MOVING OBJECT IDENTIFICATION AND EVENT RECOGNITION IN VIDEO SURVEILLANCE SYSTEMS

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

BY

BURKAY BİRANT ÖRTEN

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

JULY 2005

Approval of the Graduate School of Natural and Applied Sciences

_____

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Prof. Dr. İsmet Erkmen
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. A. Aydın Alatan
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Tolga Çiloğlu	(METU,EE)	_____
Assoc. Prof. Dr. A. Aydın Alatan	(METU,EE)	_____
Assoc. Prof. Dr. Gözde Bozdağı Akar	(METU,EE)	_____
Dr. Çağatay Candan	(METU,EE)	_____
Prof. Dr. Adnan Yazıcı	(METU,CENG)	_____

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.

Burkay Birant Örten

ABSTRACT

MOVING OBJECT IDENTIFICATION AND
EVENT RECOGNITION IN
VIDEO SURVEILLANCE SYSTEMS

Örten, Burkay Birant
MSc., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. A. Aydın Alatan

August 2005, 73 Pages

This thesis is devoted to the problems of defining and developing the basic building blocks of an automated surveillance system. As its initial step, a background-modeling algorithm is described for segmenting moving objects from the background, which is capable of adapting to dynamic scene conditions, as well as determining shadows of the moving objects. After obtaining binary silhouettes for targets, object association between consecutive frames is achieved by a hypothesis-based tracking method. Both of these tasks provide basic information for higher-level processing, such as activity analysis and object identification. In order to recognize the nature of an event occurring in a scene, hidden Markov models (HMM) are utilized. For this aim, object trajectories, which are obtained through a successful track, are written as a sequence of flow vectors that capture the details of instantaneous velocity and location information. HMMs are trained with sequences obtained from usual motion patterns and abnormality is detected by measuring the distance to these models. Finally, MPEG-7 visual descriptors are utilized in a regional manner for object identification. Color structure and homogeneous texture parameters of the independently

moving objects are extracted and classifiers, such as Support Vector Machine (SVM) and Bayesian plug-in (Mahalanobis distance), are utilized to test the performance of the proposed person identification mechanism. The simulation results with all the above building blocks give promising results, indicating the possibility of constructing a fully automated surveillance system for the future.

Keywords: Moving Object Detection, Object Tracking, Event Recognition, Hidden Markov Models, Object Identification.

ÖZ

GÜVENLİK AMAÇLI VİDEO SİSTEMLERİNDE
HAREKETLİ NESNELERİN TANINMASI VE
OLAY ANALİZİ

Örten, Burkay Birant
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. A. Aydın Alatan

Ağustos 2005, 73 Sayfa

Bu tez, otomatize bir görsel güvenlik sistemi için gerekli olan temel parçaların tanımlanması ve geliştirilmesi üzerine bir çalışmadır. Bu tür bir sistemde en temel parçalardan birisi hareketli nesnelerin bulunmasıdır. Bu amaçla, değişken ortam şartlarını öğrenme kapasitesine sahip, ayrıca gölgeleri de nesne maskelerinden ayırabilen bir arkaplan modelleme yöntemi tanımlanmaktadır. Nesne maskeleri elde edildikten sonra birbirini takip eden kareler arasında nesneleri eşleyebilmek için hipotez tabanlı bir takip algoritması önerilmektedir. Bu iki parça, daha üst seviye işlemler için gerekli olan temel bilgileri sağlamaktadır. Sahnedeki hareketlerin anlamlandırılması ve türlerinin belirlenmesi için saklı Markov modelleri kullanılmaktadır. Nesne eşlemesi sonucunda elde edilen rota bilgileri, hareketin hız ve pozisyon bilgisini içeren akış vektörleri kullanılarak bir seri halinde yazılmakta ve bu vektörler K-ortalama yöntemi kullanılarak gruplanmaktadır. Sahnedeki genel olağan hareket rotalarına ait seriler saklı Markov modellerinin eğitilmesinde kullanılmakta, herhangi bir rotanın bu modellere olan uzaklığı hesaplanarak da hareketin doğası belirlenmektedir. Son olarak, MPEG-7 görsel tanımlayıcıları nesne tanımlama konusunda bölgesel olarak kullanılmaktadır. Hareketli nesne bölütlemesi sonucunda

elde edilen bölgelerden renk yapısı ve homojen doku parametreleri çıkarılmaktadır. Destek vektör makinesi ve Mahalanobis uzaklığı kullanılarak yapılan testler sonucunda önerilen tanıma sisteminin performansı gösterilmektedir. Yukarıda tanımlanan sistem ile yapılan simülasyonlar, otomatik bir video gözlem sistemi oluşturulması açısından olumlu sonuçlar vermektedir.

Anahtar Kelimeler: Hareketli Nesnelerin Bulunması, Nesne Takibi, Olay Analizi, Saklı Markov Modelleri, Nesne Tanıma.

ACKNOWLEDGEMENTS

I would like to express my deep appreciation to my supervisor Assoc. Prof. Dr. Aydın Alatan for his guidance, positive suggestions and for his efforts on providing a stimulating research environment.

I feel indebted to my friend and colleague Engin Tola for his invaluable ideas and support. Without him, preparation of this thesis would be a lot more difficult.

I am grateful to Assoc. Prof. Dr. Tolga Çiloğlu for valuable information he has provided with me and also grateful to Yağız Yaşaroğlu for smart suggestions and discussions.

Finally, I would like to thank my family for their emotional support and belief in me. Especially, very special thanks to my grandmother, who has put us in the center of her life for the past 6 years.

TABLE OF CONTENTS

PLAGIARISM III

ABSTRACT IV

ÖZ VI

ACKNOWLEDGEMENTS VIII

TABLE OF CONTENTS IX

CHAPTERS

1 INTRODUCTION 1

1.1 Scope of the Thesis 2

1.2 Outline of the Thesis 2

2 STATE-OF-THE-ART IN VIDEO SURVEILLANCE SYSYTEMS 4

2.1 Moving Object Detection 4

2.2 Object Tracking 5

2.3 Event Recognition 6

2.4 Person Identification 8

3 MOVING OBJECT DETECTION 9

3.1 Comparison of Moving Object Segmentation Methods 10

3.1.1 Frame Differencing 10

3.1.2 Moving Average Filtering 11

3.1.3 Eigenbackground Subtraction 12

3.1.4 Hierarchical Parzen Window Based Moving Object Detection 13

3.1.5 Simulation Results for Moving Object Detection 16

3.2 Noise Removal 19

3.2.1 Morphological operators for noise removal 19

3.2.1.1 Erosion 20

3.2.1.2 Dilation 20

3.2.2 Connected Component Labeling (CCL) and Area Filter 21
3.3 Shadow Removal 23
4 OBJECT TRACKING 25
4.1 Matching Criterion 25
4.1.1 Match Matrix 27
4.2 Track Hypotheses 28
4.3 Foreground Object Color Modeling 31
5 EVENT RECOGNITION 33
5.1 Hidden Markov Models 34
5.1.1 Discrete Markov Processes 34
5.1.2 Extension to HMMs 36
5.2 Basic Problems of HMMs 37
5.2.1 Problem 1 – Evaluation 37
5.2.2 Problem 2 – Decoding 38
5.2.3 Problem 3 – Learning 39
5.3 Recognizing Events by Using HMM 41
5.3.1 Selection of the Number of Models 44
5.3.2 Training HMMs with Multiple Sequences 46
5.4 Simulation Results 48
6 OBJECT IDENTIFICATION 53
6.1 Brief Review of MPEG-7 Standard 54
6.1.1 Color Structure Descriptor 55
6.1.2 Homogeneous Texture Descriptor 56
6.2 Classifiers 57
6.2.1 Support Vector Machine 58
6.2.2 Mahalanobis Distance 59
6.3 Combining Classifiers 60
6.3.1 Sum Rule 60
6.3.2 Product Rule 60
6.3.3 Max Rule 60
6.3.4 Min Rule 61
6.3.5 Geometric Mean Rule 61

6.3.6 Absolute Max and Min Rules 61
6.4 Simulation Results 62
7 CONCLUSIONS 66
7.1 Main Contributions 66
7.2 Discussions 67
7.3 Future Directions 69
REFERENCES 70

CHAPTER 1

INTRODUCTION

In recent years, with the latest technological advancements, off-the-shelf cameras became vastly available, producing a huge amount of content that can be used in various application areas. Among them, visual surveillance receives a great deal of interest nowadays. Until recently, video surveillance was mainly a concern only for military or large-scale companies. However, increasing crime rate, especially in metropolitan cities, necessitates taking better precautions in security-sensitive areas, like country borders, airports or government offices. Even individuals are seeking for personalized security systems to monitor their houses or other valuable assets.

Old-fashioned security systems were vastly relying on human labor instead of system hardware. As a result, detection and assessment of threat was limited with the concentration of the human operator. Additionally, area under surveillance may be too large to be monitored by a few 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 technological equipments' being much cheaper than the human resource.

The sole answer for this increasing demand for personal and societal security is automation. The vast amount of data acquired from video imagery should be analyzed by an intelligent and useful autonomous structure. This intelligent system should have the capacity to observe the surrounding environment and extract useful information for subsequent

reasoning, like detecting and analyzing the activity (motion), or identifying the objects entering the scene. Besides, monitoring should be done 24-hours-a-day, without any interruption. This sort of a system will achieve the surveillance task more accurately and effectively, saving a great amount of human effort.

1.1 Scope of the Thesis

This thesis deals with the problems of defining and developing the basic building blocks of an automated surveillance system. Initial problem is the detection of object motions in the scene. Background modeling algorithms, which are capable of coping with the changes in the scene (i.e., adaptable), are described for extracting isolated moving objects and a hypothesis-based algorithm is utilized for tracking the detected objects.

Higher-level tasks, such as event recognition and object identification, are also to be handled in such an automated system. Event recognition is achieved by modeling object trajectories by the help of hidden Markov models (HMM). Finally, visual feature-based object identification is also examined. The performances of the proposed system blocks are validated by the simulation results.

1.2 Outline of the Thesis

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

Moving object detection algorithms are described in Chapter 3. These algorithms include adaptive background modeling, moving object segmentation and noise cleaning methods. Additionally, removal of shadows from binary masks is also explained in this chapter.

After the detection of the moving objects, the next problem turns out to be tracking of these objects. Chapter 4 details a hypothesis-based object-tracking algorithm, which utilizes a color model for each foreground object to handle occlusions.

Chapter 5 mentions a hidden Markov model based event recognition algorithm. HMMs are briefly introduced in this chapter and utilization of object trajectories for event recognition is described together with some simulation results.

Visual features used for object identification, utilized classifiers and the expert combination schemes are presented in Chapter 6. Simulation results on different data sets are also provided.

The thesis is concluded in Chapter 7 and some future extensions are also discussed in this chapter.

CHAPTER 2

STATE-OF-THE-ART IN VIDEO SURVEILLANCE SYSYTEMS

The described framework in this study includes 4 main building blocks for an automated surveillance system, which can be listed as moving object detection, object tracking, event recognition and object identification. This chapter describes the related work and the latest studies in the literature on each of these building blocks.

2.1 Moving Object Detection

Detecting changes in image sequences of the same scene, captured at different times, is of significant interest due to a large number of applications in several disciplines. Video surveillance is among the important applications, which require reliable detection of changes in the scene.

There are several different approaches for such a detection problem. These methods can be separated into two conventional classes: *temporal differencing* and *background modeling and subtraction.* The former approach is possibly the simplest one, also capable of adapting to changes in the scene with a lower computational load. However, the detection performance of temporal differencing is usually quite poor in real-life surveillance applications. On the other hand, background modeling and subtraction approach has been used successfully in several algorithms in the literature.

Haritaoglu, *et al.* [1], model the background by representing each pixel with its maximum intensity value, minimum intensity value and intensity difference values between consecutive pixels. The limitation of such a model is its susceptibility to illumination changes.

Oliver, *et al.* [2] have proposed an eigenspace model for moving object segmentation. In this method, dimensionality of the space constructed from sample images is reduced by using Principal Component Analysis (PCA). Their claim is that, after the application of PCA, the reduced space will represent only the static parts of the scene, yielding moving objects, if an image is projected on this space. Although the method has some success in certain applications, it cannot model dynamic scenes completely. Hence, it is not very suitable especially for outdoor surveillance tasks.

Another statistical method is proposed by Wren, *et al.* [3], which models each point in a scene by using a Gaussian distribution with an estimated mean intensity value. The drawback of the model is that it can only handle unimodal distributions. Later, in a general approach, a mixture of Gaussians is also proposed, instead of a single Gaussian [4].

Elgammal, *et al.* [5] use 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.

2.2 Object Tracking

Obtaining the correct track information of the moving objects is crucial for subsequent actions, like event modeling and activity recognition. For this purpose, many different types of tracking algorithms have been proposed [3,6,9,10,11,12]. Most of these algorithms can be listed under the following 4 different groups: model-based, region-based, contour-based and feature-based algorithms [6].

Model-based algorithms track objects by the aim of fitting them into a predetermined model. The models are usually produced off-line by using a priori knowledge about the nature of the object and the scene under consideration [7]. An obvious disadvantage of this type of tracking is the need for priori information about *all* the objects that might appear in the observed environment. In addition to that, construction of such a model is not a trivial task.

Region-based approaches extract relevant object information like color or texture from regions and track these regions by utilizing such information [3][8].

Unlike other tracking methods, *contour-based* approaches rely on the contour information of the moving object instead of the whole set of pixels inside the object region. Object boundaries are extracted and updated in successive frames and a simpler representation is achieved in this way [6][9][10]. The performance of such a tracker is quite sensitive to initialization, making it difficult to adapt to an automated surveillance system.

The final class of trackers is the *feature-based* approaches. They aim to find and track relevant features of the object like perimeter, area of the object region [11] or more local features, like corners or vertices inside a given region [12]. Feature-based methods are usually not very effective for handling occlusions between objects.

2.3 Event Recognition

Event recognition is probably the ultimate purpose of a fully automated surveillance system. Even though it is quite important and useful to recognize an activity, it is not easy to define the type of motion that is interesting and meaningful within surveillance context. Hence, there are many studies addressing different types of events. Polana and Nelson [11] compute the optical flow fields between consecutive frames and sum up the vector magnitudes in object regions to obtain high dimensional feature

vectors that are used for recognition. Activities are classified by using the nearest neighbor algorithm. In another attempt to find simple motion characteristics, Fujiyoshi et al. [13] proposes a “star” skeletonization method. The objects are detected by using background subtraction, then their boundaries are extracted and a skeleton is produced. The authors claim that skeletonization provides important motion cues, such as body posture and cyclic motion of skeleton segments, which in turn are utilized in determining human activities, such as walking or running [13].

Instead of analyzing simplistic object motions, activity patterns in time might also be observed. Oliver, *et al*. [2] propose a state-based learning architecture with coupled hidden Markov models (CHMM), to model object behaviors and interactions between them. Johnson, *et al*. [14] represent object motion by using flow vectors, which involve spatial location and instantaneous velocity of the object. Afterwards, the trajectories are constructed as a sequence of flow vectors and a competitive learning network is adapted to model the probability density functions of flow vector sequences. Similarly, Rao, *et al*. [15] produce probabilistic models to describe the normal motion in the scene. The flow vectors are further quantized to obtain a prototype representation and trajectories are converted into prototype vector sequences. Thereafter, these sequences are evaluated using the probabilistic trajectory models.

Stauffer, *et al*. [16] produces a codebook of prototype representations from input representations (*x, y,* v_x*,* v_y*, size of object, binary mask*) by using on-line *Vector Quantization* (VQ). Then, a co-occurrence matrix is defined over the prototypes in the codebook and a hierarchical classifier is formed by using co-occurrence data. Lee, *et al*. [17] also work with prototype vectors and aim to classify both local and global trajectory points. They use *Support Vector Machines* for local point abnormality detection whereas global trajectories (sequences of vectors) are classified by using HMMs. As a final step, a rule-based system incorporates local and global information to decide on the abnormality of the motion pattern [17].

2.4 Person Identification

Understanding the identity of persons entering the scene is another important part of a surveillance system. Latest studies on person identification demonstrate the popularity of architectures based on biometrics (distinctive personal features). Face and gait are the main biometric features that can be observed within passive surveillance context [6]. Research on face recognition has a longer history and there are several studies on face detection, face tracking, extraction of facial features and face recognition [25][26][27].

Gait-based recognition has gained more attention in recent years. These studies can be classified into three main categories: model-based methods, statistical methods and physical feature-based methods. Model-based methods use anatomical models to analyze gait of a person. Parameters like joint trajectories or angular speeds are used to build the models [28][29]. In statistical methods, moment features of object regions are utilized for identifying individuals [30][31]. Finally, physical feature-based models make use of the geometric structural properties of human body to identify the motion pattern of an individual. Among these properties are the height, stride length and cadence [32]. A more detailed discussion on gait-based recognition studies can be found in [6].

CHAPTER 3

MOVING OBJECT DETECTION

Performance of an automated visual surveillance system considerably depends on its ability to detect moving objects in the observed environment. A subsequent action, such as tracking, analyzing the motion or identifying persons, requires an accurate extraction of the foreground objects, making moving object detection a crucial part of the system.

The problem of detecting changes in a scene can be described as follows: Images of the same scene is acquired in time by a static camera and the aim is to detect changes between consecutive frames. Pixels that have a significant difference compared to the previous ones are marked as *foreground* pixels, whereas other pixels are labeled as *background,* resulting in a *change mask*. The set of pixels in this change mask yields the segmentation of the moving objects.

In order to decide on whether some regions in a frame are foreground or not, there should be a model for the background intensities. This model should also be able to capture and store necessary background information. Any change, which is caused by a new object, should be detected by this model, whereas unstationary background regions, such as branches and leafs of a tree or a flag waving in the wind, should be identified as a part of the background.

In this thesis, several different methods are tested to decide on their performance for such a detection problem.

3.1 Comparison of Moving Object Segmentation Methods

The moving object segmentation methods, which are used in some comparative tests, can be listed as follows:

- Frame differencing
- Moving average filtering
- Eigenbackground subtraction
- Hierarchical Parzen window-based moving object detection

All of these methods have both advantages and disadvantages, which are provided below together with some brief descriptions. Additionally, simulation results are included to demonstrate the performance of each algorithm on some real-life data.

3.1.1 Frame Differencing

The simplest method for moving object detection is frame differencing. The model for the background is simply equal to the previous frame.

$$m(x,y,t) = \begin{cases} 0 \text{ if } |I(x,y,t) - I(x,y,t-1)| < th \\ 1 \text{ if } |I(x,y,t) - I(x,y,t-1)| > th \end{cases} \quad (3.1)$$

In the above formula, *I(x,y,t)* is the intensity at pixel location (x,y) at time *t*, *th* is the threshold value and *m(x,y,t)* is the change mask obtained after thresholding. Instead of using the previous frame, a single frame, which does not include any moving objects, can also be used as a reference. Although this method is quite fast and has an adaptation ability to the changes in the scene, it has a relatively low performance in dynamic scene conditions and its results are very sensitive to the threshold value, *th*. Additionally, based on a single threshold value, this method cannot cope with multi-modal distributions [18]. As an example for the intensity variation

of single background pixel in time having two “main” intensity values, a sample multi-modal distribution (histogram) can be seen in Figure 3-1.

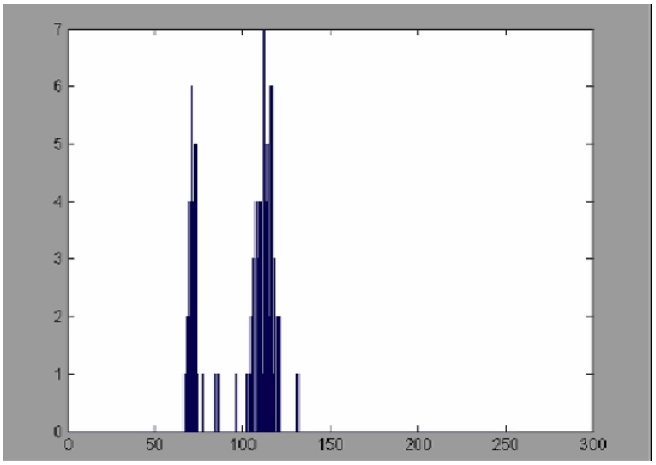


Figure 3-1. Multi-modal distribution

3.1.2 Moving Average Filtering

In this method, the reference background frame is constructed by calculating the mean value of the previous N frames. A change mask is obtained as follows:

$$m(x,y,t) = \begin{cases} 0 \text{ if } |I(x,y,t) - I_{ref}| < th \\ 1 \text{ if } |I(x,y,t) - I_{ref}| > th \end{cases} \tag{3.2}$$

where the update equation of the background model is

$$I_{ref,t} = \alpha \times I(x,y,t-1) + (1-\alpha) \times I_{ref,t-1} \tag{3.3}$$

As in the frame differencing method, mask, *m*(*x,y,t*), is obtained after thresholding by *th*. In the update equation, α is the *learning parameter*.

Moving average filtering also suffers from threshold sensitivity and cannot cope with multi-modal distributions, whereas yields a better background modeling with respect to the frame differencing.

3.1.3 Eigenbackground Subtraction

Eigenbackground subtraction [2] proposes an eigenspace model for moving object segmentation. In this method, 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 [18]:

- 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), is 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.

This method has an elegant theoretical background, if it is compared to the previous two methods. Nevertheless, it cannot model dynamic scenes as expected, even though it has some success in some restricted environments. Hence, eigenbackground subtraction is still not very suitable for outdoor surveillance tasks.

3.1.4 Hierarchical Parzen Window Based Moving Object Detection

In this section, a hierarchical Parzen window-based method [38] is proposed for modeling the background. This approach depends on nonparametrically estimating the probability of observing pixel intensity values, based on the sample intensities [5]. An estimate of the pixel intensity can be obtained by,

$$p(x) = \frac{1}{N}\sum_{k} \varphi(x - x_k) \tag{3.4}$$

where the set $\{x_1, x_2, \ldots, x_N\}$ gives the sample intensity values in the temporal history of a particular pixel in the image. The function $\varphi(.)$ in (3.4) is the window function, which is used for interpolation and usually denoted as *Parzen window* [24], giving a measure for the contribution of each sample in the estimate of *p(x)*. When the window function is chosen as a Gaussian function, (3.4) becomes:

$$p(x) = \frac{1}{N}\sum_{k}\prod_{i=1}^{3}\frac{1}{\sqrt{2\pi\sigma_i^2}} e^{\frac{-(x_i - x_{ki})^2}{2\sigma_i}} \tag{3.5}$$

The above equation can be obtained for three color channels (*R, G, B*) by using the assumption that they are all independent, where σ_i is the window function width of the i^{th} color channel window function. Considering the samples $\{x_{1i}, x_{2i}, \ldots, x_{Ni}\}$ are background scene intensities, one can decide whether a pixel will be classified as foreground or background

according to the resulting value in (3.5). If the resulting probability value is high (above a certain threshold), this indicates the new pixel value is close to the background values. Hence, it should be labeled as a background pixel. On the contrary, if the probability is low (below threshold) the pixel is decided to be part of the moving object and marked as foreground. This process yields the first stage detection of objects. However, change mask obtained as a result of this first stage calculation usually contains some noise.

In order to improve the results, a second stage should also be utilized. At this stage, by using the sample history of the neighbors of a pixel (instead of its own history values), the following probability value is calculated,

$$p_N(x) = \max_{y \in N(x)} p(x \mid B_y) \tag{3.6}$$

where $N(x)$ defines a neighborhood of the pixel x and B_y is the sample intensity values in the temporal history of y where $y \in N(x)$. Probability p_N can be defined as the *pixel displacement probability* [5] and it is the maximum probability that the observed value is the part of the background distribution of some point in the neighborhood of x. After performing a similar calculation as in (3.5) on foreground pixels (by using the history of y instead of x), which are obtained as the result of the first stage calculations, one can also find $p(x|B_y)$. After thresholding, a pixel can be decided to be a part of a neighboring pixel's background distribution. This approach reduces false alarms due to dynamic scene effects, such as tree branches or a flag waving in the wind. Another feature of the second stage is the connected component probability estimation. This process yields, whether a connected component is displaced from the background or it is an appeared object in the scene. The second stage helps reducing false alarms in a dynamic environment providing a robust model for moving object detection.

Although the above-mentioned method is effective for background modeling, it is slow due to calculations at the estimation stage. Performing both the first and the second stage calculations on the *whole* image is computationally expensive. Hence, a hierarchical version of the above system is proposed in this thesis, which includes multilevel processing to tailor the system suitable for real-time surveillance applications.

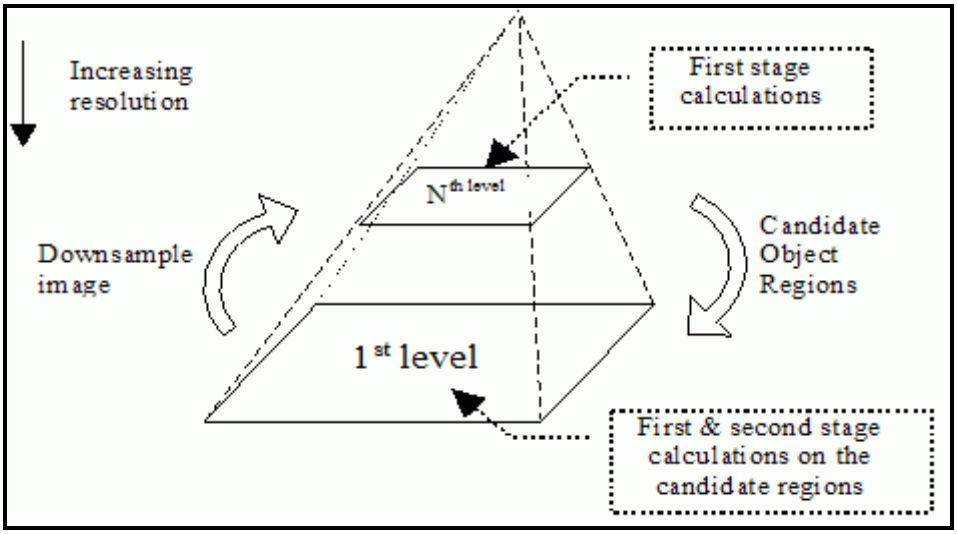


Figure 3-2. Hierarchical detection of moving objects

Figure 3-2 illustrates the hierarchical structure of the proposed system. When a frame from the sequence arrives, it is downsampled and first stage detection is performed on this *low-resolution* image. Due to the high detection performance of the nonparametric model, the object regions are captured quite accurately even in the downsampled image, providing object *bounding boxes* to the upper level. The upper level calculations are performed only on the candidate regions instead of whole image, ensuring faster detection performance. Indeed, processing the whole frame in a sequence takes approximately 5 sec. (in a Pentium IV PC with 1 GB RAM), whereas the hierarchical system makes it possible to process the same

frame around 150-200 msecs. Besides, providing a bounding box to the upper level only makes the processing faster without causing any performance degradation in the final result.

3.1.5 Simulation Results for Moving Object Detection

In this section, the simulation results for moving object detection is presented and discussed. For each video, a comparison of the following algorithm outputs is shown: frame differencing, moving average filtering, eigenbackground subtraction and hierarchical Parzen window-based moving object detection. The simulations are performed on two different sequences.

The first sequence is obtained from MPEG-7 Test Set, (CD# 30, ETRI Surveillance Video), which is in MPEG-1 format recorded at 30 fr/s with a resolution of 352x240. In Figure 3-3, a sample frame from ETRI Surveillance video is given together with the outputs of four algorithms. The results for eigenbackground and hierarchical Parzen window methods are both satisfactory, whereas moving average produces a ghost-like replica behind the object due to its use of very recent image samples to construct a reference background frame. The final result is for frame differencing, which also results with a very noisy change mask.

(a)

(b)

(c)

(d)

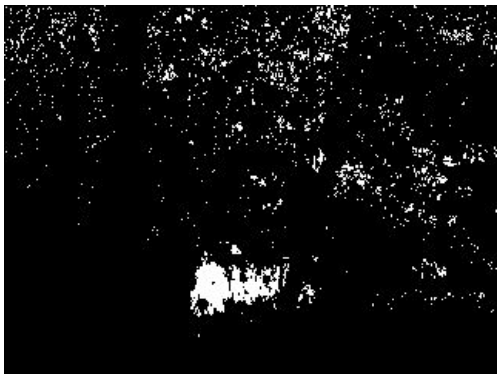
(e)

Figure 3-3. Detection results for Sequence-1
a) Original frame b) Frame differencing
c) Moving average filtering d) Eigenbackground subtraction
e) Hierarchical Parzen windowing

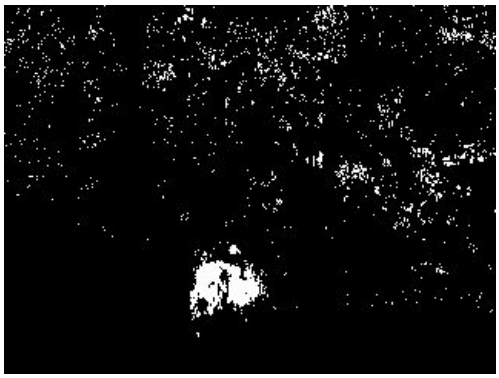
The other test sequence is in MPEG-1 format, 30 fr/s with a resolution 320x240 (it can be downloaded from http://www.cs.rutgers.edu/~elgammal). This video contains a dynamic background due to dense tree leaves and branches waving in the wind (Figure 3-4). The hierarchical Parzen windowing extracts the object silhouette quite successfully. However, moving average, eigenbackground subtraction and frame differencing approaches yield either noisy or inaccurate outputs. Obviously, noise filtering or morphological operations can also be used to improve the results of these methods at the risk of distorting object shape.

(a)

(b)



(c)



(d)

(e)

Figure 3-4. Detection results for Sequence-2
a) Original frame b) Frame differencing
c) Moving average filtering d) Eigenbackground subtraction
e) Hierarchical Parzen windowing

3.2 Noise Removal

The strategy defined in Section 3.1.4 for detecting moving objects produces quite accurate silhouettes. However, it is still highly expected to observe some noise that cannot be handled by the background model. This noise affects the outputs of many calculation stages during the processing of a frame and the overall mask becomes inaccurate due to noise. In order to get improved results, noise removal is a crucial step. For this purpose, some simple, but effective algorithms are used in the proposed system. These algorithms are:

- Morphological operators: erosion and dilation,
- Connected component labeling and area filtering.

Although connected component labeling (CCL) is a powerful tool that gives important information about the objects in the change mask, it is not only utilized primarily for noise removal. Its usage for noise removal is described briefly.

3.2.1 Morphological operators for noise removal

Morphological operators work usually on binary images by using a structuring element and a set operator (intersection, union, etc). Structuring element determines the details of the operations to be performed on the input image. Generally, the structuring element is 3×3 in size and has its origin at the center pixel. 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 (0 or 1 for binary images). For the basic morphological operators, the structuring element contains only foreground pixels (1's) and background

pixels (0's). The operators of interests in this context are *erosion* and *dilation* [19].

3.2.1.1 Erosion

As its name implies, the basic effect of erosion operator is to erode away the boundaries of the regions for the foreground pixels. A structuring element for this purpose is shown in Figure 3-5. Each foreground pixel in the input image is aligned with the center of the structuring element. 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 background value. The effect of this operation is to remove any foreground pixel that is not completely surrounded by other white pixels (Figure 3-5). As a result foreground regions shrink and holes inside a region grow.

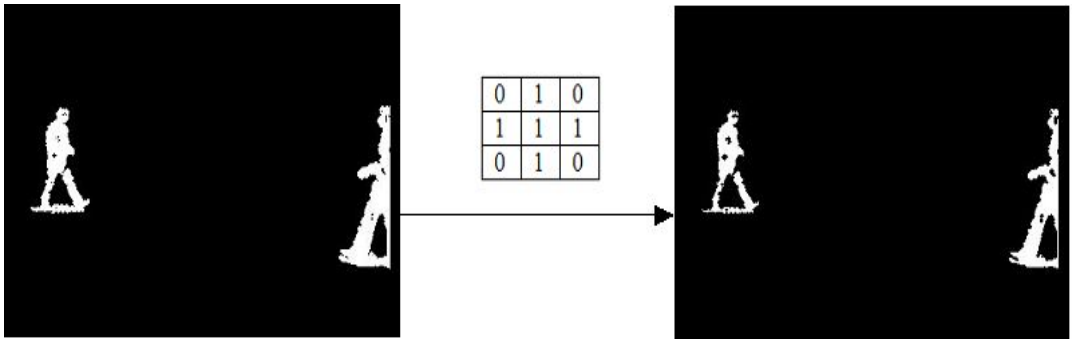


Figure 3-5. Erosion operation

3.2.1.2 Dilation

Dilation is the dual operation of erosion. A sample structuring element is shown in Figure 3-6. The structuring element works on background pixels instead of foreground pixels, with the same methodology defined in erosion operator (considering 8-connectedness). This time, foreground regions grow, while holes inside the regions shrink.

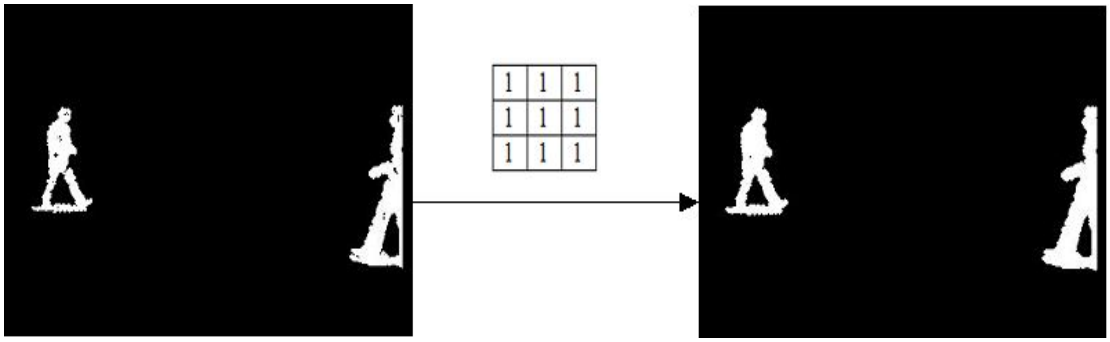


Figure 3-6. Dilation operation

By using erosion and dilation operators in turn, some of the noise (grainy noise) can be removed from the mask. Apart from the noise removal, erosion operation might disconnect the links between loosely connected regions, which are not the desired foreground objects most of the time, such as tree branches or leafs moving in the wind. When the connectedness of a region is lost and the region area is below a threshold, it is not treated as a foreground object any more. On the other hand, strongly connected regions are not affected from this operation (except from their boundaries) and a subsequent dilation operation recovers the shrinkage caused by erosion.

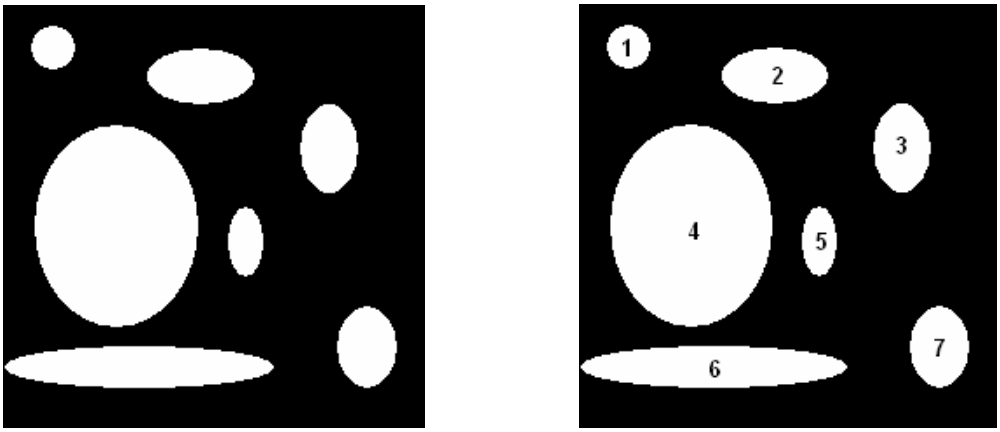
3.2.2 Connected Component Labeling (CCL) and Area Filter

Connected component labeling groups pixels in an image into components based on pixel connectivity. The algorithm adapted to the system in this thesis works as described below [19]:

1. Image is raster scanned
2. If the pixel under consideration is a foreground pixel (having value 1):
    a. If one of the pixels on the left, on the upper-left, on top or on the upper right is labeled, this label is copied as the label of the current pixel.

b. If two or more of these neighbors has a label, one of the labels is assigned to the current pixel and all of the labels are marked as equal (as being in the same group) and an equivalence table is formed.
c. If none of the neighbors has a label, current pixel is given a new label

3. All pixels on the image are scanned considering the rules defined in Step 2.
4. Classes representing the same group of pixels in the equivalence table are merged and given a single label.
5. Image is scanned once more to replace old labels with the new ones. All isolated groups of pixels are given a distinct label as a result of the algorithm (Figure 3-7).



(a) (b)

Figure 3-7. Connected component labeling on a binary image

As it is described in Section 3.2, the area of each isolated object region is obtained after CCL algorithm. Considering the average area of moving objects in the scene, a threshold value is determined. Objects having an area below this threshold are not considered as desired moving objects and they are removed from the change mask. The same threshold value is utilized after several tests conducted both in indoor and outdoor environments.

Apart from the area of a region, number of independent moving objects in the scene and the bounding boxes of these objects (width, height and center) are extracted as a result of connected component labeling, which are both very crucial in such an automated image analysis system. Indeed, tracking algorithm is based on such information.

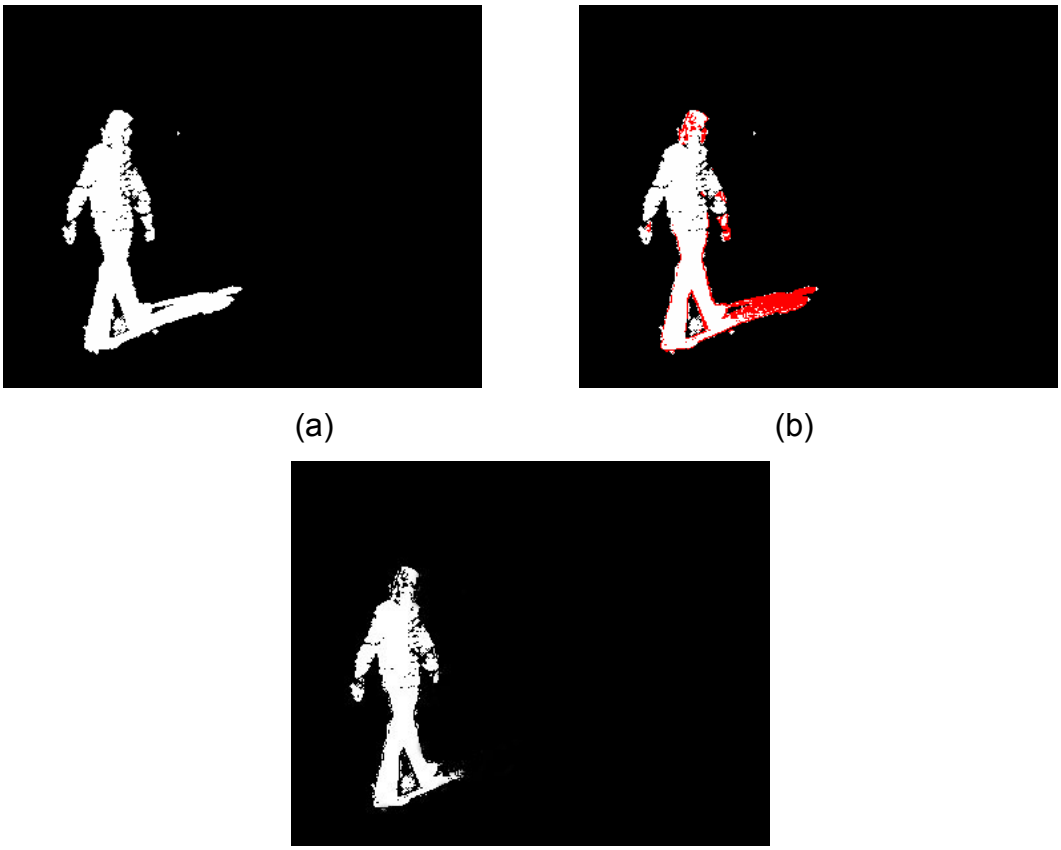
3.3 Shadow Removal

During segmentation of the objects from the background, moving cast shadows are always misclassified, as a part of the moving object. This result is expected, since the shadow causes a significant intensity change on the surface it is cast upon. However, desired segmentation of the moving objects should not contain shadows. In order to remove them, an algorithm is applied on the change mask [20]. The idea behind the algorithm is as follows: If a shadow is cast upon a surface, the intensity value decreases significantly, whereas normalized color value does not change much.

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

$$I_s(x,y) = \alpha\, I(x,y) \quad , \quad \alpha < 1 \tag{3.7}$$

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. After detection, regions of shadow are removed from change mask as shown in Figure 3-8.



(a) (b)

(c)

Figure 3-8. Shadow removal result

a) Moving object detection

b) Shadow detection

c) Mask with shadows removed

CHAPTER 4

OBJECT TRACKING

Background subtraction algorithm identifies the moving objects in the scene and separates them from the background, while producing accurate change masks. After the object segmentation is achieved, the problem of establishing a correspondence between object masks in consecutive frames should arise. Indeed, initializing a track, updating it robustly and ending the track are important problems of object mask association during visual data flow. Obtaining the correct track information is crucial for subsequent actions, such as event modeling and activity recognition.

As it was described in Chapter 2, there are several different trackers that can be utilized according to the nature of the application. In the framework defined in this thesis, the image frames of a scene are recorded by a *static* camera and the moving objects are segmented from the background before initializing a track hypothesis. Hence, after these initial steps, tracking process can be considered as a region mask association between temporally consecutive frames. Details of the tracking mechanism are described in the following sections.

4.1 Matching Criterion

Background subtraction algorithm produces accurate masks for the moving objects in the scene. Hence, after connected component labeling is applied, the bounding boxes and centroids of the moving objects can be easily obtained. In the proposed system, object region matching is achieved by

simply using box overlapping. In this approach, the bounding box of the mask of an object in the previous frame is compared to the bounding boxes of the masks in the current frame. A metric, yielding the percentage of the overlapping regions of the boxes, provides a measure for associating the masks in two consecutive frames. At this point, object displacement is assumed to be small compared to the spatial extent of the object itself. Besides, object velocity (distance between centroids of two regions) is recorded at each frame and helps to make an initial guess about the position of the object at current frame (Figure 4-1).

$O_i(t)$: Object i at time = t (current frame)

$O_j(t)$: Object j at time = t - 1 (previous frame)

$B_i(t)$: Bounding box of object i at time = t

$B_j(t-1)$: Bounding box of the object j at time = t - 1

$v_j(t-1)$: Velocity of the object j at time = t - 1

If $BoxOverlapping(B_i(t), B_j(t-1) + v_j(t-1)) > threshold$

$O_i(t) \equiv O_j(t-1)$

Else

$O_i(t) \neq O_j(t-1)$

Figure 4-1. Basic notations and matching criterion for tracking

Although small displacement assumption is valid generally, in some cases this hypothesis does not hold due to delays in preprocessing (background subtraction) stage during real-time performance. Object regions may be detected to be apart from each other so that they do not match according to simple box overlapping. Hence, object velocity information is especially useful in these situations, since it will yield an initial prediction (in the direction of previous motion) about the new position of the object.

According to the results of above defined matching criterion, a matrix is formed indicating the matches between the objects in the current frame (new objects) and that of the previous frame (old objects).

4.1.1 Match Matrix

Let *m* be the number of objects in previous frame (at time=t-1) and *n* be the number of objects in current frame (at time=t). *Match matrix*, M, is an mxn matrix denoting the matches between objects in consecutive frames, as shown in Figure 4-2. Every entry of this matrix shows whether the respective objects match according to box overlapping. A "1" value at position M_{ij} means that object *i* of the previous frame can be associated with object *j* of the current frame. Conversely, if the entry has a value of "0", there is no matching between objects *i* and *j*.

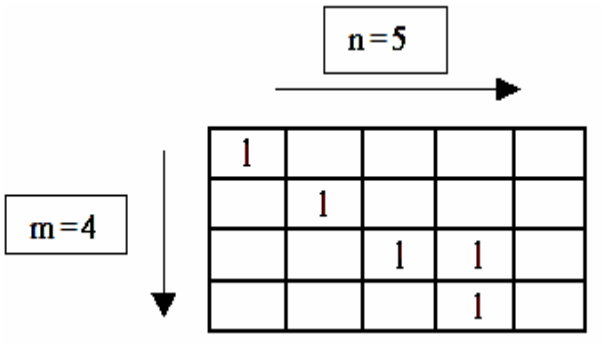


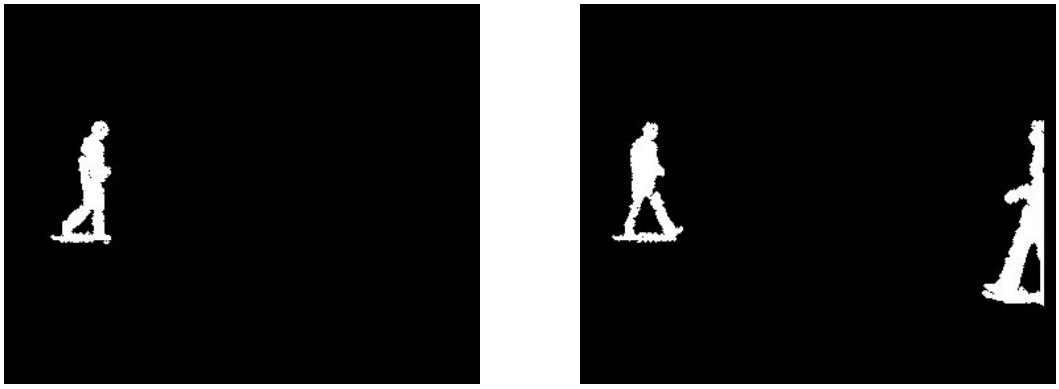
Figure 4-2. Match matrix, M

Entries having a value of "0" in matrix M are not explicitly shown in Figure 4-2. Observing the arrangement of M, one can see that more than one entry in a row or in a column might obtain a value of 1. In some cases, a row or a column may not have a single match at all. It is indeed this property of the match matrix that allows producing track hypotheses. These hypotheses are described in more detail in the next section with the help of illustrative examples.

4.2 Track Hypotheses

As pointed out in the previous section, match matrix is the starting point for the track hypothesis generation. It has primarily two sources for distinct information content: rows and columns. Rows provide information about the relation of an old object with the new objects. Likewise, columns give the relation between a new object and the old ones. There are 3 different hypotheses for both rows and columns. Hence, it will be convenient to analyze these cases denoting the rows by “R” and the columns by “C”.

Case C1. *No “1” value in a column* means a new object does not match any of the old objects known by the system (e.g. Figure 4-3). In this case a new track is initialized for the new object. Initializing a track in the described framework corresponds to recording the initial bounding box, velocity and the entrance time of the object.



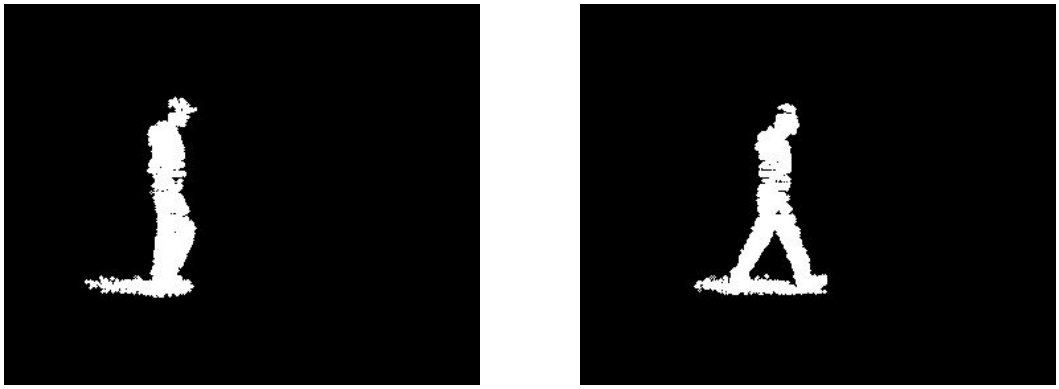
(a) (b)

Figure 4-3. A new object appears in the scene

a) Change mask at time = t-1, single object
b) Change mask at time = t, one old and one new object

Case C2. *Single “1” value in a column* stands for the situation in which a new object has only a single match. This is the desired tracking result since

an isolated moving object should have a single match between consecutive frames (e.g. Figure 4-4).

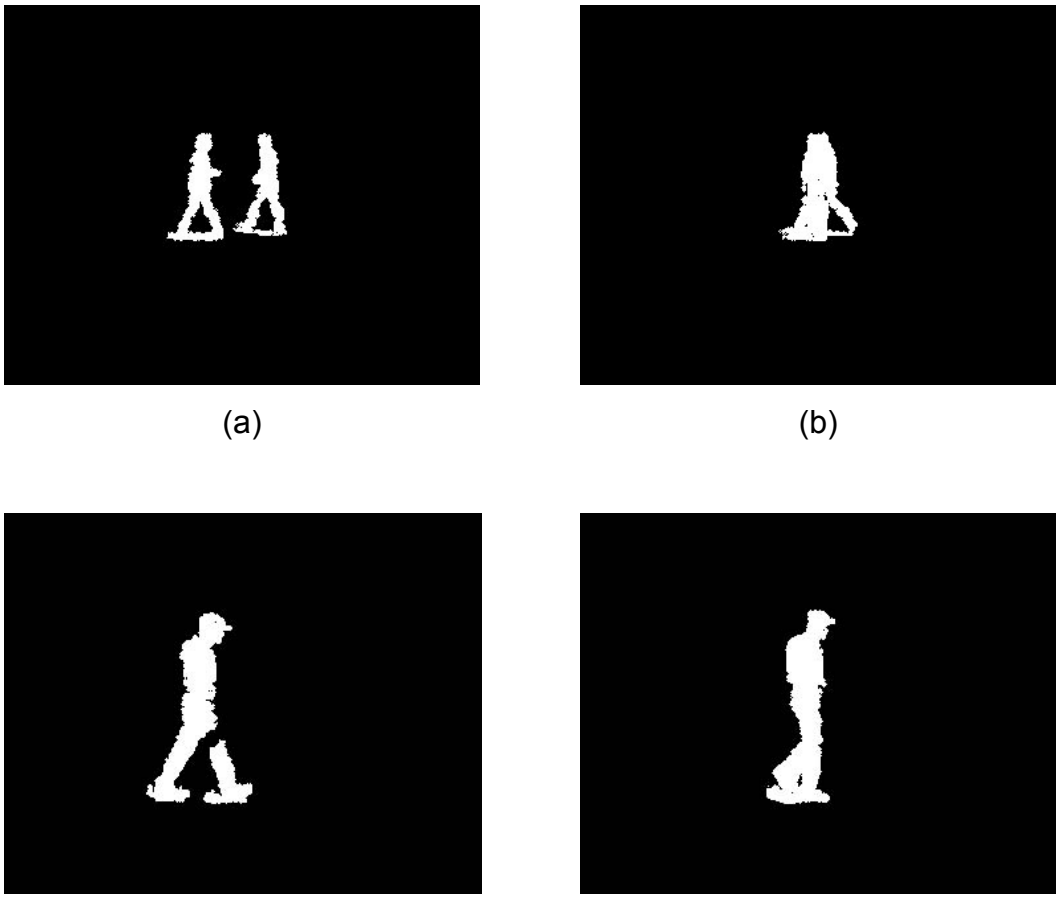


(a) (b)

Figure 4-4. New object matches a single old object

a) Old object at time = t-1
b) New object at time = t

Case C3. *More than a single "1" value in a column* means a new object matches with more than one old object. This situation can be observed, if isolated objects come together to form a group (e.g. Figure 4-5(a)(b)) and a track is initialized for this newly formed group object. During the tracking of this new group object, its trajectory data is used to update the track information of every single object in the group. Another possible reason for having several "1" values in a column is merging of the object parts, which are previously detected as isolated moving entities by background subtraction module (e.g. Figure 4-5(c)(d)). Since one of the objects is considered as a part of the other one, its track is terminated. Usually, such separated parts can be merged with the main object in a few frames. Therefore, the track history (duration for the object being tracked) is utilized to discriminate between the two different cases described above.



(a) (b)

(c) (d)

Figure 4-5. New object matches multiple old objects

a) 2 isolated objects at time = t-1
b) A single object (group object) at time = t
c) An isolated object and an object part at time = t-1
d) A single object (merged object parts) at time = t

Case R1. *No "1" value in a row* stands for the situation in which a previous object does not have a match in the current frame. This situation may occur when the moving target is temporarily blocked by another object in the scene or when the target leaves the scene. In order to account for the first case and to be able to keep track of the object, while it is out of sight, its position is estimated for a few frames by using its last available bounding box position and velocity vectors. Certainly, the resulting estimate should be

in the direction of the prior motion, since it is mostly a valid approach to assume temporal motion consistency.

Case R2. *Single "1" value in a row* means a previous object has only a single match in the current frame. This is the same situation described in *column single match* case. Tracking parameters are updated based on the information obtained from the new frame.

Case R3. *More than a single "1" value in a row* means a previous object has more than one match among the objects in the current frame. There are mainly three different reasons for this situation. The first reason is the splitting of object parts, which is exactly the opposite of the situation illustrated by Figure 4-5(c)(d). For this case, the separated part is merged with its own object and a new track is *not* initialized. The second case is the splitting of group objects that were previously merged, as described in part C3. Although it is not mentioned so far, every single object in the group has a color model, as a part of its track information, which will be described in the next section. This color model is used to identify the object leaving the group and its track is continued as described previously for an isolated object. In addition to the above-mentioned two cases, some objects enter the scene together and detected as a single target. When they are separated from each other, the track history of the group is passed to each single object and they are continued to be tracked as isolated targets.

4.3 Foreground Object Color Modeling

Change mask yields some local regions for the moving targets in the scene. The most important visual information that can be obtained from these local regions is color information. Hence, as a part of object's track information, color histogram is utilized. Color histogram can be obtained by counting the number of occurrences of a particular (*R*, *G*, *B*) value in the mask region. A

distribution is obtained for each color channel after normalizing it with the total number of pixels in the mask, as

$$P(r) = \frac{CH(r)}{Total\ number\ of\ pixels} \tag{4.1}$$

where *CH*(*r*) denotes the number of pixels having red value '*r*' and *P* is the resulting distribution. The distributions for blue and green channels can also be obtained similarly.

In order to compare the color model of two objects, a distance metric is required. For this purpose, *Kullback-Liebler divergenc*e [21] is utilized, which is mostly used to obtain the distance between any two probability distributions. If h_1 and h2 are assumed to be two probability distributions obtained from the color histograms of two distinct objects, the distance between h_1 and h_2 is given by:

$$D(h_1, h_2) = \sum h_1 \, log \frac{h_1}{h_2} \tag{4.2}$$

Since the distance metric provided above is not symmetric ($D(h_1,h_2) \neq D(h_2,h_1)$) , the following form is usually preferred instead:

$$D(h_1, h_2) = \sum h_1 \, log \frac{h_1}{h_2} + \sum h_2 \, log \frac{h_2}{h_1} \tag{4.3}$$

Appearance models are required to solve ambiguities that might arise in identifying different objects. These ambiguities might occur during occlusions or when an object leaves a group of objects. Therefore, color modeling facilitates robust tracking of each isolated object under cluttered scene conditions.

CHAPTER 5

EVENT RECOGNITION

The main purpose of an automated surveillance system is to analyze the visual changes in the observed environment, which includes detection of motion and understanding its nature. In this thesis, up to this point, the algorithms for moving object detection and tracking are discussed. They are both crucial stages in a surveillance task; however they mainly serve as a backbone for a higher-level task, such as activity analysis, which provides semantic description for the motion and interaction of objects in the scene.

As the diverse studies on event analysis point out, there is not a well-defined set of meaningful activity types that is of significant interest. Instead, they are strongly application dependent. However, detection of “abnormal” motion patterns should be the ultimate aim of every robust surveillance system. Abnormal, in this context, can be defined as an unusual event, which does not have any previous occurrences throughout the observation interval. One can say that people running around a park may look quite normal, whereas they can be marked as suspicious objects, if they do the same inside a building. Therefore, instead of labeling every motion pattern as normal or abnormal by the help of user intervention, it is a more suitable approach to observe usual activities in a scene and label the rest (which does not resemble usual behavior) as suspicious.

In the proposed framework, trajectory information is obtained after successful tracking of an object. The resulting motion patterns are used to

train a predefined number of Hidden Markov Models and subsequent event recognition is performed by using the trained HMMs.

5.1 Hidden Markov Models

Hidden Markov model (HMM) is a statistical model where the system being modeled is assumed to be a Markov process. In order to understand the idea behind HMM, it is convenient to review discrete Markov processes first.

5.1.1 Discrete Markov Processes

A Markov process is a process, which moves from state to state depending (only) on the previous n states. A collection of discrete-valued random variables $\{q_t\}$ forms an n^{th}-order Markov chain, if:

$$P(q_t = S_t \mid q_{t-1} = S_{t-1}, \ldots, q_1 = S_1) = P(q_t = S_t \mid q_{t-1} = S_{t-1}, \ldots, q_{t-n} = S_{t-n}) \quad (5.1)$$

for all $t \geq 1$ and all $q_1, q_2, \ldots, q_t$. In other words, given the previous n random variables (q_i's), the current variable (S_i = states) is conditionally independent of every variable earlier than the previous n. In the above equation, "q" is a stochastic process and "$q_t = S_t$" can be explained as event *q* being in state S_t at time *t*. The simplest Markov chain is the first order chain, where the choice of state is made purely on the basis of the previous state. Hence, expression in (5.1) simplifies into:

$$P(q_t = S_t \mid q_{t-1} = S_{t-1}, \ldots, q_1 = S_1) = P(q_t = S_t \mid q_{t-1} = S_{t-1}) \quad (5.2)$$

In a Markov chain, state transitions occur according to a set of probability values associated with the system's current state. Therefore, $\{q_t = S_i, q_{t+1}=S_j\}$ can be explained as the event of a transition from state S_i to state S_j starting at time t with a probability of:

$$a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i) \quad (5.3)$$

with a_{ij}'s having the following properties:

$$a_{ij} \geq 0, \quad \sum_{j} a_{ij} = 1 \tag{5.4}$$

Explanation of the above-mentioned ideas with a weather forecast model [22] is quite helpful. Assume, at every observation, the weather is found to be in one of the three states: *sunny*, *rainy* and *cloudy* (Figure 5-1). According to the model, weather for today can be forecasted, if one has knowledge about the weather of yesterday. Additionally, the transition from one state into another occurs with a certain probability, which is independent of time.

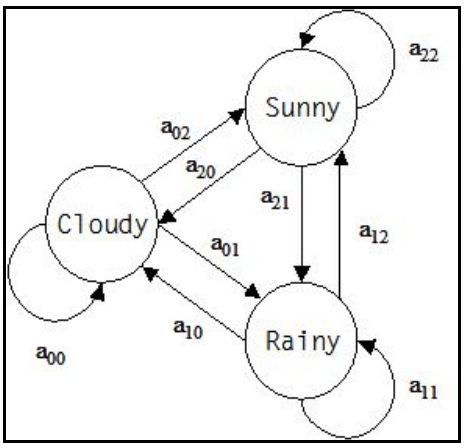


Figure 5-1. A simple Markov chain modeling weather condition

In order to characterize the state transitions, following matrix can be utilized.

$$A = a_{ij} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix}$$

Matrix *A* is called the *state transition matrix* and each row of it contains probability values for going form state *i* to state *j*. Using the information

provided by *A*, one can find the probability of having a cloudy day after 5 sunny days or obtain the expected number of consecutive rainy days. However, output of the forecast system depends on the starting point. Utilized notation for initial state probabilities are:

$$\pi_i = P(q_0 = S_i)$$

A thorough model description is achieved by providing state transition matrix, *A*, and the initial state probabilities, π_i.

5.1.2 Extension to HMMs

The idea behind hidden Markov models can be well described by building upon the example provided for discrete Markov processes. Assume that a weather model is still required but there is no available information about the previous state of it. Instead, humidity values are measured for the past few days. High humidity values strongly imply rainy weather, whereas lower values can be observed during sunny days. The observed sequence of states (humidity levels) are probabilistically related to the hidden process (weather states) and such processes can be modeled by using a hidden Markov model, where there is an underlying *hidden* Markov process changing over time, and a set of observable states which are closely related to the hidden states. In short,

Hidden states: Actual states of the system, modeled by a Markov process.
Observable states: 'Visible' states of the same process.

As it was stated previously, the state transitions are given by the probabilities, a_{ij}, in an N-by-N transition matrix, A. In addition to that, for M observation symbols v_1, v_2,…, v_m the observation probability distribution is given by matrix B, whose elements are defined as [22]:

$$b_j(k) = P(v_k \text{ at } t \mid q_t = S_j) \text{ for } 1 \le j \le N,\ 1 \le k \le M$$

When the number of states, *N*, number of observation symbols, *M*, state transition matrix, *A*, observation probability matrix, *B*, and the initial state probabilities, π, are specified, HMM is characterized completely. Following shorthand notation is used to denote the model:

$$\lambda = (A, B, \pi)$$

5.2 Basic Problems of HMMs

Once a system is described as an HMM, there are three basic problems to be solved [22]: Finding the probability of an observed sequence given an HMM (*evaluation*), finding the sequence of hidden states that most probably generated an observed sequence (*decoding*) and generating an HMM given a sequence of observations (*learning*). Although, the problems of interest in the scope of this thesis are training and evaluation, solution to all three problems are provided for completeness.

5.2.1 Problem 1 – Evaluation

Assume a sequence of observations $O = O_1O_2\ldots O_T$ and a model $\lambda = (A, B, \pi)$ is given. The problem is to compute *P(O|λ)*, the probability of sequence O given the model λ. In order to solve this problem, a method known as *forward-backward algorithm* [22] is used. A forward variable, $\alpha_t(i)$, is used throughout the calculations and it is defined as follows:

$$\alpha_t(i) = P(O_1O_2...O_t, q_t = S_i \mid \lambda)$$

The forward variable is the probability of observing the *partial* sequence $O = O_1O_2....O_t$ until time t and the system being in state S_i at that time instance. Note that probability of the overall sequence can be calculated using $\alpha_t(i)$'s. Indeed,

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{5.5}$$

Efficient calculation of the above probability value can be achieved inductively [22]:

Initialization: $\alpha_1(i) = \pi_i b_i(O_1) \quad 1 \le i \le N$

Induction: $a_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \quad \begin{cases} 1 \le j \le N \\ 1 \le t \le T-1 \end{cases}$

Termination: $P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i).$

5.2.2 Problem 2 – Decoding

In this case, the problem is to find the most probable state sequence that should have generated the output sequence, O, given the model λ. One can find the most probable sequence of hidden states by listing all possible sequences of hidden states and finding the probability of the observed sequence for each of the combinations. The most probable sequence of hidden states gives the maximum probability for the observed output. Although this approach proposes a solution, it is not viable due to its computational burden. The most popular method for solving this problem is *Viterbi Algorithm* [22], which finds the single best state sequence as a whole. A definition should be made before describing the algorithm steps.

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(q_1 q_2 \ldots q_{t-1} = i, O_1 O_2 \ldots O_t \mid \lambda)$$

$\delta_t(i)$ is the highest probability among the probabilities of all single paths, at time t, accounting for the first t observations and ending in state i. By using induction, one can arrive at the following result :

$$\delta_t(j) = \left[\max_j \delta_{t-1}(i) a_{ij} \right] b_j(O_t) \tag{5.6}$$

The main steps of the Viterbi algorithm can be listed as follows:

Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N$$
$$\psi_1(i) = 0.$$

Induction:

$$\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(O_t), \quad \begin{cases} 2 \le t \le T \\ 1 \le j \le N \end{cases}$$
$$\psi_t(j) = \arg\max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right], \quad \begin{cases} 2 \le t \le T \\ 1 \le j \le N \end{cases}$$

Termination:

$$p^* = \max_{1 \le i \le N} \left[\delta_t(i)\right]$$
$$q_t^* = \arg\max_{1 \le i \le N} \left[\delta_t(i)\right]$$

Path Backtracking: $q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1$

In the above formulation $\psi_t(i)$ is an array that holds the argument maximizing $\delta_{t+1}(i)$ for all i and t.

5.2.3 Problem 3 – Learning

The final problem to be solved is the learning problem. The aim is to detect model parameters λ = (A, B, π) for maximizing the probability of the observation sequence, O. This is by far the most challenging problem among all. Fortunately, *Baum-Welch algorithm* [22][23] proposes an iterative procedure for locally maximizing the probability. An additional parameter, $\beta_t(i)$, is to be defined before providing the details. $\beta_t(i)$ is the backward variable, which is similar to forward variable but it defines the probability of partial observation sequence from t+1 to the end provided that the system given by model λ being in state *i* at time t.

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T \mid q_t = i, \lambda)$$

The probability of being in state i is given by:

$$\gamma_t(i) = P(q_t = i \mid O, \lambda) \tag{5.7}$$

The above expression can be written as:

$$\gamma_t(i) = P(q_t = i \mid O, \lambda) = \frac{P(q_t = i, O \mid \lambda)}{P(O \mid \lambda)} = \frac{P(q_t = i, O \mid \lambda)}{\sum_{i=1}^{N} P(q_t = i, O \mid \lambda)} \tag{5.8}$$

Note that probability of system being in state i, $\gamma_t(i)$, also equals $\alpha_t(i)\beta_t(i)$, where the forward variable accounts for the observations up to time t and backward variable for the rest. Hence, (5.8) takes the following form:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{5.9}$$

As a final definition $\xi_t(i,j)$, probability of system being in state *i* at time t and in state j at time t+1, is provided below:

$$\xi_t(i,j) = p(q_t = i, q_{t+1} = j \mid O, \lambda)$$

It can also be written in a few steps in terms of the forward and backward variables. The result is obtained as:

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \tag{5.10}$$

The model parameters of λ can be calculated using the concept of counting event occurrences [22]:

$$\overline{\pi_i} = \text{expected number of times in state } i \text{ at time instant } 1 = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to } j}{\text{expected number of transitions from state } i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_i(k) = \frac{\text{expected number in state } i \text{ and observing } v_k}{\text{expected number of times in state } i} = \frac{\sum_{\substack{t=1 \\ O_t = v_k}}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$$

It is stated that [22], when calculations are initiated with a model λ = (A, B, π) and the parameters are updated according to the above defined formulas, either λ = $\bar{\lambda}$ or P(O| $\bar{\lambda}$) > P(O| λ), where $\bar{\lambda}$ denotes the updated model. In other words, re-estimating the parameters will either make no difference or make the model better. Hence, by using iterative estimation process, the most likely HMM according to the provided observation sequence can be found.

5.3 Recognizing Events by Using HMM

Once tracking of an object is successfully achieved, its trajectory information is obtained for every point it has visited. This information involves the position of the centroid and object's instantaneous velocity at each point, which are then utilized to construct a *flow vector*, *f*:

$$f = (x, y, v_x, v_y)$$

The flow vectors capture the details of object's instantaneous motion, as well as the location in the image. Assuming the object is observed for N frames, its trajectory, $T=\{(x_1,y_1,v_{x1},v_{y1}), (x_2,y_2,v_{x2},v_{y2}), \ldots, (x_N,y_N,v_{xN},v_{yN})\}$, can be written as a sequence of flow vectors: $T = f_1 f_2 \ldots f_N$. The motion sequence of an object constitutes a useful pattern in time. Therefore,

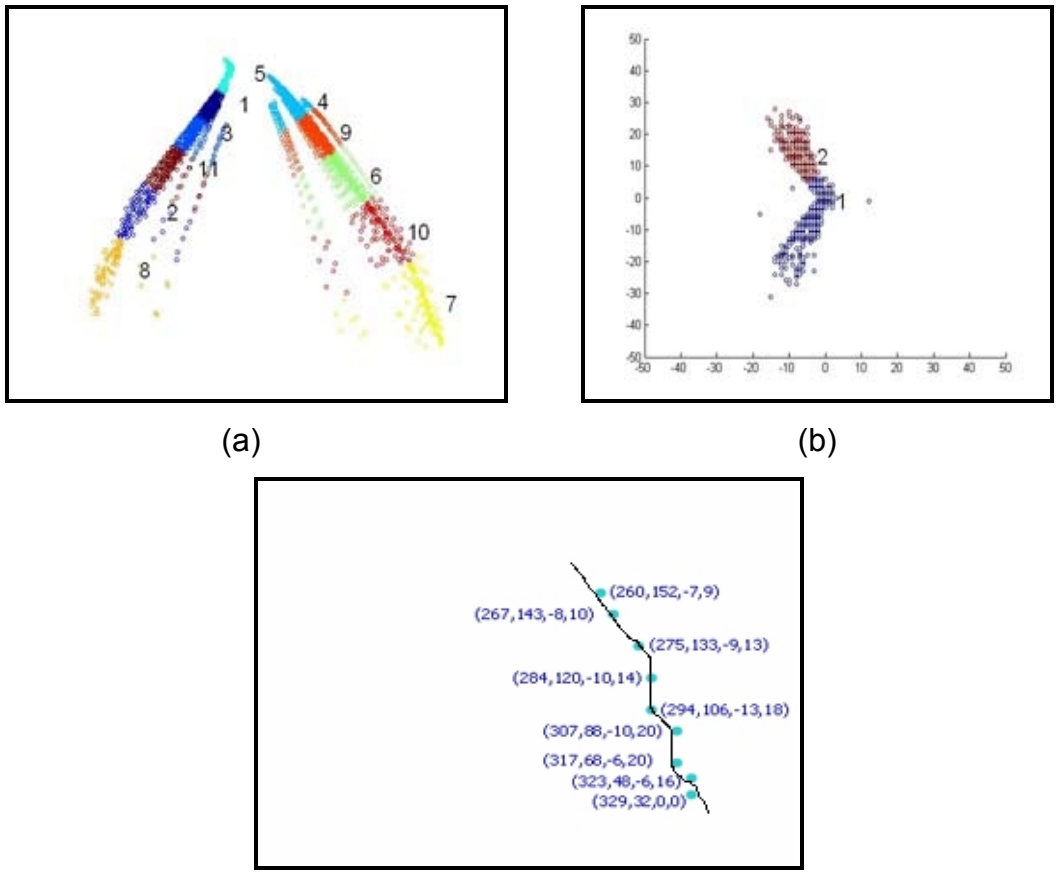
underlying event can be modeled stochastically via a hidden Markov model. The idea is to observe usual trajectories of the moving targets in the scene and train the HMMs by using *normal* trajectories. In this context, the definition of *abnormal* is any type of motion that has not any previous occurrences (i.e., not resembling normal). This method does not need any prior information about the scene and it does not require user supervision, such as defining the normal and abnormal activities for modeling the events. Hence, it is independent on the environment under observation.

In order to train hidden Markov models, a “useful sequence” is required. However, all possible combinations of velocity and location vectors constitute a large number of distinct trajectories, which cannot be modeled properly. Hence, a more compact (quantized) representation is required. For this purpose, a set of prototype vectors is produced from the flow vectors by using unsupervised clustering methods and each flow vector is represented by the index of its closest prototype.

In order to account for the contributions of position and velocity information equally, (x,y) and (v_x,v_y) are clustered separately. Assuming N clusters are needed for representing the centroids and M clusters for velocities, total number of prototypes generated is NxM. An example for the clustering of coordinates and velocities in a scene is provided in Figure 5-2. 11 clusters are used for position vectors and 2 clusters are utilized for velocities. Hence, total number of prototype vectors obtained after clustering is 22. Additionally, a sample trajectory is shown in Figure 5-2(c), which belongs to an object that has been tracked for 9 frames. Table 5-1 lists the corresponding coordinate and velocity clusters followed by this object and the prototype vector indexes.

Table 5-1. Prototype representation of an object trajectory

Coordinate clusters of each point	7,7,7,10,10,10,6,6,9
Velocity clusters of each point	2,2,2,2, 2, 2, 2,2,2
Prototype vector indexes	14,14,14,20,20,20,12,12,18



(a) (b)

(c)

Figure 5-2. Clustering of (a) coordinates and (b) velocity vectors and (c) a sample object trajectory.

The method used for clustering data is the *K-Means algorithm* [24]. K-Means algorithm tries to classify objects into K clusters such that some metric relative to the centroids of the clusters is minimized, where K is a positive integer. Several different metrics can be used for minimization such as the minimization of the sum of square distances between data and the corresponding cluster centroid, which is preferred in this study. K-Means algorithm can be summarized in a few steps:

Step 1: Begin by choosing K random points from feature space, which constitute the initial clusters

Step 2: Calculate the distance of each object to every cluster and assign the object to the closest one.

Step 3: After the assignment of all objects is complete, recalculate centroids (means) of the K clusters.

Step 4: Repeat steps 2 and 3 until convergence (according to the minimization metric described before) is achieved and the cluster centroids do not move any more.

K-Means algorithm is an elementary but effective way of unsupervised data clustering. Object centroid and velocity vectors can be successfully clustered by using this method.

Once the clustering is achieved, the motion trajectory of an object can be obtained in terms of the prototype vector indexes. In the next step, the problem of training HMMs with these sequences is solved. At this point, two problems arise: Selection of the *number of models* to be trained and training the models with *multiple observations* (previous learning algorithm is utilized for a single sequence).

5.3.1 Selection of the Number of Models

As it was stated previously, there is usually no restriction on the type of activity that can be observed in a scene. During the training stage, there might be several different trajectory sequences, which cannot be accurately modeled by a single HMM. There should be a mechanism to identify distinct motion patterns and each pattern is to be modeled separately. However, it is neither possible nor an easy task to exactly specify the number of different observable motion types. If the images are obtained from a highway (Figure 5-3(a)), it can be deduced that there are mainly two types of motion; one going from bottom to top (right lane) and the other from top to bottom (left lane). On the other hand, the scene of interest may contain a great deal of human activity (Figure 5-3(b)), which makes it even harder to define the number of usual motion patterns.

(a) (b)

Figure 5-3. Sample activity types in a (a) highway (b) campus

In this study, a novel approach is adopted for specifying the number of models in which the clues provided by centroid clustering are used. As an example, Figure 5-4(a) illustrates the trajectory clusters of objects acquired from the highway sequence. As one might easily notice, right lane traffic goes normally through the sequence 7-10-6-9-4 among the 11 clusters. Similarly, left lane has another sequence to be followed. Another observation is that, objects enter the scene at cluster 7 on the right and at cluster 4 or 5 (depending on the first detection location) on the left. In light of these observations, it can be suggested to use two HMMs for modeling the motion in this sequence, one for the trajectories starting at cluster 7 and the other for trajectories starting at 4^{th} or 5^{th} cluster (see Figure 5-4(b)). Instead of utilizing human supervision for identifying distinct motion patterns, a general and straightforward extension of this idea would be to fit a model for each centroid cluster. In other words, instead of 2 HMMs, 11 HMMs will be utilized and each model will be trained using the trajectories starting at respective cluster. This approach is based on the assumption that the motion patterns starting at the same location can be represented by the same model. Although some deviations from this assumption might be observed, it is generally valid for the usual activity types in a scene, making it a useful system part in an autonomous event detection module.

In this context, evaluation of a trajectory for abnormality detection is achieved using the entry point to the scene. One might argue that some HMMs may not be trained due to lack of trajectory data starting at that cluster. When an object enters the scene from such a cluster, it is compared against the trained models and the minimum distance is accepted as the trajectory evaluation of an object result.

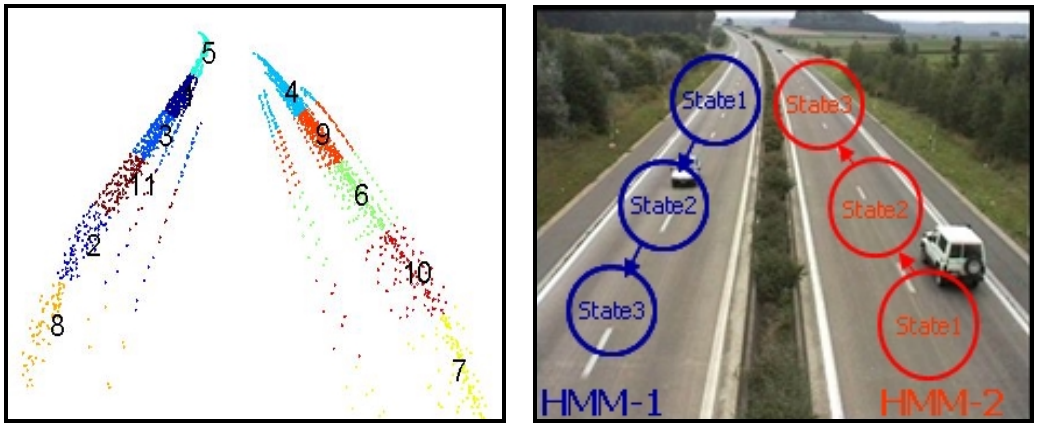


Figure 5-4. a) Trajectory clusters of the highway sequence
b) Two HMMs for modeling distinct motion patterns in the scene

As it was described above, each HMM is to be trained with all the sequences having an initial cluster belonging to the respective model. The training problem is discussed in Section 5.2.3 with a single sequence. In order to train the model with multiple sequences, some modifications should be achieved with respect to the previous algorithm and these changes are described in the next subsection.

5.3.2 Training HMMs with Multiple Sequences

In order to obtain better estimates of the parameters, multiple sequences should be used to train the model. Extension of single sequence training to multiple sequences is as follows [22]. An observation sequence was previously described as:

$$O = O_1 O_2 ... O_T$$

Multiple sequences of observations can be written as:

$$\overline{O} = \left\{ O^{(1)}, O^{(2)}, ..., O^{(K)} \right\}$$

where $O^{(i)}$ denotes the i^{th} observation sequence. Instead of maximizing the probability of observing a single sequence O (i.e., $P(O|\lambda)$), $P(\overline{O}|\lambda)$ is to be maximized. Assuming independence between observations, one can write that:

$$P(\overline{O} \mid \lambda) = \prod_{k=1}^{K} P(O^{(k)} \mid \lambda) = \prod_{k=1}^{K} P_k \qquad (5.11)$$

Parameter estimation is achieved by considering individual number of occurrences for each observation sequence. Modified formulas can be given as [22]:

$$\overline{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{i=1}^{T_k - 1} \alpha_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{i=1}^{T_k - 1} \alpha_t^k(i) \beta_t^k(i)}$$

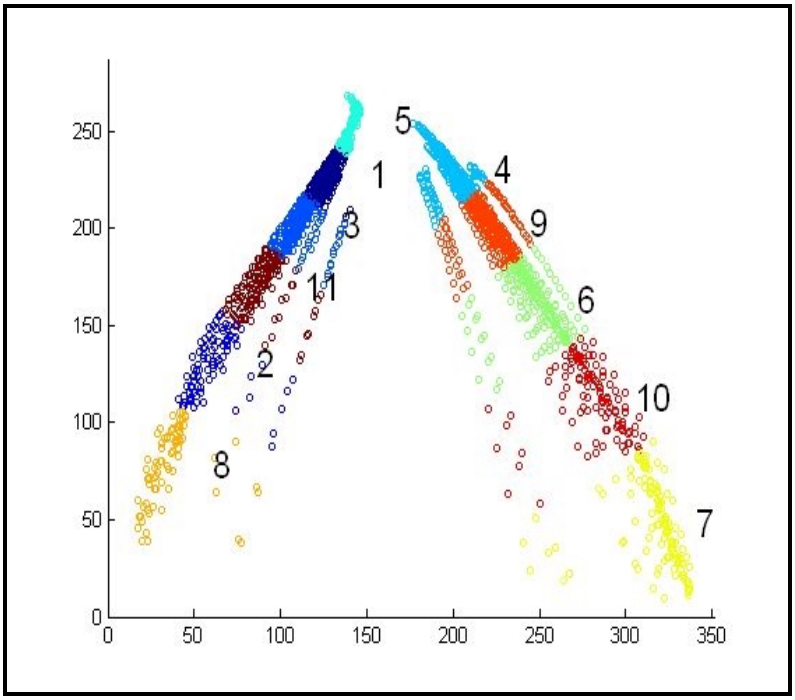
$$\overline{b}_j(\ell) = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k - 1} \alpha_t^k(i) \beta_t^k(i)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k - 1} \alpha_t^k(i) \beta_t^k(i)} \quad O_k = v_\ell$$

accounting for the scaling factor for each sequence.

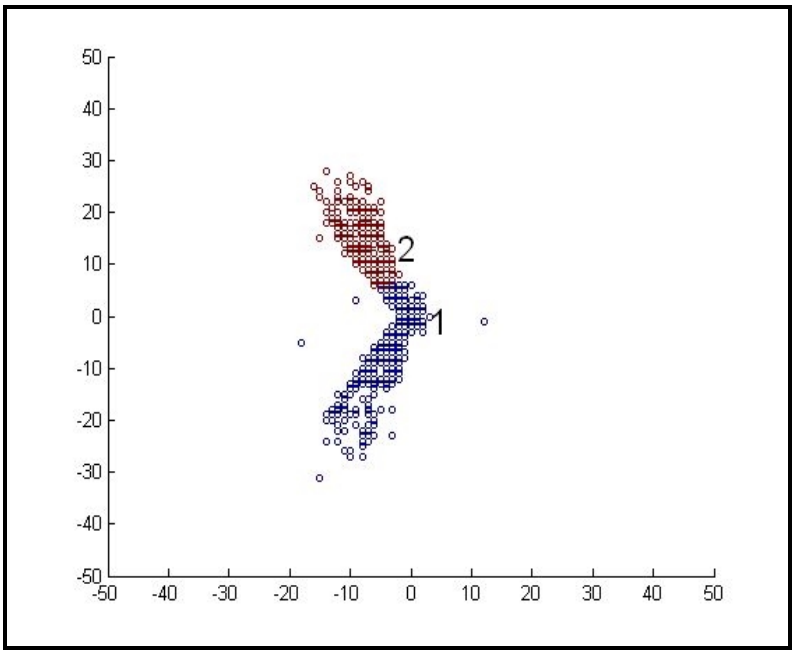
5.4 Simulation Results

In this section, the implementation details for clustering and HMMs are given together with the test results. The simulations are performed on two different sequences, as shown in Figure 5-3; *Highway* (MPEG-7 Test CD-30) and *Campus* sequences. In both cases, 3-state fully connected HMMs are used to model trajectories. State transition matrix and initial state probabilities are randomly initialized, whereas the observation matrix probabilities are assigned values according to the number of prototype vectors (i.e., according to the number of centroid and velocity clusters).

The first results are presented for the Highway sequence. The obtained trajectories are clustered according to centroid and velocity data separately, as it can be seen from Figure 5-5. 11 clusters are used for centroids and 2 clusters for the velocities, which make a total of 22 prototype vectors. Hence, observation probabilities matrix is of size 3 by 22, and each entry in a row is initialized with the value $1/22$.



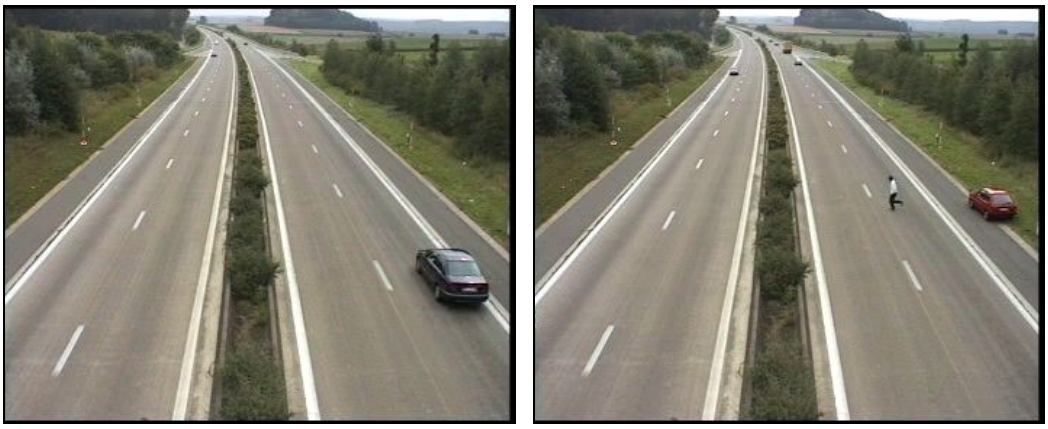
(a)



(b)

Figure 5-5. a) Centroid and b) velocity clusters for highway sequence.

As described previously, a “normal” motion follows one of the paths, given in Figure 5-5(a) with a velocity falling into one of the two clusters depicted in Figure 5-5(b). Figure 5-6(a) illustrates a typical normal trajectory that can be observed in this sequence. A car is moving from the bottom entry point towards top. On the other hand, an abnormal motion is provided in Figure 5-6(b) in which a person is trying to cross the road and arrive at the other lane. Cluster sequences followed by each object and their log likelihood (Equation 5.5) according to the 7^{th} and 6^{th} HMMs respectively are given in Table 5-2.



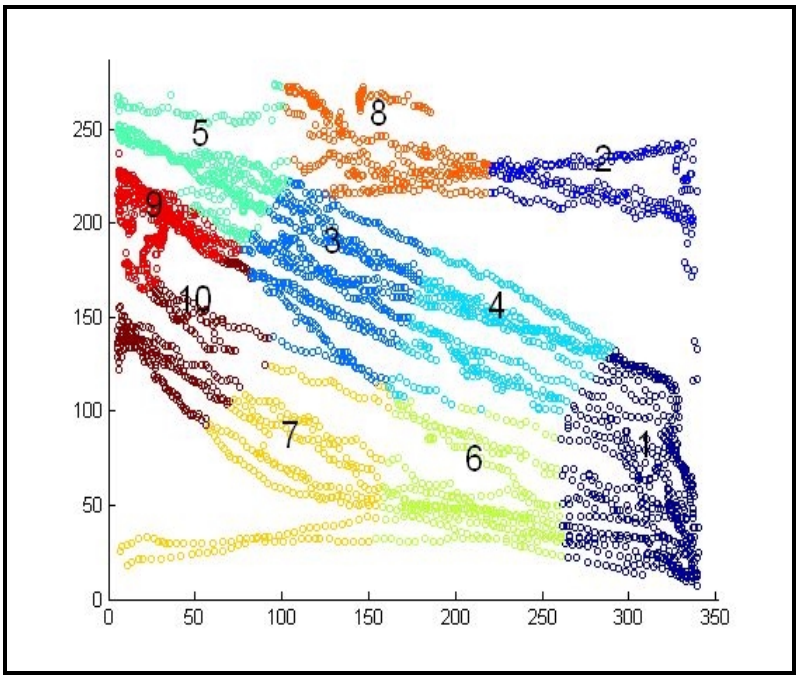
(a) (b)

Figure 5-6. a) A typical normal motion. b) A sample abnormal motion.

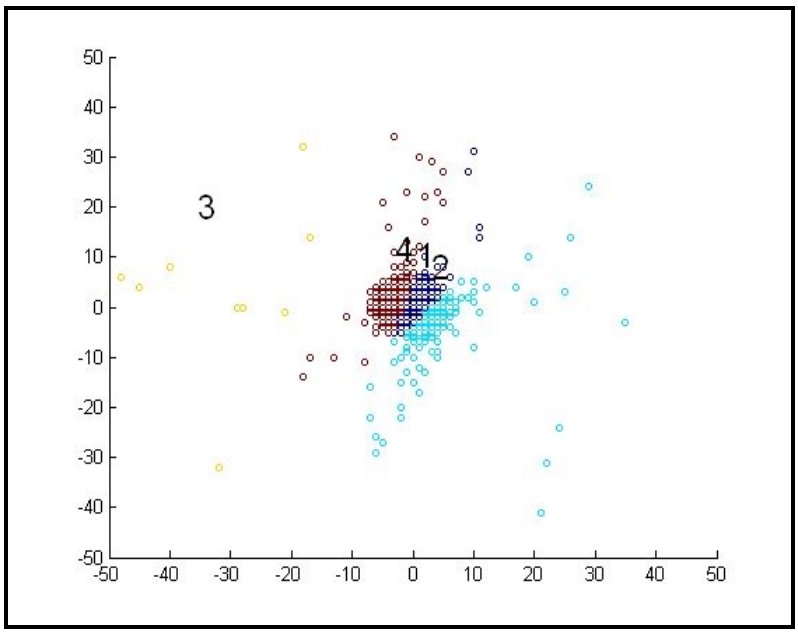
Table 5-2. A typical example from cluster sequences and log likelihood of normal and abnormal motion for highway video.

	Cluster Sequence	*Log Likelihood (Eq.5.5)*
Normal motion	7-10-6-9	12.65
Abnormal motion	6-11	119.52

More results are obtained from the Campus sequence (Figure 5-7). 10 clusters are utilized for centroids whereas velocities are divided into 4 clusters. The observation probability matrix is 3 by 40 and the initialization values are $1/40$.



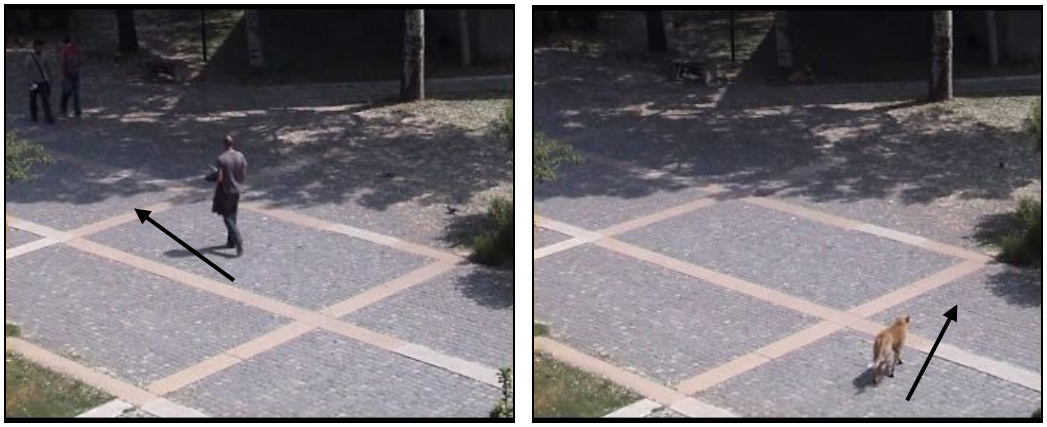
(a)



(b)

Figure 5-7. a) Centroid and b) velocity clusters for campus sequence.

Similar to the previous case, Figure 5-8(a) and 5-8(b) depicts a normal and an abnormal trajectory, respectively. Although, it is even harder to define the abnormal event in this case, one can still consider some unusual behavior. Table 5-3 lists the details of each motion.



(a) (b)

Figure 5-8. a) A typical normal motion. 7) A sample abnormal motion.

Table 5-3. A typical example from cluster sequences and log likelihood of normal and abnormal motion for campus video.

	Cluster Sequence	*Log Likelihood (Eq. 5.5)*
Normal motion	1-4-3-5	19.2
Abnormal motion	6-1	96.4

CHAPTER 6

OBJECT IDENTIFICATION

Determining some statistical information about automatically identified people, who enter into the scene, can be another important capability of an automated surveillance system. Hence, person identification should be achieved as reliable as possible. Latest studies are mainly focused on biometrics [25-32]: identification based on distinctive personal properties. As stated in Chapter 2, the main features that are of significant interest are face and gait related features. Face recognition research has reached certain maturity, whereas gait-based identification is receiving more attention, recently [28-32]. Although, biometrics is quite important for the future of automated identification systems, there are still some major problems related with it. First of all, it is not easy to represent and recognize biometric patterns. Additionally, success of biometric-based identification is excessively sensitive to measurements. In order to obtain useful information, some restrictions (like walking in a predefined style for gait recognition or looking directly towards camera for face detection) are usually required.

It was emphasized in Chapter 5 that the system proposed in this study does not put any restrictions on the motions of the targets in the scene. Hence, some other information source has to be utilized for identification purposes. The most suitable and useful information is due to the visual content of the object regions, which are obtained after segmentation from the background. For such a person recognition problem, it can be assumed

that *color* and *texture* are the most important and invariant visual features. Obviously in different applications, shape might also be used to discriminate between human, animal and vehicle classes.

In order to represent color and texture features, the well-accepted MPEG-7 descriptors are preferred [33]. Color structure [33] descriptor is used to represent color, whereas homogeneous texture [33] descriptor is utilized for representing the texture information. Before moving into the details of classification mechanism, it will be convenient to provide brief information about MPEG-7 standard and the preferred descriptors. Afterwards, some simulation results will be presented to demonstrate the performance of the proposed system.

6.1 Brief Review of MPEG-7 Standard

The latest advances in technology enable easy production and storage of multimedia data. Everyday, an increasing amount of audiovisual information is gathered from different sources such as images, audio, speech, video, and various others. When the amount of the data grows, management of this content becomes a major challenge. There has to be a way of representing the audiovisual information beyond waveforms or compression based approaches. For efficient content identification and management, International Standards Organization Moving Pictures Expert Group (ISO MPEG) established a new standard, MPEG-7, Multimedia Content Description Interface.

The same committee has also successfully developed other well-known standards, such as MPEG-1, MPEG-2 and MPEG-4. Their recent standard MPEG-7 defines a representation of multimedia information with a set of well-defined requirements [33][34]. However, MPEG-7 substantially differs from all other MPEG standards due to its target. While all the others represent the content itself, MPEG-7 represents information about the content. Therefore, it is not a coding standard, but a multimedia content description interface. Moreover, it should also be noted that MPEG-7, does

not standardize the way the information is to be extracted or consumed, whereas it standardizes which information is to be extracted and utilized. It is stated in [33] that it should be possible to create an MPEG-7 description of an analogue movie or of a picture that is printed on paper, in the same way as of digitized content.

MPEG-7 standard supports both manual and automatic annotation alternatives. Although it includes many detailed media descriptions for manual annotation, automatic annotation is strongly supported by many audiovisual low-level descriptors based on native properties of the multimedia content (i.e. color, texture, shape, melody, etc.). Obviously, a material can be described in many ways in the context of MPEG-7 [33]. A lower abstraction level would be to describe the visual content by utilizing shape, size, texture, color, movement and position information or audio content by key, mood or tempo. On the other hand, the same content can be described semantically at a higher level as follows: "*This is a scene in which a car is parked in front of a green building and a child crying out loudly.*" Moreover, MPEG-7 addresses the interoperability issues and aims at providing a standard way for multimedia content description and allows the exchange of content and its descriptions across different systems. In the next sections, some of the popular visual MPEG-7 descriptors are explained shortly.

6.1.1 Color Structure Descriptor

MPEG-7 Color Structure descriptor is used in the proposed person identification system to represent the color feature of an image. Color Structure descriptor specifies both color content (like color histogram) and the structure of this content by the help of a structure element [33]. This descriptor can distinguish between two images in which a given color is present in identical amounts, whereas the structure of the group of pixels is different (see Figure 6-1). In the person recognition system, 64-bin version of Color Structure descriptor is utilized.

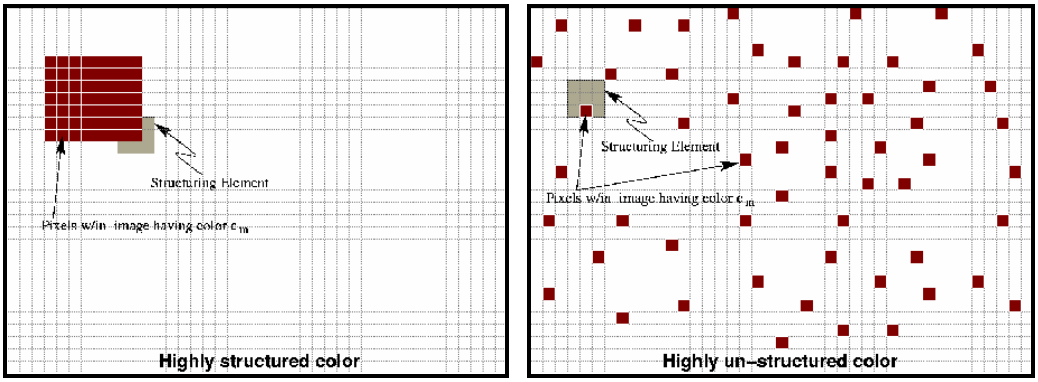


Figure 6-1. Two images with similar color histograms but different structure of color [33]

6.1.2 Homogeneous Texture Descriptor

The second fundamental feature of an image, texture, is represented by MPEG-7 Homogeneous Texture descriptor [33], characterizing the region texture by mean energy and energy deviation from a set of frequency channels (see Figure 6-2). Definition of the descriptor in MPEG-7 standard permits its use on arbitrary shaped regions, in which the background is neglected.

In Homogeneous Texture descriptor, the frequency channels are modeled by Gabor functions and the 2-D frequency plane is divided into 30 channels. In order to construct this descriptor, the mean and the standard deviation of the image in pixel domain is calculated and combined into a single feature vector with the means and energy deviations computed in each of the 30 frequency channels. As a result, a feature vector of 62 dimensions is extracted from each image [33].

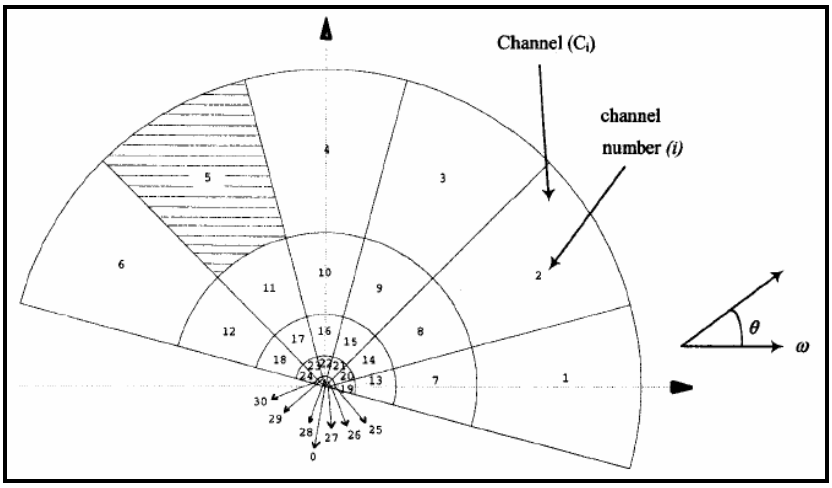


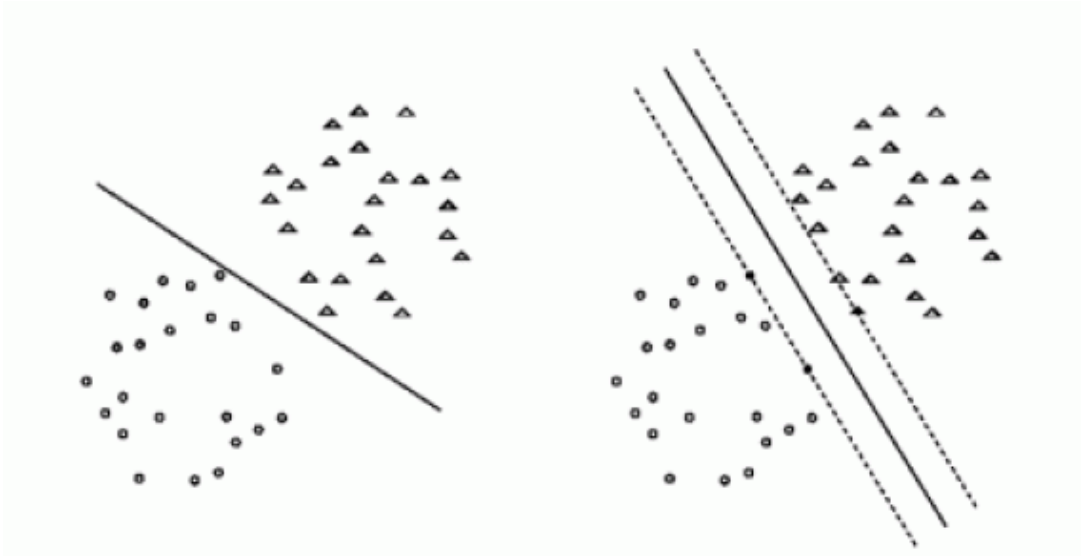
Figure 6-2. Frequency channels on 2-D image plane [33]

6.2 Classifiers

In order to be able to discriminate between people in the scene by using the selected features, a robust classifier should be used. For this purpose, a popular classifier, Support Vector Machine (SVM) [37] is selected. There can be two distinct classifiers one of which is working on color structure feature space and the other on homogeneous texture. Due to its computational burden, SVM-based classification is performed offline, while training the classifiers with a manually labeled set of object images. However, the identification of persons entering the scene has to be completed in real time, without user interaction. On the other hand, extraction of both color and texture parameters is computationally expensive. Assuming that texture provides less information than color in such low-resolution surveillance videos, the color structure feature is preferred and it is used in a Bayesian plug-in classifier, which simply utilizes the *Mahalanobis* distance between feature vectors with the assumption of Gaussian distribution for conditional densities [24]. Both of these online (Mahalanobis Distance) and offline (SVM) classifiers are described below.

6.2.1 Support Vector Machine

Support Vector Machine performs classification between two classes by finding a decision surface via certain points of the training set. This approach is different in the way that it handles the risk concept. Although other classical classifiers try to classify training sets with minimal errors, SVM can sacrifice from training set performance for being successful on yet-to-be-seen samples [37]. Briefly, one can say that SVM constructs a decision surface between samples of two classes, maximizing the margin between them (Figure 6-3) [34].



(a) (b)

Figure 6-3. Boundary obtained by an a) Ordinary classifier b) SVM.

SVM classifies test data by calculating the distance of samples from the decision surface with its sign signifying which side of the surface they reside.

In this study, in order to improve the classification results, both single and combined classifier performances are investigated. In order to combine the classifier outputs, each classifier should produce calibrated posterior probability values. In order to obtain such an output, a simple logistic link function method, proposed by Wahba [35], is utilized:

$$P(in\text{ - }class \mid x) = \frac{1}{(1 + exp(-f(x)))}$$

In this formula, $f(x)$ is the output of an SVM, which is the distance of the input vector from the decision surface. The details about the classifier combination schemes are provided in Section 6.3.

6.2.2 Mahalanobis Distance

Mahalanobis distance is based on the correlations between variables by which different patterns can be identified and analyzed. It is the normalized distance between two N-dimensional vectors, which are scaled by the *statistical variation in each component* of these vectors. Consider a cloud of points having a mean, μ, and a covariance matrix, C. Mahalanobis distance of a vector, ***x***, to this cloud of points can be defined as follows:

$$D_M(x) = \sqrt{(x - \mu)^T C^{-1} (x - \mu)}$$

It is a useful way of determining *similarity* of an unknown sample set to a known one. It differs from Euclidean distance in that it takes into account the correlations of the data set. Obviously, such an approach is equivalent to finding the minimum-error-rate classifier with the assumption that the conditional probabilities are Gaussian distributed [24].

Mahalanobis distance is utilized during real-time person identification. Color structure feature of an object is extracted from every observed frame and its statistics (mean and covariance matrix) is obtained. Afterwards, the object is compared with the stored data in the archive by utilizing Mahalanobis distance in between. If the object does not resemble any of the known classes, it is labeled as a new identity. The overall process is completed quite fast, enabling real-time operation.

6.3 Combining Classifiers

Visual features and the classifiers that work on these features are described in the previous sections. It was noted that both the color and texture based classifiers are SVM-based. In order to improve classification performance, calibrated posterior output probabilities that are obtained from each classifier are combined by using several different combination schemes [36]. After such a combination, the resultant probability value is compared against a threshold value and the final classification result is achieved. These combination methods are listed below.

6.3.1 Sum Rule

The first approach is *Sum rule*, which in the two expert case simplifies into an arithmetic average of the two probabilities.

$$P(p_k \mid x, y) = \frac{P_1(p_k \mid x) + P_2(p_k \mid y)}{2}$$

where $P_1(p_k|x)$ and $P_2(p_k|y)$ are the single expert probabilities (one working on color structure, other on homogeneous texture) of a person being p_k according to the features x and y.

6.3.2 Product Rule

Product rule is another well-known combination method, which is specified by the following formula:

$$P(p_k \mid x, y) = \frac{P_1(p_k \mid x) \times P_2(p_k \mid y)}{P_1(p_k \mid x) \times P_2(p_k \mid y) + (1 - P_1(p_k \mid x)) \times (1 - P_2(p_k \mid y))}$$

6.3.3 Max Rule

Max rule is given by a similar formula to product rule. Instead of multiplying probabilities, maximum of them is utilized.

$$P(p_k \mid x,y) = \frac{max(P_1(p_k \mid x), P_2(p_k \mid y))}{max(P_1(p_k \mid x), P_2(p_k \mid y)) + max((1 - P_1(p_k \mid x)), (1 - P_2(p_k \mid y)))}$$

6.3.4 Min Rule

The formulation is similar to the Max rule:

$$P(p_k \mid x,y) = \frac{min(P_1(p_k \mid x), P_2(p_k \mid y))}{min(P_1(p_k \mid x), P_2(p_k \mid y)) + min((1 - P_1(p_k \mid x)), (1 - P_2(p_k \mid y)))}$$

6.3.5 Geometric Mean Rule

Calculating the geometric mean of the probabilities is another approach for combining classifiers:

$$P(p_k \mid x,y) = \sqrt{P_1(p_k \mid x) \times P_2(p_k \mid y)}$$

6.3.6 Absolute Max and Min Rules

Absolute max and *absolute min* rules are used as special cases of majority vote rule. Absolute max rule picks the highest of the probabilities as the last decision, whereas absolute min rule picks the lowest probability and assigns it to the sample. Figure 6-4 illustrates the use of experts in a combination scheme.

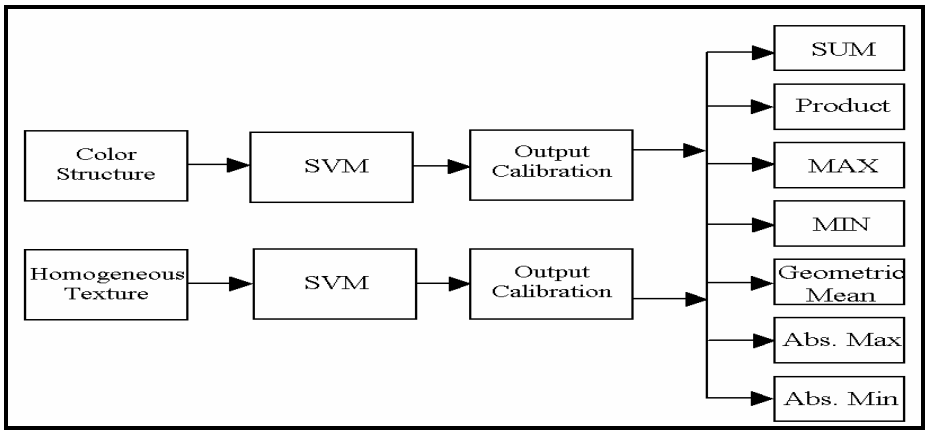


Figure 6-4. Expert combination scheme.

6.4 Simulation Results

In this section, both classifier combination simulation results, which are performed using SVM and classification performance results of Bayesian plug-in classifier, are provided. Classifier combination simulations are performed on two different sequences. The first one is from MPEG-7 Test Set, (CD# 30, ETRI Surveillance Video), in which two persons are to be identified out of seven distinct identities. Typical examples of frames, containing these distinct identities, are given in Figure 6-5. The selected objects are first divided into equal-sized train and test sets. Then from these sets each person class is trained and tested according to one-against-all scheme.

Figure 6-5. (Above) Two persons to be identified, person-1 (left) and person-2 (right).
(Below) Five Other identities.

In addition to the decisions of single classifiers based on color and texture, seven different classifier combination schemes are also evaluated. The performance results for the identities in this sequence are presented in Table 6-1.

Table 6-1. Person identification results for ETRI video

		Single Expert		Combined Experts						
		Color	**Texture**	**Sum**	**Product**	**Max**	**Min**	**Geo. Mean**	**Abs Max**	**Abs Min**
Person 1	**Accuracy**	92,94%	44,77%	67,64%	67,64%	67,64%	67,64%	62,04%	**85,64%**	52,07%
	Precision	100,00%	0,00%	95,71%	95,71%	95,71%	95,71%	97,67%	**84,85%**	0,00%
	Recall	85,28%	0,00%	34,01%	34,01%	34,01%	34,01%	21,32%	**85,28%**	0,00%
Person 2	**Accuracy**	66,15%	88,82%	90,37%	90,37%	90,37%	90,37%	85,40%	**90,06%**	64,91%
	Precision	100,00%	89,84%	100,00%	100,00%	100,00%	100,00%	100,00%	**90,00%**	100,00%
	Recall	53,02%	95,26%	86,64%	86,64%	86,64%	86,64%	79,74%	**96,98%**	51,29%

As it can be observed from Table 6-1, color and texture features of the person to be recognized affect the performance of single expert significantly. For instance, in Person-1 of Figure 6-3, color-based expert outperforms the texture-based one, since the texture of the clothes of this person has no significant difference from that of the other identities. On the other hand, for the Person-2 case, texture-based expert yields better results. Such a problem seems to be solved by combining these experts in an appropriate scheme. According to Table 6-1, it can be observed that absolute max combination yields the most stable and promising results for this case.

The second sequence is produced in METU campus, with 7 distinct persons moving in the scene (Figure 6-6). Classification strategy is similar to the previous one and 6 of them (excluding bottom-right identity) are to be identified according to one-against-all scheme. The results are listed in Table 6-2.

Figure 6-6. Persons in METU sequence.

Table 6-2. Person identification results for METU video

		Single Expert		Combined Experts						
		Color	Texture	Sum	Product	Max	Min	Geo. Mean	Abs Max	Abs Min
Person 1	Accuracy	85,71%	72,45%	**88,76%**	86,63%	86,76%	84,76%	87,76%	68,37%	89,80%
	Precision	70,83%	56,36%	**76,00%**	75,00%	74,32%	72,00%	76,19%	52,31%	81,58%
	Recall	100,00%	91,18%	**98,06%**	96,06%	97,06%	94,06%	94,12%	100,00%	91,18%
Person 2	Accuracy	98,61%	75,00%	**90.1%**	79.25%	87,50%	76.82%	84,72%	98,61%	75,00%
	Precision	100,00%	0,00%	**100,00%**	100,00%	100,00%	100,00%	100,00%	100,00%	0,00%
	Recall	94,44%	0,00%	**52.41%**	48.73%	50,00%	47.82%	38,89%	94,44%	0,00%
Person 3	Accuracy	100,00%	62,99%	**97,06%**	93,06%	95,06%	96,48%	97,64%	86,61%	76,38%
	Precision	100,00%	56,41%	**95,34%**	92,34%	94,64%	95,56%	100,00%	75,36%	100,00%
	Recall	100,00%	42,31%	**97,15%**	94,15%	96,32%	96,15%	94,23%	100,00%	42,31%
Person 4	Accuracy	98,58%	58,16%	**95,04%**	93,08%	95,04%	94,46%	95,04%	65,96%	90,78%
	Precision	97,50%	58,41%	**91,76%**	90,85%	91,76%	91,89%	94,94%	61,90%	98,51%
	Recall	100,00%	84,62%	**100,00%**	95,68%	98,68%	98.64%	96,15%	100,00%	84,62%
Person 5	Accuracy	99,09%	77,27%	**96,36%**	92,36%	93,36%	95,44%	94,55%	90,00%	86,36%
	Precision	96,77%	60,00%	**88,24%**	86,24%	87,24%	88,24%	90,00%	73,17%	100,00%
	Recall	100,00%	50,00%	**100,00%**	98,04%	99,09%	100,00%	90,00%	100,00%	50,00%
Person 6	Accuracy	87,39%	57,98%	**86,55%**	76.32%	85.83%	65.44%	86,55%	60,50%	84,87%
	Precision	78,13%	50,53%	**76,12%**	58.12%	75.17%	52.45%	76,12%	52,04%	77,05%
	Recall	98,04%	94,12%	**100,00%**	100,00%	100,00%	100,00%	100,00%	100,00%	92,16%

Table 6-2 depicts that color outperforms texture in almost all cases. Moreover, the best combination results are obtained by the Sum rule.

Considering both of these simulations, one can infer important results about classifier combination. First of all, it can be observed that if one of the classifier outputs is significantly worse than the other, the overall performance degrades and the better classifier cannot compensate for it. Additionally, it is not possible to determine a "best" combination scheme, which works in all situations. However, it can be deduced that combining classifiers yields a more robust output compared to single classifier, although the performance is sometimes lower than the single case.

Unlike the offline identification scheme (SVM), there is not a binary classification approach in Bayesian plug-in classifier. Each identity, which is observed in the scene, belongs to one of the classes in the archive or it is labeled as a new one. In Table 6-3, person recognition rates achieved with Bayesian plug-in classifier are presented, which demonstrate that this classifier gives satisfactory real-time identification results even though its performance is below that of the SVM-based classifier. For example, it cannot distinguish between person-1 and person-4 well due to the colors in their clothes.

Table 6-3. Person identification results for METU video with Bayesian plug-in classifier

	Person 1	**Person 2**	**Person 3**	**Person 4**	**Person 5**	**Person 6**
Recognition Rate	20,97%	73,33%	41,67%	82,91%	69,40%	79,52%

CHAPTER 7

CONCLUSIONS

There is an increasing demand for personal and public security systems. However, utilizing human resources in such systems builds up the expenses, as well as inconsistencies due to subjective perceptions. Besides, technological devices are vastly available in this era. All of these factors indicate the inevitable utilization of automated systems. In this thesis, an automated surveillance system is described, which includes the following four main building blocks: moving object detection, object tracking, event recognition and person identification.

7.1 Main Contributions

In this thesis, several novel contributions are obtained in the moving object detection, event recognition and person identification building blocks.

In order to extract moving objects in real-time, a hierarchical structure (two level processing) is proposed. In this way, a considerable speed-up is obtained during the segmentation stage without any degradation in object silhouettes.

In the HMM-based event recognition scheme, the selection of the number of models is achieved by utilizing coordinate clustering information, without human supervision. Hence, the proposed system might be utilized in any scenario without giving a priori information about the scene, but only some training data with typical object motion.

Final contributions are achieved in the person identification framework. The color and texture features of the segmented object regions are utilized for recognizing persons and combination of classifiers is utilized to obtain a better performance.

7.2 Discussions

Moving object detection segments the moving targets form the background and it is the crucial first step in surveillance applications. Four different algorithms, namely frame differencing, moving average filtering, eigenbackground subtraction and Parzen window-based moving object detection, are described and their performances in different outdoor conditions are compared. Parzen window approach is proved to be accurate and robust to dynamic scene conditions, considering the simulation results. A novel multi-level analysis stage is also introduced and a considerable speed up is obtained for the tested sequences. Additionally, a simple algorithm is presented to remove shadows from the segmented object masks for obtaining better object boundaries.

Object tracking follows the segmentation step and it is used to associate objects between consecutive frames in a sequence. Using the objects in the previous frame and the current frame, a match matrix is formed. Simple bounding box overlapping is used as a matching criterion while constructing this matrix. The information obtained from the match matrix is utilized in a hypotheses-based tracking algorithm. The simulation results indicate the acceptable performance of such a system in case of small number disjoint targets. However, a better association, as well as tracking method, should be required for real life scenes with many crossing and jointly moving objects.

After segmentation and tracking of the moving objects are achieved, higher-level tasks can be incorporated into the system. Event recognition is an example of such semantic processing. A hidden Markov model-based event analysis scheme is described for this purpose. Object trajectories,

which are obtained during the period of training, are utilized to form flow vectors, which contain information about the instantaneous position and velocity of the object (x, y, v_x, v_y). The position and velocity vectors are clustered separately by using K-Means algorithm and a prototype representation is achieved. Sequence of flow vectors (written in terms of prototype vector indexes) belonging to the “normal” (usually observed) motion patterns are used to train HMMs. Abnormality of a given trajectory (a sequence of vectors) is evaluated by calculating its distance to each previously trained model. Since the models are trained with normal sequences only, the distance should be high, if the trajectory is abnormal. It was observed during simulations that a single HMM is not sufficient to successfully model every possible motion in the scene. Hence, number of position clusters is utilized for the selection of model count. The simulations demonstrate the success of the presented self-learning recognition module.

Finally, a novel approach for object identification is proposed in which color structure and homogeneous texture descriptors of MPEG-7 standard are utilized to represent the visual content of the segmented object regions. However, it is observed that the seperability of color and texture features of samples varies greatly even in a single domain. Classifier combination is proposed to address this problem and 7 different combination rules are tested. Considering the results of simulations, it is concluded that the inferior classifier output degrades the overall performance significantly. Besides, it is not easy to determine a combination rule, which will give the best performance in all situations. However, combining classifiers yields more robust results compared to single classifier case. Support Vector Machine is utilized in identification tests; but its computational burden necessitates the use of another classifier, which is a simple Bayesian plug-in, for real-time operation. Although the results obtained with the Bayesian plug-in are satisfactory, SVM classifier yields better identification performances.

7.3 Future Directions

As some future work, shadow removal process can be achieved with a more robust algorithm. This case will both improve object silhouettes and tracking results. As for the identification part, an automated combination scheme should be incorporated into the system, which will automatically decide on the best combination rule with respective weights of color and texture features.

REFERENCES

[1] Haritaoglu, I., D. Harwood and L.S. Davis, “W4: A Real-Time System for Detecting and Tracking People in 2 ½ D.” 5th European Conference on Computer Vision. 1998. Freiburg, Germany: Springer.

[2] Oliver, N., B. Rosario, and A. Pentland. “A Bayesian Computer Vision System for Modeling Human Interactions.” Int’l Conf. on Vision Systems. 1999. Gran Canaria, Spain: Springer.

[3] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, “Pfinder: Real-Time Tracking of the Human Body,” IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 19, no. 7, pp. 780-785, July 1997.

[4] W. E. L. Grimson and C. Stauffer, “Adaptive background mixture models for real-time tracking.” Proc. IEEE Conf. CVPR, Vol. 1, pp 22-29, 1999.

[5] A. Elgammal, D. Harwood, and L. S. Davis. “Non-parametric Model for Background Subtraction.” In Proc. IEEE ICCV’99 FRAME-RATE Workshop, 1999.

[6] Hu, W., Tan T., Wang L., Maybank S., "A Survey on Visual Surveillance of Object Motion and Behaviours" IEEE Transactions on Systems, Man, and Cybernatics, Vol. 34, no. 3, August 2004.

[7] D. Koller, K. Danilidis, and H. Nagel. "Model-based object tracking in monocular image sequences of road traffic scenes." International Journal of Computer Vision 1993, Vol:10-3, pp.257-281.

[8] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, “Tracking groups of people,” Comput. Vis. Image Understanding, vol. 80, no. 1, pp. 42–56, 2000.

[9] N. Paragios and R. Deriche, “Geodesic active contours and level sets for the detection and tracking of moving objects,” IEEE Trans. Pattern Anal. Machine Intell., vol. 22, pp. 266–280, Mar. 2000.

[10] N. Peterfreund, “Robust tracking of position and velocity with Kalman snakes,” IEEE Trans. Pattern Anal. Machine Intell., vol. 22, pp. 564–569, June 2000.

[11] R. Polana and R. Nelson, “Low level recognition of human motion.” Proc. IEEE Workshop Motion of Non-Rigid and Articulated Objects, Austin, TX, 1994, pp. 77–82.

[12] Carlo Tomasi and Takeo Kanade. "Detection and Tracking of Point Features." Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[13] Fujiyoshi, H., Lipton, A.J., “Real-time human motion analysis by image skeletonization.” Applications of Computer Vision, 1998. WACV '98. 19-21 Oct. 1998, pp.15-21.

[14] N. Johnson and D. Hogg, “Learning the distribution of object trajectories for event recognition.” Image Vis. Comput., Vol. 14, no. 8, pp. 609–615, 1996.

[15] Rao, S. and Sastry, P.S, "Abnormal activity detection in video sequences using learnt probability densities." TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region Vol. 1, 15-17 Oct. 2003, pp. 369 – 372.

[16] C. Stauffer, W.E.L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, August 2000, pp. 747-757.

[17] Lee K. K., Yu M.; Xu Y., “Modeling of human walking trajectories for surveillance.” Intelligent Robots and Systems, 2003 (IROS 2003), 27-31 Oct. 2003, Vol.2, pp.1554–1559.

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

[19] Jain, R., Kasturi, R., Schunk, B.G., "Machine Vision", McGraw-Hill Inc., pp. 63- 69.

[20] Cavallaro, A.; Salvador, E.; Ebrahimi, T., " Detecting shadows in image sequences." Visual Media Production, 15-16 March 2004. (CVMP), pp. 165-174.

[21] S. Kullback and R. A. Leibler. "On information and sufficiency." Annals of Mathematical Statistics 22(1):79–86, March 1951.

[22] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.” Proceedings of the IEEE, Vol. 77, no. 2, February 1989, pp. 257-286.

[23] Y. Yasaroglu, "Multi-modal Video Summarization Using Hidden Markov Models For Content Based Multimedia Indexing." MS Thesis, September 2003, pp. 28-37

[24] R. Duda, P. E. Hart, D. G. Stork, "Pattern Classification." 2nd Edition, John Wiley and Science, Inc., New York, 2001, pp. 526-528

[25] H. Rowley, S. Baluja, and T. Kanade, “Neural network based face detection,” IEEE Trans. Pattern Anal. Machine Intell., Vol. 20, pp. 23–38, Jan. 1998.

[26] Turk, M.A. Pentland, A.P., “Face Recognition using eigenfaces.” IEEE Proceedings CVPR ’91, 3-6 Jun 1991, pp. 586-591.

[27] Hjelmas E., Low, B.K. “Face Detection: A Survey.”, Computer Vision and Image Understanding, 2001. Vol: 83, pp. 236-274.

[28] C. Y. Yam, M. S. Nixon, and J. N. Carter, “Extended model-based automatic gait recognition of walking and running.” Proc. Int. Conf. Audio- and Video-Based Biometric Person Authentication, 2001, pp. 278–283.

[29] R. Tanawongsuwan and A. Bobick, “Gait recognition from time-normalized joint-angle trajectories in the walking plane.” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2001, pp. 726–731.

[30] J. D. Shutler, M. S. Nixon, and C. J. Harris, “Statistical gait recognition via temporal moments.” Proc. IEEE Southwest Symp. Image Analysis and Interpretation, 2000, pp. 291–295.

[31] L. Lee, “Gait Dynamics for Recognition and Classification.” MIT AI Lab, Cambridge, MA, Tech. Rep. AIM-2001-019, 2001.

[32] C. BenAbdelkader, R. Culter, and L. Davis, “Stride and cadence as a biometric in automatic person identification and verification.” Proc. Int. Conf. Automatic Face and Gesture Recognition, Washington, 2002, pp. 372–377.

[33] B. S. Manjunath, P. Salembier, and T. Sikora, “Introduction to MPEG-7 Multimedia Content Description Interface.” John Wiley & Sons Ltd., England, 2002.

[34] M. Soysal, "Combining Image Features for Semantic Descriptions." MS Thesis, September 2003.

[35] J.C. Platt, “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.” Advances in Large Margin Classifiers, MIT Press, Cambridge, MA, 1999.

[36] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. "On Combining Classifiers." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 3, March 1998.

[37] V.N. Vapnik, "The Nature of Statistical Learning Theory." Springer-Verlag, New York, 1995.

[38] B. Orten, M. Soysal, A. A. Alatan, “Person Identification in Surveillance Video by Combining MPEG-7 Experts.” WIAMIS 2005, Montreux.