when the posture changed from 'stand' to 'lie' in a short time), and a fairly robust posture recognition result (about 90% for three different datasets) was reported in [63].

Similar projection histogram features were also used in [62] with an improvement by using a statistical scheme to reduce the effect of human body upper limb activities, and a more common k-nearest neighbour classifier was applied for posture classification purpose. A fall activity was then confirmed if the time difference between a 'stand' posture and 'lie' posture is less than a threshold. This is determined by statistical hypothesis testing for distinguishing a fall event from lying down. In the experiment, it was presented that the obtained threshold for fall confirmation is 0.4s and a correct detection rate of 84.44% is obtained on fall detection and lying down event detection according to their experimental results.

In [64], B. Ni et al. proposed a computer vision based fall prevention system for hospital ward application. A Microsoft sensor which can obtain the colour and depth information was applied; motion features and shape features, such as motion history image (MHI), histogram of oriented gradients (HOG) and histogram of optic flows (HOF) from both colour and depth image sequences were extracted. These were then fused via a multiple kernel learning framework [65] for training the fall event detector. Experimental results demonstrated the high

accuracy that can be achieved by the proposed system with an activity recognition accuracy of 98.76%.

B. Mirnahboub et al. [66] proposed a view-invariant fall detection system by using a single camera. The silhouette area extracted from background subtraction and combined with inclination angle was extracted from a video sequence as features. These were then fed into the popularly used SVM for classifying fall activities and non-fall activities. Different kernels were tested in this work and the experimental results on a public dataset showed that the polynomial kernel of 2^{nd} degree can achieve the best performance with 100% fall detection rate and less than 1% of mistaking non-fall activities as falls.

The extracted features from short video sequences can be used to build an hidden Markov model (HMM) for activity recognition to detect fall activities. For [67], a bounding box and motion information were extracted from consecutive silhouettes as features. These features were then used to train HMMs for classifying fall and non-fall activities. Preliminary results were presented by constructing three HMMs for walking, kneeling and falling activities from several training sequences. The most likely state sequence for a particular test sequence can be successfully estimated by the corresponding HMM.

In [68] a method was presented based on short video sequence activity classification. In this work, a novel method was proposed to extract a person's three-dimensional orientation information from multiple non-calibrated video-cameras. Using this extracted orientation information from short video sequence; an improved version of HMM, layered hidden Markov model (LHMM) was trained and used to test falls. The experimental results on falling and walking sequences showed that a

fall detection rate of 98% can be achieved by using two cameras, with no walking activities mistaken as falls.

Htike et al. [69] presented a vision-based framework that could detect falls using a single video-camera, irrespective of the viewpoint of the camera with respect to the subjects. The proposed system made use of invariant pose models which performed view-invariant human pose recognition by using the chord distribution of the resampled points along the contour of the extracted foreground region. Based on the chord distribution information, inference with an expectation-maximization algorithm was performed on an ensemble of pose models and the probability value that the given frame contained a corresponding pose was then calculated. The system finally detected falls by analysing a sequence of frames using a fuzzy hidden Markov model (FHMM) based on the estimated pose probability values for every frame. This system achieved a 94.1% success rate when it was tested on a challenging multiple view dataset. A multi-camera based HMM approach was proposed in [49], where in projection histogram features were extracted from every single frame for posture recognition by a posture probabilistic template, the results were then fed into an HMM model which exploited the temporal coherence of the postures for detecting falls for an acquired sequence. Multiple calibrated cameras were used to transfer the appearance information to solve the initial occlusion problem when the person passes to another monitored room.

Apart from the features extracted from postures or short video sequences, some other features can also be applied to construct the corresponding supervised classifiers.

In [70], Mihailidis et al. used a single camera to classify fall and non-

fall activities. Carefully engineered features, such as silhouette features, lighting features and flow features were extracted to allow the system to be robust to lighting, environment and the presence of multiple moving objects. Three pattern recognition methods were compared (Logistic Regression (LR), Neural Network (NN) and SVM); the NN technique achieved the best performance with a fall detection rate of 92% and a false detection rate of 5%.

Foroughi et al. [71] proposed a new method for fall detection based on human shape variations using an multi-class support vector machine (MCSVM). Several new features extracted from segmented foreground were used to detect fall and other actions. A combination of best fit approximated ellipse around the human body, projection histograms of the segmented silhouette and temporal changes of head pose were used to obtain useful clues for detection of different behaviors. Extracted feature vectors were fed to an MCSVM for precise classification of motions and determination of a fall event. A reliability rate of 88.08% was achieved in the experimentation. Although this approach did not need tight clothes to be worn, the system had restricted functionality due to the occlusion problem.

Three-dimensional features were applied in [67] by constructing a three-dimensional voxel person from multiple calibrated cameras. Based on the extracted three-dimensional features (including the three-dimensional centroid and orientation information), Anderson proposed a fuzzy logic based linguistic summarisation for fall detection. A hierarchy of fuzzy logic was used, where the output from each level was summarized and fed into the next level for inference. Corresponding fuzzy rules were designed under the supervision of nurses to ensure that they reflect the

manner in which elderly people perform their activities. The proposed framework was extremely flexible and rules can be modified, added, or removed to allow for per-resident customization. This system was tested on a dataset which contained 14 fall activities and 32 non-fall activities, all the fall activities were correctly detected and only two non-fall activities were mistaken as fall activities (100% fall detection rate and 6% false detection rate), which showed an acceptable level of performance.

The main problem for supervised fall detection methods is that they do not provide a person-specific solution for individuals. A large dataset needs to be constructed initially (which should contain the data collected from many people in different views) for a supervised fall detection system, if a person does not fit the dataset very well (such as if he/she is obese), a good performance can definitely not be obtained for this specific person. Moreover, supervised fall detection methods will be affected by occlusions which happen in a real home environment.

2.4 Comparison of the fall detection methods

Both intelligent vision and non-vision techniques have different strengths and weaknesses. The following Table 2.1 summaries the advantages and disadvantages of the two techniques of fall detection.

2.5 Summary

In this chapter a pertinent review of fall detection techniques and certain existing products has been presented. It focuses on the two main categories: intelligent vision and non-vision based methods.

Table 2.1. Comparison of the two approaches of fall detection

Sensor description	Definition	Equipment	Merits	Demerits
Non-vision based methods	User wears some devices and multiple installed sensors to detect posture, sound, vibration and motion	Accelerometer, gyroscope and/or pressure sensor to obtain users location	Cheap; easy to set up and non-intrusive	High rate of false alarm; the sensor is intrusive
Intelligent vision based methods	Computer vision techniques applied to the data captured by video-cameras	Single or multiple digital-cameras	Monitor multiple events simultaneously; less intrusive; the recorded video for remote and post verification	The accuracy is very sensitive to lighting condition, dim light in the night with poor performance. There may be a privacy issue

For non-vision based methods, the focus was mainly on accelerometer, gyroscope and acoustic or vibration based sensors. The principle of fall detection based on these sensors is that a fall has a different pattern of motion data from other activities.

Intelligent vision based methods are more in use nowadays; because they avoid the drawback of non-vision based methods. They use digital video-cameras for image capture. The issue of privacy is not covered in this thesis. However, they still suffer from the problem that accuracy is very sensitive to lighting condition, shadow and similar colour, which needs to be solved by modern intelligent computer vision techniques. Therefore, for the current intelligent vision based techniques, three main problems exist:

1. For the analysis of image sequences involving humans body shape needs to be improved in order to obtain robust person extraction in an indoor environment for fall detection system.

2. Most of the two-dimensional features used in the fall detection

works are not invariant to directions; either direction invariant three-dimensional features need to be used or two-dimensional features captured from different directions should be used to build a supervised classifier model which is invariant to directions.

3. For the posture classification methods for fall detection, an improved classifier should be applied for achieving a better posture classification performance. Besides, the current posture classification based methods are not easy to distinguish fall activities from fast lying activities. Additional information, such as floor region information is needed to distinguish these two activities.

Different techniques are proposed in the next contribution chapters to solve these problems with the aim of achieving more robust fall detection methods for better detection performance.

Chapter 3

BACKGROUND SUBTRACTION TECHNIQUES

3.1 Introduction

In this chapter background subtraction (BGS) techniques will be discussed in detail, including the approximate media filter (AMF), mixture of Gaussians (MoG), and codebook (CB) methods. In addition, a selective updating technique is introduced for adapting the change of the background model to ensure that the human body region will not be absorbed into the background model when static for prolonged periods of time.

Extracting moving objects from sequences of images or video is an interesting problem and remains a challenging topic in intelligent computer vision. The aim of the chapter is to develop a robust video-camera processing technique that reliably extracts the region of a person in the recording of a room environment based on BGS techniques. BGS is the first step when implementing a detection algorithm as mentioned in [72] and [73]. It is the process of generating a foreground mask for

each frame that flags pixels that are considered to be foreground. Foreground in this context refers to any set of pixels that do not conform to the background scene for a given frame. The challenge when selecting a BGS algorithm is to find a solution that best suits the video scene in question. The main performance parameters when selecting an algorithm are the ability to cope with changing background scenes (moving furniture for example), the ability to cope with sudden and gradual changes in illumination and the ability to reliably detect a foreground object even if it has been static for a significant length of time.

Since the background scenes in this thesis are indoor environments, the foreground is likely to be static for prolonged periods of time (as the person sleeps or is seated on a sofa for example) and the background model is not subject to change to the same extent as some outdoor scenes such as in a car park [74].

In the following section, three methods of BGS are explained, each of which has been implemented in the MATLAB environment. Each method is objectively BGS analysed and a selective updating technique is proposed for adapting the background model change. Some post-processing steps, such as noise-removal, holes-filling and shadow-removal [75] required for each method to improve the quality of the BGS result which will be used in this research work for given environments are chosen.

3.2 Background subtraction techniques, review and implementation

Much research has been carried out in the field of background subtraction techniques [76] and these are summarised in the following three

subsections. Specific methods have their own range of computational complexity and their differing algorithmic approaches to background modelling and foreground detection however, each method follows the same basic steps. An initial background model is produced using a training sequence before the video is processed. Then on a per-frame basis, the background model is modified over time to reflect changes in the background scene. Each frame is compared to the running background model in order to extract the estimated foreground. As in [72], in this chapter $I_t(x, y)$ and $B_t(x, y)$ denote the intensity of a pixel with spatial coordinates (x, y) within the input video frame and the background model respectively at time t. The following three methods are the most common methods for BGS [76] each with their own chosen strengths and weaknesses for a given environment:

- Approximate median filter (AMF)
- Mixture of Gaussians (MoG)
- Codebook method (CB)

In the following subsection, the different methods of BGS techniques will be discussed in detail to enable the selection of a suitable method which can be applied in an indoor residential environment.

3.2.1 Approximate median filter

The AMF method of BGS which was proposed by McFarlane and Schofield in [77] corresponds to a simple recursive filter to estimate the median intensity value. This technique has also been used in background modelling for urban traffic monitoring [78]. Median filtering is a nonlinear operation often used in image processing to reduce 'salt'

and 'pepper' noise. A median filter is more effective than a linear convolution based filter when the goal is to simultaneously reduce noise and preserve edges [79].

In [80], Cucchiara et al. computed the median on a set of sub-sampled frames, which increased the ability of detection in the background model. The median filter has the disadvantage that computationally requires a buffer with recent pixel values and there is no deviation measure for adapting the subtraction threshold.

The initial background model consists of a single frame that depicts the background with no foreground objects visible. The background model $B_t(x, y)$, adapts to the video sequence over time by running an estimate of the median which is incremented by one if the input pixel is larger than the estimate, and decreased by one if smaller depending on the current input frame as in equation (3.2.1):

$$B_t(x,y) = \begin{cases} B_{t-1}(x,y) + 1, & I_t(x,y) > B_{t-1}(x,y) \\ B_{t-1}(x,y) - 1, & I_t(x,y) < B_{t-1}(x,y) \end{cases} \tag{3.2.1}$$

Over time, this estimate eventually converges to a value (the median) for which half of the input pixels are larger and half are smaller than [77]. Foreground detection is achieved by subtracting the background model from the current frame and thresholding the result as in equation (3.2.2). A pixel within a frame at time t is detected as foreground if the following conditional is true as in equation (3.2.2) with a threshold T.

$$|I_t(x,y) - B_t(x,y)| > T \tag{3.2.2}$$

A drawback of the AMF is that it adapts slowly toward a large change in background. It needs many frames to learn the new background region revealed by an object that moves away after being stationary for a long time. The AMF is clearly a very simple algorithm and performs extremely well under good lighting conditions, but its inability to cope or handle poorer lighting conditions make it an unsuitable method for indoor application, as presented in the experimental section.

3.2.2 Mixture of Gaussians

The mixture of Gaussians model is among the most fundamental and widely used statistical models for BGS techniques, and was initially proposed in [81] for BGS and since then it has been a popular method of BGS according to the application [72].

According to [72] and [81] the basis of this algorithm is to model the pixel distribution on a per-pixel basis with the sum of K Gaussian probability distributions (referred to as components). The component sets for each pixel represent a set of modalities that correspond to the distribution of a pixel, which can be expressed mathematically as follows:

$$B_t(x,y) = \sum_{i=1}^{K} \omega_{\hat{i},t} \cdot \ \eta(x_t; \mu_{\hat{i},t}, \sigma_{\hat{i},t}) \tag{3.2.3}$$

where K is the number of Gaussian distributions, $\omega_{\hat{i},t}$ is a weight associated to the i^{th} Gaussian at time t with mean $\mu_{\hat{i},t}$ and standard deviation $\sigma_{\hat{i},t}$. η is a Gaussian probability density function:

$$\eta(x_t, \mu_{\hat{i},t}, \sigma_{\hat{i},t}) = \frac{1}{(2\pi)^{\pi/2} \mid \sigma_{\hat{i},t} \mid^{\frac{1}{2}}} \exp\left(\frac{-1}{2} \ \frac{(x_t - \mu_{\hat{i},t})^2}{\sigma^2_{\hat{i},t}}\right) \tag{3.2.4}$$

The weights for a given component sum to unity. The first step in implementing the algorithm is to identify the component of $B_t(x, y)$ which is most closely matched to $I_t(x, y)$. This is referred to as component $\hat{i}$. A component is declared to be a matched component based on the following condition:

$$| I_t(x, y) - \mu_{\hat{i},t-1} | \leq D \cdot \sigma_{\hat{i},t-1} \tag{3.2.5}$$

where D is the deviation threshold with $D > 0$. When a matched component is identified then the parameters of the matched component are updated as follows:

$$\begin{aligned} \omega_{\hat{i},t} &= (1 - \alpha)\omega_{\hat{i},t-1} + \alpha \\ \mu_{\hat{i},t} &= (1 - \rho)\mu_{\hat{i},t-1} + \rho \\ \sigma^2_{\hat{i},t} &= (1 - \rho)\sigma^2_{\hat{i},t-1} + \rho(I_t - \mu_{\hat{i},t})^2 \end{aligned} \tag{3.2.6}$$

where α is the learning rate set by the user as a value between 0 and 1 and ρ is a second learning rate and can be approximately calculated as follows rather than defined by the user:

$$\rho = \frac{\alpha}{\omega_{\hat{i},t}} \tag{3.2.7}$$

This is only when no matched component can be found, and the weights of the components exponentially decay as in:

$$\omega_{\hat{i},t} = (1 - \alpha)\omega_{\hat{i},t-1} \tag{3.2.8}$$

The component with the least weight is replaced by a new component. The next step is to re-configure the weights to sum to unity whether a matched component has been identified or not. Up to this point the process is used to train the background with a series of training frames. These training frames need to make up a copy of the background scene with as little foreground present as possible. The same process is performed on a per-frame basis to modify the background model over time during processing. Foreground extraction is the final step which occurs alongside its model modification process.

With regards to the AMF, the structure involves the simple task of subtracting the background from the current frame and thresholding the result. The process is more complex in MoG than in AMF and starts with the computationally intensive step of ranking all of the components associated with each pixel in order of $\omega_{\hat{i},t}/\sigma_{\hat{i},t}$. This ranking scheme favours components with high weightings and low standard deviations. Once weighted, the first M components that satisfy the following conditional against the weight threshold Γ are declared as background components as in:

$$\sum_{k=i_1}^{i_M} \omega_{k,t} \;\geq\; \Gamma \tag{3.2.9}$$

Finally, $I_t(x, y)$ is declared as foreground if it is at least D times the standard deviation from the mean of any one of the background components.

3.2.3 Codebook method

Besides the AMF and MoG BGS method [82], another popularly used BGS method is the codebook (CB) BGS method [73]. For the CB BGS method, there is no parametric assumption on the CB model and it shows the following merits as proposed in [73]: (1) resistance to artifacts of acquisition, digitisation and compression, (2) capability of coping with illumination changes, (3) adaptive and compressed background models that can capture structural background motion over a long period of time under limited memory, (4) unconstrained training that allows moving foreground objects in the scene during the initial training period.

As for the AMF BGS and the MoG BGS method, the CB BGS algorithm is also a pixel-wise method, and each pixel is modelled by a number of codewords. One codeword $\mathbf{c}$ is composed of an RGB vector $\mathbf{v} = (\bar{R}, \bar{G}, \bar{B})$ and a 6-tuple $\mathbf{aux} = \langle \check{I}, \hat{I},\ f,\ \lambda,\ p,\ q \rangle$. The meanings of the elements in the 6-tuple **aux** are shown in Table 3.1:

Table 3.1. The meanings of the elements in the tuple.

$\check{I}, \hat{I}$	The min and max brightness of all pixels assigned to this codeword
f	The frequency with which the codeword has occurred
λ	Maximum negative run-length (MNRL) defined as the longest interval during the training period that the codeword has not recurred
p, q	The first and last access times, respectively, that the codeword has occurred

The CB BGS algorithm is divided into model training and BGS process. For model training, the codewords used to model every single pixel will be obtained from a training sequence. The procedure for constructing

the codewords for a particular pixel is shown in Table 3.2. Initially, the

Table 3.2. The training procedure for constructing the codewords for a pixel.

(I). L ←0, $\Phi \leftarrow \phi$(empty set)

(II). For t=1 to N do
(i) $\mathbf{x}_t = (R, G, B)$, $I \leftarrow \sqrt{R^2 + G^2 + B^2}$
(ii) Find the codeword $\mathbf{c}_m$ in
Φ (codewords class for a pixel) $= \{\mathbf{c}_i | 1 \leq \mathbf{c}_i \leq L\}$ matching to $\mathbf{x}_t$ based on:
(a) Colourdist($\mathbf{x}_t$,$\mathbf{v}_m$)$\leq \varepsilon_t$
(b) Brightness(I,$\langle \hat{I}_m, \check{I}_m \rangle$)=true
(iii) If $\Phi = \phi$ or there is no match,then L←L+1.
Create a new codeword $\mathbf{c}_L$ by setting:
•$\mathbf{v}_L \leftarrow (R, G, B)$
•$\mathbf{aux}_L \leftarrow \langle I, I, 1, t-1, t, t \rangle$
(iv) Otherwise, update the matched codeword $\mathbf{c}_m$, consisting of
$\mathbf{v}_m = (\bar{R}_m, \bar{G}_m, \bar{B}_m)$ and
$\mathbf{aux}_m = \langle \check{I}_m, \hat{I}_m, f_m, \lambda_m, p_m, q_m \rangle$, by setting:
•$\mathbf{v}_m \leftarrow (\frac{f_m \bar{R}_m + R}{f_m + 1}, \frac{f_m \bar{G}_m + G}{f_m + 1}, \frac{f_m \bar{B}_m + B}{f_m + 1})$
•$\mathbf{aux}_m \leftarrow$
$\langle min\{I, \check{I}_m\}, max\{I, \hat{I}_m\}, f_m + 1, max\{\lambda_m, t - q_m\}, p_m, t \rangle$.
end for

(III). For each codeword $\mathbf{c}_i$, i=1,.....,L, wrap around λ_i by setting
$\lambda_i \leftarrow max\{\lambda_i, (N - q_i + p_i - 1)\}$, removing the codewords whose λs are larger than a particular threshold.

codewords set for a pixel are set to be empty so that the number is zero. Codewords are constructed and updated by matching the existing codewords with the incoming pixel in the training set. If matched, the matched codeword will be updated and a new codeword will be constructed if there is no match.

Finally, the codeword set is refined by deleting the codewords which do not recur for a certain interval measured by the maximum negative run-length (MNRL) value λ to form a more compact CB model. For a

particular codeword $\mathbf{c}$, it is said to match the incoming pixel $\mathbf{x}_t$ if the following two conditions are met:

$$\begin{aligned} colordist(\mathbf{x}_t, \mathbf{c}) &\leq \varepsilon \\ brightness(I, \langle \hat{I}, \check{I} \rangle) &= true \end{aligned} \tag{3.2.10}$$

where ε is a preset threshold value for comparison, I represents the norm of $\mathbf{x}_t$, $\hat{I}$ and $\check{I}$ are the first two parameters of the 6-tuple **aux** vector of the codeword $\mathbf{c}$. The $colordist(\mathbf{x}_t,\mathbf{c})$ measures the chromatic difference between two colour vectors, which can be calculated as:

$$colordist(\mathbf{x}_t, \mathbf{c}) = \sqrt{\| \mathbf{x}_t \|^2 - \frac{\mathbf{x}_t \cdot \mathbf{v}}{\| \mathbf{v} \|^2}} \tag{3.2.11}$$

where $\mathbf{v}$ represents the RGB vector $\mathbf{v} = (\overline{R}, \overline{G}, \overline{B})$ of codeword $\mathbf{c}$. The $brightness(I, \langle \hat{I}, \check{I} \rangle)$ is defined as:

$$brightness(I, \langle \hat{I}, \check{I} \rangle) = \begin{cases} true & if \quad I_{low} \leq I \leq I_{hi} \\ false & \text{otherwise} \end{cases} \tag{3.2.12}$$

where $I_{low} = \alpha \hat{I}$ and $I_{hi} = min\{\beta \hat{I}, \frac{\check{I}}{\alpha}\}$. In the experimental studies, α and β are fixed to be 0.5 and 2 respectively for BGS.

The CB model training procedure is applied for every pixel are constructed, the trained CB models are then used for BGS, the procedure is shown in Table 3.3.

Sometimes, the background model will change after the training process (due to the movement of furniture, for example) and therefore the corresponding CB model for every pixel should be updated.

For the model updating, an additional model $\hbar'$ called a cache and three

Table 3.3. The codebook background subtraction procedure.

Step I	For each pixel $\mathbf{x}_t$=(R,G,B) (assuming the time instance of the frame is t), calculate the intensity from the (R,G,B) value of a colour image by I$\leftarrow \sqrt{R^2+G^2+B^2}$
Step II	Find the first codeword $\mathbf{c}_m$ from the corresponding CB matching to $\mathbf{x}_t$ based on two conditions: 1) $colourdist(\mathbf{x}_t, \mathbf{c}_m) \leq \varepsilon_2$, 2) $brightness(I, \langle \hat{I}_m, \check{I}_m \rangle) = true$ Update the matched codeword
Step III	If there is no match, then the pixel $\mathbf{x}_t$ is categorised as foreground; otherwise, it is regarded as a background pixel.

parameters $T_{\hbar'}$, T_{add} and T_{delete} are defined. The updating procedure is then described as in Table 3.4.

Table 3.4. The updating procedure for the CB background model.

Step I	After training, the background model $\hbar$ for a pixel is obtained. Create an empty model $\hbar'$ as a cache.
Step II	For an incoming pixel value $\mathbf{x}_t$, find a matching codeword in $\hbar$. If found, update the codeword.
Step III	Otherwise, try to find a matching codeword in $\hbar'$ and update it. For no matching, a new codeword $\mathbf{h}$ is created and added to $\hbar'$.
Step IV	Filter out the cache codewords based on $T_{\hbar'}$: $\hbar' \leftarrow \hbar' - \{\mathbf{h}_i | \mathbf{h}_i \in \hbar', \lambda\ of\ \mathbf{h}_i\ is\ longer\ than\ T_{\hbar'}\}$
Step V	Move to the cache codewords staying for enough time, to $\hbar$: $\hbar \leftarrow \hbar \cup \{\mathbf{h}_i | \mathbf{h}_i \in \hbar', \mathbf{h}_i\ stays\ longer\ than\ T_{add}\}$
Step VI	Delete the codewords not accessed for a long time from $\hbar$: $\hbar \leftarrow \hbar - \{\mathbf{c}_i | \mathbf{c}_i \in \hbar, \mathbf{c}_i\ not\ accessed\ for\ T_{delete}\}$
Step VII	Repeat the process from the Step II.

The CB model is updated so any changes in the background model will be taken as the new background model after certain iterations of the above steps.

3.3 Selectively adaptive modification of background model updating

It should be noticed that for the traditional BGS method, the human body will be absorbed into the background model if he/she stays still after a certain time period due to the updating of the background model. To ensure that humans are not absorbed into the background scene, the background model must be updated selectively. The codewords of the pixels in the human body blob should not be updated but other pixels need to be updated according to Table 3.4 so that this human body region will always be taken as foreground and extracted even when the person is static for a very long time while the background model is updated.

In order to selectively update the background model, head tracking is performed to recursively estimate the location of a person's head in each frame. The location of the head is then used to determine the human body blob (which includes the head's location). The tracking problem can be treated as a statistical estimation problem where the dynamics of the human head are estimated based on a sequence of noisy measurements. The particle filter has been employed to perform the estimation recursively, which is explained more comprehensively in [83].

Two types of models are applied in the particle filter, they are state model and measurement model.

3.3.1 State model

A mathematical model of the human head state and transition characteristics between frames must be defined. The model used in this research work is an elliptical one based on [84], which is described as:

$$\begin{aligned} \mathbf{x}_k &= A\mathbf{x}_{k-1} + \mathbf{w}_k \\ \mathbf{z}_k &= H\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{3.3.1}$$

where $\mathbf{x}_k$ is the models state vector at sample k, A is the state transition matrix, $\mathbf{z}_k$ is the measurement vector at sample k whose relationship to $\mathbf{x}_k$ is defined by measurement matrix $H.\mathbf{w}_k$ and $\mathbf{v}_k$ are additive noise vector variables that model the process and measurement noise respectively.

A simple ellipse is used to model the head with fixed size for a given person. The ratio of the minor to major axes of the ellipse is fixed at $1 : 1.2$. The state vector $\mathbf{x}$ is represented as $x = [x, y, \dot{x}, \dot{y}]^T$ where x and y are the pixel coordinates of the ellipses centre and $\dot{x}$ and $\dot{y}$ are respectively the horizontal and vertical velocity components of the ellipse in $pixels/frame$. The state transition behaviour is assumed to be simply the previous state coordinates plus the effects of the velocity from the previous time step. This gives the state transition matrix:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3.2}$$

3.3.2 Measurement model

Tracking methods usually involve measuring the similarity between a target model and one or more hypothesised models in order to decide which hypothesis is more likely to be the real state in the new frame. For measuring the similarity, multiple measurement cues exist and for the particular head tracking problem the mostly used ones include the intensity gradient [84] and colour histogram [85]. In this work, these two measurement cues are applied together to increase the tracker's robustness.

3.3.2.1 Intensity measurement

The intensity gradient measurement analyses the perimeter pixels of a hypothesised elliptical head model and compares it to that of the target.

Figure 3.1 shows how the intensities of the perimeter pixels of an hypothesis and a target model can be treated as vectors $T[i]$ and $H_n[i]$ respectively where i is the perimeter pixel index and n is the index given to different hypotheses. To determine a measure of similarity between $T[i]$ and $H_n[i]$, the $min(.)$ function is used. This function chooses the minimum value of the two vectors for each pixel. The scalar measure of similarity with respect to intensity ϕ_{ig} is then given by equation (3.3.3) where N represents the number of perimeter pixels. This coefficient of similarity is bounded by 0 and 1.

$$\phi_{ig} = \sum_{i=1}^{N} \frac{min(T[i], H_n[i])}{T[i]} \tag{3.3.3}$$

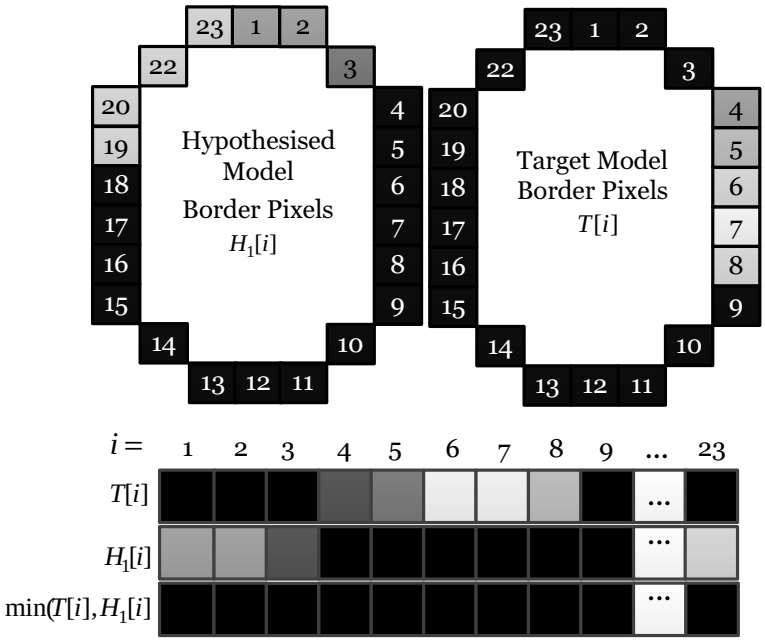


Figure 3.1. Intensity vectors of hypothesis and target elliptical models.

3.3.2.2 Colour histogram measurement

The colour histogram module analyses the interior pixels bounded by a hypothesised elliptical head model and compares it to that of the target. The histogram is produced by assigning each of the interior pixels of a given ellipse to a bin in three-dimensional RGB colour space.

Once a target histogram $U[i]$ and the n^{th} hypothesis histogram $G_n[i]$ have been computed, they can be compared in a very similar way to the intensity. The scalar measure of similarity with respect to colour distribution ϕ_{cd} is given by equation (3.3.4) where N represents the number of bins in the three dimensional histogram (for example, if the histogram was size 2-by-4-by-4 then $N = 32$) and $i = [1, 2, ..., N]$. This coefficient of similarity is bounded by 0 and 1 to give a normalised

comparison between this and ϕ_{ig} as equation (3.3.3):

$$\phi_{cd} = \sum_{i=1}^{N} \frac{min(U[i], G_n[i])}{U[i]} \tag{3.3.4}$$

3.3.3 Particle filter tracker

Based on the state model and measurement model, the particle filtering technique is applied for head tracking purpose. The objective of the particle filter, is to estimate the posterior state distribution $p(x_k \mid z_{1:k})$ which in this application yields an estimated head location in the current frame. The particle filter is a recursive Bayesian approach to state estimation. It is able to incorporate elements of non-linearity and non-Gaussianity into the tracking process, which makes the particle filter a valuable estimator for many modern applications where the restrictive assumptions of the Kalman Filter cannot be applied [86].

The principle behind the particle filtering approach is to represent the posterior state distribution $p(x_k \mid z_{1:k})$ as a sum of weighted particles, where each particle is a hypothesised state. This point-mass approach to representing probability distributions along with its recursive operation makes the particle filter a sequential Monte Carlo method.

For this work, the i^{ih} particle at sample time k is represented by the vector S_k^i with the form $[x, y, \dot{x}, \dot{y}]^T$ where x and y are the spatial coordinates of the elliptical head model and $\dot{x}$ and $\dot{y}$ are respectively the horizontal and vertical velocity components of the ellipse in $pixels/frame$. The i^{th} weight at sample time k is represented by the scalar W_k^i. The weight of each particle can be thought of as a measure of how likely it is that a hypothesised state is correct. When N particles are summed together in weighted delta function form the particles approximate the

posterior distribution without making assumptions about the distribution's shape as shown below:

$$p(x_k \mid z_{1:k}) \approx \sum_{i=1}^{N} W_k^i \delta(x_k - S_k^i) \tag{3.3.5}$$

where $\sum_{i=1}^{N} W_k^i = 1$. The particle filter algorithm for head tracking can be split into four main steps which are repeated for every frame of the video sequence:

1. Select particles from the previous time step.

2. Propagate particles into the current time step according to state transition model.

3. Measure the current frame at each and every particle to assign a new weight to each particle.

4. Estimate the head location based on the distribution given by the newly drawn particles and their associated new weights.

3.3.3.1 Select particles

The particle filter is implemented within the sequential importance sampling (SIS) framework. This means that a resampling operation is carried out at every time step which takes into account the importance of each particle in the previous time step. Drawing N particles means that particles with high weighting are likely to be resampled whereas particles with a low weight are likely to be lost in the resampling process. This has two important functions. Firstly, it reduces wasted computation on particles (hypothesised elliptical head states) that are not at all likely to be similar to that of the target. Secondly, it militates against

the scenario where all but one of the particles is given negligible weight after several successive iterations.

3.3.3.2 Propagate particles

Since all of the particles that have been drawn are from the previous time step, they must be propagated using the model dynamics defined by equation (3.3.1). This automatically takes into account the motion of the head since velocity forms part of the state model. The additive noise is given by multivariate Gaussian random variable r_k:

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + r_k \tag{3.3.6}$$

3.3.3.3 Measurements

A new set of particle weights is now assigned by observing the frame at each hypothesised state (given by each particle) using the intensity gradient and colour histogram cues described in Section 3.3. Weights are calculated independently based on each measurement cue and are then combined to give a single scalar weight for each particle. The i^{th} weight as a result of each cue at sample time k is calculated as follows:

$$W^i_{ig_k} = \frac{1}{\sqrt{2\pi\phi_{ig}}} e^{-\frac{1-\phi_{ig}}{2\sigma^2_{ig}}} \tag{3.3.7}$$

$$W^i_{cd_k} = \frac{1}{\sqrt{2\pi\phi_{cd}}} e^{-\frac{1-\phi_{cd}}{2\sigma^2_{cd}}} \tag{3.3.8}$$

$$W^i_k = \sqrt{(W^i_{ig_k})^2 + (W^i_{cd_k})^2} \tag{3.3.9}$$

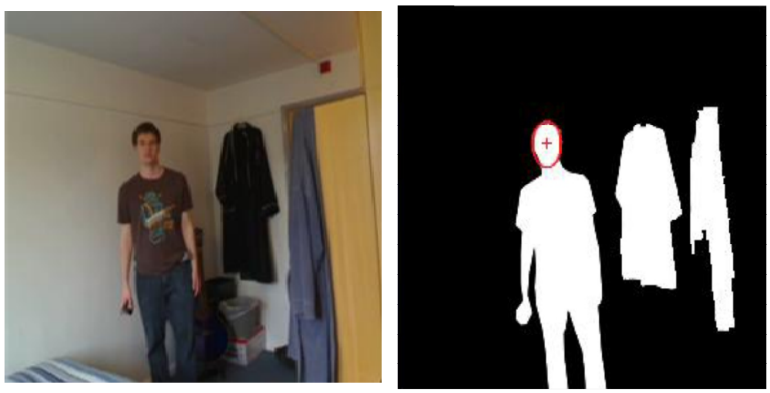
where ϕ_{ig} and ϕ_{cd} are as in Section 3.3, $W^i_{ig_k}$ and $W^i_{cd_k}$ are the weights as a result of the intensity gradient and colour histogram cues respectively and $\sum_{i=1}^{N} W^i_k = 1$. These new weights and their associated particles represent the a posteriori distribution as in equation (3.3.5).

3.3.3.4 Estimate

The estimate of the head's location is simply the expected value of the posterior distribution:

$$E\{p(x_k \mid Z_{1:k})\} \approx \sum_{i=1}^{N} W^i_k S^i_k \tag{3.3.10}$$

This concludes the head tracking process using the particle filter.
After the head tracking, the head's position in the two-dimensional image plane is estimated, the blob which contains the head's position is then taken as the human body blob and will not be updated.
As presented in Figure 3.2 if a person remains static after taking off the clothes, after the selective updating, the human body blob (which contains the head) is not absorbed into the background model while the non-human blobs (clothes) are absorbed into the background model after a certain time period. One important problem with head tracking is initialisation. The initial position of the human head should be known for successive tracking in later frames. This can be done either manually or by adapting particular head detection algorithms such as in [87] and [88]. The later approach is adopted in this work.



(a) (b)

(c)

Figure 3.2. The illustration of selective updating. (a) The original image. (b) The BGS result, the tracked head is contained in the human body blob. (c) The non-human blobs are absorbed into the background model after a certain time while the human blob still remains as the foreground.

3.4 Post-processing techniques

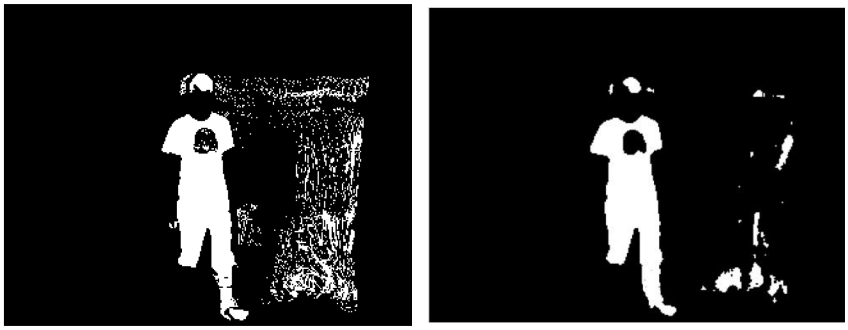
When a BGS method is applied to a video sequence, the result is binary; a zero value represents background model where as a unity value represents foreground region. It should be noted, however that inescapably some foreground pixels are detected as background pixels and vice-versa and therefore post-processing is introduced to increase the accuracy of the output. The survey by Parks et al. [89] provides a useful summary of post-processing techniques and some of them are applied here to

improve the BGS results.

3.4.1 Median filtering

An effective way of excluding inconsistent values within the set and reducing 'salt' and 'pepper' noise is to take the median value of the set of pixels. This type of filtering is an ideal candidate for the CB algorithms post-processing [54].

To solve this problem a 5×5 window is applied to the set of pixels to perform the median filtering. If half of the pixels in the windows have '0' values, then the pixel value in the centre of the window is also set to '0'. In this way, some small 'salt' and 'pepper' noises will be removed, as seen in Figure 3.3.



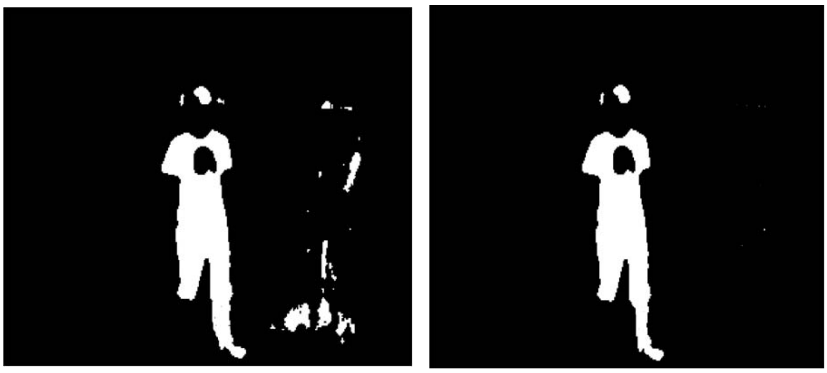
(a) (b)

Figure 3.3. A 5 x 5 median filter applied to the BGS output showing considerable success in removing 'salt' + 'paper' noise. (a) BGS output. (b) Median Filtering.

3.4.2 Advanced post-processing techniques

To further improve the BGS result, some advanced post-processing techniques can also be used. One of these methods is called 'blobs' post-processing. Essentially, it classifies the image as a number of segments of foreground or 'blobs' representing different groups of foreground pix-

els which are 4-connected or 8-connected [75]. If the distance between two blobs is less than a threshold, these two blobs will be taken as one. A threshold is set and the blobs whose pixel numbers are less than the set threshold are removed and the pixels in these blobs are taken as background pixels. One example is seen in Figure 3.4, from which it can be seen that by the 'blobs' post-processing, further improvement of the BGS result is obtained and all the 'noisy blocks' are removed.



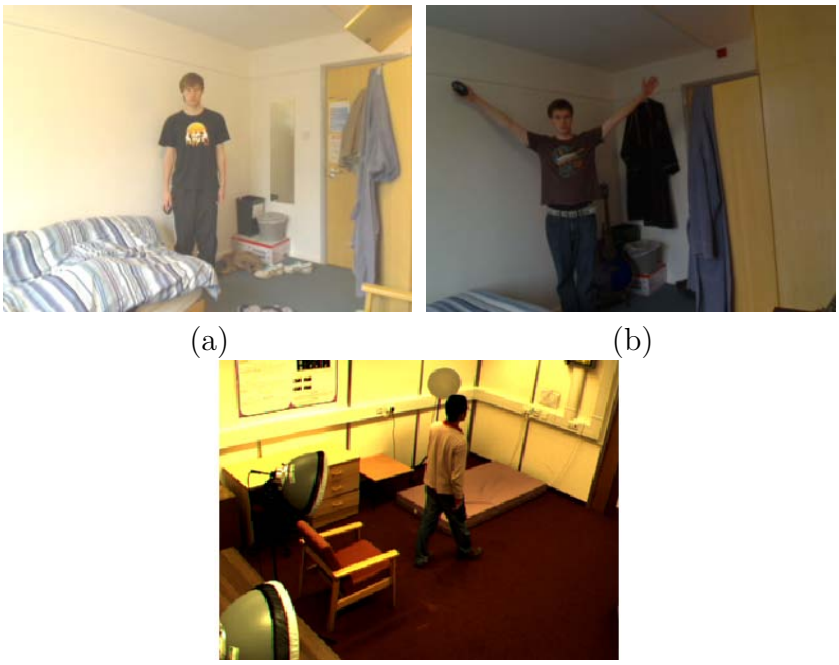
(a) (b)

Figure 3.4. Applying blob post-processing to the CB output video. (a) BGS output video. (b) Blob post-processing technique.

3.5 Performance analysis of background subtraction techniques

This section will focus on performance analysis of the BGS techniques which have been mentioned in Section 3.2. For comparing the performance of three BGS algorithms, three datasets were used in a room environment. Frames samples from datasets 1, 2 and 3 are shown in Figure 3.5. To test the performance of the BGS techniques different real video sequences from a single video-camera in an enclosed environment are used to represent one person in a room environment and under different light conditions (good, poor and typical lighting conditions).

These colour datasets contain around 50 video frames of size 320×240. The criteria for comparison are based on speed and accuracy.



(a) (b)

(c)

Figure 3.5. Different sample of datasets used. (a) Dataset 1 in good lighting conditions. (b) Dataset 2 in poor lighting conditions. (c) Dataset 3 in typical lighting conditions.

3.5.1 Speed comparison of different background subtraction methods

Table 3.5 shows the comparison of speed for the various techniques. The AMF BGS method just compares the values between the current frame and background model for a particular pixel, as shown in [76], the complexity is defined as O(1). The MoG method has $O(m)$ time complexity for one pixel, with m the number of Gaussian distributions used, typically in the order of 3-5 [91].

Finally, for the CB algorithm which is a quantisation method [73] mod-

Table 3.5. Performance of speed (theoretically) per pixel.

Method	Speed
Approximate median filter (AMF) [90] and [80]	$O(1)$
Mixture of Gaussians (MoG) [91]	$O(m)$
Codebook (CB) [73] [54]	$O(n_c)$

els are built from a long observation of a video-camera sequence [54] and [73] for each pixel. This model, will build a CB consisting of one or more codewords and may be different from pixel to pixel. Therefore, the CB has time complexity as $O(n_c)$, n_c is the total number of codewords.

3.5.2 Accuracies of different background subtraction methods

For evaluating the accuracy two parameters are obtained, true positive rate (TPR) and false positive rate (FPR) for each implementation, and are defined as follows in equation (3.5.1) and (3.5.2) to quantify how well each algorithm matches the ground truth images [73].

$$TPR = \frac{Number\ of\ foreground\ pixels\ correctly\ identified\ by\ the\ algorithm}{Actual\ number\ of\ foreground\ pixels\ in\ ground-truth} \tag{3.5.1}$$

$$FPR = \frac{Number\ of\ foreground\ pixels\ incorrectly\ identified\ by\ the\ algorithm}{Actual\ number\ of\ background\ pixels\ in\ ground-truth} \tag{3.5.2}$$

Figure 3.6 shows the ground truth images manually framed for the actual foreground image [92], in which the foreground is shown in white

and corresponds to the human body.

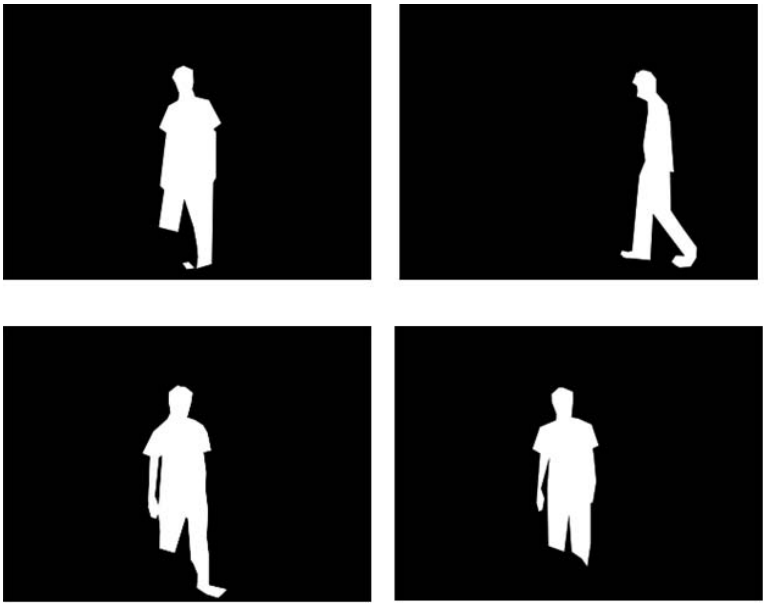


Figure 3.6. Samples of the ground truth images manually generated from actual foreground pixels.

In order to calculate TPR and FPR, knowledge of the true foreground must be known. For example TPR aims to answer the question 'how much of the actual foreground was correctly identified as foreground?' whilst FPR aims to answer 'how much of the areas identified as foreground are truly foreground pixels?' [72].

To achieve this, a process called framing was used to manually generate frames which indicate the foreground of the true areas and these frames could be used as a reference to obtain the corresponding TPR and FPR values.

It is vitally important to obtain results from a variety of data sets so that consistent characteristics of each algorithm can be identified and to enable the performance of the techniques to be examined under different datasets with different lighting conditions.

The Figures in 3.7, 3.8, 3.9, 3.10, 3.11 and 3.12 show the results of the simulation of the three techniques for three different data sets which are explained in detail in the previous subsection.

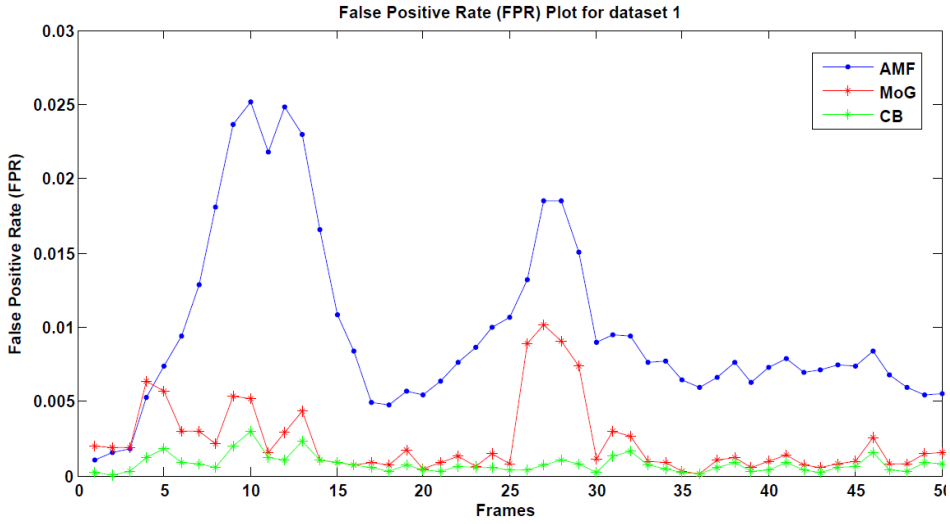


Figure 3.7. Simulation of the AMF, MoG and CB methods of BGS, dataset 1 in good lighting conditions.

Figure 3.8. Simulation of the AMF, MoG and CB methods of BGS, dataset 1 in good lighting conditions.

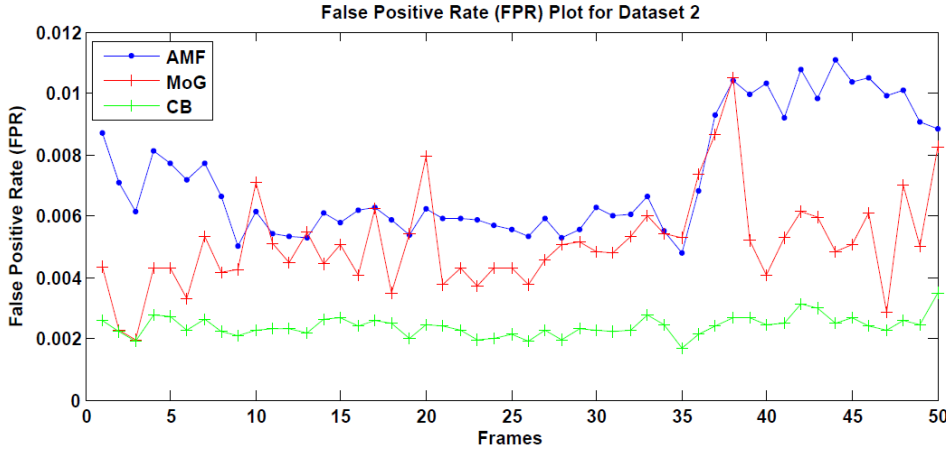


Figure 3.9. Simulation of the AMF, MoG and CB methods of BGS, dataset 2 in poor lighting conditions.

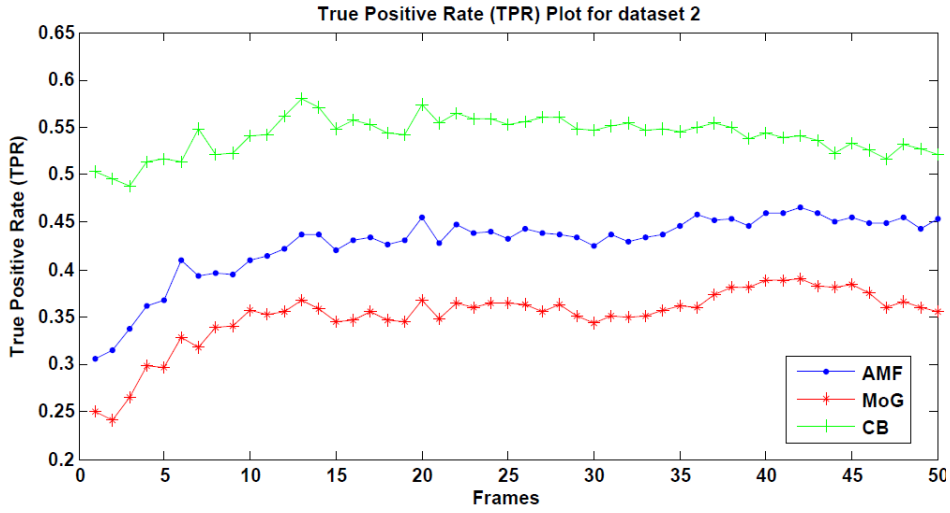


Figure 3.10. Simulation of the AMF, MoG and CB methods of BGS, dataset 2 in in poor lighting conditions.

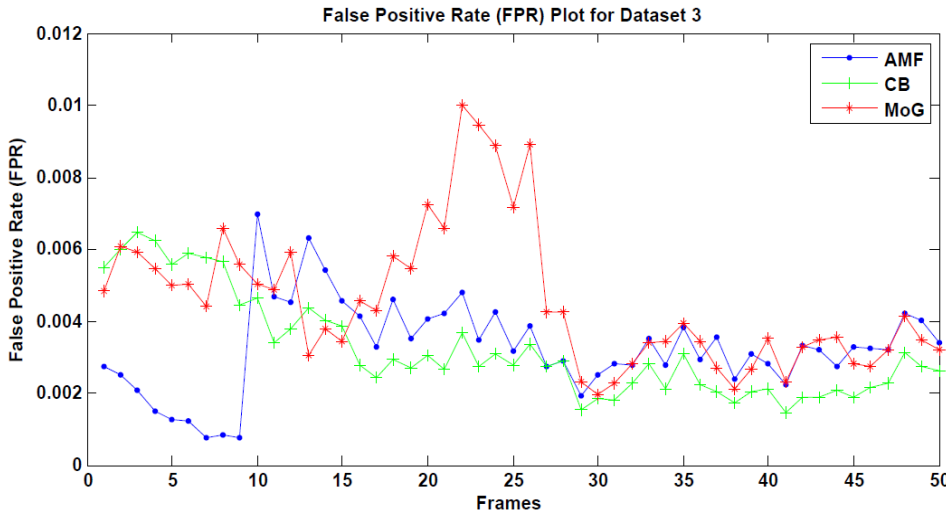


Figure 3.11. Simulation of the AMF, MoG and CB methods of BGS, dataset 3 in typical lighting conditions.

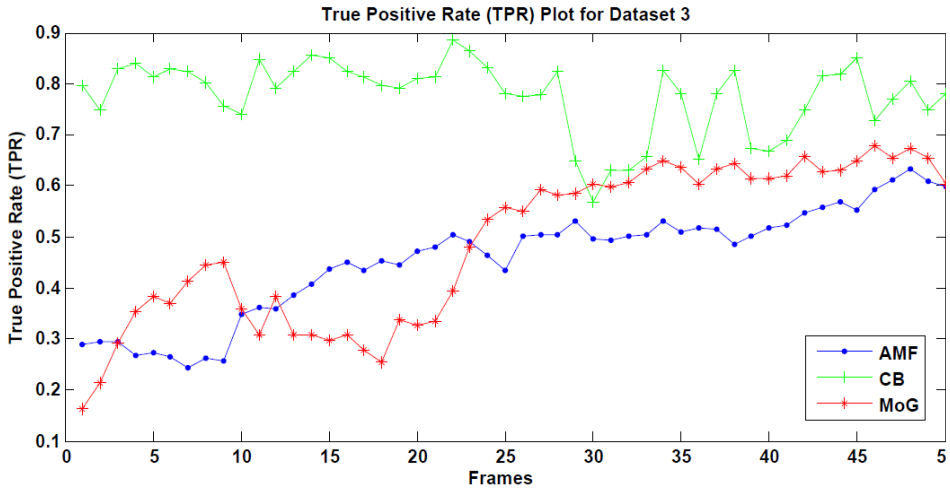


Figure 3.12. Simulation of the AMF, MoG and CB methods of BGS, dataset 3 in typical lighting conditions.

For each dataset evaluation, the parameters for each algorithm are tuned empirically to be optimal and the corresponding post-processing techniques are adopted to improve the BGS results. For a good detection algorithm, it will offer a high level of TPR without sacrificing FPR.

Figures 3.7, 3.8 show the performance of three algorithms on dataset 1 with good lighting conditions, from which it can be observed that the CB method has the lowest FPR curve (Figure 3.7) and highest TPR curve (Figure 3.8) among the three methods. For the other two methods (MoG and AMF), high values of TPR are obtained at the expense of high FPRs. In conclusion, the CB method has better performance than the other two methods on this dataset.

Similar phenomena can be observed on the other two datasets with typical and poor lighting conditions (Figures 3.9, 3.10, 3.11 and 3.12); the CB method always obtains a higher TPR curve and a lower FPR curve than other two methods. This is because the CB method makes use of both the colour information and the intensity information for distinguishing the background model and foreground, which makes the background and foreground pixels more distinguishable than the AMF and MoG methods which only use the intensity information.

Besides, it is also noted that the performance of each algorithm is related to the lighting condition; for the better lighting condition, a higher TPR curve and a lower FPR curve are obtained. As an example, the TPR curve of the CB method fluctuates around 0.8 and the FPR curve is near to zero for the dataset 1 (good lighting condition); however, when it comes to the poor lighting condition case, the value of the TPR curve could drop below 0.6 and higher FPR values (more than 0.005) are obtained. This phenomenon can be explained by the fact that the contrast between the foreground and background model is usually larger under the better lighting condition, making it easier to obtain a good BGS result.

For an overall quantitative analysis, the average TPRs and FPRs for

the frames in all the three datasets are calculated, the results are presented in Tables 3.6, 3.7 and 3.8. From these tables, the advantages

Table 3.6. Average value of TPR and FPR for dataset 1

	TPR	FPR
AMF	0.6368	0.00099
MoG	0.5748	0.0023
CB	0.8139	0.0007

Table 3.7. Average value of TPR and FPR for dataset 2

	TPR	FPR
AMF	0.4272	0.0072
MoG	0.3513	0.0051
CB	0.5416	0.0024

Table 3.8. Average value of TPR and FPR for dataset 3

	TPR	FPR
AMF	0.4556	0.0033
MoG	0.488	0.0046
CB	0.77	0.0032

of the CB method over the AMF and MoG methods can be observed; higher TPRs and lower FPRs are obtained in all the three datasets for the CB method. Considering the CB method's advantages, it is applied in the work for extracting the human body for fall detection throughout this research work.

3.6 Selectively updating evaluation

3.6.1 The evaluation of head tracking

For selectively updating the background model, head tracking is performed to determine the human body blob, some examples of the head tracking results for a video sequence are presented in Figure 3.13 where an, ellipse is used to model the head. The performance of the parti-

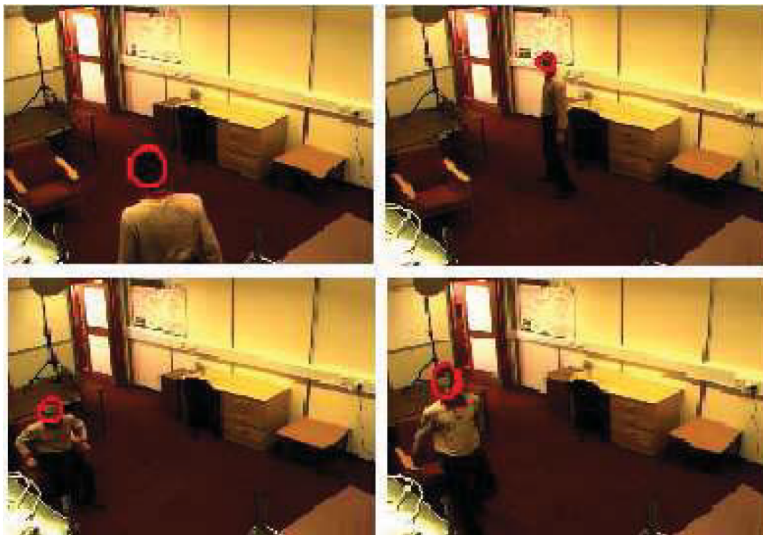


Figure 3.13. Some head tracking results, the tracked head position is tracked by an ellipse.

cle filter is dependent on the number of particles; the performance of the particle filter improves with the increasing number of particles. To investigate how many particles are necessary for this application using the intensity gradient and colour histogram cues, Figure 3.14 was produced which shows the cumulative spatial error between the true head centroid and that of the particle filter estimate. It can be seen that increasing the number of particles N by a factor of 2 from 50 to 100 causes a significant decrease in cumulative error. Increasing by a

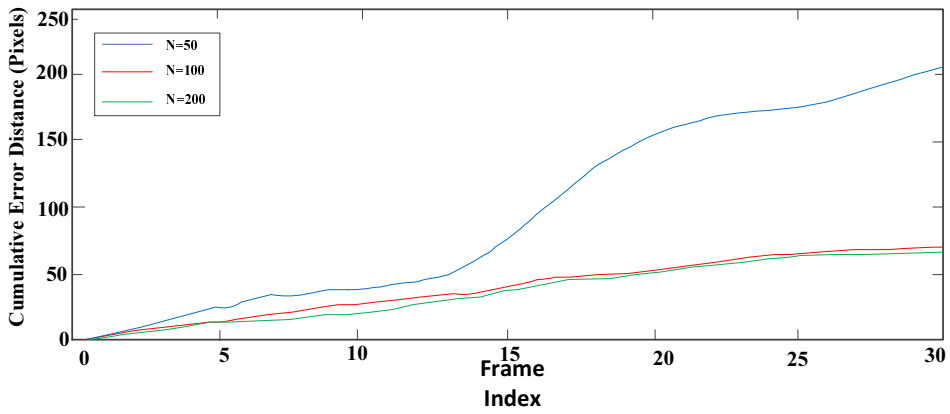


Figure 3.14. Effect of number of particles (N) on head tracking performance

further factor of 2 from 100 to 200 results in much less of a positive impact. This is an important result as it shows the relationship between N and that the error is not a linear one. It also shows that, there is a point at which increasing N does not result in a significant performance advantage. The number of particles used plays a large role in the computational complexity of the filter and so should be set as low as possible whilst maintaining a satisfactory level of error. For this application a value of $N = 100$ was seen as an acceptable engineering trade-off between complexity and performance.

3.6.2 Comparisons of different updating schemes

For comparing different background updating schemes, a video-camera is recorded which shows two people moving in a scene, a chair is moved by one person and after moving the chair, the two persons are static in the scene. The CB BGS method is chosen for BGS because it achieves the best performance for different indoor scenarios from previous analysis. Three different types of CB BGS versions are implemented, they

are: CB BGS without updating (CB), CB BGS with updating (ACB), CB BGS with selective updating (SACB) using head tracking. Figure 3.15 shows some results of the three versions of CB BGS method, the corresponding ground truth images are shown in the bottom line for comparison purpose. From the last column of Figure 3.15, it can be

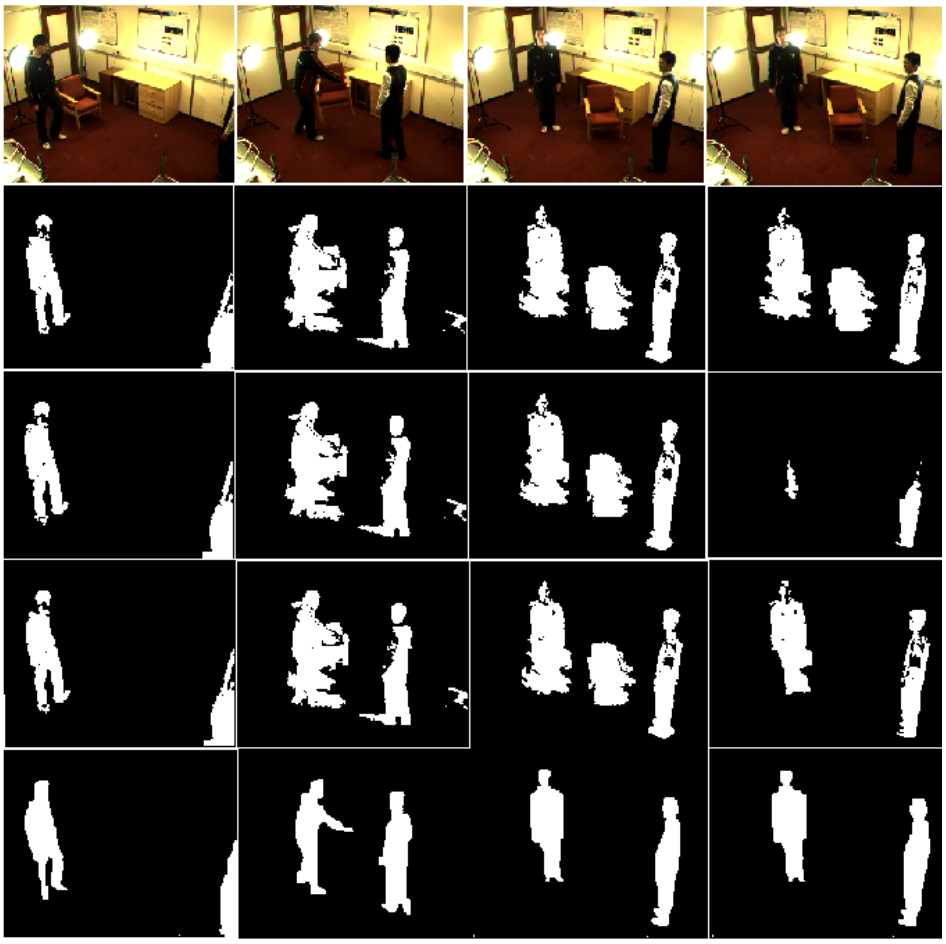


Figure 3.15. The comparison of three CB BGS methods, first row: original image, the second, third, fourth and fifth rows are the results of CB, ACB, SACB and ground truth images respectively.

observed intuitively the SACB method proposed works best. Not only is the chair absorbed into the background model, the two static persons are also successfully segmented.

For an objective analysis, the TPR curves and FPR curves for the three versions of CB BGS method on this video sequence are presented in Figure 3.16 and 3.17, from which it can be seen that initially, these three versions of CB methods work similarly; however, after a certain time interval, the SACB can achieve both higher TPR and lower FPR by selective background model updating.

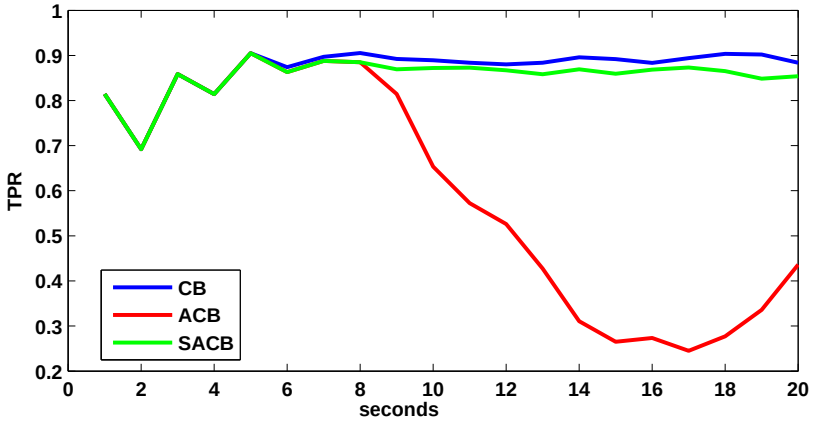


Figure 3.16. The comparison of TPR for three CB methods.

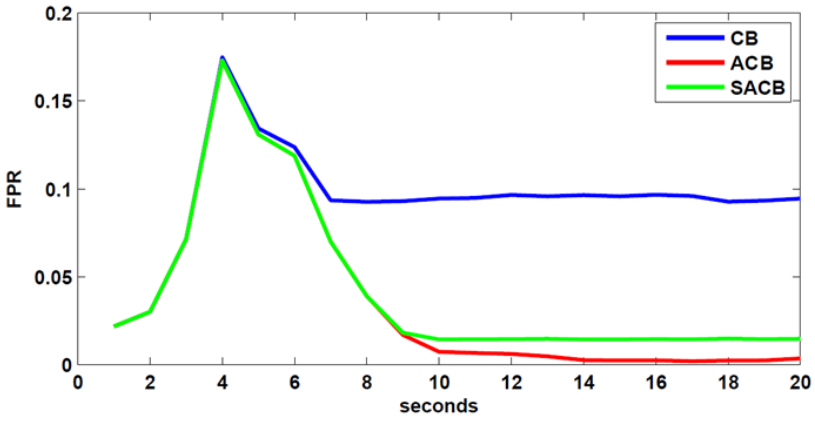


Figure 3.17. The comparison of FPR for three CB methods

3.7 Summary

Different BGS techniques were compared and corresponding improvements were presented in this chapter. Compared with the AMF and MoG methods, the CB BGS method achieved the best performance with the highest TPR and lowest FPR, based on three different datasets under different lighting conditions. This is because compared with the AMF and MoG methods, the CB method makes use of both intensity and colour information, so that the foreground pixels and background pixels will be properly discriminated and better BGS results can be achieved. For this reason, CB method is applied for BGS for fall detection in indoor environment. The complexity of the CB method is related to number of codeword $O(n_c)$ as mentioned in [73], real-time performance can be achieved.

Additionally, with the aid of the head tracking, selective updating of the background model was achieved, which efficiently coped with the change of the background while not taking the long-time static people as the background. As a result, good BGS results can still be obtained after a person is still for a certain time interval, the person will be detected and the background model will be updated to absorb the changes (such as moving a chair/furniture) in the background model.

Chapter 4

VIDEO-CAMERA CALIBRATION BASED ON THE TSAI'S MODEL

4.1 Introduction

Video-camera calibration is an important step in three-dimensional computer vision to obtain certain types of three-dimensional information (such as the three-dimensional position for a particular pixel in the image). The Tsai's model for calibration detailed in this Chapter is the basis of Chapter 5. The derivations of the three-dimensional space line and constructions of three-dimensional voxel person rely upon Tsai's model. As such, the content of this chapter facilitates the derivations of Chapter 5; besides, it also gives readers the corresponding background knowledge about camera calibration. It involves estimating the external and internal parameters of a video camera based on certain camera models, which correspond to various geometries of the image formation process as shown in [93], [94], [95] and [96].

In a general sense, a video-camera model procedure can be regarded as a set of equations involving several video-camera parameters that pro-

vides a mathematical interpretation of the process of projecting a three-dimensional world scene onto a two-dimensional image plane coordinate system. The estimation of these video-camera model parameters can be achieved from experimentally acquired point correspondences (a set of three-dimensional object points and corresponding two-dimensional image points).

There are many different video-camera calibration techniques. In general, existing video-camera calibration techniques can be classified into three types [96]:

1. Linear techniques (direct linear transform model (DLT)) [93] and [94].

2. Nonlinear techniques (Z. Zhang's model) [95].

3. Multiple steps techniques (Tsai's model) [96].

The following Table 4.1 summaries the merits and demerits of the three techniques of video-camera calibration.

To calibrate the uncalibrated video-cameras and extract a three-dimensi onal voxal block from two-dimensional images, Roger Y. Tsai's versatile video-camera calibration algorithm (aptly called the Tsai's model) described in [96] has been used due to its ability to estimate the internal and external parameters separately for an uncalibrated video-camera. To estimate these internal and external parameters of a video-camera, a calibration target with points whose three-dimensional real world coordinates and corresponding two-dimensional image coordinates are known, must be recorded. These real world coordinates and image coordinates are then put in files in an appropriate order before passing on to the calibration routines steps.

Table 4.1. Comparison of the three geometric calibration techniques

Methods	Computational	Accuracy	Merits	Demerits
Linear calibration	More expensive and use expensive equipment such as two or three orthogonal planes	Generates large errors	Widely used for their simplicity and simple implementation and fast operation	Distortion not considered
Non-linear calibration	Use a large number of unknowns and a large-scale nonlinear optimization	Low accuracy	Easy to use; and flexible	In highly distorted environments the optimization may be unstable if the iterative procedure is not appropriately designed
Multiple steps calibration	Computationally complex	Accurate	Can be used with linear and non linear model, possibility to fix the internal parameters	Other types of distortion not considered such as tangential [97]

4.2 Calibration steps based on Tsai's model

This section describes in detail the video-camera calibration model steps based on the Tsai's model [96]. It will also identify and define the calibration parameters. The Tsai's model has internal and external parameters and is based on the pinhole model of three-dimensional to two-dimensional perspective projection with 1^{st} order radial lens distortion. The parameters are as listed below:

Internal parameters:

- Effective focal length of the pinhole video-camera (f).
- 1^{st} order radial lens distortion coefficient (k).
- Scale factor, (s_x) to account for any uncertainty in the frame grabber's resampling of the horizontal scan line.

External parameters:

- Rotation angles for the transformation between the world coordinate system and video-camera coordinate system frames (R_x, R_y, R_z)
- Translational components for the transform between the world coordinate system and video-camera coordinate system (T_x, T_y, T_z) frames.

Tsai's model involves a four step video-camera calibration procedure of transformation from three-dimensional world coordinate system to two-dimensional image coordinate system captured by multiple digital video-cameras, which is based on the video-camera geometry as shown in Figure 4.1 which displays the basic geometry of the video-camera model processing based on Tsai's model.

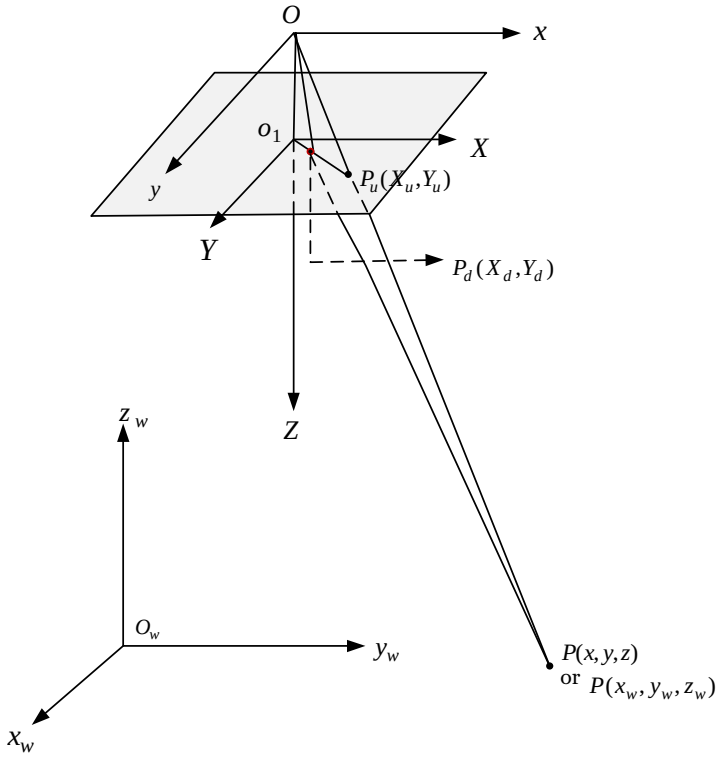


Figure 4.1. Video-camera geometry with perspective projection based on Tsai's model and radial lens distortion [96].

The flowchart in Figure 4.2 shows the four steps of transformation from three-dimensional world coordinate system to video-camera coordinate system.

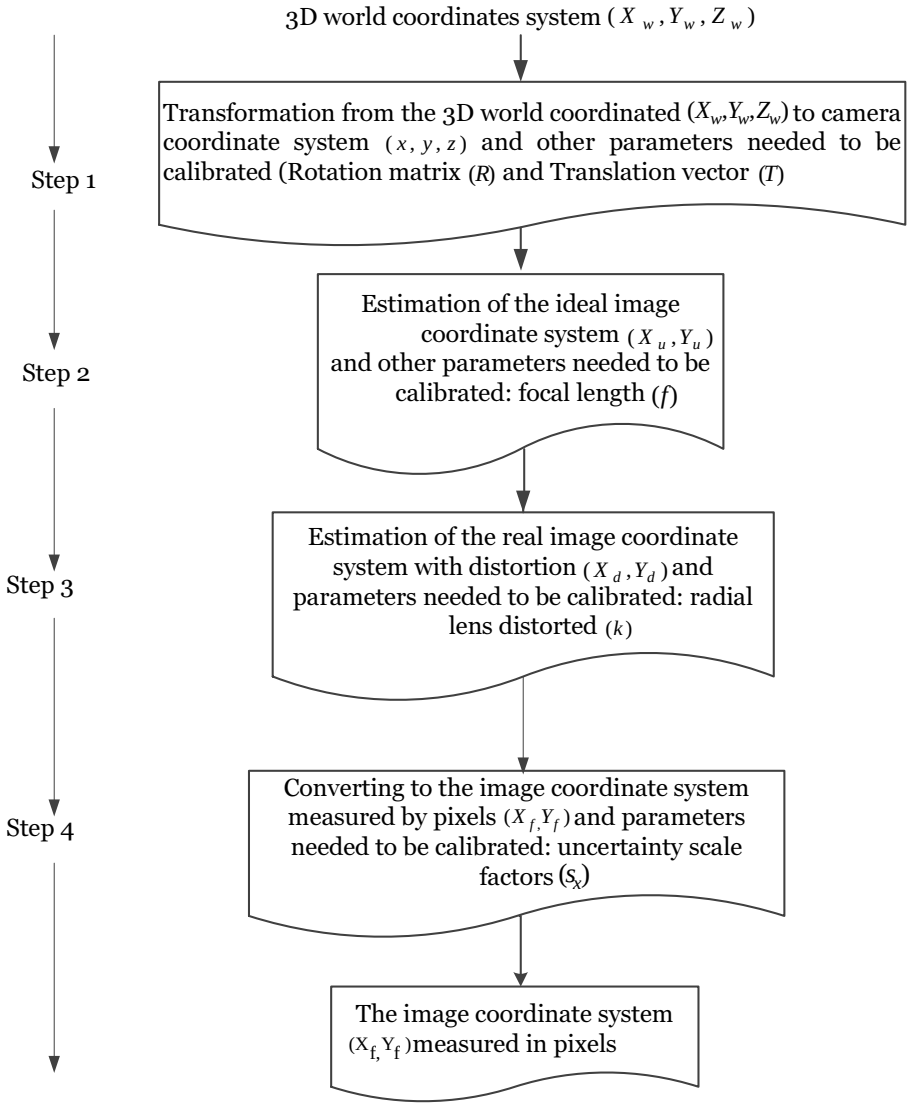


Figure 4.2. The flowchart of converting from the three-dimensional world coordinate to the two-dimensional image coordinate system measured by pixels and the parameters needed to be calibrated [96].

Step 1. For a three-dimensional point, P is the object point in a scene as shown in Figure 4.1 and it has a corresponding three-dimensional world coordinate system (x_w, y_w, z_w) (based on the

three-dimensional world coordinate system (O_w, x_w, y_w, z_w) as in Figure 4.1).

The video-camera itself also has its own coordinate system, which is called the video-camera coordinate system denoted as (O, x, y, z) as shown in Figure 4.1 (with the origin being the lens center of the video-camera).

The world coordinate system can be aligned to the video-camera coordinate system by certain rotating and translating operations.

The relationship between the three-dimensional world coordinate system (x_w, y_w, z_w) and the video-camera coordinate system (x, y, z) of the point P is as shown:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \tag{4.2.1}$$

where R is a 3×3 rotation matrix which can be represented by three rotation angles in three-dimensional space (Euler angles yaw θ, pitch ϕ and tilt ψ) as:

$$R = \begin{bmatrix} \cos(\psi)\cos(\theta) & \sin(\psi)\cos(\theta) & -\sin(\theta) \\ -\sin(\psi)\cos(\phi) + \cos(\psi)\sin(\theta)\sin(\phi) & \cos(\psi)\cos(\phi) + \sin(\psi)\sin(\theta)\sin(\phi) & \cos(\theta)\sin(\phi) \\ \sin(\psi)\sin(\phi) + \cos(\psi)\sin(\theta)\cos(\phi) & -\cos(\psi)\sin(\phi) + \sin(\psi)\sin(\theta)\cos(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \tag{4.2.2}$$

or it can be written in a simpler form:

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \tag{4.2.3}$$

and T is a 3×1 translation vector with:

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{4.2.4}$$

Step 2. After the video-camera coordinate system (x, y, z), of the object point P is obtained, the coordinates system of the pinhole video-camera model [96], that is the image plane coordinate system (X_u, Y_u) of the ideally projected point P_u can be obtained as:

$$\begin{aligned} X_u &= f\frac{x}{z} \\ Y_u &= f\frac{y}{z} \end{aligned} \tag{4.2.5}$$

where f is the focal length of the video-camera, and also measures the distance between the center of lens and image plane.

Step 3. In comparison to the pinhole video-camera model, the real video-camera outputs contain certain types of distortion.

One important distortion and one whose impact is most felt is the radial lens distortion. The lens distortion, the actual object point on the image plane (denoted as P_d) differs from the ideally projected point P_u.

Note that image plane coordinate system P_d is denoted as (X_d, Y_d). As

proposed in [96], the relationship between (X_u, Y_u) and (X_d, Y_d) is:

$$\begin{aligned} X_u &= X_d(1 + kr^2) \\ Y_u &= Y_d(1 + kr^2) \end{aligned} \tag{4.2.6}$$

where $r = \sqrt{X_d^2 + Y_d^2}$ and k is the coefficient of the radial lens distortion.

Step 4. The last step, is to convert the image plane coordinate system (X_d, Y_d) to the image plane coordinate system (X_f, Y_f) measured in pixels with the following transformation:

$$\begin{aligned} X_f &= s_x \dot{d}_x^{-1} X_d + C_x \\ Y_f &= \dot{d}_y^{-1} Y_d + C_y \end{aligned} \tag{4.2.7}$$

where C_x and C_y are the center coordinates system of the captured image, s_x is the uncertainty scaling factor due to camera scanning and acquisition time error, $\dot{d}_x$ and $\dot{d}_y$ represent the corresponding size of a pixel, and are calculated as:

$$\begin{aligned} \dot{d}_x &= d_x \frac{N_{cx}}{N_{fx}} \\ \dot{d}_y &= d_y \frac{N_{cy}}{N_{fy}} \end{aligned} \tag{4.2.8}$$

where d_x and d_y are the center to center distance (CCD) between adjacent sensor elements in the X and Y directions; N_{cx} and N_{cy} are the numbers of CCD sensor elements in the X and Y directions and N_{fx} and N_{fy} are the numbers of image pixels in the X and Y directions.

The four steps highlighted above show the relationship between the three-dimensional real world coordinate system and two-dimensional image pixel coordinate system for a particular point. From these four steps, a three-dimensional point can be converted to a two-dimensional point on the image.

4.3 Parameter estimation for video-camera model

The procedures of estimating the internal and external parameters are shown in the following section.

4.3.1 Estimation of external and internal parameters

The estimations will involve the following parameters, $T_y^{-1}s_x r_1$, $T_y^{-1}s_x r_2$, $T_y^{-1}s_x r_3$, $T_y^{-1}s_x T_x$, $T_y^{-1}r_4$, $T_y^{-1}r_5$, and $T_y^{-1}r_6$.

If points $P(x, y, z)$ and $P_{oz}(0, 0, z)$ are connected (as shown in Figure 4.3), $\overline{O_1P_d}//\overline{P_{oz}P}$ (// means parallel) can be obtained by the fact that $\overline{O_1P_d}$ and $\overline{P_{oz}P}$ are the intersections of the plane (O, P, P_{oz}) and two parallel planes ((O_1, x, y) and (P_{oz}, x, y)). As $\overline{O_1P_d}//\overline{P_{oz}P}$ and $\overline{O_1P_d} \times \overline{P_{oz}P}$=0 ($\times$ denotes the cross product), then:

$$(X_d, Y_d) \times (x, y) = 0 \tag{4.3.1}$$

which yields $X_d \cdot y - Y_d \cdot x = 0$. If equations (4.2.3) and (4.2.4) are substituted into equation (4.2.1), the following is obtained:

$$\begin{aligned} x &= r_1 x_w + r_2 y_w + r_3 z_w + T_x \\ y &= r_4 x_w + r_5 y_w + r_6 z_w + T_y \end{aligned} \tag{4.3.2}$$

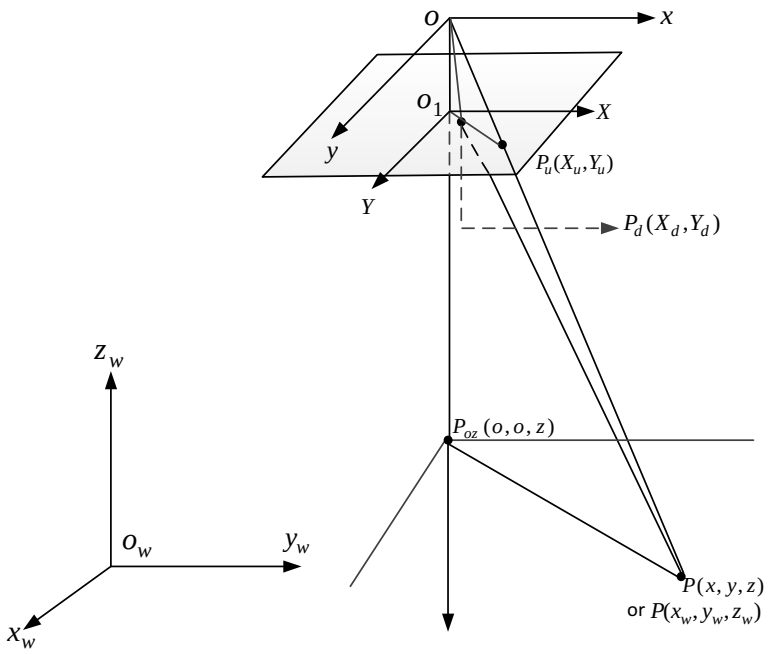


Figure 4.3. $\overline{O_1P_d}//\overline{P_{oz}P}$ by connecting P and P_{oz} by the fact that $\overline{O_1P_d}$ and $\overline{P_{oz}P}$ are the intersections of the plane (O, P, P_{oz}) with two parallel planes (O_1, x, y) and (P_{oz}, x, y) [96].

Thus, using equation (4.3.1), it can be derived that:

$$X_d(r_4x_w + r_5y_w + r_6z_w + T_y) - Y_d(r_1x_w + r_2y_w + r_3z_w + T_x) = 0 \quad (4.3.3)$$

Dividing each side by T_y and after applying some algebraic operations, equation (4.3.4) is obtained:

$$\left[\begin{array}{cccccccc} Y_d x_w & Y_d y_w & Y_d z_w & Y_d & -X_d x_w & -X_d y_w & -X_d z_w \end{array}\right] \cdot \left[\begin{array}{c} T_y^{-1} r_1 \\ T_y^{-1} r_2 \\ T_y^{-1} r_3 \\ T_y^{-1} T_x \\ T_y^{-1} r_4 \\ T_y^{-1} r_5 \\ T_y^{-1} r_6 \end{array}\right] = X_d \tag{4.3.4}$$

Multiplying both sides by s_x and replacing the result $X_d \dot{s}_x$ with $\dot{X}_d$ a new equation (4.3.5) is obtained:

$$\left[\begin{array}{cccccccc} Y_d x_w & Y_d y_w & Y_d z_w & Y_d & -\dot{X}_d x_w & -\dot{X}_d y_w & -\dot{X}_d z_w \end{array}\right] \cdot \left[\begin{array}{c} T_y^{-1} s_x r_1 \\ T_y^{-1} s_x r_2 \\ T_y^{-1} s_x r_3 \\ T_y^{-1} s_x T_x \\ T_y^{-1} r_4 \\ T_y^{-1} r_5 \\ T_y^{-1} r_6 \end{array}\right] = \dot{X}_d \tag{4.3.5}$$

Initially, a set of points is chosen and for each point i, its three-dimensional world coordinate system (x_{wi}, y_{wi}, z_{wi}) and two-dimensional image coordinate system (X_f, Y_f) are known. Using equation (4.2.7), $\dot{X}_{di}$ and

Y_{di} can be calculated as:

$$\dot{X}_{di} = (X_{fi} - C_x)\dot{d}_x$$
$$Y_{di} = (Y_{fi} - C_y)\dot{d}_y \tag{4.3.6}$$

Equation (4.3.5) can be denoted as:

$$\mathbf{a}_i \cdot \mathbf{x} = b_i \tag{4.3.7}$$

where

$$\mathbf{a}_i = \left[\begin{array}{cccccc} Y_{di}x_{wi} & Y_{di}y_{wi} & Y_{di}z_{wi} & Y_{di} & -\dot{X}_{di}x_{wi} & -\dot{X}_{di}y_{wi} & -\dot{X}_{di}z_{wi} \end{array} \right],$$

$$\mathbf{x} = \left[\begin{array}{c} T_y^{-1}s_x r_1 \\ T_y^{-1}s_x r_2 \\ T_y^{-1}s_x r3 \\ T_y^{-1}s_x T_x \\ T_y^{-1} r_4 \\ T_y^{-1} r_5 \\ T_y^{-1} r_6 \end{array} \right], \; b_i = \dot{X}_{di}, \text{ each element of the vector } \mathbf{a}_i \text{ and } b_i$$

are all known. If there are N points, then equation (4.3.8) holds:

$$\begin{bmatrix} \mathbf{a}_1 \\ . \\ . \\ \mathbf{a}_i \\ . \\ . \\ \mathbf{a}_N \end{bmatrix} \mathbf{x} = \begin{bmatrix} b_1 \\ . \\ . \\ b_i \\ . \\ . \\ b_N \end{bmatrix} \tag{4.3.8}$$

and if $\mathbf{A}$ is used to represent $\begin{bmatrix} \mathbf{a}_1 \\ . \\ . \\ \mathbf{a}_i \\ . \\ . \\ \mathbf{a}_N \end{bmatrix}$ which is a $N \times 7$ matrix, and $\mathbf{b}$ is used to represent $\begin{bmatrix} b_1 \\ . \\ . \\ b_i \\ . \\ . \\ b_N \end{bmatrix}$ which is a $N \times 1$ vector then equation (4.3.9) can be obtained:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \tag{4.3.9}$$

and the least squares solution for equation (4.3.9) is:

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \tag{4.3.10}$$

from which the seven parameters:
$T_y^{-1}s_xr_1,\ T_y^{-1}s_xr_2,\ T_y^{-1}s_xr_3,\ T_y^{-1}s_xT_x,\ T_y^{-1}r_4,\ T_y^{-1}r_5,\ T_y^{-1}r_6$ are estimated.

4.3.2 Determination of T_y and s_x

The result of equation (4.3.10) can be used to estimate $|T_y|$. If the i^{th} element of $\mathbf{x}$ is denoted as x_i, $|T_y|$ can be computed as:

$$|T_y| = (x_5^2 + x_6^2 + x_7^2)^{-1/2} \tag{4.3.11}$$

by using $r_4^2 + r_5^2 + r_6^2 = 1$ according to the correspondence between equation (4.2.2) and (4.2.3). The uncertainty scale factor s_x can be estimated from $|T_y|$ as:

$$s_x = (x_1^2 + x_2^2 + x_3^2)^{-1/2}|T_y| \tag{4.3.12}$$

by using $r_1^2 + r_2^2 + r_3^2 = 1$. To determine the sign of T_y an object point i whose two-dimensional image pixel coordinate (X_{fi}, Y_{fi}) is away from the image center (C_x, C_y) is selected; its three-dimensional world coordinate is denoted as (x_{wi}, y_{wi}, z_{wi}). Initially, the sign of T_y is assumed as $+1$ and this point's three-dimensional camera coordinate (x_i, y_i) is computed as:

$$\begin{aligned} x_i &= r_1x_{wi} + r_2y_{wi} + r_3z_{wi} + T_x \\ y_i &= r_4x_{wi} + r_5y_{wi} + r_6z_{wi} + T_y \end{aligned} \tag{4.3.13}$$

where r_1, r_2, r_3, r_4, r_5, r_6 and T_x can be calculated as:

$$
\begin{aligned}
r_1 &= (T_y^{-1} s_x r_1) \cdot T_y s_x^{-1} = x_1 T_y s_x^{-1} \\
r_2 &= (T_y^{-1} s_x r_2) \cdot T_y s_x^{-1} = x_2 T_y s_x^{-1} \\
r_3 &= (T_y^{-1} s_x r_3) \cdot T_y s x^{-1} = x_3 T_y s x^{-1} \\
r_4 &= (T_y^{-1} r_4) \cdot T_y = x_5 T_y \\
r_5 &= (T_y^{-1} r_5) \cdot T_y = x_6 T_y \\
r_6 &= (T_y^{-1} r_6) \cdot T_y = x_7 T_y \\
T_x &= (T_y^{-1} T_x) \cdot T_y = x_4 T_y
\end{aligned} \tag{4.3.14}
$$

Moreover, by using equation (4.2.7), the distorted image coordinate (X_{di}, Y_{di}) can be obtained from (X_{fi}, Y_{fi}). In addition, from the relationship between (X_{di}, Y_{di}) and (x_i, y_i) as shown in equations (4.2.5) and (4.2.6), it can be concluded that X_{di} and x_i, together with Y_{di} and y_i should have the same signs. Therefore, if the calculated (X_{di}, Y_{di}) and (x_i, y_i) meet this condition, the initial assumption is right and the sign of T_y is $+1$; otherwise, the sign of T_y is -1.

4.3.3 Computation of R and T_x

After obtaining T_y, s_x and the solution $\mathbf{x} = [T_y^{-1} s_x r_1,\ T_y^{-1} s_x r_2,\ T_y^{-1} s_x r_3,\ T_y^{-1} s_x T_x,\ T_y^{-1} r_4,\ T_y^{-1} r_5,\ T_y^{-1} r_6]$, some elements of the rotation matrix $(r_1, r_2, r_3, r_4, r_5, r_6)$ and T_x can be obtained in the same way as equation (4.3.14).

The remaining three elements r_7, r_8, r_9 of the rotation matrix R, can be estimated by using the fact that the third row of R can be computed as the cross product of the first two rows according to the form of R

equation (4.2.3), which yields:

$$\begin{aligned} r_7 &= r_2 * r_6 - r_3 * r_5 \\ r_8 &= r_3 * r_4 - r_1 * r_6 \\ r_9 &= r_1 * r_5 - r_2 * r_4 \end{aligned} \tag{4.3.15}$$

4.3.4 Computations of f, T_z and k

The remaining parameters needed to be estimated include focal length f, T_z and radial lens distortion coefficient k. To obtain these parameters, a two step procedure is applied:

Step 1. For a particular point P_i in a set of N points $\{P_1, ..., P_N\}$, combining equations (4.2.5), (4.2.6) and (4.3.2), the following is obtained:

$$\begin{aligned} X_{di}(1 + kr^2) &= f\frac{r_1 x_{wi} + r_2 y_{wi} + r_3 z_{wi} + T_x}{r_7 x_{wi} + r_8 y_{wi} + r_9 z_{wi} + T_z} \\ Y_{di}(1 + kr^2) &= f\frac{r_4 x_{wi} + r_5 y_{wi} + r_6 z_{wi} + T_y}{r_7 x_{wi} + r_8 y_{wi} + r_9 z_{wi} + T_z} \end{aligned} \tag{4.3.16}$$

if k is set to zero, then:

$$\begin{aligned} \begin{bmatrix} x_i & -X_{di} \end{bmatrix} \begin{bmatrix} f \\ T_z \end{bmatrix} &= X_{di} w_i \\ \begin{bmatrix} y_i & -Y_{di} \end{bmatrix} \begin{bmatrix} f \\ T_z \end{bmatrix} &= Y_{di} w_i \end{aligned} \tag{4.3.17}$$

where $x_i = r_1 x_{wi} + r_2 y_{wi} + r_3 z_{wi} + T_x$, $y_i = r_4 x_{wi} + r_5 y_{wi} + r_6 z_{wi} + T_y$ and $w_i = r_7 x_{wi} + r_8 y_{wi} + r_9 z_{wi}$.

Extending equation (4.3.18) to N points, gives

$$\begin{bmatrix} x_1 & -X_{d_1} \\ y_1 & -Y_{d_1} \\ & . \\ & . \\ x_i & -X_{d_i} \\ y_i & -Y_{d_i} \\ & . \\ & . \\ x_N & -X_{dN} \\ y_N & -Y_{dN} \end{bmatrix} \begin{bmatrix} f \\ T_z \end{bmatrix} = \begin{bmatrix} X_{d1}w_1 \\ Y_{d1}w_1 \\ . \\ . \\ X_{di}w_i \\ Y_{di}w_i \\ . \\ . \\ X_{dN}w_N \\ Y_{dN}w_N \end{bmatrix} \tag{4.3.18}$$

Similarly, a least squares solution of $\begin{bmatrix} f \\ T_z \end{bmatrix}$ can be obtained by using equation (4.3.10), with $A = \begin{bmatrix} x_1 & -X_{d1} \\ y_1 & -Y_{d1} \\ & . \\ & . \\ x_i & -X_{di} \\ y_i & -Y_{di} \\ & . \\ & . \\ x_N & -X_{dN} \\ y_N & -Y_{dN} \end{bmatrix}$ which is a $2N \times 2$ ma-

trix and $\mathbf{b} = \begin{bmatrix} X_{d1}w_1 \\ Y_{d1}w_1 \\ . \\ . \\ X_{di}w_i \\ Y_{di}w_i \\ . \\ . \\ X_{dN}w_N \\ Y_{dN}w_N \end{bmatrix}$ which is a $2N \times 1$ vector.

Step 2. In order to obtain more precise estimations of f, T_z and k, an error function $e(f, T_z, k)$ with respect to f, T_z and k is minimized, which becomes:

$$\begin{aligned} e(f, T_z, k) = & \sum_{i=1}^{N} \left(X_{di}(1 + kr^2) - f\frac{r1x_{wi} + r2y_{wi} + r3z_{wi} + T_x}{r7x_{wi} + r8y_{wi} + r9z_{wi} + T_z} \right)^2 \\ & + \sum_{i=1}^{N} \left(Y_{di}(1 + kr^2) - f\frac{r4x_{wi} + r5y_{wi} + r6z_{wi} + T_y}{r7x_{wi} + r8y_{wi} + r9z_{wi} + T_z} \right)^2 \end{aligned} \tag{4.3.19}$$

$e(f, T_z, k)$ is a non-linear function with respect to f, T_z and k and in order to minimise it, some non-linear optimisation methods, such as the steepest gradient method, Gaussian Newton method and Levenberg-Marquardt method (damped Gaussian-Newton method) [98] can be applied.

The optimisation method adopted in the fall detection system is the Levenberg-Marquardt method. The Levenberg-Marquardt method obtained its operating stability from the steepest descent method and

gains its accelerated convergence in the minimum vicinity from the Newton method as shown in [98]. The f, T_z and k values calculated in Step 1 are used as the initial point values for the Levenberg-Marquardt method. By minimising $e(f, T_z, k)$, the difference between the undistorted image plane coordinate system obtained from two-dimensional image pixel and that obtained from three-dimensional real world coordinate system will be small, which indicates an accurate camera model for the correspondence between two-dimensional image pixel coordinate system and three-dimensional real world coordinate system.

At this stage, all the video-camera parameters are estimated and can then be applied to obtain the corresponding three-dimensional types of information as shown in the next chapter.

4.4 Summary

This chapter has given a technical overview of the video-camera calibration based on Tsai's model. By applying the Tsai's camera calibration using a set of correspondent points (three-dimensional points and corresponding two-dimensional image points), both the external and internal parameters of a video camera can be estimated.

The Tsai camera model, together with estimated parameters can be further used to obtain particular three-dimensional types of information, such as the approximated three-dimensional person region, which is presented in details in the next chapter.

Chapter 5

SINGLE GAUSSIAN MODEL BASED FALL DETECTION AND THREE-DIMENSIONAL FEATURE EXTRACTION

5.1 Introduction

In this chapter, an effective new fall detection system for intelligent indoor environments is proposed based on three-dimensional features and a single Gaussian model. Initially, the codebook (CB) background subtraction (BGS) method as described in [73] is performed on multiple calibrated video-cameras (the video-camera calibration procedure based on Tsai's model is described in the previous Chapter 4 and will be applied as an important step in this chapter). The approximated three-dimensional person is then reconstructed and corresponding three-dimensional features are extracted from the obtained BGS results. The extracted three-dimensional features are applied to construct a single Gaussian model using the maximum likelihood technique, which can be used to distinguish falls and non-falls by comparing with a single preset

threshold. The performance of the proposed fall detection system with different threshold values is validated and the results are presented in the experimental section.

5.2 Three-dimensional human body reconstruction and feature extraction

In this section, the three-dimensional human body shape is reconstructed from the multiple calibrated video-cameras and how the corresponding three-dimensional features are extracted will be explained in more details in the following subsections.

5.2.1 Three-dimensional reconstruction of a human body based multi-view

Once the video-camera parameters have been estimated (illustrated in previous Chapter 4 section 4.3) the three-dimensional human body shape can subsequently be reconstructed as presented in [99]. Firstly, the room space (assumed to be cubic) is divided into non overlapping blocks called 'voxels block', (O_w, x_w, y_w, z_w) and (O, x, y, z) represent the three-dimensional world coordinate system and three-dimensional camera coordinate system respectively as shown in Figure 5.1. For a two-dimensional image pixel with two-dimensional image pixel coordinate $P\ (X_f, Y_f)$, from equation (5.2.1) and (5.2.2),

$$\begin{aligned} X_u &= X_d(1 + kr^2) \\ Y_u &= Y_d(1 + kr^2) \end{aligned} \tag{5.2.1}$$

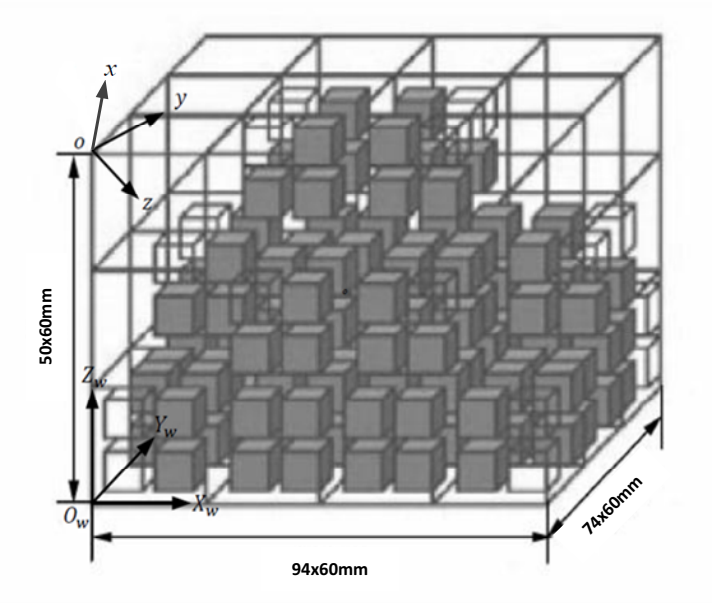


Figure 5.1. Discretisation of the three-dimensional room space with the dimension 5.64m x 2.8m x 3m.

where $r = \sqrt{X_d^2 + Y_d^2}$ and k is the coefficient of the radial lens distortion

$$X_f = s_x \dot{d}_x^{-1} X_d + C_x$$
$$Y_f = \dot{d}_y^{-1} Y_d + C_y \tag{5.2.2}$$

where C_x and C_y are the center coordinates system of the captured image, s_x is the uncertainty scaling factor due to camera scanning and acquisition time error, $\dot{d}_x$ and $\dot{d}_y$ represent the corresponding size of a pixel, and are calculated as in equation (4.2.8). The image plane coordinate system (X_u, Y_u) of the corresponding ideally projected point P_u on the image plane coordinate system can be obtained from (X_f, Y_f)

as:

$$X_u = \frac{(X_f - C_x)\dot{d_x}}{s_x}(1 + kr^2)$$
$$Y_u = (Y_f - C_y)\dot{d_y}(1 + kr^2) \tag{5.2.3}$$

The three-dimensional world coordinate system of the point P_u is (X_u, Y_u, f) where f is the focal length and represents the z-axis of every point on the image plane converted using equation 5.2.4:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \tag{5.2.4}$$

where R is a 3×3 rotation matrix and T is a 3×1 translation vector The point P_u can be denoted as (P_{xw}, P_{yw}, P_{zw}).

Using the above, the origin point $O(0, 0, 0)$, of the three-dimensional camera coordinate system which represents the center of the camera lens can also be converted to the three-dimensional world coordinate system and is (O_{xw}, O_{yw}, O_{zw}).

Points (P_{xw}, P_{yw}, P_{zw}) and (O_{xw}, O_{yw}, O_{zw}) thus determine a three-dimensional line (as the line OP_u in Figure 5.2) in the three-dimensional world coordinate system (x_w, y_w, z_w) which intersects with voxels block in three-dimensional space as:

$$\frac{x_w - O_{xw}}{P_{xw} - O_{xw}} = \frac{y_w - O_{yw}}{P_{yw} - O_{yw}} = \frac{z_w - O_{zw}}{P_{zw} - O_{zw}} = t \tag{5.2.5}$$

where t is a scalar indicating the slope of the three-dimensional line.

Using this technique, a simple and efficient method is applied for ob-

taining the voxels block with which the three-dimensional line intersects which is illustrated as follows:

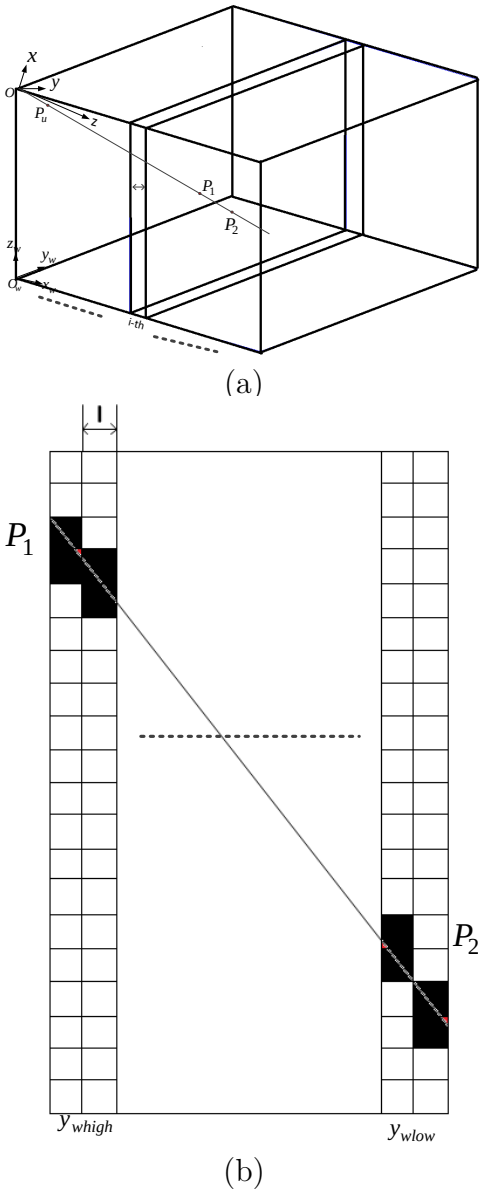


Figure 5.2. The procedure of obtaining the intersected voxel block for the i^{th} bin along the x_w axis. (a) The three-dimensional line connecting the video-camera coordinate system origin O and point P_u intersect with the i^{th} bin along the x_w axis, the intersected line segment is P_1P_2 and (b) The coordinate system range of P_1P_2 in the y_w direction is denoted as $[y_{wlow}, y_{whigh}]$, every bin in $[y_{wlow}, y_{whigh}]$ is tested to obtain the final intersected voxel person (marked in black).

1. If it is assumed that the size of a voxel block is $[l \times l \times l]$ and the number of voxels blocks along the x_w, y_w and z_w directions (which coincide with the length, width and height of the three-dimensional space as shown in Figure 5.1 in three-dimensional room space) are N_{x_w}, N_{y_w} and N_{z_w} respectively, the room space can then be divided into N_{x_w}, N_{y_w} and N_{z_w} bins along the three axes of the three-dimensional world coordinate system, with each bin's length being l.

2. For the i^{th} bin along the x_w direction, its x_w coordinate range is $[(i-1)*l, i*l]$. As such $(i-1)*l$ and $i*l$ can be substituted into equation (5.2.5) to obtain the y_w range of the line segment in that bin, denoted as $[y_{wlow}, y_{whigh}]$ with

$$\begin{aligned} y_{w_{low}} = \min\{&Oy_w + \frac{(i-1)*l - Ox_w}{Px_w - Ox_w}(Py_w - Oy_w), \\ &Oy_w + \frac{i*l - Ox_w}{Px_w - Ox_w}(Py_w - Oy_w)\} \\ y_{w_{high}} = \max\{&Oy_w + \frac{i*l - Ox_w}{Px_w - Ox_w}(Py_w - Oy_w), \\ &Oy_w + \frac{(i-1)*l - Ox_w}{Px_w - Ox_w}(Py_w - Oy_w)\} \end{aligned} \tag{5.2.6}$$

3. This can be converted to the bin index range $[I^{yw}_{low}, I^{yw}_{high}]$ along the y_w axis as:

$$\begin{aligned} I^{yw}_{low} &= int(\frac{y_{wlow}}{l}) + 1 \\ I^{yw}_{high} &= int(\frac{y_{whigh}}{l}) + 1 \end{aligned} \tag{5.2.7}$$

where $int(.)$ represents the operation of getting the integer part of the result.

4. Finally, $[I_{low}^{yw}, I_{high}^{yw}]$ intersects with $[1, N_{yw}]$ to obtain the bin index range confined by the three-dimensional room space, denoted as $[\tilde{I}_{low}^{yw}, \tilde{I}_{high}^{yw}]$, if the intersection result is non-empty.

In a similar way, for every bin index from $\tilde{I}_{low}^{yw}$ to $\tilde{I}_{high}^{yw}$, the corresponding z_w direction's bin index range confined by the three-dimensional room space (denoted as $[\tilde{I}_{low}^{zw}, \tilde{I}_{high}^{zw}]$) can be calculated. In the end, the bin indices of voxels block which are intersected with the three-dimensional line are obtained in the i_{th} bin along the x_w direction.

Figure 5.2 shows the procedure of obtaining the intersection of the voxels block in the i_{th} bin along the x_w direction with the line $\overline{P_1P_2}$, the same procedure is applied for every bin along this direction and to obtain the voxels block with which the three-dimensional line intersects in the three-dimensional room space.

For every pixel on the image, the voxel blocks intersected by the corresponding three-dimensional line are obtained which form a voxel block set corresponding to that pixel.

Multiple video-cameras of viewpoints are applied for three-dimensional human body reconstruction (as shown in Figure 5.3 for a two video-camera case).

For each video-camera the voxel person set which corresponding to the pixels on the image plane initially obtained using the the CB method based on BGS techniques as shown in Chapter 3 is applied for extraction of the moving object (the CB BGS techniques can achieve the best performance in the indoor environments as shown in the experimental

analysis in Chapter 3 Section 3.5).

The union of voxels person corresponding to pixels in the moving object for the i_{th} video-camera is then calculated and denoted as $\mathbf{V}_t^i = \{\mathbf{V}_{t,1}^i, ..., \mathbf{V}_{t,Pi}^i\}$, where t is the captured time, i is the index of the video-camera, $\mathbf{V}_{t,1}^i$ represents voxels corresponding to the background subtraction result of the $i-th$ camera and P_i is the number of cameras. The voxel person can be obtained by intersecting the union sets of multiple camera as: $\mathbf{V}_t' = \bigcap_{i=1}^{C} \mathbf{V}_t^i$ where $\mathbf{V}_t'$ denotes the voxel person set corresponding to the three-dimensional reconstructed human body shape and C is the number of video-cameras.

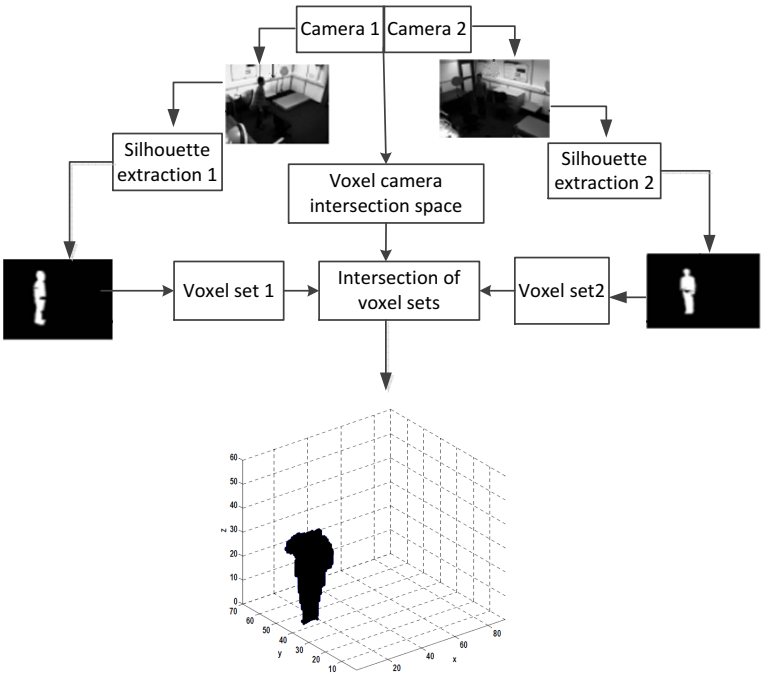


Figure 5.3. The procedure for three-dimensional voxels person reconstruction from two video-camera measurements.

5.2.2 Three-dimensional feature extraction

The next step is to evaluate both the centroid position and the orientation value (denoted as Δ_t and θ), which reflects the three-dimensional angle between the constructed human body and the ground floor plane. The centroid of the voxel person at time t, $u_t = [x_t, y_t, z_t]$ is evaluated using:

$$\mathbf{u}_t = (\frac{1}{P}) \sum_{j=1}^{P} \mathbf{V}'_{t,j} \tag{5.2.8}$$

where $\mathbf{V}'_{t,j}$ is the j^{th} voxel block $\mathbf{V}'_t$ at time t.

Within a time interval of Δ_t the centroid's height information and differences of the centroid's horizontal and vertical position can both be used as features for the fall recognition system. The horizontal variation of the centroid is calculated as: $\sqrt{(x_{t+\Delta_t} - x_t)^2 + (y_{t+\Delta_t} - y_t)^2}$ and the vertical variation is: $|z_{t+\Delta_t} - z_t|$. The covariance matrix used to define the orientation information is:

$$(\frac{1}{P}) \sum_{j=1}^{P} (\mathbf{V}'_{t,j} - \mathbf{u}_t)(\mathbf{V}'_{t,j} - \mathbf{u}_t)^T \tag{5.2.9}$$

where $(\mathbf{V}'_{t,j} - \mathbf{u}_t)$ is the difference between the three-dimensional position of the j^{th} voxel and voxel person's centroid.

If the largest eigenvector corresponding to the largest eigenvalue at time t is $eigenvec_t$, then θ_t is calculated as:

$$\theta_t = (eigenvec_t.\langle 0, 0, 1\rangle^T, -eigenvec_t.\langle 0, 0, 1\rangle^T) \tag{5.2.10}$$

where the eigenvalues and eigenvector are calculated as shown in [100]. If the elderly person is upright, the value is near unity; if the elderly

person for example lies on the ground, then the value is near zero.
This value and its difference during Δ_t $(|\theta_{t+\Delta_t} - \theta_t|)$ are chosen as the remaining elements of the feature vector.
Finally, a 5-dimensional feature vector is obtained, which consists of the following five elements:

1. The centroid's horizontal position change over Δ_t
2. The centroid's vertical position change over Δ_t
3. The centroid's vertical position at particular time t+Δ_t
4. The θ_t value change over Δ_t
5. The θ at the particular time t+Δ_t

These five-dimensional features are chosen based on previous research work done on characteristics of falls [47], [55], [59], [61], [66] and [68]. The centroid position and the three-dimensional orientation angle contain sufficient information to distinguish fall and non-fall activity. For example, when a person falls the centroid height is very low and the orientation angle between the human body and the ground plane is almost zero, which is different from other common activity, such as standing, sitting and so on. It is for this reason that features (3) and (5) which provide static information besides other dynamic phenomena that can be observed for fall activities which includes the centroid horizontal and vertical position changing dramatically over a short time interval (fall is a fast activity). The human body will change from standing to lying on the floor during the same interval. This provides the dynamic information to distinguish fall and non-fall activities. It is for this reasons that features (1), (2) and (4) were chosen. Note that a time interval of

one second was chosen because fall activity usually occurs within one second.

Therefore, in this fall detection system, Δ_t is chosen to be 1s. The three-dimensional video-camera features are obtained and the ones which correspond to fall activities are used to construct models representing falling, which are then used to distinguish fall and non-fall activities. The construction of three different kinds of density models is shown in the next section.

A single Gaussian model based on maximum likelihood parameter estimation is used to distinguish unusual activity (falls) and is presented in the next section.

5.3 Single Gaussian model based fall detection

The single Gaussian model, also known as the normal distribution, is a widely used model for the probability distribution of continuous variables.

In the case of a single variable x, the Gaussian distribution can be written in the form (5.3.1):

$$P(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2\sigma^2}(x-\mu)^2 \right\} \tag{5.3.1}$$

where μ is the mean and σ^2 is the variance. An example of a single Gaussian model with $\mu = 0$ and $\sigma^2 = 1$ is presented in Figure 5.4. For a D-dimensional vector x, the multivariate Gaussian distribution takes

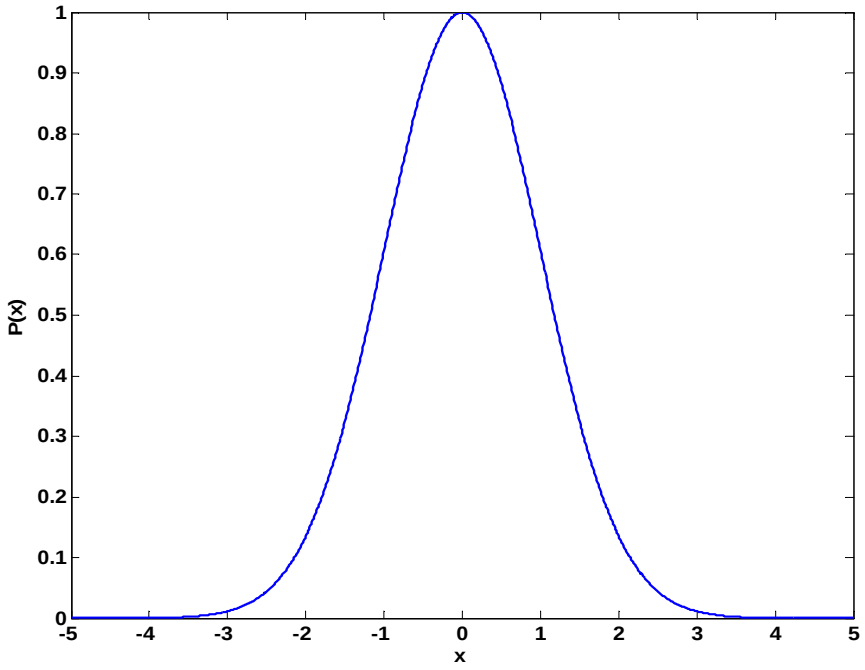


Figure 5.4. The single variable single Gaussian model.

the form (5.3.2):

$$P(\mathbf{x} \mid \mathbf{u}, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{\mid \Sigma \mid^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\mathbf{u})^T \Sigma^{-1} (\mathbf{x}-\mathbf{u}) \right\} \quad (5.3.2)$$

where D is the dimension of variable $\mathbf{x}$, $\mathbf{u}$ and Σ represent the mean vector and covariance matrix respectively. Figure 5.5 shows an example of two-dimensional single Gaussian distribution with $\mathbf{u} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. If N dataset samples $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]$ are provided, these samples can be used to fit a single Gaussian model. The corresponding model parameters $\mathbf{u}$ and Σ can be obtained from the maximum

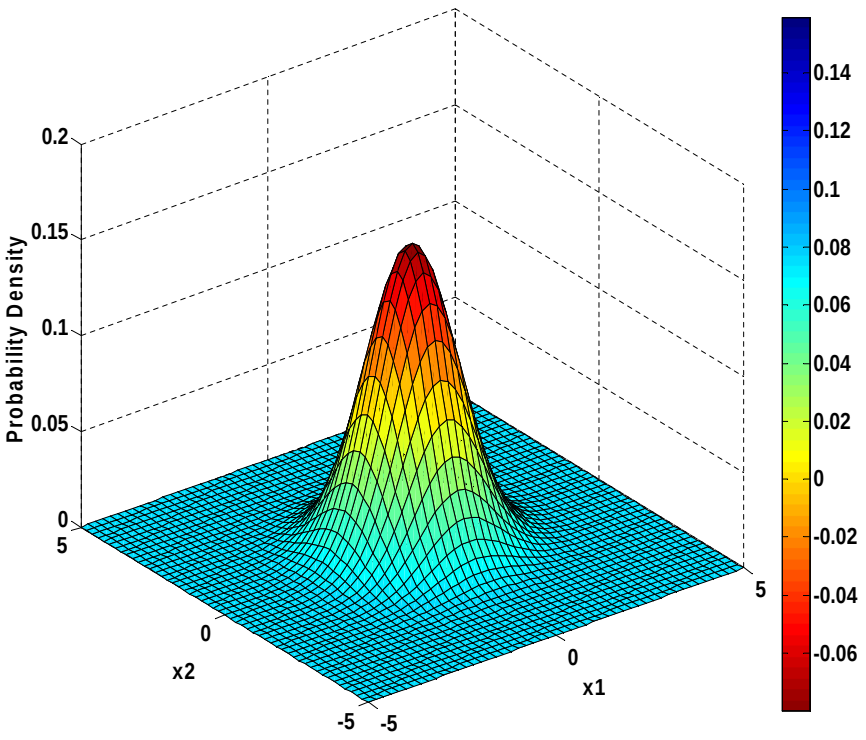


Figure 5.5. The multivariate single Gaussian model.

likelihood (ML) function as:

$$\begin{aligned} d(\ln((P(\mathbf{X}))) &= \sum_{i=1}^{N} \ln(d(P(\mathbf{x}_i))) \\ &= \sum_{i=1}^{N} \ln(\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(\mathbf{x}_i - \mathbf{u})^T \Sigma^{-1}(\mathbf{x}_i - \mathbf{u}))) \\ &= -\frac{Nd}{2}\ln(2\pi) - \frac{N}{2}\ln|\Sigma| - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{u})^T \Sigma^{-1}(\mathbf{x}_i - \mathbf{u}) \end{aligned} \tag{5.3.3}$$

As mentioned in [101], to determine the ML estimators of $\mathbf{u}$ and Σ, the first order differential of equation (5.3.3) is calculated as:

$$\begin{aligned}
\mathbf{d}\ln((P(\mathbf{X})) &= -\frac{N}{2}\mathbf{d}\ln|\Sigma| - \frac{1}{2}\mathbf{d}Tr(\Sigma^{-1}\mathbf{Z}) \\
&= -\frac{N}{2}\mathbf{d}\ln|\Sigma| - \frac{1}{2}Tr(\mathbf{d}(\Sigma^{-1})\mathbf{Z}) - \frac{1}{2}Tr(\Sigma^{-1}\mathbf{d}(\mathbf{Z})) \\
&= -\frac{N}{2}Tr(\Sigma^{-1}\mathbf{d}\Sigma) + \frac{1}{2}Tr(\Sigma^{-1}\mathbf{d}(\Sigma)\Sigma^{-1}\mathbf{Z}) \\
&+ \frac{1}{2}Tr(\Sigma^{-1}(\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{u})(\mathbf{du})^T + (\mathbf{du})\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{u})^T)) \\
&= \frac{1}{2}Tr(\Sigma^{-1}(\mathbf{d}\Sigma)\Sigma^{-1}(\mathbf{Z} - N\Sigma)) + (\mathbf{du})^T\Sigma^{-1}\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{u}) \\
&= \frac{1}{2}Tr(\Sigma^{-1}(\mathbf{d}\Sigma)\Sigma^{-1}(\mathbf{Z} - N\Sigma)) + N(\mathbf{du})^T\Sigma^{-1}(\bar{\mathbf{x}} - \mathbf{u})
\end{aligned} \tag{5.3.4}$$

where $\mathbf{d}(\cdot)$ is the differential operation, $Tr(\cdot)$ represents the trace of a matrix and $\bar{\mathbf{x}}$, $\mathbf{Z}$ can be represented as:

$$\begin{aligned}
\bar{\mathbf{x}} &= \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \\
\mathbf{Z} &= \sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{u})(\mathbf{x}_i - \mathbf{u})^T
\end{aligned} \tag{5.3.5}$$

The ML estimators of $\mathbf{u}$ and Σ, denoted as $\hat{\mathbf{u}}$ and $\hat{\Sigma}$, are then obtained by making the first differential of equation (5.3.4) equal to zero to obtain:

$$\begin{aligned}
\bar{\mathbf{x}} - \hat{\mathbf{u}} &= 0 \\
\mathbf{Z} - N\hat{\Sigma} &= 0
\end{aligned} \tag{5.3.6}$$

from which the ML estimators of $\hat{\mathbf{u}}$ and $\hat{\Sigma}$ are evaluated as:

$$\hat{\mathbf{u}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$$
$$\hat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{u})(\mathbf{x}_i - \mathbf{u})^T \tag{5.3.7}$$

To prove that the obtained $\hat{\Sigma}$ and $\hat{\mathbf{u}}$ are the local maximum estimator (instead of the local minimum estimator which can also make the first differential of equation is equal to zero), the second differential of equation (5.3.8) is calculated, which is:

$$\begin{aligned}\mathbf{d}^2\ln((P(\mathbf{X})) &= \frac{1}{2}Tr(\mathbf{d}(\Sigma^{-1})(\mathbf{d}\Sigma)\Sigma^{-1}(\mathbf{Z}-N\Sigma))\\ &+ \frac{1}{2}Tr(\mathbf{d}(\Sigma^{-1})(\mathbf{d}\Sigma)\mathbf{d}(\Sigma^{-1})(\mathbf{Z}-N\Sigma))\\ &+ \frac{1}{2}Tr(\Sigma^{-1}(\mathbf{d}\Sigma)\Sigma^{-1}(\mathbf{dZ}-N\mathbf{d}\Sigma))\\ &+N\ (\mathbf{du})^T(\mathbf{d}\Sigma^{-1})(\bar{\mathbf{x}}-\mathbf{u})\\ &- N(\mathbf{du})^T\Sigma^{-1}\mathbf{du}\end{aligned} \tag{5.3.8}$$

For the obtained $\hat{\Sigma}$ and $\hat{\mathbf{u}}$ in equation (5.3.7), the second differential in equation (5.3.8) becomes:

$$\mathbf{d}^2\ln((P(\mathbf{X})) = -\frac{N}{2}Tr(\hat{\Sigma}^{-1}(\mathbf{d}\Sigma)\hat{\Sigma}^{-1}(\mathbf{d}\Sigma)) - N(\mathbf{du})^T\hat{\Sigma}^{-1}(\mathbf{du}) \tag{5.3.9}$$

which is always smaller than zero, so that the obtained $\hat{\Sigma}$ and $\hat{\mathbf{u}}$ are the local maximum estimator of the ML function in equation (5.3.4).

For this particular fall detection problem, multiple fall activities are initially simulated and three-dimensional video features are then extracted from the video sequences of fall activities to form the dataset $\mathbf{X}$. A single Gaussian model is then constructed from the dataset $\mathbf{X}$

and the constructed model is then used to distinguish fall and non-fall activities. For a three-dimensional video feature vector (denoted as $\mathbf{x}$) of a certain type of activity, if $P(\mathbf{x})$ (the probability value for the single Gaussian model) is high, then $\mathbf{x}$ is from the supporting region corresponding to $\mathbf{X}$ and the corresponding activity is regarded as a fall; otherwise, the activity is regarded as a non-fall activity. A threshold is set and the following formula shows the criteria of detecting falls for an incoming feature vector $\mathbf{x}$:

$$Activity_{detection} = \begin{cases} Fall & P(\mathbf{x}) \geq threshold \\ NonFall & P(\mathbf{x}) < threshold \end{cases} \tag{5.3.10}$$

where a threshold value is set and if the corresponding probability density function of $P(\mathbf{x})$ is no less than the threshold for an extracted feature vector $\mathbf{x}$, then the corresponding activity from which the feature vector is extracted belongs to fall activity; otherwise, the activity belongs to non-fall activity.

As proposed in [102], direct calculation of the density estimate is avoided to solve the problem of numerical instability (the determinant of the covariance matrix is zero), only the Mahalanobis distance is used. The Mahalanobis distance is defined as:

$$f(\mathbf{x}) = (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \tag{5.3.11}$$

The obtained Mahalanobis distance $f(\mathbf{x})$ for an incoming feature $\mathbf{x}$ is then compared with a proper threshold for detecting falls, the equation

(5.3.10) and can be rewritten as:

$$Activity_{detection} = \begin{cases} Fall & f(\mathbf{x}) < threshold \\ NonFall & f(\mathbf{x}) \geq threshold \end{cases} \tag{5.3.12}$$

The flowchart of the proposed single Gaussian model fall detection system is presented in Figure 5.6. Online video frames are captured from

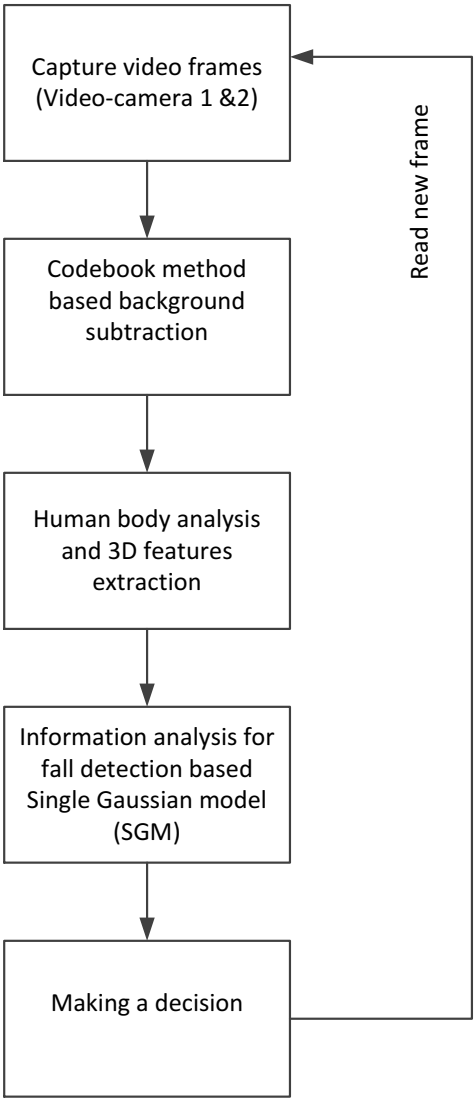


Figure 5.6. The flow-diagram of the proposed fall detection system based single Gaussian model.

two calibrated video-cameras and three-dimensional human body shape is reconstructed. An efficient CB BGS algorithm is firstly applied to extract the human body foreground and some post-processing is applied to improve the results.

Different features are extracted from the three-dimensional human body reconstruction, which are used for classification purposes. These features are fed into the single Gaussian model and by setting proper threshold value are used to determine fall or non-fall when it happens. In this research work, different threshold values for the Mahalanobis distance are chosen and tested for the single Gaussian model to distinguishing fall and non-fall activities when they happen; the corresponding results are illustrated in the next experimental analysis section.

5.4 Experimental analysis

5.4.1 Experimental description

The experiments are performed in the Intelligent Audio/Video Experimental Laboratory within the Advanced Signal Processing Group (ASPG) at Loughborough University. Two Basler A312fc video-cameras and the Streampix 5 software were applied to capture the video sequences. The recorded sequences were then processed using VC++ 6.0 (with OpenCV 1.0) and MATLAB R2010. Figure 5.7 shows the room scene captured by two video-cameras.

5.4.2 Dataset collection and description

One person was invited to participate in the experiment, for which he simulate 40 fall activities (including 10 frontal falls, 10 backward falls

(a) (b)

Figure 5.7. The room's scenes captured by two video-cameras. (a) The room scene captured by video-camera 1, and (b) The room scene captured by video-camera 2.

and 20 side falls), which are used to compose a training dataset for model construction. He also simulate another 40 fall activities (which also include 10 frontal falls, 10 backward falls and 20 side falls) and 40 non-fall activities (including 8 walking actions, 8 rapid moving actions, 8 bending actions, 8 sitting actions and 8 lying actions), which are used for testing purpose, as outlines in Table 5.1. Each activity recorded

Table 5.1. The characteristic of dataset used

Activity	Simulated activity
80 fall	20 front, 20 backward, 40 side
40 non-fall	8 walking, 8 rapid moving, 8 bending, 8 sitting, 8 lying

is listed on Table 5.1 lasts about 1-2s (with frame rate 15fps). The final output video format is audio video interleaved (AVI) and three-dimensional features were extracted from the recorded video clips of these activities for training and testing.

5.4.3 Video-camera calibration and three-dimensional person construction

For the calibration of each video-camera, a large chessboard is printed and adhered to a plate, which is placed at a particular position in the room and captured by the video-camera needed to be calibrated. The corner points of the chessboard blocks are used for video-camera calibration (these length and width of the chessboard plate are parallel to the world coordinate axes x_w and y_w). The two-dimensional image pixel coordinates are obtained manually from the image. The chessboard images captured by two video-cameras are presented in Figure 5.8. The block corner points are used for video-camera calibration and are marked as red stars. Table 5.2 and 5.3 describe the

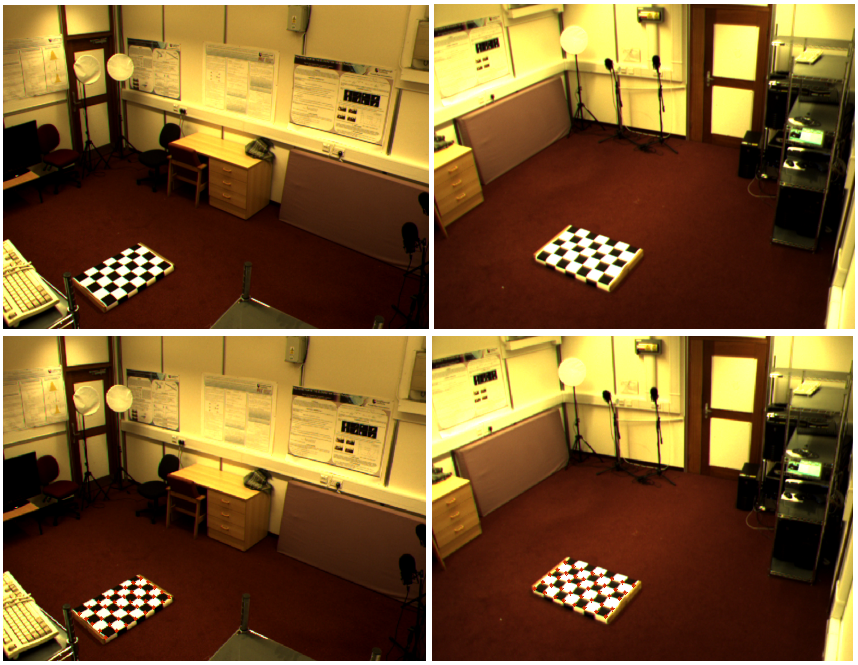


Figure 5.8. The chessboard plate used for video-camera calibration with the block corner points marked as red stars.

video-cameras parameters estimated from the corresponding points set.

To construct the three-dimensional voxel person, the room is divided

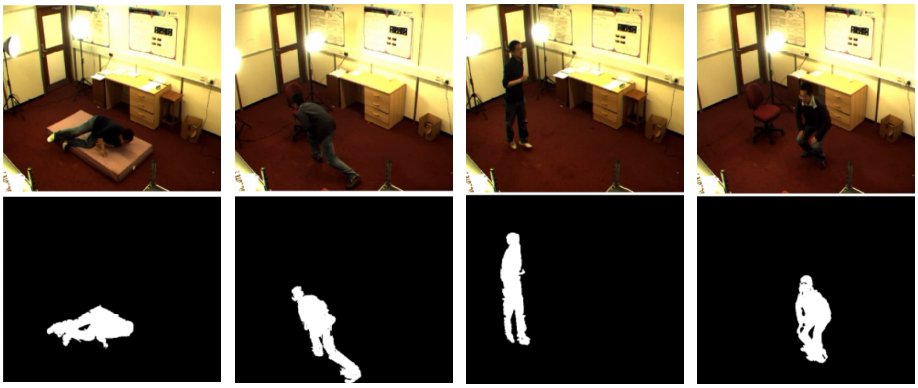
Table 5.2. Calibrated video-camera parameters for video-camera no.1.

Parameters needed to be estimated	Video-camera Calibration results
$\begin{bmatrix} r1 & r2 & r3 \\ r4 & r5 & r6 \\ r7 & r8 & r9 \end{bmatrix}$	$\begin{bmatrix} -0.7001 & -0.7105 & -0.0717 \\ -0.1772 & 0.2563 & -0.9502 \\ 0.6935 & -0.6525 & -0.3053 \end{bmatrix}$
$[T_x, T_y, T_z]$ (mm)	$[760.4500, 1122.9000, 5620.2000]$
$f(mm)$	7.5100
k	0.0033
s_x	0.9719

Table 5.3. Calibrated video-camera parameters for video-camera no.2.

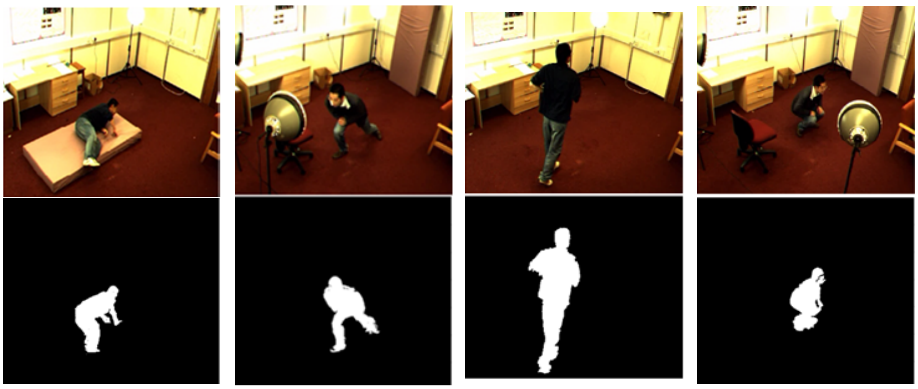
Parameters needed to be estimated	Calibration results
$\begin{bmatrix} r1 & r2 & r3 \\ r4 & r5 & r6 \\ r7 & r8 & r9 \end{bmatrix}$	$\begin{bmatrix} -0.4405 & 0.8959 & 0.0569 \\ 0.3853 & 0.1964 & -0.9016 \\ -0.8190 & -0.3753 & -0.4318 \end{bmatrix}$
$[T_x, T_y, T_z]$ (mm)	$[-1867.9000, 588.9082, 4325.1000]$
f (mm)	6.2373
k	0.0028
s_x	0.9462

into $94 \times 74 \times 50$ blocks along the x_w, y_w and z_w axes of the three-dimensional world coordinate system with each block having the size of $60mm \times 60mm \times 60mm$. Figures 5.9, 5.10 and 5.11 show the BGS techniques and three-dimensional person construction results for four different activities. Initially, the CB method based on BGS techniques is applied on both video-cameras to obtain the moving object, and the three-dimensional person is then constructed from the BGS results and the estimated video-camera parameters. After three-dimensional person construction, the three-dimensional features are then extracted and the features corresponding to fall activities are applied to build the models for fall detection.



(a) (b) (c) (d)

Figure 5.9. Four activities and corresponding background subtraction results for video-camera 1. (a) Lying, (b) Bending, (c) Walking, and (d) Crouching.



(a) (b) (c) (d)

Figure 5.10. Four activities and corresponding background subtraction results for video-camera 2. (a) Lying, (b) Bending, (c) Walking, and (d) Crouching.

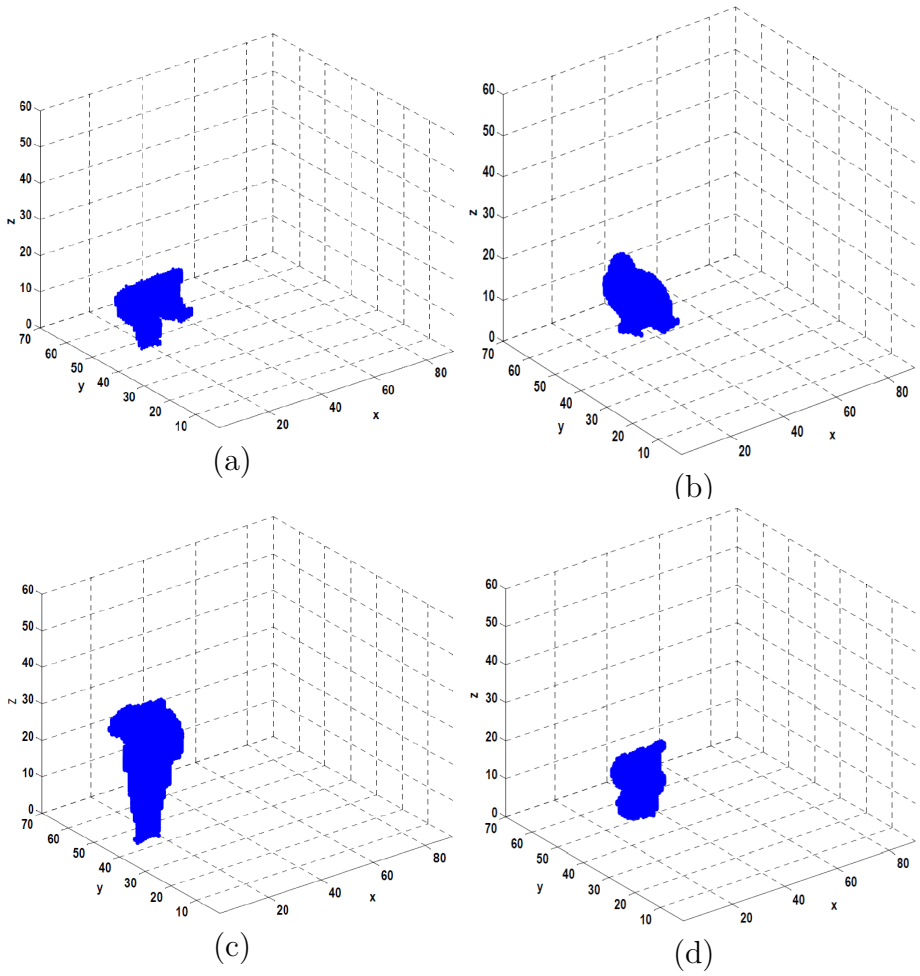


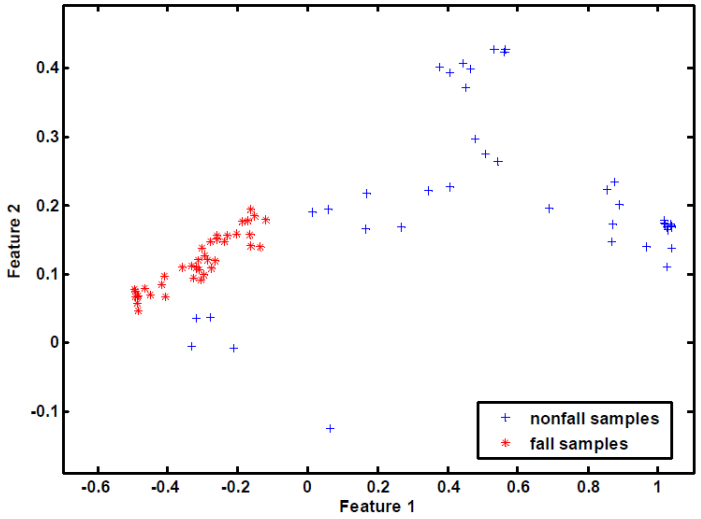
Figure 5.11. Three-dimensional person construction by using background subtraction results from video-cameras 1. and 2. (a) Lying, (b) Stretching, (c) Walking, and (d) Crouching.

5.4.4 Single Gaussian model analysis

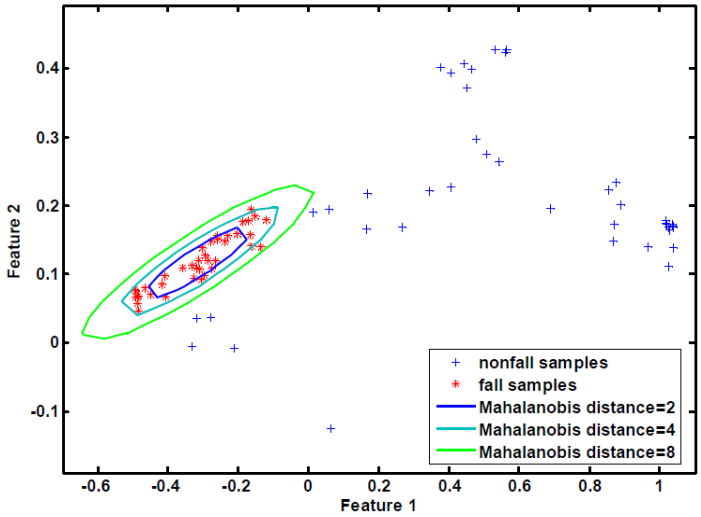
In this section, the performance of the single Gaussian model is analysed for the recorded dataset. This includes 40 fall activities (including 10 frontal falls, 10 backward falls and 20 side falls), which are used to compose a training dataset for model construction. Also another 40 fall activities (which also include 10 frontal falls, 10 backward falls and 20 side falls) and 40 non-fall activities (including 8 walking actions, 8 rapid moving actions, 8 bending actions, 8 sitting actions and 8 lying

actions), were recorded and used for testing purpose, as outlined in Table 5.1.

Figure 5.12 (a) shows the two-dimensional principle component analysis (PCA) [60] projections of the features extracted from 80 fall activities and 40 non-fall activities, and Figure 5.12 (b) shows that a single Gaussian model can be applied to fit the projected two-dimensional features corresponding to falling activities; different contours of the single Gaussian model corresponding to different Mahalanobis distance values are plotted and the projected fall features and non-fall features can be successfully distinguished by a proper contour (with the Mahalanobis distance being 8).



(a)



(b)

Figure 5.12. (a) Projected two-dimensional feature by principle component analysis, and (b) The projected two-dimensional features by PCA and the fitted single Gaussian model with different Mahalanobis distances.

In order to analyse the performance of the single Gaussian model on the training and testing datasets with full-dimensional features, Receiver operating characteristic (ROC) analysis [102] is applied for the single Gaussian model. Different thresholds are chosen and the true positive rate (TPR), which represents the percentage of falls which are correctly detected and false negative rate (FNR), which represents the

percentage of non-falls which are wrongly detected as falls are calculated for these thresholds, and the results are plotted as a ROC curve in Figure 5.13. To reveal more details, the performance with different thresholds corresponding to the Mahalanobis distances are listed in Table 5.4. Ideally, for a perfect fall detection system, TPR should be 1 and FNR should be 0.

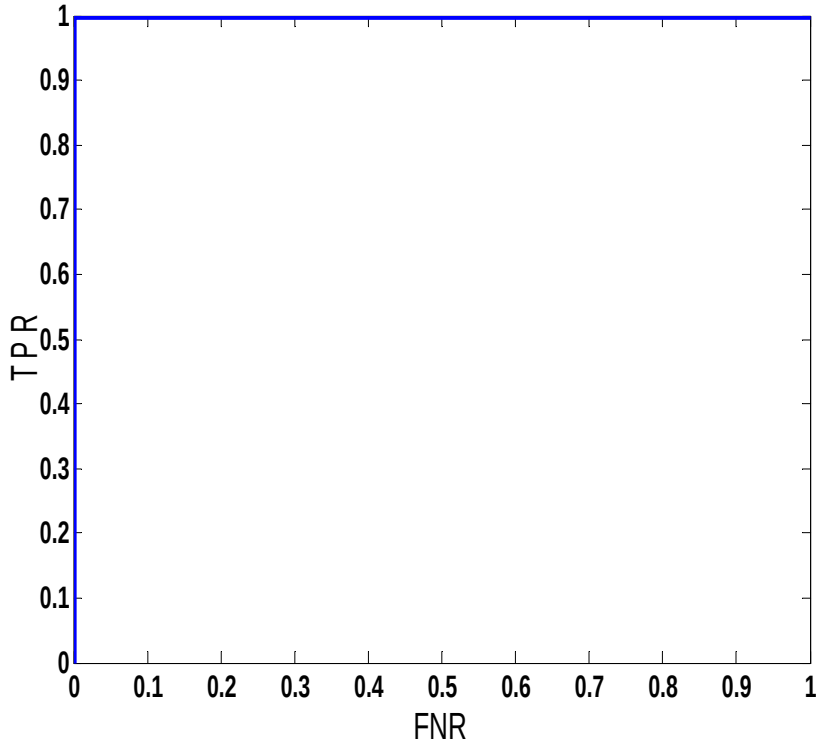


Figure 5.13. The ROC curve of the single Gaussian model.

Table 5.4. The performance of the single Gaussian model with different thresholds value (The experiment is performed by applying the 5-dimensional feature (not the 2-dimensional features obtained from PCA)).

Thresholds	0	10	20	30	40	50	60	70	80	90	100
TPR	1	1	1	1	1	1	1	0.975	0.95	0.925	0.85
FNR	1	0.25	0.05	0	0	0	0	0	0	0	0

From Figure 5.13 and Table 5.4, it can be observed that the single Gaussian model can effectively distinguish different fall activities and non-fall activities at a proper threshold setting (between 30 and 60 for the Mahalanobis distance as shown in Table 5.4 and Figure 5.12 (b)). This is due to the fact that the obtained five-dimensional features based on three-dimensional person construction are distinguishable for fall and non-fall activities, as shown in Figure 5.12 (a) for the projected two-dimensional images.

5.5 Summary

This chapter proposed a new fall detection scheme based on three-dimensional human body features extraction and a single Gaussian model to distinguish unusual behaviour (falls) for enclosed environments from multiple calibrated video-camera.

A three-dimensional person was initially constructed from the obtained codebook background subtraction results from multiple calibrated video-cameras.

The position, velocity and orientation information corresponding to fall activities were then extracted from the three-dimensional human body to build the Gaussian model for distinguishing fall activities and non-fall activities. The performance of the single Gaussian model with different thresholds was validated and results show that the single Gaussian model can effectively distinguish falls and non-falls with the proper threshold setting.

The shortcoming of this process lies in the fact that in order to obtain the three-dimensional person construction results and extract the corresponding features, at least two video-cameras are used thus increasing

the financial costs; furthermore, the video-cameras need to be individually calibrated and the room dimension also needs to be measured beforehand. This will also cause inconvenience in the real application. The level of Computational complexity in obtaining the pixels voxel correspondence limits its suitability for real time application.

Therefore, to obtain a simple but effective fall detection system, a two-dimensional postures recognition based fall detection scheme is proposed. This scheme is implemented using only one un-calibrated video-camera, which can effectively distinguish fall activities and non-fall activities performed in different directions.

Chapter 6

SUPERVISED MULTI-CLASS CLASSIFIER FOR FALL DETECTION BASED ON POSTURE FEATURES

6.1 Introduction

This chapter describes the development of a new effective fall detection system based on posture features. Based on the codebook (CB) method based background subtraction (BGS) technique for segmenting the human body posture, in addition to new advanced post-processing techniques, an improved background subtraction technique is applied for a real home indoor environment.

Special types of features which can be used to describe the segmented human body posture are then extracted. A supervised fall detection system is proposed, the posture features are obtained and can be fed into an efficient supervised classifier. This system uses a multi-class support vector machine (MCSVM) classifier, for which the features from various types of postures simulated by different persons are used to

build the corresponding classifier for posture classification. The results of the classifier, together with certain rules derived from characteristics of fall activities, can then be used for detecting whether a fall happens or not.

Results based on experimental studies are used to assess the fall detection performance in a real home environment.

6.2 Codebook method reconsidering based BGS techniques

In Chapter 3 Section 3.2, different BGS techniques have been discussed and the results show that an efficient codebook based BGS algorithm can be employed to obtain the best performance of extracting the foreground object in indoor environments.

Post-processing techniques which include (a blob operation and a morphological technique) were applied to obtain an improved background subtraction result. However, in a real home environment, three problems arise:

1. Sometimes the furniture (chair or table) in the home environment will be moved.

2. The person may be static in the room for a long period of time (for example sitting on a chair or lying down on a couch) and the static human body region will be absorbed into the background.

3. The light condition may change dramatically when a light is turned on/off, or curtain is opened.

These three problems will definitely generate large background subtraction errors, which can not be solved by the basic post-processing

technique as mentioned in Chapter 3 Section 3.4.
In order to obtain good background subtraction results when these problems occur, some other post-processing techniques must be applied.

6.2.1 Advanced post-processing technique based codebook method

In this research, a three step blob operation strategy is adopted after the codebook method based BGS technique procedure in order to remove the errors that are introduced as a result of the movement of furniture and the long static period for an elderly person. This three stage blob operation strategy is:

Step 1. Blob merging

In this stage, a blob merging operation is applied on the original background subtraction result. If the distance between two blobs is less than a pre-set threshold, these two blobs will be merged together (as shown in Figure 6.1 (d), where blobs $B2$ and $B3$ contain several separate blobs which are near to each other).
The distance between two blobs is defined as the minimum 4-distance [75] between two rectangles which enclose the blobs given by:

$$Distance(B1, B2) = min_{p1 \in R1, p2 \in R2} d_4(p1, p2) \tag{6.2.1}$$

where $B1$ and $B2$ are two blobs, $R1$ and $R2$ are two rectangles which enclose them, and $p1$ and $p2$ are points belonging to $R1$ and $R2$. Figure 6.2 shows examples of the distance between two blobs with respect to their positions.

Step 2. Human body blob determination:

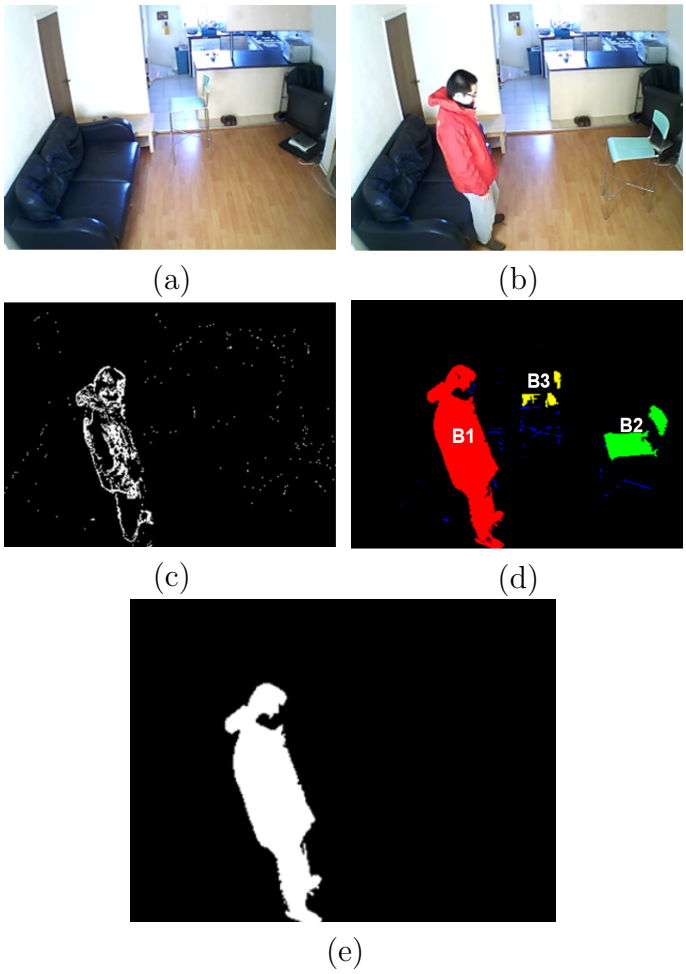


Figure 6.1. The background subtraction and the human body blob determination. (a) Background image. (b) Image with object. (c) Frame difference result obtained from two consecutive frames. (d) Original background subtraction result, there are three large blobs ($B1$, $B2$ and $B3$) after the blob merging operation and they are marked red, green and yellow, and the blue colour represents the small noise like blobs. (e) The final result obtained human body blob.

After the blobs merging step, small blobs are removed by the post-processing technique as previously mentioned in Chapter 3 Section 3.4. According to the number of remaining blobs and assuming that the elderly person lives alone, implying that there should be only one human moving object, three possible cases are given as follows:

Case 1: The number of remaining blobs is zero, which means that no

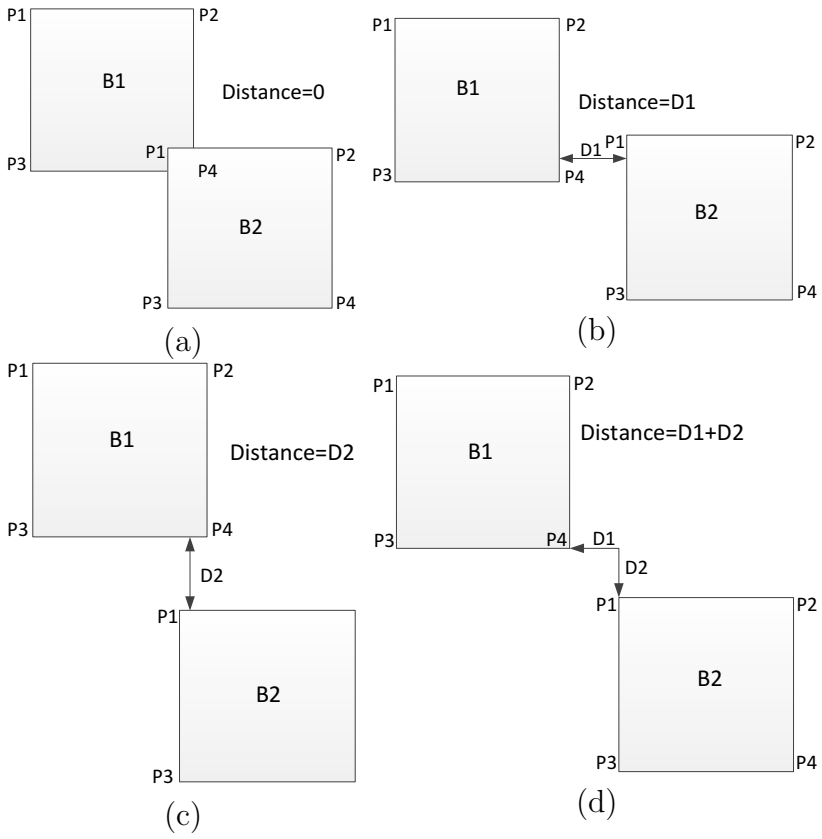


Figure 6.2. Four cases of the distance between two blobs with respect to their relative positions.

large foreground object (human body) is in the scene and the room is empty.

Case 2: The number of remaining blobs is one, which indicates that the blob represents the human body region.

Case 3: The number of blobs after blob merging is greater than one, which suggesting that there are some other regions (such as the chair regions at the new and previous positions as shown in Figure 6.1 (d)) which are mistaken as a foreground object. In this case, the human body blob is determined by using the frame difference technique as:

$$D_t(x, y) = |I_t(x, y) - I_{t-1}(x, y)| \tag{6.2.2}$$

where $I_t(x, y)$ and $I_{t-1}(x, y)$ are the gray level pixel values of consecutive frames I_t and I_{t-1} at position (x, y), $|\cdot|$ denotes the operation to obtain the absolute value and $D_t(x, y)$ is the frame differencing result for the position (x, y) at time t.

Frame differencing is applied to obtain the moving pixels Figure 6.1 (c) and the blob with the greatest number of moving pixels is taken as the human body blob. From Figure 6.1 (e), it can be seen that the blob $B1$ contains the most moving pixels and so $B1$ is finally taken as the human body blobs.

Step 3. Selective updating:

As shown in Figure 6.1 (e), the non-human body blobs $B2$ and $B3$ are removed from the final background subtraction result, and their pixel values form new codewords to be added to the background codebook immediately for background model updating. In this case, no updating is performed for pixels in the human body blob.

Furthermore, the errors generated by the movement of furniture are absorbed into the background model immediately and a better background subtraction result is obtained. Besides, the foreground human body object is not absorbed into the background even though he/she has been static for a long time.

6.2.2 Background model retraining for the sudden illumination change

The trained background CB model can be affected in various ways, including dramatic change in illumination due to for example a sudden turning on/off the light. When this occurs, the CB model needs to be re-trained because the previous CB is no longer available. The dramatic change in illumination can be detected by frame difference results, if the value of the active pixels in an image is larger than a threshold (50% is set), then the dramatic global illumination change is said to have occurred and the background model is retrained.

By using selective updating and background model retraining, the practical problems (movement of the furniture, the elderly person being stationary for a long period and sudden light change) existing in the real home environment can be solved to obtain a better human body region extraction result, which is used for the next step in the posture feature extraction.

6.3 Features used for posture description

The extracted human body postures can be described in detail by certain types of features. These are then fed into some supervised classifier for classification. In this research, two types of features are applied, they are ellipse features and projection histogram features.

The first set of features extracted from the human body silhouette is obtained from ellipse fitting proposed in [40]. This is, a moment based approach which is applied to fit the ellipse features. For a binary image

$f(x, y)$, the moments are given as:

$$m_{pq} = \sum_{x,y} x^p y^q f(x, y) \qquad \text{with } p, q = 0, 1, 2, 3... \tag{6.3.1}$$

By using the first and zero order spatial moments, the center of the ellipse can be obtained $(\bar{x}, \bar{y})$ as: $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$. The angle between the major axis of the person and the horizontal axis x gives the orientation of the ellipse, and it is computed as:

$$\theta = \frac{1}{2} arctan(\frac{2u_{11}}{u_{20} - u_{02}}) \tag{6.3.2}$$

where the central moment can be calculated as:

$$u_{pq} = \sum_{x,y} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \qquad \text{with } p, q = 0, 1, 2, 3... \tag{6.3.3}$$

The remaining parameters to describe an ellipse are the major and minor semi-axes a and b respectively, these two parameters are calculated by evaluating the greatest and least moments of inertia, which are denoted here as I_{max} and I_{min}. They can be obtained by evaluating the eigenvalues of the covariance matrix:

$$J = \begin{pmatrix} u_{20} & u_{11} \\ u_{11} & u_{02} \end{pmatrix} \tag{6.3.4}$$

I_{min} and I_{max} are the smallest and largest eigenvalues of matrix J respectively, given as:

$$I_{min} = \frac{u_{20} + u_{02} - \sqrt{(u_{20} - u_{02})^2 + 4u_{11}^2}}{2} \tag{6.3.5}$$

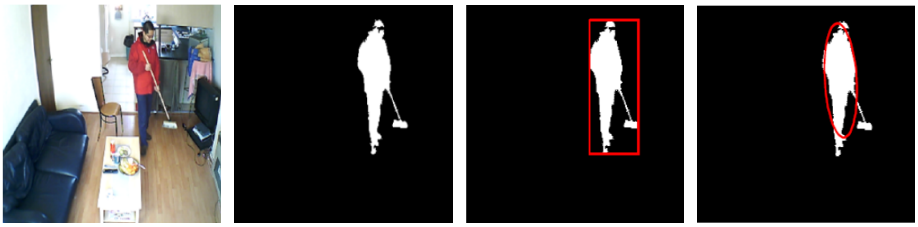
$$I_{max} = \frac{u_{20} + u_{02} + \sqrt{(u_{20} - u_{02})^2 + 4u_{11}^2}}{2} \tag{6.3.6}$$

After obtaining I_{min} and I_{max}, the major semi-axis a and minor semi-axis b can be calculated using the following equations:

$$a = (4/\pi)^{1/4} \left[\frac{I_{(max)^3}}{I_{min}} \right]^{1/8} \tag{6.3.7}$$

$$b = (4/\pi)^{1/4} \left[\frac{I_{(min)^3}}{I_{max}} \right]^{1/8} \tag{6.3.8}$$

The results from the ellipse fitting experiment are shown in Figure 6.3. For comparison, the simple blob-based rectangle fitting result used



(a) (b) (c) (d)

Figure 6.3. The rectangle fitting and ellipse fitting results. (a) Original image for a person with a broom. (b) Background subtraction based codebook result. (c) Rectangle fitting. (d) Ellipse fitting result.

in [61] is also presented, which shows that the ellipse fitting is better fitted to the human body region for a person with a broom. After ellipse fitting, the orientation of the ellipse (denoted as θ) and the ratio between a and b (denoted as ρ) are taken as features to describe a human body posture's general property. Features obtained from the ellipse fitting can describe postures in a general way, however, it is evident that 2-dimensional features alone can not fully describe postures in detail for distinguishing different postures. In order for a more detailed posture description, other features such as the projection histogram features

discussed in the next subsection, are needed.

6.3.1 Projection histogram features

A commonly used feature which describes the posture's detail information is the projection histogram [61], [62] and [49]. This feature is computationally efficient which can be applied in a real time application [103], while achieving a good performance for posture classification purpose of the supervised classifier. In this research, the projection directions of the corresponding projection histogram are along the major and minor axes of the fitted ellipse. One example is shown in Figure 6.4 where the projection histograms of the long and short axes of the fitted ellipses are obtained for different types of postures. The results show that there are differences in the patterns within the histograms between different postures, which are helpful for posture classification. The numbers of bins of the major axis projection and minor axis projection histograms are all set to 30 for this research. This value was found empirically and provides suitable detail whilst not introducing undue complexity.

For particular bins of the projection histograms along the ellipse's major and minor axes, their values are calculated as in (6.3.9) and (6.3.10):

$$bin_{major}(i) = \frac{NoFP_{major}(i)}{L_{major}} \tag{6.3.9}$$

$$bin_{minor}(i) = \frac{NoFP_{minor}(i)}{L_{minor}} \tag{6.3.10}$$

where i is the index of bins, $NoFP_{major}(i)$ and $NoFP_{minor}(i)$ denote the number of foreground pixels along the $i-th$ projection line in the directions of the major and minor axes respectively. The results are nor-

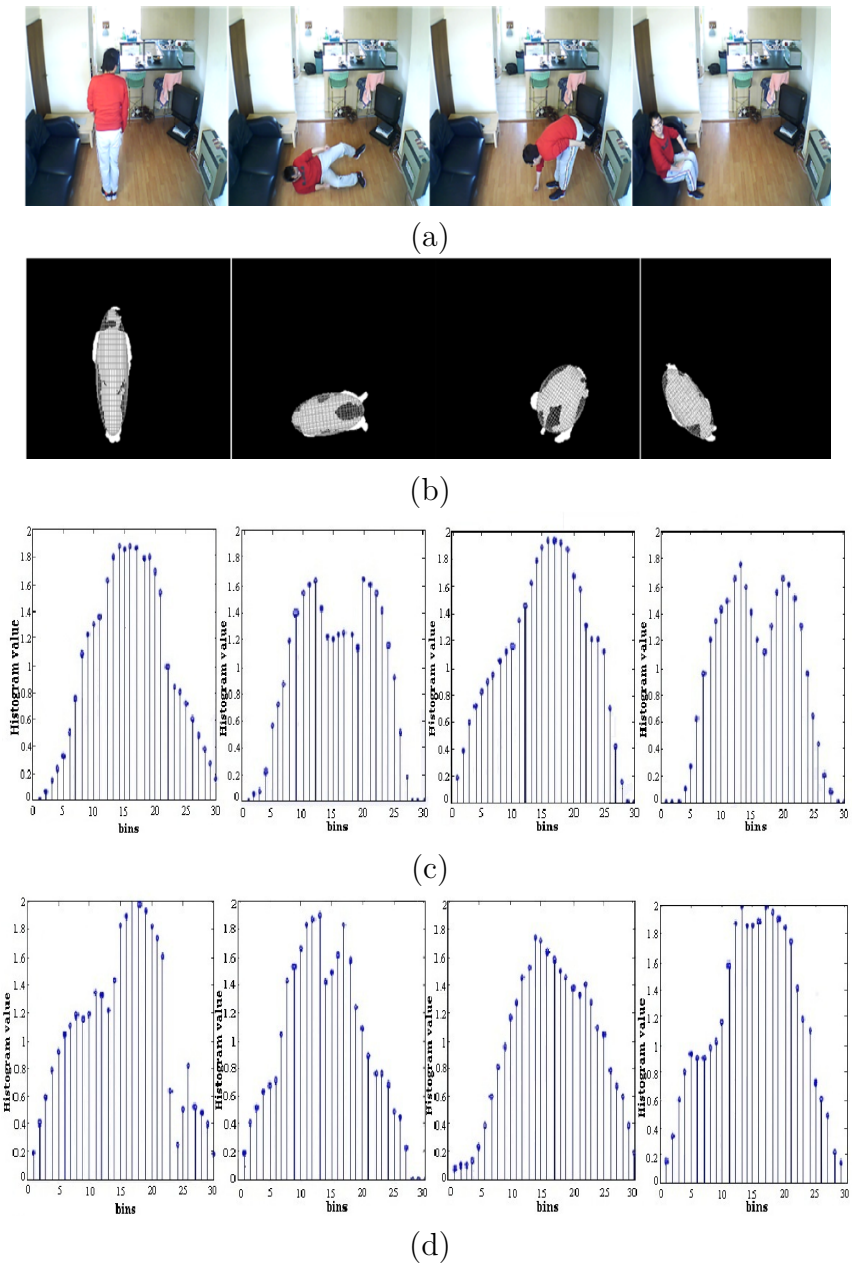


Figure 6.4. Projection histograms of four different types of postures. (a) Original frames. (b) Background subtraction results with fitted ellipses and projection lines. (c) Projection histograms along the major axis of the ellipse. (d) Projection histograms along the minor axis of the ellipse. The horizontal axis of the projection histogram represents the index of bins and the vertical axis represents the value of the projection histogram.

malised by L_{major} and L_{minor}, which represent the length of the major and minor axes. The purpose of the normalisation is to make sure this feature is invariant to the size of the foreground human body region (which will vary according to a person's distance from the camera).
The projection histogram features, together with two features obtained from the fitted ellipse result: the ratio between the major and minor axes *rho* and the orientation angle *theta*, can be applied for posture classification by a support vector machine, which can deal with both multi-class classification and high-dimensionality features. This is will lead to overcome the problem with single Gussain model that can not handle with the multiple classes classification problem and high-dimensionality features. The results are presented in the experiment section.

6.4 Support vector machine based supervised learning algorithm

The extracted features for describing the postures are then be fed into the supervised classifier. These are then used for the detection of different postures. In this section, supervised learning methods are described.

6.4.1 Support vector machine based supervised classifier

6.4.1.1 2-class support vector machine

A support vector machine (SVM) is a recently emerging classification technique based on statistical learning theory [104] and it has good generalisation performance compared with the traditional classification methods, such as the nearest neighbour method and neural network based techniques. For a two class classification problem with

the training dataset: $\{\mathbf{x}_i, y_i\}, y_i \in \{-1, 1\}, \mathbf{x}_i \in \mathbf{R}^d$, a hyperplane $h(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b$ in a particular feature space is estimated to separate these two classes while making the margin (the smallest distance between the hyperplane and any of the samples) maximum, as presented in Figure 6.5.

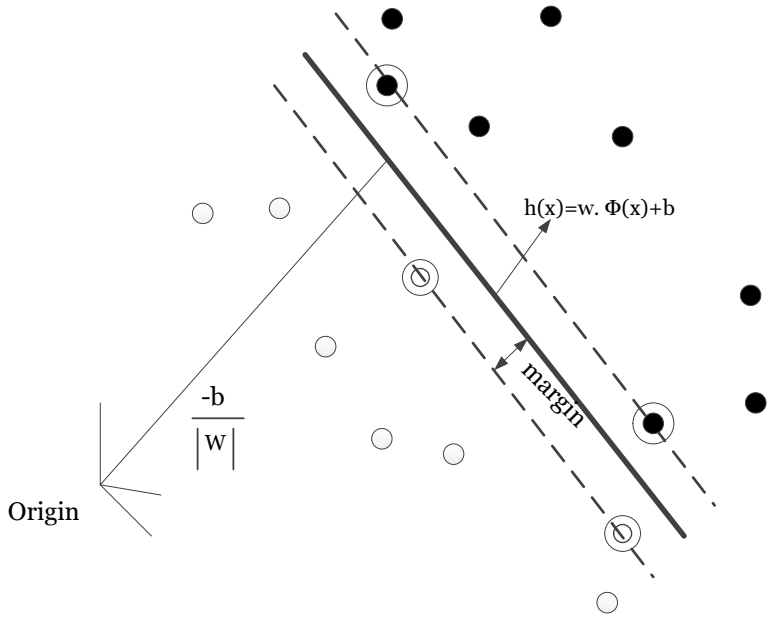


Figure 6.5. The illustration of a hyperplane to separate samples from two classes (white and black) in a particular feature space.

In order to obtain the hyperplane, the following quadratic problem needs to be solved to obtain the hyperplane's parameters $\mathbf{w}$ and b:

$$\begin{aligned} \min_{\mathbf{w},\xi,b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) - b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned} \tag{6.4.1}$$

where $\mathbf{w}$ and b are the parameters determining a hyperplane, $\xi = [\xi_1, ..., \xi_N]$ are slack variables to cope with noise, C is called a penalty parameter which balances the noises and the margin and $\Phi(\cdot)$ is a map-

ping operation which maps the original sample into a feature space for better separation purpose as mentioned in [60] and [104].

A Lagrangian function L is obtained by introducing multipliers $\alpha = [\alpha_1, ..., \alpha_N], \beta = [\beta_1, ..., \beta_N]$ as:

$$L(\mathbf{w}, \xi, b, \alpha, \beta) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_i \xi_i - \sum_i \alpha_i (y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) - 1 + \xi_i) - \sum_i \beta_i \xi_i \tag{6.4.2}$$

The derivatives of the above Lagrangian function with respect to $\mathbf{w}$, ξ and b are set to zeros, which yields:

$$\begin{aligned} \mathbf{w} &= \sum_i \alpha_i y_i \Phi(\mathbf{x}_i) \\ \alpha_i &= C - \beta_i \\ \sum_i \alpha_i y_i &= 0 \end{aligned} \tag{6.4.3}$$

From the results of (6.4.2), the Lagrangian function L in (6.4.6) then becomes:

$$\begin{aligned} L(\mathbf{w}, \xi, b, \alpha, \beta) &= \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_i \xi_i - \sum_i \alpha_i (y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) - 1 + \xi_i) - \sum_i \beta_i \xi_i \\ &= \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) + C \sum_i \xi_i - \sum_{ij} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \\ &- b \sum_i \alpha_i y_i + \sum_i \alpha_i - \sum_i \alpha_i \xi_i - \sum_i \beta_i \xi_i \\ &= -\frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) + \sum_i \alpha_i \end{aligned} \tag{6.4.4}$$

According to [105], a dual form of the problem (6.4.1) is obtained by maximizing (6.4.4) with respect to α considering the constraints of α

in (6.4.3). Thus,

$$\max_{\alpha} \quad -\frac{1}{2}\sum_{ij}\alpha_i\alpha_j y_i y_j \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)+\sum_i \alpha_i$$
$$\text{subject to } 0\le\alpha_i\le C, \quad \sum_i \alpha_i y_i = 1 \tag{6.4.5}$$

can be obtained. Besides, a Gaussian kernel function $k(\mathbf{x},\mathbf{y})$ is applied to replace the dot product of samples in the feature space $(\Phi(\mathbf{x})\cdot\Phi(\mathbf{y}))$, and (6.4.6) is rewritten as:

$$\max_{\alpha} \quad -\frac{1}{2}\sum_{ij}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j)+\sum_i \alpha_i$$
$$\text{subject to } 0\le\alpha_i\le C, \quad \sum_i y_i\alpha_i = 1 \tag{6.4.6}$$

From the relationship of $\mathbf{w}$ and α as mentioned in (6.4.3), the solution of α in (6.4.5) (denoted as α^*) is related to the solution of $\mathbf{w}$ in (6.4.1) (denoted as $\mathbf{w}^*$) with:

$$\mathbf{w}^* = \sum_i \alpha_i^* y_i \Phi(\mathbf{x}_i) \tag{6.4.7}$$

The solution of parameter b in (6.4.1) (denoted as b^*) can be estimated from the Karush-Kuhn-Tucker (KKT) conditions as mentioned in [105], from which it follows that:

$$\alpha_i^*(y_i(\mathbf{w}^*\cdot\Phi(\mathbf{x}_i)+b^*)-1+\xi_i^*)=0$$
$$\beta_i^*\xi_i^*=0 \tag{6.4.8}$$

where ξ_i^* and β_i^* denote the i^{th} optimal solutions of ξ and β. If $\alpha_i^* \neq 0$ and $\beta_i^* \neq 0$, then from the KKT conditions in (6.4.8):

$$b^* = y_i - \mathbf{w}^* \cdot \Phi(\mathbf{x}_i) \tag{6.4.9}$$

if $\mathbf{w}^*$ is replaced with $\sum_i \alpha_i^* y_i \Phi(\mathbf{x}_i)$ by (6.4.7), then

$$\begin{aligned} b^* &= y_i - \sum_j \alpha_j^* y_j (\Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i)) \\ &= y_i - \sum_j \alpha_j^* y_j k(\mathbf{x}_j, \mathbf{x}_i) \end{aligned} \tag{6.4.10}$$

Finally, after α^* and b^* are obtained, the hyperplane can be determined as:

$$h(\mathbf{x}) = \sum_i \alpha_i^* y_i k(\mathbf{x}_i, \mathbf{x}) + b^* \tag{6.4.11}$$

This value of $h(\mathbf{x})$ is then used to determine to which class the sample $\mathbf{x}$ belongs. For an incoming test sample $\mathbf{x}$, if $h(\mathbf{x}) \geq 0$, then the corresponding symbol y is then determined as '+1' and $\mathbf{x}$ belongs to the '+1' class; or else, the symbol value is -1 and $\mathbf{x}$ belongs to the '-1' class.

6.4.1.2 Directed acyclic graph support vector machine for multi-class classification

The traditional 2-class SVM can only solve the 2-class classification problem, hence the need to consider a multi-class approach. In addition, several schemes have been developed to solve the multi-class classification problem, the representative one-versus-one (1-v-1) cases [106] and the one-versus-rest (1-v-r) [104]. In comparison to these two methods, a new concept called the directed acyclic graph support vector machine (DAGSVM) was introduced by Platt et al. [107], which has a theoreti-

cally defined generalisation error bound and is more efficient than 1-v-1 and 1-v-r schemes with respect to the training and computation time. This DAGSVM is the same as operating on a list, initialised with all classes. Therefore, for input sample data $\mathbf{x}$, it is firstly evaluated by a 2-class support vector machine corresponding to the first and last class element. After the evaluation by the 2-class support vector machine, the sample $\mathbf{x}$ is determined to be one of the two classes and the class element that $\mathbf{x}$ does not belong to will be eliminated from the list. This procedure is repeated until only one class element remains in the list and this class element is taken as the class to which $\mathbf{x}$ belongs. In this way, it can be seen that for a problem with N classes, N-1 decisions will be evaluated in order to derive an answer.

An example of the decision procedure of the DAGSVM for a four class classification problem is shown in Figure 6.6. This figure presents a

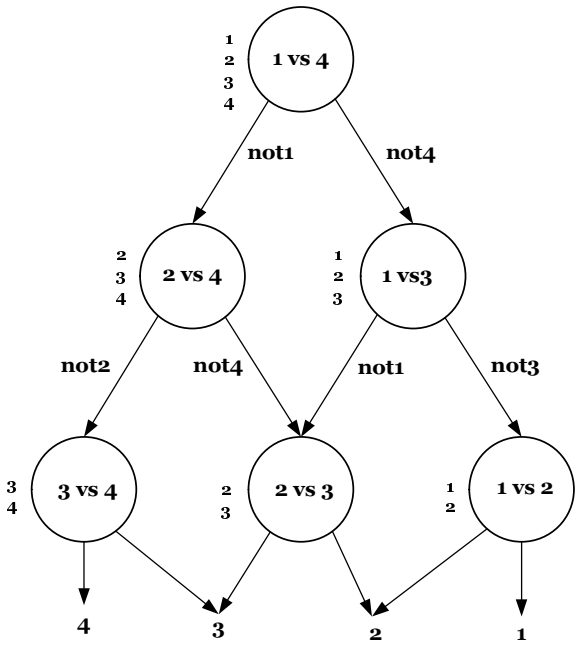


Figure 6.6. The decision process for the traditional DAGSVM for a four class problem [107]

tree-like structure and each node in this structure corresponds to a 2-class SVM. The decision process just follows the structure and is based on a sequence of 2-class operations, a decision is made when a bottom node is reached.

As for all other classifiers, one step for training the DAGSVM classifier is to determine optimal parameters so that the classifier can achieve the optimal performance. There are two sets of parameters required to be tuned for the DAGSVM classifier:

1. The list sequence for the DAGSVM, which is equivalent to the sequence of different 2-class support vector machines for making a decision as shown in Figure 6.6; the list sequence is related to the performance of the DAGSVM and a proper list sequence is essential to guarantee good performance.

2. The parameters of each 2-class SVM node, including the kernel parameters (Gaussian kernel is used here for non-linear classification) and the penalty parameter in the 2-class SVM scheme, also need to be tuned optimally.

Although traditional cross-validation [60] can be used to find the optimal parameters, when there is a large number of parameters needed to be tuned, it proves to be time consuming. As such, in this research, a new parameter optimisation scheme is proposed to reduce the training time to a large extent. For tuning the kernel parameters, the concept of distance between two classes (DBTC) is exploited, as shown in [108]. The calculation of DBTC in a feature space with n_1 and n_2 samples for

two classes is defined as follows:

$$\begin{aligned}\text{DBTC} &= \|\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi\|^2 \\ &= (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi) \\ &= \frac{1}{n_1^2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} k(\mathbf{x}_{1,i}, \mathbf{x}_{1,j}) + \frac{1}{n_2^2}\sum_{i=1}^{n_2}\sum_{j=1}^{n_2} k(\mathbf{x}_{2,i}, \mathbf{x}_{2,j}) \\ &- \frac{2}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} k(\mathbf{x}_{1,i}, \mathbf{x}_{2,j})\end{aligned} \tag{6.4.12}$$

where $k(\mathbf{x}, \mathbf{y})$ is the kernel function mentioned previously and $\mathbf{m}_1^\Phi$, $\mathbf{m}_2^\Phi$ are the means of the two classes in the feature space: $\mathbf{m}_1^\Phi = \frac{1}{n_1}\sum_{i=1}^{n_1}\Phi(\mathbf{x}_{1,i})$ and $\mathbf{m}_2^\Phi = \frac{1}{n_2}\sum_{i=1}^{n_2}\Phi(\mathbf{x}_{2,i})$. For the Gaussian kernel used in this work, the following equation holds:

$$\begin{aligned}\text{DBTC} &= (2 - \frac{2}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} k(\mathbf{x}_{1,i}, \mathbf{x}_{2,j})) \\ &- (1 - \frac{1}{n_1^2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} k(\mathbf{x}_{1,i}, \mathbf{x}_{1,j})) \\ &- (1 - \frac{1}{n_2^2}\sum_{i=1}^{n_2}\sum_{j=1}^{n_2} k(\mathbf{x}_{2,i}, \mathbf{x}_{2,j})) \\ &= \frac{d(C_1, C_2)}{n_1 n_2} - \frac{d(C_1, C_1)}{n_1^2} - \frac{d(C_2, C_2)}{n_2^2}\end{aligned} \tag{6.4.13}$$

where $d(C_i, C_j)$ is calculated as: $d(C_i, C_j) = \sum_{\mathbf{x}\in C_i}\sum_{\mathbf{y}\in C_j}\|\Phi(\mathbf{x}) - \Phi(\mathbf{y})\|^2$, which measure the distance between two classes.

From equation (6.4.13), it can be observed that a large DBTC value means a large inter-distance value $d(C_i, C_j)$ and small intra-distance values $d(C_i, C_i)$ and $d(C_j, C_j)$, imply, the two classes have high separation level.

For each two class SVM, the optimal Gaussian kernel parameter can then be obtained by maximising the DBTC value between the corresponding two classes, which reduces the training time because compared

with the traditional cross-validation method [101] to tune this parameter, the time consuming 2-class SVM training procedure is avoided. The cross-validation method is still employed in tuning the penalty parameter of the 2-class SVM. Assuming that the parameters for every 2-class SVM node have already been tuned, the 2-class SVMs are trained and the DBTCs are calculated under the tuned parameters to obtain two lists:

$$DBTC_{list} = DBTC_{1,1}, ..., DBTC_{1,n}, ..., DBTC_{i,i+1}, ..., DBTC_{i,n}, ..., DBTC_{n-1,n}$$
$$SVM_{list} = SVM_{1,1}, ..., SVM_{1,n}, ..., SVM_{i,i+1}, ..., SVM_{i,n}, ..., SVM_{n-1,n} \quad (6.4.14)$$

For an incoming sample, the sequence of different 2-class support vector machines for making a decision is determined by $DBTC_{list}$, which is summarised in Table 6.1 and a four class example is presented in Figure 6.7. The procedure in Table 6.1 guarantees that at every step,

Table 6.1. Optimal sequence of 2-class support vector machines for decision making

Step 1	Initially, the largest value in the $DBTCl_{ist}$ is chosen. Assuming the largest value is $DBTC_{x,y}$, $SVM_{x,y}$ is then used to make a decision.
Step 2	After the decision, one class is eliminated. Assuming the eliminated class is x, all the $DBTCs$ and $SVMs$ whose indexes contain x will be eliminated from the $DBTC_{list}$ and SVM_{list}.
Step 3	Choosing the largest value among the $DBTC$ values whose index contain y for the remaining elements in the $DBTC_{list}$. And the corresponding SVM in the SVM_{list} will be applied for the second round classification.
Step 4	Repeating step 2-3, until one element is left in $DBTC_{list}$ and SVM_{list}, and the final classification is then made.

the two classes used to build up the 2-class SVM for classification are always the most separable (with the largest DBTC value in the current $DBTC_{list}$), thus good generalisation performance can be achieved. One other merit of this decision making scheme is that it avoids the trial of every possible list sequence of DAGSVM for cross validation, thus greatly saving on the parameter optimization time. For this particular

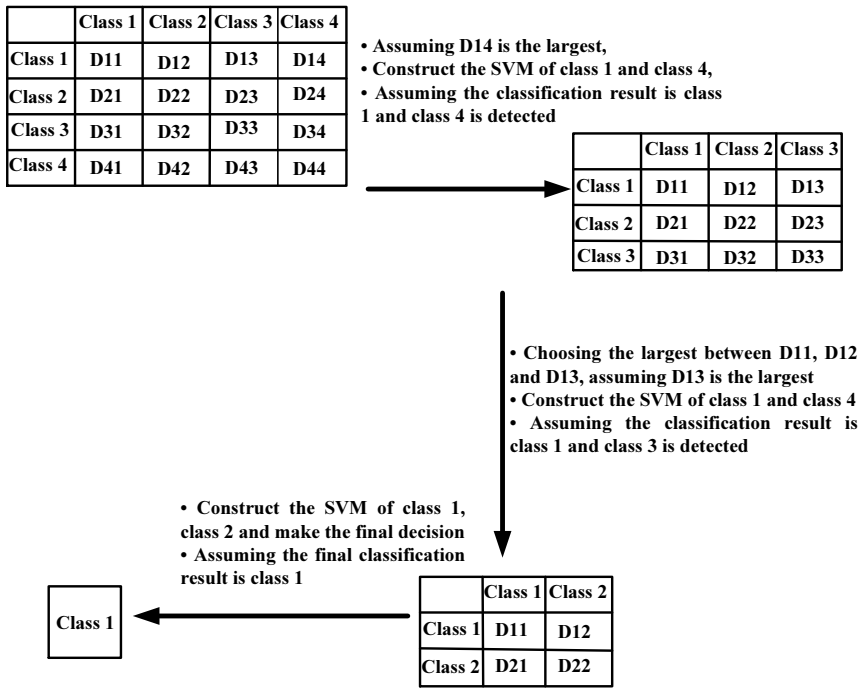


Figure 6.7. The decision process for the DAGSVM based on the DBTC values for a four class example, the DBTC value between classes i and j is denoted as D_{ij}.

fall detection problem, the projection histogram features of a posture and features obtained from the posture's ellipse fitting result (the ratio between the major and minor axes rho and the orientation angle $theta$) are used for DAGSVM construction or posture classification.

In this sub-section, a supervised classification method for classifying different types of postures for fall detection is proposed.

The classification results, together with certain rules can be used together to construct a robust fall detection system, the corresponding rules are presented in the next section.

6.5 Rules used for fall detection

After classifying a particular posture by a supervised classifier, some rules can be used to further confirm whether a fall has happened or not, these rules and the results obtained from postures are used together to achieve a robust fall detection system.
A fall activity is determined if the following three rules are met:

1. The posture is classified as 'lie' or 'bend'.
2. The posture is inside the ground region.
3. The above two conditions are kept for a certain time, which exceeds a preset time threshold (20s is used).

These three rules are set according to the characteristic of fall activity, in most cases, fall activities end up with a 'lie' posture and this 'lie' posture usually remains for a certain time due to the period of immobility of an elderly person after the fall. Compared to lying on the bed/sofa, the posture should be inside the ground region (or at least a large part is inside the ground region). Also it is considered that an elderly person rarely 'bends' for a long time in the ground region (here the 'bend' class is defined as postures of bending to fasten a shoe lace or bending to pick up something, which very commonly occurs in an elderly person's daily life). So that if a 'bend' posture is detected in the ground region for a certain time, it is also regarded as an abnormal

activity (this can happen when an elderly person falls while ending up with a bend-like posture, an example will be given in the experimental section).

In order to detect falls by the three rules, the ground region needs to be determined initially. Before the fall detection phase, floor detection is carried out. In this phase, when the posture is classified as stand or sit, the region nearby the lower extreme point of the ellipse (an 8×8 block is used here) is marked as the ground region. Figure 6.8 shows the result of floor detection. Note, as shown in (d) and (e), sometimes the furniture is moved and the floor region has to be updated accordingly. The detected floor region is extremely helpful to distinguish a fall on the floor from lying on a sofa, which is shown in the experimental part.

The flowchart of the proposed supervised fall detection system is presented in Figure 6.9. An efficient codebook background subtraction algorithm is initially applied to extract the human body foreground and some post-processing is applied to improve the results. From the extracted foreground silhouette, features are extracted from the fitted ellipse and projection histogram, which are used for classification purposes. These features are fed into the DAGSVM (which is trained from a dataset containing features extracted from different postures in different orientations) and the extracted foreground silhouette is classified as one of four different postures (bend, lie, sit and stand). The classification results, together with the detected floor information, are then used to determine fall or non-fall activities.

The evaluations of performance of this supervised fall detection systems are presented in the experimental part.

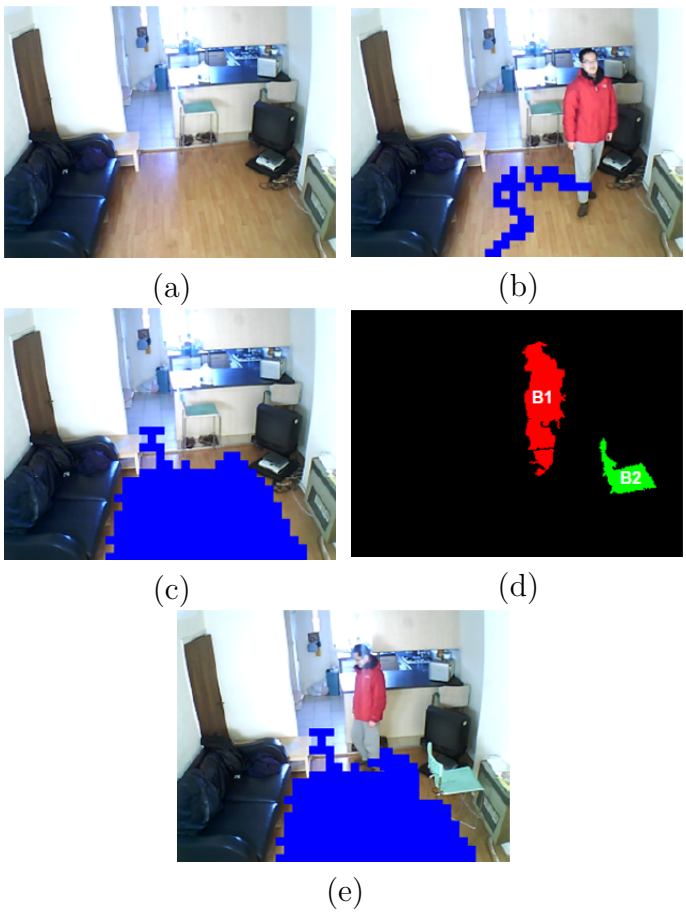


Figure 6.8. The floor detection results. (a) Original image; (b) Detected floor region while a person is walking; (c) Floor detection result after some time; (d) More than one blob after the furniture is moved, the moving blob (human body) is marked in red, the static blob (furniture) is marked green; (e) The updated floor region result after moving furniture. The region nearby the new position of the furniture is unmarked and that nearby the person's feet is marked as the floor region.

6.6 Experimental analysis

In this section, the performance of the proposed supervised fall detection system will be presented. The experiments were carried out in a simulated real elderly home environment. A USB camera was used for recording the real video sequence with the image size of 320×240, the recorded video sequence is processed by using VC++ 6.0 (with OpenCv library 1.0) and MATLAB on an Intel(R) Core(TM)2 Duo CPU lap-

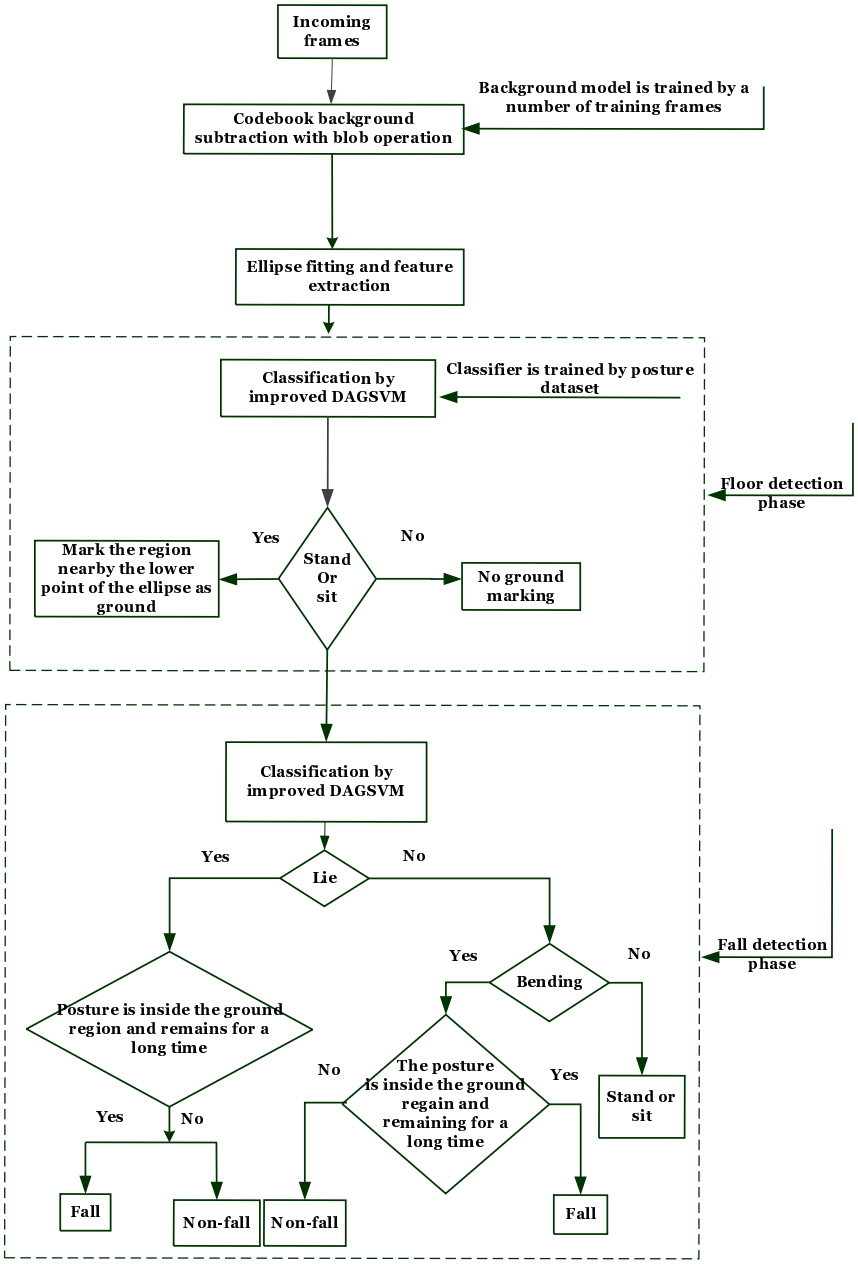
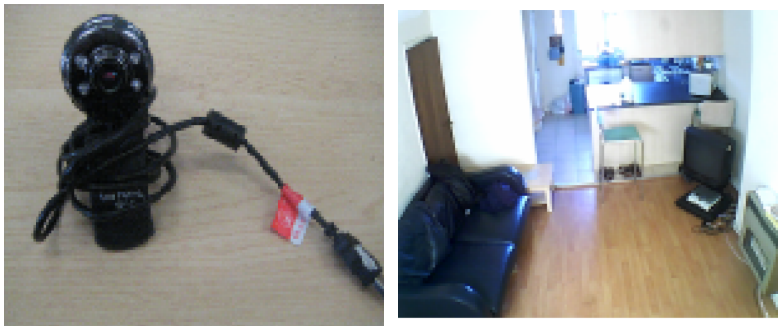


Figure 6.9. The flow chart of the proposed supervised DAGSVM classifier based fall detection system.

top with 1.00GB Memory. Figure 6.10 shows the camera used in the experiment and the background image it records.



(a) (b)

Figure 6.10. (a) The USB camera used in the experimental room environment. (b) The experimental environment.

6.6.1 Background subtraction results

Some background subtraction results in this more realistic home environment are shown in Figures 6.11, 6.12 and 6.13, in which three challenging scenarios which occur commonly in a home environment are analysed. Initially, a short video-camera clip which contains 100 frames is applied for training the original background model, which will be updated with the evolution of time.

Figure 6.11 shows the background subtraction results at different times of day with gradual light change. In Figure 6.11 images (a) and (c) show a frame captured at noon time and the corresponding background subtraction result. Figure 6.12 shows the background subtraction results in the presence of moving objects. The background model is updated to cope with the gradual light change and the results are shown in (b) and (d), where (b) is a frame captured later in the afternoon and (d) is the background subtraction result with the updated background model. (a) shows four frames sampled in a short video sequence, which shows that a person moves the table and fruit plate. (b) shows the background subtraction results by directly applying the codebook background subtrac-

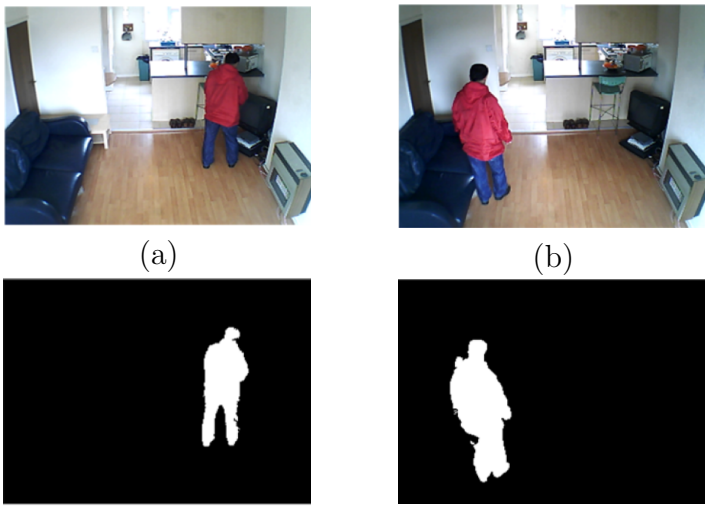


Figure 6.11. Background subtraction results at different times of a day; (a) and (c) show an original frame captured at noon time and the corresponding BGS result; (b) is a frame captured in the afternoon with the light condition changed and (d) is the BGS result of (b) with the updated background model.

tion method. It can be seen that there are many segmentation errors due to the movement of the table and fruit plate. Frame differencing results are shown in (c), which indicate active pixels and help to find the active blob representing the human body. By the post-processing technique as discussed in Subsection 6.2.1, improved background subtraction results are obtained in (d).

Figure 6.13 shows a case of sudden illumination change. At frame (c), the light is turned on and a large illumination change can be observed. This sudden change of illumination can be detected by the frame differencing result as shown in (g), with more than 50% of pixels being marked as active ones (white). The background model is then retrained to cope with this sudden illumination change. As shown in (d) and (h), good segmentation is obtained by the retrained background model.

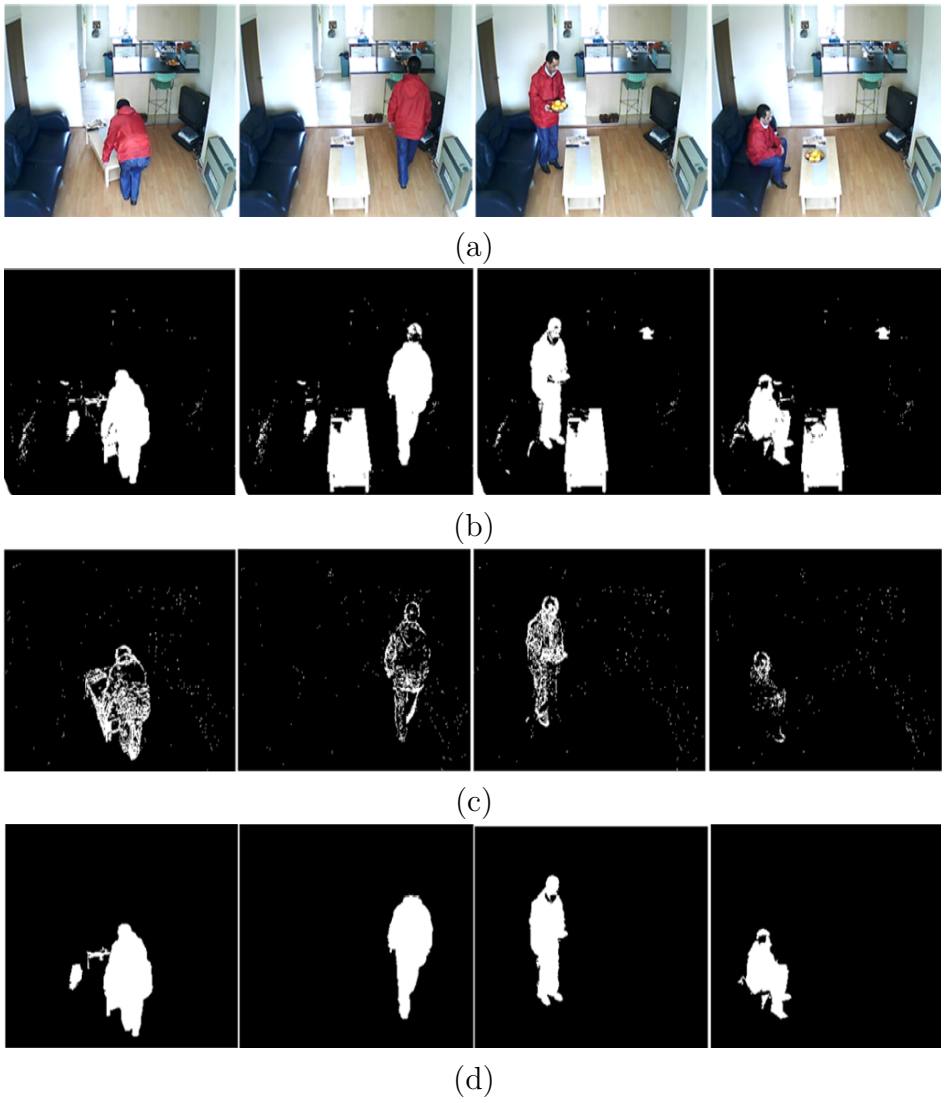


Figure 6.12. Background subtraction results for moving furniture. (a) Shows original frames of a person moving the table and fruit plate. Codebook background subtraction results are shown in (b), (c) Shows the frame differencing results which indicate active pixels. From the frame differencing results and blob operations, improved background subtraction results are obtained in (d).

6.6.2 Results for the supervised fall detection system

For the experiment of the fall detection based on supervised classifiers, 15 people (11 males and 4 females) were invited to attend the experiments for simulating different postures and activities. The char-

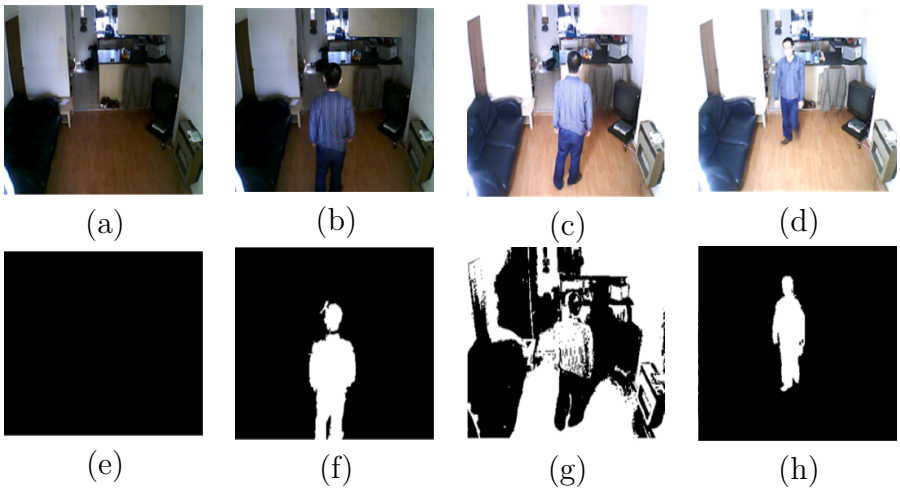


Figure 6.13. Background subtraction for sudden illumination change. Frames (a) and (b) are captured with the light off, at frame (c) the light is turned on and a drastic illumination change can be observed. Frame (d) is captured after the light is turned on for a certain time. Frames (e) and (f) are the background subtraction results of (a) and (b). Image (g) is the frame difference result for (c), sudden illumination is detected because more than 50% of the pixels are marked as active ones and the background model is retrained. Frame (h) is the subtraction result of (d) by the retrained background model.

acteristics of the 15 people are summarised in Table 6.2. It has to be noted that real elderly people were not invited to participate in the experiments because it is unsafe for an elderly person to simulate fall activities; instead, younger people were invited to mimic elderly people.

Table 6.2. The characteristics of 15 participators in the experiments.

No. of people	15
Male/Female	11/4
Age	25-49
Weight	57-94 (kg)
Height	158-187 (cm)

6.6.3 Posture classification results

To form the posture dataset, 3200 postures (comprising of 800 stands, 800 sits, 800 lies and 800 bends) from 15 people were recorded. As in [109], each person was asked to simulate postures in different directions. This ensures that the constructed classifier is robust to view angles. Some samples are shown in Figure 6.14.

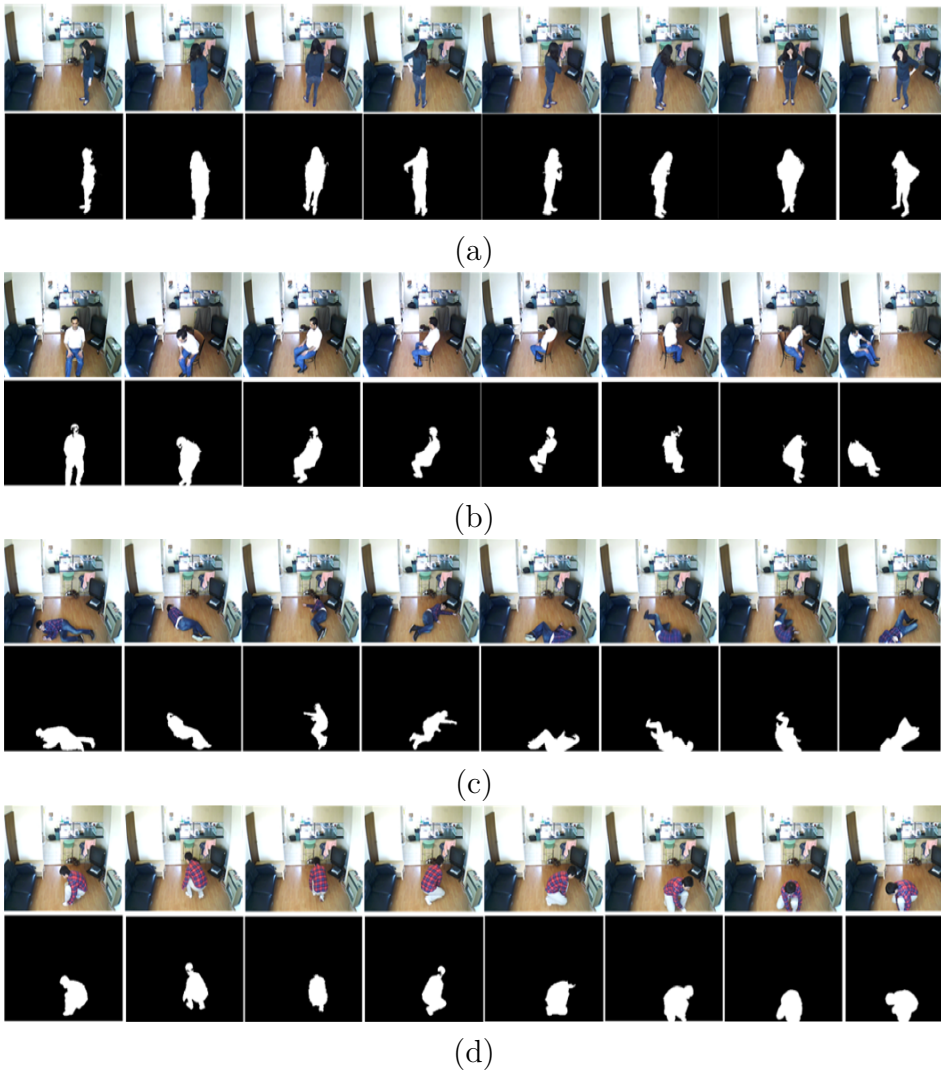


Figure 6.14. Posture samples simulated by different participates in different orientations: (a) Stand (b) Sit (c) Lie and (d) Bend.

Then a 10-fold validation [60] was applied to evaluate the performance

of the posture classification. Three types of comparisons were made: The first evaluation is the feature comparison, shown in Table 6.3. The feature classification results are compared when using the ellipse features or projection histogram features alone and when using a combination of these two features. The DAGSVM is then applied to classification.

Table 6.3. Classification result by different types of features.

	Global features	Local features	Combined features
Classification rate	89.72%	76.70%	96.09%

Table 6.3 shows that, the classification result by the combined feature presents a higher classification rate than using either feature alone. For the second assessment, two types of parameter optimisation methods are compared for DAGSVM: the 10-fold validation based method and the DBTC based method, which is shown in Table 6.4.

Table 6.4. Comparison of different parameter optimisation methods.

	10-fold validation method	DBTC based method
Training time (s)	2453.40	397.87
Classification rate	95.63%	96.09%

From Table 6.4, it can be seen that the DBTC based parameters optimisation method can greatly reduce the training time of DAGSVM while achieving an even slightly better performance. For this reason, the DBTC based parameters optimisation method is preferred in this research work. In the third assessment, the performance of DAGSVM is compared with other traditional classifiers K-nearest neighbour (KNN),

multi-layer perceptron neural network (MLPNN) , naive bayes classifier (NBC) and binary decision tree (BDT). All classifiers are implemented using PRtools, a well-known software package for pattern recognition [110]). The results are presented in Table 6.5. To get a fair result, the parameters of the comparison classifiers are tuned to be optimal by using the cross validation function in PRtools. Table 6.5 presents the results from the cross validation tests. It can be observed

Table 6.5. Comparison of different classifiers.

	DAGSVM	KNN	MLPNN	NBC	BDT
Classification rate	96.09%	92.64%	92.53%	75.27%	83.72%

that DAGSVM achieves the best performance with a slightly better classification rate performance than k-nearest neighbour and the MLP neural network.

It should be noted that in the real life scenario, the training dataset is not usually perfect and it is common to find some outliers existing in the training dataset. Outliers in posture classification are mainly caused by extremely bad background subtraction results and wrong labeling (for example, the feature of one class may be mislabelled as another class). In Table 6.6, the comparison results of the classifiers are presented on a dataset which includes 10% outliers. This table presents another ad-

Table 6.6. Comparison of different classifiers on a dataset which is corrupted by 10% outliers.

	DAGSVM	KNN	MLPNN	NBC	BDT
Classification rate	95.51%	84.07%	85.93%	72.42%	58.72%

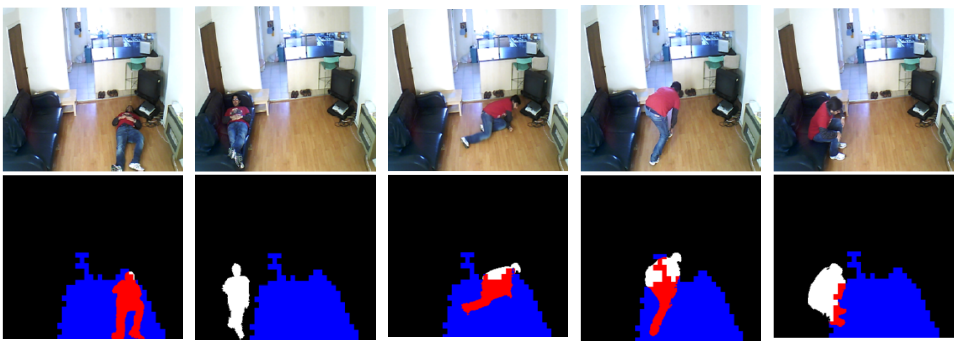
vantage of the classification rate of the DAGSVM over other classifiers on this noisy dataset. Also, compared with other classifiers, DAGSVM

is the most robust to such noise due to the reason that slack variables are introduced in the 2-class SVMs of the DAGSVM classifier to cope with the noises as mentioned in Section 6.4 (there is only a 0.58% drop in classification rate compared with the result of Table 6.5. Therefore, from the results presented in Table 6.5 and Table 6.6, DAGSVM performs better for posture classification than other traditional classifiers.

6.6.4 Fall detection by the supervised directed acyclic graph support vector machine classifier

Using the three conditions presented in the previous section, posture classification results along with the detected floor information may be used to detect falls. To illustrate this further, five cases are presented in Figure 6.15. Figure 6.15, (a) shows a person who has fallen on the floor, and a 'lie' posture is detected with most parts of the human body region in the detected ground region; in addition, this posture is kept for a certain time (longer than 20s), therefore, a fall is detected. For Figure 6.15 (b), although a 'lie' posture is detected, the human body blob is not in the floor region, so the lying on the sofa case is correctly classified as non-fall. In figure 6.15 (c), (d) and (e), the postures are all classified as 'bend'. For figure 6.15 (c), a large portion of the human body is in the ground region and the 'bend' posture remains for longer than 20s. It is assumed to be generally impossible for an elderly person to bend for a long time in the ground region, so this is classified as an abnormal activity and it is detected as a fall. For (d) and (e), either the detected 'bend' posture does not hold for a long time (for case (d), a person ties his shoe lace and the 'bend' posture recovers to 'stand' posture in a short time), or the posture is not in the ground region (only

a small portion of the human body region is in the ground), so they are not detected as falls. To evaluate this fall detection system, each



(a) (b) (c) (d) (e)

Figure 6.15. Different cases of fall and non-fall activities. a) Fall on the floor; (b) Lie on the sofa; (c) Fall on the floor; (d) Bend to fasten the shoe tie; (e) Sit on the sofa. For (a) and (b), the postures are classified as 'lie' and for (c), (d) and (e), the postures are classified as 'bend'. The blue region represents the detected floor, the red region represents the intersected part of the foreground human body with the detected floor region, the white region represents the foreground human body part which is not intersected with the detected floor region and the remaining background region is marked as black. The proposed system successfully classifies (a), (c) As falls and (b), (d) and (e) as non-falls.

person is asked to simulate 16 fall activities and 16 non-fall activities in different directions. A total number of 240 fall activities and 240 non-fall activities are recorded. These are used to evaluate the proposed fall detection system. To classify an individual activity, the postures from other people in the recorded posture dataset are used to construct the DAGSVM classifier. The final results are given in Table 6.7. The table shows that 233 out of 240 (97.08%) falls can be detected while only 2 out of 240 (0.8%) non-falls are mistaken as falls. A high fall detection rate (true positives) is obtained with a very low false alarm rate.

Table 6.7. Evaluation of the proposed fall detection system.

	No. of activities	Detected as falls	Detected as non-falls
Falls	240	233	7
Walking around	60	0	60
Sitting on the sofa/chair	60	2	58
Bending	60	0	60
Lying on the sofa	60	0	60

6.7 Summary

This chapter presented an efficient fall detection system based on a supervised approach, the advantages of this proposed system include:

1. Using the codebook background subtraction technique, extraction of a silhouette was performed with further post-processing techniques applied to solve the background subtraction errors caused by environmental changes in a real home environment.

2. The combination of ellipse and projection histogram features were used, and an improved version of DAGSVM classifier was proposed.

3. Classification results of DAGSVM, together with some properly designed rules were effectively used to distinguish falls and non-falls.

Experimental results showed that acceptable fall detection results can be obtained for the supervised fall detection system. The supervised fall detection system has both merits and drawbacks. One demerit is that at the initial stage, a large posture dataset composed of postures

from different persons is needed for training the classifier, this is inconvenient. However, the trained classifier can be immediately applied for classifying postures for fall detection for a particular elderly person as long as the classifier is well trained.

Besides the information obtained from postures, as a means of improving the system, other types of information can be applied to achieve a more robust fall detection system. One such type of information can be obtained from audio as proposed in [111] and [112]. The floor sounds generated for fall activities and non-fall activities are different, and this sound information is helpful to distinguish fall activities and non-fall activities when poor results of posture features are obtained (such as the elderly person wearing clothes whose colour is very similar to the background thus generating bad posture segmentation results).

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

This thesis proposed different types of intelligent vision based fall detection techniques for detecting fall activities of an elderly person living alone at home. Three background subtraction techniques were used to develop a robust video processing technique that reliably extracted the presence of a person in the recording of a room environment based on background subtraction techniques. Two-dimensional and three-dimensional features were extracted from video sequences recorded by one or multiple video-cameras, and an analytical algorithm (Chapter 3 and Chapter 4) and machine learning algorithms (Chapter 5 and Chapter 6) were then exploited to process the extracted features for detecting fall activities. Evaluations were performed in both the intelligent signal processing lab at Loughborough University and also in a real home environment.

In Chapter 3, background subtraction techniques based on AMF, MOG and CB were compared and datasets were used and tested based on different lighting conditions (good lighting, poorer lighting conditions and

typical lighting conditions). The results showed that the CB method based BGS technique, provides a fixed background model after training. This implies that new modalities in the scene, including foreground items that remain static for long periods of time will never become part of the background model as they would do in the other methods. Therefore, the CB method of BGS is the one among the three popular BGS methods that is most suited to robust person extraction in an indoor environment.

Chapter 4 gives a technical overview of the video-camera calibration based on Tsai's model. By applying the Tsai's camera calibration using a set of correspondent points (three-dimensional points and corresponding two-dimensional image points), both the external and internal parameters of a video camera can be estimated. This calibration is a very important step to obtaining certain types of three-dimensional information (such as the three-dimensional position for a particular pixel in the image).

Chapter 5 proposed a fall detection scheme based on three-dimensional features extraction and single Gaussian model based method to distinguish unusual behaviour (falls) for enclosed environments. Video-cameras were firstly calibrated by the popular Tsai's camera calibration method and a three-dimensional person was then constructed from the obtained CB BGS results from two calibrated cameras. Three-dimensional features, including the three-dimensional position, velocity and orientation information corresponding to fall activities were extracted to build the model for distinguishing fall activities and non-fall activities. Single Gaussian models were constructed from a training dataset including 80 short video clips of different fall activities simu-

lated by one person. From these models, three-dimensional features were extracted for model construction. The test dataset included 40 fall activities and 40 non-fall activities simulated by the same person. Results showed that the single Gaussian model achieved the best performance with 100% fall detection rate and 0% false detection rate with the optimal threshold.

Chapter 6 presented fall detection systems based on supervised learning methods. The codebook background subtraction approach was used to extract the postures and certain post-processing techniques were applied to solve the background subtraction errors caused by some environmental changes in a real home environment. Some features (ellipse features and projection histogram) which can describe postures in detail were extracted and used to construct the corresponding DAGSVM classifier with some simple features (ellipse features and position features). The classification results of DAGSVM, together with certain rules were used to distinguish falls from non-falls. Experimental results in a real home environment showed that acceptable fall detection results can be obtained, with 97.08% fall detection rate and 0.8% false detection rate for a 15 person dataset by using proposed supervised fall detection system.

In summary, this thesis has proposed effective schemes for solving the problems listed at the end of Chapter 2:

1. The codebook background subtraction technique is improved to obtain better background subtraction results in a home environment (e.g. by head tracking in Chapter 3 and some other advanced post-processing techniques in Chapter 6).

2. Directional invariant three-dimensional features were extracted

from the reconstructed 3-D person. These were obtained from the background subtraction results from video sequences recorded by calibrated video-cameras, and the three-dimensional features were used to train the model to distinguish fall and non-fall activities by using only one threshold.

3. Supervised classifiers based on posture features, with certain types of information (floor information and movement amplitude information) were applied to construct more robust fall detection systems. These were then thoroughly evaluated on datasets recorded in a real home environment representative of an assisted living application.

7.2 Future work

The research work can be extended in different ways such as algorithm aspects, information aspects and hardware equipment aspects:

From algorithm aspects, elegant intelligent vision algorithms can be applied as the components of the proposed fall detection system. A more efficient background subtraction method as proposed in [113] can be applied for better human body segmentation, with online self-adaptive mechanism to update model parameters.

In this way, the change of the illumination, which is a common phenomenon in the indoor environment, could be avoided.

On the other hand, considering that there could sometimes be more than one moving object at home (such as the elderly person with a pet), some object classification algorithms in [114] and [115] could be applied to determine the moving blob corresponding to the human re-

gion. These algorithms can be added into the current fall detection system as new modules, which is helpful in enhancing the performance of the proposed fall detection systems.

From information aspects, instead of using only the video information, more types of information (such as acoustic information) could be extracted to compensate for the limitations of video information (such as poor video information will be obtained when an elderly person wears clothes whose colour is similar to the background). For hardware equipment aspects, a more robust multimodal fall detection system could be constructed by fusing different types of information (video information and audio information) for improving the performance.

The sound and posture information together can be used to compose a more robust multimodal (audio and video) fall detection system, which is a possible next research.

The limitation of the audio information is that it can easily be affected by background noises, especially TV; however, this problem can be ameliorated by using the modern Blind Source Separation (BSS) technique [116] and [117] to reduce this type of interference.

Instead of using only ordinary video-cameras, more advanced hardware equipment could be used. As presented in [118], a new-emerging Kinect sensor could be applied, which can obtain additional depth information for better human body segmentation when an elderly person wears clothes whose colour is similar to the background.

References

[1] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *BioMedical Engineering OnLine*, vol. 12, no. 66, pp. 1–24, 2013.

[2] S. Paul, S. Chaplin, and R. Legood, "Incidence and costs of unintentional falls in older people in the united kingdom," *Journal of epidemiology and community health*, vol. 57, no. 9, pp. 740–744, 2003.

[3] M. Rantz, M. Skubic, C. Abbott, C. Galambos, Y. Pak, D. Ho, E. Stone, L. Rui, E. Back, and S. Miller, "In-home fall risk assessment and detection sensor system," *Journal of GerontoloGical nursing*, vol. 39, pp. 18–22, 2013.

[4] Nuffield Institute for Health and NHS Centre for Reviews and Dissemination, "Preventing falls and subsequent injury in older people," *Bulletin*, vol. 2, pp. 1–16, 1996.

[5] M. Williams, "Falls, injuries due to falls, and the risk of admission to a nursing home," *New England Journal of Medicine*, vol. 337, pp. 1279–1284, 1997.

[6] J. Rizzo, R. Friedkin, C. Williams, J. Nabors, D. Acampora, and M. Tinetti, "Health care utilization and costs in a medicare population by fall status," *Medical Care*, vol. 36, no. 8, pp. 1174–1188, 1998.

[7] S. Abbate, M. Avvenuti, P. Corsini, J. Light, and A. Vecchio, "Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor network: a survey," *Wireless Sensor Networks: Application-Centric Design*, pp. 147–166, 2010.

[8] T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," IEEE *International Conference on Computer Vision* (ICCV)*, Corfu, Greece*, vol. 99, pp. 1–19, 1999.

[9] Ed Pilkington, "Census population ageing global," tech. rep., Ageing population, http://www.guardian.co.uk/world/2009/jul/20/census-population-ageing-global , 2009.

[10] R. Suzman and J. Beard, "Global health and aging," *United Nations, World Population Prospects, The 2012 Revision, http://esa.un.org/unpd/wpp.*, pp. 1–25, 2012.

[11] F. Kemp and R. Acheson, "Care in the community elderly people living alone at home," *Community Medicine*, vol. 1, no. 1, pp. 21–26, 1989.

[12] L. Lin, F. Chiou, and H. Cohen, "Slip and fall accident prevention: a review of research, practice and regulations," *Journal of Safety Research*, vol. 26, no. 4, pp. 203–212, 1995.

[13] S. Sadigh, A. Reimers, R. Andersson, and L. Laflamme, "Falls and fall-related injuries among the elderly: a survey of residential-care facilities in a Swedish municipality," *Journal of Community Health*, vol. 29, no. 2, pp. 129–140, 2004.

[14] R. Sutherland, "Number of older people admitted to hospital in uk increased by two thirds," http://www.cardi.ie/news/numberofolderpeopleadmittedtohospitalinuk increasedbytwothirds, Centre for Ageing Research and Development in Ireland (CARDI), Dublin Office Centre for Ageing Research and Development, 2010.

[15] C. Lord and D. Colvin, "Falls in the elderly: detection and assessment," *Annual international conference of the* IEEE *Engineering in Medicine and Biology Society*, vol. 13, no. 4, pp. 1938–1939, 1991.

[16] X. Yu, "Approaches and principles of fall detection for elderly and patient," *10th International Conference on e-*Health *Networking, Applications and Services-Healthcom, Missouri, Columbia*, pp. 42–47, 2008.

[17] M. Rantz, "TigerPlace: An innovative 'aging in place' community," *University of Missouri, Columbia, http://http://eldertech.missouri.edu/files/Papers/Rantz/AJN-01-201320Profiles-Rantz.pdf*, vol. 113, no. 1, 2013.

[18] E. Stone and M. Skubic, "Passive, in-home gait measurement using an inexpensive depth camera: Initial results," *6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), San Diego, CA*, pp. 183–186, 2012.

[19] M. Gibson, R. Andres, B. Issacs, T. Radebaugh, and J. Peterson, "The presention of falls in later life. A report of the kellogg international work group on the prevention of falls by the elderly," *Danish Medical Bulletin*, vol. 34, no. 4, pp. 1–24, 1987.

[20] S. Lord, C. Sherrington, and H. Menz, "Falls in older people: risk factors and strategies for prevention," *Cambridge University Press*, 2007.

[21] N. Noury, A. Fleury, P. Rumeau, A. Bourke, G. Laighin, V. Rialle, and J. Lundy, "Fall detection-principles and methods," *29th Annual International Conference of the* IEEE *Engineering in Medicine and Biology Society* (EMBS)*, Lyon*, pp. 1663–1666, 2007.

[22] Y. Charlon, N. Fourty, W. Bourennane, and E. Campo, "Design and evaluation of a device worn for fall detection and localization: Application for the continuous monitoring of risks incurred by dependents in an alzheimer care unit," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7316–7330, 2013.

[23] H. Tronics, "A beginner's guide to accelerometers," *http://www.hobbytronics.co.uk/accelerometer-info*, 2011.

[24] P. Veltink, H. Bussmann, W. Vries, W. Martens, and R. Lummel, "Detection of static and dynamic activities using uniaxial accelerometers," IEEE *Transactions on Rehabilitation Engineering*, vol. 4, no. 4, pp. 375–385, 1996.

[25] F. Hijaz, N. Afzal, T. Ahmad, and O. Hasan, "Survey of fall detection and daily activity monitoring techniques," *International Conference on Information and Emerging Technologies* (ICIET)*, Karachi*, pp. 1–6, 2010.

[26] M. Kangas, A. Konttila, I. Winblad, and T. Jamsa, "Determination of simple thresholds for accelerometry-based parameters for fall detection," *29th International Conference of the* IEEE *Engineering in*

Medicine and Biology Society (EMBS)*, Oulu, Finland*, pp. 1367–1370, 2007.

[27] P. Jantaraprim, P. Phukpattaranont, C. Limsakul, and B. Wongkittisuksa, "Evaluation of fall detection for the elderly on a variety of subject groups," *Proceedings of the 3rd International Convention on Rehabilitation Engineering and Assistive Technology, Singapore*, pp. 1–4, 2009.

[28] T. Nguyen, M. Cho, and T. Lee, "Automatic fall detection using wearable biomedical signal measurement terminal," *Annual International Conference of the* IEEE *Engineering in Medicine and Biology Society* (EMBC), pp. 5203–5206, 2009.

[29] S. Zahra, "Gyro and accelerometer fusion," *Engineering Projects Electonics and Control Engineering, http://www.electronics-control.info/gyro-acc.htm*, 2011.

[30] A. Bourke and G. Lyons, "A threshold based fall-detection algorithm using a bi-axial gyroscope sensor," *Medical Engineering and Physics*, vol. 30, no. 1, pp. 84–90, 2008.

[31] L. Tong, W. Chen, Q. Song, and Y. Ge, "A research on automatic human fall detection method based on wearable inertial force information acquisition system," *International IEEE Conference on Robotics and Biomimetics* (ROBIO)*, Guilin*, pp. 949–953, 2009.

[32] M. Nyan, F. Tay, and E. Murugasu, "A wearable system for pre-impact fall detection," *Journal of Biomechanics*, vol. 41, no. 16, pp. 3475–3481, 2008.

[33] Y. Li, Z. Zeng, M. Popescu, and K. Ho, "Acoustic fall detection using a circular microphone array," *Annual 32nd International Conference of the* IEEE *Engineering in Medicine and Biology Society* (EMBC)*, Buenos Aires, Argentina*, pp. 2242–2245, 2010.

[34] M. Popescu, Y. Li, M. Skubic, and M. Rantz, "An acoustic fall detector system that uses sound height information to reduce the false alarm rate," *30th Annual International Conference of the* IEEE *Engineering in Medicine and Biology Society* (EMBS)*, Vancouver, BC*, pp. 4628–4631, 2008.

[35] G. Sessler and J. West, "Self-biased condenser microphone with high capacitance," *Journal of the acoustical society of America*, vol. 34, no. 11, pp. 1787–1788, 1962.

[36] M. Alwan, P. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, "A smart and passive floor-vibration based fall detector for elderly," *2nd Information and Communication Technologies* (ICTTA)*, Damascus, Syria*, vol. 1, pp. 1003–1007, 2006.

[37] J. Tao, M. Turjo, M. Wong, M. Wang, and Y. Tan, "Fall incidents detection for intelligent video surveillance," *Fifth International Conference on Information, Communications and Signal Processing*, pp. 1590–1594, 2005.

[38] M. Wang, C. Huang, and H. Lin, "An intelligent surveillance system based on an omnidirectional vision sensor," IEEE *Conference on Cybernetics and Intelligent Systems*, pp. 1–6, 2006.

[39] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Monocular 3d head tracking to detect falls of elderly people," *28th Annual*

International Conference of the IEEE *Engineering in Medicine and Biology Society* (EMBS)*, New York City, New York,* USA, 2006.

[40] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," *21st International Conference on Advanced Information Networking and Applications Workshops* (AINAW)*, Niagara Falls, Ont.*, vol. 2, pp. 875–880, 2007.

[41] B. Toreyin, Y. Dedeoglu, and A. Cetin, "HMM based falling person detection using both audio and video," IEEE *14th Signal Processing and Communications Applications, Antalya*, pp. 1–4, 2006.

[42] S. Miaou, F. Shih, and C. Huang, "A smart vision-based human fall detection system for telehealth applications," *The Third International Conference on Telehealth* (IASTED), pp. 7–12, 2007.

[43] S. Miaou, P. Sung, and C. Huang, "A customized human fall detection system using omni-camera images and personal information," *1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, Arlington,* VA*,* USA, 2006.

[44] A. Nasution and S. Emmanuel, "Intelligent video surveillance for monitoring elderly in home environments," IEEE *9th Workshop on Multimedia Signal Processing* (MMSP), pp. 203–206, 2007.

[45] T. Lee and A. Mihailidis, "An intelligent emergency response system: preliminary development and testing of automated fall detection," *Journal of Telemedicine and Telecare*, vol. 11, no. 4, pp. 194–198, 2005.

[46] H. Nait-Charif and S. McKenna, "Activity summarisation and fall detection in a supportive home environment," *Proceedings of the 17th*

International Conference on Pattern Recognition (ICPR), pp. 323–326, 2004.

[47] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He, "Recognizing falls from silhouettes," *28th Annual International Conference of the* IEEE *Engineering in Medicine and Biology Society* (EMBS), pp. 6388–6391, 2006.

[48] N. Thome and S. Miguet, "A HHMM-based approach for robust fall detection," *9th International Conference on Control, Automation, Robotics and Vision* (ICARCV), pp. 1–8, 2006.

[49] R. Cucchiara, A. Prati, and R. Vezzani, "A multi-camera vision system for fall detection and alarm generation," *Expert Systems Journal*, vol. 24, pp. 334–345, 2007.

[50] Y. Hsu, J. Hsieh, H. Kao, and M. Liao, "Human behavior analysis using deformable triangulations," IEEE *7th Workshop on Multimedia Signal Processing*, pp. 1–4, 2005.

[51] A. Williams, D. Ganesan, and A. Hanson, "Aging in place: fall detection and localization in a distributed smart camera network," *Proceedings of the 15th International Conference on Multimedia*, pp. 892–901, 2007.

[52] C. Lin, Z. Ling, Y. Chang, and J. Kuo, "Compressed-domain fall incident detection for intelligent home surveillance," IEEE *International Symposium on Circuits and Systems* (ISCAS), pp. 3781–3784, 2005.

[53] C. Lin and Z. Ling, "Automatic fall incident detection in compressed video for intelligent homecare," *Proceedings of 16th Interna-*

tional Conference on Computer Communications and Networks (ICCCN), pp. 1172–1177, 2007.

[54] M. Yu, S. Naqvi, and J. Chambers, "Fall detection in the elderly by head tracking," IEEE *15th Workshop on Statistical Signal Processing* (SSP)*, Cardiff, United Kingdom*, pp. 357–360, 2009.

[55] M. Shoaib, R. Dragon, and J. Ostermann, "View-invariant fall detection for elderly in real home environment," *Fourth Pacific-Rim Symposium on Image and Video Technology* (PSIVT)*, Singapore*, 2010.

[56] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier, "Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution," IEEE *Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 290–300, 2011.

[57] A. Leone, G. Diraco, and P. Siciliano, "An automated active vision system for fall detection and posture analysis in ambient assisted living applications," IEEE *International Symposium on Industrial Electronics* (ISIE)*, Bari, Italy*, pp. 2301–2306, 2010.

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

[59] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier, "Fall detection with multiple cameras: An occlusion-resistant method based on 3D silhouette vertical distribution," IEEE *Transac-*

tions on Information Technology in Biomedicine, vol. 15, no. 2, pp. 290–300, 2011.

[60] C. Bishop, "Pattern recognition and machine learning," *Springer*, 2006.

[61] C. Juang and C. Chang, "Human body posture classification by a neural fuzzy network and home care system application," IEEE *Transactions on Systems Man and Cybernetics Part A: Systems and Humans*, vol. 37, no. 6, pp. 984–994, 2007.

[62] C. Liu, C. Lee, and P. Lin, "A fall detection system using k-nearest neighbor classifier," *Expert Systems with Applications*, vol. 37, no. 10, pp. 7174–7181, 2010.

[63] R. Cucchiara, A. Prati, and R. Vezzani, "An intelligent surveillance system for dangerous situation detection in home environments," *Intelligenza Artificiale*, vol. 1, no. 1, pp. 11–15, 2004.

[64] B. Ni, N. Dat, and P. Moulin, "RGBD-camera based get-up event detection for hospital fall prevention," IEEE *International Conference on Acoustics Speech and Signal Processing* (ICASSP)*, Kyoto, Japan*, pp. 1405–1408, 2012.

[65] F. Bach, G. Lanckriet, and M. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," *Proceedings of the Twenty-first International Conference on Machine Learning, Banff, Alberta, Canada*, pp. 1–8, 2004.

[66] B. Mirmahboub, S. Samavi, N. Karimi, and S. Shirani, "View-invariant fall detection system based on silhouette area and orientation,"

IEEE *International Conference on Multimedia and Expo* (ICME)*, Melbourne, Australia*, pp. 176–181, 2012.

[67] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He, "Recognizing falls from silhouettes," *28th Annual International Conference of the* IEEE *Engineering in Medicine and Biology Society* (EMBS)*, New York,* NY, pp. 6388–6391, 2006.

[68] N. Thome, S. Miguet, and S. Ambellouis, "A real-time, multiview fall detection system: A LHMM-based approach," IEEE *Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1522–1532, 2008.

[69] Z. Htike, S. Egerton, and K. Chow, "A monocular view-invariant fall detection system for the elderly in assisted home environments," *7th International Conference on Intelligent Environments* (IE)*, Nottingham,* UK, pp. 40–46, 2011.

[70] M. Belshaw, B. Taati, J. Snoek, and A. Mihailidis, "Towards a single sensor passive solution for automated fall detection," *33rd Annual International Conference of the* IEEE EMBS*, Boston, Massachusetts* USA, pp. 1773–1776, 2011.

[71] H. Foroughi, A. Rezvanian, and A. Paziraee, "Robust fall detection using human shape and multi-class support vector machine," *Sixth Indian Conference on Computer Vision, Graphics Image Processing* (ICVGIP)*, Bhubaneswar, India*, pp. 413–420, 2008.

[72] S.-C. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *Visual Communications and Image Processing*, vol. 5308, no. 1, pp. 881–892, 2004.

[73] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using code-book model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[74] O. Barnich and M. Van Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011.

[75] R. Gonzalez, "Digital image processing," *Third Edition, Pearson Education*, 2008.

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

[77] N. McFarlane and C. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, vol. 8, pp. 187–193, 1995.

[78] S.-C. Cheung and C. Kamath, "Robust background subtraction with foreground validation for urban traffic video," *Visual Communications and Image Processing*, vol. 5308, pp. 2330–2340, 2005.

[79] Image Processing Toolbox 7.0, "Perform image processing analysis and algorithm development," *Mathworks Evaluation of Background Subtraction Algorithms with Post-ProcessingInc.*, 2010.

[80] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1337–1442, 2003.

[81] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," *the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence* (UAI), pp. 175–181, 1997.

[82] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *International Conference Computer Vision and Pattern Recognition, Fort Collins,* CO*,* USA, vol. 2, 1999.

[83] N. Bergman, "Recursive bayesian estimation navigation and tracking applications," *PhD thesis, Linköping Studies in Science and Technology Dissertations No. 579, Linköping University*, 1999.

[84] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 232–237, 1998.

[85] L. Mihaylova, P. Brasnett, N. Canagarajah, and D. Bull, "Object tracking by particle filtering techniques in video sequences," *Advances and Challenges in Multisensor Data and Information, NATO Security Through Science, Series 8, ISO Press*, no. 2, pp. 260–268, 2007.

[86] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[87] N. Bouaynaya, W. Qu, and D. Schonfeld, "An online motion-based particle filter for head tracking applications," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, pp. 225–228, March 2005.

[88] S. A. B. Ristic and N. Gordon, "Beyond the kalman filter: particle filters for tracking applications," *In Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, 2001.

[89] D. Parks and S. Fels, "Evaluation of background subtraction algorithms with post-processing," IEEE *Computer Society Fifth International Conference on Advanced Video and Signal Based Surveillance* AVSS*, Santa* Fe*,* NM, pp. 192–199, 2008.

[90] B. P. L. Lo and S. Velastin, "Automatic congestion detection system for underground platforms," *International symposium on intelligent multimedia, video and speech processing, Hong Kong*, pp. 158–161, 2001.

[91] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *International Conference Computer Vision and Pattern Recognition, Fort Collins, CO, USA*, pp. 246–252, 1999.

[92] C. Manning, P. Raghavan, and H. Schutze, *An introduction to information retrieval*. Cambridge University Press, second ed., 2008.

[93] Y. Abdelaziz and H. Karara, "Direct linear transformation into object space coordinates in close-range photogrammetry," *American Society of Photogrammetry, In Proceedings of the Symposium on Close-Range photogrammetry, Urbana, Illinois, United states*, vol. 1, pp. 1–18, 1971.

[94] S. Shih, Y. Hung, and S. Lin, "Accurate linear technique for camera calibration considering lens distortion by solving an eigenvalue problem," *Optical Engineering*, vol. 32, no. 1, pp. 138–149, 1993.

[95] Z. Zhang, "A flexible new technique for camera calibration," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[96] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off the shelf TV cameras and lenses," IEEE *Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.

[97] R. Burns, *Manual of photogrammetry.* American Society of Photogrammetry, fourth ed., 1980.

[98] K. Madsen, H. Nielsen, and O. Tingleff, "Methods for non-linear least squares problems," *Lecture note, Informatics and Mathematical Modelling, Technical University of Denmark,* DTU, 2004.

[99] D. Anderson, R. Luke, J. Keller, M. Skubic, M. Rantz, and M. Aud, "Modeling human activity from voxel person using fuzzy logic," IEEE *Transaction on Fuzzy Systems*, vol. 17, no. 1, pp. 39–49, 2009.

[100] G. Strang, *Introduction to Linear Algebra.* Wellesley-Cambridge Press, fourth ed., 2009.

[101] J. Magnus and H. Neudecker, *Matrix differential calculus with applications in statistics and econometrics.* New York, NY, USA: John Wiley and Sons, 1988.

[102] D. Tax, "The manual of data description toolbox for MATLAB 1.9.1," *Version 2.0.1*, 2012.

[103] S. Huwer and H. Niemann, "2D-object tracking based on

projection-histograms," *The 5th European Conference on Computer Vision*, vol. 1406, pp. 861–876, 1998.

[104] V. Vapnik, "Statistical learning theory," *Springer*, 1998.

[105] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, California*, pp. 232–237, 1998.

[106] J. Friedman, "Another approach to polychotomous classification," *Technical Report, Stanford Department of Statistics*, 1996.

[107] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," *Advances in Neural Information Processing Systems,* MIT *(Massachusetts Institute of Technology), Press*, vol. 12, pp. 547–553, 2000.

[108] J. Sun, C. Zheng, X. Li, and Y. Zhou, "Analysis of the distance between two classes for tuning SVM hyperparameters," IEEE *Transactions on Neural Networks*, vol. 21, no. 2, pp. 305–318, 2010.

[109] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[110] F. Heijden, R. Duin, D. Ridder, and D. Tax, "Classification, parameter estimation and state estimation: An engineering approach using MATLAB," *John Wiley and Sons*, 2005.

[111] Y. Li, K. Ho, and M. Popescu, "A microphone array system for au-

tomatic fall detection," IEEE *Transactions on Biomedical Engineering*, vol. 59, no. 2, pp. 1291–1301, 2012.

[112] Y. Zigel, D. Litvak, and I. Gannot, "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*, vol. 56, no. 12, pp. 2858–2867, 2009.

[113] H. Bhaskar, L. Mihaylova, and A. Achim, "Video foreground detection based on symmetric alpha-stable mixture models," IEEE *Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 8, pp. 1133–1138, 2010.

[114] Y. Chen, L. Zhu, A. Yuille, and H. Zhang, "Unsupervised learning of probabilistic object models (POMs) for object classification, segmentation and recognition," *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR)*, Anchorage,* AK, pp. 1–8, 2008.

[115] F. Lecumberry, A. Pardo, and G. Sapiro, "Simultaneous object classification and segmentation with high-order multiple shape models," IEEE *Transactions on Image Processing*, vol. 19, no. 3, pp. 625–635, 2010.

[116] W. Wang, D. Cosker, Y. Hicks, S. Sanei, and J. Chambers, "Video assisted speech source separation," *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '05), Philadelphia, USA*, vol. 5, pp. 425–428, 2005.

[117] M. Naqvi, M. Yu, and J. Chambers, "A multimodal approach to blind source separation of moving sources," IEEE *Journal of Selected Topics in Signal Processing*, vol. 4, no. 5, pp. 895–910, 2010.

[118] Z. Zhang, W. Liu, V. Metsis, and V. Athitsos, "A viewpoint-independent statistical method for fall detection," *21st International Conference on Pattern Recognition* (ICPR), *Tsukuba Science City, Japan*, 2012.