2010

Occlusion handling in multiple people tracking

Mohammad Omair Alam
University of Windsor

OCCLUSION HANDLING IN MULTIPLE PEOPLE TRACKING

by

Mohammad Omair Alam

A Thesis
Submitted to the Faculty of Graduate Studies
through Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

2010

Your file Votre référence
ISBN: 978-0-494-70585-8
Our file Notre référence
ISBN: 978-0-494-70585-8

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Object tracking with occlusion handling is a challenging problem in automated video surveillance. Occlusion handling and tracking have always been considered as separate modules. We have proposed an automated video surveillance system, which automatically detects occlusions and perform occlusion handling, while the tracker continues to track resulting separated objects. A new approach based on sub-blobbing is presented for tracking objects accurately and steadily, when the target encounters occlusion in video sequences. We have used a feature-based framework for tracking, which involves feature extraction and feature matching.

DEDICATION

To my Parents

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank and express my sincere gratitude to my supervisor Dr. Boubakeur Boufama, for his continuous support, encouragement, and stimulating suggestions throughout my Master's degree.

I would also like to thank Dr. Imran Ahmad and Dr. Ben Chaouch for being in the committee and providing me with valuable feedback.

Finally, I am grateful to all my colleagues for their moral support and valuable suggestions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The field of computer vision is concerned with problems that involve interfacing computers with their surrounding environment through visual means. One such problem, video surveillance, has an objective to monitor a given environment and report the information about the observed activity that is of significant interest. Video surveillance systems have long been in use to monitor security sensitive areas. The application of video cameras for monitoring and surveillance is common in both public places like banks, shopping centers, airports, metro stations and private firms.

1.1 Automated Video Surveillance

Video surveillance can be described as the task of analyzing video sequences to detect abnormal or unusual activities. An intelligent video surveillance is a system where there is no human intervention. The system does both the low-level tasks, like motion detection and tracking, and also high -level decision making tasks like abnormal event detection.

Moving object detection is the basic step for video tracking. It handles segmentation of moving objects from stationary background objects. Commonly used techniques for object detection are background subtraction, statistical methods, temporal differencing and optical flow.

In recent years, detection and tracking of human beings using computer vision techniques have been receiving a great deal of interest. Video surveillance systems from the security point of view are increasingly becoming the most widespread tools for monitoring, management, and law enforcement in public areas. Nowadays, even individual houses are equipped with personalized security cameras to monitor their houses. For instance in the UK, which is a world leader in video and digital surveillance, there are 4.5 million CCTV cameras in use [Liberty], a significant number for an estimated population of 61.8 million[Statistics]. This means there is one CCTV for an average of every 15 people. Needless to say, human operators cannot be employed to monitor each and every camera. These surveillance cameras, generally, are used to gather evidence after an incident has taken place. Therefore, in many cases, major mishaps such as a crime or suicide attempt, which could have been averted, are known after they have taken place. Therefore, the automation of visual surveillance systems becomes a very important research area.

There are many commercial visual surveillance systems available in the market but these are manual and continuously need human operators to monitor the video feed [Shah07]. Therefore, to overcome this limitation, a lot of research is being done to develop systems which can automatically monitor people, vehicles, etc.

1.2 Motivation

Human tracking in the field of video surveillance has always been a challenging scientific problem and a very active research area. Several researchers, institutions and commercial companies are continuously trying to build many promising applications [Wang03]. Significant progress has been made in human tracking over the last few years. However the problem of how to handle the trade-off between tracking precision and its computational cost still remains. Our motivation in studying this problem is to develop an intelligent video surveillance system capable of real-time detection of moving objects and tracking in the case of occlusions. Solving the problem of tracking individuals or members of identified groups during encounters has been very sparsely addressed and is usually avoided completely by merely creating a new multi-object individual [Mabey03]. We would like to deal with the explicit occlusion reasoning under the multiple target tracking scenario, where the occlusion relationships (i.e. who is occluding whom) between the tracked targets will be explicitly characterized.

1.3 Problem Statement

This thesis investigates the problem of tracking humans during occlusion. The aim in this work is to improve occlusion performance and recapture rate with a single camera. Humans do this exceptionally well because of high resolution attentive vision and a well calibrated low-level vision system to identify and separate objects, even when some are partially or fully occluded. Humans also recapture a tracked object after occlusion with high accuracy. It involves the following sub-tasks.

Segmentation: This is the first step of almost every tracking system. It detects the moving objects in the image. The output of segmentation is a binary image which differentiates foreground from background. This output is given as input to the next step, namely, tracking.

Tracking: Tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene [Yilmaz06]. Our goal is to track the humans between frames along with their features such as location, bounding box, color histogram, etc.

Occlusion Handling: In multi-object tracking, occlusions are bound to occur, when the objects interact or cross each other. In this thesis, we present a novel method for tracking moving objects in an accurate manner during occlusions.

1.4 Challenges

In many instances, humans can appear the same. For example, if two persons are of the same height and build, and their clothes are of similar color too, it is difficult to distinguish the two even though they can be detected as two separate objects. It is necessary to tell persons apart from each other so that correspondence between detected objects in two different frames may be established.

1.5 Overview

The organization of the thesis is as follows. Chapter 2 presents a background on moving object detection, tracking and occlusion handling. In particular, the challenges of tracking under occlusions are discussed in this chapter with examples. Chapter 3 presents an overview of existing methods. Chapter 4 describes a robust framework for tracking multiple people under occlusions. Experimental results of the proposed system are presented in Chapter 5. Finally, Chapter 6 concludes the thesis with the suggestions for future research.

CHAPTER II

BACKGROUND OF VIDEO SURVEILLANCE

In the last few years, visual surveillance has become one of the most active research areas in computer vision, especially due to growing demands for security [Abdelkader06]. Detection, localization and tracking of moving objects such as people or vehicles are key components for many applications including surveillance, monitoring and elderly care [Wang06].

Analysis of human movement is currently one of the most active research topics in the field of video surveillance [Murali09]. Human Motion Analysis includes detection, tracking and recognition of people. Human Motion Analysis can be classified into 3 categories [Thomas06, Hu04], low level (Detection), intermediate level (Tracking) and high level (Behavioural Analysis). However, most of the current video surveillance systems suffer from a major obstacle; the need of a human operator to be available for constant monitoring [Ko08]. This also suggests that the effectiveness of these systems is largely dependent on the level of human involvement, and not the technological capabilities. To overcome this limitation, a major effort is underway to develop automated system for real-time monitoring of objects.

An intelligent video surveillance system is expected to detect people and monitor their actions and subsequently need to analyse their behaviour in order to prevent any untoward incident. The problem of classifying and understanding human behaviour is a difficult task to perform in practice, and we concentrate on tracking robustly during occlusions.

2.1 Smart Video Surveillance

The term "Intelligent Video Surveillance" or "Automatic Video Surveillance" is used interchangeably with smart video surveillance. The ultimate goal of a smart video surveillance system is to detect, recognize and track objects from video sequences as automatically as possible. Generally, an automated video surveillance system would contain four main modules (Figure 1). Firstly in the object detection module, the moving objects in the video are detected. Secondly in the object classification module, detected objects in the scene are classified such as humans or vehicles for example. Thirdly, in the object tracking module, the detected objects are tracked as long as they are in the field of view. Fourthly, in the object behavioural analysis module, the behaviour of each tracked object is monitored and analysed, and then acted upon if necessary.

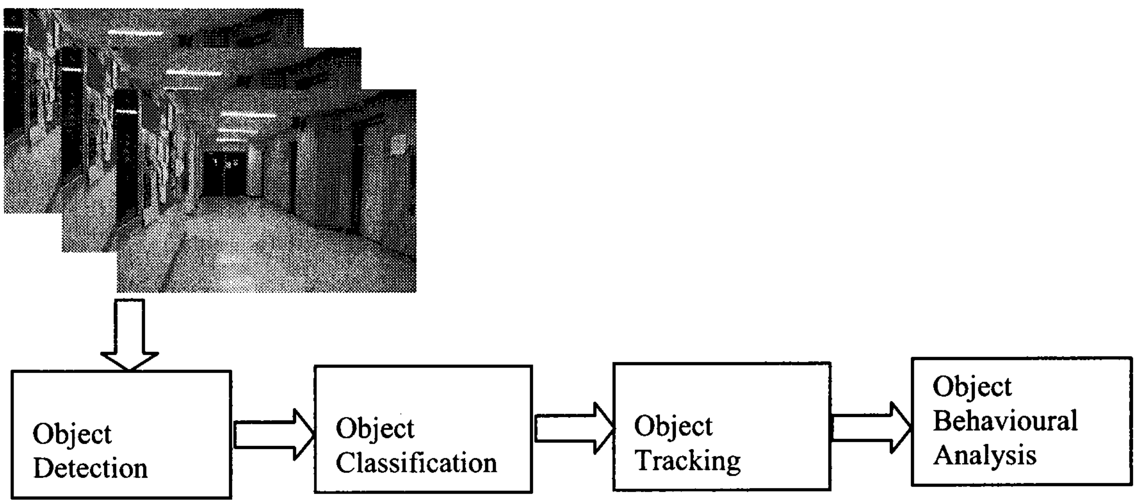


Figure 2.1: Smart Video Surveillance Modules

2.1.1 Smart Video Surveillance in real-time applications

Listed below are some of the scenarios, in which smart video surveillance systems are useful.

- Vigilance on national and international borders

- Monitoring of banks, schools, airports, museums, stations, etc
- Monitoring elderly and sick people for activities requiring help
- Measuring traffic flow and pedestrian congestion
- Monitoring speeding vehicles, and detecting accidents
- Preventing theft in homes and parking lots

2.2 Object Detection

The detection of moving objects is a common feature of every video surveillance system. Detecting regions that correspond to moving objects such as people and vehicles is the first basic step of almost every video surveillance system. Indeed most of the behaviour we wish to track, is associated with some kind of body movement. This process aims at segmenting regions of an image corresponding to moving objects over subsequent frames. Since we are separating the foreground objects from the background, this process is also known as foreground object segmentation. Subsequent processes such as object classification, object tracking, and behavioural analysis depends on this process. Techniques that are frequently used to identify foreground objects include background subtraction, temporal differencing and optical flow.

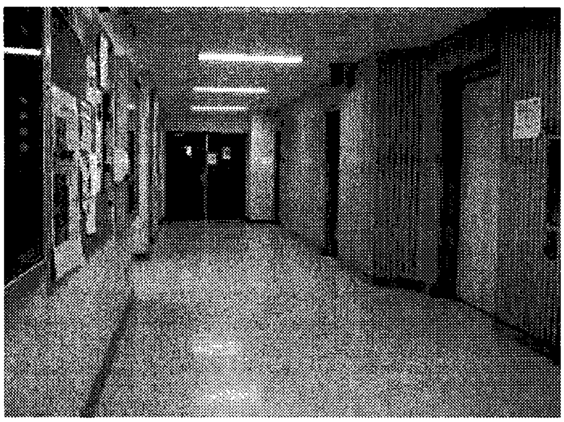
2.2.1 Background Subtraction

Background subtraction simply subtracts a previously acquired reference background image from the currently observed image pixel by pixel. Short for *Picture Element*, a pixel is a single point in a graphic image [Webopedia]. The reference background image is known as background model, since it has been modelled from a set of frames, and is updated regularly. The pixels where the difference in color intensity is above a certain threshold are classified as foreground pixels. Due to this method's simple nature it is extremely sensitive to sudden illumination changes. Therefore, the reference background is updated with new images over time to adapt to dynamic scene changes.

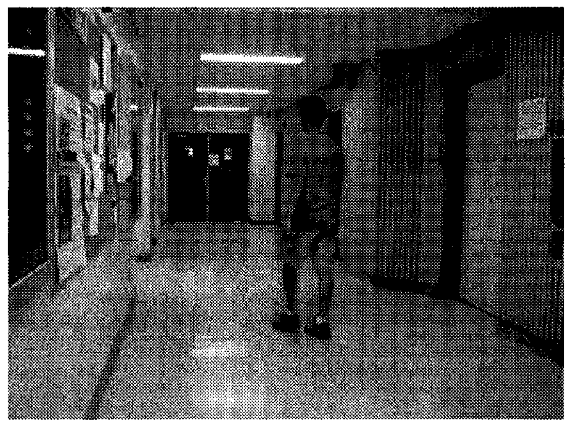
Mathematically, the color of a pixel at location (x,y) in the current image I_t is marked as foreground [Heikkila99] if

$$|I_t(x, y) - B_t(x, y)| > T$$

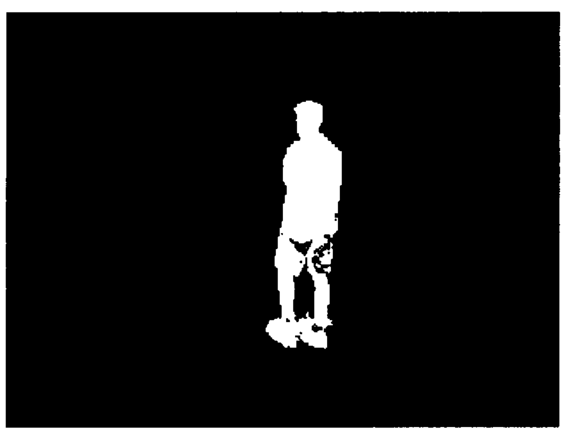
is satisfied where T is a predefined threshold, and B_t is the background image. Pixels that are classified as foreground pixels are represented in white color in figure 2.2(c), whereas the background pixels are shown in black color.



(a) Background model



(b) Random frame from video



(c) Result of background subtraction

Figure 2.2: Background Subtraction

2.2.2 Temporal Differencing

Temporal differencing detects foreground objects by making use of difference in pixel values over consecutive frames (two or three). This approach is possibly the simplest one and highly adaptive to dynamic scene changes, however, it generally fails in detecting whole relevant pixels of some types of moving objects. Also this method fails to detect stopped objects in the scene. Hence the detection performance of temporal differencing is usually quite poor in real-life surveillance systems.

Lipton et al. presented a two-frame differencing scheme where the pixels that satisfy the following equation are marked as foreground. [Lipton98].

$$|I_t\,(x, y) - I_{t-1}\,(x, y)| > T$$

where, $I_t\ (x,\ y)$ is the intensity of pixel at location (x, y) in the current frame captured at time t, $I_{t-1}\ (x,\ y)$ is the intensity of pixel at location (x, y) in the previous frame, and T is the threshold.

2.2.3 Optical Flow

Optical flow methods make use of flow vectors of moving objects over time to detect the moving objects in the scene. However, most of the optical flow methods are computationally complex and cannot be used for real-time tracking, unless with the help of specialized hardware. [Candamo10].

2.2.4 Challenges in moving object detection

Illumination changes: The change in illumination with the time of day for outdoor systems, and lighting variations for indoor systems can result in the detection of false moving objects.

Shadow: Shadows are a major problem in every tracking system. Objects cast shadows that can also be classified as foreground objects.

Camouflage: Similarity in the features of foreground objects and the background. When a moving object has similar color and intensity as background region, then it is difficult to distinguish between them.

2.3 Object Classification

After finding the moving objects in the scene, the next step would be to classify them (i.e. human or car). For example, there would be no difference between a vehicle violating a red light traffic signal and a pedestrian crossing the street, if they have not been classified. Object classification is not discussed in this thesis, since we would be dealing with only human tracking. Object classification methods in the field of video surveillance can be classified into three categories [Candamo10]: shape-based, motion-based and feature-based classification methods.

1. *Shape-Based Classification:* The shape or the geometry (i.e. blobs, silhouette, box, etc.) of the moving objects are used to classify the objects.
2. *Motion-Based Classification:* Object motion characteristics are used to distinguish between objects. Motion can be used for human recognition as well as identification of the type of human movement, such as walking or running. In general, human motion exhibits a periodic property, so this has been used as a strong cue for moving object classification.
3. *Feature-Based Classification:* Important features such as the color of the dress are used for classification purpose.

2.4 Object Tracking

Object tracking can be defined as finding the appearance and location of an object in a sequence of frames. It is described as a correspondence problem and involves finding which object in a video frame is related to which object in the next frame [Javed02]. If the object is continuously observable and its shape, size or motion does not vary over time, then tracking isn't difficult. However, in real-life environments, some objects such as people, do not walk at a constant motion. Similarly shapes are also subject to change. The biggest challenge in any video surveillance system are occlusions, however, a robust tracking system can overcome this difficulty.

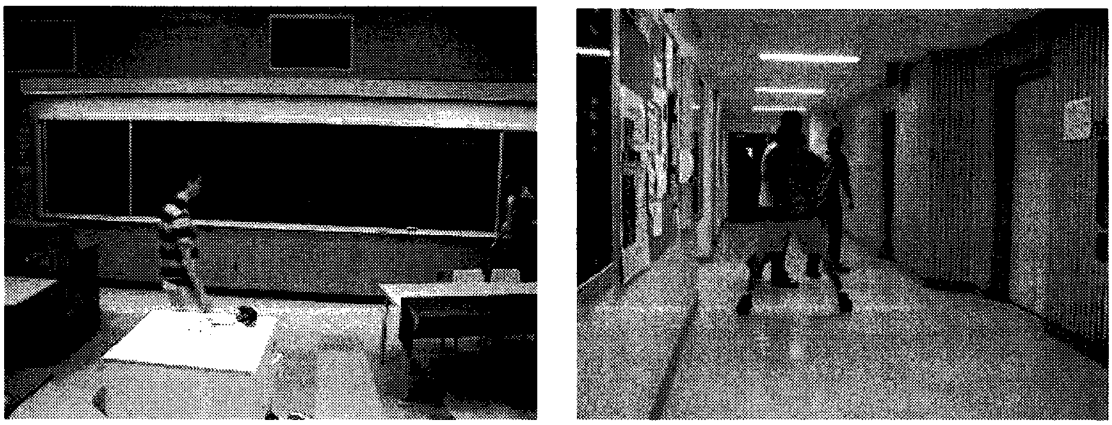
2.5 Handling Occlusions

In an environment where multiple objects are interacting, objects may undergo occlusions, i.e. the view of one object is blocked by another object or structure. Handling occlusions has become an important issue for practical tracking algorithms. Occlusions affect the global shape by overlapping partially or completely the tracked object and therefore subsequent recognition suffers. When tracking human interactions in the presence of occlusions, a remaining issue is to differentiate between objects when they reappear as individual blobs. Occlusion handling can be broadly classified into three categories: self occlusion, inter-object occlusion and occlusion by the background scene [Yilmaz06].

Self occlusion occurs mostly when one part of the object occludes another. For example, arms occluded by the chest. Self occlusion will always be a problem in a silhouette-based posture recognition framework.

Inter-object occlusion occurs when two objects being tracked occlude each other. These scenarios occur when two or more moving people cross each other, or they are walking together with one of them obstructing the others from the camera's field of view.

Occlusion by the background occurs when some structure in the background occludes the tracked object. For example, when a person walks behind a tree which is the part of background, the tree would occlude the person.



(a) Occlusion by background (b) A person occluding two persons

Figure 2.3: Types of Occlusion

2.6 Conclusion

In this chapter, we have introduced the key logical components of a general automated surveillance system, noting important technical challenges in the real-world deployment of such a system. Tracking non-rigid objects and classifying their motion is a challenging problem. Many of the key obstacles such as occlusions are not solved yet. Occlusions lead to discontinuity in the observation of objects, and its one of the major issues that a tracking algorithm should solve.

CHAPTER III

REVIEW OF LITERATURE

3.1 Introduction

Tracking multiple objects plays an important role in many applications such as intelligent human computer interaction and video surveillance. It is the process of following an object of interest within a sequence of frames, from its first appearance to its disappearance. During the time that the tracked object is present in the scene, it may be occluded (either partially or fully) by other objects within the scene. A tracking system should be able to predict the position of any occluded objects through the occlusion, ensuring that the object is not temporarily lost and only detected again when the object appears after the occlusion. Thus occlusion becomes a special and difficult problem for multiple object tracking. This thesis addresses the problem of detecting and tracking multiple moving humans in a complex environment with unknown background, especially in the case of occlusions. The problem of understanding and classifying human behaviour is not addressed in this thesis.

Nearly every system in a video surveillance system starts with segmentation [Hu04]. The current motion segmentation methods are mainly based on background subtraction, temporal differencing or optical flow [Thomas06]. Background subtraction is a commonly used technique to detect moving regions in an image [Shireen08]. Development of a reliable background model adaptive to dynamic changes in complex environments is still a challenge [Shireen08, Hu04].

There are two different kinds of methodologies used in visual tracking algorithms, bottom-up and top-down. [Ying01] Bottom-up approaches respond to stimulus, i.e. they construct object states by analyzing the content of the images. On the other hand, top-down approaches are goal oriented, and a majority of video surveillance systems are designed in this manner. This methodology adopts the idea of analysis-by-synthesis, by directly verifying a number of hypotheses. [Ying01, MacCormick99].

A top-down approach is the most popular method for developing real time tracking systems. Systems following top-down approach basically follow three steps as shown in Figure 3.1.

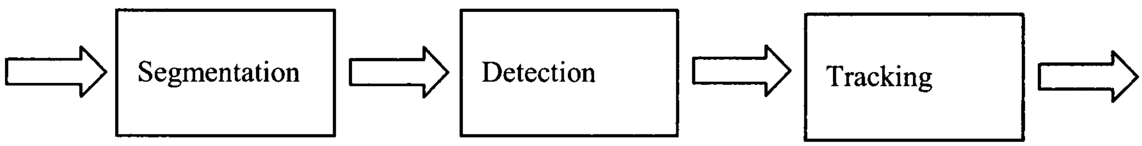


Figure 3.1: A basic top-down approach

Recently, systems using particle filtering techniques [Wu08] have begun to use both top-down and bottom-up approaches within one system. Using the stimulus from the input images, the particle filters are used to track and update previously detected objects. New objects are detected and added using a top-down approach.

Tracking methods can be divided into four major categories [Ko08]

1. *Region-based Tracking:* Region based tracking algorithms track objects according to the variations of the image regions, corresponding to the moving objects. They do not work well against a cluttered background or with multiple objects [Hu04]. Moreover it cannot reliably handle occlusions.
2. *Contour-based Tracking:* In this method, bounding contours are used to represent the object's outline, which are updated dynamically in successive frames [Hu04]. These

methods, instead of tracking the whole set of pixels comprising an object, track only the contour of the object. These methods perform well under partial occlusions. However, the recovery of a 3-D object from its contour comes with a high computational cost.

3. *Model-based Tracking:* This approach tracks objects by building a projected object model, and matching the objects to it. Construction of 2D or 3D human body models is the base of model-based human tracking. In general, the more complex a human body model, the more accurate the tracking results and better performance during occlusions, but the more expensive the computational cost.
4. *Feature-based Tracking:* Feature based methods perform tracking by extracting features of the moving object, and matching the feature between frames. We used feature-based approach in our tracking system because it is faster compared to other methods, and it is easier to extract the features, and the characteristics of a feature normally persist through a sequence of frames.

For tracking systems to function robustly in all kinds of conditions is a big challenge. Environmental changes (indoor and outdoor), changing weather conditions (rain, fog) and illumination changes are significant hurdles for a tracking system. Other challenges include shadows, partial and full occlusion of objects with each other and with stationary objects in the scene. Some of these problems like occlusions can be overcome using multi-camera setups [Khan03]. However this introduces additional challenges such as camera calibration, utilizing the Pan-Tilt-Zoom (PTZ) cameras and handing over the tracking between views. However since we have used a single fixed camera for tracking, we do not have the benefit of multiple views, and thus our main focus is on how to handle occlusions.

3.2 Foreground Segmentation

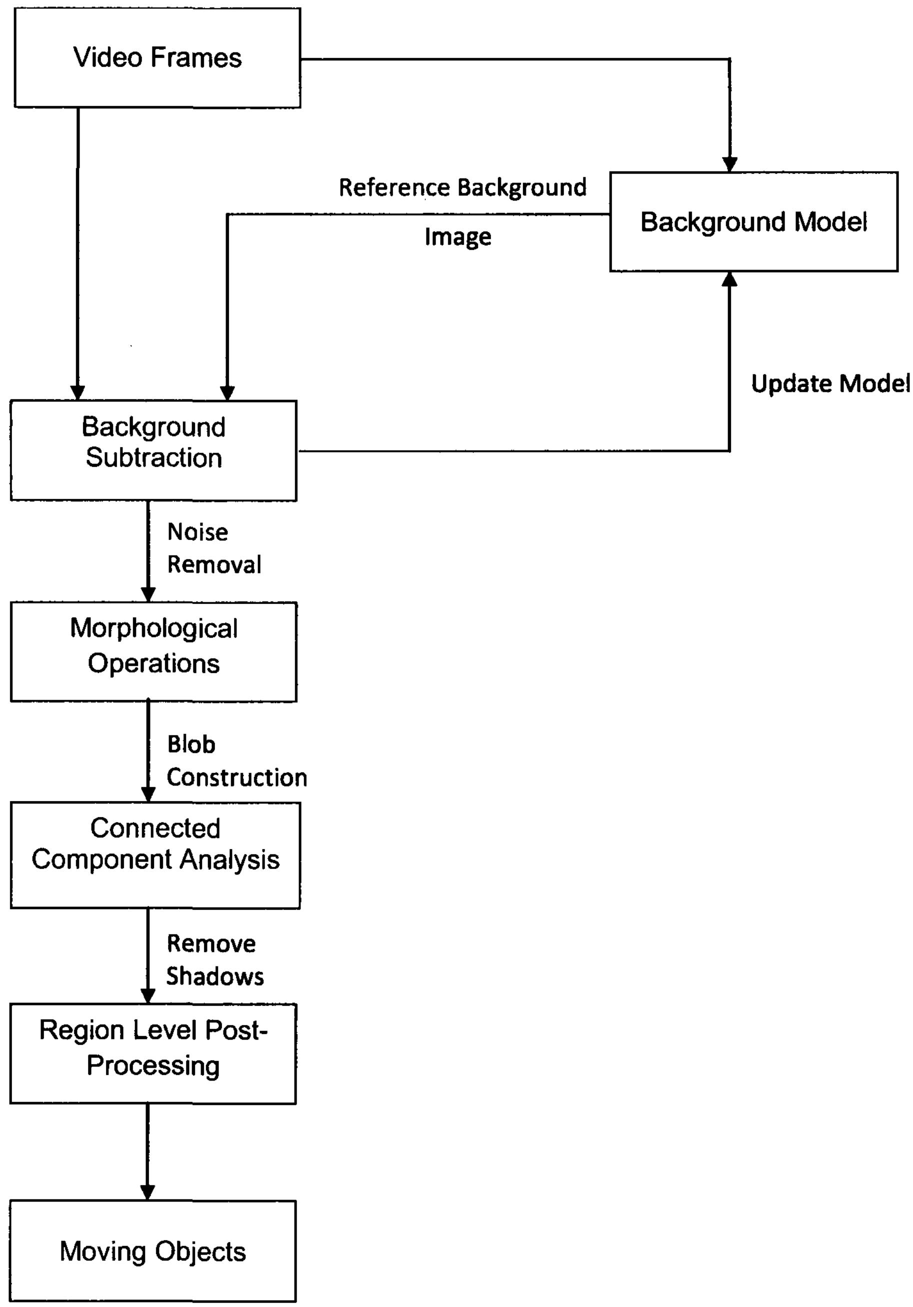


Figure 3.2 Foreground Object Detection Steps

The first step of every visual tracking system is to detect foreground objects. Proper segmentation creates a focus of attention for higher processing levels such as tracking, classification and behaviour understanding and reduces computation time considerably, since only pixels belonging to foreground objects need to be dealt with. In the literature, most of the works on foreground segmentation is based on background subtraction technique [Zhou01]. Background subtraction is the commonly used technique for motion segmentation in static areas [McIvor00]. A good overview of the existing background removal methods is given by Shireen *et al.* [Shireen08] and Yilmaz *et al.* [Yilmaz06]. Even though they produce good results, most of the suggested methods suffer from a drawback. The background model is usually maintained for every image pixel. Hence these methods lead to substantial computational cost, depending on how the pixel values are modelled. Another drawback is that the reliability of the methods strongly depends on the choice of some model parameters or some thresholds. Heikkila *et al.* used the simple version of background subtraction where a pixel is computed as foreground or background based on the absolute difference between each new frame $I(k)$ and an adaptive background frame $B(k)$ [Heikkila99]. To set an optimal value for the threshold is quite challenging. The pixels are classified as foreground if they exceed a particular threshold. The background is updated using the following first order recursive filter, where α is an adaption coefficient.

$$B(k+1) = (1-\alpha)B(k) + \alpha I(k)$$

There are several problems that must be addressed by a good background removal algorithm to correctly detect moving objects [Elgammal02]. A good background subtraction technique should be able to handle the relocation of background objects, nonstationary background objects, e.g. grass, waving branches of the trees, and image changes due to camera motion, which is common in outdoor application. A background removal system should also adapt to sudden illumination changes, whether gradual changes (time of day) or sudden changes (light switch) [Shireen08].

Frame differencing is the simplest moving object detection method, which works by determining the difference between the intensity of each pixel with its corresponding pixels in consecutive frames. This technique is sensitive to noise and variations in illumination, and does not consider local consistency properties of the change mask [Radke05]. This method also fails to segment the non-background objects if they stop moving [Huwer00]. Since it uses only a single previous frame, frame differencing may not be able to identify the interior pixels of a large, uniformly-coloured moving object. This is commonly known as aperture problem [Shireen08]. In order to overcome shortcomings of two frame differencing in some cases, three frame differencing have been proposed [Wang03]. However the disadvantage of temporal thresholding processes is that they leave gaps behind moving objects, therefore making them undesirable for use in applications that require accurate segmentation. To overcome this, methods that combine temporal thresholding and background subtraction have been proposed [Xiaofeng10, Abdelkader06]. The mean and variance of each pixel is calculated over several frames, and recursively updated for each new frame. A second set of mean and variance is calculated to model the background. The background model's update process is selective in that only pixels (according to variance) not belonging to foreground are incorporated. The confidence weight of the pixel belonging to the foreground is extracted from the background model, and multiplied with the variance obtained from the temporal thresholding. This value is then compared with the set threshold to see if the pixel is in motion. Another method uses a combination of spatial clustering and image fusion to fill the holes [Murali09, Girisha09]. The methods proposed however consider only a single mode of background, so only simple scenes can be effectively modelled.

Not many foreground object detection methods have been proposed using optical flow technique. [Barron92] gives a detailed description of optical flow. Optical flow is a vector-based approach that estimates motion in video by matching points on objects over multiple frames [Candamo10]. A moderately high frame rate is required for accurate measurements. It is robust to multiple and simultaneous camera and object motions, making it ideal for crowd analysis and conditions that contain dense motion. The disadvantage with optical flow is, when due to clutter or varying lightening conditions, if the objects are not defined properly, errors can occur in the optical flow output. Also this approach is too slow and computationally expensive for a real-time tracking system, as it requires specialized hardware.

Many background subtraction methods based on the AdaBoost algorithm have been proposed [Yeh09, Grabner06]. The systems modelled on this approach are able to detect changes in low contrast scenes; however, this also makes the approach susceptible to noise. Also this method is not suitable for real-time systems, because it is reported to run slowly

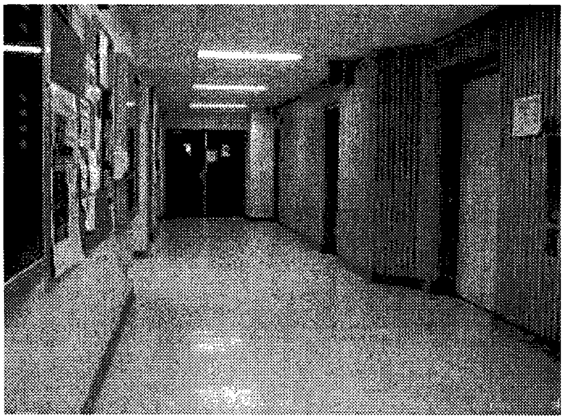
Background subtraction technique requires modeling the scene background. Background modelling, also referred to as background maintenance, is core of any background subtraction algorithm. Background modelling methods build a model of the background and compare this to each incoming frame. This approach suffers from illumination changing environment; in addition it is incapable of removing shadows of moving objects. Recently a mixture model that combines the Eigen-background with Gaussian models [Wu10] have been proposed that is able to cope better with real world problems such as lighting fluctuations and shadow removal. Although the terms background subtraction and background modelling are often used interchangeably, they are separate and distinct processes. Background modelling refers to the process of creating, and subsequently maintaining a model of the appearance of the background, whereas background subtraction refers to the process in which an image frame is compared to the background model in order to determine whether individual pixels are part of the background or the foreground [Shireen08].

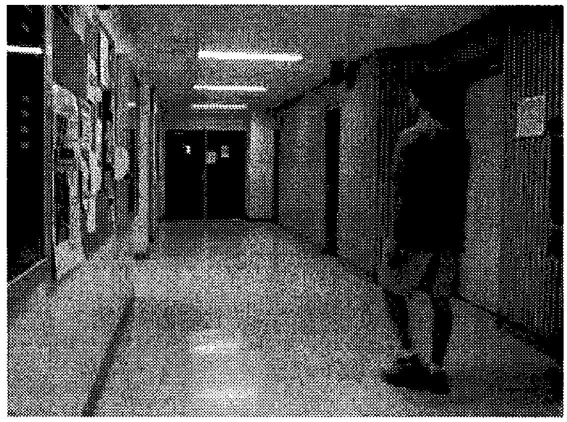
Stauffer et al. [Stauffer99] is the first paper to model pixel intensity as mixture of Gaussians. Unimodal representations which store single mean per pixel, are prone to false detections in the case of moving background. For computational reasons K Gaussians are employed for each pixel. Another background subtraction technique using Gaussian Mixture Models (GMM) which can handle large variations in the background intensity distribution is proposed by Bouttefroy [Bouttefroy10]. However, in GMM, finding a correct number of Gaussians is difficult. Another problem of this approach is the computational cost and therefore not feasible for real-time processing.

Edge segmentation had been one of the most popular segmentation algorithms for years. Edge-based segmentation techniques apply edge detection to extract discontinuities in the scene and segment the image [Hsiao05]. In order to segment the image correctly, the identified edges should form close boundaries. However, the resulting edge maps are often disconnected. Hence, additional processing should be performed on edge boundaries to connect isolated edges if they are within a distance-threshold of each other. Another main drawback of edge-based range segmentation algorithms is that discontinuities are smooth and hard to locate for curved surfaces in depth images, resulting in under-segmentation of range images. Thus, it is essential to inspect each of the edge-separated regions iteratively to assure that no object of interest is missed. As a result, edge detection segmentation methods are computationally intense and therefore have limited applications in real-time vision systems.

The outputs of foreground object segmentation algorithms generally contain noise and therefore not appropriate for further processing without special post-processing. Therefore, some well-known noise removal methods, known as morphological operators, are mostly used to remove the noise from the foreground mask. Morphology gets its name from the study of shapes, and it employs many techniques for processing images based on mathematical set operators like intersection, union, complement, etc. Thus, the noise in the image is filtered by morphological operations such as erosion and dilation [Xiaofeng10, Li09]. Finally when the area of the connected regions is larger than a given threshold, the regions are labelled as foreground objects. A dilation operation enlarges a region, while erosion makes it smaller.

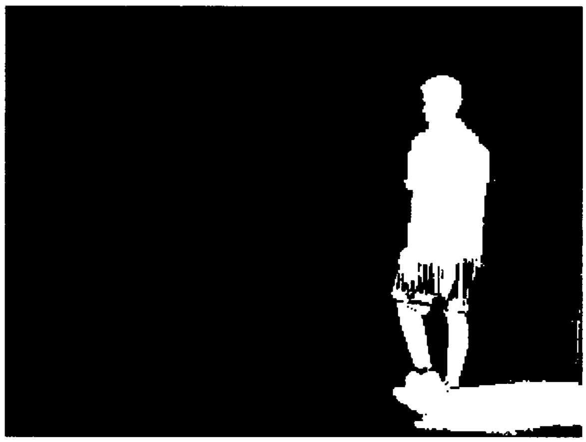
Shadows are a big problem in most of the foreground detection algorithms, which can cause inaccurate foreground object segmentation. Additionally, large shadows cause occlusions in the foreground regions. In the literature, different techniques have been used to remove shadows from the foreground region.





(a) Background model

(b) Random frame from video



(c) Background subtraction result

Figure 3.3: Example of shadows during background subtraction

Shor *et al.* proposed a method which requires the user to only provide a seed for the shadow by a single mouse click in the interior of the shadow [Shor08]. The method applies an iterative procedure to reconstruct the shadow patch by patch originating at the seed. The authors considered a non-uniform shadow field in the shadow region and designed a pyramid-based approach to recover it. Although this method requires minimal user assistance, it works only when the shadow region can be segmented from the background. Another method proposed a shadow editing tool that can perform a variety of shadow manipulations [Mohan07]. However, it requires the specification of boundary points, edges, sharpness and amplitude, which can be a tedious task for the user.

Edges have also been used in shadow detection. Zhang *et al.* proposed a method for detecting shadows based on the fact that the ratio edge is illumination invariant. The ratio edge represents the ratio between neighbouring pixels [Zhang06]. The location of this edge can be analysed to segment shadows from foreground objects. Also, Xu *et al.* proposed to use a canny edge detector on the foreground image to separate the shadow and the non-shadow regions [Xu04]. The shadow regions can be identified and removed through multi-frame integration, region growing and edge matching. Wang *et al.* proposed a method which analysed each shadow as a series of sub-regions [Wang06]. But these shadow detection methods are multi-step and quite complex processes and are not ideal for use in a real-time tracking system.

After the foreground pixels have been identified, a labelling scheme based on pixel connectivity forms these pixels into a blob. [Wu10, Shireen08, Zhou01].

3.3 Object Tracking

Object tracking is one of the most important parts of a video surveillance system. After foreground object segmentation has been achieved, then comes the task of establishing a match between object masks in consecutive frames. In order to track an object accurately, features that can uniquely identify the object in all the frames must be extracted. Feature selection is closely related to the object representation [Yilmaz06], such as object edges are usually used as features for contour-based representation, and color is mostly used for histogram-based representation. Feature-based tracking algorithms model objects using a set of extracted features like size, shape, color histogram, etc. Tracking is performed by matching the object between frames. Based on the information available, local features like points, edges, corner vertices etc. and global features like color [Lu01], texture, etc. can be extracted. Selecting good features that can be used for future tracking or identification is a necessity, since the object's appearance may change in a latter frame, due to illumination, orientation, scale, or other natural changes.

Color is the most commonly used feature for object tracking in the context of video surveillance. Cheung et al. claim that color is better than luminance at identifying objects in low-contrast areas and suppressing shadow cast by moving objects [Cheung04]. Color feature approaches have focused on using histograms, as they are simple to compute and compare. The main benefit of color histogram is that it is not affected by pose change or motion, and hence is a reliable feature for matching [Boufama07, Lu01]. There are also other potential features like color correlograms [Park07], color moments [Duanmu10], MPEG-7 descriptors [Manjunath01], etc. Every pixel in an image can be represented as a point in a three dimensional color space. There are many color spaces available, with each of them being appropriate for a specific application, but the RGB and HSV color spaces (see Figure 3.4 [Cardani01]) are the most commonly used for object tracking, and can be easily converted from one to another. Overall, there is no such proof as to which color space is better than another one for every application, so a variety of color spaces have been used [Yilmaz06].

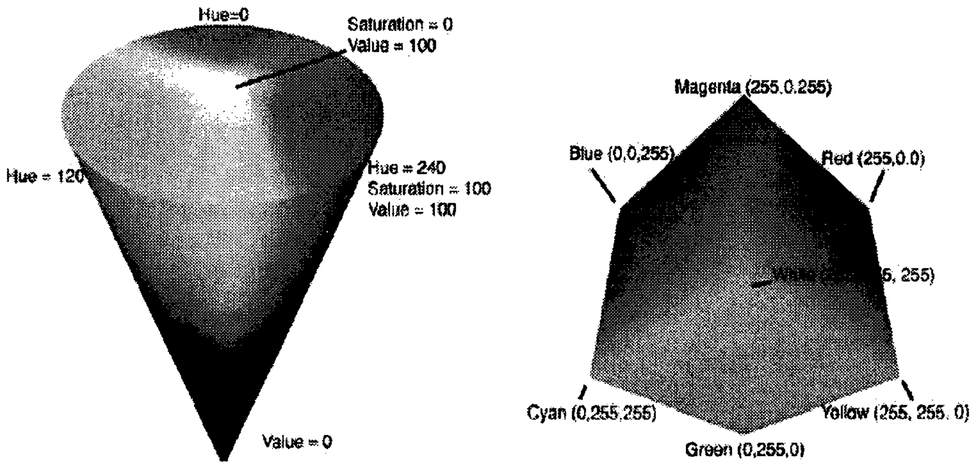


Figure 3.4 HSV and RGB color spaces

Elgammal *et al.* uses the chromaticity coordinates as r = R/s, g = G/s and b = B/s, where s = (R+G+B) and $r+g+b=1$ [Elgammal00]. This has the advantage of being more insensitive to small changes in illumination that arises due to shadows. However, they have the disadvantage of losing illumination information which is related to the differences in whiteness, blackness, and greyness between different objects.

The basic mean-shift algorithm [Comaniciu03] which is based on a color model has proven to be quite simple and robust in the field of video tracking. However the performance of the results diminishes, when an object and its background have similar colors. The color of the object depends on illumination, viewpoint, camera parameters, and so on which tend to change during a long tracking process.

Another feature useful for object tracking is edge. An edge is a set of connected pixels, which defines the boundary of a surface. One of the common edge descriptor is edge orientation histogram. Changjiang *et al.* proposed an approach to extract the edges using Sobel operator, and then the edge orientation histogram is used as one of the descriptors to represent the object [Changjiang05]. To build the edge orientation histogram, after converting the color image to greyscale, the edges are extracted using horizontal and vertical Sobel operators. Then the strength and orientation of edges are calculated as

$$\begin{aligned} S(x,y) &= \sqrt{G_x^2(x,y) + G_y^2(x,y)}, \\ \theta &= \arctan\left(G_y(x,y)/G_x(x,y)\right) \end{aligned}$$

Where G_x is the horizontal gradient and G_y is the vertical gradient. Finally the edges are counted into K bins based on their strengths and orientation. Edge orientation histogram feature is not invariant to rotation and scale, and therefore it is not a very reliable feature for many object tracking algorithms.

One more blob feature which have been used in some of the proposed algorithms is compactness [Beynon03], which can be defined as

$$C = \frac{area}{perimeter}$$

This feature represents how "stretched out" a shape is. The perimeter is the number of pixels that are part of an object and have at least one 4-connected neighbour that is not in the object. For example, circle has the minimum perimeter for a given area, hence exhibits high compactness. For people tracking, this was found to be not a very stable feature, since humans do not always walk in the same gait pattern.

Polana *et al.* proposed a method where a person is bounded with a rectangular box, and the centroid of the box is selected as a feature for tracking [Polana94]. Even when occlusions happen between two persons during tracking, as long as the velocity of the centroids can be distinguished effectively, tracking is done successfully.

ZinBer et al. proposed a robust feature tracking method [Zinber04] based on the well known Tomasi-Kanade tracker [Lucas81], but the tracking result becomes unreliable, as the target appearance undergoes major changes. Czyzewski *et al.* used Kalman filter with RGB color-based approach for matching moving objects between the frames [Czyzewski08]. The similarity between the foreground objects is measured using a threshold, which fails in fully occluded scenarios.

Bird et al. preferred blob-based tracking over kernel-based method, and used the shirt color as the main feature for tracking purpose [Bird05]. A blob is a group of connected regions, according to spatial constraints. As such, a detected blob may be formed by a single, large, connected (either 8 or 4 connected) region, or a cluster of several smaller connected components located close to each other. Blob-based tracking offers a computational advantage kernel, since the latter has to be first initialized, which would redundantly require blob extraction to be performed. Blob-based tracking methods are extremely popular in the literature; for example Boufama *et al.* performs blob-based color histogram tracking [Boufama07].

3.4 Occlusion Handling

Objects in the real world exhibit complex interactions [Senior06]. When these interactions prevent a tracked object to be completely in the field of view of the camera, then the object is said to be occluded. In multiple objects tracking, the most common occlusion that can happen in a scene is inter-object occlusion. A robust tracking system should be able to track objects which are partially or even fully occluded.

Tracking during occlusion requires detection of occlusion and tracking under occlusion. Lu *et al.* sets a threshold for the size of the bounding box for the object being tracked [Lu01]. If the size of bounding box increases unexpectedly, occlusion has possibly occurred. Occlusion can be anticipated if the object's predicted bounding box overlaps with another object's predicted bounding box. Zhou *et al.* detected occlusion by background object by checking reduction of size exceeding a certain threshold for few successive frames [Zhou01]. But finding an average size of a person is a difficult task.

In the majority of the systems used for tracking people, the problem of occlusion is solved within the framework. Earlier algorithms deal with occlusions by estimating positions and velocities of the objects [Rosales98]. After the occlusion, the correct object identities are re-established using estimated object locations. Rosales *et al.* performed this estimation using Kalman filters. But the disadvantage with these systems was, if during the occlusions, one of the objects suddenly changes its direction the estimate will be inaccurate, leading to a tracking error.

Another approach suggests mounting the camera above the observed scene [Yilmaz06]. Persons observed from such a view point cannot occlude each other. A multi-camera approach can also be used where ambiguities caused by occlusion are resolved by combining information from different cameras placed in different places [Yilmaz06]. But the problem with these approaches is that they can be used only in a few, controlled environments and their use in mobile applications is quite difficult.

Feature points can also be used to overcome occlusions. For a given object, it is likely that several feature points can be extracted. As the object moves, it can be expected that some feature points will not be visible at all times, either due to self occlusion or occlusion with other objects, but some feature points are likely to be visible. Coifman *et al.* proposed a system that used feature points to track vehicles [Coifman98]. As it is likely that each object will have multiple feature points, the use of these for tracking helps to mitigate the problems with occlusions as it is likely that even in the event of an occlusion, one of the features points will be visible. This approach is good only for partial occlusions.

Many systems rely on the appearance features of objects to classify the pixels during occlusion. For example, probability mask appearance models have been defined for objects and these models are used to localize objects during partial occlusion [Senior06]. Then pixel classification is used to identify the regions belonging to each occluding/occluded object. Capellades03 *et al.* proposed a method which used color histogram and correlogram for the modelling of the objects and then a segmentation method was used to identify the individual objects during occlusions [Capellades03]. This approach fails in most cases if an object walks behind another moving object and completely disappears in a frame.

Another approach to overcome occlusion problems is to use depth information [Koller94]. This use of depth information allows the order of the objects, according to the closeness to the camera, to be determined in the scene. This further allows the occluded objects to be segmented separately provided they are at sufficiently distinct depths. This method relies on proper segmentation at the base of the object, and thus is prone to inaccuracies.

Swain *et al.* proposed using the histogram backprojection technique for occlusion handling [Swain91]. The limitation of this technique is that, when two people have similar color, the backprojection cannot segment the occluded blob into individuals, rather they segment the whole blob as a candidate region for the target. Also it fails in the case of complete occlusions.

Handling occlusions by using human models has been proposed by many authors. Zhang *et al.* used a human model divided into five parts: head, left and right arms, and left and right legs [Zhang08]. Six features were used for matching purpose. The head was used as an anchor for aligning the human model onto the candidate blobs for matching. The authors used a layered data association approach, where direct comparison between features and subsequent matching between parts of the same bodies led to a final decision for a global match. An updating procedure is used to update the human model's parts independently of one another. The method presented was not able to handle long term occlusions. In this tracking approach, the core assumption made about the scene was that the tops of the heads are visible at all times, which was also a limitation for different kind of scenarios where the object is completely occluded.

An object tracking system based on the inter frame displacements of detected objects was proposed, and this same information was used to handle occlusions [Rashid09]. The humans are separated from the occlusion group based on information of direction of movement. The direction of walking was used to find out the matching point and was computed as an angle from -180 degree to +180 degree. This method assumed that the inter frame displacement of human is smaller than half the local width of the human. But this method fails to track people, if the people stop for a long time in the state of occlusion. This method also fails if the occluded object changes direction after being occluded for a long time.

Jin *et al.* proposed a method for handling occlusions for a single target tracking in a complex environment [Jin10]. If the similarity between the tracking result of each frame and target template becomes smaller than a preset threshold, full occlusion is said to have occurred. In such cases, large non-zero regions are searched in the object's probabilistic distribution map near the place where the object disappeared. If the large non-zero region is found, it will be regarded that the lost object reappears and the tracking resumes automatically. If no large region is found, the searching window range is extended continuously. Since this approach is tested and works with a single target tracking, its applications for handling occlusions in case of multiple objects are limited.

Youssef *et al.* proposed different techniques for handling partial and full occlusions [Youssef10]. Partial occlusion is solved by scanning the difference image pixel by pixel from top to bottom. Two peaks appear representing the heads of the two occluded objects. The modified algorithm splits the bounding box containing the two occluded objects into two bounding boxes separating the two objects from each other. Then each object is matched by comparing it to previously detected objects in previous frames. Full occlusion is solved by comparing the object in front to all objects previously detected. Since the two objects are already defined before occlusion occurs and the object in front is just labelled, therefore the hidden object is known as well. This approach is good enough for tracking just two persons.

3.5 Conclusion

In this chapter, the related work in foreground segmentation, object tracking and occlusion handling has been discussed. Regarding foreground segmentation, we discussed in brief about the frequently used techniques – background subtraction, temporal differencing and optical flow. In object tracking, we saw what kind of features could be used for tracking. Finally we end the chapter with the state of art approaches in handling occlusions during tracking.

The lack of common evaluation data makes direct comparison of individual techniques difficult, as the vast majority of evaluation is performed on privately captured datasets, with any performance metrics used varying from author to author. For this reason, no comparison of performance is given.

CHAPTER IV

DESIGN AND METHODOLOGY

4.1 Multiple People tracking with Occlusion Handling

In this chapter the main idea for our tracking model is discussed. This chapter describes the computational models employed in our approach to build a robust tracking system. In the framework defined in this thesis, the image frames of a scene are recorded by a static camera, and the foreground objects are segmented from the background, before starting the tracking hypothesis. The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video [Yilmaz06]. The computational complexity and even the constant factors of the algorithms we use are important for robust performance. Hence, our decision on selecting the following algorithms for various problems is affected by their computational run time performance as well as quality. This framework is capable of detection occlusion and recovering the occluded regions.

The first step of our approach is separating the foreground objects from a stationary background. To achieve this, we use K-means clustering combined with other low-level image processing methods to create a foreground pixel map at every frame. We then group the connected regions in the foreground map to extract individual object features, such as color histogram, bounding box, centroid, etc. This framework then utilizes the extracted object features together with a correspondence matching scheme to track objects from frame to frame. Sub-blobbing has been used in the case of occlusions. We assume a stationary camera.

The remainder of the chapter present the computational models and methods adopted for object detection and tracking. Our approach of handling occlusions is explained in the subsequent chapter.

4.2 Background Subtraction/Foreground Segmentation

Every tracking method requires an object detection mechanism either in every frame, or when the object first appears in the video. In our tracking system we have used a combination of histogram-based method for background modelling and k-means clustering method to identify the foreground objects [Indupalli06]. Object detection can be achieved by building an ideal background model and then comparing each incoming frame with the model to find significant changes in the corresponding pixels. Because of changing lightening conditions, as well as in crowded scenes, it becomes very necessary to have a dynamic background model, which keeps updating with the passage of time. To overcome this problem, we have used a histogram-based method for background modelling [Induppali06].

The background was modelled in the HSV color space. HSV stands for Hue, Saturation and Value. HSV color space was chosen, because it enables a better usage of color information and it is much better in dealing with noise and shadow when compared with other color spaces, and thus can be used to gain control over certain aspects of tracking. The following algorithm has been used.

1. Convert each frame in HSV color space. 2. Calculate the Histogram of H, S and V component for a particular pixel in all the frames. 3. Find the histogram bin with highest value and assign the median of this bin to the H, S and V component of the pixel in the background model. 4. Perform step 1, 2 and 3 for all the pixels in the background model until the pixel values are stabilized.

Table 4.1 Background Modeling

A histogram of four bins was constructed which represented the range of pixel intensity values, and then the median of the largest bin in the histogram was chosen as the pixel intensity of the background model.

Our next step in the detection process is detecting the foreground pixels by using the background model and the current frame. This pixel level detection process is dependent on the background model in use and it is used to update the background model to adapt to dynamic scene changes. Previous methods suggested setting a threshold value for differentiating between the foreground pixels and the background. But a global threshold value would not work for different kinds of background, especially where the background keeps changing dynamically. To overcome this problem, we used k–means clustering technique proposed by Indupalli et al. [Indupalli06]. The value of k represents the number of clusters to be formed, and for our system, we chose the value of k to be 2, one each for background pixels and foreground pixels.

1. Convert input frame into HSV color space. 2. Subtract the H, S and V components of the background model from the H, S and V components of the video frame and store the absolute value into a difference image. 3. Find minimum and maximum values in the difference matrix. The minimum value corresponds to the seed of the background cluster, denoted by M1 and maximum value corresponds to the seed of foreground cluster, denoted by M2. 4. For every pixel i in video frame, calculate Dij=|xi – Mj|, j= 1 & 2, and xi is the value of pixel i 5. If Di1>Di2 then assign i to FG cluster, otherwise assign i to background cluster. 6. Calculate mean of the background cluster, M1 and the mean of the foreground cluster, M2. 7. Repeat step 4 and 5 until M1 and M2 does not change significantly. 8. Report pixels in foreground cluster as foreground region and vice versa.

Table 4.2: Segmentation by k-means clustering

4.3 Pixel Level Post-Processing

Also due to environmental effects and camera noise, the detected foreground pixel map may contain noise. Pixel-level post processing operation, erosion and dilation [Xiaofeng10, Li09], are performed to remove noise in the foreground pixels. Morphological operators work mostly on binary images, where a structuring element with its origin at the center pixel is shifted over the image and at each pixel of the image its elements are compared with the set of underlying pixels. If the two sets of elements match the condition defined by the set operator, the pixel underneath the origin of the structuring element is set to a pre-defined value (0 or 1) [Jain]

Dilation, as the name implies, expands one-pixel thick boundary of the foreground region. Erosion is the reverse of dilation and erodes the foreground regions with one-unit thick pixels. The order and the amount in which we apply these morphological operations are very important. The required number of morphological operations depends on the type of environment, and the amount of noise that is generated. Based on our experimental results, we have found that three erosions followed by three dilations were good enough to remove most of the noise.

Since our further steps such as object classification and tracking depends on the correctness of object segmentation, it is very important to cope with the shadows, as moving shadows in the scene might cause false segmentation results in the silhouettes of the moving objects.

In our tracking system, the advantage of HSV color space was utilized for eliminating shadows [Cucchiara05]. In HSV color space, if the object changes intensity from frame to frame because of shadows, this change can be seen mostly in the Value component of the HSV space. Therefore in order to make tracking less sensitive to shadows and illumination, the weight of the value component can be set to a very small number.

4.4 Connecting Regions

After detecting foreground regions and applying post-processing operations to remove noise and shadow regions, the filtered foreground pixels are grouped into connected regions (blobs). We have used modified 8-connectivity component labelling algorithm [Latzel05, Boufama07]. Instead of the usual way of assigning labels to pixels connected directly to each other, and then clustering them in a group, distant pixels which are connected to each other, if they were located within a neighbourhood distance *D*. Connected-component grouping can then be used to identify all connected foreground regions, and eliminates those that are too small to correspond to real moving objects. The algorithm is described below.

1. Find a foreground pixel which is not considered in any blob. Consider this pixel as a seed for a new blob. Take a buffer and a queue for the new blob. Insert the seed into the queue.
2. Dequeue a pixel from the front of the queue and insert into a buffer. Check 8 neighbours of the current pixel, and if they fall within a neighbourhood distance D, insert those pixels in the queue.
3. Repeat step 2 until the queue becomes empty.
4. Report all the pixels in the memory as one blob.
5. Repeat step 1 until all foreground pixels are assigned to any blob.

Table 4.3: Connected Component Labelling

4.5 Region Level Post-Processing

The morphological operations can remove discrete noise, but they cannot remove a group of noisy pixels that cluster together and form a blob. In order to eliminate this type of regions, blobs that have a smaller size than a pre-defined threshold are deleted from the foreground pixel map.

4.6 Object Feature Extraction and Tracking

The aim of object tracking is to establish a correspondence between objects in consecutive frames and to extract temporal information about objects such as trajectory, speed and direction. We have used object level tracking in our system, i.e. we do not track human parts such as head or limbs, but we track humans as a whole from frame to frame. For tracking, we have used a feature-based tracking algorithm where region matching is performed using object features. Feature tracking estimates the motion of the objects, by extracting features of the moving objects from the first frame, and finding those features in the corresponding objects, in the second frame. Therefore, features that can be accurately tracked must be selected. An ideal feature usually has the following characteristics: It is unique and identifiable, it exists from frame to frame, and it provides new information that is useful for stabilization process. The algorithm uses pre-extracted spatio-temporal features in the feature matching stage to increase tracking accuracy, as well as identify possible occlusions. In order to avoid cases where one feature fails, because of illumination changes, change in shape or occlusion, we have used multiple features.

Color histogram is chosen because of its robustness to view changes, noise and partial occlusion, it is easy to build and fast to process, and the best part is the histogram normally differs for each person being tracked. Therefore, its usage is well justified for multiple object tracking. However, histogram by itself suffers from illumination changes, color similarity with background objects and mainly the loss of spatial information. Therefore the other features that were used along with histogram are the size, blob bounding box, centre of mass (centroid) and the motion vector. We calculated the size of the object by counting the number of the foreground pixels that were contained in the bounding box of the object. In order to calculate the location of the centroid ($Centre_x$, $Centre_y$) of an object, we use the following equation:

$$Centre_x = \frac{\sum_{i}^{n} x_i}{n}, \qquad Centre_y = \frac{\sum_{i}^{n} y_i}{n}$$

where i is the i^{th} pixel, and n is the total number of pixels for that object.

Features are initially built when the tracked object is first detected, and are updated every subsequent frame. A combination of Euclidean distance and Pearson correlation coefficient has been used for matching feature-vector with temporal templates [Boufama07]. Euclidean distance measures the amount of difference between two vectors, whereas Pearson correlation coefficient calculates the measure of similarity.

4.7 Handling Occlusions

Tracking detected objects through occlusions, frame by frame in video, is a significant and difficult task. Occlusion handling is a crucial part of intelligent video surveillance since without occlusion handling, the system cannot extract cohesive temporal information objects, while they are hidden (partially or completely), and this affects the overall tracking accuracy. Most of the object detection methods are not able to track during occlusions. Therefore special techniques are required to continue tracking objects even in occlusion cases. The most logical solution used is to keep the camera at a height where occlusions could be avoided, by using the top view of the tracked subjects. But camera position is quite often dictated by architectural constraints. It may not be desirable to have a point of view that is too high with respect to the observed scene, since this may limit the ability to classify the tracked objects. Hence there is every possibility that a tracked object get occluded (at least partially) by another moving object or by a static object in the scene.

In general, tracking through occlusion is a highly ill-posed problem as the hidden part of the object can acquire any shape. Moreover, if the object is completely occluded, then theoretically it is not possible to determine the object's location and shape. Thus only partially occluded objects will be considered in this thesis, where information is available for the unoccluded part. Furthermore, it can be safely assumed for many cases, that, objects undergo a smooth deformation from frame to frame.

The occlusion handling by sub-blobbing algorithm works in a top-down approach. After dividing the occluded blob into sub-blobs, the resulting list of sub-blobs (candidates) is compared to the list of tracked objects. The sub-blob and object pair which yields the highest match score is matched followed by the next lowest until all sub-blobs have a valid match (determined by a threshold). The overall tracking performance varies as the threshold is changed. For example, in case of an object that is occluded by more than half, the rate at which a correspondence is established is reduced as the threshold value is reduced. On the other hand, too low a value of the threshold would increase the number of false detections of the occlusion primitives. This can lead to incorrect matches.

Occlusions are detected and handled at the same time the objects are being tracked. The possibility of occlusion or split occurs in any of the following conditions:

- If the size of the bounding box exceeds a given threshold between frames, then it can be assumed that the box contains more than one person. This case is possible when two or more persons enter together as a group in the field of view.

- By examining the bounding boxes of the persons, if they are too close, occlusion is predicted.
- If two or more objects match to the same foreground region.
- There is a huge change in direction and velocity for a blob.

Once occlusion is detected, we use our proposed method of sub-blobbing to handle the occlusions, and ensure tracking continues in a robust manner. Below the steps for occlusion handling are listed:

1. Divide the occluded blob into sub-blobs. 2. Compare the feature vectors of the sub-blobs with the temporal templates of the tracked objects. 3. Store the matching results in a score matrix. 4. The sub-blobs with a complete blob template match are the unoccluded objects. 5. Sub-blobs without a complete match, but with a match above a preset threshold are the occluded objects 6. Update the temporal templates of the objects with the temporal templates of these sub-blobs.

Table 4.4: Occlusion Handling by Sub-blobbing

When the feature vector of a sub-blob best match with the temporal template of an object, then the sub-blob is considered as the new position of the person, and the template is updated with the information of the sub-blob. For sub-blobs, which do not have a perfect match, but the match is above T_{min} (threshold for assuming the sub-blob is a match for the object), this means these objects are occluded, and the template is updated with this new information. And for sub-blobs, whose match value in the score matrix with the objects is below T_{min}, we assume either the object is completely occluded, or the sub-blob corresponds to a new object. We have matched the feature vectors of the sub-blobs with only those templates that are within a threshold. This assumption is based on the fact that a blob should not move a long distance in two consecutive frames, because of the frame rate of the camera.

In the next frame, temporal templates of the sub-blobs of the previous frame are also available for matching with the sub-blobs of the current frame. This allows matching to be more efficient and faster. As shown in Figure 4.1 the template also consists of temporal templates of sub-blobs S_{ni} which were recently matched to the object.

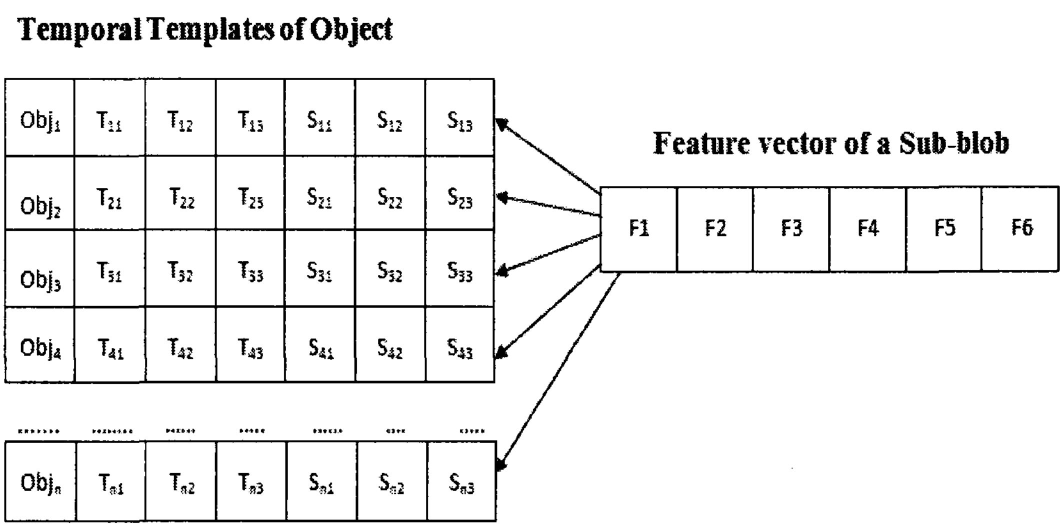


Figure 4.1: Matching features of sub-blob with temporal templates of objects [Ali06]

In case of complete occlusions, we have used a simple motion model predictor. Motion models predict the position of the object in the next frame based on the velocity in a number of past observations. The velocity can be determined as shown in equation

$$v(t) = \frac{p(t) - p(t-F)}{F}$$

Where, *v(t)* is the velocity at the current time step, *p(t)* is the position of the object at the current time step, and F is the size of the history being used. Using the current velocity, we can predict the position of the object in the next frame.

$$p(t+1) = v(t) + p(t)$$

4.8 Conclusion

In this thesis, we have proposed a novel algorithm for the detection and correction of occlusions in multiple persons tracking for video surveillance applications. We have used a feature-based model for tracking and analyze the spatial and temporal features of the object to detect sudden variations indicating a possible occlusion. Occlusion is handled by separating the occluded objects using sub blobbing.

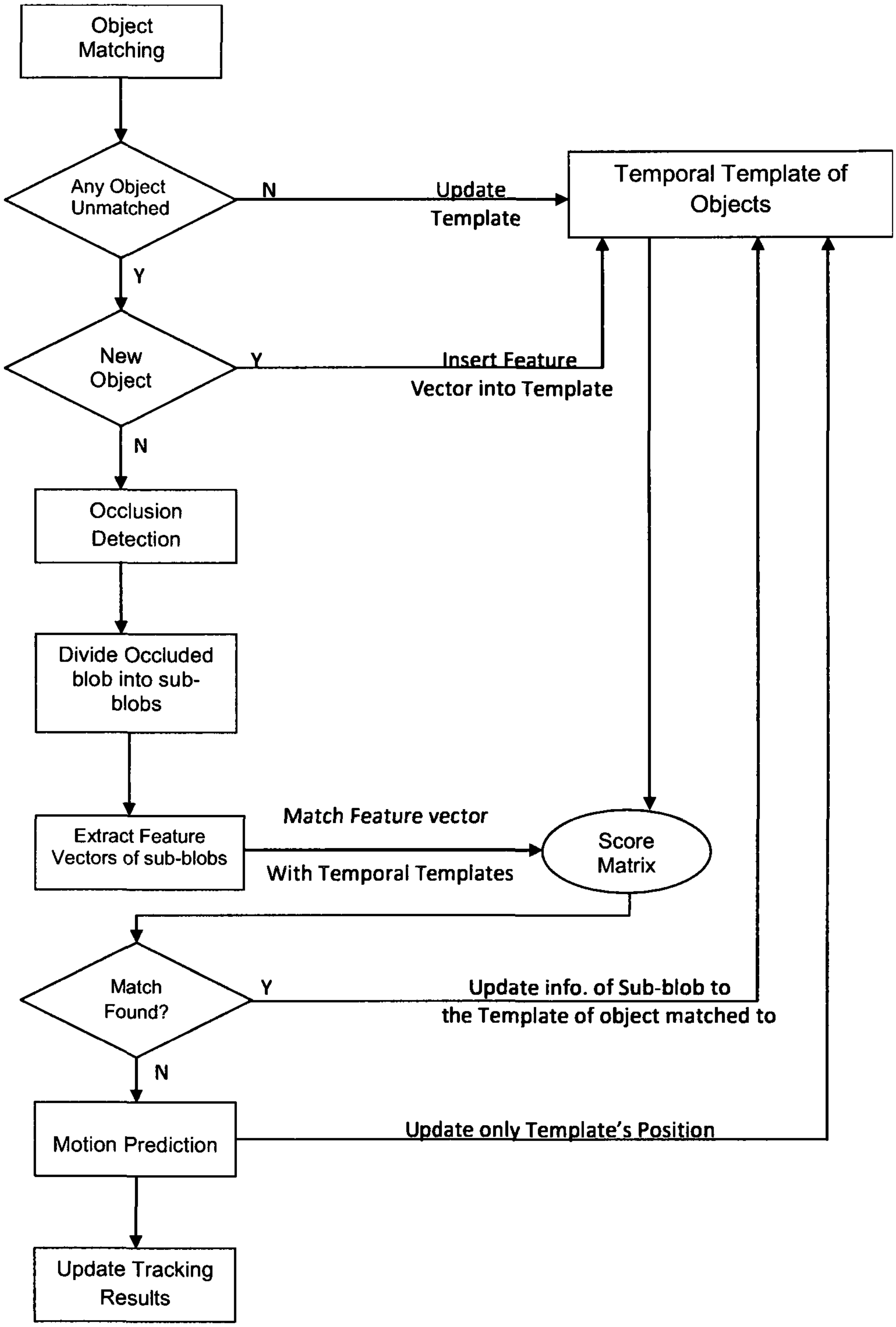


Figure 4.2 Flowchart for sub-blobbing algorithm

CHAPTER V

ANALYSIS OF RESULTS

5.1 Experimental Results

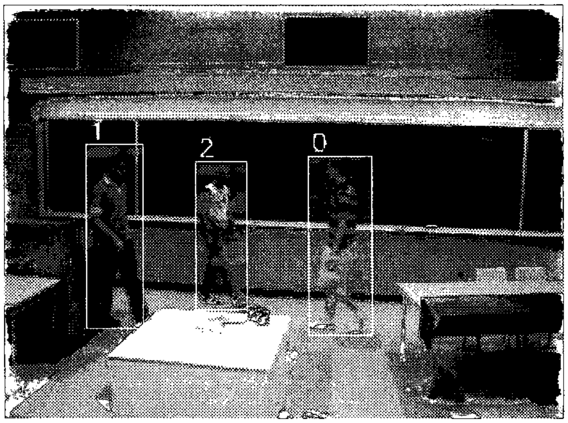
The framework has been developed in C++ using the popular computer vision library (OpenCV) as a base to provide basic image processing functionality such as:

- Image structures for image storage and manipulation
- Morphological operations such as erosion and dilation
- Associated maths functions

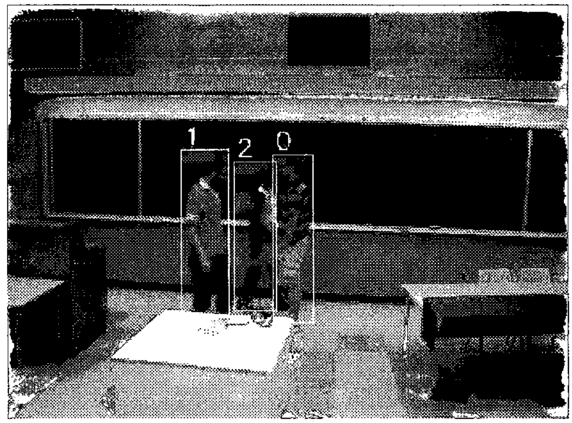
The resolutions of the images used in the video were resized to 320x240 pixels. Processing is performed at the smaller and uniform dimensions, to improve system speed and conserve storage space when performing the testing. The average number of frames processed per second (fps) is used to evaluate the system performance in terms of data throughput.

The tracking result is shown with the object labelling at the top of each blob bounding box. The label is assigned at the beginning of the tracking, and those labels are correctly kept even during occlusions, and after occlusions. For each video, the following configuration parameter had to be changed: Expected person size (minimum and maximum width of the blob bounding box). It is unrealistic to expect a single configuration to be optimal for all possible video feeds.

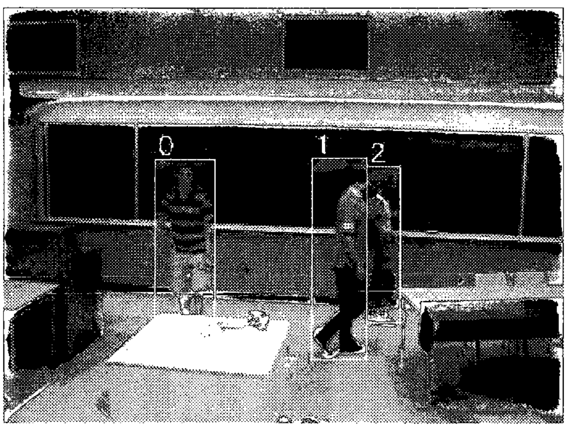
This video is taken in a large classroom and not in very bright conditions. Frame 626 shows three persons under occlusion, and how our tracker successfully distinguishes between them. Frame 687 and 715 are cases of two people occluding. Our system successfully handled the occlusion problems of merge and split.



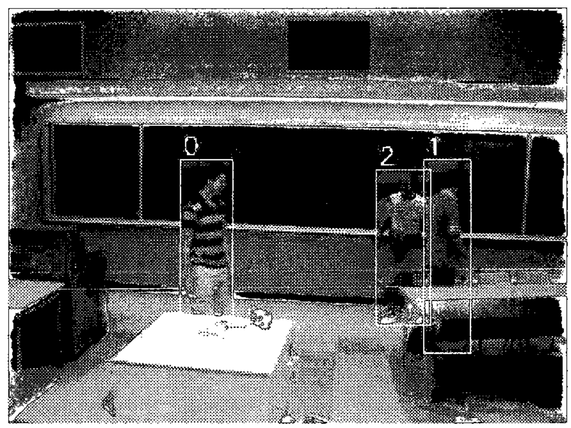
(a) Frame 597



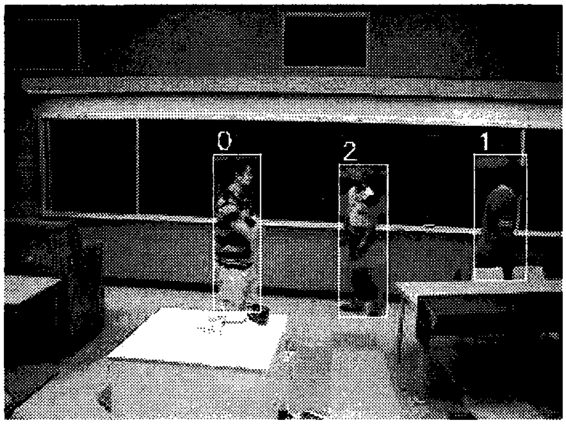
(b) Frame 626



(c) Frame 687



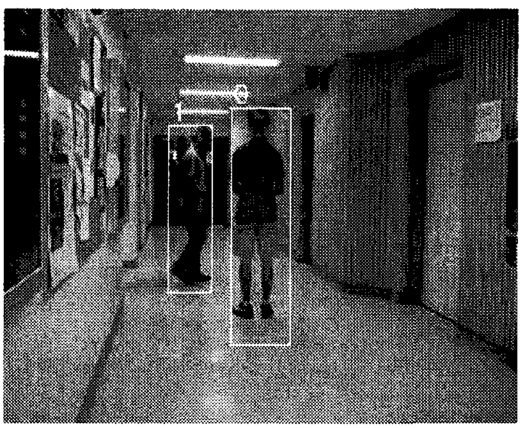
(d) Frame 715



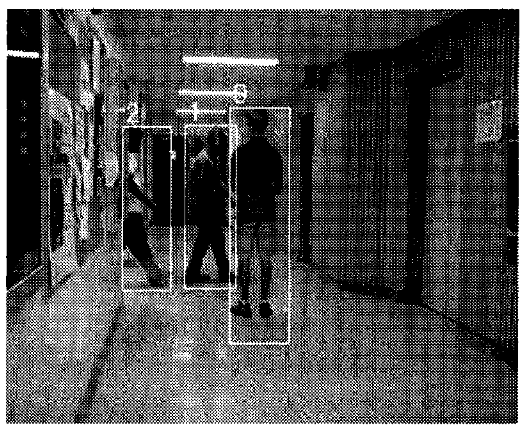
(e) Frame 734

Figure 5.1: Occlusion handling using sub-blobbing while tracking three people in a huge classroom

This video has been taken in the corridor of a building. The light condition in this scenario was quite bright. Frame 290 in fig 5.2 shows two persons tracked under partial occlusions. Frame 315 and frame 327 in fig 5.3 shows three person tracking under heavy occlusions. As you can see from the tracking result, sub-blobbing method worked well in all these cases to find the unoccluded blob.

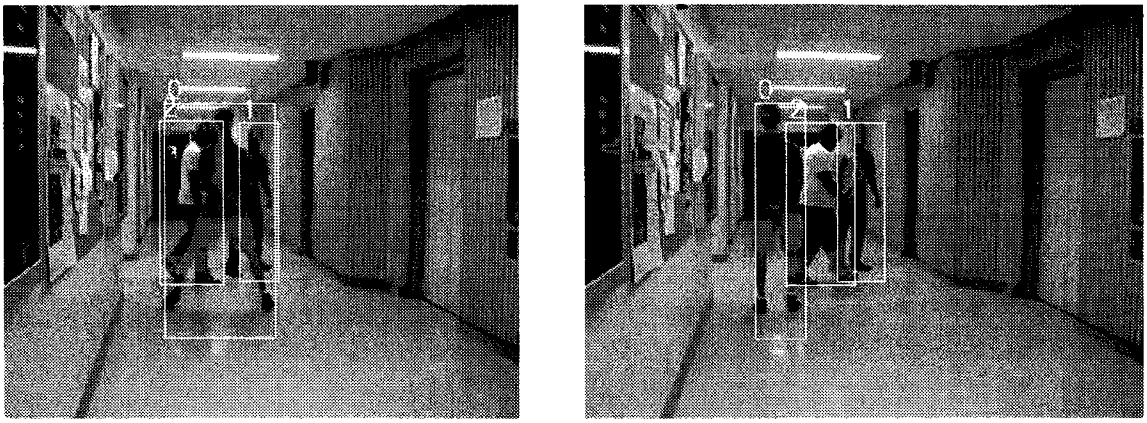


(a) Frame 281



(b) Frame 290

Figure 5.2: Occlusion Handling by Sub-blobbing of two people in the corridor



(a) Frame 315 (b) Frame 327

Figure 5.3: Occlusion handling in the case of 3 people using sub-blobbing

Errors, however, occur in some of the frames, in the size of the bounding boxes, which tend to increase progressively. This is because the scene has shadows which change the luminance of the objects in certain regions. Normally these subtle changes would be captured by the system, since the variations occur gradually from frame to frame. However, when these changes occur when the object is under occlusion, then the stored intensity templates may not remain accurate.

5.2 Analysis and Discussion

Performance metric used in the evaluation of the sub-blobbing method is

$$P = \frac{TP}{TP + FP}$$

- True Positive (TP) - number of sub-blobs matched to an object, and it corresponds to the correct object.
- False Positive (FP) – number of sub-blobs matched to an object, and it corresponds to the wrong object.

The metrics has been computed over individual frames, and then averaged over the whole sequence of frames.

5.2.1 Sensitivity to Thresholds

Our tracking system uses a large number of thresholds during the tracking process. In any system where decisions need to be made, thresholds are required. In many cases these thresholds are fixed for all cases, while in some cases thresholds need to be set at an appropriate level for different datasets. Ideally our system is functional with the thresholds set within a range of values. However for some dataset videos, there will be some values of thresholds, which may not give optimal results. Especially for cases in sub-blobbing, where only a small part of the body is visible, we would have to be very careful with the threshold values. To assess the performance of our algorithm to different threshold values, the following threshold has been evaluated.

T_{min} = Threshold for assuming the sub-blob is a match for the object

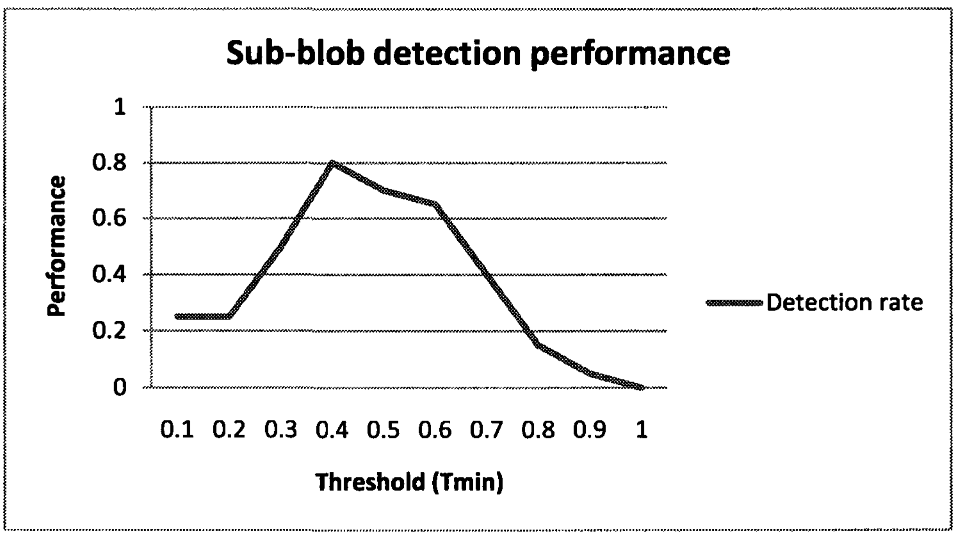


Figure 5.4: Sensitivity of tracker performance to variations of threshold

From the results we can see that if the threshold value (T_{min}) is set to low, any tracked object can be matched to any sub-blob. This can further lead to incorrect tracking. As the threshold values are increased performance increases. However as the T_{min} is increased further, performance again declines, because parts of occluded objects (sub-blobs) which do match the correct object at around 0.5 threshold value, are not declared a match because of the high value of T_{min}.

The tracking system has been tested on a 2.20 GHz Pentium® Dual core CPU. No multi-threading was employed. The tracking system was able to process 10-15 frames per second, depending on the number of people in the scene and the number of occlusions.

CHAPTER VI

CONCLUSION

There has been an increasing demand for personal and public security systems. However, utilizing human resources in such systems builds up the expenses, as well as inconsistencies due to subjective perceptions. Besides, technological devices are getting cheaper and computing power is getting advanced. All of these factors motivated me to conduct a research on the utilization of automated systems. In this thesis, we have analysed fundamental building blocks of a single camera video surveillance systems, and their challenges in the real world.

One of the main problems of any surveillance system is occlusion, which might result in the complete failure of the tracking procedure. In this thesis, we have identified and presented a solution to the very common problem of occlusions encountered by smart video surveillance systems, when used in realistic scenes. This shows that robust object detection and tracking in a single camera in a variety of environments is possible. In this thesis, we have presented an algorithm for multiple people tracking under occlusion by combining multiple features (color histogram, size, bounding box, etc.) based on their importance in different situations. The proposed method does not require the user to specify any prior shape models or motion models, and thus performs satisfactorily in unconstrained environments.

The presented sub-blobbing method successfully tracks objects through occlusions in consecutive frames. Our tests in sample videos show that feature-based correspondence matching approach is able to recognize the identities of objects during occlusions. A video surveillance system must be user friendly. Unlike other multi-camera tracking systems, our system does not need camera calibration. It can be placed anywhere and put to work immediately. Our method is also capable of handling prolonged occlusions and sees that tracking does not suffer.

Some limitations of the proposed approach are – tracking a person when it is completely occluded (motion projection) is the only option, and differentiating between multiple persons entering the scene together. These problems are significantly more difficult.

Also, due to the nature of the heuristic we use, our occlusion handling algorithm would fail in distinguishing occluding objects if they are of the same size and color.

In short, the method we presented for handling occlusions in automated video surveillance systems shows promising results and can be both used as part of a real-time surveillance system and utilized as a base for more advanced research such as activity analysis in a video.

Eventually, it would be important to extend these ideas to work in more situations, like for cameras that move, and also for detecting occluding vehicles.

For any video surveillance system, it will be useful to understand the human behaviour and predict his/her actions. For example, is a person chasing somebody? Is a person depositing some dangerous object? Is the person shoplifting? This level of understanding requires research at a greater level. In future sub-blobbing techniques could be used in handling occlusions for human behavioural analysis too.

REFERENCES

[Abdelkader06] Abdelkader, M.; Chellappa, R.; Zheng, Q.; Chan, A.; "Integrated Motion Detection and Tracking for Visual Surveillance", IEEE International Conference on Computer Vision (ICVS), p. 28, 2006

[Ali06] Ali, M.A.; "Feature-based tracking of multiple people for intelligent video surveillance", M.Sc. Thesis, School of Computer Science, University of Windsor, 2006

[Barron92] Barron, J.L.; Fleet, D.J.; Beauchemin, S.S.; Burkitt, T.A.; "Performance of Optical Flow Techniques", Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1992

[Beynon03] Beynon, M.D.; Van Hook, D.J.; Seibert, M.; Peacock, A.; Dudgeon, D.; "Detecting abandoned packages in a multi-camera video surveillance system", Proceedings, IEEE Conference on Advanced Video and Signal Based Surveillance", pp. 221-228, 2003

[Bird05] Bird, N.D.; Masoud, O.; Papanikolopoulos, N.P.; Isaacs, A.; "Detection of loitering individuals in public transportation areas", IEEE Transactions on Intelligent Transportations Systems, pp. 167-177, 2005

[Boufama07] Boufama, B.; Ali, M. A.; "Tracking multiple people in the context of video surveillance", ICIAR 2007

[Bouttefroy10] Bouttefroy, P. L. M.; Bouzerdoum, A.; Phung, S. L.; Beghdadi, A.; "On the analysis of background subtraction techniques using Gaussian Mixture Models", IEEE International Conference on Acoustics Speech and Signal Processing," pp. 4042-4045, 2010

[Capellades03] Capellades, M. B.; Doermann, D.; DeMenthon, D.; Chellappa, R.; "An object based approach for humans and object tracking", Proceedings, International Conference on Image Processing (ICIP), 2003

[Candamo10] Candamo, J.; Shreve, M.; Goldgof, D.B.; Sapper, D.B.; Kasturi, R.; "Understanding transit scenes: A survey on human behaviour recognition algorithms", IEEE Transactions on Intelligent Transportation Systems, pp. 206-224, 2010

[Cardani01] Cardani, D.; "Adventures in HSV space", The advanced Developers Hands on Conference, July 2001

[Changjiang05] Changjiang, Y.; Duraiswami, R.; Davis, L.; "Fast multiple object tracking via a hierarchical particle filter", Tenth IEEE International Conference on Computer Vision, pp. 212-219, 2005

[Cheung04] Cheung, S.-C.; Kamath, C.; "Robust techniques for background subtraction in urban traffic video", Proceedings, Video Communications and Image Processing, SPIE Electronic Imaging, 2004

[Comaniciu03] Comaniciu, D.; Ramesh, V.; Meer, P.; "Kernel-based object tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 564-577, 2003

[Coifman98] Coifman, B.; Beymer, D.; McLauchlan, P.; Malik, J.; "A real-time computer vision system for vehicle tracking and traffic surveillance", Transportation Research, Part C, vol. 6, pp. 271-288, 1998

[Cucchiara05] Cucchiara, R.; Grana, C.; Prati, A.; Vezzani, R.; "A computer vision system for in-house video surveillance", IEE Proceedings – Vision, Image and Signal Processing, pp. 242-249, 2005

[Czyzewski08] Czyzewski, A.; Dalka, P.; "Examining Kalman filters applied to tracking objects in motion", Ninth International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS, pp. 175-178, 2008

[Duanmu10] Duanmu, X.; "Image retrieval using color moment invariant", Seventh International Conference on Information technology: New Generations (ITNG), pp. 200-203, 2010

[Elgammal02] Elgammal, A.; Duraiswami, R.; Harwood, D.; Davis, L.S.; "Background and foreground modelling using nonparametric kernel density estimation for visual surveillance", Proceedings of the IEEE, pp. 1151-1163, 2002

[Elgammal00] Elgammal, A.; Harwood, D.; Davis, L.; "Non-parametric model for background subtraction", Proceedings of the 6th European Conference on Computer Vision-Part II, 2000

[Girisha09] Girisha, R.; Murali, S.; "Segmentation of motion objects from surveillance video sequences using partial correlation", 16th IEEE International Conference on Image Processing, pp. 1129-1132, 2009

[Grabner06] Grabner, H.; Bischof, H.; "On-line Boosting and Vision", IEEE Computer Science Conference on Computer Vision and Pattern Recognition, pp. 260-267, 2006

[Heikkila99] Heikkila, J.; Silven, O.; "A real-time system for monitoring of cyclists and pedestrians", Proc. of Second IEEE Workshop on Visual Surveillance, pp. 74–81, 1999

[Hsiao05] Hsiao, Y.-T.; Chuang, C.-L.; Jiang, J.-A.; Chien, C.-C.; "A contour based image segmentation algorithm using morphological edge detection", IEEE International Conference on Systems, Man and Cybernetics, 2005

[Hu04] Hu, W.; Tan, T.; Wang, L.; Maybank, S.; "A survey on visual surveillance of object motion and behaviors", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, pp. 334-352, 2004

[Huwer00] Huwer, S.; Niemann, H.; "Adaptive change detection for real-time surveillance applications", Third IEEE International Workshop on Visual Surveillance, pp. 37-46, 2000

[Indupalli06] Indupalli, S.; Ali, M.A.; Boufama, B.; "A novel clustering-based method for adaptive background segmentation", The 3rd Canadian Conference on Computer and Robot Vision, 2006

[Jain] Jain, R.; Kasturi, R.; Schunck, B. G.; "Machine vision", McGraw-Hill, 1995

[Javed02] Javed, O.; Shah, M.; "Tracking and Object Classification for Automated Surveillance", Proceedings of the 7th European Conference on Computer Vision-Part IV, pp. 343-357, 2002

[Jin10] Jin, Z.; Hongguang, S.; Weizhou, G.; Jialiang, W.; Yannan, X.; Bingnan, S.; "Robust human tracking algorithm applied for occlusion handling", Fifth International Conference on Frontier of Computer Science and Technology (FCST), pp. 546-551, 2010

[Khan03] Khan, S.; M, Shah.; " Consistent labelling of tracked objects in multiple cameras with overlapping field of view", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 25, no. 10, pp. 1355-1360, 2003

[Ko08] Ko, T.; "A survey on behavior analysis in video surveillance for homeland security applications", 37th IEEE on Applied Imagery Pattern Recognition Workshop (AIPR), 2008

[Koller94] Koller, D.; Weber, J.; Malik, J.; "Robust multiple car tracking with occlusion reasoning", ECCV, pp. 189-196, 1994

[Latzel05] Latzel, M.; Darcourt, E.; Tsotsos, J. K.; "People tracking using robust motion detection and estimation", The 2nd Canadian conference on Computer and Robert Vision, pp. 270-275, 2005

[Li09] Li, Jianyu.; Li, Feng.; Zhang, Min.; "A real-time detecting and tracking method for moving objects based on color video", Sixth International Conference on Computer Graphics, Imaging and Visualization (CGIV), pp. 317-322, 2009

[Liberty] http://www.liberty-human-rights.org.uk/issues/3-privacy/32-cctv/index.shtml

[Lipton98] Lipton, A.J.; Fujiyoshi, H.; Patil, R.S.; "Moving target classification and tracking from real-time video", In Proc. of Workshop Applications of Computer Vision, pp. 129–136, 1998

[Lu01] Lu, W.; Tan, Y-P.; "A color histogram based people tracking system", IEEE International Symposium on Circuits and Systems, pp. 137-140, 2001

[Lucas81] Lucas, B. D.; Kanade, T.; "An Iterative Image Registration Technique with an Application to Stereo Vision", Proceedings of the 7th International Joint Conference on Artificial Intelligence, pp. 674–679, 1981

[Mabey03] Mabey, G.W.; Gunther, J.; "A robust motion-estimation algorithm for multiple-target tracking at close proximity based on hexagonal partitioning", Proceeding, IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 107-112, 2003

[MacCormick99] MacCormick, J.; Blake, A.; "A probabilistic exclusion principle for tracking multiple objects", In Proc. IEEE International Conference on Computer Vision, pp. 572-578, 1999

[Manjunath01] Manjunath, B.S.; Ohm, J.–R.; Vasudevan, V.V.; Yamada, A.; "Color and texture descriptors", IEEE Transactions on Circuits and Systems for Video Technology, pp. 703-715, 2001

[McIvor00] McIvor, A.M.; "Background subtraction techniques", In Proc. of Image and Vision Computing, Auckland, New Zealand, 2000

[Mohan07] Mohan, A.; Tumblin, J.; Choudhury, P.; "Editing soft shadows in a digital photograph", IEEE Computer Graphics and Applications, pp. 23-31, 2007

[Murali09] Murali, S.; Girisha, R.; "Segmentation of motion objects from Surveillance Video Sequences using Temporal Differencing Combined with Multiple Correlation", Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 472-477, 2009

[Park07] Park, J.; An, Y.; Jeong, I.; Kang, G.; Pankoo, K.; "Image indexing using spatial multi-resolution color correlogram", IEEE International Workshop on Imaging Systems and Techniques (IST), 2007

[Polana94] Polana, R.; Nelson, R.; "Low level recognition of human motion", Proceedings, IEEE Workshop on Motion of Non-Rigid and Articulated Objects, pp. 77-82, 1994

[Radke05] Radke, R.J.; Andra, S.; Al-kofahi, O.; Roysam, B.; "Image change detection algorithms: A systematic survey", IEEE Transactions on Image Processing, pp. 294-307, 2005

[Rashid09] Rashid, M.E.; Remya, S.; Wilscy, M.; "Fast tracking of humans in frequently occurring entry, exit and occlusion scenarios", International Conference on Computer Technology and Development (ICCTD), pp. 327-330, 2009

[Rosales98] Rosales, R.; Sclaroff, S.; "Improved tracking of multiple humans with trajectory prediction and occlusion modeling", Proceedings IEEE Conference on Computer Vision and Pattern Recognition, 1998

[Senior06] Senior, A.; Hampapur, A.; Tian, Y.; Brown, L.; Pankanti, S.; Bolle, R.; "Appearance models for occlusion handling", Image and Vision Computing, pp. 1233-1243, 2006

[Shah07] Shah, M.; Javed, O.; Shafique, K.; "Automated Visual Surveillance in realistic scenarios", IEEE Multimedia, pp. 30-39, 2007

[Shireen08] E. Shireen, Y., E.-S. Khaled, M., and A. Sumaya, H. "Moving object detection in spatial domain using background removal techniques", state-of-art. Recent Patents on Computer Science, 2008

[Shor08] Shor, Y.; Lischinski, D.; "The shadow meets the mask: Pyramid-based shadow removal", Computer Graphics Forum, Vol. 27, Issue. 2, pp. 577-586, April 2008

[Statistics] http://www.statistics.gov.uk/cci/nugget.asp?id=6

[Stauffer99] Stauffer, C.; Grimson, W.E. L.; "Adaptive background mixture models for real time tracking", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 2, pp. 246-252, June 1999

[Swain91] Swain, M.; Ballard, D.; "Color Indexing", International Journal of Computer Vision, Vol. 7, pp. 11-32, 1991

[Thomas06] Thomas, B.M.; Hilton, A.; Kruger, V.; "A Survey of Advances in Vision Based Human Motion Capture and Analysis", Computer Vision and Image Understanding, October 2006

[Wang06] Wang, M.; Huang, C.; Lin, H.; "An intelligent surveillance system based on an omnidirectional vision sensor", IEEE Conference on Cybernetics and Intelligent Systems, 2006

[Wang03] Wang, L.; Hu, W.; Tan, T.; "Recent developments in human motion analysis", Pattern Recognition, pp. 585–601, March 2003

[Webopedia] http://www.webopedia.com/TERM/P/pixel.html

[Wu10] Wu, X.; Yang, L.; Yang, C.; "Real-time Foreground Segmentation Based on a Fused Background Model", The 2nd International Conference on Computer and Automation Engineering (ICCAE), pp. 585-588, 2010

[Wu08] Wu, Y.; Tong, X.; Zhang, Y.; Lu, H.; "Boosted Interactively Distributed Particle Filter for automatic multi-object tracking" , 15th IEEE International Conference on Image Processing, pp.1844-1847, 2008

[Xiaofeng10] Xiaofeng, L.; Tao, Z.; Zaiwen, L.; "A novel method on moving-objects detection based on background subtraction and three frames differencing", International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), pp. 252-256. 2010

[Xu04] Xu, D.; Liu, J.; Liu, Z.; Tang, X.; "Indoor shadow detection for video segmentation", IEEE International Conference on Multimedia and Expo (ICME), pp. 41-44, 2004

[Yeh09] Yeh, Y.; Hsu, C.; "Online Selection of Tracking Features Using AdaBoost", IEEE Transactions on Circuits and Systems for Video Technology, pp. 442-446, 2009

[Yilmaz06] A. Yilmaz, O. Javed, and M. Shah. "Object tracking: A survey", ACM Computing Surveys, 2006

[Ying01] Ying, W.; Huang, T.S.; "A co-inference approach to robust visual tracking", Eight IEEE international conference on Computer Vision, ICCV 2001

[Youssef10] Youssef, S.M.; Hamza, M.A.; Fayed, A.F.; "Detection and tracking of multiple moving objects with occlusion in smart video surveillance systems", 5th IEEE International Conference on Intelligent Systems (IS), pp. 120-125, 2010

[Zhang08] Zhang, Z.; Gunes, H.; Piccardi, M.; "Tracking people in crowds by a part matching approach", IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 88-95, 2008

[Zhang06] Zhang, W.; Fang, X.; Yang, X.; "Moving cast shadows detection based on ratio edge", 18th International Conference on Pattern Recognition (ICPR), pp. 73-76, 2006

[Zhou01] Zhou, Q.; Aggarwal, J. K.; "Tracking and classifying moving objects from video", Proceedings of 2nd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2001

[ZinBer04] Zinber, T.; Grabl, C.; Niemann, H.; "Efficient feature tracking for long video sequences", Lecture notes in computer science, pp. 326-333, 2004

VITA AUCTORIS

Mohammad Omair Alam was born in the year 1983 in Aligarh, India. He received his Bachelor of Technology degree with the specialization in Information Technology from Anna University, Tamil Nadu, India. He joined the University of Windsor in September 2007 and is currently a candidate for the Master's degree in Computer Science under the supervision of Dr. B. Boufama.