

MULTIPLE TARGET TRACKING USING MULTIPLE CAMERAS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET YILMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

MAY 2008

Approval of the Thesis

MULTIPLE TARGET TRACKING USING MULTIPLE CAMERAS

Submitted by MEHMET YILMAZ in partial fulfillment of the requirements of the degree of Master of Science in Electrical and Electronics Engineering, Middle East Technical University by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of Natural and Applied Sciences

Prof. Dr. İsmet Erkmén _____
Head of Department, Electrical and Electronics Engineering

Prof. Dr. M. Kemal Leblebicioğlu _____
Supervisor, Electrical and Electronics Engineering Dept., METU

Examining Committee Members:

Prof. Dr. Uğur Halıcı _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. M. Kemal Leblebicioğlu _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Volkan Atalay _____
Computer Engineering Dept., METU

Assoc. Prof. Dr. Aydın Alatan _____
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. İlkey Ulusoy _____
Electrical and Electronics Engineering Dept., METU

Date: 07/05/2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Mehmet YILMAZ

Signature:

ABSTRACT

MULTIPLE TARGET TRACKING USING MULTIPLE CAMERAS

Yilmaz, Mehmet

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. M. Kemal Leblebicioğlu

May 2008, 115 pages

Video surveillance has long been in use to monitor security sensitive areas such as banks, department stores, crowded public places and borders. The rise in computer speed, availability of cheap large-capacity storage devices and high speed network infrastructure enabled the way for cheaper, multi sensor video surveillance systems. In this thesis, the problem of tracking multiple targets with multiple cameras has been discussed. Cameras have been located so that they have overlapping fields of vision. A dynamic background-modeling algorithm is described for segmenting moving objects from the background, which is capable of adapting to dynamic scene changes and periodic motion, such as

illumination change and swaying of trees. After segmentation of foreground scene, the objects to be tracked have been acquired by morphological operations and connected component analysis. For the purpose of tracking the moving objects, an active contour model (snakes) is one of the approaches, in addition to a Kalman tracker. As the main tracking algorithm, a rule based tracker has been developed first for a single camera, and then extended to multiple cameras. Results of used and proposed methods are given in detail.

Keywords: Visual Surveillance, Moving Object Detection, Background Subtraction, Rule-Based Moving Object Tracking, Multiple Object Tracking Using Multiple Cameras.

ÖZ

BİRDEN FAZLA KAMERA İLE ÇOKLU HEDEF TAKİBİ

Yılmaz, Mehmet

Yüksek Lisans, Elektrik-Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Kemal Leblebicioğlu

Mayıs 2008, 115 sayfa

Video gözetimi, bankalar, mağazalar, kalabalık alanlar ve sınırlar gibi güvenlik hassasiyeti olan bölgelerde uzun süredir kullanılmaktadır. Bilgisayar hızındaki artış, ucuz ve yüksek kapasiteli depolama aygıtları ve yüksek hızlı ağ altyapıları, ucuz, çoklu kameralı video gözetim sistemlerine olanak sağlamaktadır. Bu tez çalışmasında, birden fazla kamera kullanılarak çoklu hedef takibi problemi sunulmuştur. Kameralar, gördükleri alanların bir kısmı üst üste gelecek şekilde yerleştirilmiştir. Hareket eden nesneleri arka plan görüntüsünden ayırmak için, aydınlanma değişimi ve sallanan ağaçlar gibi dinamik görünüm değişikliklerine ve periyodik hareketlere uyum sağlayan dinamik arka plan modelleme algoritması sunulmuştur. Ön plan görüntüsü ayrıldıktan sonra takip edilen nesnelerin elde edilmesi için morfolojik işlemler ve bağlı eleman analizi kullanılmıştır.

Hedef takibi sırasında Kalman süzgecine ek olarak aktif dış çevritlerden yararlanılmıştır. Ana takip algoritması olarak önce tek kamera için kural tabanlı bir takip algoritması geliştirilmiştir. Daha sonra bu algoritma çoklu kamera sistemine uygun olacak şekilde genişletilmiştir. Uygulanan ve önerilen metotlarla ilgili sonuçlar detaylı olarak verilmiştir.

Anahtar Kelimeler: Görsel Gözetim, Hareketli Nesnelerin Tespiti, Arkaplan Çıkarımı, Hareketli Nesnelerin Kural Tabanlı İzlenmesi, Birden Fazla Kamera İle Çoklu Hedef Takibi.

To my family

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Prof. Dr. M. Kemal Leblebicioğlu for his supervision, guidance and encouragement throughout this study.

I would like to also express my thanks to my friends and my colleagues for their precious support and fellowship.

Finally, I would like to thank my family for their love, support and patience over the years. This thesis is dedicated to them.

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ	vi
ACKNOWLEDGEMENTS	ix
TABLE OF CONTENTS	x
LIST OF FIGURES.....	xiii
LIST OF TABLES.....	xvii
INTRODUCTION.....	1
1.1. Scope of the Thesis	4
1.2. Outline of the Thesis	5
A SURVEY ON VISUAL SURVEILLANCE.....	6
2.1. Moving Object Detection	6
2.1.1. Background Modeling and subtraction.....	7
2.1.2. Optical Flow	8
2.2. Moving Object Tracking	8
2.3. Multiple Object Tracking	10
2.3.1. Global Nearest Neighbor Approach (GNN).....	11
2.3.2. Joint Probabilistic Data Association (JPDA).....	12
2.3.3. Multiple Hypothesis Tracking (MHT)	12
MOVING OBJECT DETECTION.....	14
3.1. Foreground Segmentation	14
3.1.1. Frame Differencing	15

3.1.2.	Running (Moving) Average	16
3.1.3.	Mixture of Gaussians	16
3.2.	Results and Discussion.....	21
3.3.	Sub-operations.....	24
3.3.1.	Shadow Removal.....	25
3.3.2.	Morphological Operations	26
3.3.2.1.	Erosion	27
3.3.2.2.	Dilation.....	28
3.3.3.	Connected Component Labeling	28
MOVING OBJECT TRACKING		30
4.1.	Kalman Tracker.....	31
4.1.1.	Kalman Filter.....	31
4.1.2.	Moving Object Tracking via Kalman Filter.....	33
4.1.3.	Interacting Multiple Model Filtering Approach	34
4.1.3.1.	Calculation of the Mixing Probabilities	37
4.1.3.2.	Mixing	37
4.1.3.3.	Mode-matched Filtering	38
4.1.3.4.	Mode Probability Update	38
4.1.3.5.	Estimate and Covariance Combination.....	38
4.2.	Active Contours (Snakes)	39
4.3.	Particle Filter Tracking	39
4.4.	Results and Discussions	40
RULE-BASED TRACKING.....		42
5.1.	Correlation Calculation Using Fuzzy Logic	43
5.1.1.	Fuzzification Interface	44
5.1.2.	Fuzzy Knowledge Base	46
5.1.3.	Defuzzification Interface.....	50
5.2.	Single Camera Tracking	50
5.2.1.	Track Initiation	54

5.2.2.	Track Confirmation	55
5.2.3.	Track Update	58
5.2.4.	Results of Singe Camera Tracking	62
5.2.5.	Evaluation of Singe Camera Tracking.....	76
5.3.	Rule-Based Tracking for Multi-Camera Configuration.....	77
5.3.1.	2D Homography.....	77
5.3.2.	The Direct Linear Transformation (DLT) Algorithm	78
5.3.3.	Calculating Track Correspondences Using Fuzzy Logic.....	81
5.3.4.	Multiple Camera Tracking	85
5.3.4.1.	Examples of Multiple Camera Tracking.....	90
5.3.5.	Results of Multiple Camera Tracking	92
	PERFORMANCE EVALUATION.....	102
	CONCLUSION	107
7.1.	Conclusion	107
7.2.	Future Work	109
	REFERENCES.....	110

LIST OF FIGURES

Figure 1.1 General scheme of single camera target tracking	2
Figure 1.2 General scheme of the developed visual surveillance system ..	3
Figure 2.1 Closely spaced multiple targets case; P1, P2 are the predicted positions of the targets and O1, O2, O3 are the observations.....	10
Figure 3.1 Results of background subtraction algorithms. (a) Input frame (b) Frame differencing (c) Running average (d) Mixture of Gaussians	21
Figure 3.2 Results of GMM method on PETS-2001 data set-3 camera-1, On the left tracking results during illumination changes and swaying trees are shown. On the right moving pixels are shown as white.	24
Figure 3.3 Result of shadow removal on PETS-2001 data set-3 camera-1, On the left tracking result is presented. On the right moving pixels are shown as white.	26
Figure 3.4 Effect of erosion operation (a) Output image of background subtraction process (b) Eroded image	27
Figure 3.5 Effect of dilation operation (a) Eroded image (b) Dilated image (resulting image of morphological operations).....	28

Figure 4.1 The Kalman filter cycle	32
Figure 4.2 One cycle of IMM algorithm	36
Figure 4.3 Kalman tracker: Estimated and real trajectories of a moving object	41
Figure 5.1 Fuzzification of input variable distance	45
Figure 5.2 Fuzzification of input variable area ratio	46
Figure 5.3 Output membership.....	50
Figure 5.4 Results of rule-based tracker on PETS-2001 data set-1 camera- 1 (a, b) Before splitting (c) During splitting (d) After splitting	63
Figure 5.5 Results for occlusion case on PETS-2001 data set-1 camera-1 (a) Before occlusion (b) During occlusion (d) After occlusion	65
Figure 5.6 Results for multiple target tracking case on PETS-2001 data set-1 camera-1	66
Figure 5.7 Results for occlusion case on PETS-2000 data set (a) Before occlusion (b) During occlusion (c) After occlusion	68
Figure 5.8 Results for multiple target tracking case on PETS-2000 data set	71
Figure 5.9 Example of a tracking failure: a person emerges from the parked car while the car being a part of the background. The identity of the car is taken by the person.....	73

Figure 5.10 Erroneous fuzzy knowledge base result on PETS-2001 data set1 camera-2	75
Figure 5.11 Erroneous fuzzy knowledge base result on PETS-2001 data set1 camera-1	75
Figure 5.12 Fuzzification of input variable Euclidean distance for multiple camera case	82
Figure 5.13 Results of target tracking using multiple camera in PETS-2001 data set-1	92
Figure 5.14 Results of multiple target tracking using multiple cameras in PETS-2001 data set-1	93
Figure 5.15 Results of multiple target tracking using multiple cameras in PETS-2001 data set-1: Tracks' occlusion in only one camera scenario ...	94
Figure 5.16 Results of multiple target tracking using multiple cameras in PETS-2001 data set-1: Track's occlusion in both cameras scenario.....	95
Figure 5.17 Results of multiple target tracking using multiple cameras in PETS-2001 data set-1: Track's occlusion in only one camera scenario ...	96
Figure 5.18 Results of multiple target tracking using multiple cameras in PETS-2001 data set-1: Occlusion ends scenario	97
Figure 5.19 Results of multiple target tracking using multiple cameras in PETS-2001 data set-2: A tracks' passing behind a stationary object scenario, left frame from camera-1, right frame from camera-2	99

Figure 5.20 Results of multiple target tracking using multiple cameras in PETS-2001 data set-2; Track which is in the FOV of one of the cameras passes behind an object..... 100

Figure 5.21 Results of multiple target tracking using multiple cameras in PETS-2001 data set-2: A tracks' passing behind a stationary object scenario-2, left frame from camera-1, right frame from camera-2..... 101

Figure 6.1 Result of single camera tracking on PETS-2001 data set-2, result of [36] is on the left (b) result of the proposed tracker is on the right 103

LIST OF TABLES

Table 5-1 Sample Observation-to-Track correlation matrix.....	43
Table 5-2 Generation of IF-THEN rules.....	47
Table 5-3 Sample occlusion matrix	53
Table 5-4 Sample Observation-to-Track matrix that is used for track initiation	54
Table 5-5 Sample Observation-to-Track matrix, after Rule-1 is applied ...	55
Table 5-6 Sample Observation-to-Track candidate matrix	56
Table 5-7 Sample Observation-to-Track candidate matrix, after Rule-1 is applied.....	56
Table 5-8 Sample number of consecutive correlated observations matrix..	57
Table 5-9 Sample occlusion matrix that is used during track update.....	58
Table 5-10 Sample Observation-to-Track correlation matrix that shows two tracks' occlusion with each other	59

Table 5-11 Sample Observation-to-Track correlation matrix that shows two tracks' occlusion with each other, after rule-1 is applied.....	59
Table 5-12 Sample Observation-to-Track correlation matrix that shows two tracks' occlusion ending	60
Table 5-13 Sample Observation-to-Track correlation matrix that shows two tracks' occlusion ending, after Rule-1 is applied	61
Table 5-14 Erroneous fuzzy knowledge base.....	74
Table 5-15 Generation of IF-THEN rules.....	83
Table 5-16 Sample Euclidean distance matrix calculated after objects in camera-1 are transformed to camera-2	86
Table 5-17 Sample area ratio matrix calculated after objects in camera-1 are Transformed to camera-2.....	86
Table 5-18 Sample matrix that shows calculated track correlation values with objects in camera-1 transformed to camera-2 and objects in camera-2	90
Table 5-19 Sample matrix that shows calculated track correlation values with objects in camera-1 transformed to camera-2 and objects in camera-2	91
Table 6-1 Results of multiple camera correspondence.....	104
Table 6-2 Results of multiple camera correspondence from [38].....	106

CHAPTER 1

INTRODUCTION

In addition to recent advances in computer speed, availability of large-capacity storage devices cheaply and high speed computer network infrastructure, the increase in crime rate has made visual surveillance popular. The history of surveillance systems consists of three generations which are called 1GSS, 2GSS and 3GSS [24]. The first generation surveillance systems (1GSS (1960-1980)) were based on analog subsystems for image acquisition, image transmission and processing which were depended on human operators for event detection. The next generation surveillance systems (2GSS (1980-2000)) used both analog and digital subsystems in order to solve the drawbacks of the 1GSS surveillance systems. They provided assistance to human operators to filter out the important events. Third generation surveillance systems (3GSS (2000-)) provided full assistance; image acquisition, image processing, storage, communication through network.

As explained above traditional security systems were greatly depending on human operators. With traditional systems, area under surveillance is restricted with the number of operators and number of cameras. These problems present the need for 3GSS surveillance systems.

In this thesis the developed visual surveillance system detects and tracks moving objects automatically. Also the developed system handles occlusion and moving objects passing through one camera view to the other camera view. This enables the objects to be tracked in entire camera views. General scheme of the developed visual surveillance system can be seen below figure.

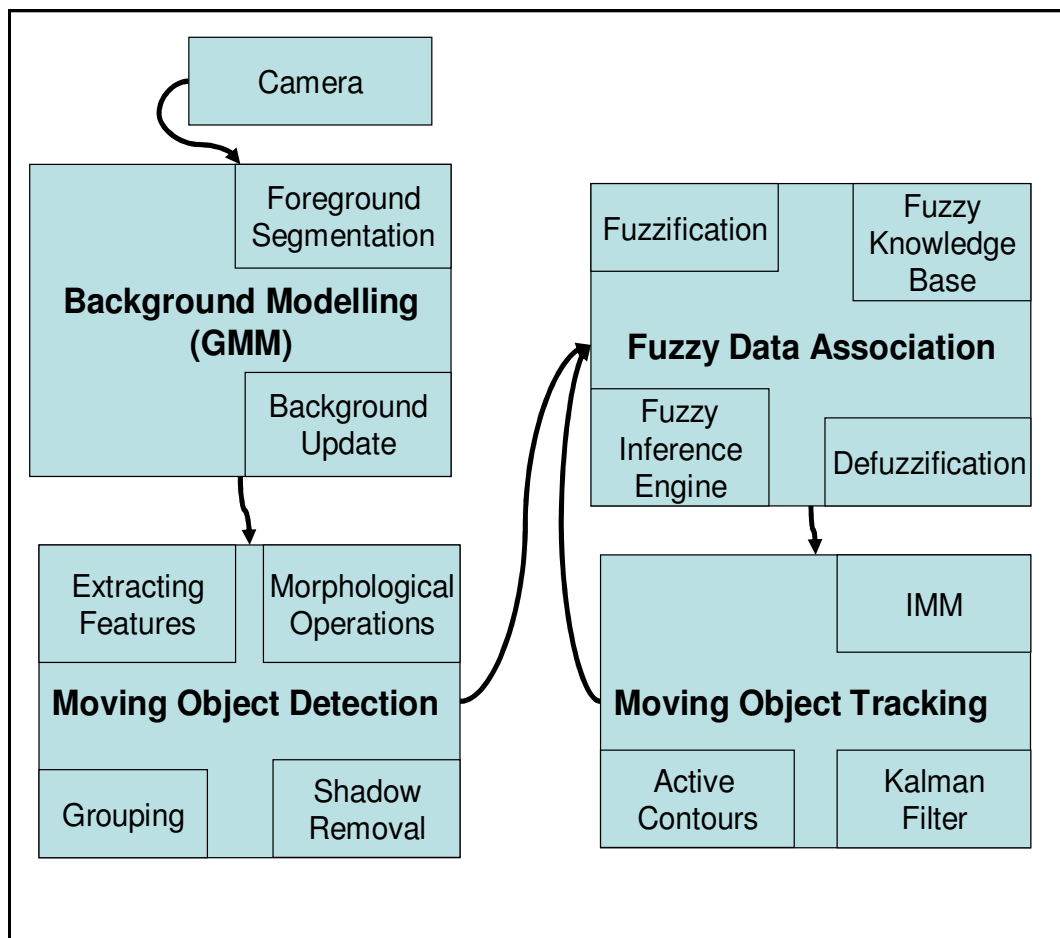


Figure 1.1 General scheme of Single Camera Target Tracking

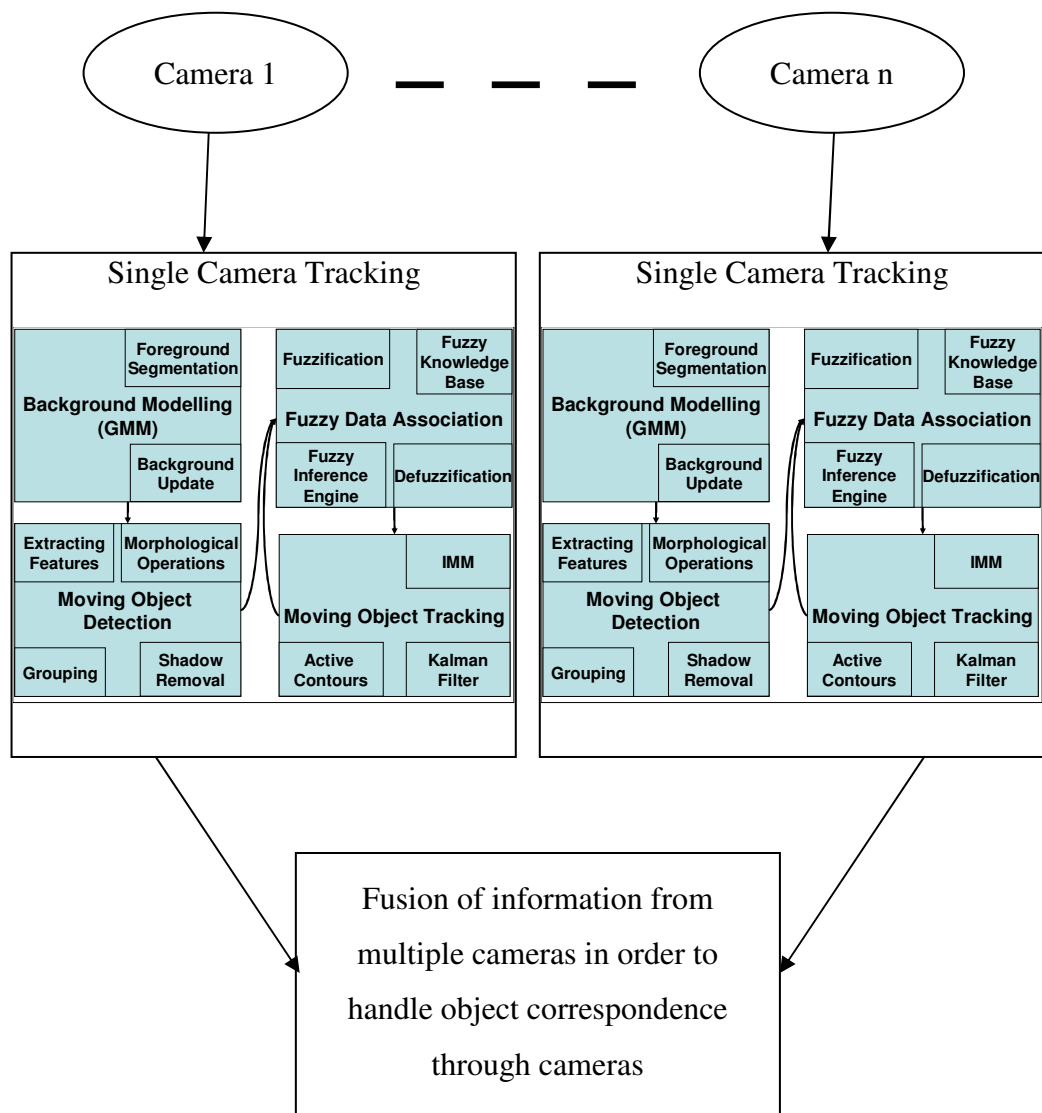


Figure 1.2 General scheme of the developed visual surveillance system

1.1. Scope of the Thesis

In this thesis, we present a smart visual surveillance system with object detection, tracking, occlusion handling and track continuation while tracks passing through one camera view to other.

Moving object detection is the basic step for tracking. It handles segmentation of moving objects from background. Due to dynamic environmental conditions such as illumination changes, shadows and swaying of tree branches in the wind, object segmentation is a difficult and for a robust surveillance system these problems should be handled carefully. Commonly used background subtraction techniques for object detection are frame differencing, running average and mixture of Gaussians which is capable of coping with the changes in the scene such as illumination change and swaying of tree branches. Morphological operations and shadow removal algorithm are used in order to remove noise and shadows.

The next section describes tracking of detected objects. Two commonly used methods have been implemented and discussed as object tracker. Kalman tracker and Particle Filter tracker have been utilized.

To cope with some difficulties, such as occlusion and track continuation over entire camera views, multiple cameras that have overlapping field of views are chosen as a solution. Using multiple cameras that have overlapping field of views and different position angles, more information can be obtained about tracked objects. Then, obtained information from cameras is used for the solution of correspondence and occlusion problems.

The tough part of multiple target tracking is the closely spaced target case. One of the best approaches to this problem is Multiple Hypotheses Tracking which solves this problem by generating track hypotheses [25]. When new data is acquired, these hypotheses are evaluated, and the true matches are found or if the data is not enough to find the true matches, new hypothesis are generated and additional data is waited for the exact match. Multiple Hypothesis Tracking is not efficient in real time since decisions are made over several scans. Thus a rule-based real time tracking algorithm is developed based on fuzzy logic.

1.2. Outline of the Thesis

In Chapter 2, related works and overview of each of the proposed system blocks are presented. Moving object detection is described in Chapter 3. Three methods are extensively described and compared by means of obtained results. Shadow removal and morphological operations are also mentioned in this chapter. Moving object tracking is given in Chapter 4. Simple Kalman filter tracking and Particle Filter tracking are presented and discussed. Rule-based real time tracking algorithm has been proposed in Chapter 5, together with a generic idea of multiple target tracking. This thesis is summarized in Chapter 6 and some future works are also discussed in this chapter.

CHAPTER 2

A SURVEY ON VISUAL SURVEILLANCE

There have been a number of surveys about object detection and tracking in the literature [26], [23]. The survey presented here includes the latest studies in the literature and covers only those works that are in the same context as our study.

2.1. Moving Object Detection

Detecting regions that correspond to moving objects in video is the first step of almost every visual surveillance system. Due to dynamic changes in scene such as illumination and weather changes, repetitive motions that cause clutter (swaying of tree branches and leaves), motion detection is a difficult problem to process reliably. The detection methods are mainly divided into two groups: optical flow and background modeling and subtraction.

2.1.1. Background Modeling and subtraction

In the context of stationary camera, background modeling and subtraction is a widely used technique [27]. A background model is constructed at each pixel location during training phase. In the new frame each pixel is classified into background or foreground according to its distance to background model. Wren et al [1] used a Gaussian model for each pixel which is sufficient for many scenarios. Stauffer et al [2] extended this approach into a mixture of Gaussians to handle the situations where the value of each pixel switches between multiple processes such as swaying of tree branches and leaves.

In W4 system Haritaoğlu [5], uses a statistical background model where each pixel is represented with its minimum (M) and maximum (N) intensity values and maximum intensity difference (D) between any consecutive frames observed during initial training period where the scene contains no moving objects. A pixel in the current image is classified as foreground if it satisfies:

$$|M(x, y) - I_t(x, y)| > D(x, y) \text{ or } |N(x, y) - I_t(x, y)| > D(x, y) \quad 2-1$$

The weakness of this model comes from its sensitivity to sudden illumination changes.

Elgammal et al. [10] uses sample background images to estimate the probability of observing pixel intensity values in a nonparametric manner without any assumption about the form of the background probability distribution. This approach yields quite accurate results under challenging outdoor conditions.

2.1.2. Optical Flow

Optical flow based motion detection uses characteristics of flow vectors of moving objects over time to detect moving regions in an image sequence. Main idea is to identify the regions that move together. For example, Meyer et al. [9] compute the displacement vector field for the extraction of articulated objects. Optical-flow-based methods can be used to detect independently moving objects even in the presence of camera motion. Phase correlation, Lucas-Kanade and Horn-Schunck methods are the mostly used methods for determining optical flow. Optical flow is a very accurate method for determining moving objects; however, most flow computation methods are computationally complex and very sensitive to noise. Thus optical flow cannot be applied to video streams in real time without specialized hardware.

2.2. Moving Object Tracking

The objective of tracking is to establish correspondence of objects and object parts between consecutive frames of video. These correspondences are established by finding the best match according to some similarity function calculated with the previous state of the objects and the new measurements. Tracking has been a difficult task to apply due to inaccurate segmentation of objects. Common problems of erroneous segmentation are illumination changes, swaying of tree branches and leaves, shadows, partial or full occlusion of objects with each other or with stationary items in the scene. Thus for robust tracking these problems should be handled carefully.

There are various methods in the literature related with visual tracking of moving objects. Tracking in video can be categorized according to the needs of the applications it is used in or according to the methods used for its solution. For outdoor video surveillance whole body tracking is generally adequate whereas objects' part tracking is necessary for some indoor surveillance and higher level behavior understanding applications.

There are two common approaches in tracking objects as a whole [28]: one is based on correspondence matching and other is based on the use of position prediction or motion estimation. The methods that track parts of objects employ model-based schemes to locate and track body parts. Stick figure, cardboard model [29], 2D contour and 3D volumetric models are some examples of model-based schemes.

W4 [5] combines motion estimation methods with correspondence matching to track moving objects. It also uses cardboard model [29] which represents relative positions and sizes of body parts, in order to track parts of people such as heads, hands, torso and feet. In order to handle matching even in merge and split cases it keeps appearance templates of individual objects.

Amer [28] presents a non-linear voting based scheme for tracking objects as a whole. It integrates object features like size, shape, center of mass and motion by voting and decides final matching with object correspondence. This method can also detect object split and fusion and handle occlusions.

Stauffer et al. [2] employs a linearly predictive multiple hypotheses tracking algorithm. The algorithm incorporates size and positions of objects for

seeding and maintaining a set of Kalman filters for motion estimation. Also, Extended Kalman filters are used for trajectory prediction and occlusion handling in the work of Rosales and Sclaroff [30].

As an example of model based body part tracking system, Pfinder [1] makes use of a multi-class statistical model of color and shape to track head and hands of people in real-time.

2.3. Multiple Object Tracking

When tracking multiple targets the most important part of tracking is the observation to track association part. Before applying a Kalman or a Particle filter, most likely measurements should be associated to tracks. The simplest method to solve the correspondence problem is to use the nearest neighbor approach. In the case of closely spaced targets there is always a chance that the correspondence is incorrect. An incorrectly associated measurement can cause the filter to do the estimation wrongly. In the below figure, we can see such a problematic case as explained.

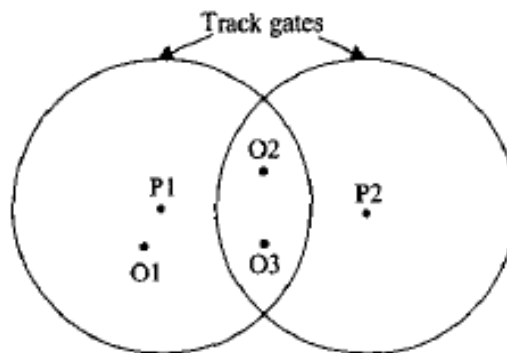


Figure 2.1 Closely spaced multiple targets case; P1, P2 are the targets predicted positions and O1, O2, O3 are the observations.

Global Nearest Neighbor Approach, Joint Probability Data Association (JPDA) and Multiple Hypothesis Tracking (MHT) are widely used techniques to tackle this problem. Here is a brief description of these techniques.

2.3.1. Global Nearest Neighbor Approach (GNN)

Global Nearest Neighbor approach finds the best assignment of observations to existing tracks [25]. The term global comes from the fact that observation to track assignment is made considering all possible observation to track associations under the constraint that an observation can be associated with at most one track. In Figure 2.1 Global Nearest Neighbor would associate O1 to track 1, O2 to track 2 (we assume that O2 is closer than O3 to the track 2) and O3 will initiate a new track. In Global Nearest Neighbor, track conformation is done by detection of the track in five (generally five but number can change) consecutive frames, and track deletion is done by missing of the track in 5 (generally five but number can change) consecutive frames. Then next states of all tracks are predicted. Kalman filter prediction covariance provides the uncertainty that is required for gating and association.

GNN approach considers the single most likely hypothesis; thus when the targets are widely spaced it gives good results. But, in the case of closely spaced targets this approach does not work well and tends to disassociations. In order to improve the performance of GNN approach one solution was to increase the Kalman filter covariance matrix in order to reflect this uncertainty.

2.3.2. Joint Probabilistic Data Association (JPDA)

JPDA is a target oriented approach; that is, for a known number of targets it evaluates the measurement-to-target probabilities and combines them into the corresponding state estimates [25]. The major limitation of the JPDA algorithm is its inability to handle new objects entering the field of view (FOV) or already tracked objects exiting the FOV. Since the JPDA algorithm performs data association of a fixed number of objects tracked over two frames, serious errors can arise if there is a change in the number of objects. Joint Probabilistic Data Association method offers updating the track by a weighted sum of all observations within its gate. For the case presented in Figure 2.1, track 1 is updated with O1, O2, O3 and track 2 is updated with O2 and O3.

2.3.3. Multiple Hypothesis Tracking (MHT)

Multiple Hypothesis Tracking is an algorithm which solves the multiple target data association problem in an exact way as the number of track hypotheses goes to infinity [25]. Algorithm begins with a set of current track hypotheses. Each hypothesis is a collection of disjoint tracks. For each hypothesis, a prediction of each object's position in the next frame is made. The predictions are then compared with actual measurements by a distance measurement and a set of associations are established for each hypothesis based on this distance measurement, then new hypotheses are generated.

Each measurement can belong to a new object entering the FOV, a previously tracked object, or a spurious measurement. Moreover, a measurement may not be assigned to an object because the object may

have exited the FOV, or a measurement corresponding to an object may not be obtained. The latter happens because either the object is occluded or it is not detected due to noise.

The MHT algorithm is computationally exponential both in memory and time. To overcome this limitation, Cox and Hingorani [31] use a special algorithm to determine best hypotheses in polynomial time for tracking interest points. There are other techniques that keep the number of hypothesis in acceptable numbers. These techniques are clustering, pruning and track merging. Clustering is performed to reduce the number of hypothesis that must be generated and evaluated. Clusters are collections of tracks that are linked by common observations. Thus clusters decompose a large problem into small and manageable problems.

In this study Multiple Hypothesis Tracking technique is not chosen as the data association technique since it is computationally exponential both in memory and time.

CHAPTER 3

MOVING OBJECT DETECTION

Performance of an automated visual surveillance system considerably depends on its ability to detect moving objects. This step is also called motion segmentation or foreground extraction. The following actions such as moving object tracking, analyzing the motion or classifying objects, requires an accurate extraction of the foreground objects. Thus, moving object detection is an important part of the visual surveillance systems.

3.1. Foreground Segmentation

The aim of the foreground segmentation is to detect changes between consecutive frames which are taken from the same scene and acquired by a static camera. The pixels that have significant difference compared to the previous ones are labeled as foreground pixels, and other pixels are marked as background. The pixels that are labeled as foreground forms the segmented moving objects. In order to decide which pixels are foreground and which are background there should be a background model. If the work is about outdoor scenes, this model should also be able to adapt to dynamic scene changes like illumination change, swaying tree leaves and shadows.

There are many works on moving object segmentation. Due to its simplicity and computational efficiency background subtraction method is chosen to be examined and implemented. Below some background subtraction methods are discussed.

3.1.1. Frame Differencing

Frame differencing is the simplest way of detecting moving objects. In this method background is always the previous frame. For detecting foreground objects previous frame is subtracted from current frame and result is thresholded. The pixels that have larger difference than the threshold are labeled as foreground.

$$M(x, y, t) = \begin{cases} 1, & I(x, y, t) - I(x, y, t-1) > Threshold \\ 0, & I(x, y, t) - I(x, y, t-1) < Threshold \end{cases} \quad 3-1$$

In the above formula, $I(x, y, t)$ is the intensity value at pixel location (x, y) at time t and $I(x, y, t-1)$ is the intensity value at pixel location (x, y) at time $t-1$. $M(x, y, t)$ is the mask image resulting from differencing and thresholding operations.

This method is fast because it does not do any background update operation and frame differencing is taken as a whole instead of pixel by pixel operation. This method is also able to adapt scene changes since it takes the previous frame as the background. Besides these advantages this method is very sensitive to threshold value, has an aperture problem and gives low performance in dynamic scene conditions.

3.1.2. Running (Moving) Average

In this method background frame is constructed by calculating the mean value of the previous N frames. Moving objects are segmented by subtracting the constructed background frame from current frame. The pixels that have difference values greater than the threshold are labeled as foreground. This is shown in the equation below:

$$m(x, y, t) = \begin{cases} 0 & \text{if } |I(x, y, t) - I_{ref}| < \text{Threshold} \\ 1 & \text{if } |I(x, y, t) - I_{ref}| > \text{Threshold} \end{cases} \quad 3-2$$

And the update equation is:

$$I_{ref,t} = \alpha * I(x, y, t - 1) + (1 - \alpha) * I_{ref,t-1} \quad 3-3$$

In the above equation $m(x,y,t)$ is the mask, α is the learning parameter. Moving average filtering is not robust to slowly moving objects and recovers slowly because of averaging; it also suffers from threshold sensitivity and cannot cope with multi-modal distributions, whereas yields a better background modeling with respect to the frame differencing.

3.1.3. Mixture of Gaussians

The tracking systems should be robust to lighting changes as well as they should deal with the movement through cluttered areas, repetitive motion like swaying trees and objects being left or removed. Traditional methods fail in these situations. Gaussian Mixture Models creates a background model that is adaptive to changes in background [32].

The Gaussian Mixture Model approach models each pixel as a mixture of Gaussians. Each pixel is classified based on whether the Gaussian distribution which represents it most effectively is considered part of the background model. This results in a stable background model which is robust to lighting changes, repetitive motions like swaying trees and long term scene changes like parking car, a left or removed object.

Instead of modeling the entire background as one particular type of distribution, GMM approach models every pixel as a mixture of Gaussians. Which Gaussian may correspond to the background is determined based on the persistence and the variance of each of the Gaussians. Pixel values that do not fit the background distributions are considered foreground until there is a Gaussian that includes them with sufficient, consistent evidence supporting it. Thus background robustly deals with lighting changes, repetitive motion of scene elements, slowly moving objects and left or removed objects. Repetitive motions are learned, and a model for the background distribution is generally maintained even it is temporarily replaced by another distribution, which leads to faster recovery when objects are removed. This method has two significant parameters, the learning constant and the proportion of the data that should be accounted for by the background.

Each time the parameters of the Gaussians are updated, the Gaussians are evaluated using simple heuristic to hypothesize which are most likely to be part of the background process. Pixel values that do not match one of the pixels background Gaussians are considered as foreground. Then these pixels are grouped and moving objects are found.

This method models a specific pixel value as a mixture of weighted K 3-D Gaussian distributions in the color space. The value of K is generally 3 to 5; it is determined by the available memory and computational power. The recent history of each pixel, $\{X_1, \dots, X_t\}$, is modeled by a mixture of K Gaussian distributions. The probability of observing the current pixel value is:

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad 3-4$$

where $\omega_{i,t}$ is an estimate of the weight (what portion of the data should be accounted for by this Gaussian) of the i th Gaussian in the mixture at time t , $\mu_{i,t}$ and $\Sigma_{i,t}$ are the mean value and covariance matrix of the i^{th} Gaussian in the mixture at time t , and η is a Gaussian probability density function:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \quad 3-5$$

For computational reasons, the covariance matrix is assumed to be of the form:

$$\Sigma_{k,t} = \sigma_k^2 I \quad 3-6$$

This assumes that the red, green, and blue pixel values are independent and have the same variances. While this is certainly not the case, the assumption allows us to avoid a costly matrix inversion at the expense of some accuracy.

Thus, the distribution of recently observed values of each pixel in the scene is characterized by a mixture of Gaussians. Each new pixel value will be represented by one of the major components of the mixture model and used to update the parameters of that component of the mixture.

Every new pixel value, X_t , is checked against the existing K Gaussian distributions (starting with the most likely background Gaussians) until the first match is found. A match is defined as a pixel value within 2.5 standard deviations of a distribution. If none of the K distributions match the current pixel value, the least probable distribution is replaced with a distribution with the current pixel value as its mean value, an initially high variance, and low prior weight. The prior weights of the K distributions at time t are adjusted as follows

$$\omega_{k,t} = \alpha * M(k,t) + (1 - \alpha) * \omega_{k,t-1} \quad 3-7$$

where, α is the learning rate and $M_{k,t}$ is '1' for matched models, and '0' for remaining models.

The mean (μ) and variance (σ) parameters for unmatched distributions remain the same. The parameters of the distribution which matches the new observation are updated as follows

$$\mu_t = (1 - \rho)\mu_{t+1} + \rho X_t \quad 3-8$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T (X_t - \mu_t) \quad 3-9$$

where $\rho = \alpha\eta(X_t | \mu_k, \sigma_k)$. These equations are logically the same as the equations used in weight updating equation and in running average method.

Measurements must be organized such that any changes in the scene must be inserted into the model. For this reason, Gaussians are evaluated according to the following approach. First, the Gaussians are ordered by the value of ω/σ . This value increases as the distribution gains more evidence and as the variance decreases. After re-estimating the parameters of the mixture, it is sufficient to sort from the matched distribution towards the most probable background distribution, because only the matched models relative value will have changed. This ordering of the model is effectively an ordered, open-ended list, where the most likely background distributions remain on top and the less probable transient background distributions gravitate towards the bottom and are eventually replaced by new distributions. Then the first B distributions are chosen as the background model, where

$$B = \arg \min_b \left(\sum_{k=1}^b \omega_k > T \right) \quad 3-10$$

where T is a measure of the minimum portion of the data that should be accounted for by the background. This takes the “best” distributions until a certain portion, T, of the recent data has been accounted for. If a small value for T is chosen, the background model is usually unimodal. If this is the case, using only the most probable distribution will save processing. If T is higher, a multi-modal distribution caused by a repetitive background motion (e.g., leaves on a tree, a flag in the wind, a construction flasher, etc.) could result in more than one color being included in the background

model. This results in a transparency effect which allows the background to accept two or more separate colors.

3.2. *Results and Discussion*

Acquired results of moving object detection methods and comparison of these methods have been presented in this section. PETS2001 video sequence has been used to obtain the results of the methods.

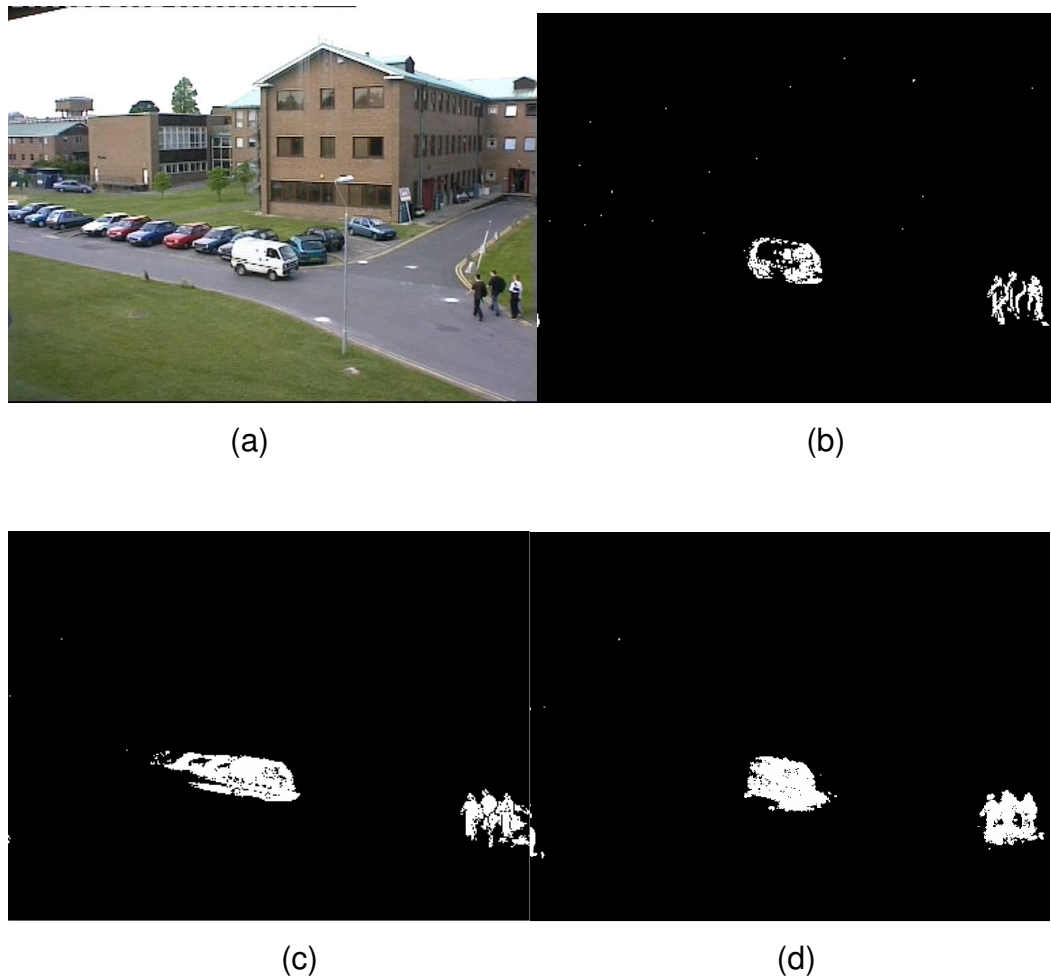


Figure 3.1 Results of background subtraction algorithms. (a) Input frame
(b) Frame differencing (c) Running average (d) Mixture of Gaussians

Frame differencing is the simplest way of detecting moving objects. This method is fast because it does not do any background update operation. Besides these advantages this method is very sensitive to threshold value, has an aperture problem and gives low performance in dynamic scene conditions.

Compared to frame differencing, moving average yields a better background modeling. But it has some deficiencies like not being robust to slowly moving objects, recovering slowly because of averaging. It also suffers from threshold sensitivity and cannot cope with multi-modal distributions.

In tracking systems the system should be robust to lighting changes as well as they should deal with the movement through cluttered areas, repetitive motion like swaying trees and objects being left or removed. Traditional methods fail because of these situations. GMM creates a background model that is adaptive to changes in background.

It can be said that among all of the methods, mixture of Gaussians is the most powerful method since it is robust to lighting changes also it deals with movement through cluttered areas, repetitive motion like swaying trees and objects being left or removed. Consequently, it has been chosen as the detection method for subsequent tracking module. Some sample results of GMM algorithm on PETS-2001 video sequence is presented.



(a) Frame-5



(b) Frame-500



(c) Frame-900



(d) Frame-1300



(e) Frame-1435

Figure 3.2 Results of GMM method on PETS-2001 data set-3 camera-1. On the left tracking results during illumination changes and swaying trees are shown. On the right moving pixels are shown as white.

3.3. *Sub-operations*

There are some sub-operations that should be applied to outputs of the foreground segmentation process since there is noise and shadow problem. A shadow removal algorithm and morphological operations are used to remove shadow and noise.

3.3.1. Shadow Removal

Shadow is a problematic case in moving object segmentation problem. Since shadows cause considerable intensity change, the parts of the scene that is affected from shadows are taken as foreground. Thus shadow removal is a crucial part of a visual surveillance system. Due to its simplicity and computational efficiency the algorithm proposed in [18] is applied. The main idea of algorithm is that the shadow changes the intensity of background but normalized intensity values of shadow pixels are approximately the same as background.

$$\begin{aligned}\frac{R_s}{R_s + G_s + B_s} &\cong \frac{R}{R + G + B} \\ \frac{G_s}{R_s + G_s + B_s} &\cong \frac{G}{R + G + B} \\ \frac{B_s}{R_s + G_s + B_s} &\cong \frac{B}{R + G + B}\end{aligned}\tag{3-11}$$

$$I_s(x,y) = \alpha I(x,y)\tag{3-12}$$

where $I(x,y)$ is the intensity value at point (x,y) and subscript “s” denotes the value after shadow. The foreground pixels, having intensity values different from the background, but normalized color values that are close to background values, are labeled as shadow region. Below, result of shadow removal on PETS-2001 video sequence is presented.

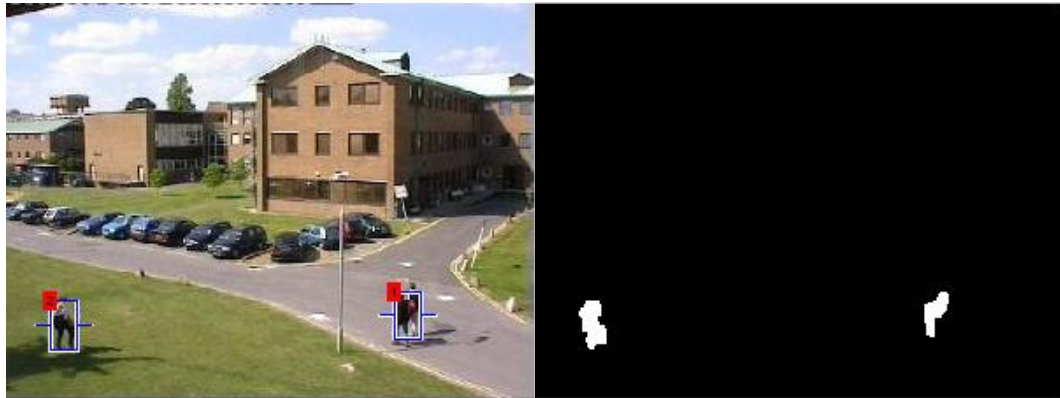


Figure 3.3 Result of shadow removal on PETS-2001 data set-3 camera-1. On the left tracking result is presented. On the right moving pixels are shown as white.

3.3.2. Morphological Operations

Morphological operations are used to remove the noise in the image which is obtained by moving object segmentation process. To remove the noise erosion and dilation operations are used.

After the motion segmentation process we have a foreground map consists of 1's and 0's. A morphological operation is a matrix which is shifted over the image and at each pixel of the image its elements are compared with the ones on the image. If the two sets match the condition defined by the set operator, the pixel underneath the origin of the structuring element is set to a pre-defined value. Erosion and dilation are two fundamental morphological operations and applied to binary image which is obtained by moving object segmentation and shadow removal.

3.3.2.1. Erosion

The basic effect of the erosion operator is to erode the small noise from the foreground pixel map. It also erodes boundaries of the regions for the foreground pixels. Thus dilation operation should be followed after erosion. A structuring element which has been utilized for this purpose is shown in (3-13). Each foreground pixel in the input image is aligned with the center of the structuring element.

$$SE_{erosion} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad 3-13$$

The effect of this operation is to remove any foreground pixel that is not completely surrounded by other white pixels as shown in Figure 3.4. As a result foreground regions shrink and holes inside a region grow.

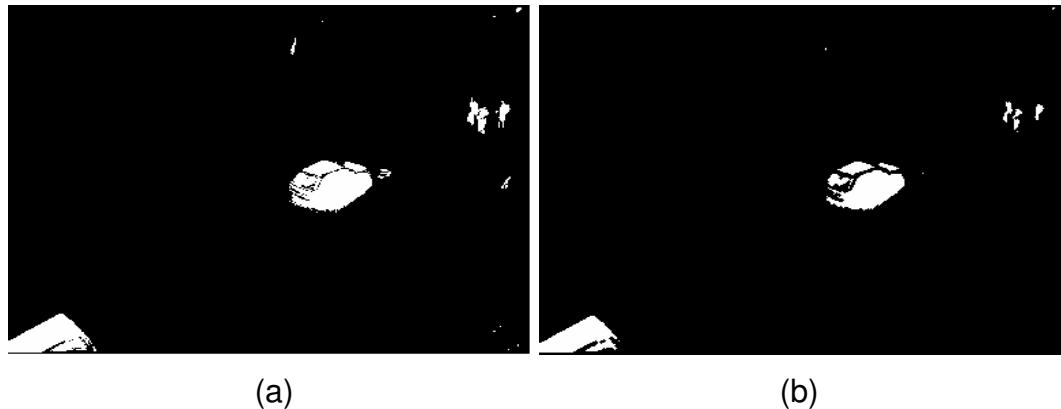


Figure 3.4 Effect of erosion operation, (a) Output image of background subtraction process, (b) Eroded image.

3.3.2.2. Dilation

Dilation is the dual operation of erosion. Foreground image borders are eroded due to erosion. Dilation operation should be applied after erosion in order to have the original image border. Since first erosion is applied, the noise that was removed by erosion operation will not be back after the dilation operation. A sample structuring element which has been utilized is given in (3.14).

$$SE_{dilation} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

3-14



Figure 3.5 Effect of dilation operation, (a) Eroded image, (b) Dilated image (resulting image of morphological operations).

3.3.3. Connected Component Labeling

In order to find moving objects separately we should apply connected component labeling. Here is a sample foreground matrix:

```

BW = [1  1  1  0  0  0  0  0
      1  1  1  0  1  1  0  0
      1  1  1  0  1  1  0  0
      1  1  1  0  0  0  1  0
      1  1  1  0  0  0  1  0
      1  1  1  0  0  0  1  0
      1  1  1  0  0  1  1  0
      1  1  1  0  0  0  0  0];

```

Result of four connected component labeling:

```

L =  [1  1  1  0  0  0  0  0
      1  1  1  0  2  2  0  0
      1  1  1  0  2  2  0  0
      1  1  1  0  0  0  3  0
      1  1  1  0  0  0  3  0
      1  1  1  0  0  0  3  0
      1  1  1  0  0  3  3  0
      1  1  1  0  0  0  0  0]

```

CHAPTER 4

MOVING OBJECT TRACKING

The objective of tracking is to establish correspondence of objects and object parts between consecutive frames of video. These correspondences are established by finding the best match according to some similarity function calculated with the previous state of the objects and the new measurements. As surveillance systems became popular, moving object tracking became an important research topic. The increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms. Most important part of the moving object tracking is associating the measurements with the targets. For a robust tracking system this association should be done correctly. In this thesis, we deal with the association problem: doing the association correctly and fast. There are several methods that have been used to track moving objects. Kalman tracker is the most popular tracker, and we will concentrate on this method.

Two methods have been used to track moving objects: Kalman tracker, Particle Filter Tracker. These methods have been evaluated using PETS2001 [8] and PETS2000 [37] data sets.

4.1. Kalman Tracker

Kalman filtering is a popular approach in tracking to estimate the next state of moving objects. In this work, standard Kalman filter is used in the concept of Interacting Multiple Model Filtering approach. Instead of using a single motion model, Interacting Multiple Model Filtering approach uses multiple motion models. According to the error calculations between the estimated state and the measurement state, the motion model which is in action is determined.

4.1.1. Kalman Filter

In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem [14]. Since that time the Kalman filter has been the subject of extensive research and application, particularly in the area automated surveillance systems.

Kalman filter is a set of equations that provides an efficient computational means to estimate the state of a process by minimizing the mean of the squared error. The filter supports estimations of past, present, and future states.

The Kalman filter addresses the general problem of trying to estimate the state $x \in R^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad 4-1$$

with a measurement defined by the equation

$$z_k = Hx_k + v_k$$

4-2

The random variables w_k and v_k represent the process and measurement noise, respectively. They are assumed to be independent of each other, white, and with normal probability distributions.

$$p(w) \sim N(0, Q)$$

$$p(v) \sim N(0, R)$$

In practice, the process noise covariance Q and measurement noise covariance R matrices might change with each time step or measurement; however they are assumed to be constant in our work.

The next state of a process is estimated by a Kalman filter by the scheme given in Figure 4.1. The time update projects the current state estimate ahead in time. The measurement update adjusts the projected estimate by an actual measurement at that time.

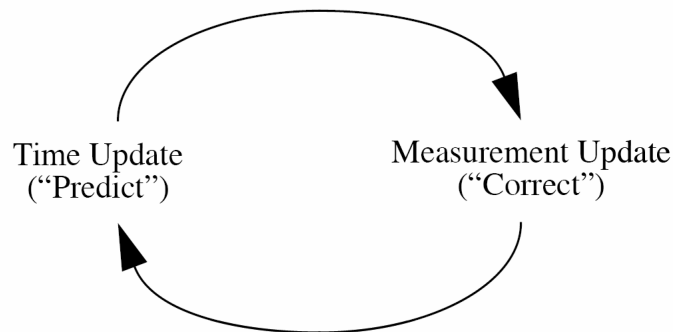


Figure 4.1 The Kalman filter cycle.

For time-update and measurements update steps of Kalman filter, there are equations by which a priori ($\hat{x}_k^-$) and posteriori ($\hat{x}_k$) estimates are determined.

Time update equations (predict):

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad 4-3$$

$$P_k^- = AP_{k-1}A^T + Q \quad 4-4$$

Measurement update (correct):

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad 4-5$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \quad 4-6$$

$$P_k = (I - K_k H)P_k^- \quad 4-7$$

4.1.2. Moving Object Tracking via Kalman Filter

After detection of moving object, Kalman tracker is used as point tracker. Center position of the moving object has been used as core point to be estimated. The state vector of Kalman filter has been defined as x and y positions and displacements in x and y directions per unit time interval.

$$X_k = \begin{pmatrix} x_c & y_c & v_x & v_y \end{pmatrix} \quad 4-8$$

The Kalman filtering algorithm estimates the state vector based on a measurement errors. State model has been assumed as linear and defined by

$$X_k = AX_{k-1} + \omega_{k-1} \quad 4-9$$

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 4-10$$

where A is the transition matrix and ω is the estimation error vector. We also assume that a linear relationship between state vector and measurements.

$$Z_k = HX_k + v_k \quad 4-11$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad 4-12$$

Where H is observation matrix and v is the measurement error. Using these values of transition and observation matrix, time-update and measurement-update equations have given the estimate of next state of the object.

4.1.3. Interacting Multiple Model Filtering Approach

Early techniques for target tracking used single motion model based approaches, which do not give good results when the target changes its motion model while tracking or the motion model of the filter does not

match the target's motion model. Advantage of using multiple model based tracking is that a target that moves according to various motion models can be tracked reliably. Motion model switching capability of multiple model framework tracks targets much more accurately than single motion model based trackers. General form of the multiple model filters assume that there is no motion model switching during the estimation process. Interacting Multiple Model algorithm can switch between motion models automatically according to a Markov chain process [33]. IMM algorithm can be used in various applications such as radar tracking, air craft tracking, and in our project we are using it for visual tracking.

Normally, to track a target we can use a single Kalman filter. For successful tracking the correct motion model should be presented to the Kalman filter, but this cannot be known in real time applications. The potential weakness of a single motion model based trackers is single motion model can lead to under-modeling and over-modeling. Under-modeling occurs when the Kalman filter state model does not describe the actual state adequately. It is possible to construct a Kalman filter that adequately describes any type of motion. This solves the under-modeling problem, but such kind of Kalman filter will have a large number of states and have complex state model. Thus although it is possible to construct such a Kalman filter, it will not be practical in noisy environment. High-order Kalman filters containing states that are derivatives of derivatives are very sensitive to noise. Also it is very difficult to handle such complex Kalman filters compared to much simpler ones.

To overcome the above problems one solution is to use a number of filters based on the possible expected motion models and then combine the outputs of the filters according to the probability of the filter to have the

correct motion model at that time. In IMM approach at time k the state estimate is computed under each possible current model using r (in our case $r = 3$ since we use 3 motion models) filters, with each filter using a different combination of the previous model conditional estimates. The figure below shows one cycle of the IMM algorithm.

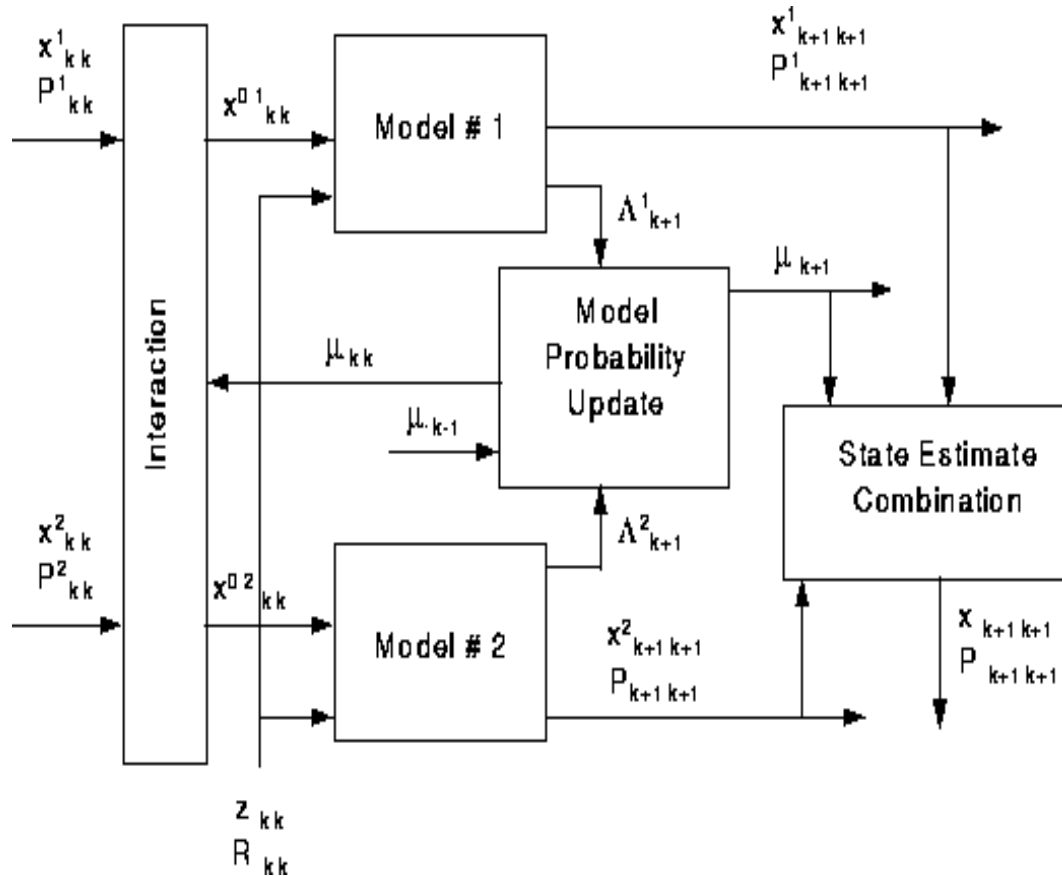


Figure 4.2 One cycle of IMM algorithm.

One cycle of the IMM algorithm consist of five steps.

4.1.3.1. Calculation of the Mixing Probabilities

The probability that mode M_i is in effect at k conditioned on Z^{k-1} measurements is given by

$$M_{i,j}(k-1|k-1) = \frac{1}{\bar{c}_j} P_{ij} M_i(k-1), \quad i, j = 1, \dots, r, \quad 4-13$$

where the mode transition probability P_{ij} which is known or decided initially and the normalizing constants are

$$\bar{c}_j = \sum_{i=1}^r P_{ij} M_i(k-1), \quad j = 1, \dots, r, \quad 4-14$$

4.1.3.2. Mixing

Starting with $\hat{x}^i(k-1|k-1)$ one computes the mixed initial condition for the filter matched to $M_j(k)$ is

$$\hat{x}^{oj}(k-1|k-1) = \sum_{i=1}^r \hat{x}^i(k-1|k-1) M_{i|j}(k-1|k-1), \quad j = 1, \dots, r \quad 4-15$$

The covariance corresponding to the above is

$$P^{oj}(k-1|k-1) = \sum_{i=1}^r M_{i|j}(k-1|k-1) \left\{ \begin{array}{l} P^i(k-1|k-1) + [\hat{x}^i(k-1|k-1) - \hat{x}^{oj}(k-1|k-1)]^* \\ [\hat{x}^i(k-1|k-1) - \hat{x}^{oj}(k-1|k-1)]^T \end{array} \right\} \quad j = 1, \dots, r \quad 4-16$$

4.1.3.3. Mode-matched Filtering

The likelihood functions corresponding to the r filters are given by

$$\Lambda_j(k) = N\left[x(k); \hat{z}^j[k|k-1], \hat{x}^{oj}(k-1|k-1), S^j[k; P^{oj}(k-1|k-1)]\right],$$

$$j = 1, \dots, r \quad 4-17$$

4.1.3.4. Mode Probability Update

This is given by

$$M_j(k) = \frac{1}{c} \Lambda_j(k) \bar{c}_j, \quad j = 1, \dots, r \quad 4-18$$

where c_j is the normalizing constant used in calculation of mixing probabilities, and $c = \sum_{i=1}^r \Lambda_i(k) \bar{c}_i$ is the normalizing constant for the equation above.

4.1.3.5. Estimate and Covariance Combination

Combination of the model conditioned estimates and covariance is done according to the mixture equations

$$\hat{x}(k|k) = \sum_{j=1}^r \hat{x}^j(k|k) M_j(k), \quad 4-19$$

$$P(k|k) = \sum_{j=1}^r M_j(k|k) \left\{ P^j(k|k) + [\hat{x}^j(k|k) - \hat{x}(k|k)] [\hat{x}^j(k|k) - \hat{x}(k|k)]^T \right\} \quad 4-20$$

4.2. Active Contours (Snakes)

Active contours have been used in several applications in computer vision such as region segmentation, contour extraction [3]. In our work snakes are used to extract exact shape of the target. This information is not used in this study, but as a future work, the use of exact shapes obtained by snakes can be useful.

4.3. Particle Filter Tracking

In many application areas it is becoming important to include nonlinearity and non-Gaussianity in order to model dynamics of physical systems accurately [34]. In target tracking the main objective of the particle filtering is to track a target as it evolves over time with a non-Gaussian and potentially multi-modal pdf. Particle filters are sequential Monte Carlo methods based on point mass representations of probability densities. The basis of particle filtering is to construct a sample-based representation of the pdf of the target. Particles from the target are associated with weights that represent the quality of that particle. Target state estimate is represented by a weighted sum of all particles. Particle filtering is recursive and has two main phases: prediction and update. In prediction phase each particle is updated according to the existing model and random noise is included in order to represent the noise on the particle. In update phase each particle's weight is re-evaluated based on the latest sensor information. Then particles that have very small weights are eliminated.

Kalman filter assumes that the posterior density is Gaussian and, thus, the density is parameterized by a mean and covariance. If these assumptions hold Kalman filter is the optimal filter for the solution of the tracking

problems. Generally densities are non-Gaussian and there should be some other solution for these cases. One of the solutions for these cases is the Extended Kalman Filter approach, which approximates the non-Gaussian densities by a Gaussian. Unscented Kalman filter is obtained from EKF framework by unscented transform. In UKF, a set of points are deterministically selected from the Gaussian approximation and propagated through the true nonlinearity. Then the parameters of the Gaussian approximation are re-estimated. Since UKF approximates the nonlinearity better than EKF, generally it gives better performance. But EKF always approximates $p(X_k | Z_{1:k})$ to be Gaussian, and if the true density is non-Gaussian than EKF can never describe the dynamics of the system accurately.

For solution of the above problems Particle Filtering is used which is a technique for implementing recursive Bayesian filter by Monte Carlo simulations. The key idea of the Particle Filtering is to represent the required posterior density function by a set of random samples with associated weights. Then, estimates are computed based on these samples and weights. As the number of samples gets larger, this characterization becomes an equivalent representation to the usual posterior pdf. Sequential Importance sampling is the optimal technique for Particle Filtering. Because of high computational load this method is not chosen. Also in this study Interacting Multiple Model filtering approach is used which is capable of solving most of the problems.

4.4. Results and Discussions

In this chapter the result of the Kalman tracker can be seen. When object motion is linear Kalman tracker gives good results, but generally this not

the case. Thus we decided to use an Interacting Multiple Model Kalman Filter approach in order to track objects in a better way. In Figure 4.2 estimated (blue) and real (green) trajectory of a moving object is shown.



Figure 4.3 Kalman Tracker: Estimated and real trajectories of a moving object.

CHAPTER 5

RULE-BASED TRACKING

The objective of tracking is to establish a correspondence of objects and object parts between consecutive frames of video. These correspondences are established by finding the best match according to some similarity function calculated with the previous state of the objects and the new measurements. In this thesis a rule based tracker which is based on fuzzy logic is first developed for a single camera then extended to multiple cameras.

For real-time tracking, rule based methodology is preferred since computational load is very low compared to the probabilistic data association techniques and Multiple Hypothesis Tracking Technique (in MHT memory load is also very high).

In multi-target tracking the tough part of data association is the case of closely spaced targets. For discrimination of the closely spaced targets the use of additional information gathered from the sensors is important. Target shape, dimensions and intensity are the features that can be used for solving image based data association problem. Fuzzy logic presents a systematic way of combining this information.

5.1. Correlation Calculation Using Fuzzy Logic

This part of the visual surveillance system is called Fuzzy Data Association Engine (FDAE) which is used to calculate the observation-to-track correlation value that is used to solve the data association problem. In order to calculate the observation-to-track correlation value Euclidean distance and area ratio features are used. Euclidean distance is the distance between the target's predicted center of area and the observation's center of area. Area ratio is ratio of target's predicted area to measurement's area. These are the inputs of Fuzzy Data Association Engine, and the observation-to-track correlation value is the output. After having correlation values between all the observation-track pairings, observation-to-track correlation matrix is formed. Below a sample observation-to-track correlation matrix is given.

Table 5-1 Sample Observation-to-Track Correlation Matrix

O\T	T-1	T-2	T-3	T-4	T-5
O-1	0.2	0.85	0	0.1	0
O-2	0.4	0	0.7	0	0
O-3	0.9	0.35	0	0.2	0
O-4	0.1	0.2	0.2	0.75	0
O-5	0.35	0	0.1	0	0

Observation-to-track correlation matrix does not directly solve the observation-to-track association problem. It only shows which track is how much correlated to which observation.

Fuzzy data association engine is composed of four sub-blocks which are Fuzzification Interface, Fuzzy Knowledge Base, Fuzzy Inference Engine and Defuzzification Interface.

5.1.1. Fuzzification Interface

Fuzzification Interface transforms each numerical data received from the sensor measurements into fuzzy variables. In this case the numerical data is ratio of target's predicted area to observation's area, and the Euclidean distance between the target's predicted center of area (output of the Kalman filter) and the observation's center of area.

Euclidean distance between the predicted position of the target and the measurement position is calculated, and this value is transformed into fuzzy variables: (Exact Position (EP), Very Very Close (VVC), Very Close (VC), Close (C), Far (F), Very Far (VF), Very Very Far (VVF)). The maximum Euclidean distance of a target in consecutive frames can be 400 pixels length (frame size is 320x240). According to maximum distance, the Euclidean distance between the target's predicted center of area and the observation's center of area is fuzzified. Fuzzification of Euclidean distance is presented in Figure 5.1.

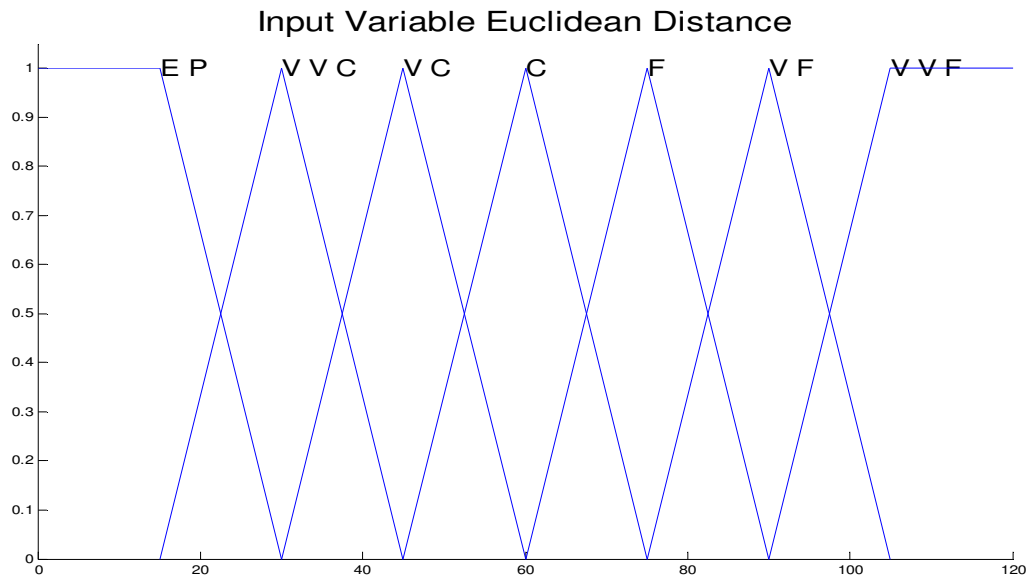


Figure 5.1 Fuzzification of input variable distance.

The ratio of target's predicted area to the observation's area is calculated and transformed into fuzzy variables: (Very Very Small Ratio (VVSR), Very Small Ratio (VSR), Small Ratio (SR), Exact Ratio (ER), High Ration (HR), Very High Ratio (VHR), Very Very High Ratio (VVHR)). As objects get closer to the camera or get away from the camera, their areas on the frames get larger or smaller. Because of this fact we are using the predicted target area instead of the target's exact area in previous frame. In Figure 5.2 fuzzification of area ratio is presented.

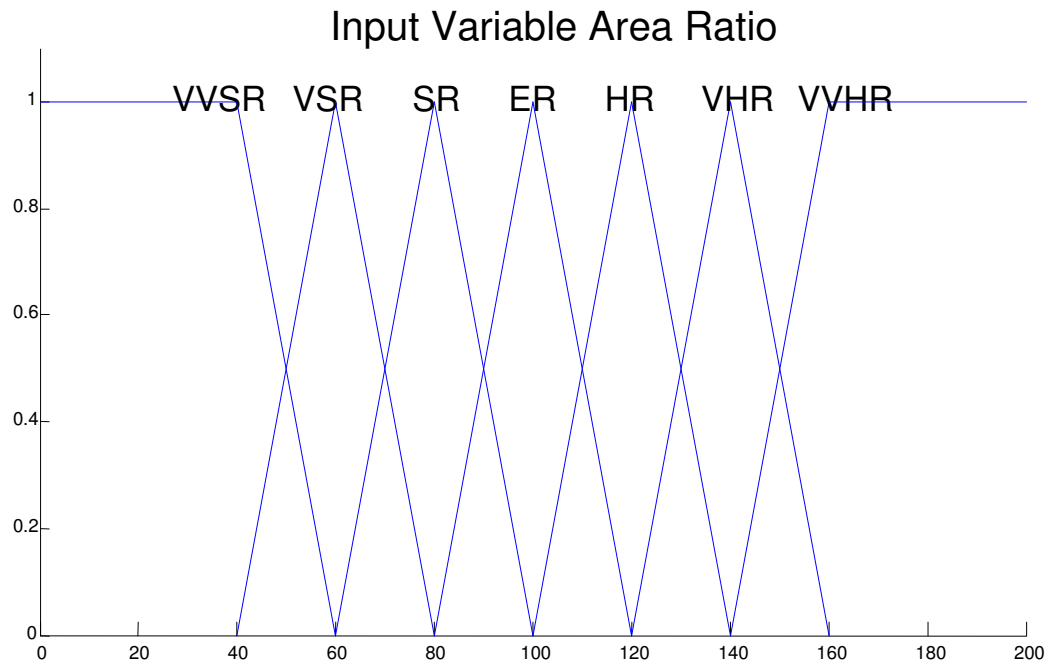


Figure 5.2 Fuzzification of input variable area ratio.

5.1.2. Fuzzy Knowledge Base

Fuzzy Knowledge Base contains the IF-THEN rules. Fuzzy Inference Engine employs particular kind of fuzzy logic. It stimulates the human decision making procedure, and employs Fuzzy Knowledge Base and fuzzy input to generate fuzzy decisions. In order to get the IF-THEN rules, fuzzified inputs are combined so that they give the degree of observation to track correlation value.

Table 5-2 Generation of IF-THEN Rules

A.R.\E.D.	EP	VVC	VC	C	F	VF	VVF
VVSR	MA	SA	VSA	NA	NA	NA	NA
VSR	HA	MA	SA	VSA	NA	NA	NA
SR	VHA	HA	MA	SA	VSA	NA	NA
ER	FA	VHA	HA	MA	SA	VSA	NA
HR	VHA	HA	MA	SA	VSA	NA	NA
VHR	HA	MA	SA	VSA	NA	NA	NA
VVHR	MA	SA	VSA	NA	NA	NA	NA

In the table, NA represents No Association, VSA represents Very Small Association, SA represents Small Association, MA represents Medium Association, HA represents High Association, VHA represents Very High Association, and FA represents Full Association.

From the combination of fuzzified inputs following IF-THEN rules are obtained:

IF Distance is EP and Ratio is VVSR THEN Association if MA
 IF Distance is EP and Ratio is VSR THEN Association if HA
 IF Distance is EP and Ratio is SR THEN Association if VHA
 IF Distance is EP and Ratio is ER THEN Association if FA
 IF Distance is EP and Ratio is HR THEN Association if VHA
 IF Distance is EP and Ratio is VHR THEN Association if HA
 IF Distance is EP and Ratio is VVHR THEN Association if MA

IF Distance is VVC and Ratio is VVSR THEN Association if SA
IF Distance is VVC and Ratio is VSR THEN Association if MA
IF Distance is VVC and Ratio is SR THEN Association if HA
IF Distance is VVC and Ratio is ER THEN Association if VHA
IF Distance is VVC and Ratio is HR THEN Association if HA
IF Distance is VVC and Ratio is VHR THEN Association if MA
IF Distance is VVC and Ratio is VVHR THEN Association if SA

IF Distance is VC and Ratio is VVSR THEN Association if VSA
IF Distance is VC and Ratio is VSR THEN Association if SA
IF Distance is VC and Ratio is SR THEN Association if MA
IF Distance is VC and Ratio is ER THEN Association if HA
IF Distance is VC and Ratio is HR THEN Association if MA
IF Distance is VC and Ratio is VHR THEN Association if SA
IF Distance is VC and Ratio is VVHR THEN Association if VSA

IF Distance is C and Ratio is VVSR THEN Association if NA
IF Distance is C and Ratio is VSR THEN Association if VSA
IF Distance is C and Ratio is SR THEN Association if SA
IF Distance is C and Ratio is ER THEN Association if MA
IF Distance is C and Ratio is HR THEN Association if SA
IF Distance is C and Ratio is VHR THEN Association if VSA
IF Distance is C and Ratio is VVHR THEN Association if NA

IF Distance is F and Ratio is VVSR THEN Association if NA
IF Distance is F and Ratio is VSR THEN Association if NA
IF Distance is F and Ratio is SR THEN Association if VSA
IF Distance is F and Ratio is ER THEN Association if SA
IF Distance is F and Ratio is HR THEN Association if VSA

IF Distance is F and Ratio is VHR THEN Association if NA
IF Distance is F and Ratio is VVHR THEN Association if NA

IF Distance is VF and Ratio is VVSR THEN Association if NA
IF Distance is VF and Ratio is VSR THEN Association if NA
IF Distance is VF and Ratio is SR THEN Association if NA
IF Distance is VF and Ratio is ER THEN Association if VSA
IF Distance is VF and Ratio is HR THEN Association if NA
IF Distance is VF and Ratio is VHR THEN Association if NA
IF Distance is VF and Ratio is VVHR THEN Association if NA

IF Distance is VVF and Ratio is VVSR THEN Association if NA
IF Distance is VVF and Ratio is VSR THEN Association if NA
IF Distance is VVF and Ratio is SR THEN Association if NA
IF Distance is VVF and Ratio is ER THEN Association if NA
IF Distance is VVF and Ratio is HR THEN Association if NA
IF Distance is VVF and Ratio is VHR THEN Association if NA
IF Distance is VVF and Ratio is VVHR THEN Association if NA

The output of the Fuzzy Data Association Engine is the observation to track correlation value. In Figure 5.3, output membership is presented.

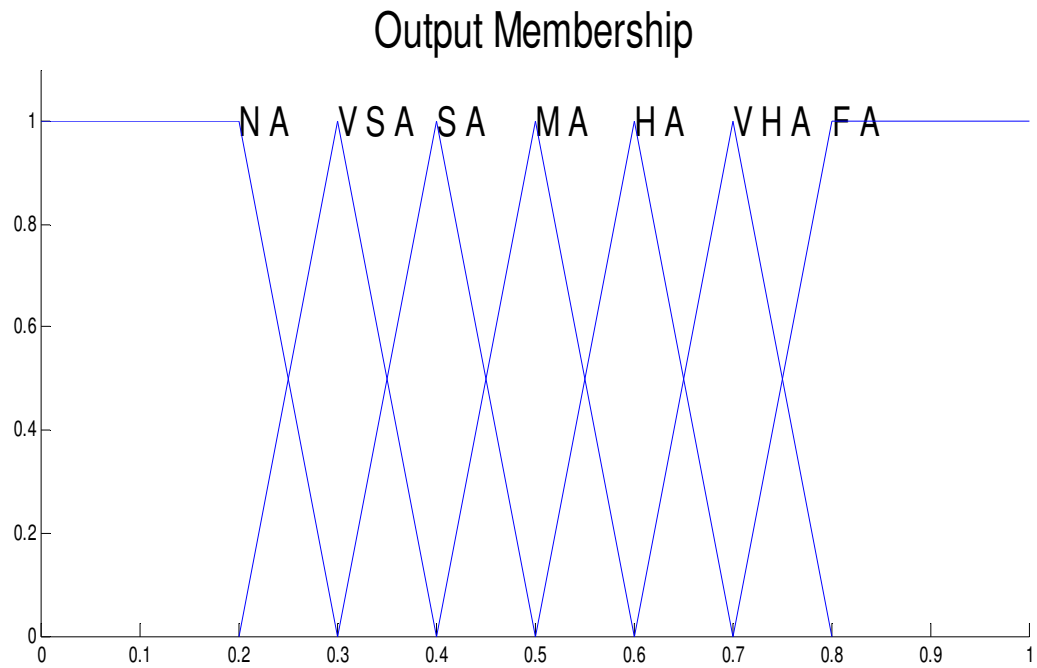


Figure 5.3 Output membership.

5.1.3. Defuzzification Interface

Defuzzification Interface transforms the fuzzy output into the numerical data for use. In this study centroid is selected as defuzzification method. The output of the defuzzification step is the correlation value. Track initiation, track confirmation and track update processes are done according to this correlation value.

5.2. Single Camera Tracking

In this part of the thesis, objects that are detected by cameras are tracked independently. Features that are obtained from single camera are used. No additional information from other camera is used. In the tracking part

we define track strength as the probability of a track being a real track. We will use track strength throughout the study.

In order to solve the observation-to-track association problem the observation-to-track correlation matrix, which is the output of the Fuzzy Data Association Engine will be used. Observation-to-track correlation matrix does not directly solve the observation-to-track association problem. It only shows which track is how much correlated to which observation. The job is to find the correct observation-track pairings. Here, additional rules are used to solve the problem. These rules are

- 1) A track can be associated with one observation. Observation-to-track correlation value that is calculated by Fuzzy Data Association Engine affects the track strength.
 - a) Full Association, Very High Association and High Association increases the track strength,
 - b) Medium Association does not affect the track strength,
 - c) Small Association and Very Small Association decreases the track strength,
- 2) An observation can be associated more than one track (occlusion case). Occluded tracks' strength is not affected.
- 3) If tracks are occluded with each other (merged), a new track is formed. This newly formed track is tracked until the occluded tracks split. When the occlusion finishes the track that is formed by occlusion is deleted.
- 4) When the occlusion finishes the occluded tracks are matched with observations. End of occlusion is declared if the threshold added area of the track that is formed by occlusion contains more than one observation. During the occlusion, occluded tracks may have

changed the direction of move and velocity. Thus the matching after the occlusion is not reliable. The track strengths are not affected by this matching procedure.

- 5) Observations that are not associated with existing tracks and track candidates initiate new track candidates.
- 6) A track candidate becomes a tracks if it seen in five consecutive frames. Every appearance of the track candidate increases its strength. After five iterations, track strength is above the track threshold. Thus the candidate becomes a real track.
- 7) If the track candidate is missed in even one frame, it is deleted.
- 8) The strengths of the tracks that are not associated with observations are decreased. These tracks are tracked in the background until there is an observation-to-track association.
 - a) At every iteration, it is checked if there is an observation-to-track-association. If not, the track strengths are decreased.
 - b) If the predicted position of the track is out of the field of view of the camera, then the track strength decreases more.
- 9) The tracks that have strengths below the threshold are deleted.

When an observation is correlated with more than one track then these tracks are assumed to be occluded. This is the case like presented in Figure 5.5. In occlusion case an occlusion matrix is calculated using the predicted positions of the tracks and track's bounding boxes. If the bounding boxes overlap or their distance is lower than a threshold, the corresponding cell of the occlusion matrix is made 1. After this calculation we have a matrix like the one given in Table 5-3.

Table 5-3 Sample Occlusion Matrix

	T-1	T-2	T-3	T-4	T-5
T-1	0	1	0	0	0
T-2	1	0	0	0	0
T-3	0	0	0	0	0
T-4	0	0	0	0	0
T-5	0	0	0	0	0

From this matrix we can say that tracks T-1 and T-2 may be occluded with each other. If there is only one observation that is associated with tracks T-1 and T-2 in the observation-to-track correlation matrix, it is proved that tracks T-1 and T-2 are occluded. In this case a new track is formed, and occluded tracks are tracked in the background. Every frame tracks are checked if their occlusion ended. Occlusion is declared as ended when there are more than one observation in the threshold added area of the track formed by occlusion. When the occlusion finishes, the track that is formed as a result of occlusion is deleted and the tracking of occluded tracks continued normally.

After observation-to-track association is done, with the help of observation-to-track association matrix, occlusion matrix and rules, tracks are updated with associated observations. During occlusion, the occluded tracks are updated by the predicted data by Kalman filter, since there is no observation data for the update phase.

In Single Camera Tracking, the observation-to-track association technique and the rules are used as explained above. Track initiation, track confirmation and track update procedures are sub-parts of Single Camera Tracking.

5.2.1. Track Initiation

Track initiation begins with the rule “Observations that are not associated with existing tracks and track candidates initiate new track candidates.” For this step, first observation-to-track correlation matrix is formed as shown below.

Table 5-4 Sample Observation-to-Track Matrix that is used for track initiation

O\T	T-1	T-2	T-3	T-4	T-5
O-1	0.2	0.85	0	0.1	0
O-2	0.4	0	0.7	0	0
O-3	0.9	0.35	0	0.2	0
O-4	0.1	0.2	0.2	0.75	0
O-5	0.35	0	0.1	0	0

After applying Rule-1, observation-to-track correlation matrix becomes as presented in Table 5-5;

Table 5-5 Sample Observation-to-Track Matrix, After Rule-1 is applied

O\T	T-1	T-2	T-3	T-4	T-5
O-1	0	0.85	0	0	0
O-2	0	0	0.7	0	0
O-3	0.9	0	0	0	0
O-4	0	0	0	0.75	0
O-5	0	0	0	0	0

According to the above matrix track T-1 is associated with observation O-3, track T-2 is associated with observation O-1, track T-3 is associated with observation O-2 and track T-4 is associated with observation O-4. Track T-5 is not associated with any observations since there are not any observations that are correlated with track T-5. Observation O-5 is not correlated with existing tracks. Thus O-5 initiates a new track candidate.

5.2.2. Track Confirmation

A track is confirmed (a track candidate becomes a true track) only when five consecutive correlated observations are received. A track candidate is deleted even only one non-correlated observation is received. When this object is detected again, whole procedure starts from track initiation procedure.

For track confirmation procedure an observation-to-track candidate matrix is formed. These observations are the ones that are not correlated with existing tracks.

Table 5-6 Sample Observation-to-Track Candidate Matrix

O\TC	TC-1	TC-2	TC-3	TC-4	TC-5
O-1	0.2	0.85	0	0.34	0
O-2	0	0	0.7	0.18	0
O-3	0.9	0	0.22	0	0
O-4	0	0.35	0	0.75	0

After applying Rule-1 this matrix becomes as presented below.

Table 5-7 Sample Observation-to-Track Candidate Matrix, After Rule-1 is applied

O\TC	TC-1	TC-2	TC-3	TC-4	TC-5
O-1	0	0.85	0	0	0
O-2	0	0	0.7	0	0
O-3	0.9	0	0	0	0
O-4	0	0	0	0.75	0

From the above matrix, track candidate TC-1 is associated with O-3, TC-2 is associated with O-1, TC-3 is associated with O-2, and TC-4 is associated with O-4. TC-5 is not associated with any observations. By using the output of this matrix, another matrix that shows the number of consecutive correlation of track candidates is formed.

Table 5-8 Sample Number of Consecutive Correlated Observations Matrix

TC\I.N.	k-4	k-3	k-2	k-1	k	NOCCO
TC-1	1	1	1	1	1	5
TC-2	0	0	1	1	1	3
TC-3	0	1	1	1	1	4
TC-4	1	1	1	1	1	5
TC-5	0	0	1	1	0	2

In this matrix TC stands for Track Candidate, I.N stands for iteration number, and NOCCO stands for number of consecutive correlated observations. 1s mean that, at that iteration corresponding track candidate has an associated observation. 0s coming after 1s mean that, that track does not have an associated observation at that time step. 0s coming before 1s mean that corresponding track candidate is not initiated yet. In fact it is initiated at time step when 1s first seen. For example, track candidate TC-2 is first seen at time step k-2.

As a result of this matrix, track candidates TC-1 and TC-4 is confirmed. TC-2 and TC-3 are carried to the next iteration. TC-5 is deleted since it is not associated with any observations.

5.2.3. Track Update

In the track update phase, tracks are updated with the associated observations. A track can be correlated with only one observation but an observation can be correlated with more than one observation (tracks merging, occlusion). When there is a case that an observation is correlated with more than one track, occlusion matrix is calculated. This is the case like presented in Figure 5.7 (b). Occlusion matrix is calculated using the existing tracks' predicted positions and bounding boxes. If the bounding boxes overlap or their distance is lower than a threshold, the corresponding cell of the occlusion matrix is made 1. After this calculation we have an occlusion matrix like the one below.

Table 5-9 Sample Occlusion Matrix that is used during track update

	T-1	T-2	T-3	T-4	T-5
T-1	0	1	0	0	0
T-2	1	0	0	0	0
T-3	0	0	0	0	0
T-4	0	0	0	0	0
T-5	0	0	0	0	0

From this matrix, we can say that tracks T-1 and T-2 might be occluded. We are sure of this case if we have an observation-to-track correlation matrix like in Table 5-10.

Table 5-10 Sample Observation-to-Track Correlation Matrix that shows two tracks' occlusion with each other

O\T	T-1	T-2	T-3	T-4	T-5
O-1	0.9	0.85	0	0.1	0.1
O-2	0.4	0	0.7	0	0
O-3	0.2	0.35	0	0.2	0.82
O-4	0.1	0.2	0.2	0.75	0
O-5	0.35	0	0.1	0	0.23

After applying Rule-1 this matrix becomes as presented below.

Table 5-11 Sample Observation-to-Track Correlation Matrix that shows two tracks' occlusion with each other, after rule-1 is applied

O\T	T-1	T-2	T-3	T-4	T-5
O-1	0.9	0.85	0	0	0
O-2	0	0	0.7	0	0
O-3	0	0	0	0	0.82
O-4	0	0	0	0.75	0
O-5	0	0	0	0	0

By looking at occlusion matrix and observation-to-track correlation matrix together, we become sure that tracks T-1 and T-2 are occluded since

there is only one observation that is correlated with two tracks and tracks bounding boxes overlap. Thus a new track is formed, and this track is given the name T-6 (last track name + 1). Occluded tracks T-1 and T-2 are tracked in the background. Every time step tracks T-1 and T-2 are checked if their occlusion is finished. This check is positive if there exist observations that are inside the threshold added area of the track formed by occlusion, and these observations are not correlated with existing tracks but correlated with T-1 and T-2. Thus T-6, the track formed by occlusion is deleted and normal tracking of T-1 and T-2 continues. This is the case like presented in Figure 5.7 (c). Below there is a sample observation-to-track correlation matrix that shows the case of occlusion ending.

Table 5-12 Sample Observation-to-Track Correlation Matrix that shows two tracks' occlusion ending

O\T	T-1	T-2	T-3	T-4	T-5
O-1	0.9	0.25	0.18	0	0
O-2	0	0.4	0.7	0.34	0
O-3	0.2	0	0	0	0.82
O-4	0	0	0.23	0.75	0
O-5	0.35	0.85	0	0	0

After applying Rule-1 this matrix becomes as presented in Table 5-13.

Table 5-13 Sample Observation-to-Track Correlation Matrix that shows two tracks' occlusion ending, after Rule-1 is applied

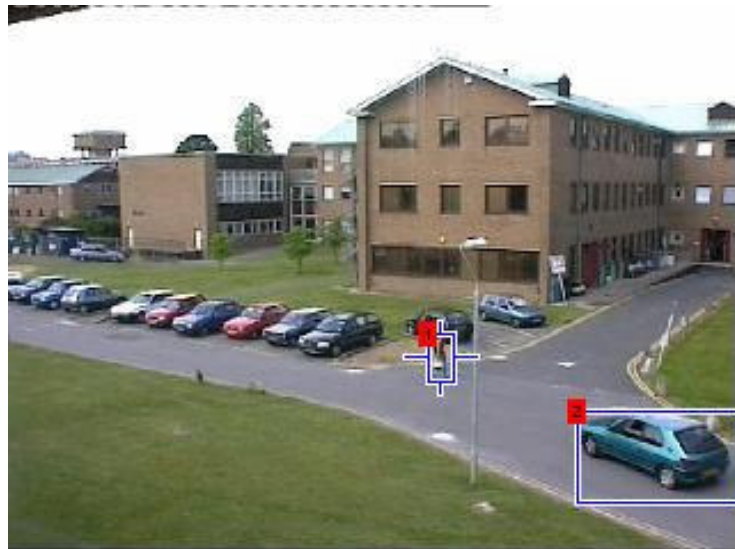
O\T	T-1	T-2	T-3	T-4	T-5
O-1	0.9	0	0	0	0
O-2	0	0	0.7	0	0
O-3	0	0	0	0	0.82
O-4	0	0	0	0.75	0
O-5	0	0.85	0	0	0

There are observations that are not associated with existing tracks and correlated with occluded tracks T-1 and T-2. Thus occlusion is ended, track T-6 that is formed by occlusion is deleted, and normal tracking procedure continues.

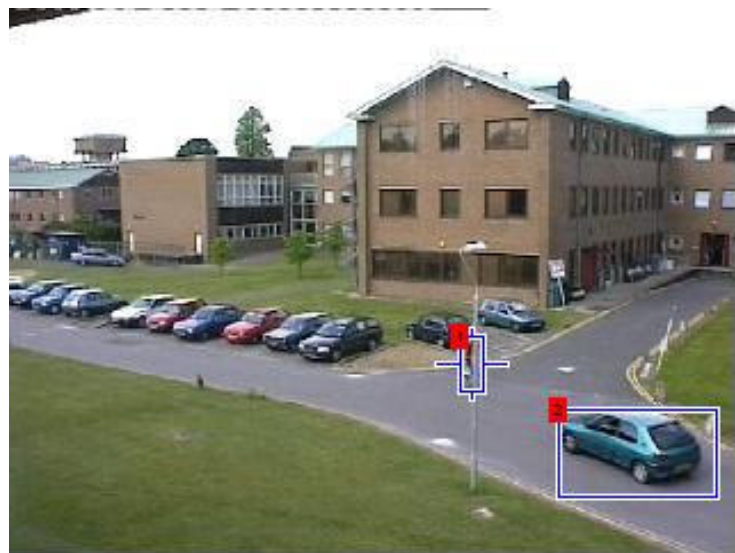
Tracks are deleted if they have strengths below the threshold. If one track is not associated with any observations then its strength is decreased and if also the predicted position of the track is not inside the field of view of the camera, then its strength is decreased more. These tracks are tracked in the background until there are observations associated with these tracks. This is the case of track's passing behind an object like a wall, a building or a car. If that track is detected again, normal tracking procedure continues.

5.2.4. Results of Singe Camera Tracking

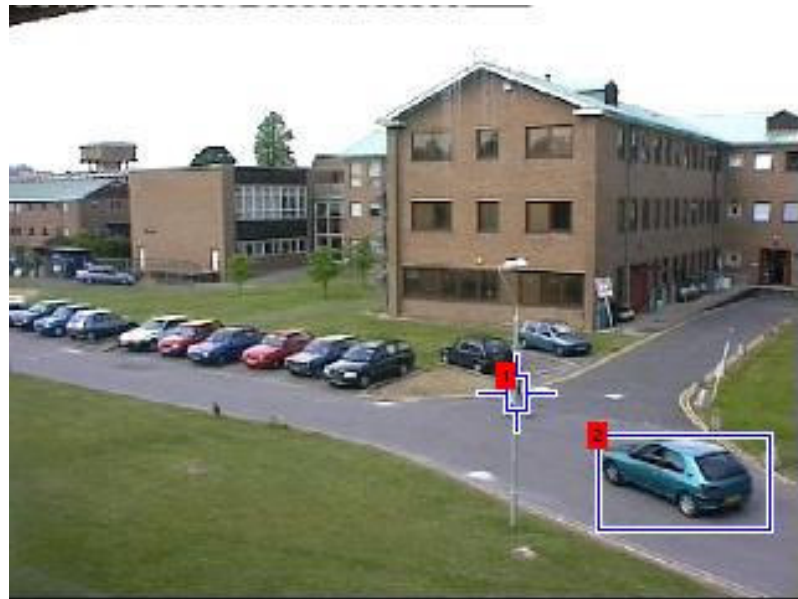
In this part the results of single camera tracking is presented. Tough cases like split and merge, occlusion, multiple target tracking are taken into account.



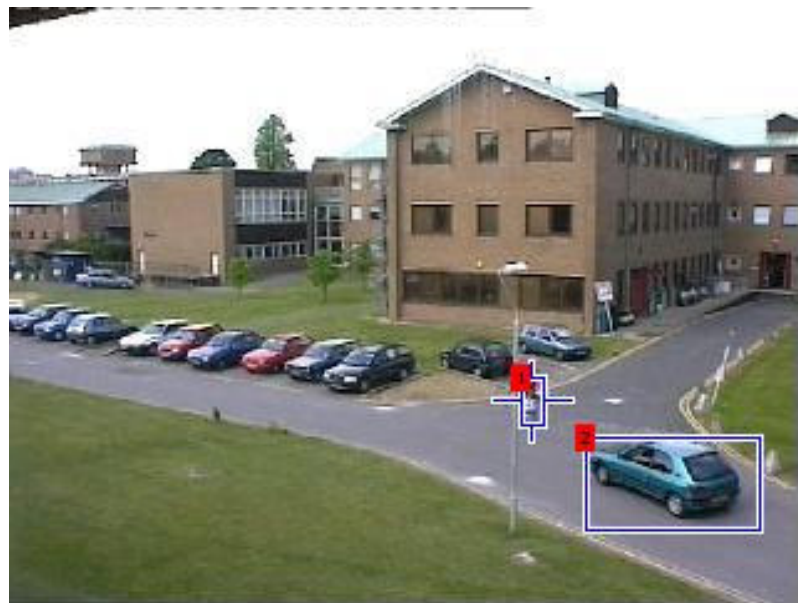
(a)



(b)

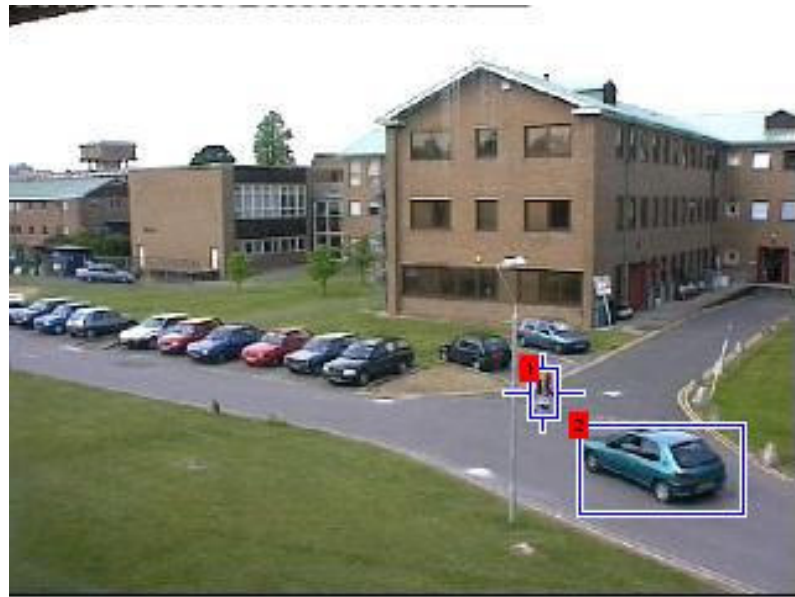


(c)

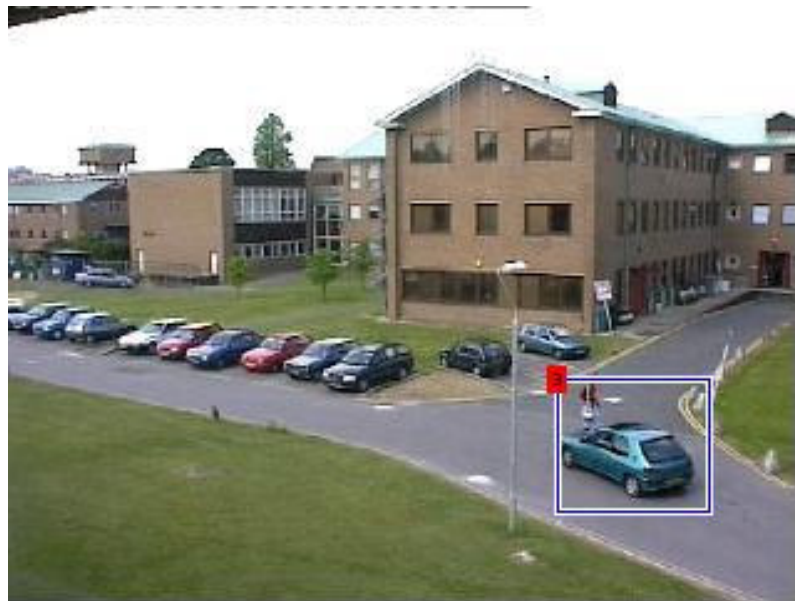


(d)

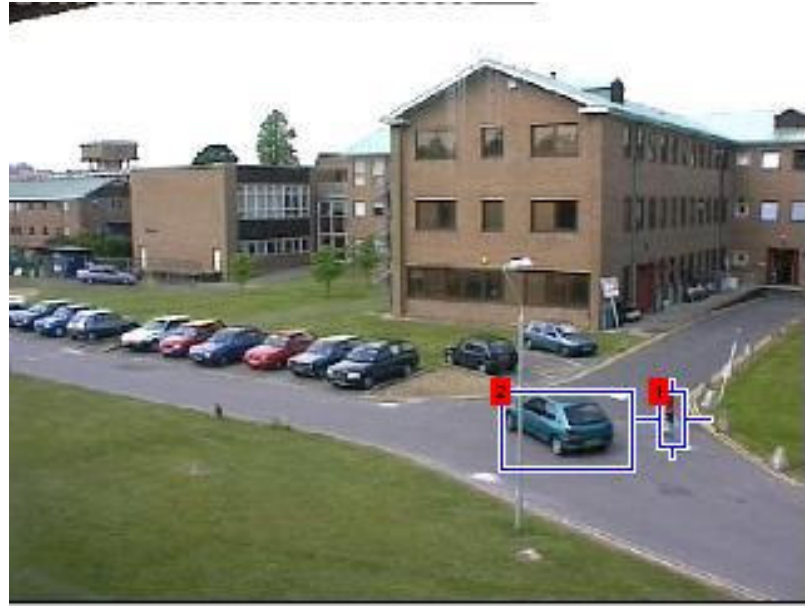
Figure 5.4 Results of rule-based tracker on PETS-2001 data set-1 camera-1, (a, b) before splitting, (c) during splitting, (d) after splitting.



(a)



(b)

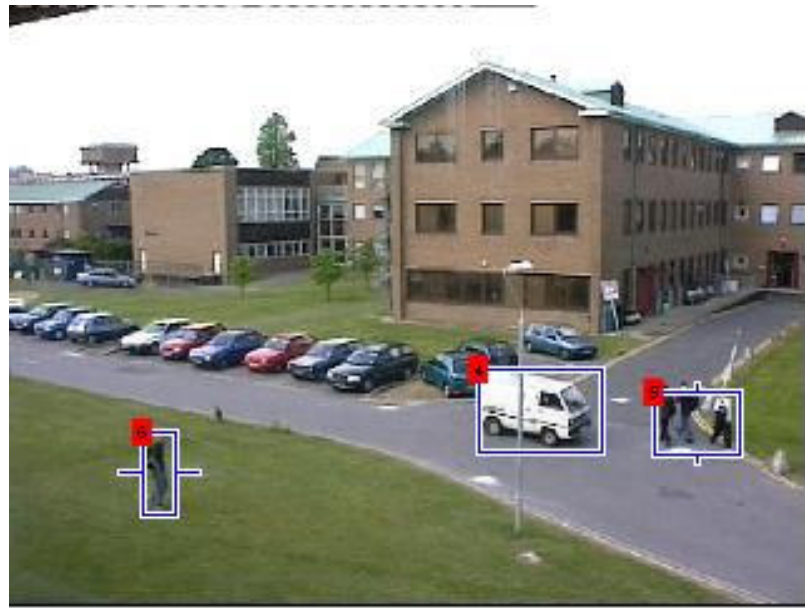


(c)

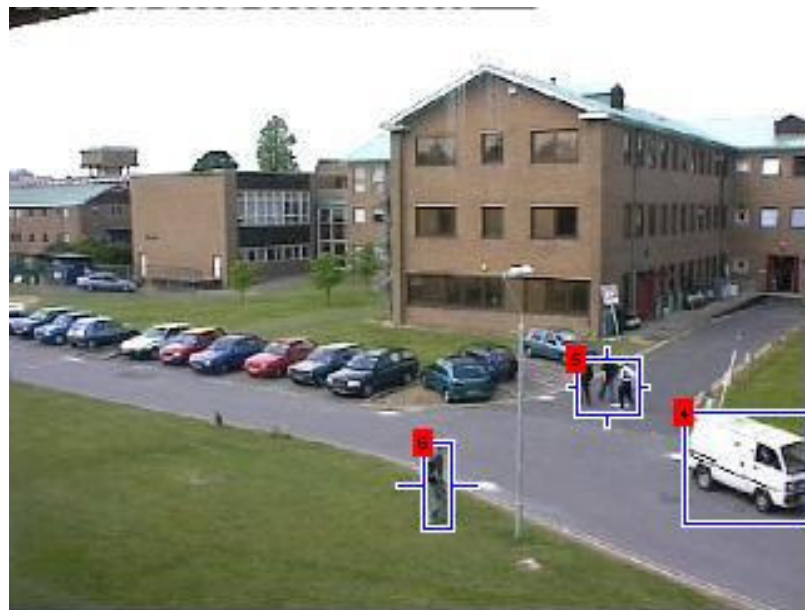
Figure 5.5 Results for occlusion case on PETS-2001 data set-1 camera-1, (a) before occlusion, (b) during occlusion, (d) after occlusion.



(a) Frame-817

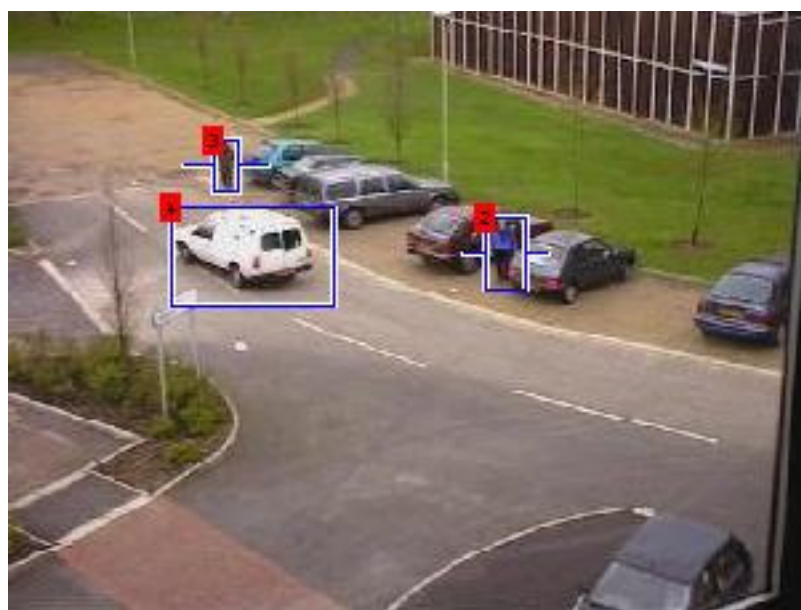


(b) Frame- 870

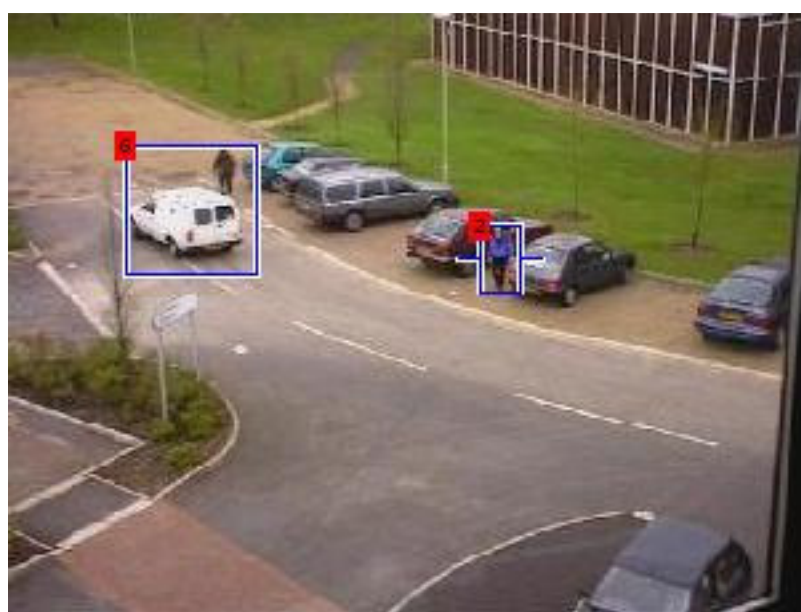


(c) Frame-960

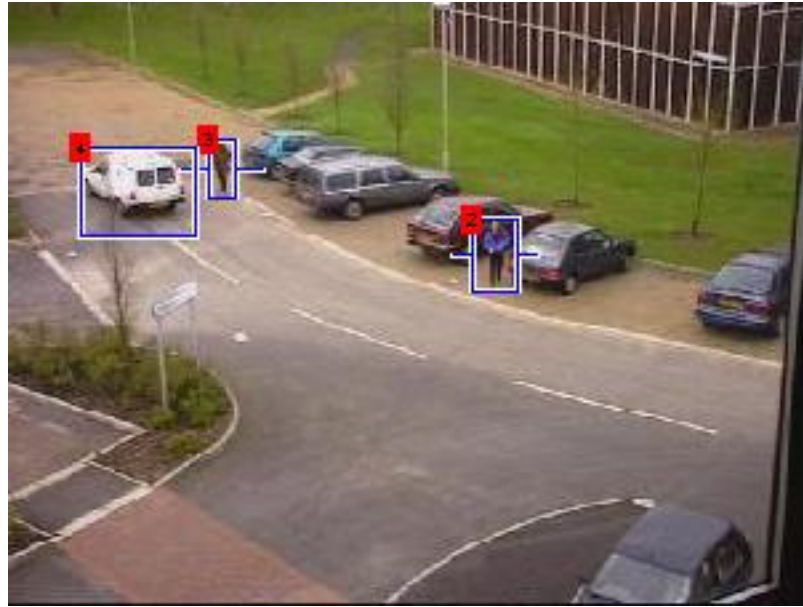
Figure 5.6 Results for Multiple Target Tracking case on PETS-2001 data set-1 camera-1.



(a) Frame-340



(b) Frame-345



(c) Frame-351

Figure 5.7 Results for occlusion case on PETS-2000 data set, (a) before occlusion, (b) during occlusion, (c) after occlusion.



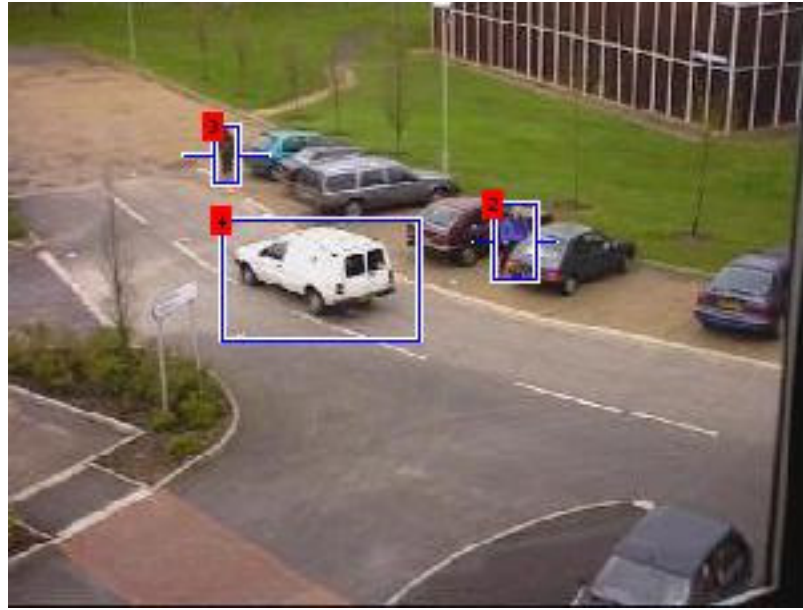
(a) Frame-195



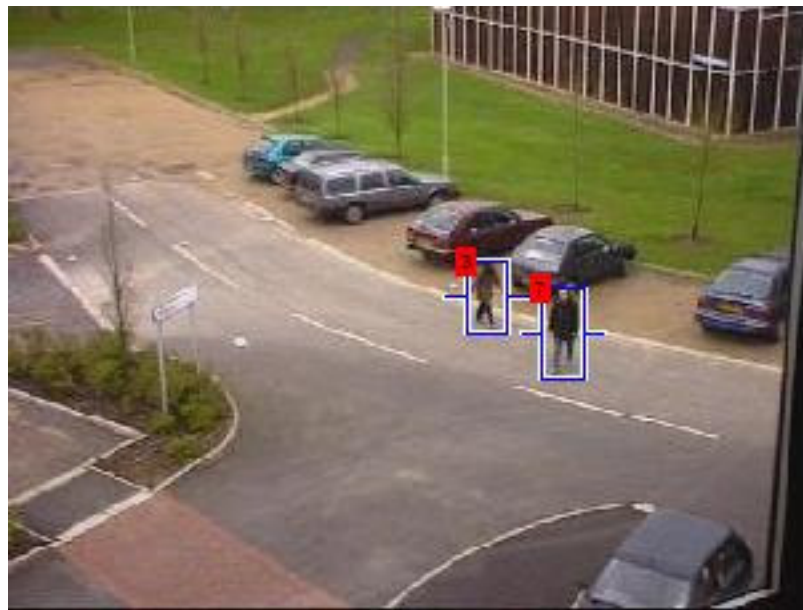
(b) Frame-271



(c) Frame-301



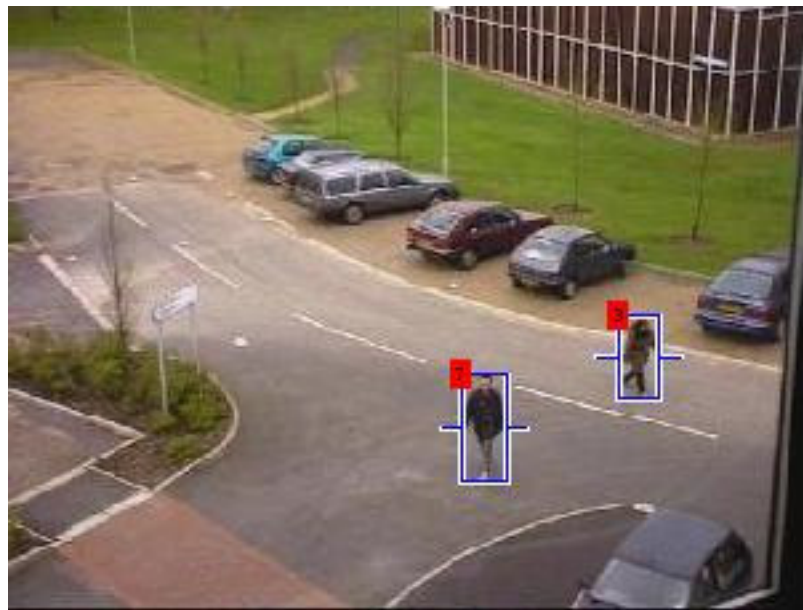
(d) Frame-353



(e) Frame-480



(f) Frame-492



(g) Frame-520

Figure 5.8 Results for Multiple Target Tracking case on PETS-2000 data set.



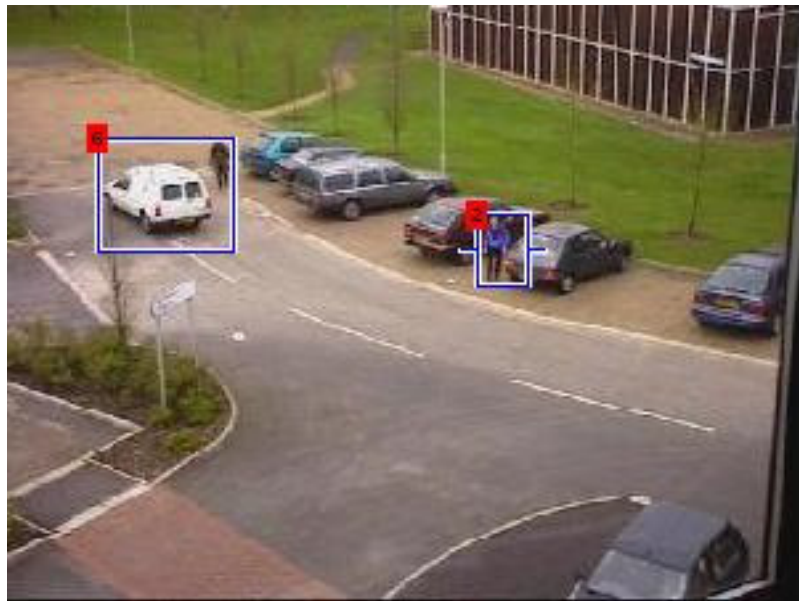
(a) Frame-272



(b) Frame-286



(c) Frame-299



(d) Frame-348

Figure 5.9 Example of a tracking failure: a person emerges from the parked car while the car being a part of the background. The identity of the car is taken by the person.

Fuzzy knowledge base and Fuzzy Inference Engine is designed to simulate the human decision mechanism. If the predicted position of the track is close to observation and area ratio is close to exact ratio then the probability of that observation belonging to that track is high. If the predicted position does not match to the observation's position or the area ratio is not close to exact ratio then probability is low. If there were some errors in fuzzy knowledge base and fuzzy inference engine, and it did not simulate the human decision mechanism the results would be faulty. If the fuzzy knowledge base is selected as below and fuzzy inference engine is operated on that then the results will be erroneous.

Table 5-14 Erroneous Fuzzy Knowledge Base

A.R.\E.D.	EP	VVC	VC	C	F	VF	VVF
VVSR	MA	SA	VSA	FA	FA	FA	NA
VSR	HA	MA	SA	VSA	NA	NA	NA
SR	VHA	HA	MA	SA	VSA	NA	NA
ER	FA	VHA	HA	MA	SA	VSA	NA
HR	VHA	HA	MA	SA	VSA	NA	NA
VHR	HA	MA	SA	VSA	NA	NA	NA
VVHR	MA	SA	VSA	FA	FA	FA	NA



Figure 5.10 Erroneous fuzzy knowledge base result on PETS-2001 dataset1 camera-2.

On the left correct Fuzzy Knowledge Base result is presented, on the right erroneous Fuzzy Knowledge Base result is presented. Because of erroneous Fuzzy Knowledge Base observation-to-track association is faulty. As explained above, if the fuzzy knowledge base has some errors and thus it does not simulate the human decision mechanism the results would have errors.



Figure 5.11 Erroneous fuzzy knowledge base result on PETS-2001 dataset1 camera-1.

On the left the track that is formed by occlusion of two tracks is shown. On the right the situation after the end of occlusion is presented. After the occlusion, the track formed by occlusion should be deleted and the tracks that are occluded should be labeled as before the occlusion. As seen on the left, the track formed by occlusion is not deleted and one of the occluded tracks is given a new label.

5.2.5. Evaluation of Singe Camera Tracking

When there are more than one track that are correlated with only one observation and the calculated occlusion matrix says that these tracks' bounding boxes overlap, occlusion is declared. In fact this case becomes true when the tracks make the expected motion. Occlusion matrix is calculated using the predicted positions of the tracks. During that time step tracks may have changed their motion type, or one of the tracks may have passed behind a stationary object. These are the possible cases other than tracks' occlusion with each other, thus in fact we not 100% sure about the occlusion.

After the occlusion, occluded tracks are matched with observations that are within the threshold added area of the track formed by occlusion. This match is made according to predicted positions and areas of the tracks. During occlusion, predicted positions and predicted areas that are predicted by Kalman filer are used in the update phase of the Kalman filter. There is no observation's data to use in the update phase during occlusion. Tracks may have changed their motion type (direction, velocity) during occlusion. Thus the match after occlusion is not reliable.

When a new object enters the field of view of the camera, and this object is close to occluded objects then it can be thought that the occlusion ended. In this case match after the end of occlusion will be definitely false. Also the new track is not initialized.

In order to solve the problems presented above the use of multiple cameras that have overlapping field of views and different angles is a good solution. In the next section this solution is presented.

5.3. Rule-Based Tracking for Multi-Camera Configuration

In this part of the thesis our main concern is to find the correspondence of tracked objects between two cameras so that the problems presented in previous section are solved. After doing the individual association and tracking job for the cameras, we should find the corresponding tracks in both cameras. In order to find corresponding tracks in different cameras we should first transform one camera view to other camera view, and then find the correspondence. For this reason 2D homography idea has been utilized.

5.3.1. 2D Homography

In the field of computer vision, a homography is defined in 2 dimensional space as a mapping between a point on a ground plane as seen from one camera, to the same point on the ground plane as seen from a second camera. Homography is a projective transformation that maps each x_i in one camera to x_i' in other camera [35]. Our problem is to compute a 3×3 matrix H such that $Hx_i = x_i'$ for each i .

We first begin with determining the number of point to be used. If the type of transformation is Euclidean transform, at least two pairs of corresponding points needed. If the transformation type is perspective transform, at least four pairs of corresponding points needed. Our case fits the second one thus four points from each camera view are enough for calculation of the homography matrix.

5.3.2. The Direct Linear Transformation (DLT) Algorithm

Direct linear transformation is a simple linear algorithm that solves a set of variables from a set of similarity relations. In our case similarity relation is $x_i' = Hx_i$ and the variable to be solved is H .

If the j -th row of the matrix H is denoted by h^jT , then we may write

$$Hx_i = \begin{pmatrix} h^{1T} x_i \\ h^{2T} x_i \\ h^{3T} x_i \end{pmatrix} \quad 5-1$$

Writing $x_i' = (x_i', y_i', w_i')$, the cross product may then be given explicitly as

$$x_i' \times Hx_i = \begin{pmatrix} y_i' h^{3T} x_i - w_i' h^{2T} x_i \\ w_i' h^{1T} x_i - x_i' h^{3T} x_i \\ x_i' h^{2T} x_i - y_i' h^{1T} x_i \end{pmatrix} \quad 5-2$$

Since $h^jT x_i = x_i'^T h^j$ for $j = 1, 2, 3$, this gives a set of three equations in the entries of H , which may be written in the form

$$\begin{bmatrix} 0^T & -w_i'x_i^T & y_i'x_i^T \\ w_i'x_i^T & 0^T & -x_i'x_i^T \\ -y_i'x_i^T & -x_i'x_i^T & 0^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \quad 5-3$$

these equations have the form $A_i h = 0$, where A_i is a 3×9 matrix and h is a 9-vector made up of the entries of matrix H ,

$$h = \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix}, \quad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad 5-4$$

with h_i is the i -th element of h . Only two out of three equations in (5.3) are linearly independent. Thus each point correspondence gives two equations in the entries of H , and third equation is omitted. Then, the set of equations becomes

$$\begin{bmatrix} 0^T & -w_i'x_i^T & y_i'x_i^T \\ w_i'x_i^T & 0^T & -x_i'x_i^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \quad 5-5$$

The homography matrix H is computed by solving a set of equations $Ah=0$, where h is the vector containing the entries of the matrix H . 8 equations are needed to solve for H . Then, the algorithm is

- (i) For each correspondence $x_i \Leftrightarrow x_i'$, compute the matrix A_i from (5.3). Only the first two rows are required in general.

- (ii) Assemble the n 2×9 matrices A_i into a single $2n \times 9$ matrix A .
- (iii) Obtain SVD of A . Solution for h is the last column of V .
- (iv) The matrix H is determined from h as in (5.4).

Once we calculate homography matrix via DLT algorithm, we have tested the success of DLT algorithm on PETS2001 sequence [8]. Homography matrix has been calculated using eight points. Four points are taken from the first camera image and four points are taken from the second camera image. Below examples of point transformation using homography matrix is presented. A point is marked on the first camera image and correspondence point is shown on the second camera image.



(a)

(b)

Figure 5-7 Original camera views, (a) camera 1, (b) camera 2.

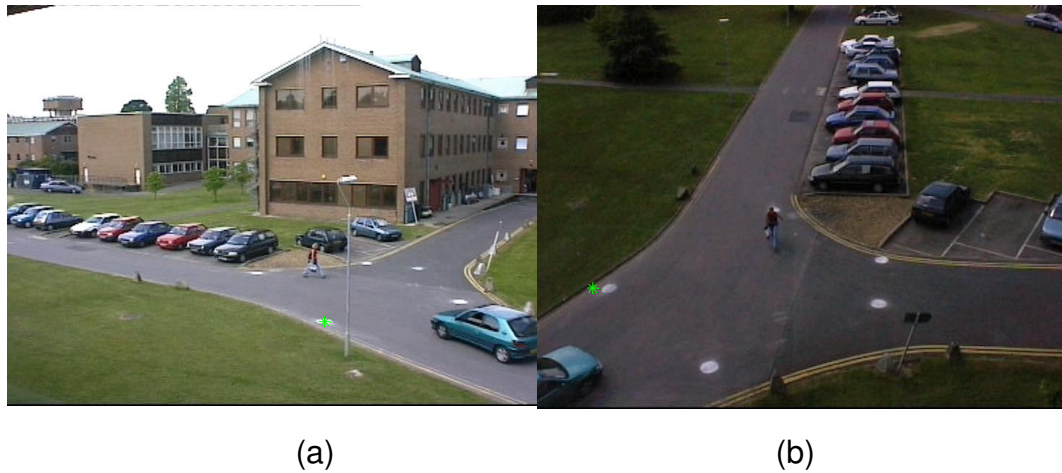


Figure 5-8 Example point transformation, (a) camera 1, (b) camera 2.

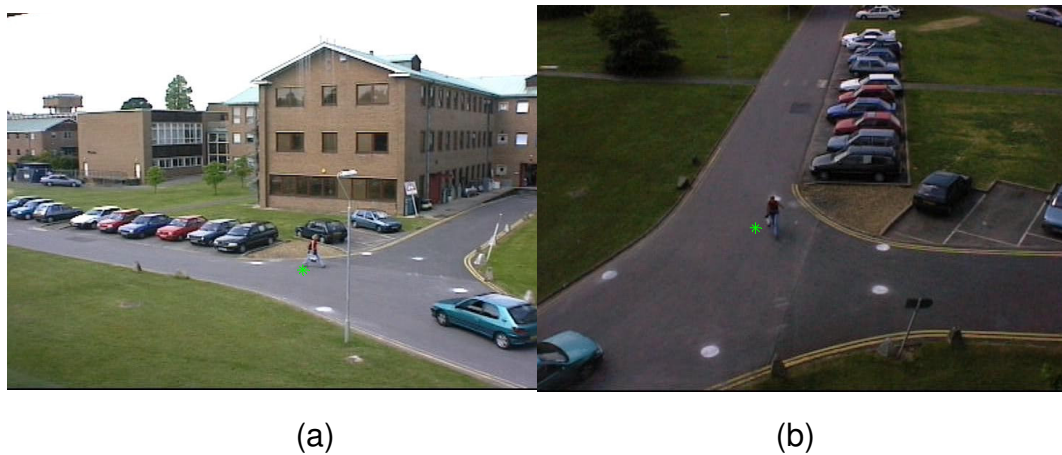


Figure 5-9 Example point transformation, (a) camera 1, (b) camera 2.

5.3.3. Calculating Track Correspondences Using Fuzzy Logic

In order to solve the correspondence problem between cameras, fuzzy logic is used. Euclidean distance and area ratio are the inputs and the track correlation value is the output of the Fuzzy Logic Data Association Engine.

Using the homography matrix we can transform a track's position in one camera to another camera with a small error. The difference between transformed position and the exact position is the Euclidean distance that is one of the inputs of Fuzzy Logic Data Association Engine. The transformed point is chosen to be left bottom corner of the object. This is because of getting rid of projection errors. As explained above there will be transformation errors, thus fuzzification of this input needs some modifications. Since transformation error will be an addition we have to widen the intervals. Below fuzzification of Euclidean distance is presented.

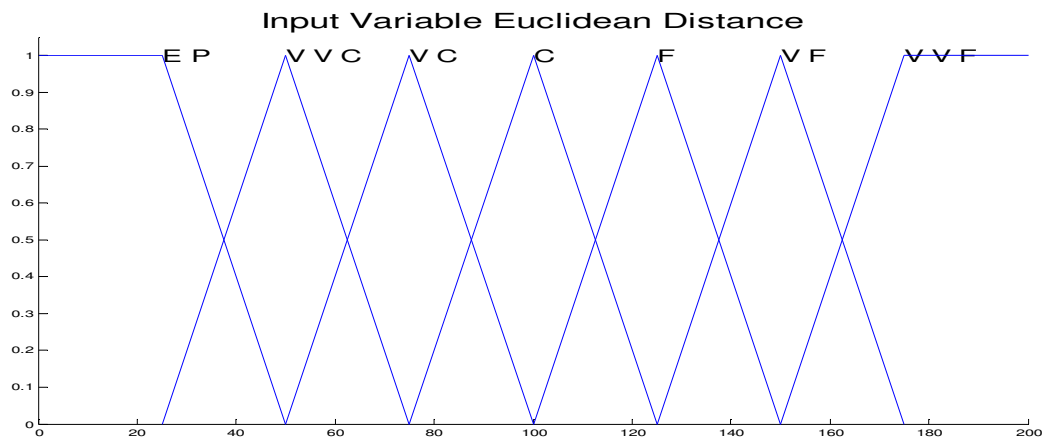


Figure 5.12 Fuzzification of input variable Euclidean distance for multiple camera case.

The above scenario is valid for the area ratio. The areas of same object in different cameras will differ very much. Thus we cannot use the same Fuzzy Logic Data Association Engine that is used in single camera tracking. There should be some differences. Changing Fuzzy Knowledge Base will solve this problem. Since Euclidean distance error is smaller, the base (more important) attribute that is used to solve the correspondence problem will be Euclidean distance. The importance of area ratio is reduced compared to single camera tracking.

In order to get the IF-THEN rules, fuzzified inputs are combined so that they give the degree of track correlation value.

Table 5-15 Generation of IF-THEN Rules

A.R.\E.D.	EP	VVC	VC	C	F	VF	VVF
VVSR	VHA	HA	MA	SA	VSA	NA	NA
VSR	VHA	HA	MA	SA	VSA	NA	NA
SR	FA	VHA	HA	MA	SA	VSA	NA
ER	FA	VHA	HA	MA	SA	VSA	NA
HR	FA	VHA	HA	MA	SA	VSA	NA
VHR	VHA	HA	MA	SA	VSA	NA	NA
VVHR	VHA	HA	MA	SA	VSA	NA	NA

In the table, NA stands for No Association, VSA stands for Very Small Association, SA stands for Small Association, MA stands for Medium Association, HA stands for High Association, VHA stands for Very High Association, and FA stands for Full Association.

From the combination of fuzzified inputs following IF-THEN rules are obtained:

IF Distance is EP and Ratio is VVSR THEN Association if VHA

IF Distance is EP and Ratio is VSR THEN Association if VHA

IF Distance is EP and Ratio is SR THEN Association if FA

IF Distance is EP and Ratio is ER THEN Association if FA

IF Distance is EP and Ratio is HR THEN Association if FA

IF Distance is EP and Ratio is VHR THEN Association if VHA
IF Distance is EP and Ratio is VVHR THEN Association if VHA

IF Distance is VVC and Ratio is VVSR THEN Association if HA
IF Distance is VVC and Ratio is VSR THEN Association if HA
IF Distance is VVC and Ratio is SR THEN Association if VHA
IF Distance is VVC and Ratio is ER THEN Association if VHA
IF Distance is VVC and Ratio is HR THEN Association if VHA
IF Distance is VVC and Ratio is VHR THEN Association if HA
IF Distance is VVC and Ratio is VVHR THEN Association if HA

IF Distance is VC and Ratio is VVSR THEN Association if MA
IF Distance is VC and Ratio is VSR THEN Association if MA
IF Distance is VC and Ratio is SR THEN Association if HA
IF Distance is VC and Ratio is ER THEN Association if HA
IF Distance is VC and Ratio is HR THEN Association if HA
IF Distance is VC and Ratio is VHR THEN Association if MA
IF Distance is VC and Ratio is VVHR THEN Association if MA

IF Distance is C and Ratio is VVSR THEN Association if SA
IF Distance is C and Ratio is VSR THEN Association if SA
IF Distance is C and Ratio is SR THEN Association if MA
IF Distance is C and Ratio is ER THEN Association if MA
IF Distance is C and Ratio is HR THEN Association if MA
IF Distance is C and Ratio is VHR THEN Association if SA
IF Distance is C and Ratio is VVHR THEN Association if SA

IF Distance is F and Ratio is VVSR THEN Association if VSA
IF Distance is F and Ratio is VSR THEN Association if VSA

IF Distance is F and Ratio is SR THEN Association if SA
IF Distance is F and Ratio is ER THEN Association if SA
IF Distance is F and Ratio is HR THEN Association if SA
IF Distance is F and Ratio is VHR THEN Association if VSA
IF Distance is F and Ratio is VVHR THEN Association if VSA

IF Distance is VF and Ratio is VVSR THEN Association if NA
IF Distance is VF and Ratio is VSR THEN Association if NA
IF Distance is VF and Ratio is SR THEN Association if VSA
IF Distance is VF and Ratio is ER THEN Association if VSA
IF Distance is VF and Ratio is HR THEN Association if VSA
IF Distance is VF and Ratio is VHR THEN Association if NA
IF Distance is VF and Ratio is VVHR THEN Association if NA

IF Distance is VVF and Ratio is VVSR THEN Association if NA
IF Distance is VVF and Ratio is VSR THEN Association if NA
IF Distance is VVF and Ratio is SR THEN Association if NA
IF Distance is VVF and Ratio is ER THEN Association if NA
IF Distance is VVF and Ratio is HR THEN Association if NA
IF Distance is VVF and Ratio is VHR THEN Association if NA
IF Distance is VVF and Ratio is VVHR THEN Association if NA

5.3.4. Multiple Camera Tracking

After single camera tracking phase is finished we come to multiple camera tracking phase. First, objects in camera-1 are transformed to camera-2. Then Euclidean distance and area ratios are calculated. In multi cameras, a track can be in the field of view of only one camera, or a track can be in

the field of view of both cameras. Tracks may be occluded in one camera view and may not be occluded in the other camera view. A track may be behind a stationary object in one camera view but in other camera this track may be seen normally. Below sample Euclidean distance and area ratio matrix is presented.

Table 5-16 Sample Euclidean Distance Matrix Calculated After Objects in Camera-1 are Trasnformed to Camera-2

C-1/C-2	T-1	T-2	T-3	T-4	T-5
T-1	200	178	15	250	375
T-2	17	234	174	287	254
T-3	354	213	155	34	90
T-4	253	67	159	198	301

Table 5-17 Sample Area Ratio Matrix Calculated After Objects in Camera-1 are Trasnformed to Camera-2

C-1/C-2	T-1	T-2	T-3	T-4	T-5
T-1	60	45	70	134	56
T-2	85	110	80	90	90
T-3	120	25	86	65	65
T-4	150	92	145	95	70

C-1 stands for Camera-1, C-2 stands for Camera-2 and T's are the tracks.

These are the inputs for Fuzzy Data Association Engine, and we get track correlation value as the output. With these values a track correlation matrix is formed.

During track correspondence and multi-camera tracking following rules are applied:

1. Tracks that are in the field of view of both cameras are matched.
 - a. If there is a match, then the strength of those tracks are increased.
 - b. If there is not a match, track strengths are not affected.
2. If both cameras does not have observations associated with a track which is seen by both cameras in the previous iteration, and the predicted position of this track is in the field of view of both cameras in the current iteration, then the strength of this track is decreased much compared to Single Camera Tracking.
3. If there is a track that is seen by both cameras in the previous iteration, and seen by only one camera in the current iteration, then this track is said to pass behind a stationary object in the camera which does not see this track. This case does not affect the track strength. The tracking in both cameras is done according to data that is gathered by the camera that sees the track.
4. If a track that is in the field of view of both cameras exits from field of view of one camera (there is no associated observations and predicted position is out of FOV), then it is deleted from the camera which it exited the FOV, and continued to be tracked in other camera.

5. If there is a case that an observation is associated with more than one track, and these tracks are marked as occluded in the occlusion matrix in one of the cameras;
 - a. If the same case occurred in other camera, then these tracks are said to be occluded in both cameras. Compared to Single Camera Tracking we are surer about occlusion. Because there can be an error when single camera is used, but there is a very small probability that there is an error in both cameras, since we have more information due to overlapping FOVs and different angles of cameras. In this case a new track is formed, and occluded tracks are tracked in the background.
 - b. If the predicted positions of occluded tracks match with tracks in other camera, and these tracks are not occluded in that camera, then we are sure about occlusion. This is the advantage of multiple cameras that have overlapping FOVs and different angles. In this case a new track is formed in the camera which has occluded tracks. Occluded tracks are tracked in the background and updated by the data that is gathered from other camera. Thus motion type changes (direction, velocity) are handled, and accuracy of the tracking is increased.
6. Occlusion is said to end when there are observations in the threshold added area of the track formed by occlusion, and they are matched with tracks in other camera. The track formed by occlusion is deleted and normal tracking of occluded tracks continue.
7. Observations that are not associated with tracks and track candidates initiate new track candidates.

- a. If these observations are seen by only one camera then track initiation is done as Single Camera Tracking.
 - b. If the track candidate is seen by both cameras then it becomes a track if it is seen in three consecutive iterations. Track strength is increased more compared to Single Camera Tracking. After three iterations track candidate's strength passes the threshold and track candidate becomes a track.
 - c. If the track candidate is in the FOV of both cameras but only seen by one camera, then track strength does not increase as (b). This may be because of track's passing behind a stationary object in one camera view. Track candidate becomes a track after five consecutive iterations.
 - d. If track candidate is not seen by both cameras, then it is deleted.
8. During track initiation a track candidate may be occluded with existing tracks in one of the cameras. In order to handle these situations, the corresponding position of the track candidate in other camera is checked. If there is a match, then normal track initiation procedure continues.
 9. Tracks that are seen only by one camera are tracked according to Single Camera Tracking.
 10. Tracks that have strengths below the threshold are deleted.

5.3.4.1. Examples of Multiple Camera Tracking

Example-1:

The case presented in this example is like the case in Figure 5.14. In the below matrix the correlation values between the Camera-1 objects' transformed to Camera-2 and Camera-2 objects are presented. In this case occlusion matrix is calculated for both cameras and there are not any tracks that are occluded with each other in both cameras.

Table 5-18 Sample Matrix that shows calculated track correlation values with objects in Camera-1 transformed to Camera-2 and objects in Camera-2

C-1/C-2	T-1	T-2	T-3	T-4	T-5
T-1	0	0.12	0.56	0	0
T-2	0.85	0	0.18	0.23	0
T-3	0.27	0	0	0.74	0
T-4	0	0	0	0	0

According to rules, Table 5-18 is interpreted. In this case it is known that there is not any tracks that are occluded with each other. Thus track T-1 in Camera-1 correspond to track T-3 in Camera-2, track T-2 in Camera-1 correspond to track T-1 in Camera-2 and track T-3 in Camera-1 correspond to track T-4 in Camera-2. Track T-4 in Camera-1 is not in the field of view of Camera-2. Tracks T-2 and T-5 in Camera-2 are not in the field of view of Camera-1.

Example-2:

The case presented in this example is like the case in Figure 5.15. In the below matrix the correlation values between the Camera-1 objects' transformed to Camera-2 and Camera-2 objects are presented. In this case occlusion matrix is calculated for both cameras. Two tracks in Camera-1 are marked as occluded in occlusion matrix calculated for Camera-1 and the corresponding positions of the occluded tracks in Camera-2 match with tracks that are not occluded with each other. Thus we are sure about the occlusion and the correspondence problem will be solved accordingly.

Table 5-19 Sample matrix that shows calculated track correlation values with objects in Camera-1 transformed to Camera-2 and objects in Camera-2

C-1/C-2	T-1	T-2	T-3	T-4	T-5
T-1	0	0.42	0.56	0	0
T-2	0.85	0	0.18	0.23	0
T-3	0.27	0	0	0.74	0
T-4	0	0	0	0	0

According to rules, Table 5-19 is interpreted. In this case it is known that there are tracks that are occluded with each other. Thus track T-1 in Camera-1 correspond to tracks T-2 and T-3 in Camera-2, track T-2 in Camera-1 correspond to track T-1 in Camera-2 and track T-3 in Camera-1 correspond to track T-4 in Camera-2. Track T-4 in Camera-1 is not in the field of view of Camera-2. Track T-5 in Camera-2 is not in the field of view of Camera-1.

5.3.5. Results of Multiple Camera Tracking



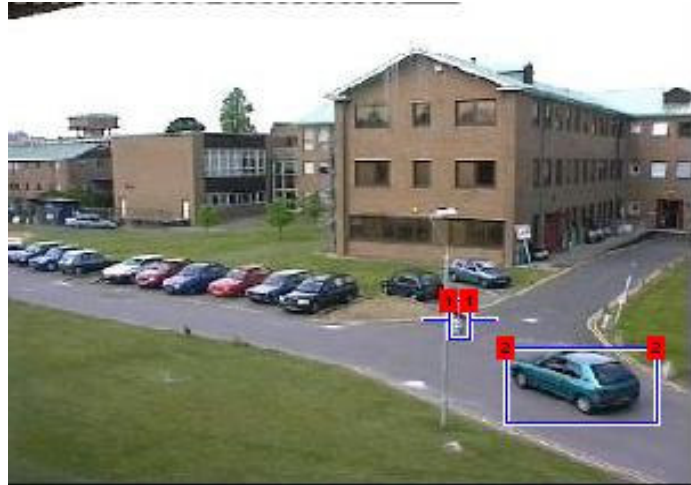
(a) Camera-1.



(b) Camera-2.

Figure 5.13 Results of target tracking using multiple cameras in PETS-2001 data set-1.

The number on the upper left corner of the object shows the track number in its own camera and the number on the upper right corner of the object shows the corresponding track in other camera. In this sample scenario track 1 in camera-1 is matched to track 1 in camera-2.



(a) Camera-1.



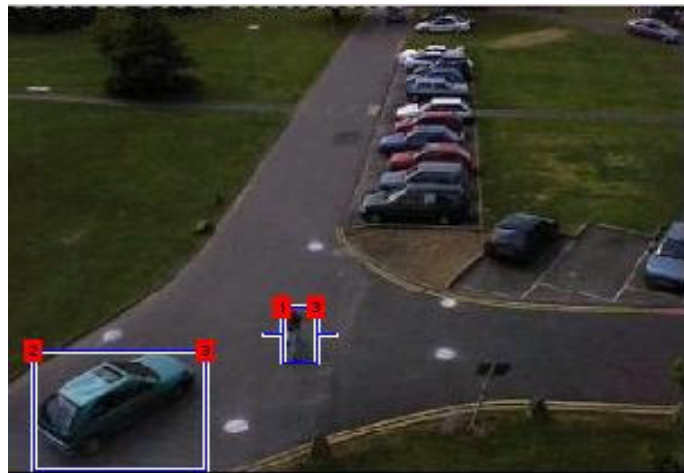
(b) Camera-2.

Figure 5.14 Results of multiple target tracking using multiple cameras in PETS-2001 data set-1.

In this sample scenario track 1 in camera-1 is matched with track 1 in camera-2 and track 2 in camera-1 is matched to track 2 in camera-2.



(a) Camera-1.

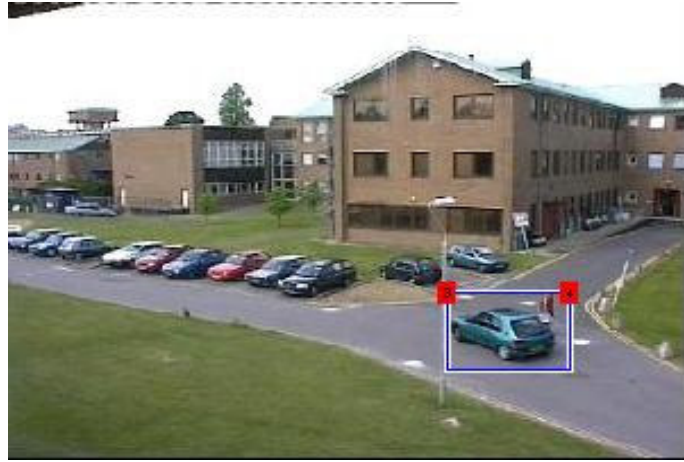


(b) Camera-2.

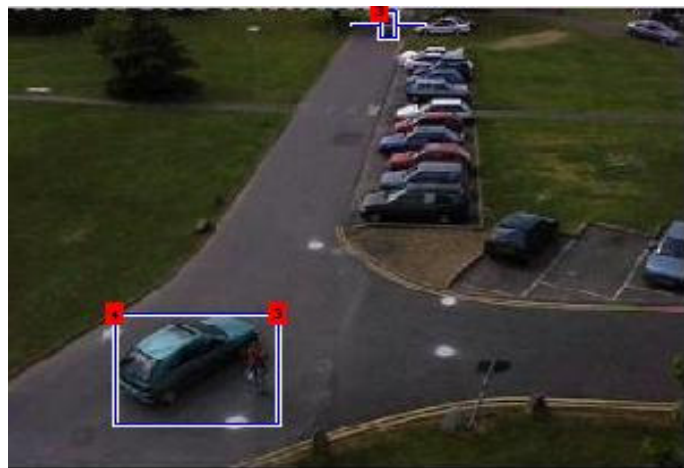
Figure 5.15 Results of multiple target tracking using multiple cameras in PETS-2001 data set-1: Tracks' occlusion in only one camera scenario.

In this sample scenario tracks 1 and 2 in camera-1 are occluded with each other, but corresponding tracks in camera-2 are not occluded. From this occlusion a new track, 3 is formed in camera-1. Track 3 in camera-1 is matched to tracks 1 and 2 in camera-2. Upper right corner of track 3 in camera-1 shows that this track is matched to tracks 1 and 2 in camera-2

(12(=1,2)). During occlusion, occluded tracks in camera-1 are updated with the information gathered by camera-2.



(a) Camera-1.

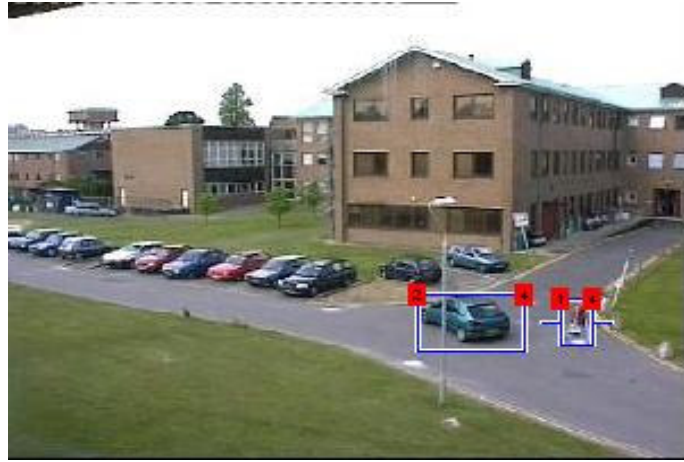


(b) Camera-2.

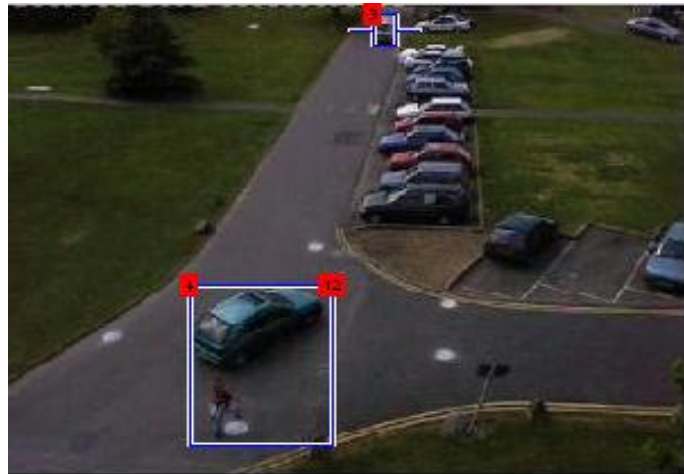
Figure 5.16 Results of multiple target tracking using multiple cameras in PETS-2001 data set-1: Track's occlusion in both cameras scenario.

In this sample scenario tracks 1 and 2 in both cameras are occluded with each other. Thus, from these occlusions new tracks, 3 in camera-1 and 4

in camera-2 are formed. Track 3 in camera-1 is matched to track 4 in camera-2. Track 3 in camera-2 does not match to any tracks in camera-1.



(a) Camera-1.

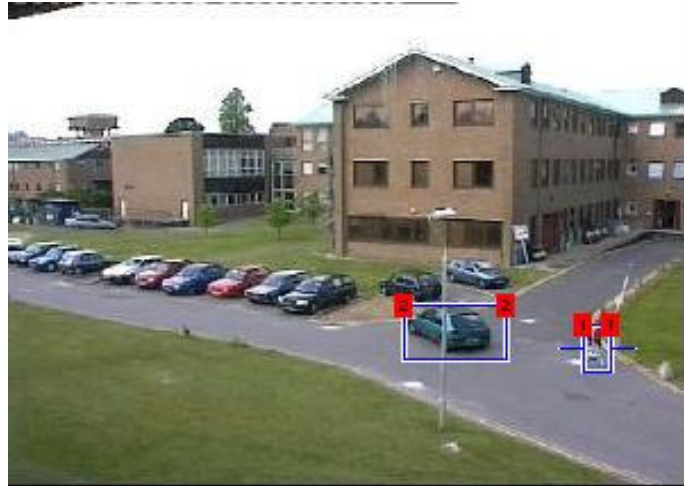


(b) Camera-2.

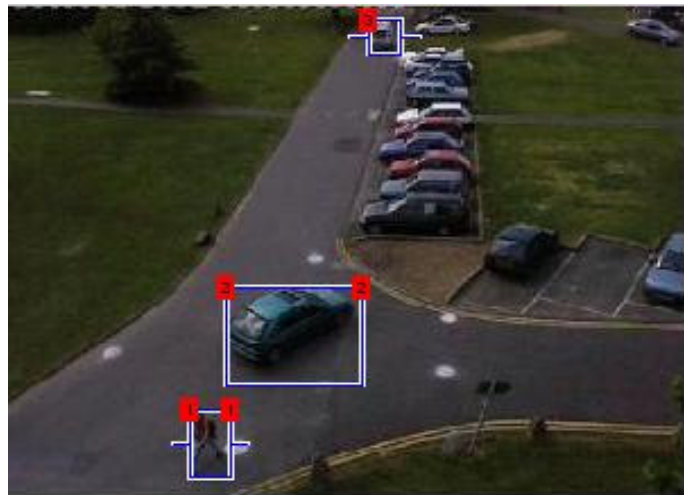
Figure 5.17 Results of multiple target tracking using multiple cameras in PETS-2001 data set-1: Track's occlusion in only one camera scenario.

In this sample scenario tracks 1 and 2 in camera-2 are occluded with each other, and from this occlusion a new track, 4 is formed. The occlusion of these tracks is ended in camera-1. Thus track 4 in camera-2 corresponds

to tracks 1 and 2 in camera-1. Upper right corner of track 4 in camera-2 shows that this track corresponds to tracks 1 and 2 in camera-1 ($12(=1,2)$). Track 3 in camera-2 does not match to any tracks in camera-1. During occlusion, occluded tracks are updated with the information gathered from camera-1.



(a) Camera-1.



(b) Camera-2.

Figure 5.18 Results of multiple target tracking using multiple cameras in PETS-2001 data set-1: Occlusion ends scenario.

In this sample scenario the occlusion of tracks 1 and 2 in camera-2 ends and track 4 which is formed by occlusion is deleted. Track 1 in camera-1 is matched to track 1 in camera-2, track 2 in camera-1 is matched to track 2 in camera-2. During the occlusion, motion type changes are handled by updating the occluded tracks with the information gathered by other camera. Thus the predicted positions of the occluded tracks are much accurate. So, matching the tracks after occlusion is done correct. Track 3 in camera-2 does not match to any tracks in camera-1.



(a) Before passing behind a stationary object.



(b) During passing behind a stationary object.



(c) Track is completely behind the object.



(d) Track appears again.

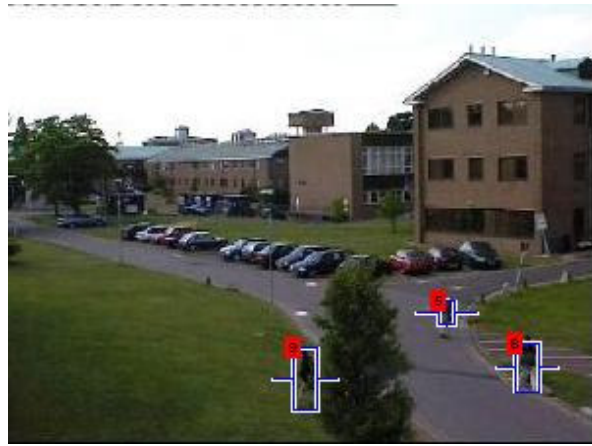
Figure 5.19 Results of multiple target tracking using multiple cameras in PETS-2001 data set-2: A tracks' passing behind a stationary object scenario, left frame from camera-1, right frame from camera-2.

In this sample scenario track-2 is passing behind a tree in camera-1. Corresponding track in camera-2, track-1 is always in sight. Track-2 in camera-1 disappears (does not have any associated observations), but predicted position of this track in camera-2 corresponds to track-1 in camera-2. Thus, it is concluded that track is passing behind a stationary object and can be seen only by one of the cameras. Track-2 in camera-1 is updated with the information gathered from camera-2 while passing behind the tree. Thus the predicted position is much accurate and the matching of the track after appearing again is done correctly.



(a) Before passing behind an object.

(b) Track completely disappears.



(c) Track appears again.

Figure 5.20 Results of multiple target tracking using multiple cameras in PETS-2001 data set-2; track which is in the FOV of one of the cameras passes behind an object.

In this sample scenario track-7 is passing behind a tree in camera-1. This track is not in the FOV of Camera-2. Thus track is lost when it is behind the tree. When the track appears again, a new label, 9 is given. In Figure 5.19 a scenario like the one above is presented. In that case the track that is passing behind a tree was seen by both cameras. Thus the track was not lost when it is behind the tree, and the match after appearing again was done correctly.



(a) Frame-600.



(b) Frame-690.



(c) Frame-763.

Figure 5.21 Results of multiple target tracking using multiple cameras in PETS-2001 data set-2: A tracks' passing behind a stationary object scenario-2, left frame from camera-1, right frame from camera-2.

CHAPTER 6

PERFORMANCE EVALUATION

From the results presented it can be seen that the proposed methods are very successful. When objects move very slowly, moving object detection part has some problems. This is because slowly moving objects become a part of the background. Swaying of trees and illumination change problems are solved. The data association part is also very fast since computational load is very low compared to probabilistic data association techniques and Multiple Hypothesis Tracking Technique. Thus it is suitable for real-time target tracking. Data association part also handles track initiation, track deletion and track merging.

Below the results of the PETS-2001 data set-2 sequence from this work and [36] is compared. In [36], Gradients of images are combined with color information for object detection, and tracking is performed using a voting scheme that utilizes color and shape cues to establish the correspondence. FOV lines which indicates the areas that are seen by both cameras are automatically determined by the objects that are entering the FOV of the camera or exiting the FOV of the camera. Then, these FOV lines are used to find corresponding objects between cameras.



(a) Frame-110.



(b) Frame-115.



(c) Frame-120.

Figure 6.1 Result of single camera tracking on PETS-2001 data set-2, on the left result of [36], (b) on the right result of proposed tracker.

From the above figure, it is noticed that the tracker in [36] has taken swaying trees as moving objects. The proposed tracker in this work solves the problem of swaying trees, and periodic motion.

The performance of the proposed rule-based tracker that is applied to multiple camera configurations on PETS-2001 data set-1 sequence is presented below.

Table 6-1 Results of Multi Camera Correspondence

Frame	Camera-1	Camera2	Comment
300	1	1	Correct
545	3	1,2	Correct
585	3	4	Correct
585	-	3	Correct
610	1	1	Correct
610	2	2	Correct
610	-	3	Correct
708	4	3	Correct
708	-	2	Incorrect
830	4	3	Correct
830	5	5	Correct
830	6	-	Correct
875	4,5	7	Correct
875	6	-	Correct
880	6	-	Correct
880	7	7	Correct
900	7	3,5	Correct

900	6	-	Correct
1041	6	-	Correct
1041	8	9	Incorrect
1066	6	-	Correct
1066	8	9	Incorrect
1066	-	11	Incorrect
1088	8	9	Correct
1088	9	11	Correct
1088	6	-	Correct
1412	13	12	Correct
1412	8	-	Correct

In this table each row is interpreted separately. For example, first row states that object-1 in camera-1 is the same as object-1 in camera-2 in frame 300. Cells that contain more than one number states that, these objects are not occluded with each other in that camera view but they are occluded in other camera view. Rows that have free cells states that object is in the field of view of only one camera.

The performance of the proposed multi camera tracker in [38] on PETS-2001 data set-1 sequence is presented below. In [38] Gaussian Mixture Models are used for modeling the background, and motion correspondence is used for tracking. FOV lines which indicates the areas that are seen by both cameras are automatically determined by the objects that are entering the FOV of the camera or exiting the FOV of the camera. Then, these FOV lines are used the find corresponding objects between cameras.

Table 6-2 Results of Multi Camera Correspondence from [38]

Frame	Camera	Object	Camera	Object	Comment
98	1	1	2	1	Correct
470	2	2	1	2	Correct
688	1	3	2	3	Correct
774	2	4	1	4	Correct
963	2	5	1	5	Correct
1074					Incorrect
1185	1	6	2	7	Correct
1423	2	8	1	8	Correct
1578	2	9	1	7	Incorrect
2106	2	10	1	9	Correct
2177	2	12	1	10	Correct

In the above table, the first row states that object 1 in camera 1 is the same as object 1 in camera 2.

When two tables are compared, it can be said that the proposed rule-based tracking method in this work is successful. It has some errors, but in general it is doing the job well.

CHAPTER 7

CONCLUSION

7.1. Conclusion

In recent years, automated visual surveillance became very important. In addition to recent advances in computer speed, availability of large-capacity storage devices cheaply and high speed computer network infrastructure, the increase in crime rate has made visual surveillance very popular.

In this thesis, important parts of an automated visual surveillance system have been presented and implemented. Moving object detection part is accomplished by Gaussian Mixture Models which is an adaptive method that solves the problem of dynamic scene changes. After detecting the foreground regions, shadow and noise removal is applied in order to get the exact objects.

Tracking is the next step after obtaining moving objects. An Interacting Multiple Model Kalman Filter approach is implemented as the tracker in order to cope with multiple motion models. Snakes are used to extract the exact shape of the target.

Multiple target tracking is a hard problem. Additional information about targets should be used in order to make successful observation-to-track association. Fuzzy logic presents a systematic way of combining information. It also stimulates the human decision mechanism. Using fuzzy logic and additional rules, rule-based tracking methodology is presented. Rule-based tracking methodology handles track initiation, track confirmation, track update and track deletion successfully. This method then applied for multi-camera configuration in order to solve the problems like track correspondence, occlusion, match after occlusion and tracks' passing behind stationary objects.

To cope with the some difficulties, such as occlusion, multiple camera tracking is chosen as solution. Data acquired from multiple cameras are connected so that multiple cameras can be used as single cameras that have very high coverage. Using multiple cameras, more information can be obtained about objects. In order to correspond information acquired from cameras homography matrix which is calculated by the use of direct linear transformation has been utilized.

The use of rule-based tracking with multiple cameras that have overlapping field of views and different angles brings the solution of occlusion problem. Due to different angles of cameras, occluded objects are tracked more reliably. Data gathered from the camera that objects are not occluded also used for occluded tracks. Thus, a motion type change that may occur during occlusion is handled, and tracks' matching after the occlusion is done more reliably. This technique also helps in the solution of problem that an observation's occlusions with an existing track during track initialization phase. Also tracks' passing behind stationary objects like parked cars, walls or buildings are handled.

7.2. Future Work

Data association part is very crucial in multiple target tracking, especially in the closely spaced targets case. Thus the proposed rule-base data association method can be improved as a future work. In our rule-based data association method we used the ratios of the areas of target and measurement. Instead of this, comparing the shape of the target and measurement that is obtained by snakes can be a better idea. Also new features can be added to the data association part in order to increase the accuracy of the association.

In order to handle complex occlusion cases like occlusion of a group of people, knowing the number of people in the group may be helpful. Thus human model analysis should be studied.

After all, adding high level behavioral analysis will make the system useful for surveillance applications. This high level analysis should be about what the tracked person does, leaves a box, paces back and forth, etc.

REFERENCES

- [1] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfindex:Real-time Tracking of the Human Body," *IEEE Trans. on Patt. Anal. and Machine Intell.*, vol. 19, no. 7, pp. 780-785, 1997.
- [2] C. Stauffer and W. E. L. Grimson. "Learning patterns of activity using real-time tracking". *IEEE Trans. on Patt. Anal. and Machine Intell*, 22(8):747–757, Aug 2000.
- [3] M. Kass, A. Witkin, D. Terzopoulos. "Snakes: Active Contour Models". *International Journal of Computer Vision*. Vol. 1(4), pp.321-331, 1987.
- [4] Y. Bar-Shalom, X.-Rong Li, T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley, 2001.
- [5] İ. Haritaoglu, D. Harwood and L. S. Davis. "W4: Real - Time Surveillance of People and Their Activities". *IEEE Trans. on Patt. Anal. and Machine Intell*, 22(8):809–830, Aug 2000.
- [6] Greg Welch and Gary Bishop. "An Introduction to the Kalman Filter", [http://www.cs.unc.edu/~\[welch,gb\]](http://www.cs.unc.edu/~[welch,gb]), Aug. 2007.

- [7] AVITRACK (Aircraft surroundings, categorized Vehicles & Individuals Tracking for apron's Activity model interpretation & Check) Project, www.avitrack.net, Aug. 2007.
- [8] University of Reading, PETS 2001, <http://ftp.pets.rdg.ac.uk/PETS2001/>, Sept. 2007.
- [9] D. Meyer, J. Denzler, and H. Niemann, "Model based extraction of articulated objects in image sequences for gait analysis," in *Proc. IEEE Int. Conf. Image Processing*, pp. 78–81, 1998.
- [10] A. Elgammal, R. Duraiswami, D. Harwood, L.S. Davis, "Background and foreground modeling using non-parametric kernel density estimation for visual surveillance", *Proc. of The IEEE*, Vol. 90, 1151-1163, 2002.
- [11] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 42–77, 1994.
- [12] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, "A system for video surveillance and monitoring," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep., CMU-RI-TR-00-12, 2000.
- [13] M. Piccardi, "Background subtraction techniques: a review." *Systems, Man and Cybernetics*, 2004 IEEE International Conference, Vol 4, pp.3099-3104, 2004.

- [14] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Transactions on the ASME Journal of Basic Engineering, Vol. 82, pp. 35-45, 1960.
- [15] Blackman, S. and Popoli, R., Design and Analysis of Modern Tracking Systems, Artech House Radar Library, Boston, 1999.
- [16] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event detection and analysis from video streams," IEEE Trans. Pattern Anal. Machine Intell., Vol. 23, pp. 873–889, Aug. 2001.
- [17] J. Black and T. Ellis, "Multi-camera Image Tracking", Image and Vision Computing, Elsevier, 2005.
- [18] E. Salvador, Andrea Cavallaro and T. Ebrahimi, "Shadow Identification and Classification Using Invariant Color Model," ICASSP 2001, May 7-11, 2001.
- [19] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, "Detecting moving shadows: Algorithms and evaluation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 25, pp. 918-923, 2003.
- [20] D.B. Reid, "An algorithm for tracking multiple targets," IEEE Trans. on Automatic Control, vol. 24, no. 6, pp. 843-854, Dec. 1979.
- [21] Blackman, S. and Popoli, R., *Design and Analysis of Modern Tracking Systems*, Artech House Radar Library, Boston, 1999.

- [22] Kurien, T., "Issues in the Design of Practical Multitarget Tracking Algorithms", *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom (Ed.), Norwood, MA: Artech House, Chapter 3, 1990.
- [23] W. Hu, T. Tan, L. Wang, and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors", *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, Vol. 34, No. 3, pp. 334-352, 2004.
- [24] F. Oberti, G. Ferrari, and C. S. Regazzoni. A Comparison between Continuous and Burst, Recognition Driven Transmission Policies in Distributed 3GSS, chapter 22, pages 267–278. *Video-Based Surveillance Systems*. Kluwer Academic Publishers, Boston, 2002.
- [25] Samuel S. Blackman. "Multiple Hypothesis Tracking for Multiple Target Tracking", *IEEE A&E Systems Magazine*, Vol. 19, No. 1, Part 2 Tutorials, January 2004.
- [26] L. Wang, W. Hu, and T. Tan. "Recent developments in human motion analysis", *Pattern Recognition*, 36(3):585–601, March 2003.
- [27] A. M. McIvor. "Background subtraction techniques", In *Proc. of Image and Vision Computing*, Auckland, New Zealand, 2000.
- [28] A. Amer. "Voting-based simultaneous tracking of multiple video objects", In *Proc. SPIE Int. Symposium on Electronic Imaging*, pages 500–511, Santa Clara, USA, January 2003.

- [29] S. Ju, M. Black, and Y. Yaccob. "Cardboard people: a parameterized model of articulated image motion", In Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition, pages 38–44, 1996.
- [30] R. Rosales and S. Sclaroff. "Improved tracking of multiple humans with trajectory prediction and occlusion modeling", In Proc. of IEEE CVPR Workshop on the Interpretation of Visual Motion, Santa Barbara, CA, 1998.
- [31] Cox, I. L., and Hingorani, S. L. "An efficient implementation of Reid's multiple hypotheses tracking algorithm and its evaluation for the purposes of visual tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence. 18, 2, 138-150, 1996.
- [32] Chris Stauffer W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," Proceedings IEEE Conf. on Computer Vision and Pattern Recognition, pages 246-252, 1999
- [33] Kirubarajan, T., Bar-Shalom, Y., Pattipati, K.R., Kadar, I., Abrams, B., Eadan E., Tracking ground targets with road constraints using an IMM estimator, Aerospace Conference, 1998. Proceedings, IEEE Volume 5, Issue 21-28, Page(s): 5 – 12, vol.5, Mar. 1998.
- [34] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, Tim Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", IEEE Transactions on Signal Processing, Vol. 50, no. 2, February 2002.

- [35] R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, UK, 2003.
- [36] Omar Javed, Real Time Automated Surveillance System, <http://server.cs.ucf.edu/~vision/projects/Knight/Knight.html>, Aug. 2007.
- [37] University of Reading, PETS 2000, <http://ftp.pets.rdg.ac.uk/PETS2001/>, June 2007.
- [38] Sohaib Khan, Omar Javed, Mubarak Shah, "Tracking with Uncalibrated Cameras With Overlapping Field of View", PETS 2001, (with CVPR 2001), Kauai, Hawaii, Dec 9th, 2001.