VISUAL DETECTION AND TRACKING OF MOVING OBJECTS

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

BY

HAMZA ERGEZER

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

NOVEMBER 2007

Approval of the Thesis

VISUAL DETECTION AND TRACKING OF MOVING OBJECTS

Submitted by **HAMZA ERGEZER** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

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

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** __________

Prof. Dr. Kemal Leblebicioğlu
Supervisor, **Electrical and Electronics Engineering Dept., METU** __________

Examining Committee Members

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

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

Prof. Dr. Mübeccel Demirekler
Electrical and Electronics Engineering Dept., METU ______________

Assist. Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering Dept., METU ______________

MSc. Umur Akıncı
MGEO, ASELSAN ______________

Date: 29.11.2007

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 : Hamza ERGEZER

Signature :

ABSTRACT

VISUAL DETECTION AND TRACKING OF MOVING OBJECTS

Ergezer, Hamza
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Kemal Leblebicioğlu

November 2007, 85 pages

In this study, primary steps of a visual surveillance system are presented: moving object detection and tracking of these moving objects. Background subtraction has been performed to detect the moving objects in the video, which has been taken from a static camera. Four methods, frame differencing, running (moving) average, eigenbackground subtraction and mixture of Gaussians, have been used in the background subtraction process. After background subtraction, using some additional operations, such as morphological operations and connected component analysis, the objects to be tracked have been acquired. While tracking the moving objects, active contour models (snakes) has been used as one of the approaches. In addition to this method; Kalman tracker and mean-shift tracker are other approaches which have been utilized. A new approach has been proposed for the problem of tracking multiple targets. We have implemented this method for single and multiple camera configurations. Multiple cameras have been used to augment the measurements. Homography matrix has been calculated to find the correspondence between cameras. Then, measurements and tracks have been associated by the new tracking method.

Keywords: Visual surveillance, moving object detection, background subtraction, moving object tracking, multiple hypothesis tracking, object tracking with multi-camera.

ÖZ

HAREKETLİ NESNELERİN GÖRSEL TESPİTİ VE İZLENMESİ

Ergezer, Hamza
Yüksek Lisans, Elektrik-Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Kemal Leblebicioğlu

Kasım 2007, 85 sayfa

Bu çalışmada, bir görsel gözetim sisteminin ilk adımları olan hareketli nesnelerin tespiti ve izlenmesi ile ilgili yapılan çalışmalar sunulmuştur. Hareketli nesnelerin tespiti için sabit bir kameradan alınan görüntüde arkaplan çıkarma (modelleme) yöntemi kullanılmıştır. Arkaplan modellemede dört yöntem gerçekleştirilmiş ve performansları karşılaştırılmıştır. Arkaplan çıkarımının ardından, takip edilecek nesnelerin düzgün bir şekilde elde edilmesi ve belirlenmesi amacıyla morfolojik operatörlerden ve bağlı eleman analizi gibi ek işlemlerden yararlanılmıştır. Tek kamerada hareketli nesnelerin takibinde ise, aktif dış çevritlerden, Kalman süzgecinden ve ortalama değer kayması yönteminden yararlanılmıştır. Çoklu kamerayla çoklu hedef takibi problemi için yeni bir metot önerilmiştir. Klasik çoklu varsayım takibi metodunda bulanık mantık kullanılarak çoklu hedef takibi yapılmıştır. Hedefler ve takipler hakkında daha çok bilgi sağlanması amacıyla ikili kamera sistemi kullanılmıştır. İki kamera arasındaki eşleştirmeyi bulmak için eşleştirme (homography) matrisi hesaplanmıştır. Kameralardan birinde hedefler arasında örtüşme olması durumunda, diğer kameradan gelen bilgilerden yararlanılmıştır. Uygulanan ve önerilen metotlarla ile ilgili test sonuçları da ayrıntılı bir şekilde verilmiştir.

Anahtar Kelimeler: Görsel gözetim sistemleri, hareketli nesnelerin tespiti, arkaplan çıkarımı, hareketli nesnelerin izlenmesi, çoklu varsayım takibi, çoklu kamerayla hedef takibi.

To my family

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Prof. Dr. 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

CHAPTER 1

INTRODUCTION

Visual surveillance has become a popular area for research and development with recent advances of computer (hardware) and camera technology. In addition to this development in technology, due to increasing crime rate, better precautions are required in security-sensitive areas, such as, country borders, airports and some government offices.

Traditional security systems were greatly depending on operator instead of an automated system. As a result, detection and judgment of events was limited with the concentration of the operator. Additionally, with traditional systems, area under surveillance must be restricted with the number of operators and number of cameras may exceed their monitoring capability. This situation forces the use of more personnel, which makes it even a more expensive task in an era of much cheaper technological equipments being than the human resources.

Visual surveillance in dynamic scenes attempts to detect, recognize and track certain objects from image sequences, and more generally to understand and describe object behaviors. The aim is to develop intelligent visual surveillance to replace the traditional passive video surveillance that is proving ineffective as the number of cameras exceeds the capability of human operators to monitor them. In short, the goal of visual surveillance is not only to put cameras in the place of human eyes, but also to accomplish the entire surveillance task as automatically as possible.

A general framework of visual surveillance is shown in Figure 1.1. Additional steps can be inserted to this scheme. However, steps in the figure are the unavoidable parts of a visual surveillance system.

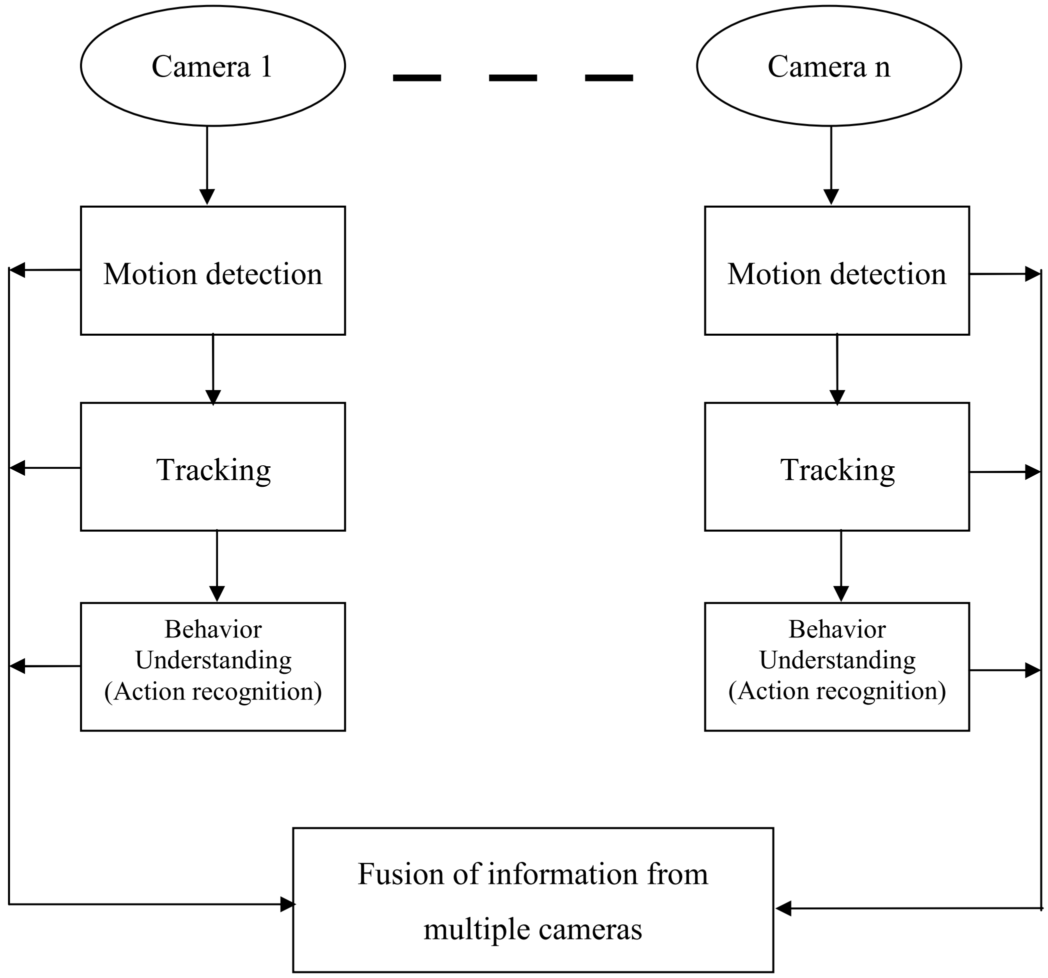

Camera 1
Camera n
Motion detection
Motion detection
Tracking
Tracking
Behavior Understanding (Action recognition)
Behavior Understanding (Action recognition)
Fusion of information from multiple cameras

Figure 1.1 General scheme of the visual surveillance systems

Visual surveillance is still one of the hot topics in computer vision area. There are many researches on the general visual surveillance theme and especially on the steps in Figure 1.1 and other related topics, such as object classification, occlusion handling, etc.

Visual surveillance has been investigated worldwide under several large research projects. For example, the Defense Advanced Research Projection Agency (DARPA) supported the Visual Surveillance and Monitoring (VSAM) project [15] in 1997, whose purpose was to develop automatic video understanding technologies that enable a single human operator to monitor behaviors over complex areas such as battlefields and civilian scenes. Furthermore, to enhance protection from terrorist attacks, the Human Identification at a Distance (HID) program sponsored by DARPA in 2000 aims to develop a full range of multimodal surveillance technologies for successfully detecting, classifying, and identifying humans at great distances. The European Union's 6th Framework Program sponsored AVITRACK [8] that aims at developing an intelligent surveillance system on the apron, addressing aircraft, vehicles and people's presence and movements.

1.1 Scope of the Thesis

This thesis is devoted to the problem of defining and developing the fundamental blocks of automated video surveillance systems via single-camera and multi-camera configurations. Initial problem is the detection of object motions in the scene. Background subtraction algorithms (frame differencing, running average, eigenbackground subtraction and mixture of Gaussians) which are capable of coping with the changes in the scene (i.e., adaptable), are described for extracting isolated moving objects. In order to remove undesired components which are part of foreground such as shadow and noises, a shadow removal algorithm and morphological operations have been utilized.

Tracking of detected objects is the next section of the thesis. Three methods have been implemented and discussed as object tracker. Kalman tracker, snake tracker and mean-shift tracker have been utilized as point tracker, contour tracker, and kernel tracker, respectively.

To cope with the some of the difficulties, such as occlusion, multiple camera tracking is chosen as solution. Using multiple cameras, more information can be obtained about objects. Also, information acquired from cameras must be corresponded. In order to find the correspondence between cameras, homography matrix has been utilized.

In order to track multiple targets, there must be a special algorithm to initiate, associate and finalize the tracks of objects. Multiple hypothesis tracking (MHT) is the only multi-target tracker which considers these three operations. This powerful feature of MHT is combined with fuzzy logic. Fuzzy logic has been used as an evaluation tool of tracks.

1.2 Outline of the Thesis

In Chapter 2, related works and broad overview for each of the proposed system blocks are presented.

Moving object detection, first step of the visual surveillance system, is described in Chapter 3. Four methods are extensively described and compared by means of obtained results. Shadow removal and morphological operations are also mentioned in this chapter.

After moving object detection, the next part, moving object tracking, is given in Chapter 4. Simple Kalman filter tracking, Active-contour (snake) based tracking, and mean-shift tracking are presented and discussed.

Rule-based multiple hypothesis tracking has been proposed in Chapter 5. General idea of generic multiple hypothesis tracking and proposed method that have been applied for both single and multiple camera have been presented. 2D homography is introduced as a correspondence tool for the multiple-camera configuration. Also, the results have been presented for single and multi-camera configurations.

This thesis is summarized in Chapter 6 and some future works are also discussed in this chapter.

CHAPTER 2

A SHORT SURVEY ON VISUAL SURVEILLANCE

The main theme of this thesis is the design of some of the primary blocks of an automated visual surveillance system. This chapter describes the latest studies in the literature related with these blocks.

2.1 Moving Object Detection

Moving object detection is generally the first block of a visual surveillance system. Detecting moving regions provides a focus of attention for later processes such as tracking and behavior analysis because only these regions will be considered in the later processes. Hence, moving object detection plays an important role in the overall performance of the system. There are several methods that aim at detecting the moving objects, but they can be divided into two conventional groups: *background subtraction (modeling)* and *optical flow*.

2.1.1 Background subtraction (modeling)

There are many methods that use a background model to extract the foreground objects. The methods in this category differ from each other by how background is modeled. Although all of methods propose a statistical model for background image, the utilized statistical model is different for each method.

Pfinder [1] uses a multi-class statistical model for the foreground objects, but the background model is a single Gaussian per pixel. After an initialization period where the room is empty, the system reports good results. There have been no reports on the success of this tracker in outdoor scenes.

Haritaoğlu [5], models the background by representing each pixel with its maximum intensity value, minimum intensity value and intensity difference values between consecutive frames in W^4 system. The limitation of such a model is its susceptibility to illumination changes.

Elgammal, *et al.* [12] 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. As a matter of fact, this theoretically well established method yields many 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. For example, Meyer *et al.* [11] computes the displacement vector field for the extraction of articulated objects. The results are used for gait analysis. Optical-flow-based methods can be used to detect independently moving objects even in the presence of camera motion. However, most flow computation methods are computationally complex and very sensitive to noise, and cannot be applied to video streams in real time without specialized hardware. More detailed discussion of optical flow can be found in Barron's work [13].

2.2 Moving Object Tracking

In visual surveillance systems, moving object detection is generally followed by tracking of the detected regions. There are various methods in the literature related with the visual tracking of moving objects. They can be classified into three major categories according to Yılmaz [14]: point tracking, kernel tracking and silhouette tracking. Furthermore, these categories have been divided into subcategories in his work.

2.2.1 Point Tracking

Objects detected in consecutive frames are represented by points, and the association of the points is based on the previous object state which can include object position and motion. Generally, center points or corners are used as tracked points. This approach requires an external mechanism to detect the objects in every frame.

Point correspondence is a complicated problem- especially in the presence of occlusions, misdetections, entries, and exits of objects. Overall, point correspondence methods can be divided into two broad categories, namely, deterministic and statistical methods. The deterministic methods use qualitative motion heuristics to constrain the correspondence problem. On the other hand, probabilistic methods explicitly take the object measurement and take uncertainties into account to establish correspondence.

2.2.2 Kernel Tracking

Kernel refers to the object shape and appearance. For example, the kernel can be a rectangular template or an elliptical shape with an associated histogram. Objects are tracked by computing the motion of the kernel in consecutive frames. This motion is usually in the form of a parametric transformation such as translation, rotation, and affine.

Kernel tracking is typically performed by computing the motion of the object, which is represented by a primitive object region, from one frame to the next. The object motion is generally in the form of parametric motion (translation, conformal, affine, etc.) or the dense flow field computed in subsequent frames. These algorithms differ in terms of the appearance representation used, the number of objects tracked, and the method used to estimate the object motion. These tracking methods can be into two subcategories based on the appearance representation used, namely, templates and density-based appearance models, and multi-view appearance models.

2.2.3 Silhouette Tracking

Tracking is performed by estimating the object region in each frame. Silhouette tracking methods use the information encoded inside the object region. This information can be in the form of appearance density and shape models which are usually in the form of edge maps. Given the object models, silhouettes are tracked by either shape matching or contour evolution. Both of these approaches can essentially be considered as object segmentation applied in the temporal domain using the priors generated from the previous frames.

Objects may have complex shapes, for example, hands, head, and shoulders that cannot be well described by simple geometric shapes. Silhouette based methods provide an accurate shape description for these objects. The goal of a silhouette-based object tracker is to find the object region in each frame by means of an object model generated using the previous frames. This model can be in the form of a color histogram, object edges or the object contour. Silhouette trackers can be divided into two categories, namely, shape matching and contour tracking. Shape matching approaches search for the object silhouette in the current frame. Contour tracking approaches, on the other hand, evolve an initial contour to its new position in the current frame by either using the state space models or direct minimization of some energy functional.

2.3 Multiple Object Tracking

When tracking multiple objects using Kalman or particle filters, there is a need to associate the most likely measurement for a particular object to that object's state, that is, the correspondence problem needs to be solved before these filters can be applied. The simplest method to perform correspondence is to use the nearest neighbor approach. However, if the objects are close to each other, then there is always a chance that the correspondence is incorrect. An incorrectly associated measurement can cause the filter fail to converge. There exist several statistical data association techniques to tackle this problem. Joint Probability Data Association (JPDA) and Multiple Hypothesis Tracking (MHT) are two widely used techniques for data association. We give a brief description of these techniques in the following.

2.3.1 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. The major limitation of the JPDAF 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. The MHT algorithm, which is explained next, does not have this shortcoming.

2.3.2 Multiple Hypothesis Tracking (MHT)

MHT is an iterative algorithm. Iteration 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 evaluating a distance measure. A set of correspondences (associations) are established for each hypothesis based on the distance measure which introduces new hypotheses for the next iteration. Each new hypothesis represents a new set of tracks based on the current measurements. Note that 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 [19] use a special algorithm to determine best hypotheses in polynomial time for tracking interest points.

CHAPTER 3

MOVING OBJECT DETECTION

In visual surveillance systems, if there is no operation on video data after observed from camera, moving object detection, sometimes called motion segmentation or foreground extraction, is the first step. The following operations, such as object tracking and object classification, take the output of moving object detection module as its input. Therefore, the performance of motion detection algorithm together with the submodules affects the overall performance of the entire system.

3.1 Foreground Segmentation

As mentioned in Chapter 2, there are many works on moving object detection. Among these methods, background subtraction has been chosen due to its simplicity and computational efficiency. Four methods have been implemented as background modeling technique. These implemented methods are: Frame differencing, running (moving) average, eigenbackground subtraction and mixture of Gaussians.

3.1.1 Frame differencing

The easiest and simplest way of detecting moving objects is frame differencing. In this method, background is taken as the previous frame and the difference between the current and the previous frame is thresholded.

$$M(x,y,t)=\begin{cases} 1, & I(x,y,t)-I(x,y,t-1)>Threshold \\ 0, & I(x,y,t)-I(x,y,t-1)<Threshold \end{cases} \tag{3.1}$$

In the formula above, 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.

Although this method is quite fast and can adapt to changes in the scene, it is very sensitive to the threshold value. Also there is an aperture problem as can be seen from Figure 3.2.b. Parts which are extracted as foreground are only the parts near the edges for the large-sized objects.

Instead of previous frame, a mean image of previous N frames can be used as a background image. However, this approach is not memory-efficient.

3.1.2 Running (Moving) Average

The main goal of background modeling is to achieve a background image, even when there are moving regions in the scene. It can be said that all techniques require a background observation (training) time.

In running average method, each pixel is modeled using an adaptive filter. The parameter, α, must be selected as considering the features (size, speed, etc.) of potential moving objects and video.

$$BG_{i+1}=\alpha * FR_{i+1}+(1-\alpha)* BG_i \tag{3.2}$$

As regards to updating parameter, α, it is generally selected about 0.05. α determines the how the background model adapts to changes in the scene such as parked car, left luggage etc.. In addition to determination of updating parameter, the background model can be updated with every new frame as well as by using the matching criteria given below.

$$BG_{i+1}(x,y)=\begin{cases}\alpha * FR_i(x,y)+(1-\alpha)*BG_i(x,y) & \text{, if FR}_\text{i}\text{(x, y) is background}\\ BG_i(x,y) & \text{, if FR}_\text{i}\text{(x, y) is foreground}\end{cases} \tag{3.3}$$

In figure 3.1, video frames (a, b, c, d) taken from an area which has dense traffic and modeled background (e) are shown.

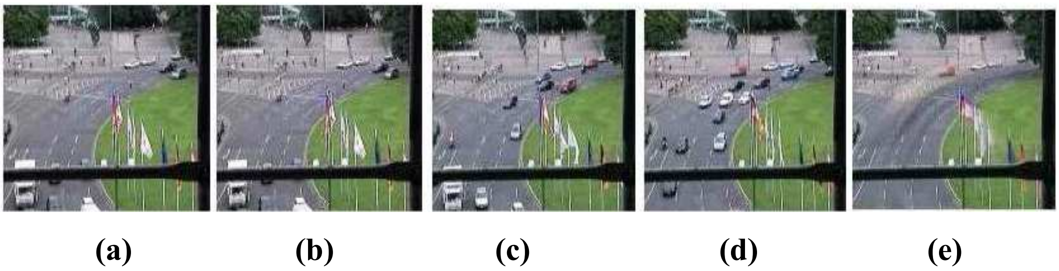


Figure 3.1 Background modeling using running average method (a) 1st frame (b) 20th frame (c) 50th frame, (d) 90th frame and (e) Modeled background

3.1.3 Eigenbackground Subtraction

In this method [16], an eigenspace that models the background is adaptively built. This eigenspace model describes the range of appearances (e.g., lighting variations over the day, weather variations, etc.) that have been observed.

The main idea of this method can be described as: since moving objects do not appear in the same location in the sample N images, they do not have significant contributions to this model. Consequently, the portions of an image containing a moving object cannot be well-described by this eigenspace model, whereas the static portions of the image can be accurately described as a sum of the various eigenbasis vectors. That is, the eigenspace provides a robust model of the probability distribution function of the background, but not for the moving objects.

Dimensionality of the space constructed from sample images is reduced by the help of Principal Component Analysis (PCA). It is proposed that the reduced space after PCA should represent only the static parts of the scene, yielding moving objects, if an image is projected on this space.

The main steps of the algorithm can be summarized as follows [17]:

- A sample of N images of the scene is obtained; mean background image, μ_b, is calculated and mean normalized images are arranged as the columns of a matrix, A.
- The covariance matrix, $C=AA^T$, is computed.
- Using the covariance matrix C, the diagonal matrix of its eigenvalues, L, and the eigenvector matrix, Φ, is computed.
- The M eigenvectors, having the largest eigenvalues (eigenbackgrounds), are retained and these vectors form the background model for the scene.
- If a new frame, I, arrives it is first projected onto the space spanned by M eigenvectors and the reconstructed frame I' is obtained by using the projection coefficients and the eigenvectors.
- The difference I - I' is computed. Since the subspace formed by the eigenvectors well represents only the static parts of the scene, outcome of the difference will be the desired change mask including the moving objects.

3.1.4 Mixture of Gaussians

Background model must be adapted gradual and fast changes in the scene for the complicated environments. Therefore, background model must cope with multi-modal distributions.

In mixture of Gaussians based moving object detection method [2], background is modeled by utilizing the recent history of each pixel, $\{X_1, ..., X_t\}$, with an approximation to a mixture of K Gaussian distributions. The probability of observing the current pixel value given in model is

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \qquad (3.4)$$

where K is the number of distributions, $\omega_{i,t}$ is an estimate of the weight (the portion of the data accounted for by this Gaussian) of the ith 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 where η is a Gaussian probability density function

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)} \qquad (3.5)$$

K is determined by the available memory and computational power. Generally, values ranging from 3 to 5 are used. Also, for computational reasons, the covariance matrix is assumed to be of the form:

$$\Sigma_{k,t} = \sigma_k^2 I \qquad (3.6)$$

This assumes that the red, green, and blue pixel values are independent and have the same variances. While the noise is certainly not spherical, this assumption allows us to avoid a costly matrix inversion at the expense of reduced accuracy. Using a diagonal covariance would allow a Gaussian to represent that a particular channel showed more variation. Using a full covariance matrix would allow each Gaussian to model its local variation with more accuracy. This would be particularly helpful in modeling of variation due to lighting, which varies significantly across the color space.

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

$$\omega_{k,t} = \alpha * M(k,t) + (1-\alpha) * \omega_{k,t-1} \tag{3.7}$$

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

$$\mu_t = (1-\rho)\mu_{t+1} + \rho X_t \tag{3.8}$$

$$\sigma_t{}^2 = (1-\rho)\sigma_{t-1}{}^2 + \rho(X_t - \mu_t)^T (X_t - \mu_t) \tag{3.9}$$

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

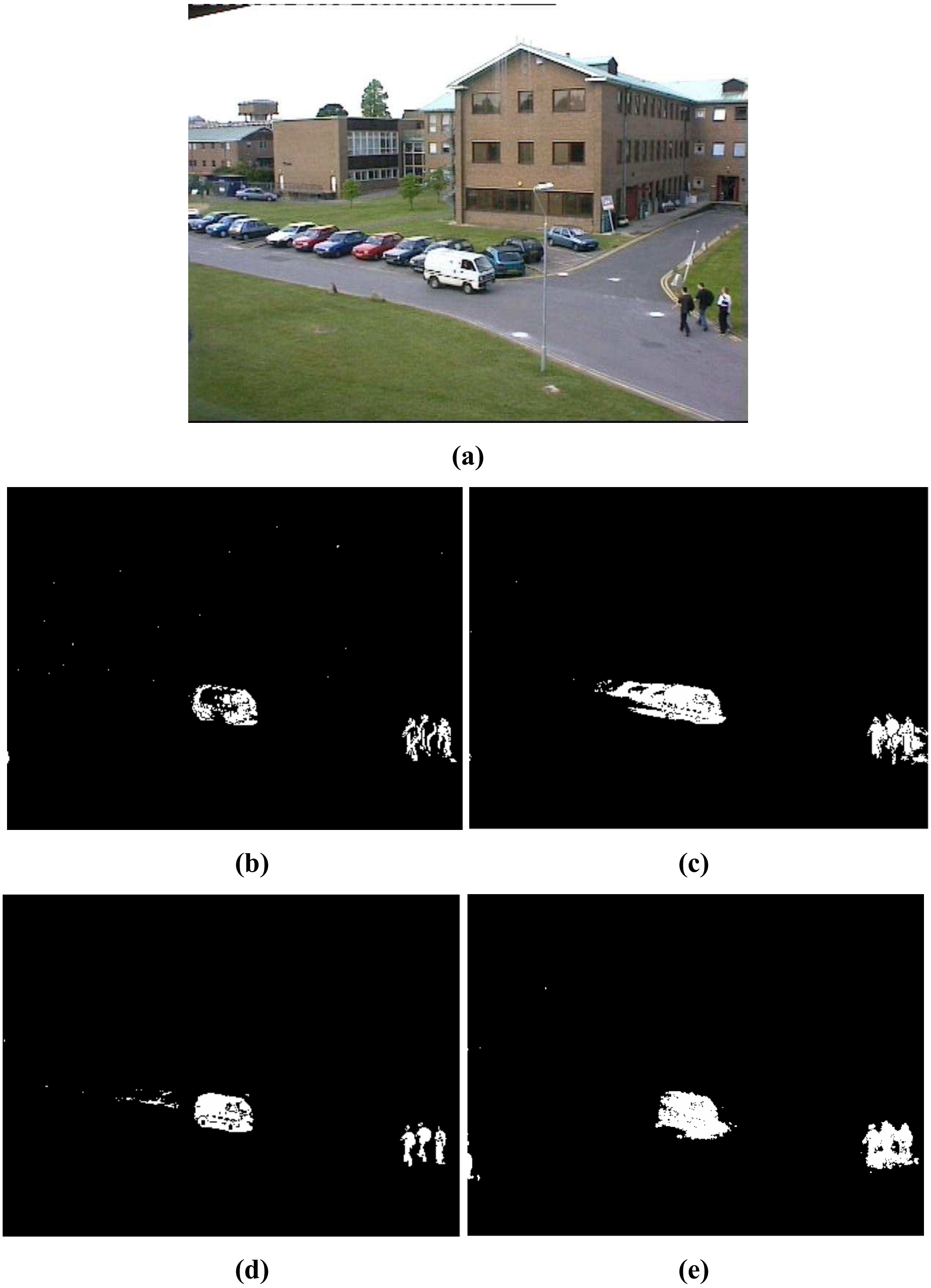
Measurements must be organized such that any changes in the scene must be inserted to the model. For this reason, Gaussians are evaluated by the following approach. First, the Gaussians are ordered by the value of ω/σ. This value increases both as a distribution gains more evidence and as the variance 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

$$B = \arg\min_b \left(\sum_{k=1}^{b} \omega_k > T\right) \tag{3.10}$$

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 simulation results of the methods.



(a)

(b)

(c)

(d)

(e)

Figure 3.2 Results of background subtraction algorithms. (a) Input frame (b) Frame differencing (c) Running average (d) Eigenbackground subtraction (e) Mixture of Gaussians

Frame differencing is the simplest approach to detect moving objects. It is very sensitive to the selected threshold value. Also, small motions which occurred in the previous frame can be regarded as moving objects. It is not suitable for complex environments such as city traffic and airports.

Running average is more complex than frame differencing, but it has also some deficiencies. Running average is not an appropriate approach for the environments where objects have different sizes. Updating parameter, α, can not be optimum for all objects. As can be seen in Figure 3.2.c a bus destroyed the real background, e.g. some part of the bus is considered as background.

Eigenbackground subtraction has a powerful theoretical background when compared to both of the previous methods. The results of eigenbackground subtraction are better than frame differencing and running average as show in Figure 3.2.d. However, there is no updating step of this approach. That is, it is not suitable in complex situations which have dynamic background. Despite its deficiency in updating the environment, it can be a solution for environments which have rare changes such as metro stations and smart rooms.

It can be said that among all of the methods, mixture of Gaussians is the most powerful method. It can adapt to all changes in the scene such as noises, gradual light changes, etc. It models all of the problems as some part of the mixture of Gaussians. Our experiment has shown that mixture of Gaussians gives good results for all environments. Consequently, it has been used as the detection method for the subsequent tracking module.

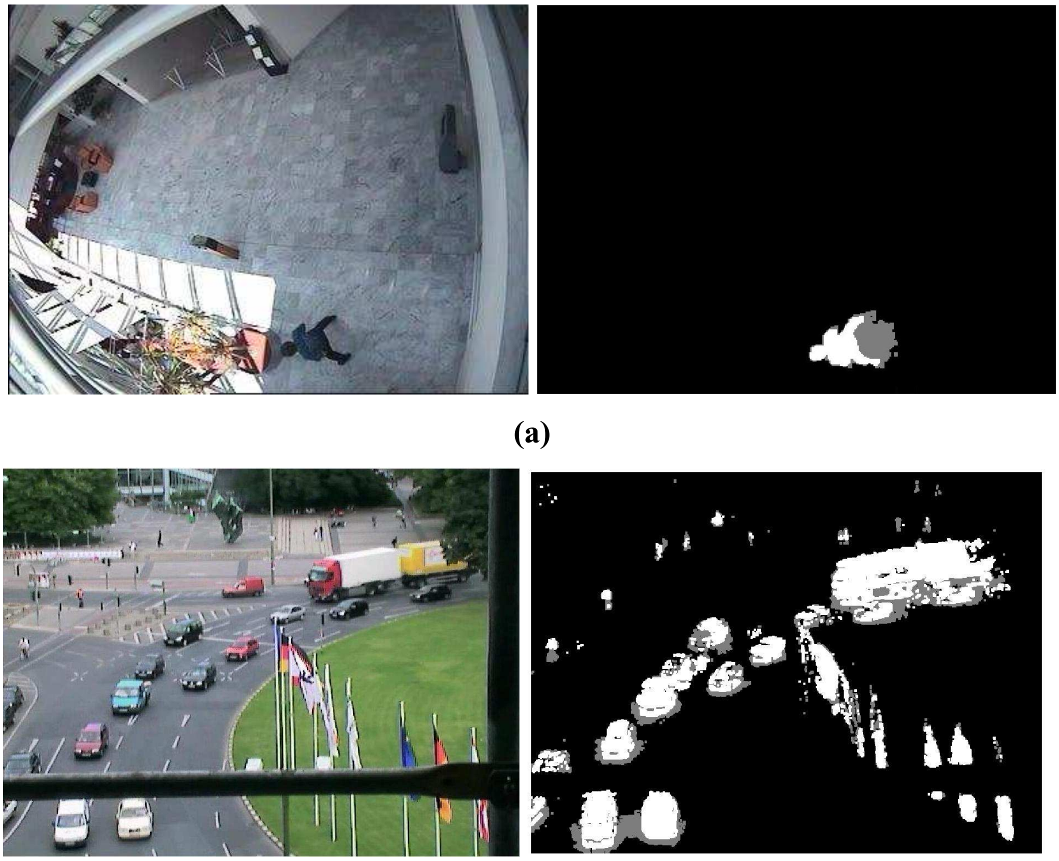
3.3 Suboperations

Outputs of background subtraction process cannot be directly used in the following processes. Shadows and noises are also part of the output. In order to remove these undesired components, a shadow removal algorithm and morphological operations have been applied after background subtraction.

3.3.1 Shadow Removal

In moving object detection process, shadows are also detected as part of the moving objects. This is an expected result, because shadows cause considerable intensity change when compared with the pixel value of background. Shadow can be a critical issue for some surveillance systems such as traffic surveillance systems.

Shadows can be a big problem especially in outdoor environments but also can be for indoor environments. In figure 3.3.a, an example of how shadows can be problem in indoor environments is shown. Depending on the angle of the incoming light, moving objects might be merged because of shadows as can be seen in Figure 3.3.b. Shadows also changes some features of the objects and creates instability related to, for example, center point of objects which will be used in later processes.



(a)

(b)

Figure 3.3 Input images and output images with shadows a) Effect of shadow for indoor environment. b) Merging problems due to shadows

In the literature, several methods can be found for shadow detection problem. Prati [27] has divided these methods into two categories: statistical and deterministic. In this work, we have applied the algorithm which is proposed in [26]. The main idea of the algorithm is that the shadow changes the intensity of the background but normalized intensity values of shadow pixels are approximately the same as background. Results of the algorithm are given in Figure 3.3.

$$\frac{R_s}{R_s+G_s+B_s} \cong \frac{R}{R+G+B} \qquad \frac{G_s}{R_s+G_s+B_s} \cong \frac{G}{R+G+B} \qquad \frac{B_s}{R_s+G_s+B_s} \cong \frac{B}{R+G+B}$$

$$I_s(x,y) = \alpha I(x,y) \qquad (3.11)$$

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.

3.3.2 Morphological Operations

After moving object detection, we have a binary image that indicates the foreground image as well as undesired noises. To get rid of these noises, morphological operations, erosion and dilation have been utilized.

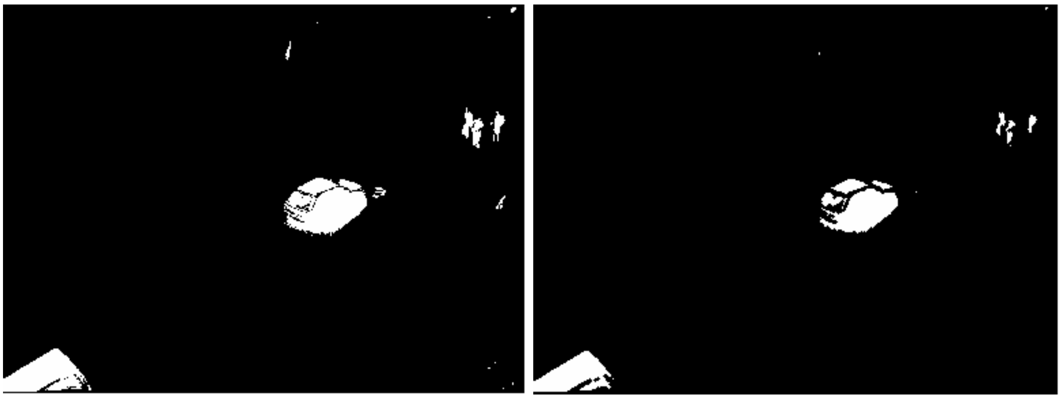
Morphological operations are generally applied in binary images by using a structuring element. Structuring element is a matrix which contains 0's and 1's and is mostly selected in sizes such as 3x3. It is shifted over the image and at each pixel of the image its elements are compared with the ones on the image. If the two sets match the condition defined by the set operator (e.g., if element by element multiplication of two sets exceeds a certain value), 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 background subtraction and shadow removal, respectively.

3.3.2.1 Erosion

The basic effect of erosion operator is to erode the boundaries of the regions for the foreground pixels. A structuring element which has been utilized for this purpose is shown in (3.12). Each foreground pixel in the input image is aligned with the center of the structuring element.

$$SE_{erosion} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad (3.12)$$

If, for each pixel having a value “1” in the structuring element, the corresponding pixel in the image is a foreground pixel, then the input pixel is not changed. However, if any of the surrounding pixels (considering 4-connectedness) belong to the background, the input pixel is also set to the background value. 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.



(a) **(b)**

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. A sample structuring element which has been utilized is given in (3.13).

$$SE_{dilation} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{3.13}$$

The structuring element works on the background pixels instead of foreground pixels, with the same methodology defined in the erosion operator (considering 8-connectedness). After the dilation operation, foreground objects become bigger and also holes inside them shrink.

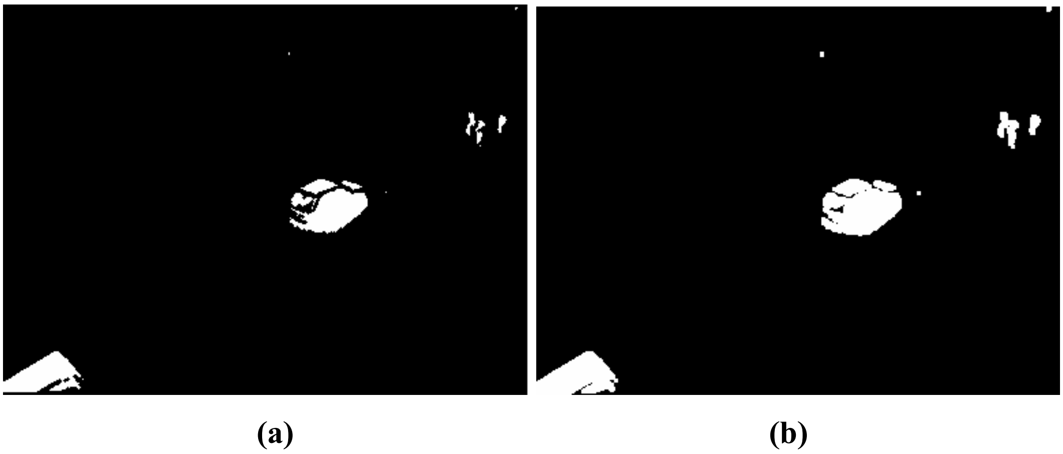


Figure 3.5 Effect of dilation operation (a) Eroded image (b) Dilated image (resulting image of morphological operations)

3.3.3 Connected Component Labeling

One of the most common operations in computer vision is finding the connected components in an image. The points in a connected component form a candidate region for representing an object.

A connected component labeling algorithm finds all connected components in an image and assigns a unique label to all points in the same object. The algorithm [20] which has been used to find the connected components is given below:

1) Run-length encodes the input image.
2) Scan the runs, assigning preliminary labels and recording label equivalences in a local equivalence table.
3) Resolve the equivalence classes.
4) Relabel the runs based on the resolved equivalence classes.

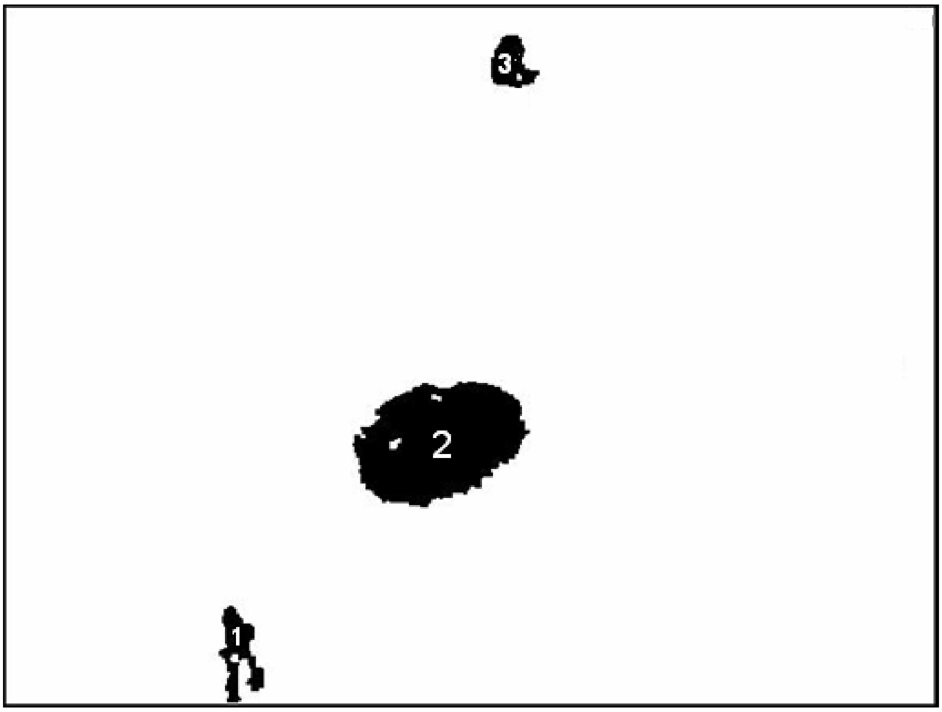


Figure 3.6 Labeling of Connected Components

CHAPTER 4

MOVING OBJECT TRACKING

Moving object tracking is one of the important steps for an automated surveillance system. The increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms. In its simplest form, tracking is defined in [4] as the estimation of the state of a moving object based on remote measurements. In our work, tracking can be defined as of finding associations of moving regions between current and previous frames.

Three methods have been used to track moving objects: Kalman tracker, active contour based tracker and mean-shift tracking. These methods have been evaluated using PETS2001 [9] and PETS2004 [10] data sets.

4.1 Kalman Tracker

Kalman filtering is a popular approach in estimation theory. In this work, Kalman tracker estimates the position of the object using standard Kalman filter with a motion model. Kalman filter has been utilized as a stand-alone tool to track the targets, as well as it has been used as supporting tool in snake tracker and rule-based MHT tracker which will be described in next chapter.

4.1.1 Kalman Filter

The Kalman filter [18] is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown.

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

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{4.1}$$

with a measurement defined by the equation

$$z_k = Hx_k + v_k \tag{4.2}$$

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 assume they are assumed to be constant in our work.

The next state of a process is estimated by 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.

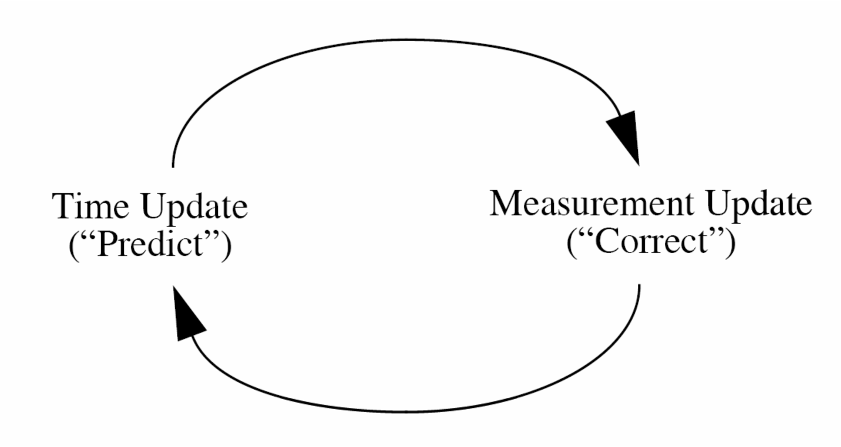


Figure 4.1 The Kalman filter cycle

For these steps of Kalman filter, namely time-update and measurement update, there are equations by which a priori (x_k^-) and posteriori (x_k) estimates are determined.

Time update equations (predict):

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{4.3}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{4.4}$$

Measurement update (correct):

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{4.5}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \tag{4.6}$$

$$P_k = (I - K_k H)P_k^- \tag{4.7}$$

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 the 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.

$$X_k = \left(x_c \;\; y_c \;\; v_x \;\; v_y\right) \tag{4.8}$$

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

$$X_k = AX_{k-1} + \omega_{k-1} \tag{4.9}$$

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

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.

$$Z_k = HX_k + v_k \tag{4.11}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.12}$$

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 given in 4.1.1 have given estimate of next state of object.

4.2 Active Contour (Snake) Tracking

Active contour (snakes) approach has been proposed in [3]. Active contours have been used in several applications in computer vision such as region segmentation, contour extraction.

In our work, snake model proposed by Hamarneh [7] has been utilized and developed. Snakes have been extended to RGB images to track the moving objects after detection of moving objects. In fact, snake tracker differs from Kalman tracker due to following reason: Kalman tracker waits for results that come from detection module. However, once snake tracker detects the moving object, the snake is initialized and it works directly on input image without output of moving object detection module.

4.2.1 Introduction to Active Contour (Snake) Concept

Active contours, belonging to the class of deformable models, have gained large acceptance as a segmentation tool. This is due to a collection of factors including the way snakes consider the boundary as a single, inherently connected, and smooth structure. Snakes also support intuitive interactive mechanisms for guiding the segmentation deformations. Many variations to the original snake formulation have been proposed to improve their performance. Snakes are energy minimizing parametric contours with smoothness constraints deformed according to image data. Snakes are designed to be semi-automatic tools supporting intuitive interactive mechanisms for guiding the segmentation deformations. Some of the problems of the classical snakes are initialization sensitivity and lack of high level automatic control that cause the snakes, for example, to leak or latch to erroneous edges.

In active contour models, a contour is initiated on the image and is left to deform in a way that, firstly, moves it toward the features of interest in the image and, secondly, maintains a certain degree of smoothness and continuity in the contour.

In order to favor this type of contour deformation, an energy term is associated with the contour and is designed to be inversely proportional to the contour's smoothness and the fit to desired image features.

The deformation of the contour in the image plane will change its energy, thus one can imagine an energy (potential) surface on top of which the contour moves (in a way that resembles the slithering of a snake and hence the name) seeking valleys of low energy. This can also be formulated using a force field that causes this energy change (analogous to physical systems). Moreover, certain forces can be designed (or derived from energy terms) in a way that the resulting contour deformations will reduce its energy, thus yielding a smooth contour located along desired image features such as edges.

A snake in the continuous spatial domain is represented as a 2D parametric contour curve $\mathbf{v}(s) = (x(s), y(s))$ where $s \in [0,1]$. In order to fit the snake model to the image data we associate energy terms with the snake and aim to deform the snake in a way that minimizes its total energy. The energy of the snake, ξ, depends on both the shape of the contour and the image data $I(x,y)$ reflected via the internal and external energy terms, $E_{int}(v)$ and $E_{ext}(v)$, respectively. The total snake energy is written as

$$E_{total} = E_{int}(v) + E_{ext}(v) \tag{4.13}$$

The internal energy term is given as

$$E_{\text{int}}(v) = \int_0^1 w_1(s)\left|\frac{\partial v}{\partial s}\right|^2 + w_2(s)\left|\frac{\partial^2 v}{\partial s^2}\right|^2 ds \tag{4.14}$$

The weighting functions w_1 and w_2 control the tension and flexibility of the contour, respectively. The external energy term is given as

$$E_{ext}(v) = \int_0^1 P(v(s))ds \qquad (4.15)$$

For the contour to be attracted to image features, the function *P*(x,y) is designed such that it has minima where the features have maxima. For example, for the contour to be attracted to high intensity changes (high gradient values) we can choose

$$P(x,y) = -c\left\| \nabla \left[G_\sigma * I(x,y) \right] \right\| \qquad (4.16)$$

where G_σ*I denotes the image convolved with a smoothing (e.g. Gaussian) filter with a parameter σ controlling the extent of the smoothing (e.g. variance of Gaussian).

4.2.2 Application of Active Contour (Snake) to Object Tracking

In our implementation a polygonal discrete active contour model is used and is represented by a set of nodes or vertices.

$$v_i(t) = \left[x_i(t), y_i(t) \right] \qquad (4.17)$$

where i = 1,2, …, N is the node number and t denotes the time or iteration number. The snake deformation is performed iteratively thanks to the minimization of an energy function. As defined in (4.13), the energy is composed of internal and external energies.

The internal energy is itself composed of two energies linked to flexural and tensile forces. The definition of the internal energy is

$$F_i^{\text{int}\,ernal}(t) = \alpha F_i^{tensile}(t) + \beta F_i^{flexural}(t) + \gamma \dot{v}_i \qquad (4.18)$$

where α, β and γ are weighting factors. $F_i^{tensile}(t)$ is a tensile force (resisting stretching) acting on node i at time t and is given by

$$F_i^{tensile}(t) = 2v_i(t) - v_{i-1}(t) - v_{i+1}(t) \tag{4.19}$$

$F_i^{flexural}(t)$ is a flexural force (resisting bending) and is given by

$$F_i^{flexural}(t) = 2F_i^{tensile}(t) - F_{i-1}^{tensile}(t) - F_{i+1}^{tensile}(t) \tag{4.20}$$

$F_i^{external}(t)$ is an external (image-derived) force. It is derived in a way that causes the snake node to move towards regions of higher intensity gradient in the image and is given by

$$F_i^{external}(t) = \nabla P(x_i(t), y_i(t)) \tag{4.21}$$

where $P(x_i(t), y_i(t))$ is given in (4.16). $F_i^{external}(t)$ is a mixture of R, G, B spaces. Deformation of the snake is done by equating internal and external forces. Therefore, main snake equation is

$$\alpha F_i^{tensile}(t) + \beta F_i^{flexural}(t) + \gamma \dot{v}_i = F_i^{external}(t) \tag{4.22}$$

Tracking of objects is done by firstly initializing the snake around the object. Then, for each frame, snake is deformed by using equations given before. Snake initialization has been done by utilizing background subtraction for images taken from stationary cameras. On the other hand, snake initialization had to be done manually for images taken from moving cameras. Once the snake is deformed for the first image, there is no need to the background subtraction operation. Our final snake in the first image is the initial snake for the second image. This assumption is valid if an acceptable amount of object is in the area covered by our final snake in second image. That is, object speed must be taken into consideration. If object speed is high, Kalman filter would be solution to initialize the snake for the next frame. The state vector of the Kalman filter is the average point and speed of this average point of our snake nodes. x_{snake} is the average of x and y coordinates of the snake nodes.

$$x_{snake} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{4.23}$$

After the calculation of the estimation of the x_{snake}, all nodes of the snake is shifted by the amount of

$$\text{Shift} = x_{snake,estimate} - x_{snake} \tag{4.24}$$

4.3 Mean-Shift Tracking

Mean shift tracking is a kernel based method for tracking moving objects in video [28]. Mean-shift tracking can be an important tool in any tracking system. It is based on normalized and smoothed color histogram of the moving objects. Color histogram smoothing is essentially equivalent to so-called kernel density estimation process described in [28]. In our work, we try to express mean-shift tracking briefly. Details of the mean-shift concept and mean-shift tracking can be found in [28].

In mean-shift tracking, next state of target is found by comparing the histogram of the target in the current frame and histogram of candidate regions in the next image frame. The mean shift iterations are employed to find the target candidate that is the most similar to a given target model, with the similarity being expressed by a metric based on the Bhattacharyya coefficient.

Discrete density, **q** = {q$_u$} u=1…m is estimated from the m-bin histogram of target model, while **p(y)** = {p$_u$(y)} u=1…m is estimated from the m-bin histogram of target candidate. The sample estimate of the Bhattacharyya coefficient is given by

$$\rho(y) \equiv \rho[p(y),q] = \sum_{u=1}^{m} \sqrt{p_u(y) q_u} \tag{4.25}$$

and the Bhattacharyya coefficient is maximized by the distance measure

$$d(y) = \sqrt{1-\rho[p(y),q]} \tag{4.26}$$

In order to find the maximum of the Bhattacharyya coefficient, the mean-shift vector which is the estimate of gradient vector's density function should be traced. When mean-shift vector converges to a certain region y$_0$, which maximizes the Bhattacharyya coefficient, y$_0$ becomes the new location of the target object in the next frame.

4.4 Results and Discussions

Three methods have been implemented to track the moving objects. Kalman tracker has utilized Kalman filter by setting state vector as center position and speed of center. In this chapter, we have only tested the power of Kalman filter in estimation of next state of targets. There is no association step for this chapter.

Kalman tracker has given good results when object motion is linear. Other Kalman filters (extended, unscented) and more complex state vectors can be selected for more complicated situations. In Figure 4.2 estimated (blue) and real (green) trajectory of a moving object is shown.

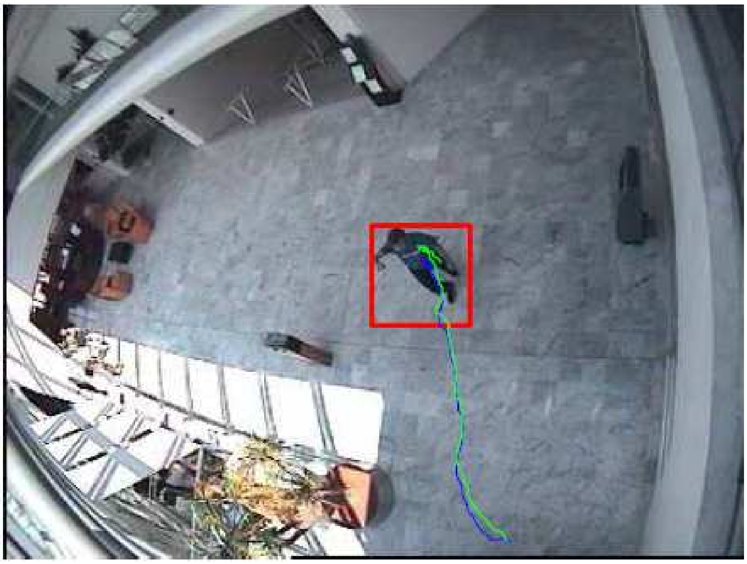
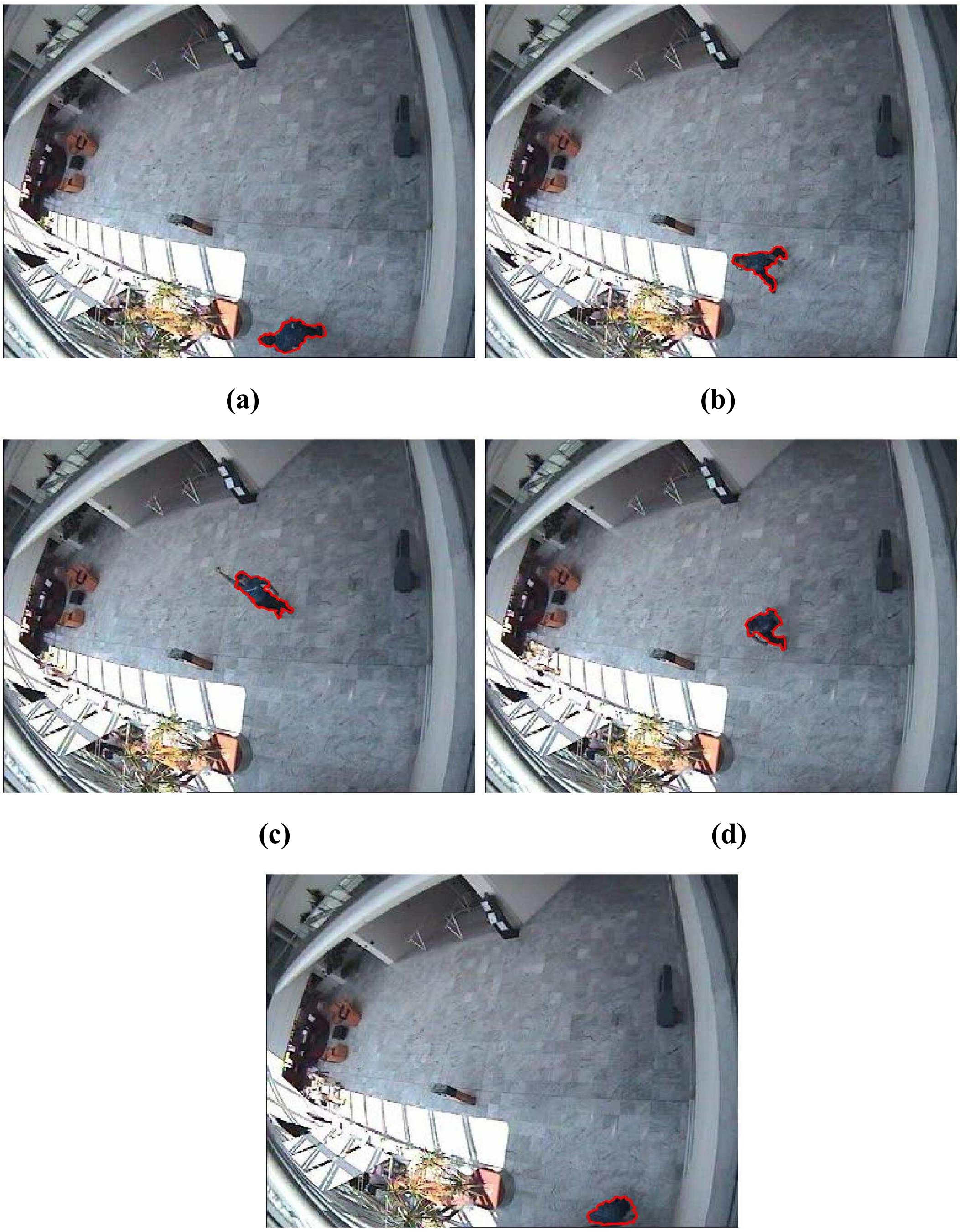


Figure 4.2 Kalman Tracker: Estimated and real trajectories of a moving object

Snake tracker is basically a solution to the problem of target localization problem. That is, snakes have been utilized to solve the problem of where target is and there is no estimation step. However, Kalman filter is again a tool for estimation in the snake tracker for the cases in which object speed is high. Actually, we have tested snake tracker with and without Kalman filter for stationary camera. For both cases, snake tracker gives sufficiently good results. In addition to stationary camera, snake tracker has been tested for moving camera. In figures 4.3 and 4.4, simulation results of snake tracker for the PETS04 sequence and a moving camera sequence have been given, respectively.



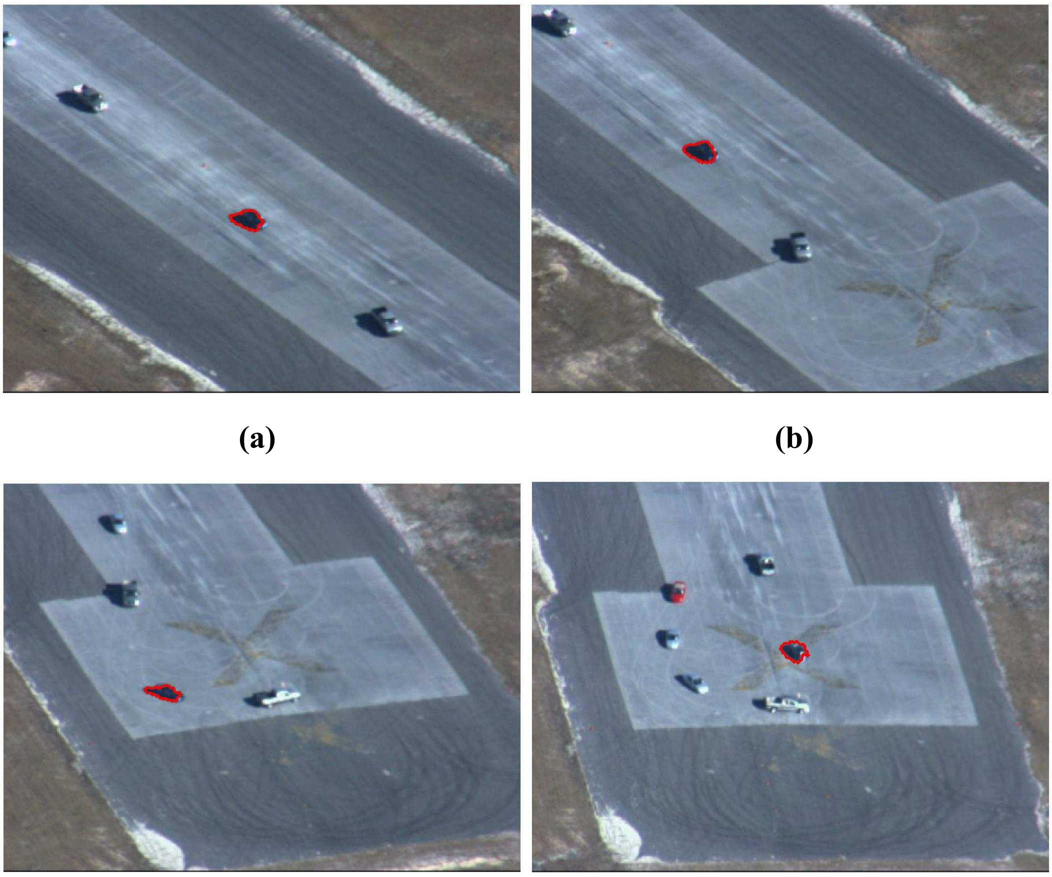
(a)

(b)

(c)

(d)

Figure 4.3 Simulation results of snake tracker for PETS04 sequence (a) 25th frame (b) 43rd frame (a) 83rd frame (a) 107th frame

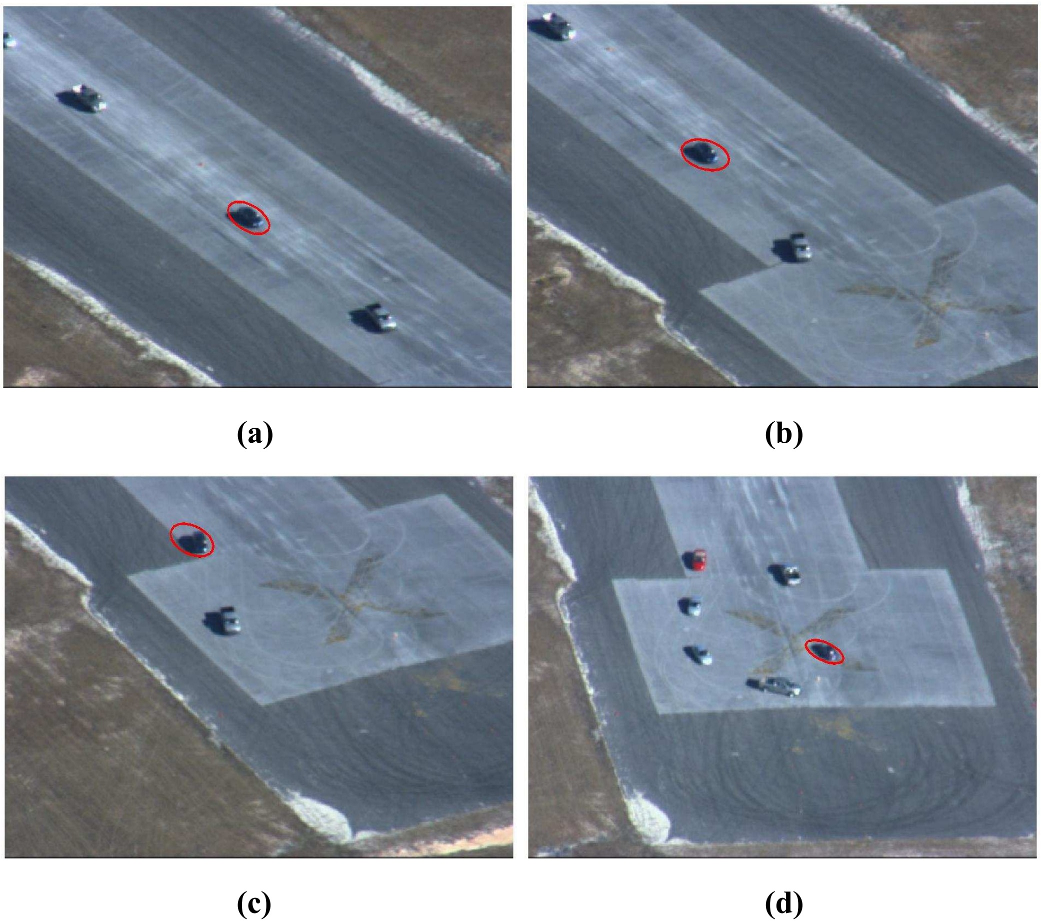


(a) (b)

(c) (d)

Figure 4.4 Simulation results of snake tracker for a sequence taken from moving camera (a) 1st frame (b) 55th frame (c) 240th frame (d) 430th frame

Mean-shift tracker is a color tracker and works well when object is visible. We can say that it is more robust to partial occlusions when compared with Kalman and snake trackers. In addition, mean-shift tracker has been tested for moving cameras. Simulation results of mean-shift tracking for moving and stationary cameras have been given in Figure 4.5 and 4.6, respectively.



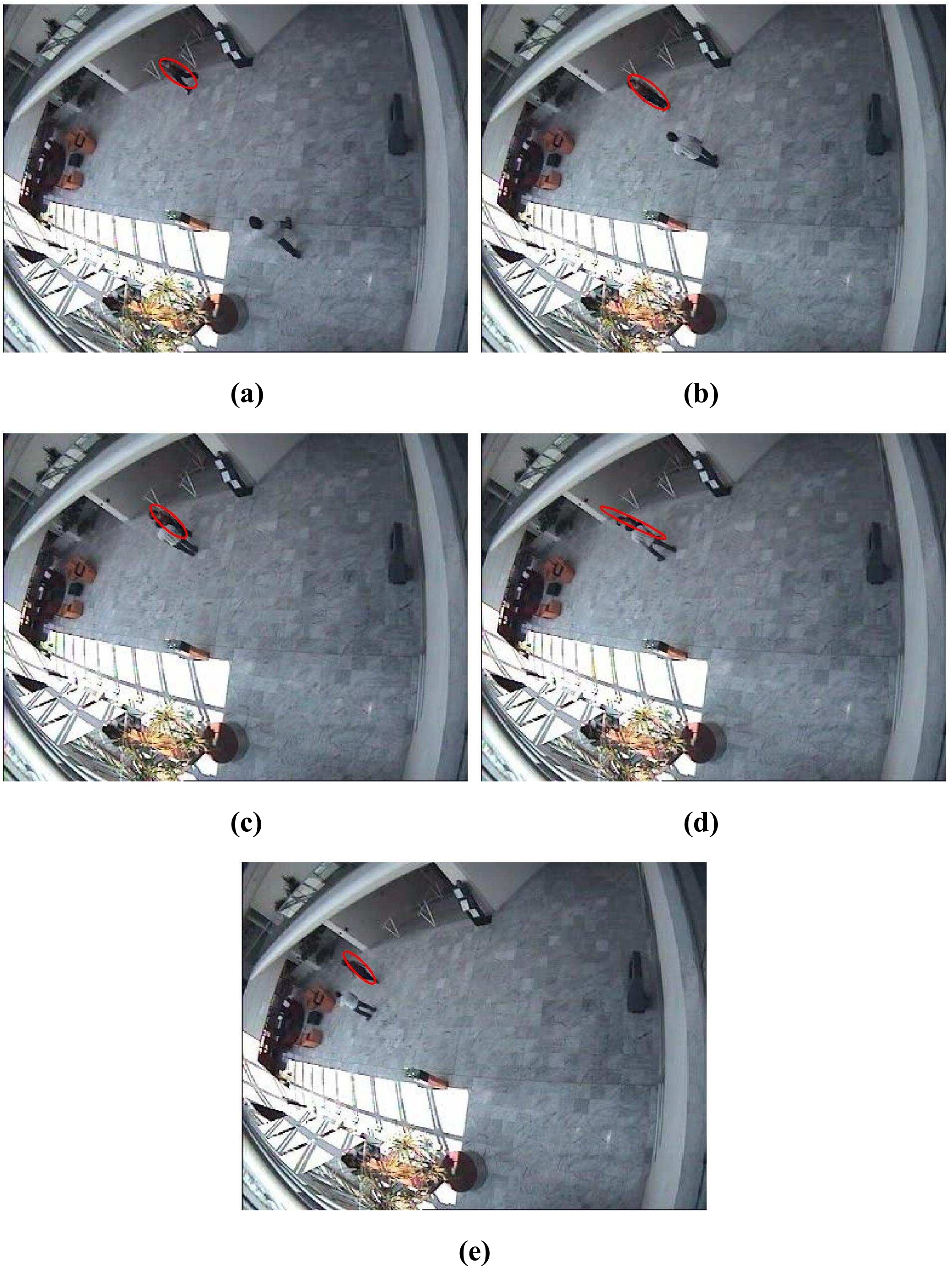
(a)

(b)

(c)

(d)

Figure 4.5 Simulation results of mean-shift tracker for a sequence taken from moving camera (a) 1st frame (b) 32nd frame (c) 113rd frame (d) 408th frame



(a)

(b)

(c)

(d)

(e)

Figure 4.6 Simulation results of mean-shift tracker (a) 1st frame (b) 36th frame (c) 83rd frame (d) 110th frame (e) 151st frame

CHAPTER 5

RULE-BASED MULTIPLE HYPOTHESIS TRACKING

Visual tracking is a concept which concerns the problem of target localization, i.e., where the target is for a specified time. In visual tracking, if there is only one target in the scene, there may be no need for a special tracking algorithm to determine the features of the target. However, for multi-target case, there must be a special algorithm to associate the targets in the current frame to the targets in previous frames.

5.1 Standard MHT Algorithm

The multiple hypothesis tracking algorithm was originally developed by Reid [29] in the context of multi-target tracking. MHT approach is measurement oriented in the sense that the probability that an established target or a new target give rise to a certain measurement sequence is obtained.

MHT approach considers the association of sequence of measurements and evaluates the probabilities of all the sequences (i.e., hypotheses). This leads to a complexity that increases exponentially with time and appropriate techniques, such as merging and pruning have to be used to limit the number of hypotheses to be constructed.

There are two basic approaches to MHT implementation. The first (hypothesis-oriented) approach follows the original work of Reid [29]. It maintains the hypothesis structure from frame (scan) to frame and continually expands and cuts back (i.e., prunes) the hypotheses as new data are received. At each frame a set of hypotheses will be carried over from the previous frame and composed of one or more tracks that are compatible with all other tracks in the hypothesis. Compatible tracks are defined to be the tracks that do no share any common observation. Then, on the receipt of new data, each hypothesis is expanded into a set of new hypotheses by considering all observation-to-track assignments for the tracks within the hypothesis. Again, as new hypotheses are formed, the compatibility constraint for tracks within a hypothesis is maintained. Direct, but inefficient, hypothesis expansion methods are given in [29] but the use of Murty's algorithm [19] will reduce the number of low probability hypotheses that are unnecessarily formed and to be deleted later.

The alternative (track-oriented) approach [32] does not maintain hypotheses from frame to frame. The tracks that are formed on each frame are reformed into hypotheses and the tracks that survive the pruning are predicted to the next frame where the process continues. A major advantage of the track-oriented approach is that hypothesis formation can be restricted to higher quality tracks. Low score tracks are deleted before hypotheses are formed. This feature of track-oriented approach reduces the computational load. Hence, track-oriented MHT has been applied to our problem.

The track-oriented approach to MHT starts by independently forming tracks. Using this approach, observations are formed into tracks without imposing the usual constraints that an observation not be used to update more than one track and that a track not be updated by more than one observation one the same frame. The tracks that are formed may not be consistent with each other; i.e., two tracks may both use the same observation. These inconsistencies are resolved through the formation and evaluation of hypotheses composed of sets of consistent tracks. To satisfy computational constraints and produce information that can readily be interpreted by a user, it is necessary to limit the number of hypotheses. The basic method of doing this is to delete (or prune) unlikely tracks. Pruning is performed at two stages. First, individual track hypotheses are compared with the hypothesis that all included observations are false alarms. Then, tracks that survive this first test (versus the false alarm option) are compared at the global level by formation, evaluation and pruning hypotheses. In upcoming subsections, elements of MHT algorithm such as pruning and merging, have been described briefly.

5.1.1 Track Formation and Maintenance

Track formation and maintenance constitutes the central track file where all tracks and the operations that are performed on those tracks are maintained. As new observations are received, gating has been exploited to determine the feasible observation-to-track pairings.

Existing tracks are updated with all observations within the gates, and extrapolated tracks (that are not updated with any current observation) are formed. Also, essentially all observations are used to form the first point of a new track. Thus, a large number of tracks are potentially formed and many of the tracks are inconsistent in the sense that same observations are used for more than one track.

The formation of too many tracks can lead to excessive computer storage requirements or to unacceptable computational time requirements. Thus, a fail-safe logic is required to limit the number of tracks that are formed as new data are received.

Track compatibility is required for the purposes of hypothesis formation. Tracks are compatible when they have no observations in common. A significant improvement can be achieved by maintaining the results of previous incompatibility tests from frame to frame. Thus, there is, in effect, for each track a list of tracks that are incompatible with that track. This incompatibility is passed along to descendant tracks that are spawned from a parent track.

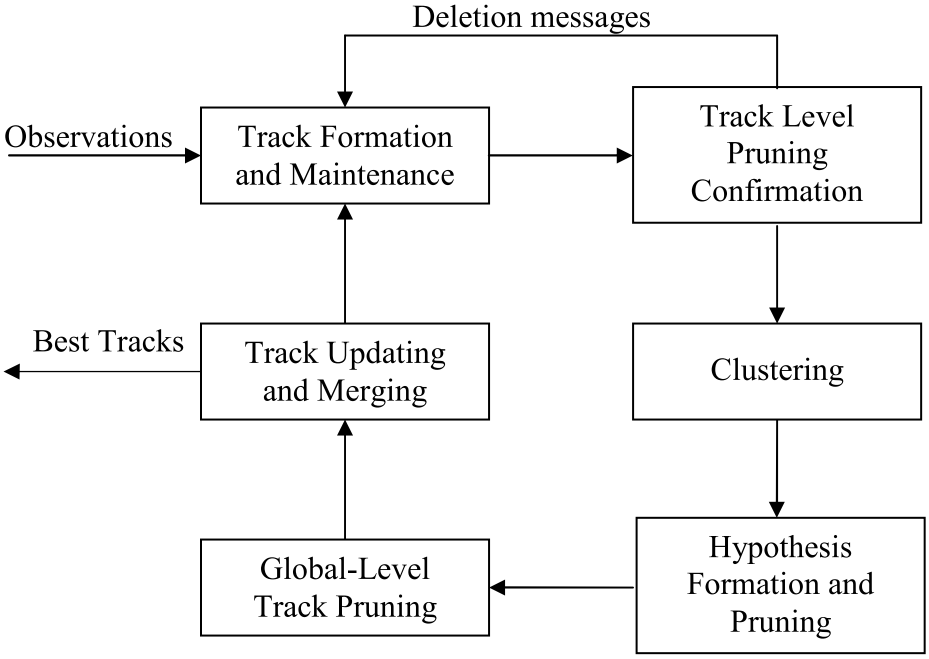

Deletion messages
Observations
Track Formation and Maintenance
Track Level Pruning Confirmation
Best Tracks
Track Updating and Merging
Clustering
Global-Level Track Pruning
Hypothesis Formation and Pruning

Figure 5.1 General flow diagram of standard MHT algorithm

5.1.2 Pruning and Confirmation

Each track has a track-level probability and resultant track score that is the likelihood ratio. The track-level pruning process just compares the track-level probability versus a suitably chosen deletion threshold. The tracks that fail this test are deleted and the surviving tracks are tested for confirmation and then passed to the next stage, which is clustering. An additional computational saving can be achieved by only allowing further processing to be performed on confirmed tracks. Also, track confirmation status is used later to determine the eligibility of tracks for presentation to the user.

5.1.3 Clustering

The process of clustering is the collection of all tracks that are linked by common observations. Tracks that share observations are defined to be incompatible and a record of incompatible tracks is maintained from frame to frame. This record is updated as tracks are deleted and as new tracks are formed from the current frame’s observations.

A cluster can include tracks that do not share observations directly but that both share observations with a third track. Thus, if track 1 shares an observation with track 2 and track 2 shares another observation with track 3, all three tracks are in the same cluster.

The formation of clusters with large numbers of tracks can lead to an unacceptable amount of time required by hypothesis formation. Thus, several techniques are employed in order to maintain clusters that contain no more than several hundred tracks. The result of clustering is a list of tracks that are interacting (linked through common observations). These tracks are ranked in order of track score. The next step is to form hypotheses of compatible tracks.

5.1.4 Hypothesis Formation and Pruning

Multiple track hypotheses are formed to represent the multiple targets in the scene. Hypotheses are defined to be sets of consistent (compatible) tracks in the sense that no two tracks within a given hypothesis share observations. There can theoretically be any number of tracks within a hypothesis.

A search routine is required to find the hypothesis that represents the most likely collection of tracks. A relatively straightforward breadth-first approach to hypothesis formation starts with the search process by the definition of one-track hypotheses and expands the hypotheses by adding new tracks to existing hypotheses. The new tracks that are added to any hypothesis as the hypothesis is expanded cannot share observations with any tracks in the existing hypothesis. This can be accomplished directly because each track has an incompatibility list, so that an incompatibility list can be inferred for the hypothesis as a whole.

Each subsequent step of hypothesis generation process begins with a set of N-track hypotheses (starting with N = 1) and expands a subset of these hypotheses into (N+1)-track hypotheses. This process is continued until the potential scores that are associated with further expansion are no longer deemed adequate to justify expansion. Initially, this expansion should be done using only positive (high) score tracks. Then negative (low) score tracks can be evaluated by their compatibility with the higher score hypotheses that were formed from positive score tracks.

5.1.5 Global-Level Track Pruning

The a posteriori probability of a given track can be computed as the sum of the probabilities of all the hypotheses containing that track. Some tracks, for example, may have only been contained in hypotheses that were deleted. Thus, since these tracks are contained in no surviving hypotheses, they will be computed (as an approximation) to have probability zero and can be immediately deleted. Also, each track whose probability is below a deletion threshold is removed from track file. Finally, an N-scan pruning approach is used to delete selected confirmed tracks.

5.1.6 Track Updating and Merging

Filtered state and covariance estimates are formed for those tracks that survive pruning. This computationally demanding Kalman filtering step should not be performed until poor tracks are deleted by pruning. Tracks that potentially share observations will have been identified during clustering. Merging logic is performed to determine which of these tracks are redundant representations of the same target. Merging rules are defined to use both common observation history and similar state vectors to identify these tracks that should be merged.

Once two tracks are determined to be similar, the track with the higher a posteriori probability is retained and other track is deleted. Thus, a single track now takes the place of two tracks that previously represented essentially the same potential target. An increment to the score of the retained track is also made in order to account for the probability of the track that is deleted.

Merging is the last logical operation performed in order to reduce the number of tracks that re to be maintained. Tracks that survive pruning and merging steps are predicted ahead to the time of the next observation data and process continues.

5.2 Introduction to Multi-View Geometry

In this section, a brief introduction to the multi-view geometry and more specifically 2D homography is discussed. Our main concern is to find the correspondence between two cameras. For this reason 2D homography idea has been utilized.

5.2.1 2D Homography

Homography is a projective transformation that maps each x_i in IP^2 to x_i' in IP^2 [30]. We consider a set of point correspondences $x_i \Leftrightarrow x_i$' between two images. Our problem is to compute a 3 x 3 matrix H such that $Hx_i = x_i$' for each i.

The first question to consider is how many corresponding points $x_i \Leftrightarrow x_i$' is required to compute the projective transformation H. A lower bound is available by a consideration of the number of degrees of freedom and number of constraints. On the one hand, the matrix H contains 9 entries, but is defined only up to scale. Thus, the total number of degrees of freedom in a 2D projective transformation is 8. On the other hand, each point-to-point correspondence accounts for two constraints, since for each point x_i in the first image the two degrees of freedom of the point in the second image must correspond to the mapped point Hx_i. A 2D point has two degrees of freedom corresponding to the *x* and *y* components, each of which may be specified separately. Alternatively, the point is specified as a homogeneous 3-vector, which also has two degrees of freedom since scale is arbitrary. As a consequence, it is necessary to specify four point correspondences in order to constrain H fully.

5.2.2 The Direct Linear Transformation (DLT) Algorithm

Direct linear transformation is a simple linear algorithm for determining H given a set of four 2D to 2D point correspondences, $x_i \Leftrightarrow x_i'$. The transformation is given by the equation $x_i' = Hx_i$. This is an equation involving homogeneous vectors; thus the vectors x_i' and Hx_i are not equal, they have same direction but may differ in magnitude by a nonzero scale factor. The equation may be expressed in terms of the vector cross product as $x_i' \times Hx_i = 0$. This form will enable a simple linear solution for H to be derived.

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

$$Hx_i = \begin{pmatrix} h^{1T}x_i \\ h^{2T}x_i \\ h^{3T}x_i \end{pmatrix} \tag{5.1}$$

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

$$\mathrm{x}_i' \times \mathrm{Hx}_i = \begin{pmatrix} y_i'\mathrm{h}^{3T}\mathrm{x}_i - w_i'\mathrm{h}^{2T}\mathrm{x}_i \\ w_i'\mathrm{h}^{1T}\mathrm{x}_i - x_i'\mathrm{h}^{3T}\mathrm{x}_i \\ x_i'\mathrm{h}^{2T}\mathrm{x}_i - y_i'\mathrm{h}^{1T}\mathrm{x}_i \end{pmatrix} \tag{5.2}$$

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

$$\begin{bmatrix} 0^T & -w_i'x_i^T & y_i'x_i^T \\ w_i'x_i^T & 0^T & -x_i'x_i^T \\ -y_i'x_i^T & -x_i'x_i^T & 0^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \tag{5.3}$$

These equations have the form $A_i\mathbf{h} = 0$, where A_i is a 3x9 matrix and h is a 9-vector made up of the entries of matrix H,

$$h = \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix}, \qquad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \tag{5.4}$$

with h_i is the i-th element of h. Although there are three equations in (5.3), two of them are linearly independent. Thus each point correspondence gives two equations in the entries of H. It is usual to omit the third equation in solving for H. Then, the set of equations becomes

$$\begin{bmatrix} 0^T & -w_i'x_i^T & y_i'x_i^T \\ w_i'x_i^T & 0^T & -x_i'x_i^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \tag{5.5}$$

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.* Since the homography matrix has 9 entries and 8 degrees of freedom, 8 equations are needed to solve for *H.* For each point correspondence, one has 2 equations, hence, 4 point correspondence is enough to solve for *H.* Then, the algorithm is

(i) For each correspondence xi ⇔ xi′, compute the matrix A_i from (**5.3**). Only the first two rows are required in general.

(ii) Assemble the n 2x9 matrices A_i into a single 2nx9 matrix A.

(iii) Obtain SVD of A. The unit singular vector corresponding to the smallest singular value is the solution h. Specifically, if $A=UDV^T$ with D diagonal with positive diagonal entries, arranged in descending order down the diagonal, then 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 [9]. Homography matrix has been calculated using four points which is taken from each image. In Figure 5.2, transfer of five points from left image to right image has been shown.

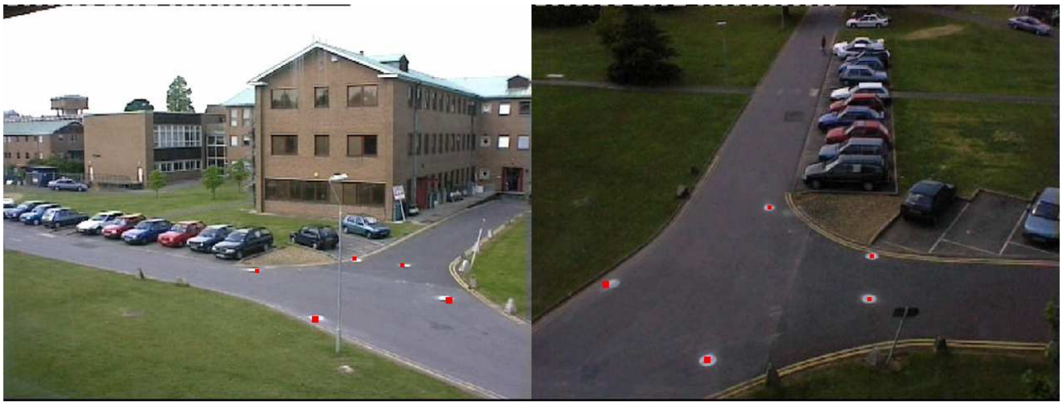


Figure 5.2 Point transfer by using homography matrix

5.3 Rule-Based MHT Algorithm

A novel method which concatenates basic steps of multiple hypotheses tracking (MHT) and fuzzy logic has been proposed. Main idea of MHT is used to determine whether a measurement is an existing target, or a new target. Evaluation of these hypotheses has been done utilizing fuzzy logic.

5.3.1 Main Idea of Rule-Based MHT

In this section, we have presented a novel method for single camera case. Some categories have been defined related with measurements. Each measurement may either belong to a previously known target or be the start of a track, e.g., a previously unseen object that has entered the field of view of the camera.

Assignments or events can be generated by creating an ambiguity or hypothesis matrix. In this matrix, each measurement is represented by a row of matrix and known targets are represented by the columns of matrix. To detect new objects or false alarms an extra column is inserted into the hypothesis matrix. As regards the situation given in Figure 5.3, our hypothesis matrix will be:

$$Hypothesis\ Matrix = \begin{array}{cccc} T1 & T2 & N & \\ \left[\begin{matrix} 1 \\ 1 \\ 0 \end{matrix}\right. & \begin{matrix} 1 \\ 1 \\ 0 \end{matrix} & \left.\begin{matrix} 1 \\ 1 \\ 1 \end{matrix}\right] & \begin{matrix} M1 \\ M2 \\ M3 \end{matrix} \end{array}$$

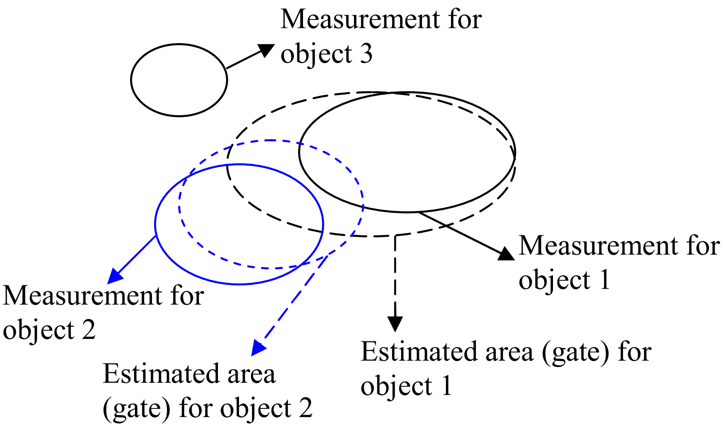


Figure 5.3 Predicted target locations and measurement of two known objects and one new measurement

Each “1” indicates that there is a possibility of measurement belongs to target. For example, generated hypotheses for the scenario given above are:

- Measurement1 belongs to Target1 and Measurement2 belongs to Target2 and Measurement3 belongs to new target.
- Measurement2 belongs to Target1 and Measurement1 belongs to Target2 and Measurement3 belongs to new target.
- Measurement1 belongs to new target and Measurement2 belongs to new target and Measurement3 belongs to new target.
- Measurement1 belongs to Target1 and Measurement2 belongs to new target and Measurement3 belongs to new target.
- Measurement1 belongs to new target and Measurement2 belongs to Target2 and Measurement3 belongs to new target.
- Measurement1 belongs to Target2 and Measurement2 belongs to new target and Measurement3 belongs to new target.
- Measurement2 belongs to Target1 and Measurement1 belongs to new target and Measurement3 belongs to new target.

For each frame, several hypotheses are generated and evaluated. Fuzzy logic has been used to evaluate these hypotheses. That is, fuzzy rules have been applied to determine category of each measurement, i.e., whether it belongs to known target(s), new object or false alarms. Membership functions used in fuzzy evaluation are related to bounding box overlapping, Euclidean distance and color histogram similarity.

Bounding box overlapping is a measure of how the estimated bounding box and a measurement are overlapped. Euclidean distance is the distance between the estimated center position and the center position of measurement. For color histogram similarity, Bhattacharyya coefficient has been utilized. Bhattacharyya coefficient has been calculated for the histogram of measurement and histogram of tracks. Basic fuzzy rules applied to obtain track scores are given below:

- If bounding box overlapping is high, track score is high
- If bounding box overlapping is medium, track score is medium.
- If bounding box overlapping is low, track score is low.
- If color similarity is high, track score is high
- If color similarity is medium, track score is medium.
- If color similarity is low, track score is low.
- If Euclidean distance is high, track score is low
- If Euclidean distance is medium, track score is medium.
- If Euclidean distance is low, track score is high.

These rules are the cues of the general concept. The working rules are the accumulation of these basic rules. Basic rules are combined with "AND" logical operator and "min" method is selected as "AND" method. Resulting rules are listed below:

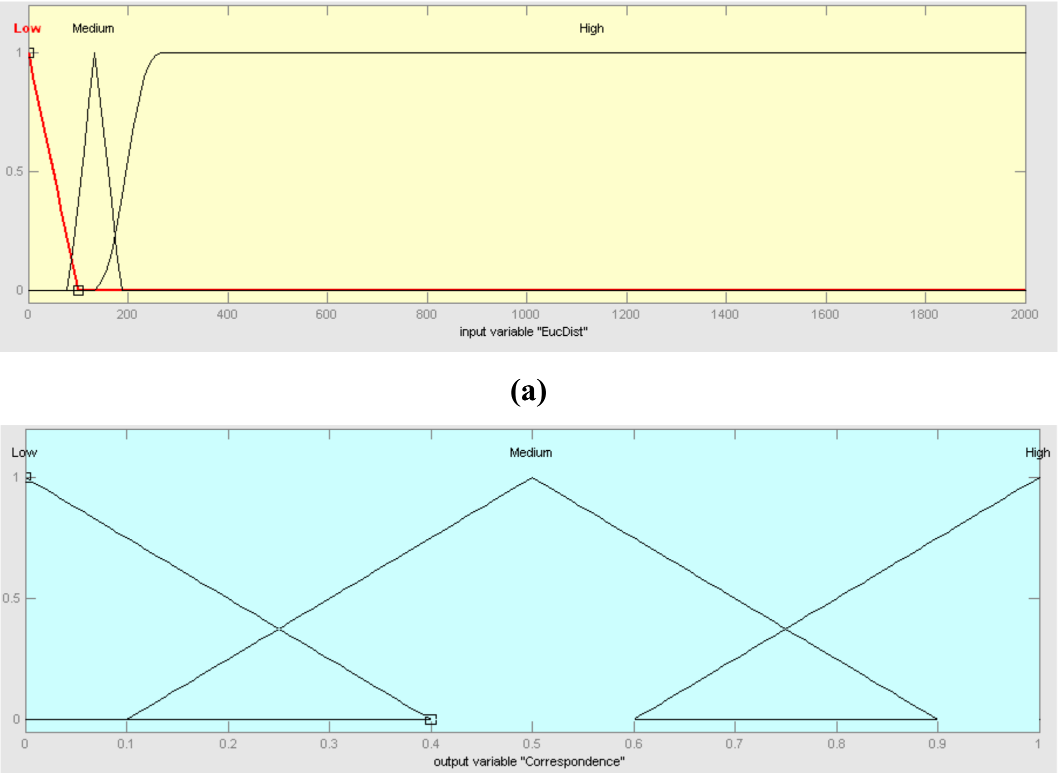
- If bounding box overlapping is high AND color similarity is high AND Euclidean distance is low, track score is high
- If bounding box overlapping is high AND color similarity is high AND Euclidean distance is medium, track score is high
- If bounding box overlapping is high AND color similarity is high AND Euclidean distance is high, track score is medium
- If bounding box overlapping is high AND color similarity is medium AND Euclidean distance is low, track score is high
- If bounding box overlapping is high AND color similarity is medium AND Euclidean distance is medium, track score is medium
- If bounding box overlapping is high AND color similarity is medium AND Euclidean distance is high, track score is medium
- If bounding box overlapping is high AND color similarity is low AND Euclidean distance is low, track score is high
- If bounding box overlapping is high AND color similarity is low AND Euclidean distance is medium, track score is medium
- If bounding box overlapping is high AND color similarity is low AND Euclidean distance is high, track score is low

- If bounding box overlapping is medium AND color similarity is high AND Euclidean distance is low, track score is high
- If bounding box overlapping is medium AND color similarity is high AND Euclidean distance is medium, track score is medium
- If bounding box overlapping is medium AND color similarity is high AND Euclidean distance is high, track score is medium
- If bounding box overlapping is medium AND color similarity is medium AND Euclidean distance is low, track score is medium
- If bounding box overlapping is medium AND color similarity is medium AND Euclidean distance is medium, track score is medium
- If bounding box overlapping is medium AND color similarity is medium AND Euclidean distance is high, track score is medium
- If bounding box overlapping is medium AND color similarity is low AND Euclidean distance is low, track score is low
- If bounding box overlapping is medium AND color similarity is low AND Euclidean distance is medium, track score is medium
- If bounding box overlapping is medium AND color similarity is low AND Euclidean distance is high, track score is low
- If bounding box overlapping is low AND color similarity is high AND Euclidean distance is low, track score is medium
- If bounding box overlapping is low AND color similarity is high AND Euclidean distance is medium, track score is medium
- If bounding box overlapping is low AND color similarity is high AND Euclidean distance is high, track score is low
- If bounding box overlapping is low AND color similarity is medium AND Euclidean distance is low, track score is medium
- If bounding box overlapping is low AND color similarity is medium AND Euclidean distance is medium, track score is low
- If bounding box overlapping is low AND color similarity is medium AND Euclidean distance is high, track score is low
- If bounding box overlapping is low AND color similarity is low AND Euclidean distance is low, track score is low
- If bounding box overlapping is low AND color similarity is low AND Euclidean distance is medium, track score is low

- If bounding box overlapping is low AND color similarity is low AND Euclidean distance is high, track score is low

These rules have been applied to form hypotheses. The output of fuzzy evaluation, track score, has been used as form, merge and prune the hypotheses. Some rules can be thought as unnecessary because of the relationship between the criteria. Bounding box overlapping is generally parallel with Euclidean distance. That is, “if bounding box overlapping is high, Euclidean distance is low” statement is generally valid. However, for unexpected cases, such as occlusion and merging, this statement cannot be valid and some rules must be inserted. For each input feature (bounding box overlapping, color similarity and Euclidean distance) and track score (correspondence), membership functions have been set as in triangular shape. Membership functions of Euclidean distance and correspondence have been shown in Figure 5.4.

Obviously, the phrase “track score is high”, does not match to a single value. For example, track score of the first case is bigger than the track score of the second case. Defuzzification is the cause of this difference. “Centroid” is selected as the defuzzification method.



(a)

(b)

Figure 5.4 Membership functions of (a) Euclidean distance and (b) Correspondence

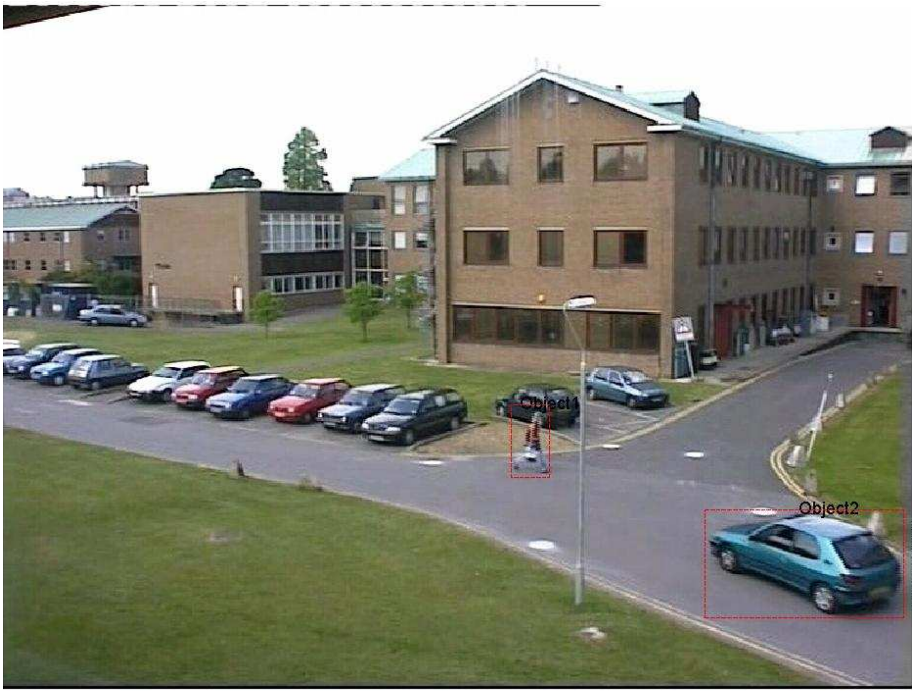
After fuzzy rules have been run, a track score in the range of [0, 1] is obtained, that is, set in the membership functions. Once the track score is obtained, some classification rules have been applied to determine the category of the measurement. That is, depending on the track score value, category of the measurement has been decided. At this point, some scenarios must be considered.

Simple tracking conditions occur when for each measurement, there is only one high track score. In other words, for this case, there is a one-to-one association for the measurements and tracks. An example of simple tracking has been shown in Figures 5.5.a and 5.5.b. If track score is high for more than one measurement for a track, it is assumed that a split is occurred. Our algorithm creates new tracks as soon as a split occurs. However, the previous track has also been updated with no measurement in Kalman filter and color histogram of the track remains same. In Figure 5.5.c, a splitting occurs due to the occlusion of the target by static objects. As can be seen from Figure 5.5.d, target has been tracked as previous object during and after splitting. Occlusion can be considered as the reverse case of splitting. In occlusion, there is only one measurement for two or more tracks. In this case, a new track has been created to track the objects during occlusion and a flag has been assigned to this track. Once the separation of the objects is detected, a match is searched between color histograms of current measurements and measurements before occlusion. Color histogram is a more reliable feature than bounding box and center point. Therefore, color histogram has been utilized to match measurements before and after occlusion. Simulation results for the occlusion case have been given in Figure 5.6. A new track, "Object3", has been created after occlusion occurs for "Object1" and "Object2" in Figure 5.5. After occlusion, by color histogram comparison, "Object1" and "Object2" have been recovered again.

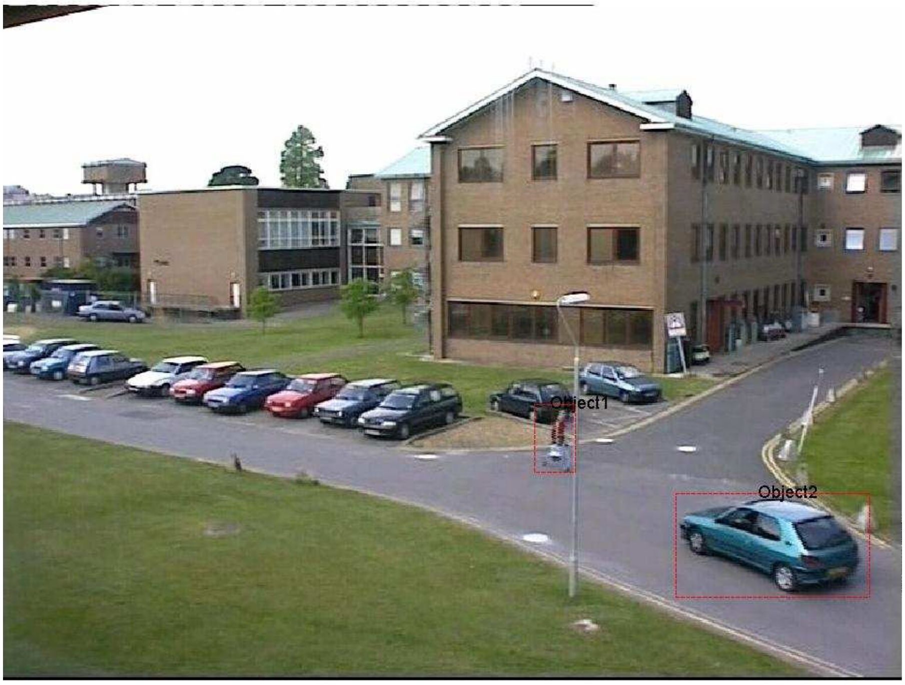
In order to estimate the next state of targets, again Kalman filter has been used. However, due to the use of bounding box overlapping criteria, our Kalman state vector for this case is different than previous cases. Area of the object is inserted as the last variable to the state vector. Hence, transition and observation matrices have also been changed in the following way:

$$X_k = \left(x_c \quad y_c \quad v_x \quad v_y \quad Area \right)$$

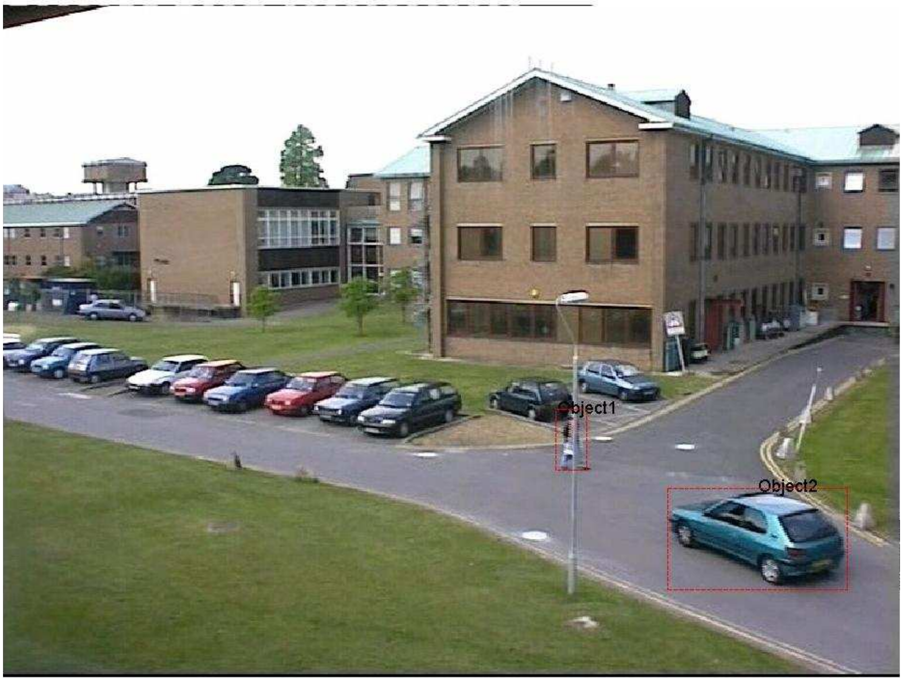
$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



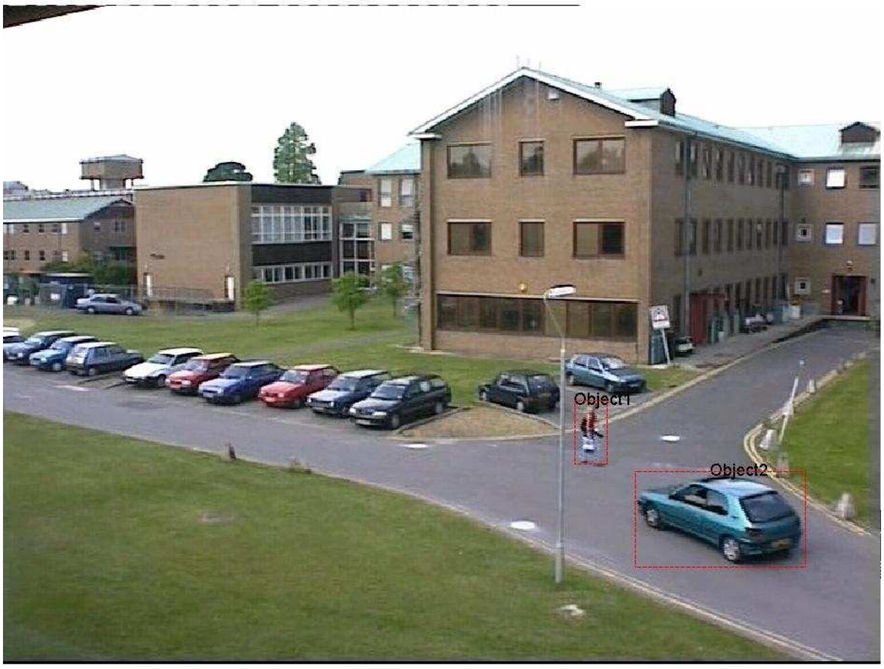
(a)



(b)

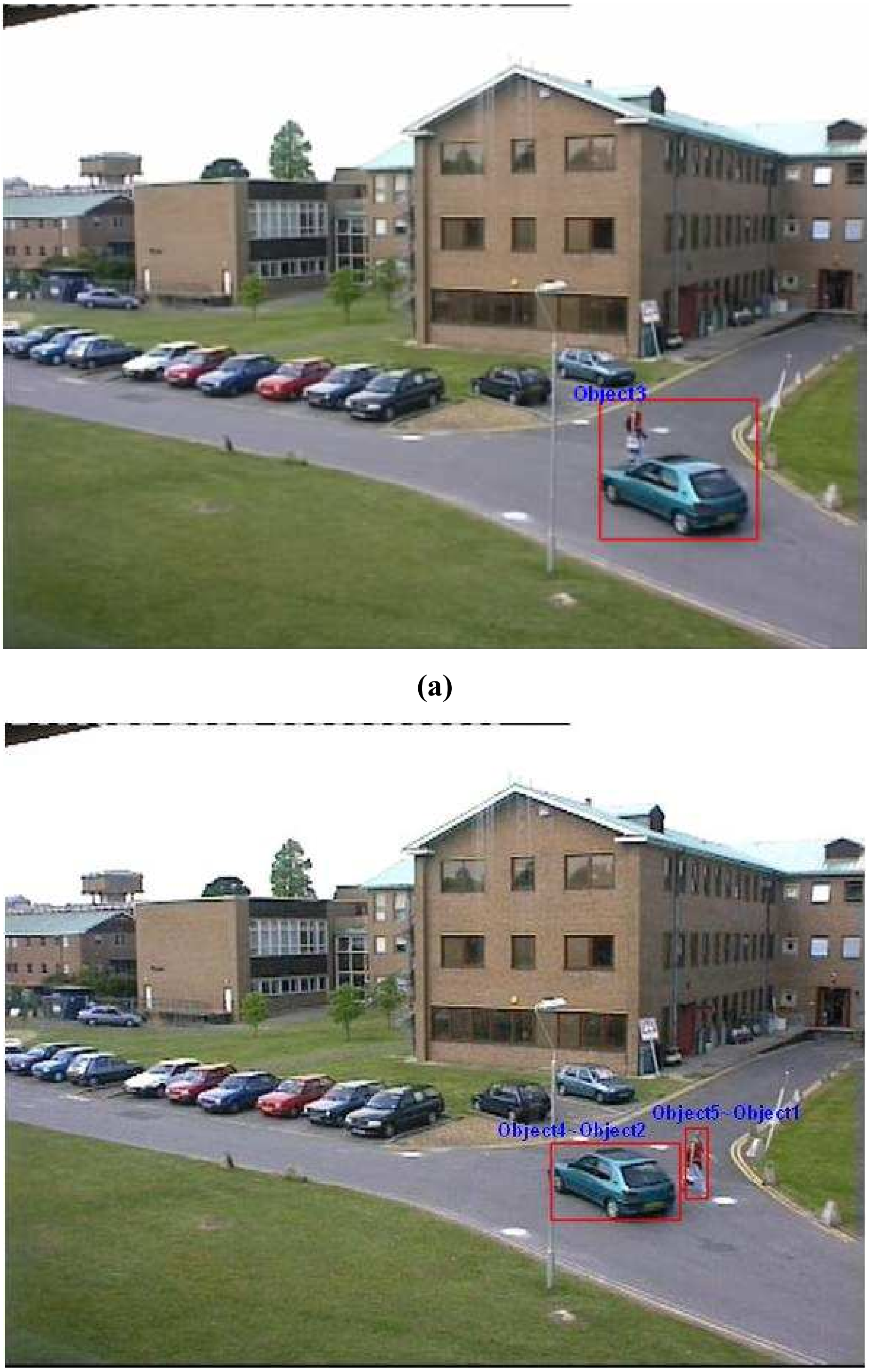


(c)



(d)

Figure 5.5 Simulation results of rule-based tracker (a, b) before splitting (c) during splitting (d) after splitting



(a)

(b)

Figure 5.6 Simulation results for occlusion case

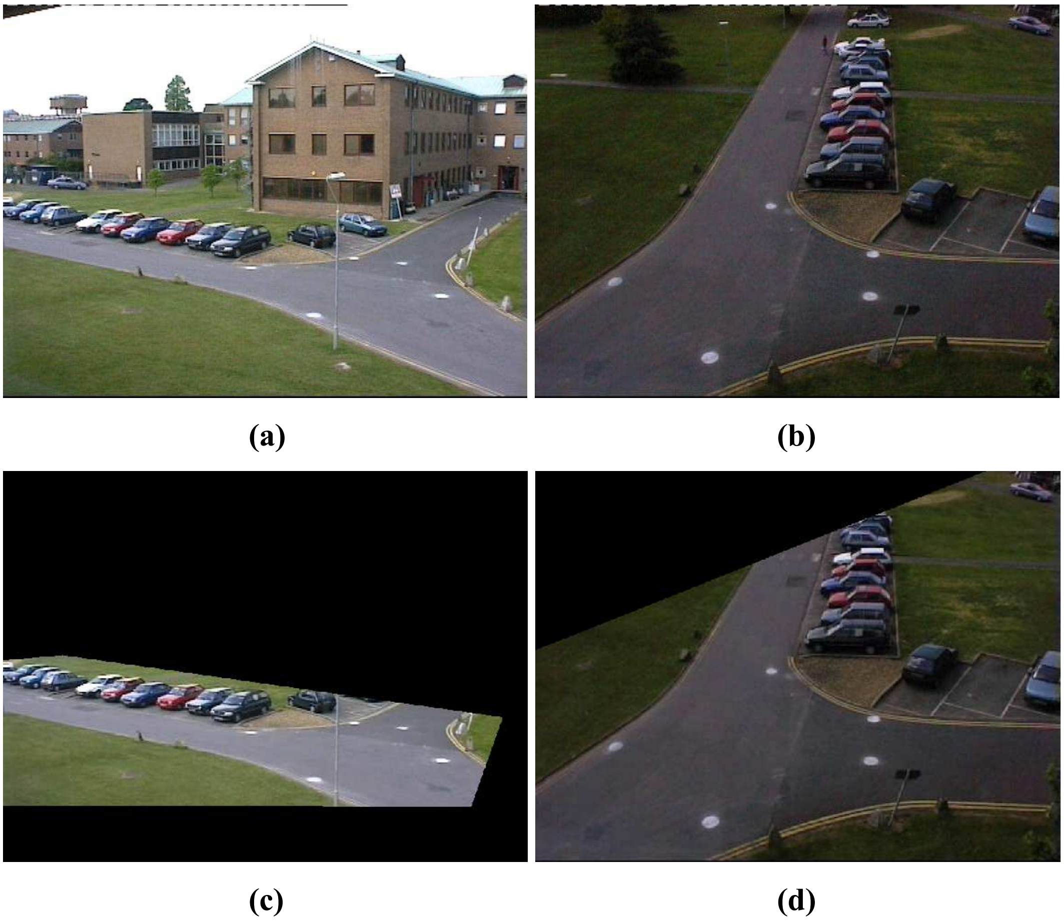
5.3.2 Rule-Based MHT Algorithm for Multi-Camera Configuration

Multiple cameras have been utilized to enhance the information about targets. In the case of occlusions and other unexpected situations, multiple camera configuration is also useful. However, the disadvantage of the use of multiple cameras is the arrangement of FOVs. For example, intersected FOVs for PETS2001 data sequence is given in Figure 5.7. As can be seen easily, any possible loss of information for camera 1 will mostly not be recovered from camera 2. For this reason, location of cameras is an important task for the multi-camera surveillance systems.

Our proposed algorithm for multiple camera configuration work in a similar way as described in section 5.3.1. The extra part of the multi-camera case is to associate the measurements to tracks that continue in the other camera. For this problem, transfer error criteria have been used. Given a set of detected moving objects in each camera view, a match between a measurement and a track is defined when the following transfer error condition is satisfied:

$$(x' - Hx)^2 + (x - H^{-1}x')^2 < \tau \qquad (5.6)$$

where x and x' are image coordinates of estimated value of track and measurement in the first and second camera views, respectively. Since homography is a point transfer approach, area can not be transferred. Hence, only position information can be updated. We have same rules as single-camera configuration for the normal conditions. Normal condition can be defined as the measurement taken from camera itself is close enough to the estimated value.

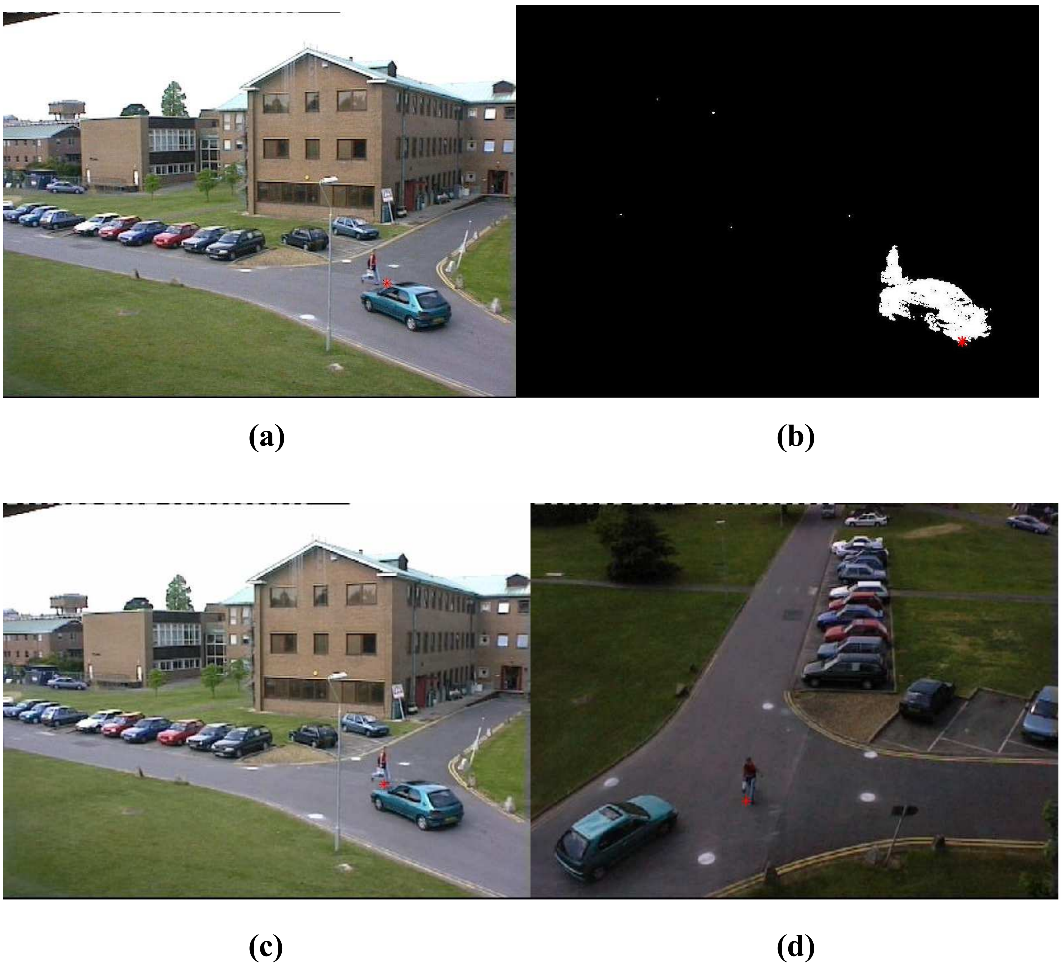


(a) (b) (c) (d)

Figure 5.7 Intersected parts of FOVs of PETS2001 dataset (a, b) Original frame of Camera1 and Camera2 (c, d) Common FOV of Camera1 and Camera2

For a camera, when a transferred measurement (other camera) gives more reliable results, updating of tracks has been done by using this transferred measurement. This situation is generally occurs in occlusions as shown in Figure 5.8. If the condition given in equation (5.7) is satisfied, track is updated with the transferred point.

$$\left| \left| x_{est} - x_{tr} \right| - \left| x_{est} - x_{meas} \right| \right| > \tau \tag{5.7}$$



(a) (b)

(c) (d)

Figure 5.8 Use of correspondence point to update the Kalman filter (a) Prediction of next state of object (b) Measurement (occlusion) (c) Transferred point from camera 2 (d) Measurement of camera 2

In Figure 5.9, it has been shown that how multiple cameras can be utilized for the occlusion case. The trajectory of the occluded track is calculated via homography and track is updated with the measurement that is obtained from the other camera.

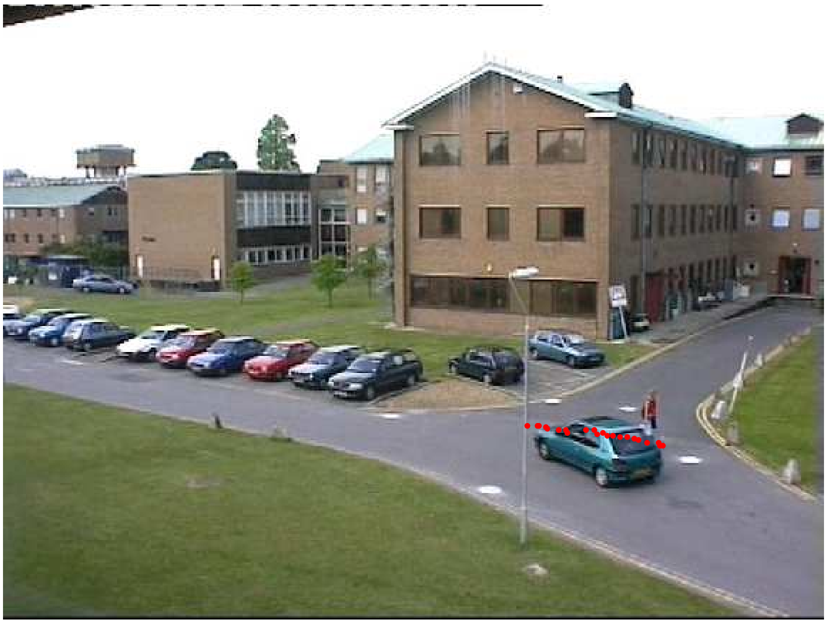


Figure 5.9 Utilization of multiple cameras

CHAPTER 6

CONCLUSION

6.1 Conclusion

In recent years, automation of the systems in both military and non-military areas has become more important. Unmanned air and land vehicles are some examples of military applications. Besides, public and personal security systems became automated as hardware and optical technologies improve.

In this thesis, important modules of automated visual surveillance system have been presented. Moving object detection has been performed utilizing four methods. Frame differencing is the simplest method in the literature and is not suitable for complex situations. Running average is one of the basic methods in moving object detection. It gives good results if parameters are well-selected for indoor environments; however, it can not adapt to sudden changes that occur in outdoor environments.

Eigenbackground subtraction has given better results than preceding methods. However, there is an updating trouble in outdoor environments for this approach. To cope with such complexities in outdoor environments, mixture of Gaussians method is used. Mixture of Gaussians has given adequate results for both indoor and outdoor environments. Therefore, mixture of Gaussians has been used as the detector while examining the trackers.

During the extraction of foreground regions, some noises and shadows have also been detected as foreground. To minimize the effects of these unwanted components, morphological and shadow removal operations has been performed.

After obtaining moving objects, tracking of these objects is the next step of this thesis. Kalman tracker, active contour based tracker and mean-shift tracker have been exploited. Kalman tracker is a point-based tracker and it has given good results when the motion of the object is linear. Active contour based tracker depends on minimization of external and internal energies. Non-flexibility of active contour based tracker is the poor part of it. That is, optimal parameters may change when environment changes. On the other hand, active contour based tracker is not dependent on detection as the Kalman tracker. This gives opportunity of skipping detection task for some frames and making the system faster.

Mean-shift tracker has given satisfactory results for trained sequences. We can say that it is robust to partial occlusions. However, it needs correct initial model for the tracked objects and this approach would fail, if the objects enter the field of view, while occluding each other. Additionally, mean-shift tracker has a high computational complexity and observed as inappropriate for multi-object tracking.

To cope with some of the difficulties, such as occlusion, multiple camera tracking is chosen as a solution. Using multiple cameras, more information can be obtained about objects. Also, information acquired from cameras must be in correspondence. In order to find the correspondence between cameras, homography matrix which is calculated by the use of direct linear transformation has been utilized.

In order to track multiple targets, there must be a special algorithm to initiate, associate and finalize the tracks of objects. Rule-based multiple hypothesis tracker has been proposed. Multiple hypothesis tracking (MHT) is the only multi-target tracker which considers these three operations. This powerful feature of MHT is combined with fuzzy logic. Fuzzy logic has been used as an evaluation tool of tracks. This novel method has been applied for multi-camera configuration.

6.2 Future Work

We have presented the main steps of typical visual surveillance systems. For the detection part, there are several works which have previously implemented successfully, such as mixture of Gaussians. The chance of making an improvement on the detection part is rare. One of the improvements which can be suggested is to extend the detection algorithms to moving camera images. However, for tracking algorithms there are more open items when compared with the detection part. Multi-target tracking is still one of the hot topics. Besides, multi-camera tracking is the best topic which can be improved even slightly.

REFERENCES

[1] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, “Pfinder: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*, vol. 22, no. 8, pp. 747–757, Aug 2000.

[3] M. Kass, A. Witkin, D. Terzopoulos. “Snakes: Active Contour Models”. *International Journal of Computer Vision,* vol. 1, no. 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. Haritaoğlu, D. Harwood and L. S. Davis. “W4: Real - Time Surveillance of People and Their Activities”, *IEEE Trans. on Patt. Anal. and Machine Intell*, vol. 22, no. 8, pp. 809–830, Aug 2000.

[6] Greg Welch and Gary Bishop, “An Introduction to the Kalman Filter”, http://www.cs.unc.edu/ [welch,gb], (18.08.2007).

[7] G. Hamarneh, A. Chodorovski, T. Gustavsson, “Active Contour Models: Application to Oral Lesion Detection in Color Images”, *IEEE Conference on Systems, Man and Cybernetics, v*ol 4, pp. 2458-2463, 2000.

[8] AVITRACK (Aircraft Surroundings, Categorised Vehicles & Individuals Tracking for apRon's Activity Model Interpretation & ChecK) Project, www.avitrack.net, (20.08.2007).

[9] PETS 2001, http://ftp.pets.rdg.ac.uk/PETS2001/, (08.10.2006).

[10] PETS 2004, http://www-prima.inrialpes.fr/PETS04/, (22.10.2006).

[11] 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.

[12] 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, pp. 1151-1163, 2002.

[13] J. Barron, D. Fleet, and S. Beauchemin, “Performance of optical flow techniques,” *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 42–77, 1994.

[14] A. Yılmaz, O. Javed, and M. Shah, “Object tracking: A survey”, *ACM Comput. Surv.”,* vol. 38, no. 4, Dec. 2006.

[15] 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.

[16] N. Oliver, B. Rosario, and A. Pentland, “A Bayesian Computer Vision System for Modeling Human Interactions”, International Conference on Vision Systems, 1999.

[17] M. Piccardi, “Background subtraction techniques: a review.” *Systems, Man and Cybernetics*, 2004 IEEE International Conference, vol 4, pp. 3099-3104, 2004.

[18] 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.

[19] I. J. Cox and S. L. Hingorani, “An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138-150, 1996.

[20] R.M. Haralick, and L.G. Shapiro, “Computer and Robot Vision, Volume I”, Addison-Wesley, pp. 28-48, 1992.

[21] S. Blackman, and R. Popoli, “*Design and Analysis of Modern Tracking Systems”*, Artech House Radar Library, Boston, 1999.

[22] Y. Jung, K. Lee, Y. Ho, “Content-Based event retrieval using semantic Scene interpretation for automated traffic surveillance”, *IEEE Trans. Intell. Transport. Syst.”,* vol. 2, pp. 151-163, 2001.

[23] 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.

[24] J. Black and T. Ellis, "Multi-camera Image Tracking", Image and Vision Computing, Elsevier, 2005.

[25] S.L. Dockstader and A.M. Tekalp, "Multiple-camera Tracking of Interacting and Occluded Human Motion", *Proceedings of IEEE*, vol. 89, no. 10, pp. 1441-1455, Oct. 2001.

[26] E. Salvador, Andrea Cavallaro and T. Ebrahimi, "Shadow Identification and Classification Using Invariant Color Model," ICASSP 2001, May 7-11, 2001.

[27] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, "Detecting of moving shadows: Algorithms and evaluation," *IEEE Trans. on Patt. Anal. and Machine Intelligence*, vol. 25, pp. 918-923, 2003.

[28] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift," *IEEE Computer Vision and Pattern Recognition*, vol 2, pp. 142-149, 2000.

[29] D.B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. on Automatic Control*, vol. 24, no. 6, pp. 843-854, Dec. 1979.

[30] R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, UK, 2003.

[31] T. Kurien, "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.

[32] B.B. Örten, Moving Object Identification and Event Recognition in Video Surveillance Systems, M.Sc. Thesis, Middle East Technical University, Turkey, July 2005.

[33] 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.