MODELLING AND RECOGNITION OF DYNAMIC EVENTS IN VIDEOS

A Thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science (by Research)
in
Computer Science

by

Karteek Alahari
200407004
karteek@research.iiit.ac.in

International Institute of Information Technology
Hyderabad, INDIA
July 2005

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Modelling and Recognition of Dynamic Events in Videos” by Karteek Alahari, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: C. V. Jawahar

_____ _____

Date Co-Advisor: P. J. Narayanan

To *Amma* and *Naanna*

Acknowledgements

I would like to thank my supervisors Dr. C. V. Jawahar and Dr. P. J. Narayanan for their support and guidance during the past four years. They have been great mentors and guided me through the ups and downs of my research career until now. I am truly grateful to Dr. Narayanan for introducing me to the field of Image Processing in the Summer of 2001; which led me into its related areas.

Many thanks to The GE Foundation for providing Graduate Scholarship during 2003-2005. A special thanks are due to my parents, who have been very supportive and understanding throughout my academic career. Last but certainly not the least, I am grateful to all the members—both past and present—of the Centre for Visual Information Technology (CVIT) for their help and stimulating company.

Abstract

Computer Vision algorithms, which mainly focussed on analyzing image data till the early 1980's, have now matured to handle video data more efficiently. In the past, computational barriers have limited the complexity of video processing applications. As a consequence, most systems were either too slow to be practical, or succeeded by restricting themselves to very controlled situations. With the availability of faster computing resources over the past couple of decades, video processing applications have gained popularity in the computer vision research community. Moreover, the advances in data capturing, storage, and communication technologies have made vast amounts of video data available to consumer and enterprise applications. This has naturally created a demand for video analysis research.

Video sequences typically consist of long-temporal objects – called events – which usually extend over tens or hundreds of frames. They provide useful cues for analysis of video information, including, event-based video indexing, browsing, retrieval, clustering, segmentation, recognition, summarization, etc. The state-of-the-art techniques seldom use the event information inherent in videos for all these problems. They either simply recognize the events or use primitive features to address other video analysis issues. Furthermore, due to the large volume of video data we need efficient models to capture the essential content in the events. This involves removing the acceptable statistical variability across all the videos. These requirements create the need for learning-based approaches for video analysis.

In this thesis, we aim to address the video analysis problems by modelling and recognizing the dynamic events in them. We propose a model to learn efficient representation of events for analyzing continuous video sequences and demonstrate its applicability for summarizing them. Further, we observe that all parts of a video sequence may not be equally important for the classification task. Based on the characteristics of each part we compute its potential in influencing the decision criterion. Another observation we make is that, a feature set appropriate for one event may be completely irrelevant for another. Hence, an adaptive feature selection scheme is essential. We present an approach to learn an optimal combination of spatial and temporal based on the events being analyzed. Finally, we describe some of our work on unsupervised framework for video analysis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the middle of the twentieth century, experts in the field of Artificial Intelligence felt that the task of making machines see was a trivial problem. Even to date, this fundamental problem remains largely unsolved and will perhaps remain so for quite some more time [120]. In the course of many efforts to achieve this dream, a new discipline has emerged – Computer Vision, which encompasses areas such as mathematics, computer science, psychology of perception, biology, neuro sciences, etc. Computer Vision deals with extracting descriptions of the world from images or sequences of images much like the human brain does from the images captured by our eyes.

The growth of Computer Vision has been helped by advances in Image Processing [41, 55], Pattern Recognition [26] and Machine Learning [76] techniques. Image processing deals with enhancing certain desired characteristics of images, the brightness of an image, for instance. In simpler terms, image processing techniques take an image as input and produce an image as output. In contrast, computer vision algorithms analyze the image (or set of images) and extract *information* as output. Image processing is thus, only a tool which aids in our objective of understanding images. For example, in Geographical Information Systems (GIS), images captured by satellites are first 'processed' to enhance their details for better (further) analysis, such as automatic extraction of roads, rivers, vegetation, etc., which are the major interest features. Pattern Recognition techniques provide strong statistical models to understand images and videos, and hence find great relevance in computer vision problems. Machine learning, in general, addresses the question of how to build computer programs that improve their performance at some task through experience. Its ultimate goal is to make computers learn and adapt over time, which would open up many new uses of computers and new levels of competence and customization [76]. Many successful machine learning applications have been developed over the years, including autonomous vehicles that learn to *see* and drive on public highways [86]. These three classes of techniques – Image Processing, Pattern Recognition, Machine Learning – form an important core of most of the computer vision solutions.

Significant progress in computer vision research has resulted in a number of applications ranging from car manufacturing to the entertainment world. Many applications in computer vision stem from the different interpretations that users seek from images [32]. Popular applications include medical image understanding, inspection of objects in manufacturing units to determine whether they are within the specifications allowed, building systems for browsing and searching image databases, understanding the geometry of the scene to introduce virtual objects into it, realistic rendering of synthetic scenes in computer graphics, military applications such as analysis of satellite image data, and video analysis to build systems for browsing, activity detection, surveillance, etc. A more comprehensive list of applications can be found in [51]. Vision techniques, which mainly focussed on analyzing image data till the early 1980's, have now matured to handle video data more efficiently. The proceedings of recent workshops [1, 3] exemplify the burgeoning interest towards video analysis in the research community.

In the past, computational barriers have limited the complexity of video processing applications. As a consequence, most systems were either too slow to be practical, or succeeded by restricting themselves to very controlled situations. With the availability of faster computing resources over the past couple of decades, video processing applications have gained popularity in the computer vision research community [98]. Moreover, the advances in the data capturing, storage, and communication technologies have made vast amounts of video data available to consumer and enterprise applications [25]. Video sequences typically consist of long-temporal objects – called events [118] – which usually extend over tens or hundreds of frames. They form a powerful cue for analysis of video information, including, event-based video indexing, browsing, clustering, segmentation, recognition, and summarization [118]. Polana and Nelson [85] classified events into three groups, namely *temporal textures* which are of indeterminate spatial and temporal extent, *activities* which are temporally periodic but spatially constrained, and *motion events* which are isolated events that do not exhibit spatial or temporal repetition. Examples of temporal textures are motion of a flock of birds, wind blown trees or grass, ripples on water, turbulent flow in cloud patterns, etc. Examples of activities include walking, running, rotating or reciprocating machinery, etc. Examples of motion events are isolated instances of throwing a ball, starting a car, smiling, throwing a ball, etc. Many previous attempts have been made to analyze these three categories of events. However, they are inefficient in handling certain aspects as we will see in the coming chapters.

Video processing and analysis has a number of promising applications in addition to the general goal of designing a machine capable of interacting intelligently with a human-inhabited environment [37]. Some of the applications include interactive virtual worlds, gaming, access control, video surveillance, gesture recognition, digital libraries and content-based video indexing, industrial monitoring, sign language recognition, wearable computing, event-based video coding, choreography of dance/ballet, gait analysis, "smart" interfaces, face expression analysis, etc. [12, 30, 37, 38, 85, 107, 116]. A more detailed discussion on video analysis applications in various domains is presented in Section 1.1.3.

The domain of video analysis is rich and challenging. Some of the challenges that the vision research community encounters are:

- The need to segment (rapidly, in most cases) changing scenes in natural environments with moving elements in the background (e.g., swaying trees).
- Robustness to lighting variations and whatever is in its visual field. The system should not depend on careful placement of cameras.
- Bulky nature of video data. The spatial and temporal redundancies in videos create the need for efficient modelling techniques.
- Demand for appropriate feature selection schemes due to the myriads of events one may observe in the real-world.
- The need for modelling techniques which handle variabilities in a large video collection as events occur with different temporal extents.
- Occlusion of objects of interest, due to other objects or different parts of the same object.
- A subject-invariant representation is essential since most applications, such as gesture recognition, activity recognition, video coding, etc., do not require identification of the subject performing the event. For example, when recognizing events performed by humans, colour of the outfits worn by them or their ethnicity is immaterial.
- Secondary issues such as the sampling rate, image resolution and nature of the video are to be considered when extracting features.

1.1 Dynamic Event Analysis in Videos : An Overview

In this section we review the state of the art in dynamic event analysis. We begin with a discussion on the modelling and characterization techniques which are popular for analyzing events in videos. In Section 1.1.2 we summarize the evolution of these techniques as a time-line. We then elaborate on the applications of event analysis (mentioned before) in various domains.

1.1.1 Modelling and Characterization Techniques

A review of the popular methods for modelling events (with the focus being on human events) can be found in [1, 37] and the references therein. Wang et al. [107] provide an extensive survey on the recent developments in motion analysis, focussing on the aspects of detection, tracking and event analysis. Most of the early techniques can be categorized into three classes, namely 2D approaches with explicit

shape models, 2D approaches without explicit shape models and 3D approaches. These methods employ segmentation and subsequent (2D or 3D) tracking of individual parts to model the dynamism in events [38, 112]. They first identify moving objects – typically referred to as blobs – which are constrained by their size or shape. Tracked trajectories or higher-order image features of these blobs are used to distinguish events. Naturally, these methods are very sensitive to the quality of segmentation and tracking of blobs. An alternate approach for modelling events is to use appearance-based features such as Motion History Images (MHI) and Motion Energy Images (MEI) [22], Pixel Change History (PCH) – a combination of MHI and Pixel Energy [111]. These methods exploit the overall form of the subject performing the event and build a feature image which describes its spatiotemporal appearance, *i.e.*, the recency and spatial density of motion. Although the early methods lead to satisfying results, they are not capable of handling the uncertainty that exists when modelling events.

The need for incorporating uncertainty when modelling events has been recognized by many researchers in the past [43, 48, 66, 81, 99]. This uncertainty occurs due to the multiple instantiations of events in a large collection of videos. Probabilistic methods such as Gaussian Mixture Models (GMMs) [43], Hidden Markov Models (HMMs) [81, 99], etc., have been used to address this issue. The ability to learn from training data and to develop internal representations with a sound mathematical framework makes models like HMM attractive. One of the first applications of HMM for event analysis was proposed by Yamato *et al.* [113]. HMMs are nondeterministic state machines which, given an input, move from state to state according to various transition probabilities [37]. Typically, in HMM-based approaches for event modelling/recognition, the set of hidden states is specified *apriori* and the transition probabilities between these states are learnt from examples [99]. Also, one HMM is used per event. Many variants of Markov models, such as Variable Length Markov Model (VLMM) [36], Layered HMM (LHMM) [81] have been proposed for analyzing events. VLMMs, which capture long-term as well as short-term temporal dependencies in events, and LHMMs, which encode the hierarchical temporal structure of a video, overcome the limitations of the traditional Hidden Markov models. Greenspan *et al.* [43] proposed a probabilistic video representation and modelling scheme using GMMs. They cluster the video into space-time blobs for detection and recognition applications. An interesting aspect of this work is the analysis of a video as a single entity rather than as a sequence of frames. This transforms the typical two-stage processing framework (frame-by-frame spatial segmentation and temporal tracking across frames) into a single-stage modelling framework of identifying spatio-temporal objects. A detailed discussion on the mathematical formulation of HMMs and other modelling techniques is provided in Chapter 2.

In certain situations, it may be desirable to model the events after extracting essential *visual* content from the video. For instance, to characterize and model activities or motion events, it is profitable to *subtract* the background. Many methods for background removal have been proposed in the past [37, 98, 107, 112, 121]. Most researchers have now abandoned non-adaptive methods of background removal since they do not account for constantly changing backgrounds. In these methods, errors in

the background accumulate over time, making them effective only in highly-supervised and short-term applications where there are no significant changes in the scene [98]. A standard method of adaptive backgrounding removal is by averaging the images over time and creating a background approximation. While this approach is useful in situations where the background is visible for a significant portion of time, it is not robust to scenes with fast-changing backgrounds. Furthermore, it cannot handle multimodal backgrounds. Stauffer and Grimson [98] proposed an adaptive background removal scheme. They model the values of each pixel as a mixture of Gaussians. The Gaussians that correspond to the background are determined based on their persistence and variance. Pixels that do not fit the "background" Gaussians are labelled as foreground as long as there is a "foreground" Gaussian that includes them with sufficient evidence. Recently an improvement to this model proposed by Zivkovic [121] chooses the number of mixtures in an adaptive fashion.

Although effective, the modelling schemes discussed so far may not be useful for some of the applications [106]. Approaches for extracting layer representations from image sequences have gained popularity [59, 65, 102, 104] ever since they were first introduced by Wang and Adelson [106]. They offer a simple yet efficient way to model and subsequently analyze video sequences. Handling occlusions is one of the important problems in event/motion analysis. Popular schemes such as those using optical flow [47] are inadequate to handle the motion discontinuities caused by occlusions. An elegant solution to this problem is decomposing the 3D scene into a set of 2D objects in layers [59]. Jojic and Frey [59] proposed a scheme to learn the appearances of multiple objects in multiple layers, over the entire video sequence. They introduce "flexible sprites", which can deform from frame to frame and thus model all the dynamic objects in the scene. This method claims to be very generic and requires the number of layers and the number of sprites as the only input. Recently, Kumar *et al.* [65] presented an approach that performs significantly better than the previously reported results.

Almost all the event-based analysis research has been accomplished to solve recognition problems. It has seldom been used for addressing higher level video analysis problems such as video summarization, indexing, browsing and searching [24, 118]. The traditional approaches are limited to detecting cuts and tracking object in extracted shots [96]. Other approaches include using the frame structure of the video in the MPEG-4 standard [46], extracting the key frames from the video [79, 97]. Zelnik-Manor and Irani [118] defined a statistical distance measure between video sequences. They used this measure to isolate and cluster events within long continuous video sequences. As a result of this clustering the long temporal sequences are temporally segmented into event-consistent subsequences. However these events cannot be identified as no prior knowledge of type of events is assumed. One of the few methods for content-based video analysis is presented by Denman *et al.* [24]. They propose three tools for this analysis – with applications to sports videos – namely, a parsing tool based on geometry (without explicit 3D computation), event detection tool, and summarization tool.

To sum up, in this section a brief review of the current state-of-the-art techniques for analyzing video sequences is presented. We infer that video analysis is still in its primitive stages of development and has many challenging problems which are unaddressed. The mathematical formulations underlying some of these models are discussed in Chapter 2.

1.1.2 Evolution of Techniques

In this section we review the evolution of various techniques for video analysis. Although motion (captured in the form of videos) plays an important role in many tasks, motion analysis in general, has received little attention in the literature compared to the volume of work on static object recognition [85]. Most computational work in motion has been concerned with various aspects of the structure-from-motion problem. There has been a spurt of interest in these problems only in the past decade [1]. Hence, we divided the time-line into five periods – before 1980, 1980-1990, 1990-1995, 1995-2000, 2000 till-date. Our findings are summarized in the table below.

Year	Characteristics	References
Before 1980	Most of the techniques dealt with understanding the perception of motion in a psychophysical sense. The influential work of Johansson [58] on Moving Light Display led to much research. The earliest attempt to recognize events was reported in [83] on synthetic images using many constraints. None of the techniques were tested on real video sequences and much of the work was rather at a high level of abstraction [15, 85].	Johansson, 1973 [58] O'Rourke and Badler, 1980 [83]
1980-1990	Video analysis techniques were restricted to recognizing simple events for surveillance applications [21]. This was achieved either by assuming explicit structural models [50] or by computing features such as spectral energy in a (temporal) difference image [8], invariant images from joint angles and angular velocities [40], event trajectories [42], etc. The general applicability of these approaches was severely limited by various factors – computation of joint angles, velocities, accuracy of the trajectory tracker. During this period some fundamental temporal pattern recognition work was done in the context of speech processing [60]. However, its applicability for event analysis was not investigated.	Cutting, 1981 [21] Hogg, 1983 [50] Anderson *et al.*, 1985 [8] Juang and Rabiner, 1985 [60] Gould and Shah, 1989 [42] Goddard, 1989 [40]

1990-1995	Common representations included space time curves [78], appearance based features [13, 16], spatio-temporal angle histograms [33]. During this period advanced modelling/recognition schemes such as HMM were first introduced in the video analysis domain [113]. A major breakthrough in the video analysis domain was The Informedia project [18]. The goal of this work was to automatically skim documentary and news videos with textual transcriptions. They create a skim video – a short synopsis of the video – by first abstracting the text using classical text summarization techniques [70], and then looking for the corresponding parts in the video. This was achieved by integration of language as well as image understanding techniques by extracting significant information, such as specific objects, audio keywords, and relevant video structure.	Yamato *et al.*, 1992 [113] Niyogi and Adelson, 1994 [78] Bobick and Wilson, 1995 [13] Christel *et al.*, 1995 [18] Campbell and Bobick, 1995 [16] Freeman and Roth, 1995 [33]
1995-2000	Event recognition in this period was characterized by methods which primarily used tracking [37, 38, 108]. It also saw a large number of DARPA funded projects on tracking, event analysis, surveillance [19, 98]. As part of the multi-institution VSAM project [19] many techniques have evolved, which includes adaptive background removal [98], multi-sensor detection and tracking [19]. The purpose of the VSAM project was to develop an automatic video understanding technology that enabled an operator to monitor events over complex areas such as battlefields and civilian scenes [107]. Methods which employed a combination of shape analysis and tracking were also popularly used [49]. The applicability of such methods for constructing the appearance-based models as well as monitoring events in outdoor environments was successfully demonstrated [89, 112]. Towards the latter half of this period, the use of example-based learning schemes started gaining momentum.	Regh and Kanade, 1995 [89] Gavrila and Davis, 1996 [38] Bregler, 1997 [15] Wren *et al.*, 1997 [108] Gavrila, 1999 [37] Yacoob and Black, 1999 [112] Stauffer and Grimson, 2000 [98] Collins *et al.*, 2000 [19] Haritaoglu *et al.*, 2000 [49]

2000 Till-date	Methods which fit event data to models fixed *apriori* are being replaced by example-based learning schemes [4, 36, 81]. Many novel techniques for representing and analyzing videos, such as layer representation [59, 65, 102], space-time blobs [43], content-based analysis [24], event clusters [118], etc., were proposed. Video sequences have also been analyzed by considering them to be made up of a sequence of events [7, 118]. As observed by Sun *et al.* [99], techniques that do not require explicit tracking or segmentation are of much interest for real-life applications.	Tao *et al.*, 2000 [102] Ali and Aggarwal, 2001 [7] Galata *et al.*, 2001 [36] Jojic and Frey, 2001 [59] Zelnik-Manor and Irani, 2001 [118] Greenspan *et al.*, 2002 [43] Nuria *et al.*, 2002 [81] Denman *et al.*, 2003 [24] Kumar *et al.*, 2005 [65]

Table 1.1 A summary of the evolution of techniques for video analysis.

1.1.3 Application Domains

Video processing finds a number of promising applications in many areas, namely, Human Computer Interaction (HCI), Storage & Digital Libraries, Interactive environments, Wearable Computing, Broadcast video, Industrial monitoring, Surveillance/Security, Entertainment, Education, etc. We will look at applications in some of these areas in detail below.

Security Systems: Building automatic systems for surveillance is a traditional application of video analysis [37]. Event recognition has become a very useful technology to monitor the situation at important locations. The recognition system can be trained to distinguish between acceptable and potentially dangerous event. This is popularly known as "unusual" event/activity detection [119]. When a system detects these events, it may trigger appropriate measures – such as raise an alarm, alert the security personnel, etc. In a parking lot setting one might want to signal suspicious behaviour such as wandering around and repeatedly looking into cars [37]. Gavrila [37] describes what are called as "smart" surveillance systems, *i.e.*, systems that do more than just motion detection to prevent false alarms (like blowing wind, animals moving around, etc.). A first requirement for these systems would be to sense if a human is present in the scene. This may be followed by face recognition for controlling access to secure installations ranging from ATMs to defence organizations. However, to get a deeper understanding of the situation, event-based video analysis is crucial.

Digital Libraries: Due to the ubiquitous nature of good quality video capture devices, a large amount of video content is available these days. Given such *libraries* of videos, one would like to have easy-to-use tools to organize them, browse through them, query them, and retrieve the snapshots of particular interest. The bulky nature of videos creates a need for efficient representation and

storage. Event-based video coding is an interesting research to achieve this [37]. Automatic content-based annotation of videos is another important application in this domain. Content-based video retrieval systems essentially depend on this annotation. Furthermore, when browsing through a large video collection it is useful to have a way of being able to *skim* through the contents. This is the problem of video summarization, which is analogous to text summarization [63, 70] in many ways. An example scenario is one where security personnel browse through the surveillance video, captured during the entire day, to detect any anomalies. It is needless to say that the process is tremendously speeded-up when they use the *skim*-video instead of a large video collection. Event analysis can be profitably used in all these applications.

Human Computer Interaction (HCI): One of the important applications of modelling and recognizing events is in the area of Human Computer Interaction. When designing machines capable of interacting intelligently with a human-inhabited environment it is inevitable for them to have an understanding of human events [37]. Such machines will know when you are looking at them, will be able to recognize your gestures, and will detect where you are pointing [27]. This analysis can also complement speech recognition and natural language understanding. These types of gesture-based interfaces are part of a growing trend toward developing non-invasive and intuitive interaction between computers and humans. Specific applications include areas where the traditional interfaces – keyboards and mouse, for instance – are not effective. This would improve the way we visualize CAD models, interact in computer games, and even control household appliances [82].

Virtual Reality: Application areas in the Virtual Reality domain lie in interactive virtual worlds, gaming, virtual studios, character animation, teleconferencing. Event analysis can enrich the interaction among the participants or objects by adding gestures, head pose, and facial expressions as cues [37]. Application systems such as FingerMouse, FingerPaint [20, 87] are a result of some of the specialized gesture recognition based devices developed for better interactivity.

Education & Entertainment: Event modelling and analysis has innumerable applications in education and entertainment domains. It could be used in choreography of dance/ballet or teaching dance steps in a controlled environment, e.g., KidsRoom [12]. Noninvasive methods to track and analyze in video sequences have helped devise realistic models. They find applications in crash simulations, synthesis of human motion, etc., which are quite commonly used in current motion pictures. An interesting application in the field of education is "intelligent" tutors [27, 61]. They judge whether a student is confused, or confident based on his/her actions and moods. Broadcast video analysis is another active application area with regard to event analysis [2]. Many tools have been developed to summarize video clips broadcast on television.

The aforementioned list of application domains is only indicative and by no means exhaustive or complete. However, it demonstrates the importance of the problem and motivates the development of robust and efficient models for modeling video events.

1.2 Motivation

The motivation for this thesis stems from the fact that machines, more specifically computer systems, have largely been "blind", with little understanding of their surroundings. Although developing techniques for making computers see has been the focus of much research over the past decades, it is mostly in the domain of image understanding. It is a known reality that videos, typically comprising of a large collection of images, contain much more information about the scene when compared to a single or multiple image(s). They may provide details of the scene from multiple view points, temporal relations that occur in the scene, an understanding of what different objects do, etc. However, video analysis research is still in its preliminary stages even to date. There have been very few attempts to learn higher level representations from videos. The limited research and breakthrough technologies in this domain may be partly attributed to the hardware limitations that existed till quite recently. But with the ubiquitous nature of good quality video acquisition devices, and the fading of most computational barriers, video analysis and processing has received great attention from computer vision researchers of late.

A desirable application for any user viewing a video (e.g., a recorded cricket video) could be to point out an interesting video segment containing an event of interest (e.g., a short clip which shows a player hitting a six in a cricket match), and request a "system" to fast-forward to the next clip which shows a similar action. Such applications require appropriate video indexing/browsing schemes or event-based similarity measures [118]. A more challenging task is to map a text query into a set of matching video clips. The current video indexing schemes which analyze the sequence frames based on the visual content alone are ill-equipped to handle these problems. We need techniques which understand *what is happening in the video*. This is a hard problem to tackle, and can be addressed by learning from examples. Understanding video contents is also useful when segmenting a video *document* into shots and scenes to compose a table of contents. We may also extract keyframes or key sequences as index entries for scenes or stories.

The availability of video cameras to the common man has created even more research challenges. These issues are succinctly stated by Dimitrova *et al.* [25] as follows.

> "At the other extreme, for consumers, the products and applications need to be extremely simple for them to be viable in the marketplace. As an example, increasing consumer access to electronic imaging devices such as still digital cameras and digital camcorders has resulted in an explosion in the volume of data being generated. For consumers to annotate, index, handle, process, and access their data, products must be designed with simple yet useful functionalities. This would preclude searching techniques that are, for example, based on color histograms. Instead, consumers might want to find all the pictures in which uncle Joe is with the baby by defining once and for all, who uncle Joe and baby

are pictorially. Clearly, from a technical point of view, this is harder to solve than searching color histograms. At the moment, no technically robust solutions exist for this problem."

In addition to this, consumers may want to query their video collection to retrieve all videos where the baby is walking. This requires the system to have *apriori* knowledge of both walking and the baby. Identification of the baby is more or less a solved problem [32], but an efficient way to understand what the baby is doing (walking, running, etc.) is an active research area which motivates this thesis. Another important factor is the voluminous surveillance data available for analysis. The ultimate goal would be to build a system which processes all this data and comes up with an understanding of most of the events that occur in real-world situations. It may also learn the *kind* objects that perform certain events, typical duration of each event, diurnal effects, etc.

1.3 Objectives

Any video sequence is essentially a set of events or temporal objects. We believe decomposing a video into individual "event components" leads to better understanding of its contents. It may also be a crucial preprocessing step in most of the applications discussed above. For instance, event-based analysis can be effectively used in video summarization, video coding, etc. The purpose of this thesis is to present efficient event-based modelling and recognition techniques. In particular, we aim to

- Build an efficient representation for videos in terms of their event content.
- Demonstrate the use of event recognition to summarize video sequences.
- Present better feature selection schemes for recognizing events.
- Develop event modelling and recognition techniques which *learn* to adapt to the set of events in consideration.
- Propose an unsupervised framework for modelling events.

1.4 Organization of the Thesis

This thesis focusses on three main issues, namely, event-based summarization, feature selection for event analysis, and an unsupervised framework for describing events. So far in this chapter we have presented an overview of dynamic event analysis in videos. In Section 1.1.2 we discussed the evolution of techniques for modelling and characterizing events. The applications of video analysis in many areas, such as Security Systems, Education & Entertainment, Virtual Reality, Human Computer Interaction, Digital Libraries, etc., are provided in Section 1.1.3. We present the motivation behind our work in Section 1.2 and discuss the objectives of this thesis in Section 1.3. The rest of the thesis is organized as follows.

- In Chapter 2 we present a review of some of the mathematical models for video analysis which are also referred in our work. These models include Principal Component Analysis (PCA), Linear Prediction Coding (LPC), Hidden Markov Model (HMM), Gaussian Mixture Models (GMM), Mixture of Factor Analyzers (MFA). PCA is one of the popular methods for finding an optimal set of features that constitute an object/event in question. Linear Prediction and HMM are schemes which model the temporal relations in data. LPC is a simple model which works by minimizing a least squared error, while HMM follows a probabilistic approach to estimate the temporal characteristics of data. GMM is a sophisticated statistical density estimation technique which is used for clustering data, and MFA is essentially a reduced dimension mixture of Gaussians.

- In Chapter 3 we present an event-based summarization technique. Video summarization is an important step in building fast and accurate content-based retrieval systems. It involves providing a description of the important constituents of videos. The state-of-the-art techniques seldom analyze the "events" in videos. They are limited to detecting shot and scene changes. We propose a method to learn a low-dimensional representation of an event set, using which we label the content in individual frames of an unseen video following a maximum likelihood approach. This is an extension of our previous work on activity recognition (refer Appendix A). We demonstrate the applicability of the new proposed scheme on various video sequences and compare our results to those reported in literature.

- In Chapter 4 we investigate the core problem of feature selection – identification of appropriate features – for event recognition. The problem of selecting features for analyzing events has seen few advancements over the years. Most of them are trivial extensions of image feature selection methods and treat all parts of a sequence as being similar. In this work we demonstrate that all parts of a video sequence are not equally important when distinguishing between classes. We propose an approach to identify the discriminatory potential of video segments and use it to compute a weighted similarity measure. We present results on hand gesture and human activity videos. Examples from online handwriting recognition, which is another form of sequential data, are used to supplement our discussion.

- In Chapter 5 we present a basic model for automated analysis of videos. Given video sequences recorded for potentially long duration, we first detect the various objects captured by the camera. We use features from these objects and find the most likely model and the parameters that describe their state over time. The formulation was tested on synthetic as well as real video sequences captured at a traffic junction.

- In Chapter 6 we present a summary of the results obtained, the contributions made, and suggestions for future work.

- In Appendix A we summarize our prior work on activity recognition. This work is based on the assumption that activities are composed of homogeneous units, *actions*, many of which are

common to more than one activity. The problem of recognizing activities is transformed to that of recognizing the actions and the temporal relations that exist between them. The actions and the transitions among them are learnt from examples in a low-dimensional space. Results on various recorded and publicly available videos are shown here. We also present a statistical justification of our model.

Chapter 2

Mathematical Models for Video Analysis

A video sequence consists of a highly correlated set of images, referred to as frames, captured at regular instants [103]. Owing to the 2-dimensional nature of the spatial extents of frames, video sequences are 3-dimensional. The additional third dimension corresponds to the temporal aspect of the video. Events, which are temporal objects spanning over tens or thousands of frames, constitute video sequences. Many mathematical models have been proposed in the past to analyze videos either directly or by the interpreting the events in them[1]. In this section, we look at some of these models relevant at various stages in the video analysis framework.

2.1 Principal Component Analysis

Due to the large size of video data, it is inefficient and impractical to model the events in videos directly. Modelling videos in a low-dimensional space is the solution to this problem. In these methods the objective is to find an optimal set of features that constitute each event. Many schemes for learning a low-dimensional representation of bulky data exist in computer vision [39, 56, 77, 105, 112, 114, 115]. Principal Component Analysis (PCA) is one such classical statistical method. It is a linear model based on the statistical representation of a random variable. PCA finds a compact representation by minimizing the correlation in the data. In terms of mean squared error, PCA is considered to be an optimal linear dimensionality reduction method.

Consider N samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ in a p-dimensional space. Let $\mathbf{y}_i$ represent the corresponding low-dimensional representation of $\mathbf{x}_i$. In this case PCA proceeds as follows. The d most significant eigen vectors of $\Sigma = \frac{1}{N}\sum_{i=1}^{N}[\mathbf{x}_i - \mu][\mathbf{x}_i - \mu]^T$ are chosen based on the corresponding d largest eigen values. The matrix $\mathcal{A}$ ($d \times p$) formed by arranging the eigen vectors as rows is used to obtain the subspace representation as $\mathbf{y}_i = \mathcal{A}\mathbf{x}_i$. The d principal components of the given examples capture the maximum variations in the data. PCA is an optimal linear dimensionality reduction method. Furthermore, it is easy

[1] Refer to Chapter 1 for an overview of video analysis techniques.

to compute the model parameters from the data directly. These benefits have made PCA an attractive scheme for computing the low-dimensional manifold. PCA representations are computed in many ways; SVD being the most popular among them [112]. Roweis [91] proposed an EM algorithm for the same. This algorithm permits an efficient computation of eigenvalues and eigenvectors even in the presence of missing data points. To overcome some of the limitations of PCA, methods such as Robust Principal Component Analysis (RPCA) [29], Sensible PCA (SPCA) [91] etc. have been devised. However, the lack of appropriate noise model in PCA remains as a major disadvantage [91].

2.2 LPC and Time Series Models

An important cue in video analysis or the events that constitute videos is the inherent dynamism. Events are marked by smooth variations over time. Linear Prediction Coding (LPC) is a popular scheme for modelling such sequential data [72, 109]. For a sequence of N data points $\mathbf{X} = \{\mathbf{x}_i\},\ i = 1, 2, \ldots, N$, a p th order linear predictor relates a sample $\mathbf{x}_i$ to its previous p samples as

$$\hat{\mathbf{x}}_i = a_1\mathbf{x}_{i-1} + a_2\mathbf{x}_{i-2} + \ldots + a_p\mathbf{x}_{i-p},$$

$i = (p+1), (p+2), \ldots, N$, where $\hat{\mathbf{x}}_i$ denotes the prediction of $\mathbf{x}_i$. The coefficient vector $\mathbf{a} = [a_1, a_2, \ldots, a_p]$ is estimated by minimizing the sum of squared errors $\sum_i ||\hat{\mathbf{x}}_i - \mathbf{x}_i||^2$. In the case of video sequence data, the vector $\mathbf{a}$ captures the temporal correlation among the samples of $\mathbf{X}$, which is an ordered sequence of frames $\mathbf{x}_i$. Using a Linear prediction scheme, video sequences can be modelled economically using the initial few frames and the coefficient vector.

Dynamic Time Warping (DTW) [94] is a method to find an optimal match between two given sequences (e.g., time series). The sequences are "warped" non-linearly to match each other using Dynamic Programming. DTW has been previously used to recognize events [44]. The cost of aligning two video sequences, $D(p, q)$, of lengths p and q is given by

$$D(i,j) = \min \left\{ \begin{array}{l} D(i-1, j-1) \\ D(i, j-1) \\ D(i-1, j) \end{array} \right\} + d(i,j),$$

where $d(i, j)$ is the local cost in aligning the i th element of the first sequence and the j th element of the second. This cost can be used to match a test pattern with a reference pattern.

2.3 Hidden Markov Models

Hidden Markov Models (HMMs) [88] are one of the most successful approaches for modeling and classifying sequential time series data. HMMs have gained increasing attention in computer vision

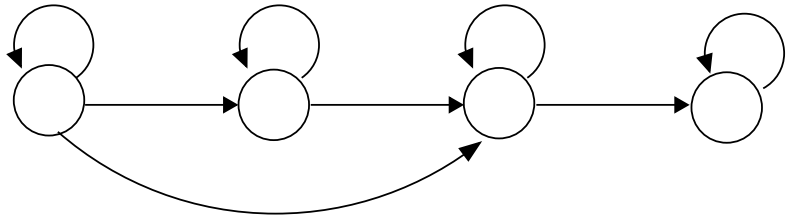


Figure 2.1 Illustration of a 4-state standard left-right HMM showing the states (denoted by circles) and the transitions.

related areas in the recent past, specifically in online handwriting [84], gesture and activity recognition [4, 81, 99]. They are popular since they offer dynamic time warping, a training scheme and a strong mathematical framework [14]. HMMs are non-deterministic state machines which, on input, move from one state to another following the transition probabilities. An example of a standard 4-state left-right HMM is illustrated in Figure 2.1. They generate output symbols probabilistically in each state. The use of HMMs involves a training and a classification phase. In the training phase the number of hidden states are specified *apriori* and the state transition and output probabilities are learnt such that the generated output symbols match the features of the respective dynamic event. In the classification (testing) phase, the probability that a particular HMM could have generated the test sequence is computed [37].

Mathematically, each event E_i is modelled by a corresponding HMM $\mathcal{H}_i$, where $i = 1, 2 \ldots K$. The parameters of the model $\mathcal{H} = \{\Xi, A, B, \pi\}$, where Ξ is the set of states, $A = \{a_{jk}\}$ is the transition probabilities matrix, $B = \{b_j\}$ is the observation symbol probability corresponding to state j and π is the initial state distribution, are estimated from the training sequences. The video sequence Φ is identified by computing the posterior $P(\mathcal{H}_i|\Phi)$, $\forall i$. A test sequence which contains only one event is labelled as E_p, whose model $\mathcal{H}_p$ gives the highest posterior score. One of the concerns in using HMMs is that the events are modelled in a high dimensional space. There have been attempts to use PCA-based features in HMMs [99] to reduce the dimensionality. However, such models are not generic enough, as they do not encode higher order temporal dependencies easily [36]. Another problem is that iterative optimization methods used for solving HMMs often lead to local optima.

2.4 Gaussian Mixture Models

Mixture Models form a class of density model which comprise a number of component functions. All these component functions combine to produce a multimodal density. Each sample $\mathbf{x}_i$ arises from a probability distribution with density given by

$$P(\mathbf{x}_i \mid \theta) = \sum_{k=1}^{m} p_k h(\mathbf{x}_i \mid \eta_k),$$

where $\theta = \{p_k, \eta_k\}_{k=1}^{m}$ denotes the set of parameters to be estimated and $h(\mathbf{x}_i \mid \eta_k)$ is the probability distribution parametrized by η_k. p_k (≥ 0), which denotes the mixing proportion of the distribution $h(.|\eta_k)$, is such that $\sum_{k=1}^{m} p_k = 1$. In the case of a Gaussian Mixture Model (GMM) the component functions follow a Normal distribution. In other words, $h(\mathbf{x}_i \mid \eta_k)$ is parameterized in terms of mean μ_k and variance Σ_k and is given by

$$h(\mathbf{x}_i \mid \eta_k) = \frac{1}{2\pi|\Sigma_k|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(\mathbf{x}_i-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}_i-\mu_k)}.$$

GMMs are one of the most widely used methods for unsupervised clustering of data, where clusters are approximated by Gaussian distributions, fitted on the provided data. The set of parameters are typically estimated following a maximum likelihood algorithm for fitting a mixture model to a set of training data. Expectation Maximization [23] is a well established technique to achieve this. In the E-step the likelihood of the distribution of the hidden variables, given the current estimates of all the parameters, is computed, while in the M-step the parameters are updated using the likelihoods computed in the E-step. These two steps are repeated until convergence[2].

2.5 Mixture of Factor Analyzers

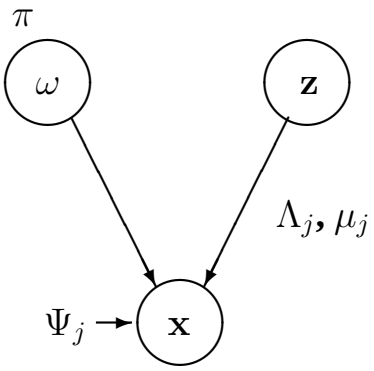


Figure 2.2 The generative model of Mixture of Factor Analysis.

Mixture of Factor Analyzers (MFA) is a linear dimensionality reduction model which also clusters the data. It combines Factor Analysis (FA) [28] with Gaussian Mixture Model. In other words, it clusters the given data probabilistically in a subspace. Let the total number of samples be N and let x_t (of dimension d), $t = 1 \ldots N$ denote the t th frame. The subsequent frames of an event are highly correlated and therefore, for each x_t, a p-dimensional $(p \ll d)$ representation z_t exists. That is, x_t is modelled as $x_t = \Lambda_j z_t + u$ where Λ_j represents the transformation basis for j th cluster and u is the associated

[2]Interested readers are encouraged to refer [11], a tutorial on Expectation Maximization algorithm for GMM parameter estimation.

noise. Multiple sequences, occurring across different events, are used to learn Λ_j for each cluster and the corresponding low-dimensional representation.

If we consider a mixture of m FAs (denoted by $\omega_j, j = 1, \ldots, m$), where each FA has the same number of p factors, but different means (μ_j) and factor loading matrices (Λ_j), the generative model is given by the following

$$p(x_t) = \sum_{j=1}^{m} \int p(x_t \mid z_t, \omega_j) p(z_t \mid \omega_j) P(\omega_j) dz_t. \tag{2.1}$$

The MFA generative model is shown in Figure 2.2. The parameters in this mixture model are given by $\{(\mu_j, \Lambda_j)_{j=1}^m, \pi, \Psi\}$, where π is the vector of adaptable mixing proportions, $\pi_j = P(\omega_j)$. These parameters are estimated with an EM algorithm [39]. The E-step (Inference) and the M-step (Learning) are discussed below.

Inference: In this step, the current estimates of parameters are used to compute the *expected* values for various interactions of the subspace representation and the mixtures. In other words, we compute $E[\omega_j \mid x_i]$, $E[z \mid \omega_j, x_i]$ and $E[zz^T \mid \omega_j, x_i]$ (for all classes ω_j), all of which can be obtained from Equation 2.1. These quantities, are computed according to

$$E[\omega_j z \mid x_i] = h_{ij} \beta_j (x_i - \mu_j) \tag{2.2}$$

$$E[\omega_j z z^T \mid x_i] = h_{ij} (I - \beta_j \Lambda_j + \Lambda_j (x_i - \mu_j)(x_i - \mu_j)^T \beta_j^T), \tag{2.3}$$

where

$$h_{ij} = E[\omega_j \mid x_i] = \pi_j \mathcal{N}(x_i - \mu_j, \Lambda_j \Lambda_j^T + \Psi)$$

$$\beta_j = \Lambda_j^T (\Lambda_j \Lambda_j^T)^{-1}.$$

Here, each $\mu_j, j = 1 \ldots m$ denotes the representative appearance for the corresponding mixture, while $\Lambda_j, j = 1 \ldots m$ denotes the corresponding subspace bases. π denotes the mixing proportions of actions in the activity set while Ψ is a measure of noise present in the data. h_{ij} can be interpreted as the membership of x_i to class j – the higher the value of h_{ij}, the more likely that x_i belongs to class j. In this manner, we *infer* the values of the subspace representations of x_i and the classes to which they belong to.

Learning: In this step, the statistics collected during the inference from *all* the training examples are used to obtain better estimates of the parameters. We solve a set of linear equations to find π_j, Λ_j, μ_j and Ψ. The interested reader may refer to the work of Ghahramani and Hinton [39] for more details. x_i is assigned to a class c_i according to

$$c_i = \arg\max_j h_{ij} \quad j = 1 \ldots m \tag{2.4}$$

The EM algorithm for MFA model can be summarized as follows.

1. Initialize the parameters π_j, Λ_j, μ_j and Ψ.
2. In the E-step, compute the expectations using the Equations 2.2 and 2.3.
3. In the M-step, estimate the parameters π_j, Λ_j, μ_j and Ψ using the equations given in [39].
4. Repeat steps 2 and 3 till convergence.
5. Assign each sample x_i to a class c_i according to the Equation 2.4.

To sum up, Mixture of Factor Analyzers model is essentially a reduced dimension mixture of Gaussians. Each factor analyzer fits a Gaussian to a portion of the data, weighted by the posterior probabilities h_{ij}.

2.6 Other Models

Many other models for representing videos (or events in videos) have been reported in literature [107]. They include Neural Network (NN), variants of HMM and NN such as Variable Length Markov Model (VLMM), Coupled HMM (CHMM), Time-Delay NN (TDNN), optical flow-based methods [85], temporal templates [12], finite automata or semantic description based approaches. For further details or relevant references for these models, the reader may refer to the comprehensive survey by Wang *et al.* [107]. Although it presents a survey of human motion analysis techniques, it covers a broad spectrum of generic frameworks.

Chapter 3

Event-based Summarization

Video summarization has been a topic of great interest over the past several years due to its innumerable applications [2, 9]. It involves providing a description of the important constituents – the summary – of the video concerned. Due to the bulky nature of video data, we need summarization algorithms to process it efficiently for browsing and retrieval applications [92]. Beginning with the Informedia Digital Video Library Project [18] many attempts have been made to interpret video content. To enable fast and accurate content access to video data, current techniques segment the video *document* into shots and scenes. The keyframes or key sequences from these shots and scenes are used for indexing the video [2, 46, 69, 79, 117]. Such an analysis provides only a low-level understanding of the video content. Therefore, the core research problem in content-based video indexing/retrieval is developing technologies to automatically parse video to identify meaningful composition structure and to extract and represent content attributes of any video sources [25].

A video can be perceived as a document. Hence, video summarization is in many ways analogous to text document summarization. Text summarization is the problem of extracting (or abstracting) from large text databases. This is achieved mostly by extracting important sentences from the text data or by abstracting the text using advanced Natural Language Processing methods [63, 70]. Unfortunately, video summarization has not reached such a stage yet. The state-of-the-art techniques do not provide a scheme for abstraction of video content, *i.e.* synthesizing a new (short) video with the essence of the original (large) video. We also limit ourselves to the extraction problem in this work, where we identify the important segments of the original video.

In this chapter we present an approach to summarize video sequences by analyzing the constituent events in them. We begin by learning a low-dimensional representation of an event set, preserving the temporal relations in it (see Section 3.2). Using this representation we label the content in individual frames of an unseen video following a maximum likelihood approach. The labellings are then grouped to build a hierarchical interpretation of the video. The proposed method finds related work in three domains: (1)

Traditional summarization techniques (e.g., Cut detection schemes), (2) Event/Activity[1] Recognition in video sequences, and (3) Event analysis in continuous videos.

Traditional summarization schemes: A popular way of summarizing videos is by detecting cuts and tracking objects in the extracted shots [96]. Other approaches include using the contents (audio/visual information) in the video [79, 97], frame structure of the video in the MPEG-4 standard [46], etc. Most of the summarization techniques developed so far present the summary as a set of *key frames* extracted from the video [64, 79]. These frames are organized spatially in many forms to indicate the video summary [64]. Such schemes are limited by the complexity of the video being summarized and are not intuitive. Although analyzing the events in the video provides more useful information, it has seldom been used in summarization techniques [24].

Activity Recognition: The problem of analyzing video sequences to characterize the activities in them has received much attention in the recent past [12, 38, 72]. A discussion on the methods for modelling activities and their importance to Human Computer Interaction and video surveillance can be found in [1, 37]. Initial attempts at solving this problem were based on segmentation and tracking of individual moving parts [38, 112]. Bobick and Davis [12] used temporal templates as an alternate approach. Although these schemes produced satisfactory results, they are limited in modelling the uncertainty in activities. Probabilistic methods such as time-series models, Gaussian Mixture Models, Hidden Markov Models are becoming popular to achieve this [48,66,81,99].

Activity analysis in continuous video: Ali and Aggarwal [7] presented a restricted model for segmentation and recognition of human activities in a continuous video. Their approach is limited to lateral views of the subject, since they use the angle of inclination of body components. We propose a method to overcome this limitation and handle videos with both lateral and frontal views of the subject performing the activity. Zelnik-Manor and Irani [118] defined a statistical distance measure between video sequences and used it to analyze continuous videos. This approach processes the videos in a high-dimensional space.

The proposed approach combines the advantages in the above three domains. We present a probabilistic method which models the activities economically in a low-dimensional manifold. In addition to the spatial redundancy (which is well studied in image processing algorithms using statistical and structural methods), videos have temporal redundancy due to the smooth variation of the scene over time. Our representation of activities exploits these redundancies. Furthermore, we model an activity as a sequence of atomic spatiotemporal units, henceforth referred to as *actions*. Thus, characterization of activities can now be modelled as that of identifying the constituent actions and their sequence order. We first estimate the individual actions and their compositional rules for the corresponding activities, given video segments. The low-dimensional representation for the set of given activities is built, which

[1] Since the technique described in this chapter is applicable to both *events* and *activities*, which are defined in Chapter 1, we use these terms interchangeably.

is used to summarize a given video. Assuming the video consists of a series of either known or unknown activities, we segment it into individual activities and subsequently annotate each of them. This annotation naturally leads to summarization of the entire video as discussed in the sections to follow.

The remainder of the chapter is organized as follows. A brief description of the event-based summarization framework is given in Section 3.1.2. In Section 3.2.1 we provide an introduction to the underlying probabilistic formulation, followed by a description of the mixture model in Section 3.2.2. The learning framework and the algorithm are outlined in the remainder of Section 3.2. The summarization and representation schemes are provided in Section 3.3. Section 3.4 illustrates the results on summarization of various video sequences (human activity, Cricket, aerobics). A comparison of our results to those reported in literature are also provided here[2]. Conclusions and avenues for future work are discussed in Section 3.5. In the following section we provide a background on the characteristics a summarization system should possess and review a traditional cut detection method.

3.1 Video Summarization

The goal of video summarization is to provide the essence of the video. Ideally, summarization systems should possess characteristics such as a browsable description of the video, a hierarchical structure, a view of the highlights. These desired qualities and the state-of-the-art in providing them are presented in further detail below.

Browsable Description: Cut detection and similar approaches generate the summary as a sequence of short clips [64]. This achieves a temporal segmentation of the video and may be used to generate a browsable description. A major drawback of these approaches is the requirement of manual intervention for labelling each clip.

Hierarchical Structure: Representing the video as a hierarchy of events makes it more structured and organized. Traditional schemes do provide this hierarchical structure [79, 96].

Table Of Contents (TOC): Viewing the video as a TOC provides easy access to the content a user may be interested in. Shot or cut detection approaches are useful for generating a table, but lack the ability of identifying the content without explicit human intervention.

Video Highlights: One of the important requirements of summarization tools is to present *highlights* of a given video sequence. Achieving this with no manual intervention is still an unsolved problem. Most techniques provide video highlights by taking explicit input from the user.

Indexing and Retrieval: To build video retrieval systems on the lines of text-based search systems, videos need to be indexed. Learning schemes such as the one presented in this chapter are ideal for indexing a large collection of videos.

[2]Part of the implementation was done by Ankit Kumar, B.Tech. 2005.

3.1.1 Cut detection

Figure 3.1 Detecting cuts in a broadcast news clip. A sports segment of the broadcast is illustrated here. The left panel shows a hierarchical structure of the video sequence. Key frames representing the shots in the video are shown in the middle panel, while the right panel shows a controller which plays the selected shot from the video.

A simple way of segmenting and summarizing videos is by using the cut detection approach [64, 101, 117]. A cut is defined as a sudden shot change in a single frame, where a shot is an unbroken sequence of frames from one camera. Some of the earliest techniques for cut detection were based on pixel differences between two consecutive frames. A block-wise comparison of every sequence frame with a few of the previous frames is also used to detect cuts. Certain methods also used colour histogram measures to confirm the results obtained, inorder to avoid spurious detections caused by camera motion. These methods developed into more stable statistical techniques at a later stage [117]. Other approaches include using edge tracking, histogram differences and motion vector information [2]. The results of cut detection schemes are typically illustrated in a browsable and hierarchical format (as can be seen in Figure 3.1).

In Figure 3.1 we show cut detection results using the block-wise scheme (16×16 sized blocks) on the sports segment of a news clip. Sports segments of news clips typically include scenes from different broadcasts (for example, Cricket broadcasts as in this case). Hence, they have a large number of abrupt shot changes making them ideal for testing cut detection approaches. Figure 3.1 shows a tool that presents a hierarchical structure of the entire video sequence in the left panel and the representative key frames of the shots are in the middle panel. The major limitation of this summarization tool is the lack of

the ability to provide a browsable description with minimal manual intervention. In order to overcome this, we need a mechanism to characterize/model and then identify the video content automatically.

To sum up, cut detection approaches provide only a low-level abstraction of the video in terms of scene changes and in some cases, the important scenes. Providing a higher level abstraction helps in many situations (for e.g., video browsing, video retrieval). Moreover, cut detection approaches are not viable for continuous video sequences. In the next section we outline our event-based summarization approach.

3.1.2 Annotation of Dynamic Events

The event-based summarization scheme we present in this chapter has most of the ideal characteristics discussed in the previous section. We believe it is an important step towards achieving an ideal video summarization system. Our approach allows for labelling video segments with minimum manual intervention. The intervention is limited to the training phase, where a labelled collection of videos is required. Given an unseen video, it is annotated automatically using the learnt representation. Furthermore, we generate the summary with different description details. An example of the summary generated can be seen in Figure 3.11. Here we perform a temporal segmentation of the video, analysis and recognition of the activities and also provide a textual description of the video contents. The video summary is also presented in an XML format (refer Section 3.3). An XML representation is useful for indexing and retrieval of a large collection of videos. Text query systems may easily incorporate this data and build on-demand video retrieval systems. User-defined video highlights can also be retrieved using this query system. For instance, the user may query the system to retrieve all Tennis video segments in which a particular player exercises forehand shots, or all Cricket video segments in which the batsman plays a Hook shot. Identifying the video contents in a learning-based framework allows for more useful and descriptive TOC unlike those given by the traditional methods.

We use a probabilistic approach for summarizing and representing video sequence data by analyzing its contents. A traditional cut detection approach is insufficient for summarizing continuous videos as there may not be any “cuts” in the video. Our approach is independent of the (in)existence of cuts in the video and hence overcomes this limitation. We begin by learning a low-dimensional representation of various activities. It may be noted that this approach is more suitable for domain-specific summarization, as it is impractical to train the system on all possible set of activities/events. The learnt representation is used to summarize any new test video. The training (offline) and testing (online) phases of our method are outlined below.

Training: Learn the representation of activities in videos.

1. Identify the actions in the given set of activities.
2. Represent the activities as a mixture model of actions.

3. Thus, learn an efficient representation of the activities.

Testing: Summarize an unseen video.

1. Follow a maximum likelihood approach to annotate all the frames of the video.
2. Generate a hierarchical structure.
3. Build an XML representation of the video, which is useful for indexing and retrieval.

3.2 Modelling and Analyzing Videos

In this section we provide details of: (a) the model, (b) the learning scheme, and (c) the method for identification of the most likely activities in the video and its subsequent annotation.

3.2.1 The Model

We consider videos to be made up of a sequence of activities/events. In a typical scenario, these videos may consist of a variety of activities. Let $A_1, \ldots A_K$ denote the subset of activities, whose videos are available during the training phase. We refer to these activities as the *known* activities and the rest as *unknown* activities. Our objective is to automatically extract the activities in the sequence, identify the known ones, and eventually summarize the entire video. We propose a method to learn the representation of the known activities and use it in segmenting the video (across time) into identifiable parts.

Let the total number of frames from examples of all the activities be N and let $x_t,\ t = 1 \ldots N$ denote the tth frame. Subsequences of x_t constitute actions, which form the basic units of our model. For every d-dimensional x_t, there exists a p-dimensional representation z_t, with $p \ll d$, due to high correlation among the frames. Using a linear dimensionality reduction model gives $x_t = \Lambda_j z_t + u$, where u is the associated noise and Λ_j denotes the transformation basis for the jth action. The transitions across actions follow a unique probabilistic structure for each activity. Thus, estimating the actions and the transitions among them provides a representation for the known activities. The low-dimensional representations for the actions are learnt from multiple subsequences occurring across different activities. The transitions are learnt by observing z_ts across the various actions for each activity. This model is similar in spirit to a standard left-to-right HMM [99]. However, we work at a lower dimension, which is simultaneously obtained while modelling the activity structure.

Given this model to analyze activities, we use it to generate summaries of videos. Consider a new video sequence with known and unknown activities. A low-dimensional representation corresponding to this sequence is constructed using the previously learnt representation. The known activities in the video are recognized, thereby annotating a subset of the given frames. This provides a higher level of abstraction

for chunks in the video. Assuming a small number of unknown activities, most of the given sequence can be replaced by textual description of the activity.

In the remaining subsections, we elaborate on the scheme for learning various actions and the associated activities. The algorithm to summarize videos is also described in detail.

3.2.2 Mixture Modelling

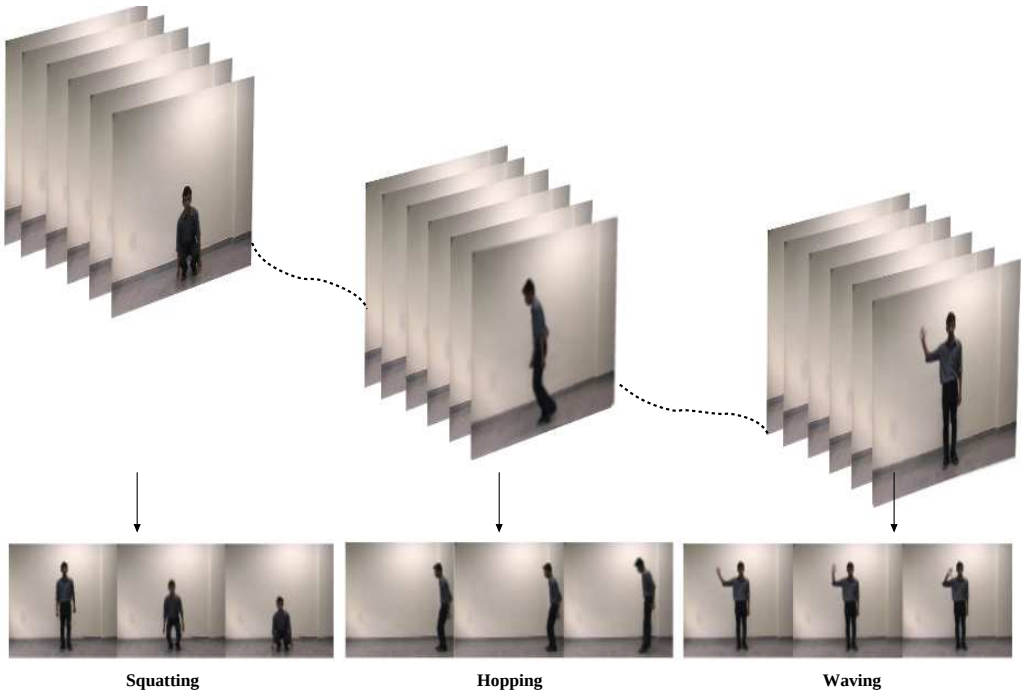


Figure 3.2 A few sample frames from a continuous video with activities such as Squatting, Hopping (Kangaroo hop) and Waving.

As can be seen in Figure 3.2, there exist actions common to different activities (the action 'Standing' can be seen here). Hence, using a mixture model of actions can be profitable. Gaussian Mixture Models (GMMs) form a special case wherein the mixtures follow a normal distribution, in a high-dimensional space. On the other hand, Mixture of Factor Analyzers (MFA) model achieves the same in a low-dimensional space. In other words, it is essentially a reduced dimension mixture of Gaussians. We use the MFA model to get a subspace representation of the actions. To model the complete activity we observe the transitions of the frames between actions.

Let us consider the process of generating a typical frame x_t using a mixture model of actions. The action to which it belongs to is chosen according to the discrete distribution $P(\omega_j),\ j = 1 \ldots m$. A continuous subspace representation z_t is generated according to $p(z_t \mid \omega_j)$, depending on the chosen action. Having

obtained z_t and the action ω_j, we can synthesize x_t according to the distribution $p(x_t \mid z_t, \omega_j)$. Thus, x_t is modelled as a "mixture model of actions" as follows

$$p(x_t) = \sum_{j=1}^{m} \int p(x_t \mid z_t, \omega_j)\, p(z_t \mid \omega_j)\, P(\omega_j)\, dz_t, \tag{3.1}$$

where $\omega_j,\ j = 1 \ldots m$ denotes the jth action. This equation describes a reduced dimensionality mixture model where the m mixture components are actions. It describes the probability of generating a frame given the action (to which it belongs) and its corresponding subspace representation. We estimate the parameters of these distributions from the frames of all the activities by inverting the generative process. Ghahramani and Hinton [39] describe an Expectation Maximization (EM) algorithm [23] to learn the parameters in an MFA model. EM alternates between inferring the expected values of hidden variables using observed data, keeping the parameters fixed, and estimating the parameters underlying the distributions of the variables using the inferred values. The following section provides details of the EM algorithm used to estimate the action representation in a low-dimensional manifold.

3.2.3 Learning to Summarize

We use the videos of all the activities, arranged as a sequence of frames, for training. In our case, the observed data corresponds to frames, the hidden variables to the low-dimensional representations of these frames and the actions to which these frames are associated. The two phases of the EM algorithm – E-Step (Inference) and M-Step (Learning) – are discussed below.

- **E-Step**: The current estimates of parameters are used to compute the *expected* values for various interactions of the subspace representation and the actions. In other words, we compute $E[\omega_j \mid x_t]$, $E[z_t \mid \omega_j, x_t]$ and $E[z_t z_t^T \mid \omega_j, x_t]$ for all frames t and actions ω_j. Figure 3.3 shows the equations used to compute these quantities. In this figure, $\mu_j,\ j = 1 \ldots m$ denotes the representative appearance for each of the actions while $\Lambda_j,\ j = 1 \ldots m$ denotes the various subspace bases for the actions. π denotes the mixing proportions of actions in the activity set while Ψ is a measure of noise present in the data. h_{tj} can be interpreted as the membership of frame t in action j – the higher the value of h_{tj}, the more likely that frame t contains a subject performing action j. In this manner, we *infer* the values of the subspace representations of the frames and the actions to which they belong to.

- **M-Step**: The statistics collected during the inference from all the training examples are used to obtain better estimates of the parameters. We solve a set of linear equations to find π_j, Λ_j, μ_j and Ψ. Interested reader may refer to [39] for more details on these equations. Each of the frames x_t is assigned to an action c_t according to

$$c_t = \arg\max_j h_{tj} \qquad j = 1 \ldots m. \tag{3.2}$$

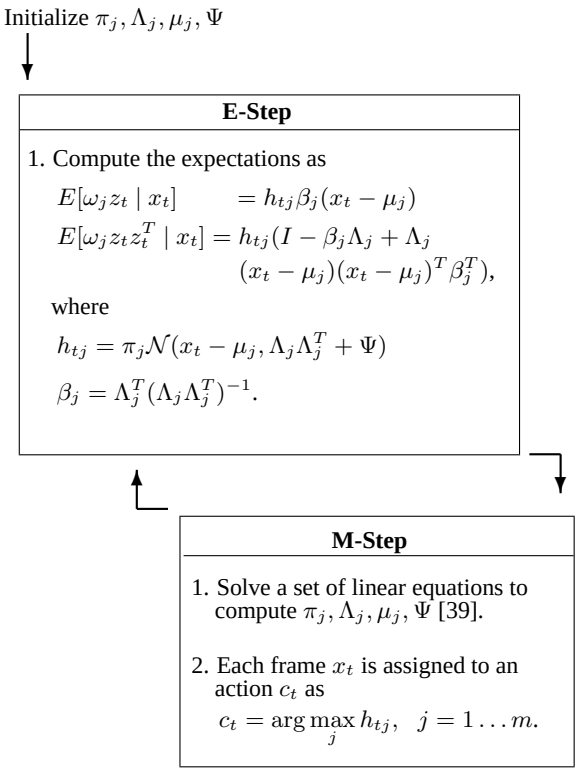


Figure 3.3 The two steps in the EM algorithm are executed in an iterative fashion till convergence. In the equations, μ_j denotes the mean appearance of each action and Λ_j denotes the corresponding subspace basis.

The two steps are executed iteratively till convergence. After convergence, we form the action transition matrix $T_k = [\tau_{pq}^k]$ for each activity A_k. The entries τ_{pq}^k of the matrix are given by

$$\tau_{pq}^k = \sum_{t=1}^{N-1} [c_t = p][c_{t+1} = q], \tag{3.3}$$

where $1 \leq p, q \leq m$. The action transitions for successive frames of the activity A_k are represented by the entries in the transition matrix T_k. This matrix encodes the temporal characteristics of the activity. Normalizing the entries gives the corresponding probability transition matrix P_k.

To sum up, we obtain the parameters of the model – $\{(\mu_j, \Lambda_j)_{j=1}^m, \pi, \Psi\}$, $\{P_k\}_{k=1}^K$ – at the end of the training phase. The model which now encapsulates the activity structure can be employed for the various tasks such as recognition, summarization, as described below

3.2.4 Likelihood Computation

Given the learnt representation, our task is to use it in a summarization framework. Let us consider a video of $\mathcal{V}$ frames which is to be summarized. To annotate each of these frames, we follow a maxi-

mum likelihood approach. First, we reduce the problem to a lower dimensional space using the factors learnt from the training data. We also compute the membership h_{tj} of every frame x_t in action j according to the equation given in Figure 3.3. x_t is then assigned a single action label by choosing the maximum $h_{tj},\ j = 1 \ldots m$ as in Equation 3.2. Let $c_1,\ c_2 \ldots c_{\mathcal{V}}$ denote the action assignments for the frames $x_1,\ x_2 \ldots x_{\mathcal{V}}$ respectively. A sequence probability (likelihood) S_k is computed using $S_k = \prod_{t=1}^{\mathcal{V}-1} P_k[c_t][c_{t+1}]$.

3.3 Summarization

The sequence probability can now be used to label the new video sequence. If the given video has only one activity, S_k denotes the likelihood of the video representing activity A_k. In this case, the unlabelled video is assigned to be the activity A_k^*, which maximizes $S_k, k = 1, 2, \ldots, K$.

Thus, the task of labelling the given video is trivial when it contains one activity. In a generic scenario, we need to summarize videos containing more than one activity. To handle this situation, we modify the likelihood computation accordingly. We consider a sliding window W of size w ($\ll \mathcal{V}$), which is moved over the entire video sequence one frame at a time. Many references to such sliding window approaches have been found in literature. In particular, Zelnik-Manor and Irani [118] use this approach to find the distance between two video sequences in a higher dimensional space. For each position i, $i = (w/2), \ldots, (\mathcal{V} - (w/2) + 1)$, of the sliding window, we compute a subsequence probability S_k^i using $S_k^i = \prod_{t=i}^{i+(w/2)-2} P_k[c_t][c_{t+1}]$. The frames between positions i and $(i + (w/2) - 2)$ are assigned to the activity A_k^*, following a maximum likelihood approach, if S_k^i is greater than a pre-determined threshold. Otherwise, the frames are labelled as *unknown*.

In our implementation, we compute the subsequence probability in an efficient manner. This is achieved by observing the correlation between any two consecutive sequence probabilities, say S_k^{i+1} and S_k^i. The subsequence probability is computed in an incremental fashion, as we slide over the frame sequence. S_k^{i+1} is obtained from S_k^i using

$$S_k^{i+1} = \frac{S_k^i P_k[c_{i+1}][c_{i+2}]}{P_k[c_i][c_{i+1}]}.$$

Once we have labelled the frames in the video sequence, we generate summaries in different ways (refer Figure 3.11). Grouping identically labelled contiguous frames segments the video temporally. These labelled chunks can be further classified as either known or unknown activity and are grouped based on their similarity. Finally, textual descriptions of the known activities are provided, which are a very useful abstraction of the video sequence.

This representation provides a hierarchical structure of the video. It also allows for a browsable description of the video contents. We then build an XML representation of the video sequence, an example of which is shown below. Efficient indexing and retrieval applications can be developed easily on this XML data.

```
<video>
  <segments number=5>
    <segment name=Squatting length=100>
    </segment>
    <segment name=Flapping length=121>
    </segment>
    <segment name=Jumping length=163>
    </segment>
    <segment name=Waving length=97>
    </segment>
    <segment name=Jumping length=116>
    </segment>
  </segments>
</video>
```

Standard Information Retrieval techniques [95] used in text retrieval systems are directly applicable on such a representation. A Table of Contents may be automatically built by parsing the XML data corresponding to the video. Furthermore, simple text querying on the video will provide user-specific video highlights. Thus, our approach for summarization and representation of videos provides most of the desirable characteristics of summarization systems discussed earlier in Section 3.1.

We now outline the entire summarization process. The given collection of videos is used to learn a representation of the constituent activities. Using this, the individual frames of a test video sequence are annotated, if the subsequence probability exceeds a certain threshold, else they are unlabelled (*i.e.*, they belong to an unknown activity). All these annotated frames are used to generate summaries in one of the ways described above.

3.4 Results

In this section, we illustrate the applicability of our model for analysis and annotation of different video sequences. We present results on human activity, Cricket and aerobics sequences. We also compare

our results to those reported by Zelnik-Manor and Irani [118] and provide a statistical validation of the proposed model.

Human activity videos: Summarization of human activity video sequences finds application in surveillance systems. It is needless to say that in such applications, browsing through a summarized video is much more efficient than going through each and every frame of the long video sequence. A typical scenario where an activity-based summarization tool may be used is that of identifying security breaches. If the system is trained on a set of acceptable activities, then any un-trained activity can trigger appropriate remedial steps such as sound an alarm, close the entry points, etc. In other words, a summarization system is a useful aid for humans to monitor unusual activities.

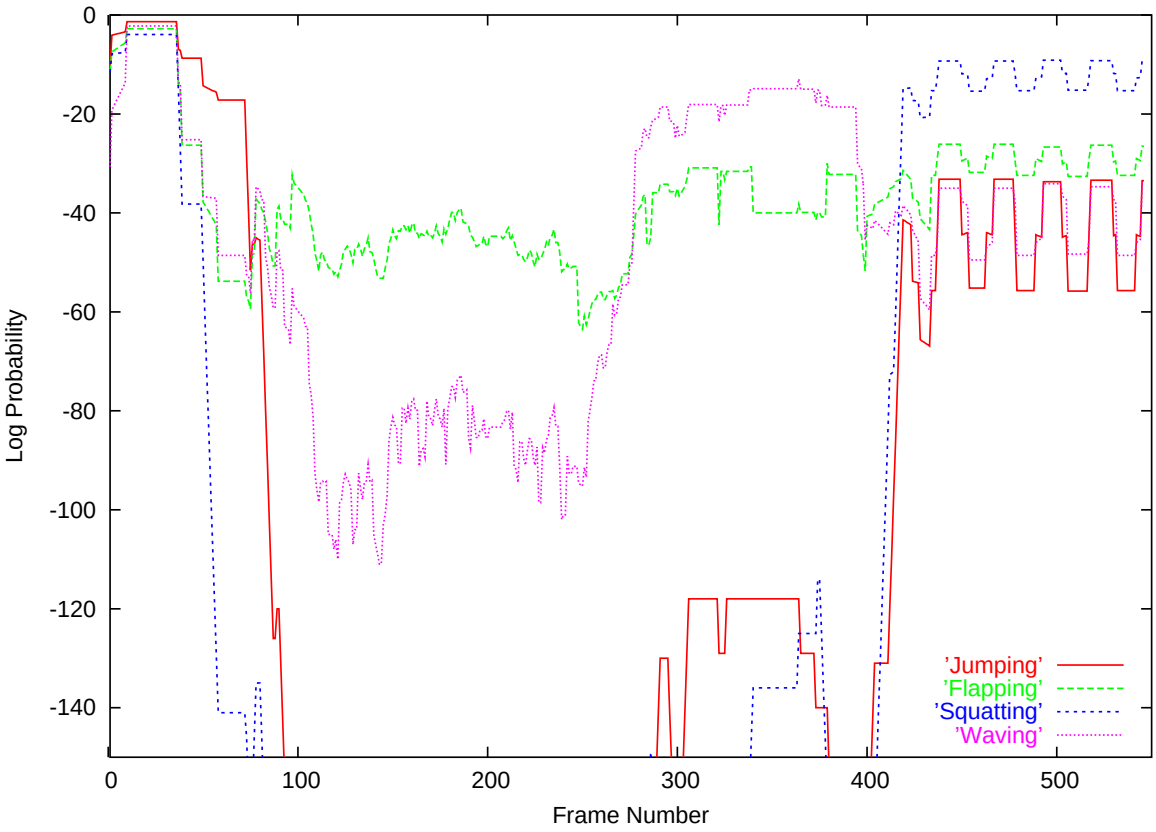


Figure 3.4 A graph showing the logarithm of sequence probability S_k^i for the frames of a video with known activities (Jumping $0-85$, Flapping $86-276$, Waving $277-390$, Squatting $391-546$). Each frame is annotated as the activity whose corresponding probability is maximum. Note that the model remains fairly accurate even during the activity transition phases. On an average, 96% of the frames are annotated correctly. The crests and troughs in the graph (frames $400-546$) clearly denote two actions (Sitting and Standing) of the activity Squatting which are performed by the subject 4 times.

For analysis of human activity videos, we considered video sequences of 7 activities – Flapping, Jumping, Squatting, Waving, Limping, Walking and Hopping. A few sample frames of some of these activi-

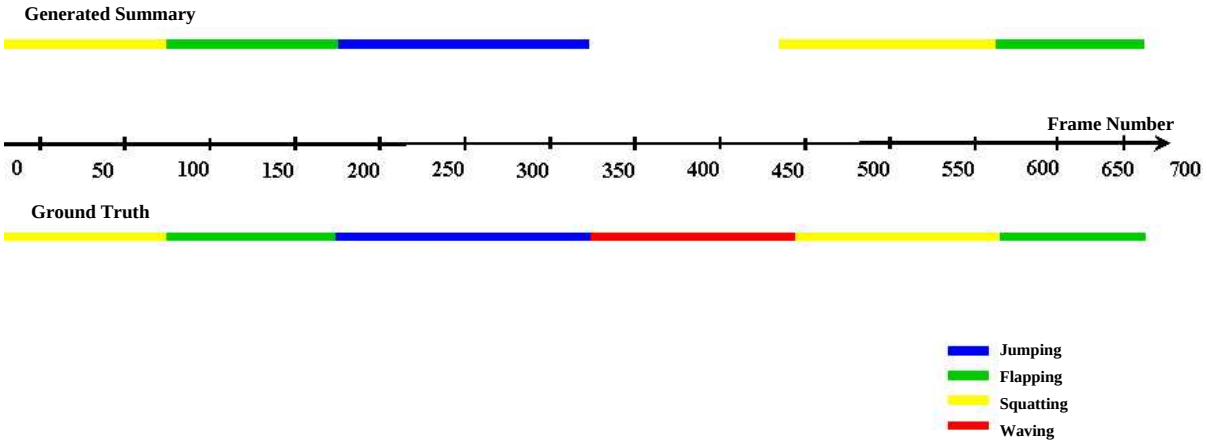


Figure 3.5 Summarization of a 670 frame long video sequence with Jumping (Blue), Flapping (Green), Squatting (Yellow) as the known activities and Waving (Red) as the unknown activity. The video is summarized based on the subsequence probability observed for each frame. The frames corresponding the the unknown activity are unlabelled due to their low probability values (represented by the blank region in the generated summary). 96% of the frames are identified correctly in this case.

ties are shown in Figure 3.2. We captured the videos at 24 fps using a Panasonic Digital Video Camera. The average duration of each activity is 5 seconds. Minimal pre-processing steps, such as normalizing the frames by centering the concerned subject, retaining only the visually significant information by subtracting the background, and thresholding were performed. The processed frames were then used to learn the low dimensional representation and the probability transition matrices for all the activities.

In the first experiment, videos comprising of the activities Flapping, Jumping, Squatting, Waving were considered. The test sequence comprised of these known activities (performed by a new subject). Using the learnt representations of the activities we computed the subsequence probability for all the four activities. A window size $w = 40$ was found appropriate for our experiments. This is based on the assumption that the activities (captured at 25 fps) in consideration are performed for atleast 1.6 seconds. It is to be noted that the window size is not a critical choice, if it is sufficiently small. The results of this experiment are shown as a probability plot against time in Figure 3.4. The x-axis denotes the frame number and the y-axis denotes the logarithm of the subsequence probability S_k^i for each of the activities. It can seen that almost all the frames are annotated correctly. The activity 'Jumping' shows a high correlation with all the other activities, hence the probabilities in the initial few frames corresponding to this activity are very similar. Nevertheless, the frames are identified correctly as Jumping, considering the maximum probability. The repetitive nature of the graph at various regions is due to the fact that each activity is performed more than once by the subject. Another interesting observation is that the activities are recognized after seeing only the initial few frames. This supports our claim that the model has effectively captured the activities and can be used in a real-time environment. By choosing the activity-set – on which the model is trained – randomly we found that our model summarizes 96% of

the frames accurately on average. Notice that the actions in the activity Squatting (frames 400 - 546) are very prominently visible in Figure 3.4. The crests and troughs visible in this region correspond to the two actions (Sitting and Standing) that constitute the activity Squatting.

Figure 3.6 Sample frames of 4 Cricket shots – Cover drive, Straight drive, Hook, Square cut. The subtle variations among these shots make the summarization task challenging.

The results of video summarization are illustrated in different ways. Figure 3.11 shows different summaries (arranged hierarchically) on a video with four known activities – Squatting, Flapping, Jumping and Waving. Temporal segmentation of the sequence provides a label for each frame, which is used to group them into similar activities. The recognition framework described in the previous section identifies each of these activities, if the model is trained on them. Such a naming is useful in building (text) query-based video retrieval systems.

In another experiment, we test the model on a video with a combination of known and unknown activities. Figure 3.5 shows the results on a video sequence with Jumping, Flapping, Squatting as the known, and Waving as the unknown activity. We learn the representation of the three known activities and test it on a video sequence with all the four activities. We observed that the unseen activity has low subsequence probabilities, unlike the activities on which the model is trained. Hence, all the frames with low probabilities are left unlabelled (or unidentified). Based on these labellings at each frame, we summarize the video as shown in Figure 3.5. We perform a leave-one-out test and find that 95.5% of the frames are identified correctly.

Cricket videos: There has been an increasing use of technology in modern day sports [2]. Analysis of video recordings of sports events to improve upon one's performance is a common phenomenon nowadays. For instance, in Cricket, statistics such as percentage of shots played on the on side, percentage of hook shots, etc. provide useful information about a player. Similar statistics are useful for other games

like Baseball, Football, Tennis. A summary of such video sequences comes in handy especially when the duration of the game is very long, as is the case with Cricket videos.

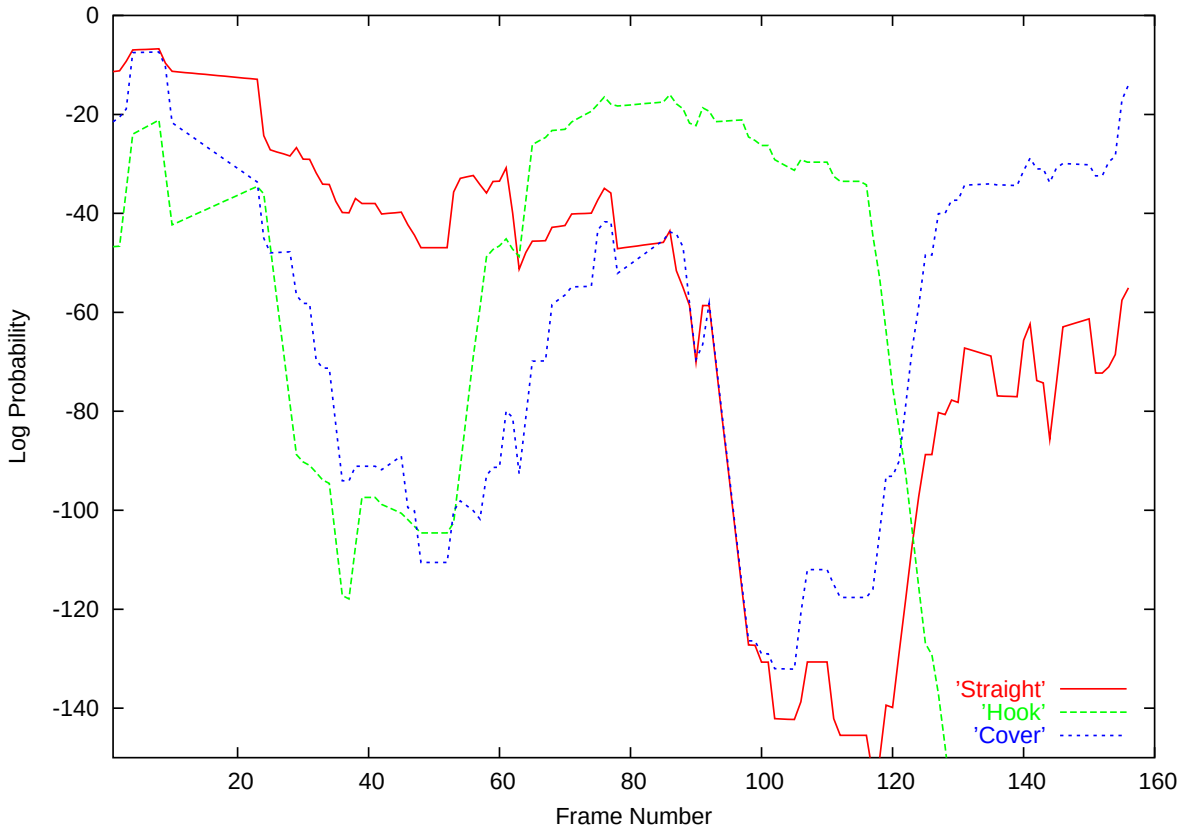


Figure 3.7 The logarithm of subsequence probability is plotted against frames of the activities – Straight drive ($0-55$ frames), Hook ($56-118$ frames), Cover drive ($125-156$ frames). Following a maximum likelihood approach, the summary generated is $0-60$: Straight drive, $61-120$: Hook, $121-156$: Cover drive. In all, only 11 frames of this video sequence were labelled incorrectly.

We provide the summarization results of videos consisting of batsmen executing various shots. The videos were recorded using a Konica Minolta Dimage Z2 camera at 30 fps and 320×240 resolution. The model was trained and tested on videos with 8 Cricket shots – Cover drive, Forward defence, Flick, Hook, Late cut, Square cut, Straight drive, Sweep. Figure 3.6 shows sample frames of some of these shots. Subtle differences among the shots (cover drive and straight drive, for instance) make the summarization task more challenging. A leave-one-out test on the data resulted in 95.5% of the frames being labelled correctly. Performance of the model on a video sequence with Cover drive, Hook and Straight drive is shown in Figure 3.7. In this case, only 11 frames (in 156 frame long video sequence) were labelled incorrectly.

Aerobics videos: Aerobics video sequences comprise of many complex activities. Summarization can prove to be useful in categorizing a large collection of such videos. As pointed out by Zelnik-Manor

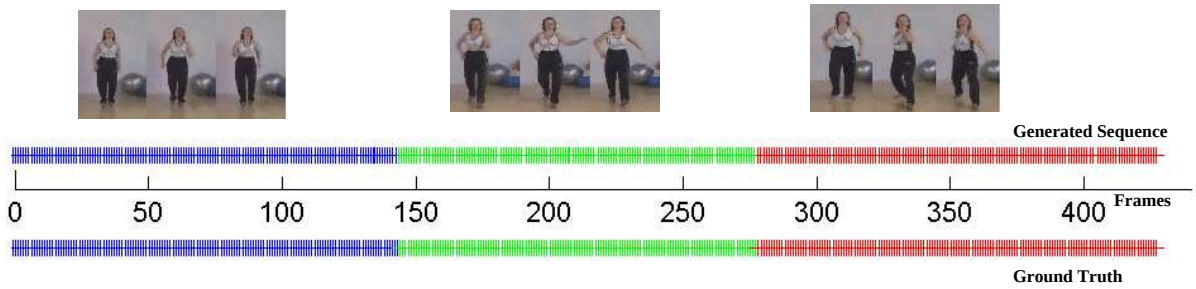


Figure 3.8 Summarization of an aerobics video sequence. The representative frames of the activities are shown in the top row. Ignoring the frames in which the sliding window falls on the boundary of two activities, 98% accuracy was observed.

and Irani [118], a desired application could be one where the user selects an interesting segment in the video and queries for an identical sequence in the collection. We trained our model on a set of aerobic activities [53] and summarized a 425 frame long video sequence. Results of this experiment are shown in Figure 3.8 along with the representative frames of the activities. As can be observed from the figure, almost all the frames are labelled correctly. If we ignore the frames in where the sliding window W consists of frames belonging to two activities (*i.e.*, the window falls on a boundary separating the two activities, which are difficult even for the human eye to identify,) the accuracy rate is nearly 98%.

Videos from [118]: Zelnik-Manor and Irani [118] proposed a statistical distance measure based on the video content. They use this measure for clustering activities (events) and thereby temporally segmenting long video sequences. As this is done without any prior knowledge of the types of events, they cannot label the frames automatically.

We used the video sequences available at [54] to test our model. Figures 3.9 shows the results obtained on the Punch-Kick-Duck video sequence. Our approach resulted in similar labellings when compared with those reported by Zelnik-Manor and Irani [118]. They are also mostly consistent with the ground truth, which was obtained by manual segmentation. Results on the summarization of Tennis video sequence are shown in Figure 3.10 along with representative frames of the three activities. The relatively low accuracy rate of 85% is attributed to mainly two reasons: (a) High degree of similarity between the activities Hop and Step, (b) High-speed nature of the game, which results in the activities being performed for a short duration. However, the results are marginally better in our approach compared to those reported by [118]. Thus, the results obtained following our approach are consistent, if not better, with those reported in literature. The main advantage is that our model affixes labels correctly after observing only the initial few frames ($5 - 10$ on average).

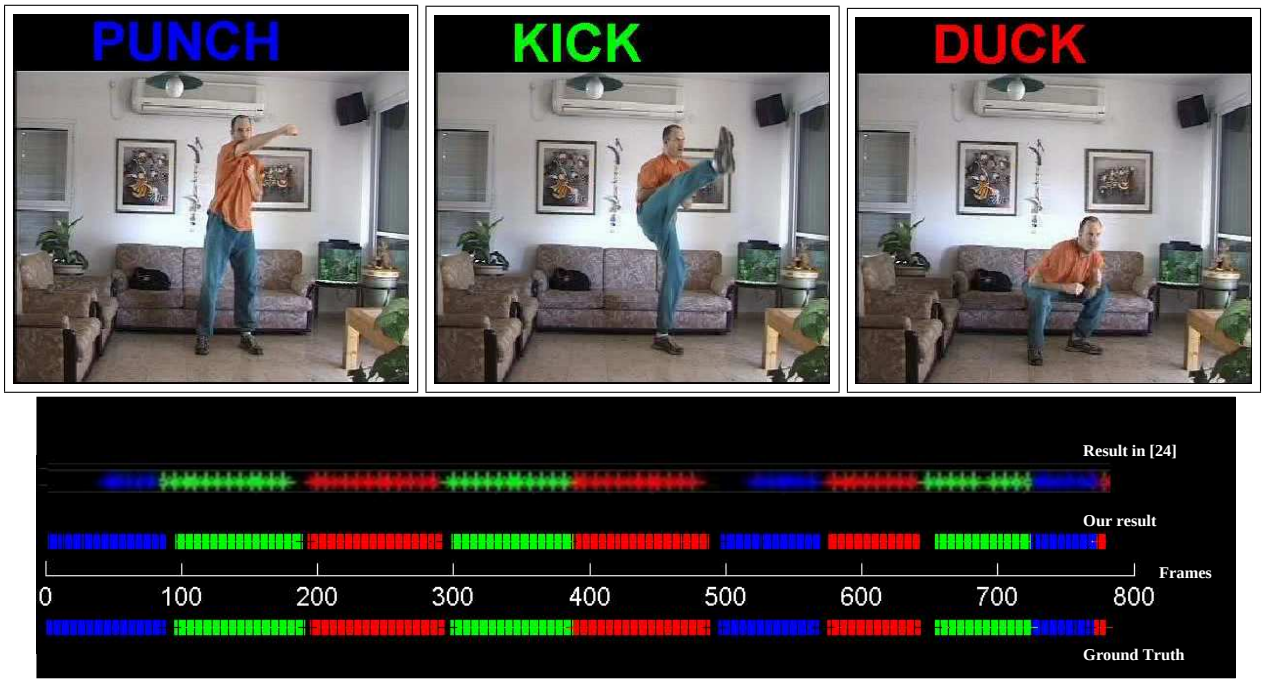


Figure 3.9 Summarization of Punch-Kick-Duck video sequence. Sample frames of the three activities (Punch - Blue, Kick - Green, Duck - Red) are also shown. Both the approaches label the frames quite accurately when compared to the ground truth, which was obtained by manually segmenting the sequence.

3.4.1 Validation of the Model

As mentioned before, the videos are represented in a low-dimensional manifold (refer Section 3.2.2). We computed a quantitative measure for the representation error. Using the low-dimensional representation z_t and the factor loading matrix Λ_j, we reconstruct the original frame, x_t. The average per pixel intensity difference between the original and the reconstructed frames of over 30 video sequences was found to be a very negligible 0.63%. Table 3.1 shows these errors for 7 different videos. We also performed a χ^2 test [118] to quantify the annotation performance of our approach on several videos. We computed the χ^2 distance between the annotation result obtained and the ground truth labelling that is available *a priori* for each frame. In general, the χ^2 distance between two sequences $a_1(i)$ and $a_2(i)$, $i = 1, 2, \ldots, N$, is given by

$$\chi^2 = \sum_i \frac{(a_2(i) - a_1(i))^2}{a_2(i) + a_1(i)}.$$

In this case an element $a_k(i)$ denotes the annotation label for the i th frame of the kth sequence. Say we have 4 activities in the video sequence, then $a_k(i) \in \{1, 2, 3, 4\}$. The results of this analysis on 5 video sequences with more than 500 frames each are illustrated in Table 3.2. On an average, χ^2 distance was 3.55. In comparison, the χ^2 distance between a hypothetical sequence, where 10% of the frames are annotated incorrectly, and the ground truth is about 18.72. The χ^2 distance between a hypothetical

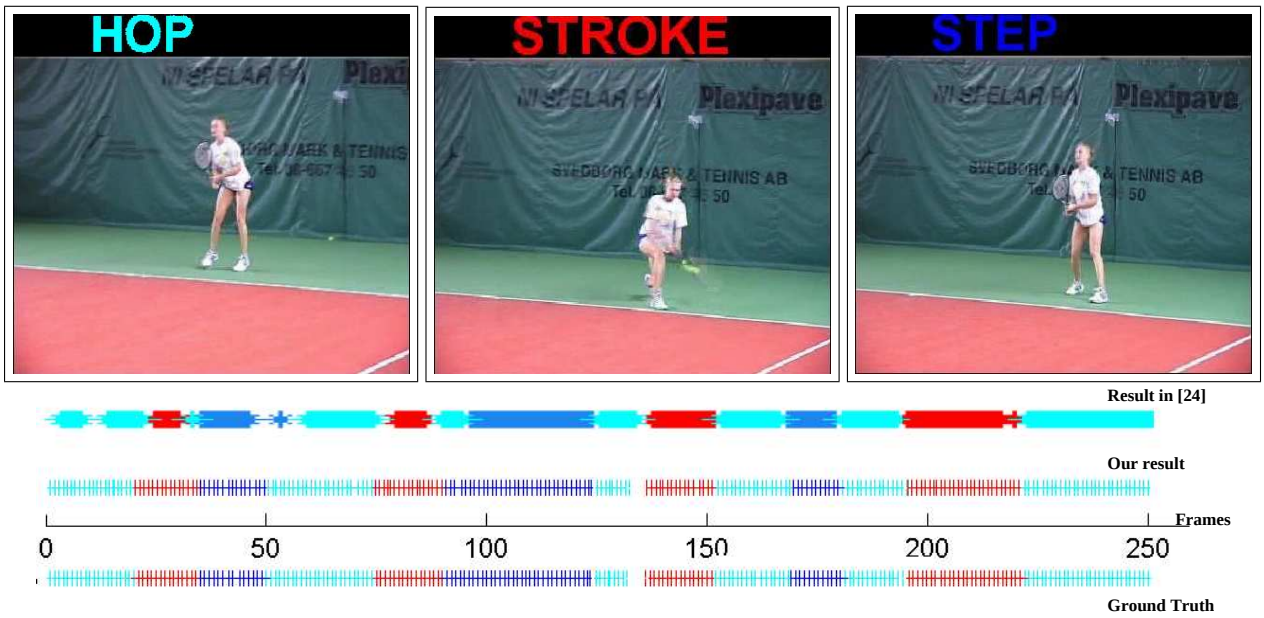


Figure 3.10 Summarization of a Tennis video sequence with three activities – Hop (Cyan), Stroke (Red), Step (Blue). It should be noted that the activities Hop and Step are fairly similar. Due to the high-speed nature of the game, each activity is performed for a short duration. This explains the relatively low accuracy rate (nearly 85%). However, they are marginally better compared to those reported by Zelnik-Manor and Irani [118].

sequence, where 20% of the frames are annotated incorrectly, and the ground truth is about 39.20. This shows that very few frames are annotated incorrectly using our approach.

3.5 Summary

In this chapter we presented an event-based approach to summarize video sequences. The summarization proceeds by temporally segmenting the video into events/activities and subsequently identifying them. We provide a hierarchical structure of the video sequences and build XML representations for them. Such high level abstractions of video have a large potential for application in browsing and retrieval systems. We demonstrated the performance of our approach on a variety of video sequences. One of the main advantages of this approach is that it analyzes the contents of the video without any explicit feature extraction. Furthermore, the transitions between activities are identified with a fairly large accuracy. As shown above, our model is applicable not only for the activities learnt, but also for unseen activities making them ideal for detecting *unusual* activities. A monitoring system built using our approach can be trained on a set of acceptable activities and any other activity that is observed can trigger appropriate remedial measures. Our approach is limited to providing video summary by "ex-

	Reconstruction Error
Video 1	0.0100
Video 2	0.0054
Video 3	0.0093
Video 4	0.0086
Video 5	0.0056
Video 6	0.0016
Video 7	0.0041

Table 3.1 A quantitative measure of the error in representing the activities in a low-dimensional space. The average per pixel intensity differences between the reconstructed and the original frames of 7 video sequences is shown here.

	χ^2 Distance
Video 1	3.67
Video 2	4.00
Video 3	1.33
Video 4	5.44
Video 5	3.33

Table 3.2 χ^2 distance between the ground truth annotations and the results obtained using our approach. As a comparison, the χ^2 distance between a hypothetical sequence, where 10% of the frames are annotated incorrectly, and the ground truth is about 18.72. This shows that very few frames are annotated incorrectly using our approach.

tracting" the essential content. The interesting problem of "synthesizing" the summary of the video is largely unaddressed.

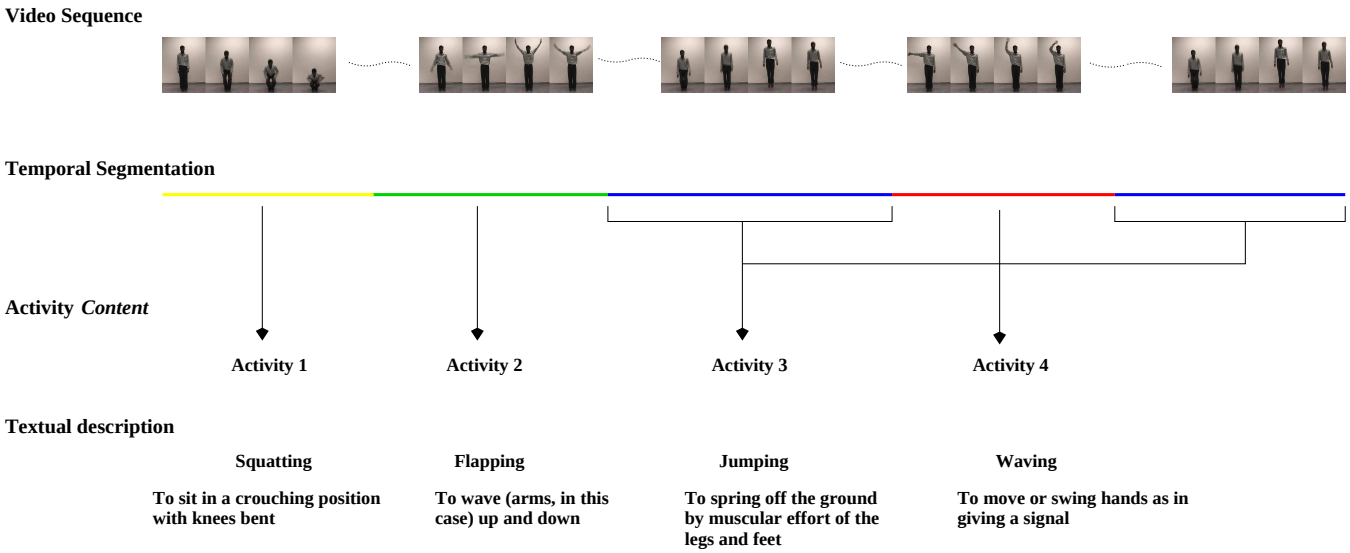


Figure 3.11 Representation of the summarization results. We present the summary of the given video sequence in three levels. In the first level, the sequence is temporally segmented. In the second, the content is identified and is grouped into similar units. And finally, if the activity is *known* (*i.e.*, the activity is seen during the training phase), we provide a textual description. This description may be used in building video retrieval systems.

Chapter 4

Feature Selection for Event Recognition

As is the case with most Pattern Recognition problems, feature selection – identification of appropriate features – plays an important role in event recognition [26, 35, 73, 80]. From a given set of features, one can select a subset of useful features [35] (e.g., Forward/backward subset selection procedure, Branch and Bound algorithms, etc.) or transform the feature space to a new basis to achieve better classification in a lower dimension [73] (e.g., PCA, LDA, ICA, etc.). These techniques are demonstrated to be effective for solving recognition problems when patterns are represented as vectors in a feature space [26]. Most of the event recognition techniques follow similar approaches. Unfortunately, the problem of feature selection for analyzing videos has seen few advancements over the years [3, 17, 37]. In this chapter we present novel feature selection schemes in the domain of video analysis, with applications to event recognition.

Due to the bulky nature of video data, we need a compact representation for efficient recognition applications. Often this is achieved using models like Linear Prediction, PCA, ARMA, Markov methods etc. [26, 103]. It has been argued that such modelling techniques, suited for efficient signal representation, (like PCA or as a matter of fact, schemes like LPC, HMM, etc.) need not be optimal for the classification task [10]. Using a discriminant-based approach along with these models provides better features for classification. However, it is not evident how discriminant analysis can be coupled with these modelling schemes to extract class-specific features for better recognition. Therefore, we need a discriminating mechanism which works at the compact modelling/representation level; and identifies the parts of the video sequence which can discriminate between two events belonging to different classes.

We begin by reviewing the state-of-the-art feature selection methods for event recognition. An exhaustive review of the classical approaches for event recognition can be found in [17, 37]. Many of these approaches employed 2D or 3D tracking to temporally isolate the object, which is performing the event, from the scene. Subsequent to tracking, the activity is recognized by extracting higher-order image features such as joint locations and inter-joint angles. Another significant direction is to extract motion features without resorting to tracking. Motion History and Motion Energy images [12], which represent

Figure 4.1 A few sample frames of events performed by humans: Squatting (top row), Flapping (bottom row). Note the presence of a common *action* – Standing – between these events. The initial few frames of the event Squatting represent the action standing while the other frames represent the action sitting. The action standing also occurs in the initial few frames of the activity Flapping. Thus, both these events share the common action Standing.

recency and spatial density of motion respectively, were used for modelling a dynamic activity. A related approach is to use low level motion features computed using an IIR filter for each frame [72]. All these schemes may be effective to *represent* the event, but not necessarily to *recognize* the event.

Kiran, *et al.* [93] proposed an approach for event recognition by identifying actions – the atomic spatiotemporal units inherent in events, which are subsequences extracted from the video sequences – and their sequencing information. They observed that most of the events (activities) performed by humans have common actions. For example, the event Squatting has two distinct actions: a standing action and a sitting action (refer Figure 4.1). Similarly, the event Flapping (flapping of hands) has standing and hands stretched out as the constituent actions. Clearly, these events share the common action 'standing'. The correlation that exists between these and other such events was profitably exploited in learning a compact representation. More details of this approach can be found in Appendix A. Although this approach led to efficient and effective representations, it may not be ideal for recognition applications. Usage of the relative importance of actions among events is limited to the representation phase. It is hardly evident in the recognition phase and does not provide an intuitive understanding of the important subsequences (actions) in the event.

We present an approach to overcome the limitations of these state-of-the-art techniques. To our knowledge, the proposed method is the first to use a discriminant-based feature extraction scheme for video sequences. We identify the parts of the videos (*i.e.,* subsequences or actions) which are more useful in discriminating between two dynamic events by analyzing their statistical characteristics. The individual actions are modelled and their discriminatory potential – the relative importance for distinguishing events – is then computed. A similarity/dissimilarity measure for the event is computed by combining the weights for individual actions, as shown in the example in Figure 4.2. We demonstrate the

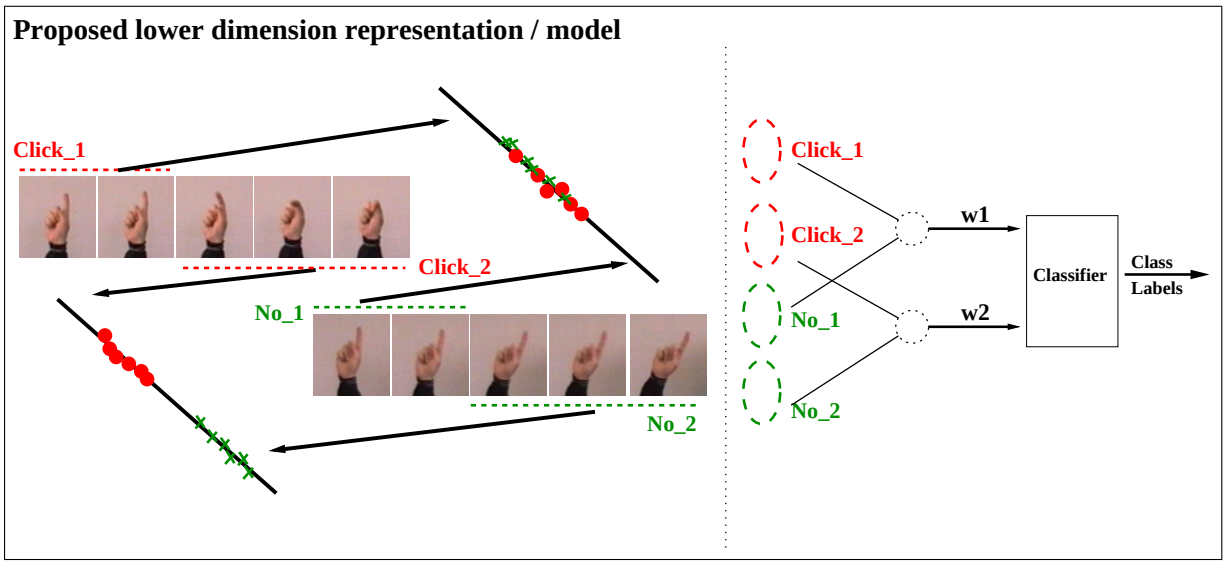


Figure 4.2 A few sample hand gesture frames showing two parts with different discriminatory potentials. In this case of events *Click* and *No*, the latter frames of the sequences are more useful in the classification task when compared to the former frames. The individual segments (two segments in this case) of the video sequences are modelled and their discriminatory potential is combined to compute a similarity/dissimilarity score.

application of these ideas on event videos and supplement our discussion with examples from Online Handwriting data [84].

The remainder of the chapter is organized as follows. In Section 4.1 we review the details of popular modelling schemes used for event representation along with a discussion on Discriminant Analysis techniques. The algorithm to obtain discriminant-based features for video sequences is given in Section 4.2. We show results on two categories of videos, namely hand gestures and human activities, along with a statistical analysis in Section 4.2.5. In Section 4.3 we present a novel scheme for integrating spatial (offline) and temporal (online) features in videos. We summarize the work in Section 4.4 and suggest directions for possible extensions. In the rest of this section we discuss an example that motivates our approach.

A Motivating Example

To better appreciate the need for discriminating features for event recognition, consider the example illustrated in Figure 4.2. Here, we show sample frames from two hand gesture [71] events: "*Click*", "*No*". Recognizing hand gestures has received a lot of attention in the recent past. It finds innumerable applications in HCI, Virtual Reality [74, 82]. One of the challenges in hand gesture recognition is the large similarity among the events (gestures). Given this scenario, we need to identify a feature space where the classes are compact (minimum inter-class variance) and distinct (maximum intra-class vari-

ance).

The two events shown in Figure 4.2 are described as follows. In the *Click* event the subject moves his index finger vertically up and down, as if clicking a mouse, while in the *No* event the subject moves his index finger sideways horizontally, as if "saying" no to something. The two events appear to posses similar properties at the beginning of the sequences (where the finger remains in an almost stationary state). As the complete video sequence begins to appear over time, the distinguishing characteristics unfold. In other words, the latter frames of the sequence are more useful for discriminating between the two events when compared to the former frames. Thus, the latter frames should contribute more towards the decision making process when compared to the former frames. As shown in Figure 4.2, our objective is to identify subsequences Click_2 and No_2 which map to a feature space wherein the events are clearly distinguishable. The other parts (Click_1 and No_1 in this example) owing to their similarity may not contribute much to the decision criteria. Popular pattern recognition approaches [26, 56] do not allow for such a scheme. They give equal importance to all parts of a video sequence during matching, which may not be the optimal, as in this case.

4.1 Preliminaries

We need efficient modelling schemes for building a compact representation of the events, by discarding the acceptable statistical variability. Efficient modelling also involves removing the redundancy inherent in video data. Spatial redundancies in individual frames are well exploited in image processing algorithms using statistical and structural methods [41, 55]. In a video, an additional temporal redundancy exists due to the smooth variation of the scene over time. Often, efficient video representations are achieved by methods like Hidden Markov Models (HMMs), Linear Prediction, Principal Component Analysis (PCA), etc. [4, 72]. Most of these modelling schemes exploit the inherent dynamism (*i.e.*, temporal relations) in videos.

HMM is a popular method for capturing the inherent dynamism in events, which are marked by smooth variations over time. Event representation, and subsequent recognition, using HMM typically proceeds as follows. The Markov model is fixed *apriori* based on the nature of the events being considered. Sun *et al.* [100] use four states to model events such as sitting, getting up from chair, and martial arts, etc. Each event (in a set of K events) is modelled by a corresponding HMM $\mathcal{H}_i$, where $i = 1, 2 \ldots K$. The parameters of the model $\mathcal{H} = \{\Xi, A, B, \pi\}$, where Ξ is the set of states, $A = \{a_{jk}\}$ is the transition probabilities matrix, $B = \{b_j\}$ is the observation symbol probability corresponding to state j and π is the initial state distribution, are estimated from the training sequences. The video sequence Φ is identified by computing the posterior $P(\mathcal{H}_i|\Phi)$, $\forall i$. The sequence is labelled as the event whose corresponding model $\mathcal{H}$ gives the highest posterior score.

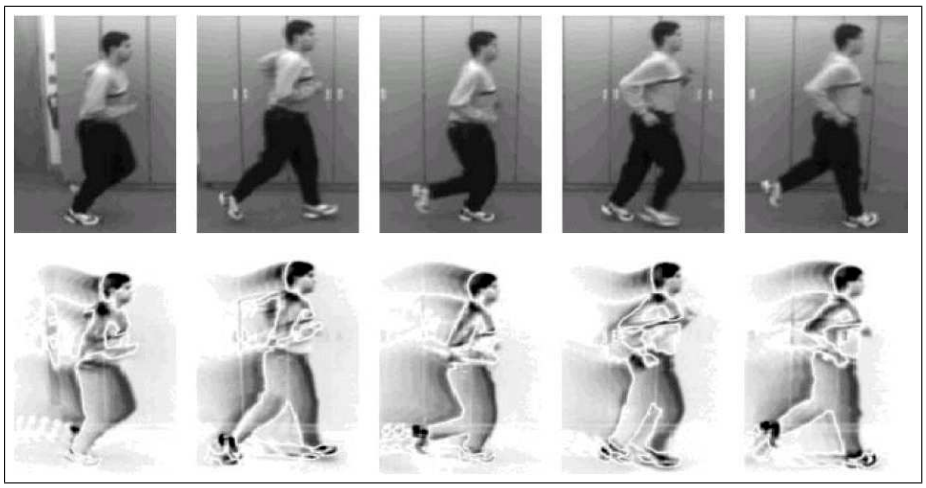


Figure 4.3 Sample frames of event *Running* (top row) modelled with Linear Prediction features (bottom row) (from Masoud and Papanikolopoulos [72], © 2003 IEEE).

Linear Prediction is another scheme for modelling the temporal relations in video sequences [72]. For a sequence of N video frames $\mathbf{X} = \{\mathbf{x}_i\},\ i = 1, 2, \ldots, N$, a pth order linear predictor relates a frame $\mathbf{x}_i$ to its previous p frames as $\hat{\mathbf{x}}_i = a_1\mathbf{x}_{i-1} + a_2\mathbf{x}_{i-2} + \ldots + a_p\mathbf{x}_{i-p}$, $i = (p+1), (p+2), \ldots, N$, where $\hat{\mathbf{x}}_i$ denotes the prediction of $\mathbf{x}_i$. The coefficient vector $\mathbf{a} = [a_1, a_2, \ldots, a_p]$ is typically estimated by minimizing the sum of squared errors $\sum_i ||\hat{\mathbf{x}}_i - \mathbf{x}_i||^2$. This vector captures the temporal correlation among the frames of $\mathbf{X}$. Masoud and Papanikolopoulos [72] used a linear prediction scheme to extract motion features from event videos. They computed feature images $F_i = |\hat{\mathbf{x}}_i - \mathbf{x}_i|$, $i = 1, 2, \ldots, N$, where $\hat{\mathbf{x}}_i = \alpha\mathbf{x}_{i-1} + (1-\alpha)\hat{\mathbf{x}}_{i-1}$. This is an Infinite Impulse Response (IIR) filter whose response is a measure of the motion in the image. Unlike the standard LPC methods, where the coefficients are estimated, the scalar α $(0 < \alpha < 1)$ is chosen based on the class of events. It signifies the importance of a frame over time. In the case of moving objects, this recursive filter produces a fading away effect. The features F_i implicitly encode the speed and the direction of motion. An example of a *Running* sequence and its corresponding feature sequence is shown in Figure 4.3. The motion trail is clearly visible in these feature images.

Due to the large size of video data, it is inefficient and impractical to model the events directly. Modelling the events in a low-dimensional space is the solution to this problem [72]. The goal is to find the optimal set of features that constitute each event. Principal Component Analysis (PCA) is one of the popular methods for achieving this. It is a linear model based on eigenvectors corresponding to the dominant eigenvalues [26]. Given normalized vectors x_i, we find the eigenvalues and the eigenvectors of $\sum_i x_i x_i^T$. Choosing the eigenvectors related to the top k eigenvalues, gives us the k most important features of the data x. In terms of mean squared error, PCA is considered to be an optimal linear dimensionality reduction method. It has been widely used for obtaining a low-dimension manifold of non-sequential data, such as images. Approaches dealing with application of PCA on video sequences

are limited to extracting a feature set based on offline patterns (images) [72, 112]. Furthermore, PCA is believed to be suitable for representing the data, unlike discriminant based approaches which are appropriate for classification problem [10].

These modelling schemes treat all parts of a video sequence uniformly. As mentioned earlier (refer Section 4), this may not ideal in most cases. To distinguish between the different parts of a sequence, we need to weigh them appropriately in computing the decision criterion. This is in the spirit of Discriminant Analysis and Statistical Pattern Recognition techniques.

4.1.1 Discriminant Analysis Techniques

Fisher Discriminant Analysis (FDA) is a popular feature extraction scheme for 2-class problems [31,73]. It finds an optimal direction φ along which the between class variance is maximized and the within class variance is minimized. The criterion function $J(.)$ is defined as

$$J(\varphi) = \frac{\varphi^T \mathbf{S_b} \varphi}{\varphi^T \mathbf{S_w} \varphi}, \tag{4.1}$$

where $\mathbf{S_w}$ and $\mathbf{S_b}$ are the within class and between class scatter matrices. The function $J(.)$ is maximized to compute optimal φ for discriminating between the patterns. It is shown that any vector φ which maximizes the Fisher criterion in Equation 4.1 satisfies $\mathbf{S_b}\varphi = \lambda \mathbf{S_w}\varphi$ for some constant λ [26]. This can be solved as an eigenvalue problem. Thus, the discriminant vector, φ is given by the eigenvector corresponding to the largest eigenvalue of $\mathbf{S_w}^{-1}\mathbf{S_b}$.

Ever since Fisher Discriminant Analysis was proposed [31], many extensions to it have evolved [68, 73, 75]. Multiple Discriminant Analysis adapts FDA to a multiclass scenario. This is achieved by generalizing the scatter matrices $\mathbf{S_w}$ and $\mathbf{S_b}$ to include multiclass information [73]. Lin *et al.* [68] recently proposed an incremental Linear Discriminant algorithm for multiple classes. They expressed the scatter matrices in terms of the means and covariances of the classes and updated them incrementally using the Sequential Karhunen-Loeve algorithm [67]. FDA has also been extended to discriminate non-linear data samples, using Kernel Discriminant Analysis methods [73, 75]. Non-linearity is handled by projecting the dataset into some linear feature space using the kernel trick [75].

Although these traditional discriminant analysis schemes provide efficient features, they are insufficient to handle video sequences directly.

4.2 Discriminative Features for Events

When recognizing events, there exist situations where some parts are more useful to distinguish than others. For example, when distinguishing two events – *"Click", "No"* – in hand gesture data [71] (see

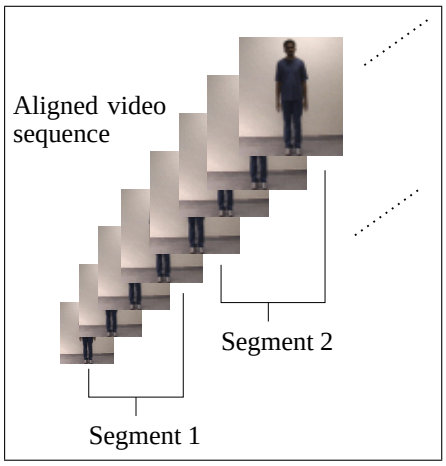


Figure 4.4 Temporal segmentation of a sample video sequence.

Figure 4.5), the common subevent (frames where the index finger appears in a vertical position) in them is less important. In this section we present the method for identifying discriminant features for event modelling and recognition, along with examples from online handwriting recognition [84][1]. We support our claims with experimentation and analysis on hand gesture and human activity videos.

Let us consider two video sequences $\mathbf{A}$ and $\mathbf{B}$ which belong to events (classes) $\mathcal{A}$ and $\mathcal{B}$ respectively. They represent a sequence of image frames where the corresponding event (like Click, No, etc.) is captured.

4.2.1 Temporal Segmentation

For the two video sequences $\mathbf{A}$ and $\mathbf{B}$, dissimilarity can be computed by comparing the sequences. If the sequences are of different lengths (say due to variation in frame rate of video capture or duration of the event), a normalization can be done by resampling. However, comparison of video data frame-by-frame is not valid since the event of interest is macro in nature and cannot be captured from one sample frame. Also, the identification of a simple model parameter (say using HMM or LPC) may not be valid for the entire sequence. Therefore, researchers have frequently decided to pick an appropriate intermediate subsequence for the representation [118]. The problem we address is identification of contribution of each of these subsequences for the global dissimilarity/discriminative information for the given video sequences.

Before segmenting the video sequences temporally, we need to align the video frames. In this work, we assume that the video sequences are already aligned with respect to a sample video (called the template video). It is to be noted that the alignment scheme and further processing steps are independent of the

[1]Thanks to Satya Lahari Putrevu, B.Tech. 2005, for conducting experiments on Online Handwriting data [5].

Figure 4.5 A few frames of hand gesture data showing two events – *"Click"* (left), *"No"* (right), and their common subevent.

choice of the template video. Moreover, if we are given that the data samples are obtained at a uniform rate (*i.e.,* the video sequences are captured at a uniform frame rate and all events have approximately similar duration), the sequences are already aligned, and can be directly segmented.

After alignment, we identify subsequences (or segments) of each video sequence by splitting it into a pre-fixed number parts. The number of segments is determined based on the set of events under consideration and their constituent actions. For our experiments on videos captured at nearly 25 fps, with about 150 frames each, we used 10 segments, with the assumption that all actions are performed in approximately 0.6 seconds. The temporal segmentation process is summarized using a sample video sequence in Figure 4.4.

At the end of the segmentation stage, we have s segments each of $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{A}^k$ and $\mathbf{B}^k, k = 1, 2, \ldots, s$, respectively.

Example: Let us look at Online Handwriting Recognition as an example. Each handwritten character/stroke is represented as a sequence of points, in the order it is written. Figure 4.7 shows sample handwritten strokes of the numerals 2 and 3. Just as in the case of video sequences, all parts of the stroke, which are referred to as substrokes, are not useful for the recognition task. To identify the relative importance of each substroke, we begin by identifying the segments. For this type data we use the well-known Dynamic Time Warping (DTW) techinque [84] to align and thus segment the sequences. DTW aligns a sequence of feature vectors using dynamic programming [26]. In this case, the feature set comprises of 2-dimensional points obtained over time. $D(p, q)$, the cost of aligning the sequences $\mathbf{A}$ and $\mathbf{B}$ (of lengths p and q respectively), is given by $D(i, j) = \min\{D(i-1, j-1), D(i, j-1), D(i-1, j)\}$ $+d(i, j)$, where $d(i, j)$ is the local cost in aligning the ith element of $\mathbf{A}$ and the jth element of $\mathbf{B}$. Backtracking along the minimum cost path obtained for $D(p, q)$ provides the alignment information. This alignment results in a correspondence between the two sequences. We extract a fixed number of segments from one of the sequences and choose the corresponding (aligned) subsequences from the others.

We will continue our discussion on modelling and recognizing handwritten strokes, as an example, in the remaining parts of this section.

4.2.2 Modelling the video segments

The segments $\mathbf{A}^k$ and $\mathbf{B}^k$ are modelled appropriately to capture their inherent properties. This is achieved by transforming the video segment features with a modelling function. We denote the model parameters of the jth sample by $\theta^k_{\mathcal{A}j}$ for the segment $\mathbf{A}^k$, and similarly $\theta^k_{\mathcal{B}j}$ for $\mathbf{B}^k$. The modelling function: (i) enforces a local continuity constraint within the segments, and (ii) preserves the ordering across segments. We use Motion History Images (MHI) proposed by Bobick and Davis [12] to model the video segments, which denote the actions that constitute events.

MHI represents *how* the motion is occuring in the event. In other words, it denotes the direction of motion. The history image of the jth sample for the segment $\mathbf{A}^k$ is denoted by $\theta^k_{\mathcal{A}j}$. From [12], the intensities at pixels in the history image at time instant t, $H_\tau(t)$, are a function of the temporal history of the motion of the corresponding points. It is defined as

$$H_\tau(t) = \begin{cases} \tau & \text{if I(t) = foreground} \\ \max(0, H_\tau(t-1) - 1) & \text{otherwise} \end{cases}$$

where τ is a pre-determined constant and I(t) = foreground denotes the set of all pixels belonging to the event-performing subject. For every segment k of the sequence $\mathbf{A}$, we compute the History Image feature for the last frame in that segment, *i.e.*, every segment has exactly one History Image feature. Motion History Image features of a few sample video segments are illustrated in Figure 4.6. The motion of these event segments is clearly visible.

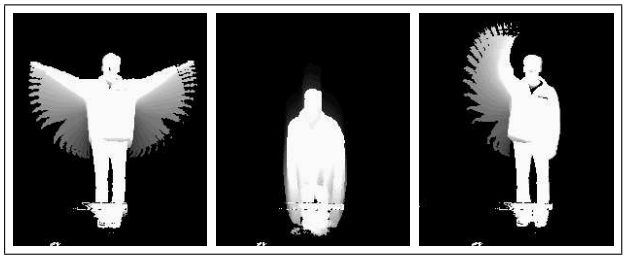


Figure 4.6 Motion History Image (MHI) features of a few sample video segments clearly illustrating the motion trails.

Example: Due to the simplicity of online handwritten data, many modelling schemes, such as LPC, HMM, DTW, etc., are viable for modelling the substrokes. For instance, using a trivial model – LPC of rth order – to capture the dynamic nature of the substrokes, we have the ith element of the jth sample, $\mathbf{A}^k_{ji} = [x^k_{ji}, y^k_{ji}]^T$ predicted using the r previous 2D observations. The scalars x^k_{ji}, y^k_{ji} denote the spatial location of the 2D data point. The prediction $\hat{\mathbf{A}}^k_{ji}$ is given by

$$\hat{\mathbf{A}}^k_{ji} = a^k_1 \mathbf{A}^k_{j(i-1)} + a^k_2 \mathbf{A}^k_{j(i-2)} + \ldots + a^k_r \mathbf{A}^k_{j(i-r)};$$

$i = (r+1), (r+2), \ldots, p^k_j$, where $\hat{\mathbf{A}}^k_{ji}$ denotes the prediction of $\mathbf{A}^k_{ji}$, and p^k_j is the length of segment $\mathbf{A}^k_j$. Most linear predictors estimate a^k_is by minimizing the sum of squared errors $\sum_i ||\hat{\mathbf{A}}^k_{ji} - \mathbf{A}^k_{ji}||^2$.

In other words, $\theta^k_{\mathcal{A}j} = \mathbf{a}^k = [a^k_1, a^k_2, \ldots, a^k_r]^T$. This scheme can be trivially extended to handle other features traditionally used in online handwriting recognition [84].

4.2.3 Discriminatory potential of segments

To identify the segments (subsequences) useful for the classification task, we find the weights $\varphi_k, k = 1, 2, \ldots, s$, for each segment such that they have optimal distinguishing characteristics along the direction of the vector φ. We obtain this vector using a Fisher-like analysis, *i.e.,* we minimize the within class scatter and maximize the between class scatter for the sequences. The scatter matrices are defined as

$$\begin{aligned} \mathbf{S_w} &= \sum_{i \in \{\mathcal{A},\mathcal{B}\}} \sum_{j=1}^{N_i} (\theta_{ij} - \bar{\theta}_i)(\theta_{ij} - \bar{\theta}_i)^T \\ \mathbf{S_b} &= (\bar{\theta}_{\mathcal{A}} - \bar{\theta}_{\mathcal{B}})(\bar{\theta}_{\mathcal{A}} - \bar{\theta}_{\mathcal{B}})^T, \end{aligned}$$

where the number of samples in class i is denoted by N_i, the symbols without the superscript k denote the sequence features with subsequences stacked as rows and the mean over the samples of a class i is given by $\bar{\theta}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \theta_{ij}$. Also, $(\theta_{ij} - \bar{\theta}_i)$ is the distance measure defined in the representation space. Here, the $s \times s$ matrices $\mathbf{S_w}$ and $\mathbf{S_b}$ capture the within class and between class scatters at the subsequence level. Each entry of $\mathbf{S_b} = \{b_{ij}\}$ represents the variance between subsequences $\mathbf{A}^i$ and $\mathbf{B}^j$ over the set of all samples. Maximizing the objective function in Equation 4.1 results in classes with large discriminating characteristics.

To sum up, the process of computing discriminant features (training phase) is outlined here.

1. Align all the sequences in the training set with respect to a template sequence, and segment them temporally to obtain s segments for all the video samples in the two classes $\mathcal{A}$ and $\mathcal{B}$.

2. Model the individual video segments using Motion History Images (MHI) to get the new set of features – $\{\theta^k_{\mathcal{A}j}, \theta^k_{\mathcal{B}j}\}^s_{k=1}$.

3. Compute the discriminant vector φ, whose elements represent the relative importance of each segment, by minimizing the objective function $J(.)$ according to Equation 4.1.

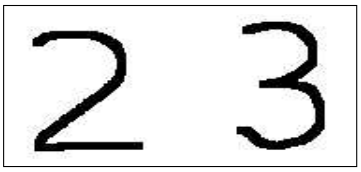


Figure 4.7 An Online Handwriting [84] example. The numerals 2 and 3 possess similar curvature properties at the beginning of the sequences. Their distinguishing characteristics unfold over time, as the complete numbers begin to appear.

Example: Let us revisit the online handwriting example discussed earlier. Consider the numeral pair $(2, 3)$ shown in Figure 4.7. The two numerals appear to possess similar curvature properties at the beginning of the sequences. As the complete numbers begin to appear, their distinguishing characteristics unfold over time. In other words, the tail portion of the numbers is more useful for distinguishing them. This vector is computed by minimizing the within-class variance and maximizing the between-class variance. This objective is achieved by minimizing the function $J(.)$ in Equation 4.1. When computing the $s \times s$ matrices $\mathbf{S_b}$ and $\mathbf{S_w}$, we build the θ_{ij} and $\bar{\theta}_i$ matrices. In this case, they are $s \times r$ matrices where each row consists of the r Linear Prediction coefficients.

In this example using two segments to model the strokes, φ was found to be $[0.410, 0.590]^T$. The elements of the discriminant vector support our claim that the head portion ($\varphi_1 = 0.410$) is less discriminatory compared to the tail portion ($\varphi_2 = 0.590$).

4.2.4 Recognition

Let $\mathbf{T}$ be the sequence we are interested in recognizing. It is labelled as class i^* according to

$$i^* = \arg \min_{i \in \{\mathcal{A}, \mathcal{B}\}} D(\mathbf{T}, i), \tag{4.2}$$

where D defines the cost of recognizing the sequence $\mathbf{T}$ as the sequence i. The matching cost $D(\mathbf{T}, \mathcal{A})$ is given by $D(\mathbf{T}, \mathcal{A}) = f(\varphi_1, \ldots, \varphi_s, \theta^1_{\mathbf{T}}, \ldots, \theta^s_{\mathbf{T}}, \theta^1_{\mathcal{A}}, \ldots, \theta^s_{\mathcal{A}})$. The function $f(.)$ models D as a combination of the subsequence (or action) level matching costs and the weights φ_k, which discriminate between the subsequences. Naturally, $f(.)$ depends on the modelling scheme used for the subsequences. Given the modelling scheme used in our approach, we define

$$f(.) = \sum_{k=1}^{s} \varphi_k d(\theta^k_{\mathcal{T}}, \theta^k_{\mathcal{A}}),$$

where $d(.)$ is the distance between the two Motion History Image feature vectors.

We now present a justification for using the elements of the Discriminant vector as weights in the decision criterion. Discriminant Analysis identifies an optimal direction ϕ along which the ratio of between-class scatter and within-class scatter is maximized. When the data points, say, $\theta^k_{\mathcal{T}}$ and $\theta^k_{\mathcal{A}}$ are projected onto this direction as $\phi^T\theta^k_{\mathcal{T}}$ and $\phi^T\theta^k_{\mathcal{A}}$ respectively, each element of ϕ acts as a weight for the corresponding dimension. In the lower dimension, the distance between two event segments $\theta^k_{\mathcal{T}}$ and $\theta^k_{\mathcal{A}}$ is expressed as a weighted linear combination of distances along each dimension, *i.e.*, $d(\phi^T\theta^k_{\mathcal{T}}, \phi^T\theta^k_{\mathcal{A}}) = \sum_k \phi_k d(\theta^k_{\mathcal{T}}, \theta^k_{\mathcal{A}})$. This relation holds for any metric distance $d(.)$.

Example: In the online handwriting case, where we modelled the data points using a Linear prediction scheme, we define the metric as the distance between the two prediction coefficient vectors. Recognition

of numerals 2 and 3 (Figure 4.7) modelled using a linear predictor of order 3, resulted in an average accuracy of 97.33% using our weighing method compared to 93.73% using equal weights – a 56.51% reduction in error. A leave-one-out approach was followed to compute this recognition accuracy. For a more detailed discussion on the experiments and an analysis of the results on online handwriting data, the reader may refer to [5].

4.2.5 Experiments and Results

In this section we present results on two classes of videos – hand gestures and human activities. We used recorded as well as publicly available videos [71] for testing the applicability of our model. We also present a statistical validation experiment.

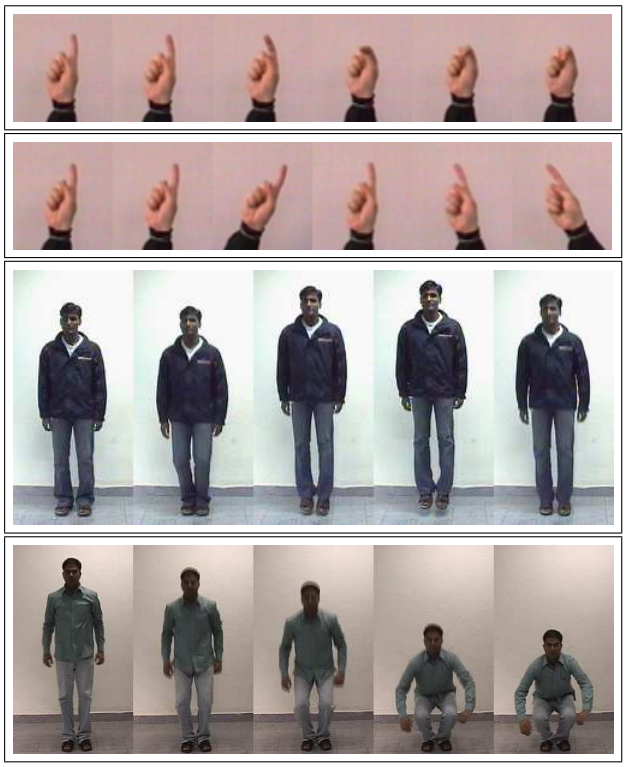


Figure 4.8 Sample frames showing four events. Hand gestures: Click, No (first two rows); Human activities: Jumping, Squatting (last two rows).

Hand gestures: Recognizing hand gestures has received a lot of attention in the recent past. It finds innumerable applications in HCI, Virtual Reality [82], wherein input to the computer can be regulated through various hand gestures, for instance controlling the visualization of a CAD model. One of the challenges in hand gesture recognition is the high similarity among the events (gestures). We test the applicability of our approach on this dataset. We used hand gestures videos from Marcel's Dynamic Hand Gesture database [71]. It consists of 15 video sequences for each of the 4 hand gestures, namely

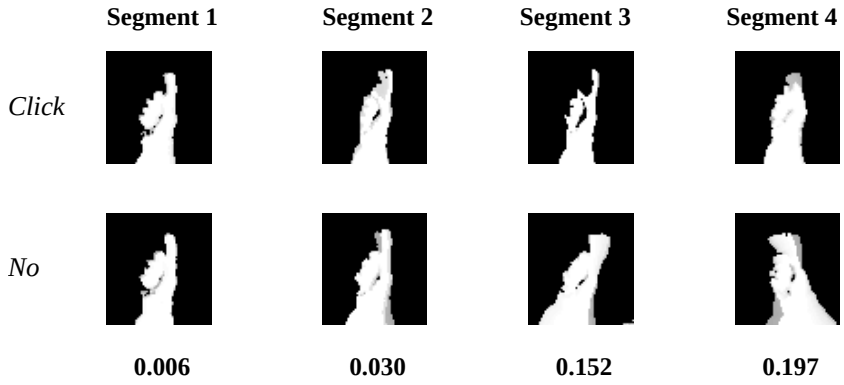


Figure 4.9 Motion History Image features computed for 4 segments of the events *Click*, *No*, and their corresponding discriminatory potential (shown in the last row). It can be observed that the first two segments have low discriminatory potential owing to their similarity. The last two segments are more useful for the classification task.

Click, No, StopGraspOk and Rotate. The data was divided into separate train and test sets. No resubmission error on the data set was observed. We present results on three of the possible pairs – Click vs No, StopGraspOk vs Rotate, Rotate vs Click – which have high degree of similarity. Sample frames of a couple of hand gestures are shown in Figure 4.8. Following is the experiment conducted on this data set.

1. We segment the aligned sequences temporally to obtain actions (video segments) for all the video samples in the two classes.

2. We model each video segment using MHI features.

3. We then compute the discriminant vector, which represents the relative importance of each segment according the method described in the Section 4.2.3.

4. Given a new video sequence to recognize, we follow the first two steps, then use the already estimated weights to compute the similarity score and label the video as discussed in the previous subsection.

The accuracy results on this data set are illustrated in Table 4.1. We compare our results to those generated by techniques which give equal importance to all parts of the sequence. Here, we observed an average improvement of 3.6% in the recognition accuracy. In Figure 4.9 we illustrate the Motion History Image features computed for 4 segments of *Click*, *No* events. Our claim that the latter frames of the sequence are more useful for the classification task is supported by the discriminatory potential of these frames.

Other activities: Recognition of events involving humans finds many applications in surveillance [4, 12, 72]. These events are marked by a considerable amount of overlap among them. We exploit this

Video Pairs	% Accuracy	
	Equal weights	Optimal weights
Click vs No	91	93
StopGraspOk vs Rotate	90	92
Rotate vs Click	87	92
Jumping vs Squatting	85	90
Limping vs Walking	87	91

Table 4.1 Recognition accuracy for over 60 video sequences. On an average a reduction of 30.29% was observed.

observation and apply our discriminant based feature selection scheme. For this experiment on human activities, we used videos of 20 human subjects performing 4 different activities, of average duration 6 seconds. These activities occur with the subject either stationary or indulging in locomotion. In the former category, we consider activities Jumping and Squatting, while in the latter category (involving locomotion), we consider Limping and Walking. The videos were captured with a Panasonic Digital Video Camera at 24 fps. The data set was divided into distinct train and test sets. Minimal preprocessing is done on the video sequences. In order to retain only the visually significant information, background subtraction and normalization is performed on all the frames. Motion compensation is performed to center the subject for activities where locomotion is involved. We then segment the event and model the individual parts using MHI features. The modelled segments are used to estimate the discriminatory potential of each segment. To recognize an unlabelled test event, the sequence is preprocessed as above and the similarity measure is computed with respect to the two learnt event representations. The test video is labelled as the event for which the weighted similarity measure is maximum (refer Section 4.2.4). The recognition accuracy results on these activities are presented in Table 4.1. On an average, our approach results in 32.05% reduction in error.

Statistical Analysis: We now present a statistical analysis of the proposed model. To quantify the performance of the model, we computed the within-event and between-event scatters before and after the feature space transformation on a set of “Click” and “No” video sequences. Optimality of the feature set is defined in terms of the compactness (low variance within the class) and the separability (high variance between classes) of the classes. Low within-event and high between-event scatters shown in Table 4.2, after transforming the features to a discriminant-based feature space, support our claim that the model identifies an optimal discriminating feature set.

Feature Space	Within-event scatter		Between-event scatter	
Standard	Class 1	5.025	Class 1 vs 2	6.174
	Class 2	4.619		
Discriminant-based	Class 1	3.907	Class 1 vs 2	15.958
	Class 2	2.794		

Table 4.2 Performance of the model in identifying an optimal discriminant-based feature set. Here we show the within-class and the between-class scatters for both the classes (*Click* and *No*) before and after the feature space transformation. The values were computed by segmenting the sequences into 3 parts. Low within-action and high between-action scatter values indicate that our approach identifies a feature space wherein the classes are compact and well-separated.

4.3 Online and Offline features for event recognition

In this section we present another scheme to select appropriate features for event recognition. As mentioned earlier, events are characterized by smooth temporal variations. These variations provide useful cues for recognizing events. The state-of-the-art event recognition schemes can be broadly classified into two categories: (a) the ones that encode temporal features as image(s) [12, 72] (offline feature based methods), and (b) the ones that explicitly encode the temporal variations into the model [93, 100] (online feature based methods). We believe that the latter type of methods, henceforth referred to online feature based methods, are more appropriate to handle a wider class of events. In particular, the model proposed by Kiran *et al.* [93] is very attractive for event recognition. It represents the temporal relations in the event sequences in a low-dimensional space. The model is based on the observation that most events, which are composed of actions (the homogeneous units), have a large amount of overlap amongst them. This overlap is evident in the form of common actions among various events (see Appendix A). They model the frames of events as a mixture model of actions and employ a probabilistic approach to learn the individual actions, and the compositional rules for the events in a low-dimensional manifold. This is achieved by using a Mixture of Factor Analyzers (MFA) model [39] combined with a probability transition matrix, which encodes the transitions among action clusters. To sum up, actions denote the offline features and the transitions between them denote the online features in the events. However, this model is restrictive since the combination of online and offline features, which is determined by the number of actions, is fixed *a priori* and hence may not be appropriate for all events.

We now present an integrated model which combines these two types – online and offline – of features. We use a mixture of models with varying combinations of online and offline features. This is achieved by changing the number of action clusters in each of the component models. A small number of clusters denotes low online and high offline feature content, while a large number of clusters signifies high online and low offline feature content. It is a hard problem to determine one combination of online and offline features that suits all the events. Therefore, the relative importance of each model component

(*i.e.*, each combination of online and offline features) for a particular event is learnt through examples. The individual components, with the extreme cases using either one of the features, are trained on the given set of events. The learnt representation is then used to classify new video sequences. The decision criterion is defined as a weighted combination of the individual components (as shown in the Figure 4.10).

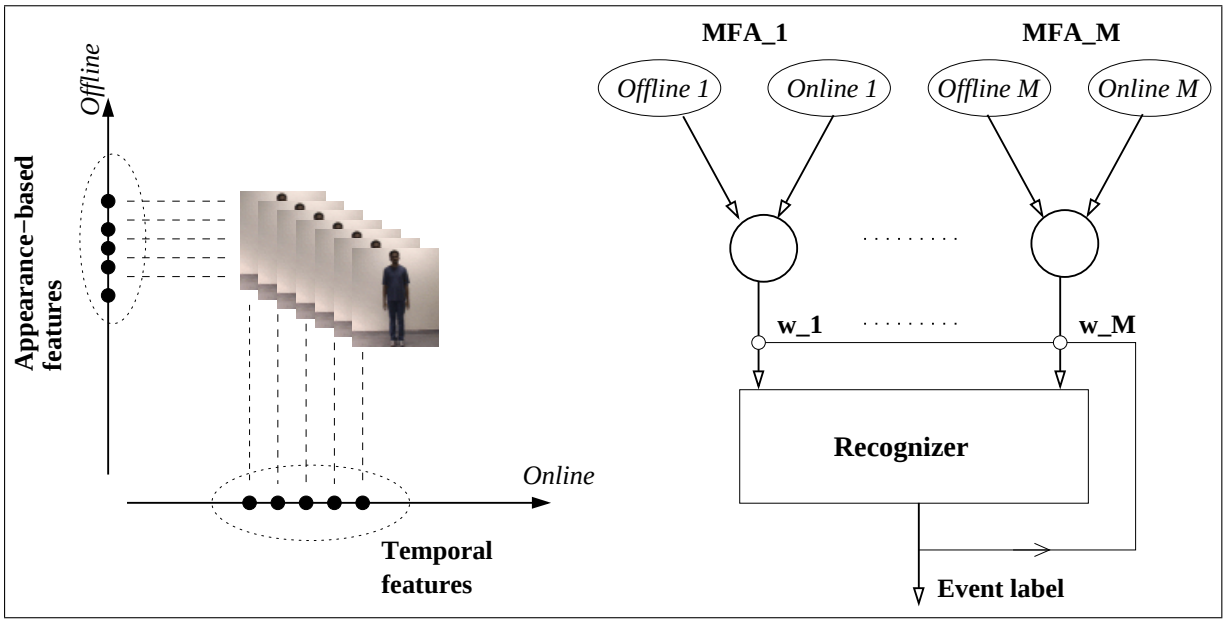


Figure 4.10 Video sequences consist of temporal (online) and appearance-based (offline) features, as shown on the left side. A summary of the proposed online and offline feature integration model is shown on the right side. We use a mixture of MFAs (MFA_1 . . . MFA_M) to have the model choose between offline (say, MFA_1), online (say, MFA_M), which are the two extreme cases, and a combination of both features (say, MFA_i) automatically. The contribution of each of these components in the decision making process is identified by its corresponding weight (w_i).

4.3.1 The Model

We begin by reviewing the basic model presented in [93]. It consists of an MFA coupled with a probability transition matrix. MFA is essentially a reduced dimension mixture of Gaussians. Let the total number of frames from examples of all the events be N and let x_t (of dimension d), $t = 1 \ldots N$ denote the tth frame. Subsequences of x_t form actions (the atomic units of an event). For instance, if we consider the event Squatting (which consists of two distinct actions – standing and sitting), the initial few frames represent the action standing and the other frames represent the action sitting (as in Figure 4.1). The subsequent frames of an action are highly correlated and therefore, for each x_t, a p ($\ll d$)-dimensional representation z_t exists. That is, x_t is modelled as $x_t = \Lambda_j z_t + u$ where Λ_j represents the transformation basis for jth action and u is the associated noise. Multiple such subsequences, occurring across different events, are used to learn Λ_j for each action and the corresponding low-dimensional rep-

resentation. Interested readers may refer to Appendix A for more details on MFA and the Expectation Maximization framework for learning its parameters.

Although MFA captures the actions effectively, it does not account for the temporality in events. This issue is addressed by modelling the dynamism in events as transitions across the learnt actions $\omega_1, \omega_2, \ldots \omega_m$. The transition probabilities are computed by observing z_ts across the various actions for each event. In the end, we obtain a compact representation of the events by automatically learning the m actions in a low-dimensional manifold and the sequencing information (which is embedded in the example frames). The structure of the ensemble of events is contained in the parameters of the actions and the probability transition matrix. More details of this approach can be found in [6, 93].

This model has a pre-fixed combination of online and offline features. We now propose an adaptive model which learns the appropriate composition of online and offline features based on the event in consideration. As mentioned earlier, the temporal (online) nature of the event is directly related to the number of action clusters in the MFA model. Hence, our method uses a set of MFAs with varying number of clusters; in some sense it is a mixture of MFAs. The two extreme cases in this framework are: modelling with (1) a single cluster for each event and, (2) a cluster for every frame of an event. In our model we vary the number of clusters between the two extremes. Our task is to define an approach which chooses the appropriate amount of offline and online features for recognizing events. In other words, we need to identify the relative importance of each model component for every event.

4.3.2 Learning the event representations

Theoretically, we can define a single cluster for each frame in the event video sequence. However, such a scheme is inefficient and impractical due to the possibly large number of transitions between these clusters. The maximum number of action clusters is typically decided by the nature of the data set, but is much lower than the total number of frames. Each MFA $\mathcal{M}_i, i = 1, 2, \ldots, M$ is trained with the frames of all the events using an Expectation Maximization (EM) algorithm. EM is a general method for finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given data set when the data has missing or unknown values [23]. In our case, the data corresponds to frames, the unknown values to the lower-dimensional representations of these frames and the actions to which these frames are associated. The procedure is explained in further detail below.

The videos of all the events of the subjects are represented as a sequence of frames and are used for training. The two phases of the EM algorithm – Inference and Learning – are executed sequentially and repeatedly till convergence. The E-step (Inference) proceeds by computing $E[\omega_j \mid x_t]$, $E[z_t \mid \omega_j, x_t]$ and $E[z_t z_t^T \mid \omega_j, x_t]$ for all frames t and actions ω_j, where z_t denotes the corresponding low-dimensional representation of the frame x_t. In the M-step (Learning) we compute the model parameters.

During the E-step we use the following equations

$$\begin{aligned} E[\omega_j z_t \mid x_t] &= h_{tj}\beta_j(x_t - \mu_j) \\ E[\omega_j z_t z_t^T \mid x_t] &= h_{tj}(I - \beta_j\Lambda_j + \Lambda_j(x_t - \mu_j)(x_t - \mu_j)^T\beta_j^T), \end{aligned}$$

where $h_{tj} = E[\omega_j \mid x_t] = \pi_j\mathcal{N}(x_t - \mu_j, \Lambda_j\Lambda_j^T + \Psi)$, $\beta_j = \Lambda_j^T(\Lambda_j\Lambda_j^T)^{-1}$. The parameters μ_j, Λ_j, $j = 1 \ldots m$, denote the mean and the corresponding subspace bases of the mixture j respectively. The mixing proportions of actions in the event are denoted by π. The noise in the data is modelled as Ψ. h_{tj} can be interpreted as a measure of the membership of x_t in class j.

After the EM algorithm converges, we form the action transition matrix $T_k = [\tau_{pq}^k]$ for each event E_k as follows.

$$\tau_{pq}^k = \sum_{t=1}^{N-1}[c_t = p][c_{t+1} = q]; \quad 1 \leq p, q \leq m \tag{4.3}$$

where c_t denotes the class label of the frame x_t and is given by $c_t = \arg\max_j h_{tj}; j = 1 \ldots m$. The entries in the transition matrix T_k represent the transitions of actions for successive frames of the event E_k. In other words, the matrix T_k encodes the temporal characteristics of the activity. Normalizing the entries gives the corresponding probability transition matrix P_k.

Thus, by the end of the MFA training phase, we obtain the parameters of the model – $\{(\mu_j, \Lambda_j)_{j=1}^m, \pi, \Psi\}$, $\{P_k\}_{k=1}^K$ for each model component.

4.3.3 Estimating the weights

Learning the representation also involves estimating the relevance of the individual components for a particular event. We identify this by optimizing an objective function $J(.)$ defined over the training set of N video sequences. The objective function $J(.)$, formulated along the lines of [57], is given by

$$J(\Gamma) = \sum_{j=1}^{N}\sum_{i=1}^{M}(\gamma_{ij}d_{ij})^2,$$

where $\Gamma \in \mathbb{R}^{MN}$ is a matrix $[\gamma_{ij}]$. γ_{ij} denotes the contribution (or the significance) of the ith MFA for the jth video sequence in the data set and d_{ij} is the distance metric signifying the cost of recognizing the jth sample with the ith MFA. We minimize this objective function over the space of γs. This is done by using Lagrange multipliers with the constraint $\sum_{i=1}^M \gamma_{ij} = 1$.

We observe that the weights for each video sequence are independent and hence the minimization can be done independently in each column. Thus, the Lagrangian is given by

$$\mathcal{J}(\lambda, \gamma_j) = \sum_{i=1}^{M}(\gamma_{ij}d_{ij})^2 - \lambda(\sum_{i=1}^{M}\gamma_{ij} - 1). \tag{4.4}$$

Differentiating Equation 4.4 with respect to γ_{pq}, we get

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \gamma_{pq}} &= 2\gamma_{pq}(d_{pq})^2 - \lambda = 0, \\ \implies \gamma_{pq} &= \lambda/2(d_{pq})^2. \end{aligned} \tag{4.5}$$

Using Equation 4.5 in the constraint $\sum_{r=1}^{M} \gamma_{rq} = 1$ gives

$$\lambda = 1 \Bigg/ \left(2 \sum_{r=1}^{M} (d_{rq})^2 \right).$$

Substituting λ into Equation 4.5 we get

$$\gamma_{pq} = 1 \Bigg/ (d_{pq})^2 \sum_{r=1}^{M} (d_{rq})^2 \ . \tag{4.6}$$

This equation provides a method for estimating the weights, given the distance metric d_{ij}. We define this metric in terms of the likelihood of the MFA $\mathcal{M}_i$ recognizing the correct event, which is the probability computed from the corresponding transition matrix. Other metrics based on HMM, SVM, NN, etc., can also be explored.

4.3.4 Recognition

Once the weights $[\gamma_{ij}]$ are identified for all the classes, we use them in the recognition framework. Given an un-trained video, we learn the low-dimensional representations using each of the MFAs, $\mathcal{M}_i, i = 1, 2, \ldots, M$. We then compute the maximum likelihood of the event being recognized as belonging to class j using all the MFAs. The decision criteria based on the weighted sum of posterior probabilities (for class j) is given by

$$p_j = \sum_{i=1}^{N} \gamma_{ij} p(j|data, \mathcal{M}_i).$$

The event is labelled as belonging to the class j^*, which maximizes the posterior probability according to $j^* = \arg\max_j p_j$.

4.3.5 Experiments and Results

The proposed framework was tested on two classes of events, namely hand gestures and human activities. We used hand gesture videos from Marcel's database [71]. For the experiment on human activities, we used videos of 20 human subjects performing 7 different activities for an average duration of 6 seconds. These activities occur with the subject either being stationary or indulging in locomotion [6]. In the former category, we consider activities Flapping, Jumping, Squatting and Waving, while in the latter category (involving locomotion), we consider Limping, Walking and Hopping. All the videos were captured with a Panasonic Digital Video Camera at 24 fps.

Events	% Accuracy	
	Single MFA	Mixture of MFAs
Hand gestures:		
Click	89	94
No	88	93
StopGraspOk	90	92
Rotate	86	90
Human Activities:		
Flapping	83	88
Jumping	80	86
Squatting	83	90
Waving	82	86
Limping	85	92
Walking	87	93
Hopping	84	90

Table 4.3 A comparison of recognition accuracy using a single MFA model (which has a fixed composition of online and offline features) and the proposed mixture of MFA model (which learns the composition of features). On an average, 35.35% reduction in error was observed. Sample frames of some of these events can be seen in Figure 4.8.

Minimal preprocessing is done on the video sequences. In order to retain the visually significant information, background subtraction and normalization is performed on all the frames. For the activities involving locomotion, the frames are motion compensated to center the subject performing the activity. Using a set of example videos as the training set, we learn the appropriate composition of online and offline features, and the parameters that describe them for all the events (refer Sections 4.3.2 and 4.3.3). To recognize an unlabelled test event, the frame sequence transitions are computed via the inference step of EM algorithm. This results in a set of sequence probabilities computed for each event. The test video is then labelled as the event whose corresponding weighted probability measure is maximum (refer Section 4.3.4). The recognition accuracy results obtained using the proposed model and an MFA model [6] are presented in Table 4.3. When compared to the single MFA model [6], we achieve 35.35% reduction in error on average.

4.4 Summary

In this chapter, we addressed the issue of feature selection for recognizing events. We highlight the importance of feature selection for *recognizing* rather than just *representing* events. We also demonstrated that a fixed feature selection scheme may not be appropriate for a wide class of events. The two schemes presented here are briefly discussed below. The first one is based on discriminant analysis of sequential data, in particular video sequences. This approach: (a) provides a mechanism to identify the video seg-

ments (actions) and their importance statistically, (b) is applicable for different modelling schemes, (c) is suitable for various domains such as video event, online handwriting, etc., (d) is straight-forward to implement as there is no need for parameter-tuning, and (e) can be extended to a multiclass scenario on the lines of Multiple Discriminant Analysis [26]. Also, using Kernel Discriminant Analysis extends the applicability of the approach to non-linear data.

The second feature selection scheme presented adapts based on the set of events being considered. It learns an optimal combination of online and offline features for every event from a set of examples. The composition of online and offline feature content is controlled by the number of mixtures in the model. Furthermore, the model captures the temporal relations that exist in events in a low-dimensional manifold. Many interesting avenues for further research are possible. Incorporating the discriminant based scheme into this framework could lead to interesting results. Also, the features used to learn this adaptive model are quite primitive. Many sophisticated techniques can be investigated to improve upon this aspect [56, 110].

Chapter 5

Learning to Describe Videos

Automated analysis of videos for surveillance applications has been one of the most important problems for Computer Vision researchers [8,47,59]. Most of these applications require detection and tracking of moving objects – which is a difficult fundamental problem to date [34,90,108]. Traditional approaches such as optical flow computation [47], temporal differencing [8,90], background subtraction (or elimination) [108] may not be generic enough to analyze videos captured under various conditions. Of late, researchers have found learning-based approaches to be more appropriate for video surveillance applications [59,98].

In this chapter, we present a technique to automatically analyze video sequences and generate a description of the contents. Given video sequences recorded for potentially long duration, we first detect the various objects captured by the camera. This involves removal of insignificant information such as the scene background. Features of each object are then extracted. Using these features, we find the most likely model and the parameters to describe the state of the object over time. The model and its parameters may change over time as the object undergoes different transformations.

Our main focus is on observing moving objects and estimating the motion model that is most likely to describe the object. Any moving object which comes to a halt and remains stationary for a considerable amount of duration is discarded as the background. In this aspect our work differs from that presented in [34], where a layered background is assumed. Moving objects which remain stationary form various layers in their work. Jojic and Frey [59] describe an approach to identify deformable sprites in video layers. Each object is described by a "flexible sprite", which can deform from frame to frame in a linear fashion, *i.e.*, for a flexible sprite $\mathbf{s}$ the transformation $\mathbf{T}$ between frames is given by $\mathbf{Ts}$. The method presented in this work, overcomes the linearity assumption. Another closely related, although supervised, approach is presented in [15]. Bregler [15] presents a probabilistic approach to recognize human dynamics by learning the model that represents the motion. However, the type of dynamic model is assumed to remain unchanged over the entire video sequence. Only the parameters of the model are allowed to vary. Furthermore, the approach seems to be have been tested only on human movements in

video and is limited to handling one "object" in the scene. In comparison, our approach identifies the models and their parameters of all the objects in the scene. We also allow for the model to change over time. For example, the motion of a person walking may be described by a linear model, and later when (s)he starts jogging the motion may be described better by a complex mixture model.

The remainder of the chapter is organized as follows. Section 5.1 presents a generative model for videos. The details of our approach for learning the contents of video sequences and an Expectation Maximization algorithm to learn the model parameters are discussed in Section 5.2. Section 5.3 presents details of the experiments conducted on synthetic and real video sequences. We tested our formulation with traffic video sequences available at [52]. Concluding remarks are provided in Section 5.4.

5.1 A Generative model for video

Let us begin by understanding the video generation process. Let $O_1, O_2, \ldots$ denote the objects in the scene undergoing different motions. Let the motion parameters of these objects be described by parametric models $\mathcal{M}_1, \mathcal{M}_2, \ldots$ respectively. Examples of $\mathcal{M}_i$ include Hidden Markov Model (HMM), Gaussian Mixture Model (GMM), etc. The objects, when undergoing motion, constitute the foreground $\mathcal{F}$ of the scene and the regions which remain stationary in the video form the background $\mathcal{B}$. Together, the background $\mathcal{B}$ and the foreground $\mathcal{F}$ describe the entire video. Thus the generative model for a generic video sequence $\mathcal{V}$ containing N distinct moving objects described by M models is given by

$$P(\mathcal{V}) = \sum_{k=1}^{M} \sum_{i=1}^{N} p(\mathcal{V}|O_i, \mathcal{M}_k) p(O_i|\mathcal{M}_k) p(\mathcal{M}_k),$$

where $p(O_i|\mathcal{M}_k)$ denotes the likelihood of the model $\mathcal{M}_k$ describing the object O_i and $p(\mathcal{M}_k)$ denotes the prior of the corresponding model. Our task is to invert the generative process and learn the parameters of the distributions mentioned above. We use the Expectation Maximization (EM) algorithm to perform this. EM is a general method of finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given data set when the data has missing or unknown values [23].

To make the estimation problem tractable, we assume that the maximum number of models that may exist in a video sequence is known *apriori*. Before using the EM algorithm to estimate the model parameters, it is necessary to remove the unwanted background in the scene. Background removal or subtraction, as it is popularly known, forms an important preprocessing step in identifying the moving objects in the scene. Many methods for background removal have been proposed in the past [37, 98, 107, 112, 121]. Most researchers have now abandoned non-adaptive methods of background removal since they do not account for constantly changing backgrounds. In these methods, errors in the background accumulate over time, making them effective only in highly-supervised and short-term applications where there are no significant changes in the scene [37]. A standard method of adaptive backgrounding removal is by

averaging the images over time and creating a background approximation [98]. While this approach is useful in situations where the background is visible a significant portion of the time, it is not robust to scenes with fast-changing backgrounds. Furthermore, it cannot handle multimodal backgrounds. An appropriate way of overcoming this limitation is by mathematically modelling the background process.

We follow the method described in [121], which is an extension of the model proposed by Stauffer and Grimson [98]. It follows a pixel-based approach wherein each pixel is classified as either background or foreground using a Bayesian decision criterion. The intensities of the pixels are modelled as a mixture of Gaussians. Each pixel is labelled based on whether the Gaussian distribution which represents it is considered part of the background model. The parameters of the Gaussians are updated over time to accommodate varying backgrounds. Also, moving objects which come to a halt and remain in the state of rest are incorporated into the background model automatically. A few sample frames from a traffic video sequence [52] and their corresponding extracted foreground frames are illustrated in Figure 5.2. To sum up, the background elimination process provides the spatial extents of all the objects in every frame. Once the objects are identified, we extract features from each of the objects. In this case we used the 2D coordinates of all the points that belong to objects as features.

5.2 Inference and Learning

After extracting features from the moving objects identified in the scene, we learn the model parameters that describe their motion. This is achieved by using the EM algorithm. The EM algorithm has two steps: E-step (Inference step) and M-step (Learning step). In the E-step, the model parameters are assumed to be correct and, probabilistic inference is used to find the values of the unobserved variables. In the M-step, the model parameters are estimated to increase the joint probability of the observations and the computed unobserved variables. These two steps are repeated until convergence.

Since the objects O_i are already identified, the effective objective function $\mathcal{Q}(.)$ is given by

$$\mathcal{Q}(.) = \sum_{k=1}^{M} \sum_{i=1}^{N} w_{ik} p(O_i|\mathcal{M}_k),$$

where w_{ik} denotes the relative importance of model $\mathcal{M}_k$ in describing the object O_i. In this case, the observed data corresponds to the features extracted from the video frames and the unobserved data corresponds to the model parameters and the weights w_{ik}. Our task is to estimate the unobserved data such that the expectation $E[\mathcal{Q}(.)]$ is maximized with respect to all the models and the objects. Note that the model parameters of one object are independent of another. Thus, we can infer the model parameters independently for each object. In other words, we optimize $E[Q_i(.)]$, where

$$Q_i(.) = \sum_{k=1}^{M} w_{ik} p(O_i|\mathcal{M}_k),$$

for each object $O_i, i = 1, 2, \ldots, N$. In general, for any function $f(x)$ that is linear in x, $E[f(x)] = f(E[x])$. Since $\sum_k$ is a linear function, we have

$$E[Q_i(.)] = \sum_{k=1}^{M} E[w_{ik} p(O_i|\mathcal{M}_k)].$$

For computing the expectation in the E-step, the above equation can also be written as

$$E[Q_i(.)] = \sum_{k=1}^{M} E[w_{ik}] p(O_i|\mathcal{M}_k), \tag{5.1}$$

because the model parameters are assumed to be constant. To compute the model parameters, which define the likelihoods $p(O_i|\mathcal{M}_k)$, in the M-step we maximize $E[Q_i(.)]$ with respect to the parameters of all the models. The exact form of the equations depends on the type and the number of models chosen *apriori*. They can be derived easily once this choice is made. If Φ denotes the estimates of the parameter set, then the two steps of the EM algorithm can be summarized as follows.

E-step: Compute the expectation $E[Q_i(.)]$, for all the objects using the Equation 5.1.

M-step: Replace the parameters Φ by maximizing $\{E[Q_i(.)]\}_i$ over the entire set of possible parameters.

The convergence criterion is determined by the computed likelihoods $p(O_i|\mathcal{M}_k)$. After convergence we get the most estimate for the parameters Φ and the weights w_{ik}. EM will converge to a global maximum likelihood estimate, if the likelihood function has a single maximum [76]. As an example, we now discuss the EM solution for estimating the means of k Gaussian distributions.

Example: Consider a situation wherein the data has been generated by a probability distribution that is a mixture of k distinct Gaussian distributions. Each instance x_i is generated according to one of the distributions selected at random from the set of k Gaussians. The process is repeated to generate all the data samples. For this discussion, we assume that all the distributions are equally likely and have an equal (known) variance, σ^2. Our objective is to estimate the means $\mu_1, \mu_2, \ldots, \mu_k$ of the k distributions. If all the samples are generated from a single distribution, then this reduces to a maximum likelihood estimation problem. However, in our case the problem involves a mixture of Gaussians and so we cannot observe which samples were generated by a particular distribution. In other words, this is a prototypical example of a problem involving estimation of hidden variables.

The observed data corresponds to the samples $x_i, i = 1 \ldots n$. For each sample x_i there exists a corresponding k-dimensional random variable $z_i = [z_{ij}], j = 1 \ldots k$, where z_{ij} denotes the membership of the sample x_i in the jth distribution. In particular, z_{ij} has the value 1 if x_i was created by the jth Gaussian and 0 otherwise. The random variable z_i corresponds to the hidden data. As discussed above,

the k-means problem is to estimate the parameters $\mu = \langle \mu_1 \ldots \mu_k \rangle$. As the first step, we compute the likelihood of the complete data $Y = \langle y_1, \ldots, y_n \rangle$ *i.e.*, observed and hidden data, where $y_i = \langle x_i, z_i \rangle$, as

$$p(x_i, z_i|\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(x_i-\mu_j)^2}.$$

Thus the log-likelihood of the complete data can be written as

$$\begin{aligned} \ln p(Y|\mu) &= \ln \prod_{i=1}^{n} p(x_i, z_i|\mu) \\ &= \sum_{i=1}^{n} \left(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{j=1}^{k} z_{ij}(x_i - \mu_j)^2 \right). \end{aligned}$$

We then take the expected value of this likelihood over the distribution governing the hidden components z_i of the data Y as follows.

$$\begin{aligned} E[\ln p(Y|\mu)] &= E\left[\sum_{i=1}^{n} \left(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{j=1}^{k} z_{ij}(x_i - \mu_j)^2 \right) \right] \\ &= \sum_{i=1}^{n} \left(ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{j=1}^{k} E[z_{ij}](x_i - \mu_j)^2 \right), \end{aligned}$$

since $\sum_i$ and $\ln$ are both linear functions[1]. Noting that $E[z_{ij}]$ is just the probability that x_i was generated by the jth Gaussian distribution [2], we have

$$E[z_{ij}] = \frac{e^{-\frac{1}{2\sigma^2}(x_i-\mu_j)^2}}{\sum_{p=1}^{k} e^{-\frac{1}{2\sigma^2}(x_i-\mu_p)^2}}.$$

The definition of the expectation $E[\ln p(Y|\mu)]$ forms the E-step in the EM algorithm. The second step (maximization or the M-step) defines the expressions for computing new estimates of the parameters $\mu_1 \ldots \mu_k$, that maximize the function $E[\ln p(Y|\mu)]$. Maximizing $E[\ln p(Y|\mu)]$ is equivalent to minimizing $\sum_{i=1}^{n}\sum_{j=1}^{k} E[z_{ij}](x_i - \mu_j)^2$, *i.e.*,

$$\begin{aligned} \mu_j &= \arg\max_{\mu_j} E[\ln p(Y|\mu)] \\ &= \arg\max_{\mu_j} \sum_{i=1}^{n} \left(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{j=1}^{k} E[z_{ij}](x_i - \mu_j)^2 \right) \\ &= \arg\min_{\mu_\mathbf{j}} \sum_{i=1}^{n}\sum_{j=1}^{k} E[z_{ij}](x_i - \mu_j)^2. \end{aligned}$$

[1] As mentioned before, for any linear function $f(z)$, $E[f(z)] = f(E[z])$.

[2] z_{ij} is a binary random variable and $E[z_{ij}] = (z_{ij} = 1)p(z_{ij} = 1)$.

The above minimization, done for each μ_j, provides the estimates of the means of the k Gaussian distributions as

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} E[z_{ij}]x_i. \tag{5.2}$$

5.3 Experiments and Results

In this section we discuss the experiments conducted to support our claims. We present results on synthetic as well as real traffic video sequences [52]. Given the videos, we identify the objects in them, extract features corresponding to each of the objects and learn the parameters which are most likely to describe their motion.

5.3.1 Toy Problem

Type of Model	Ground Truth	Estimated value
Gaussian	$\mu = \begin{bmatrix} 30 \\ 25 \end{bmatrix}$	$\mu = \begin{bmatrix} 30.0059 \\ 25.0235 \end{bmatrix}$
	$\sigma^2 = \begin{bmatrix} 5 & 0 \\ 0 & 7 \end{bmatrix}$	$\sigma^2 = \begin{bmatrix} 5.0377 & 0 \\ 0 & 7.0513 \end{bmatrix}$
Linear Predictor	$\begin{bmatrix} 0.7 \\ 0.5 \\ 0.2 \end{bmatrix}$	$\begin{bmatrix} 0.6907 \\ 0.5010 \\ 0.2049 \end{bmatrix}$

Table 5.1 A comparison of ground truth and estimated model parameters from the 2 objects (refer Section 5.3.1). The feature points from Object 1 follow a Gaussian distribution and those from Object 2 follow a Linear Predictive model.

We wish to test our formulation for estimating the model parameters on synthetic data. The advantages of using such data are: (1) ground truth is available for comparison, and (2) the additional preprocessing steps, such as background removal, feature extraction, etc., are eliminated. The toy problem involves estimating the parameters that are likely to describe a certain number of objects, whose features are generated according to pre-determined distributions. The estimated parameters are then compared to the ground truth data that is available. In the rest of this discussion we assume two objects ($N = 2$) in the scene and two possible model types ($M = 2$), namely Gaussian distribution and Linear Predictive models. For our experiments we used a Gaussian distribution with mean $\begin{bmatrix} 30 \\ 25 \end{bmatrix}$, variance $\begin{bmatrix} 5 & 0 \\ 0 & 7 \end{bmatrix}$ and a third order linear predictor with coefficients $\begin{bmatrix} 0.7 & 0.5 & 0.2 \end{bmatrix}$. At every time instant t, we generate the feature points of each object according to the known distributions. We repeat this process for 100 frames and generate the feature points (about 50 2D coordinates in this case) of the 2 objects in the scene. Without loss of generality, we generate the feature points from object O_1 such that they follow

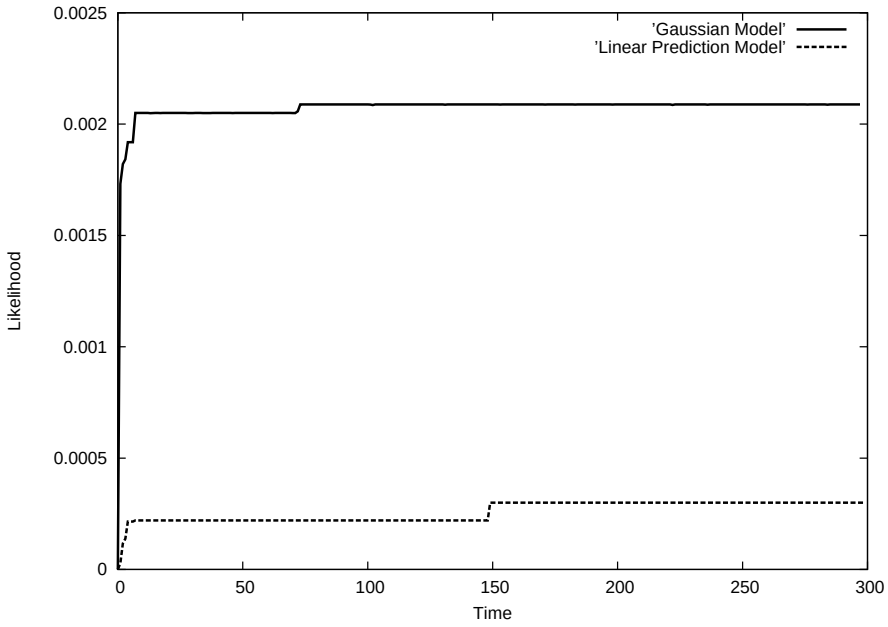


Figure 5.1 Graph showing the likelihood of the parameters of the two models describing the object O_1, whose features were generated according to a Gaussian distribution. The likelihood corresponding to the Gaussian (with the estimated parameters shown in Table 5.1) is greater at all time instants.

the Gaussian distribution and those from object O_2 follow the linear predictive model defined previously.

We begin our estimation process by initializing the unknown parameters of the two models. We assume both the models are equally likely and initialize the two weights w_{i1}, w_{i2} to 0.5. The feature points of the first frame are used to estimate the parameters through the EM algorithm described in the previous section. After the EM algorithm converges, the new parameters and the weights computed are used as initial estimates when processing the second frame. This method is followed for the entire collection of 100 frames. Table 5.1 shows a comparison of the ground truth and the estimated model parameters for the two objects. It can observed that the error in estimation is almost negligible. The learning characteristics of the algorithm can be observed from the graph illustrated in Figure 5.1. It shows the likelihood of the parameters of the two models describing the object O_1. Since this object's features follow a Gaussian distribution, the likelihood corresponding to this model is greater at all time instants when compared to likelihood computed using the linear prediction model. Furthermore, the likelihood increases gradually since the estimates of the model parameters improve over time.

5.3.2 Analysis of Real Videos

In another experiment we compute the model parameters of objects present in traffic surveillance videos [52]. Automatic analysis of such videos finds innumerable applications, such as detecting traffic congestion, unusual activity detection, etc. [62]. A few sample frames from the video are shown in

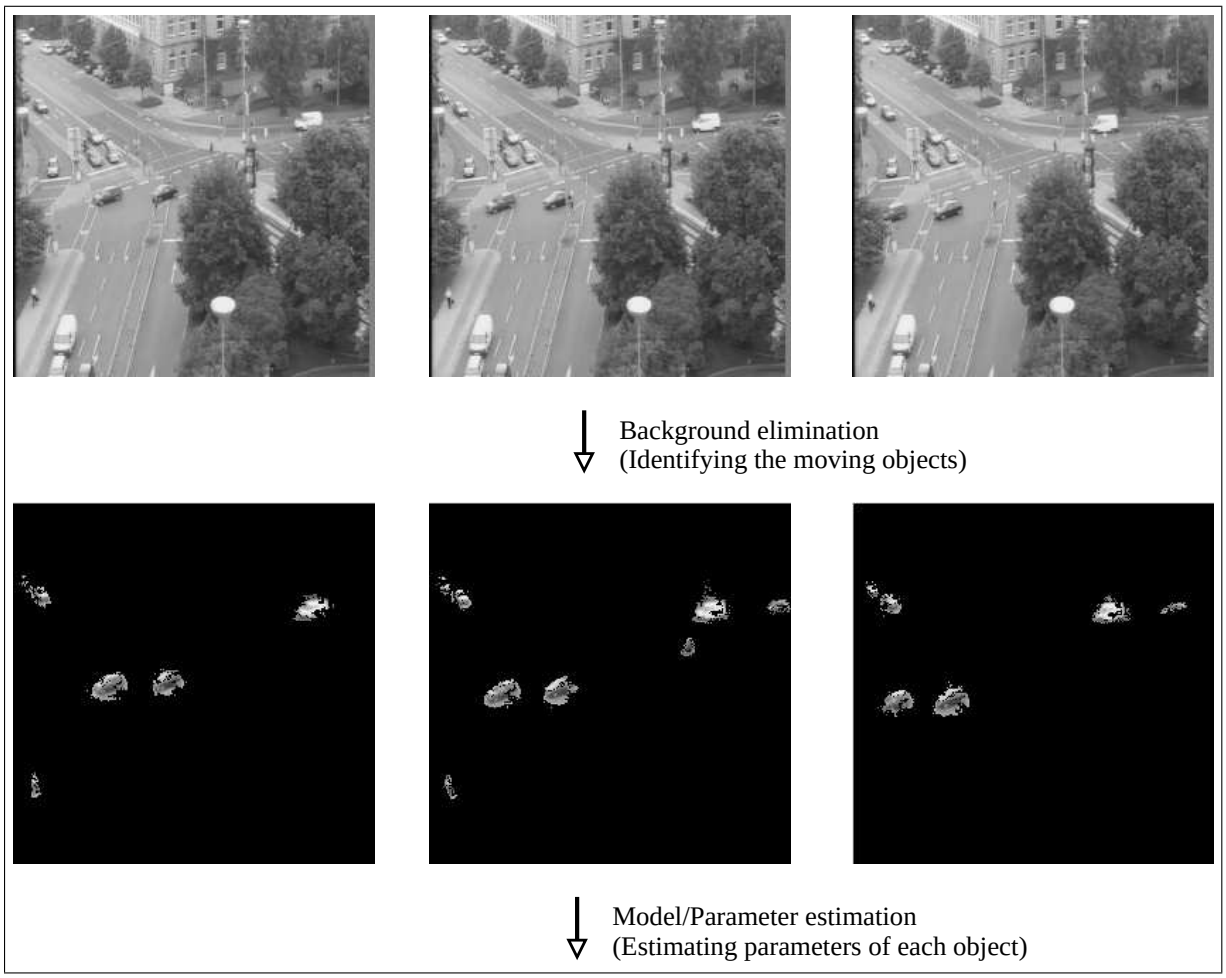


Figure 5.2 An overview of the video analysis procedure. First the objects (defined as consistently moving regions) are detected. The features (2D coordinates in this case) are extracted from each object. Then, the most likely model and its parameters representing each object are computed.

Figure 5.2. The analysis of these videos proceeds in three stages: (i) detecting objects, (ii) extracting features for each object, and (iii) estimating model parameters which describe the motion of each object.

We used a pixel-based approach developed by Zivkovic [121] to *subtract* the background and detect the objects in the scene. In this method, the intensity of each pixel is modelled as a mixture of Gaussians. The pixels are classified as either background or foreground using a Bayesian decision criterion. The adaptive nature of this method makes it robust to variations in the background. As stated previously, moving objects which come to a halt and remain in the state of rest are incorporated into the background model. In other words, we identify only moving regions as objects. Figure 5.2 illustrates the quality of foreground extraction on a few sample frames. The features corresponding to the objects are then used to estimate the model parameters which describe the object motion. Figure 5.3 illustrates the likelihoods

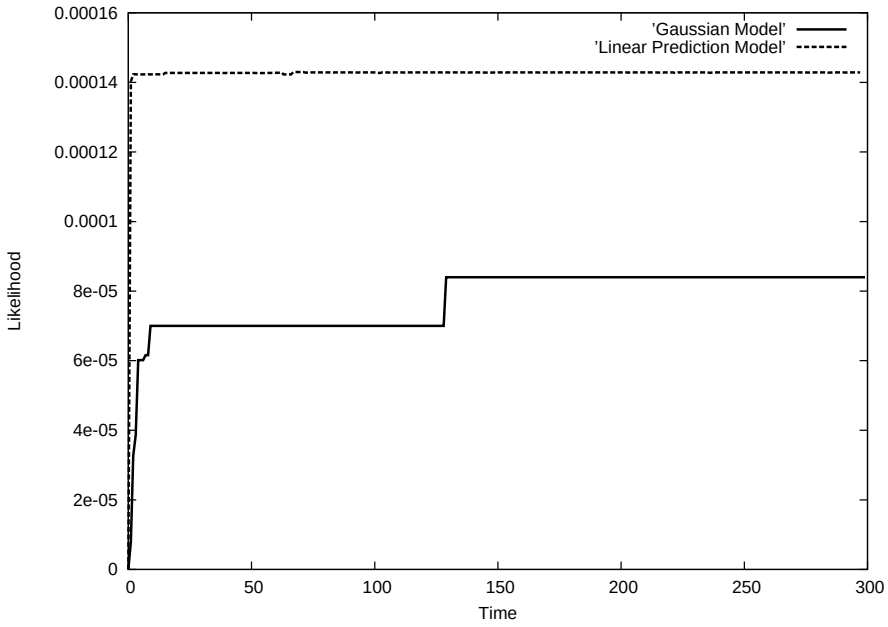


Figure 5.3 A plot showing the variation of likelihoods of the car object (refer Figure 5.2) with respect to Gaussian and linear predictor models. It can be observed the car is most likely described by a linear predictor model, which is acceptable because the car undergoes a linear motion in the video sequence.

computed for one of the cars in the video sequence (refer Figure 5.2). It can be observed that the Linear Prediction model likelihood represents the motion of the car better.

5.4 Summary

In this chapter we presented a technique to automatically analyze video sequences and identify the motion parameters, which are most likely to describe the objects in the scene. We discussed an unsupervised scheme for estimating these parameters. The parameters as well as the models are allowed to vary over time. The work presented here is only a step towards building learning-based video analysis systems. It promises to open many new avenues to *understand* video content. Firstly, the feature extraction scheme can be investigated. Other statistically strong models (refer Chapters 2 and 4) can be successfully used here. Secondly, an integrated framework for identifying the objects and then estimating the model parameters could be a promising approach. In our method we distinguish between object extraction and model parameter estimation. Thirdly, an interesting direction of research would be to associate the motion models to known events in the world.

Chapter 6

Conclusions

We have presented techniques for analyzing videos by interpreting the dynamic events in them. Our contributions are mainly in three main aspects, namely, event modelling, feature selection, and event recognition.

We proposed a model which, given a continuous video, builds a hierarchical representation of the video and also generates its XML content. Such high level abstractions of video have a large potential for application in browsing and retrieval systems. The model first learns efficient representations of events from an example set of video sequences. We achieved results comparable to those reported in literature, using a stronger mathematical model. We also demonstrated the use of this model for identifying *unusual* activities. Although this approach resulted in efficient event representations, it may not be optimal for recognizing events. We then proposed a method for selecting features which are useful for *recognizing* rather than just *representing* events. We found that all the video segments (actions) of an event may not be equally important for the recognition task. We presented a discriminant-based algorithm for identifying the video segments and their relative statistical importance. Using these relative weights we computed a similarity measure for comparing two video sequences. The main advantage of this approach is that it does not involve a careful choice of parameters. It is sufficiently robust to any parameter initializations. We illustrated the superiority of this method by testing it on hand gesture and human activity videos. We also presented an adaptive feature selection technique which chooses an optimal combination of spatial (offline) and temporal (online) features in events. Our claim that a fixed feature selection method is not appropriate for a set of events is supported by the results we achieved. A significant improvement in the recognition accuracy, when tested on hand gesture and human activity events, was observed. In some cases it may be difficult to obtain a set of example videos for training the learning-based systems described above. Such situations necessitate unsupervised learning frameworks. We describe one such framework for analyzing videos and identifying the motion parameters, which are most likely to describe the objects in the scene. The parameters are well as the models are allowed to vary over time. A preliminary evaluation of the system was done on synthetic as well as real traffic

surveillance videos.

The work presented in the thesis can be considered as a step towards building autonomous video analysis systems. The ultimate goal of these systems is to capture video recordings for long durations, and generate an "intelligent" summary of the observations. The unsupervised learning framework is the most significant part in the system. This framework can be improved upon by using stronger mathematical models. The feature extraction scheme in this case also needs more investigation. The feature selection and video summarization schemes discussed here form important components of the system. They come in handy once the system has enough data accumulated for using them as training examples. However, our summarization approach is limited to providing video summary by "extracting" the essential content. The interesting problem of "synthesizing" the summary of the video is largely unaddressed. Also, the discriminant-based feature selection scheme can also extended to multiple classes for wider applicability.

Appendix A

MFA-based Activity Recognition*

Most activities[1] are characterized by considerable amount of spatiotemporal variation. Activities are composed of homogeneous units, henceforth referred to as *actions*, many of which are common to more than one activity as shown in Figure A.1. To develop an effective recognition framework, it is essential to have a representation which can capture these activities efficiently.

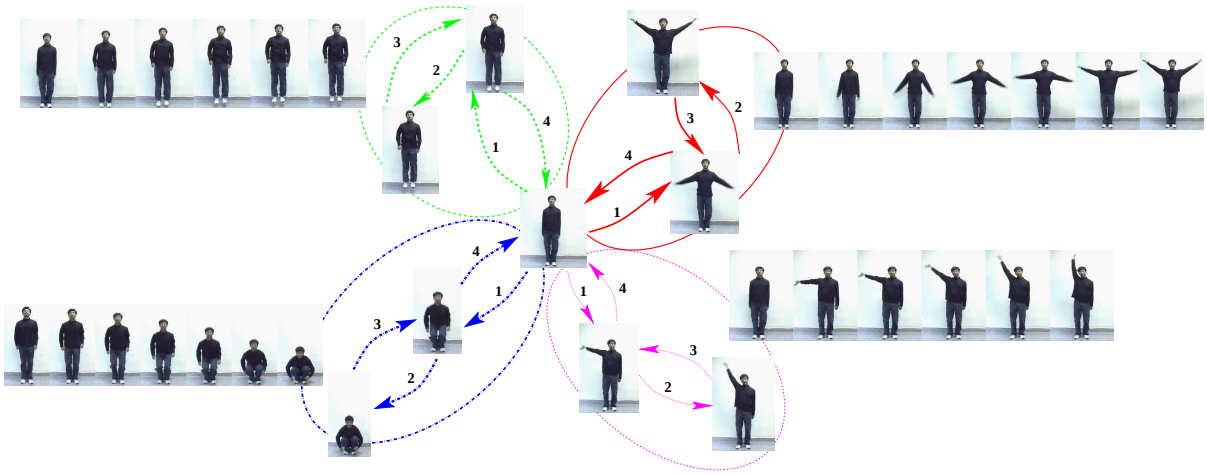


Figure A.1 A sample of human activities (image strips) and their action representatives (individual frames). A set of actions and the transitions among them constitute an activity. Four activities and their corresponding actions are shown as distinct groups here (Green (Top Left) - *Jumping*, Red (Top Right) - *Flapping*, Blue (Bottom Left) - *Squatting*, Magenta (Bottom Right) - *Waving*). The arrows denote the temporal transitions between the actions and the number on each arrow denote the temporal sequencing of the activity. In addition, there are self-loops for each action (not shown in the figure). Note that the action 'standing' is common to all of these activities.

Here we discuss a model to learn a compact representation, exploiting the redundancies like HMM, etc. An activity is modelled as a sequence of atomic spatiotemporal units called *actions*. Human activities are constrained by the degrees of freedom allowed for joints and muscles of the human body and hence,

*This appendix is based on our prior work [6, 93].

[1]The terms *activities* and *events* are used interchangeably in this appendix.

Figure A.2 A few sample frames of human activities: Squatting (top row), Flapping (bottom row). Note the presence of a common *action* – Standing – between these activities in the initial few frames.

limited to a finite set of actions. The problem of characterizing human activities can, therefore, be modelled as that of identifying the constituent actions and their sequencing information. Given a large number of video segments, we employ a probabilistic method to learn these individual actions and their compositional rules for the corresponding activities. These actions, in turn, are represented in a lower dimensional space exploiting the spatial redundancy. Identifying the actions from a given video is not trivial and therefore, we learn the actions from examples. Our approach is comparable to the probabilistic models, such as HMM, GMM, etc., popularly used in activity recognition [48, 66, 81, 99]. However, we capture the activities in a very low-dimensional space which speeds up the entire recognition process.

As described before, the activities are modelled in a low-dimensional subspace. We performed a quantitative analysis of the sub-space by reconstructing the original sequences from the learnt representations. In other words, using Λ_j and the low-dimensional representation, z_t, we recover the original frame, x_t, $\forall t$, thereby generating the entire sequence. The reconstruction error was found to be 0.5%. A comparison of some of the original and recovered frames is shown in Figure A.3. To quantify the performance of the model in identifying the inherent m actions, we compute the within-action and between-action scatter. These statistics are shown in Table A.1. Low within-action scatter values indicate that the frames grouped to belong to a particular action are similar, while high between-action scatter values indicate that the actions (clusters) are well separated – a useful property when clustering data.

Within-action scatter		Between-action scatter	
1	4.9945	1 - 2	12.6427
2	4.5310	2 - 3	11.6940
3	3.9053	3 - 1	13.4344

Table A.1 Performance of the model in identifying the actions among the activities. The values were computed with the 3 actions learnt from activities Squatting and Jumping. Low within-action and high between-action scatter values are observed.

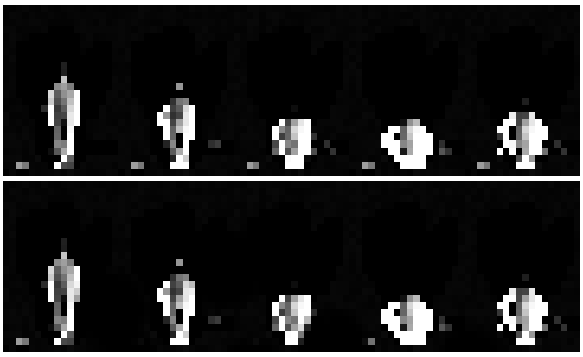


Figure A.3 A comparison of the original (top) and reconstructed (bottom) frames of the activity Squatting. Even though we achieve 99.94% reduction in size, the reconstruction error is negligible (0.5%).

A.1 Modelling of Activities

We consider a generative process for the ensemble of activities based on the MFA model. An activity (captured as a set of frames) is composed of various actions. A typical frame of the activity, x_t, can be generated as follows. The action to which it belongs to is chosen following the discrete distribution $P(\omega_j), j = 1 \ldots m$. Depending on the chosen action, a continuous subspace representation z_t is generated according to the distribution $p(z_t \mid \omega_j)$. Having learnt z_t and action ω_j, we obtain the observed x_t according to the distribution $p(x_t \mid z_t, \omega_j)$. That is, x_t is modelled as a "mixture model of actions" as follows:

$$p(x_t) = \sum_{j=1}^{m} \int p(x_t \mid z_t, \omega_j) p(z_t \mid \omega_j) P(\omega_j) dz_t, \tag{A.1}$$

where $\omega_j, j = 1 \ldots m$ denotes the jth action. The above is essentially a reduced dimensionality mixture model where the m mixture components are individual actions. The equation describes the probability of generating a frame given the action (to which it belongs) and its corresponding subspace representation. Our task is to invert the generative process and learn the parameters of these distributions from the frames of *all* the activities. We perform this by using an EM algorithm. It is a general method of finding the maximum-likelihood estimate of the parameters of an underlying distribution (Equation A.1 in this case) from a given data set when the data has missing or unknown values [23]. In the context of video sequences, the data corresponds to frames, the unknown values to the lower-dimensional representations of these frames, and the actions to which these frames are associated. The procedure is explained in further detail below.

A.1.1 EM Framework for Learning

EM alternates between inferring the expected values of hidden variables (subspace representation and actions) using observed data (frames), keeping the parameters fixed and estimating the parameters un-

derlying the distributions of the variables using the inferred values. The videos of all the activities of the subjects are represented as a sequence of frames and are used for training. The two phases of the EM algorithm – Inference and Learning – are executed sequentially and repeatedly till convergence. The E-step (Inference) proceeds by computing $E[\omega_j \mid x_t]$, $E[z_t \mid \omega_j, x_t]$ and $E[z_t z_t^T \mid \omega_j, x_t]$ for all frames t and actions ω_j, while in the M-step (Learning), we compute parameters π_j, Λ_j, μ_j and Ψ.

During the E-step we use the following equations

$$\begin{aligned} E[\omega_j z_t \mid x_t] &= h_{tj}\beta_j(x_t - \mu_j) \\ E[\omega_j z_t z_t^T \mid x_t] &= h_{tj}(I - \beta_j\Lambda_j + \Lambda_j(x_t - \mu_j)(x_t - \mu_j)^T\beta_j^T), \end{aligned}$$

where

$$\begin{aligned} h_{tj} &= E[\omega_j \mid x_t] = \pi_j\mathcal{N}(x_t - \mu_j, \Lambda_j\Lambda_j^T + \Psi), \\ \beta_j &= \Lambda_j^T(\Lambda_j\Lambda_j^T)^{-1}. \end{aligned} \tag{A.2}$$

Here each $\mu_j, \Lambda_j,\ j = 1 \ldots m$ denotes the mean and the corresponding subspace bases of the mixture j respectively. The mixing proportions of actions in the activity are denoted by π. The noise in the data is modelled as Ψ. h_{tj} can be interpreted as a measure of the membership of x_t in class j. More details about MFA and its EM solution can be found in [39].

After the EM algorithm converges, we form the action transition matrix $T_k = [\tau_{pq}^k]$ for each activity A_k as follows.

$$\tau_{pq}^k = \sum_{t=1}^{N-1}[c_t = p][c_{t+1} = q]; \quad 1 \le p, q \le m \tag{A.3}$$

where c_t denotes the class label of the frame x_t and is given by $c_t = \arg\max_j h_{tj}; j = 1 \ldots m$. The entries in the transition matrix T_k represent the transitions of actions for successive frames of the activity A_k. In other words, the matrix T_k encodes the temporal characteristics of the activity. Normalizing the entries gives the corresponding probability transition matrix P_k. In the M-step the statistics collected during the inference from *all* the training examples are used to obtain better estimates of the parameters. We solve a set of linear equations to find π_j, Λ_j, μ_j and Ψ. The interested reader may refer to [39] for more details. x_i is assigned to a class c_i according to

$$c_i = \arg\max_j h_{ij} \quad j = 1 \ldots m \tag{A.4}$$

Thus, by the end of training phase, we obtain the parameters of the model – $\{(\mu_j, \Lambda_j)_{j=1}^m,\ \pi, \Psi\}$, $\{P_k\}_{k=1}^K$. The model, which now encapsulates the activity structure, can be employed for various tasks such as recognition, summarization. In this work we only discuss the recognition scheme.

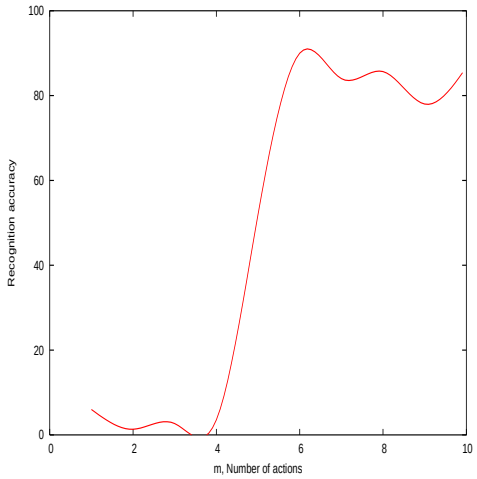


Figure A.4 Graph showing the recognition accuracy (y axis) with respect to the number of actions (x axis), considering Flapping, Jumping, Squatting and Waving activities.

A.1.2 Recognition

We recognize activities in an unlabelled video using the parameters obtained in the training phase. Let the activity being recognized have N_s frames. We reduce the dimensionality of the problem by using the factors *learned* from the training data. We also compute the membership of the frames in each of the actions (from Equation A.2). Each frame is then assigned a single action label using the Equation A.4. Let $c_1, c_2 \ldots c_{N_s}$ be the action assignments for the respective frames. Then, the probability of the video frames to be from the kth activity, S_k, is computed using $S_k = \prod_{t=1}^{N_s-1} P_k[c_t][c_{t+1}]$. The unlabelled video is assigned to be the activity A_k^*, which maximizes S_k. If the test video has more than one activity, we can obtain each of the activities present by observing the ranges of selected features extracted from the subject performing the activity.

There are two subjective decisions to be made in this approach: choosing the number of (1) factors, and (2) mixtures. Our experience shows that the change in accuracy is insignificant beyond a certain number of factors or mixtures. As seen in Figure A.4, recognition accuracy is dependent on the number of actions, m, we assume in the model. However, beyond a certain limit, increase in m does not show any appreciable improvement in the performance.

A.2 Implementation, Results and Discussion

We first describe the implementation details of the two phases in our approach, namely, modelling (training phase) and recognition (testing phase). In both these phases, we begin by preprocessing the video data in a similar fashion. The collection of videos is preprocessed by subtracting the background and

then binarizing the individual frames of all the activities performed by various subjects. In the activities which involve movement of the subject across the field of view (as in the case of the activity Hopping, shown in Figure A.5), motion compensation is performed to center the subject in every frame. This data is stacked into a matrix and is normalized to have a zero mean and unit covariance. The normalized data is used to learn the representation of the activities in terms of actions (in a low-dimensional subspace) and their corresponding probability transition matrix (refer Section A.1.1). In the testing phase, the learnt parameters are used to compute the subspace representation for the new preprocessed video which is to be recognized. In other words, we execute only the E-step during the testing phase (refer Section A.1.2).

The proposed approach differs from various time-series models in many aspects. Our techniques for preprocessing, feature extraction, representation and recognition have considerable advantages, as described below.

- **Preprocessing and Feature Extraction:** Attempts have been made in the past to extract features which summarize an activity by modelling its recency and spatial density [12, 72]. Higher-order image features, which correspond to attributes of various body parts such as joint locations and inter-joint angles (obtained by temporal isolation via tracking [112]), have also been used. Other popular approaches for feature extraction are based on motion parameter vectors [99], measurements of relative distances and velocities [48], colour and motion densities [12, 81]. In contrast, we perform minimal preprocessing and avoid any explicit feature extraction. It is limited to background subtraction and binarization of the individual frames. Some of the approaches seek to obtain low-dimensional features by exploiting the covariance structure of the activity via methods such as PCA. In case of our model, the relevant lower dimensional representation is *automatically* obtained from the observed intensity distribution.

- **Representation:** For many approaches, the extracted features themselves represent the activity [12]. Some of the methods assume that the form of data distribution is known. They use a compact feature information to represent the activity in terms of the distribution parameters [112]. Probabilistic methods such as GMMs and HMMs are popularly used to achieve this [48,66,81,99]. Our model is similar in spirit to a standard left-to-right HMM. However, we work at a lower dimension, which is simultaneously obtained while modelling the activity structure. Typically, separate HMMs are trained for modelling each activity [99] in the ensemble of activities. In our case, a single observation model achieves the same. Conceptually, we believe that multiple activities share the same actions (observation model) and therefore one model is enough for their representation.

- **Recognition:** For many of the methods based on explicit feature extraction, K-nearest neighbour classifier and its variants are used for recognition [12, 72]. In this aspect, we employ a procedure similar to methods with probabilistic representations [48, 81, 99]. Typically, when the model is

applied for a recognition task, the likelihood of the observation sequence is computed. Instead, we compute the likelihood for the sequence of actions *inferred* from the observations (refer Section A.1.2). The video is assigned to the activity which maximizes this value.

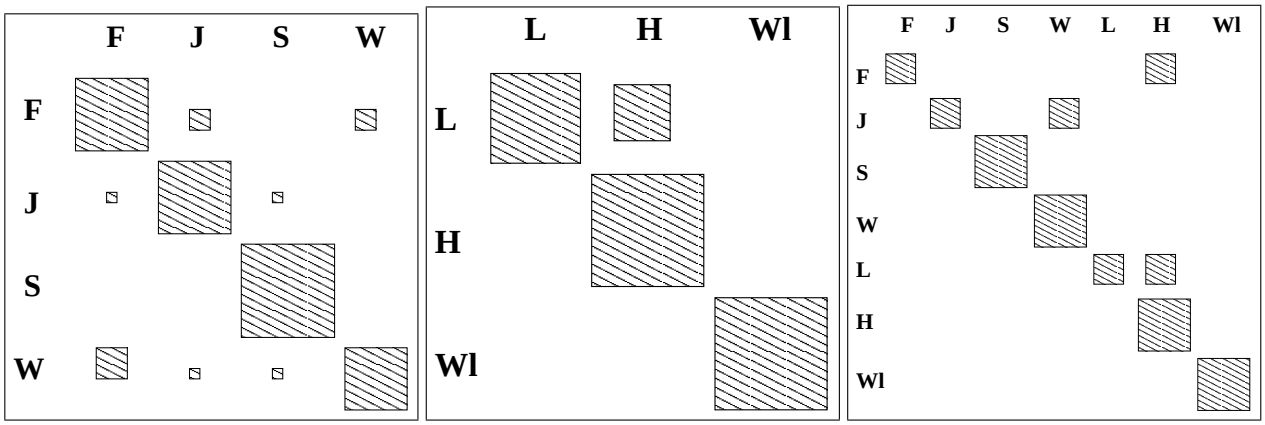
We demonstrate the applicability of the model for different kinds of human activity recognition through the following examples.

A.2.1 Example 1: Recorded data

Recognition of activities involving the whole body finds many applications in surveillance. These activities usually occur with the subject either stationary or indulging in locomotion. In the former category, we consider activities Flapping, Jumping, Squatting and Waving (top row of Figure A.5), while in the latter category (involving locomotion), we consider Limping, Walking and Hopping (bottom row of Figure A.5). We use videos of 20 human subjects performing 7 different activities, of average duration 6 seconds. The videos were captured with a Panasonic Digital Video Camera at 24 fps. As mentioned before, minimal preprocessing is done on the recorded video. In order to retain only the visually significant information, background subtraction and normalization is performed on all the frames. Motion compensation is performed to center the subject for activities where locomotion is involved. To recognize an unlabelled test activity, the frame sequence transitions are computed via the inference step of EM algorithm, which is used to calculate the sequence probability for each activity. The test video is labelled as the activity for which this probability is maximum (refer to Section A.1.2).

Figure A.5 Sample frames of in-place activity, Waving (top row) and activity involving motion, Hopping (bottom row).

The ability of the model to accommodate considerable variation in the range and variety of spatial motion is highlighted by the results (Figures A.6(a), A.6(b) and Figure A.6(c) (the entire ensemble of the 7 activities)). The occasional misclassification is present between activities which share spatial coherence to a large degree, for example Flapping and Waving. The recognition accuracy was found to be $88 - 91\%$ for various activities.



(a) (b) (c)

Figure A.6 Confusion Matrices for *in-place* (F - Flapping, J - Jumping, S - Squatting, W - Waving), locomotion (L - Limping, H - Hopping, Wl - Walking) and the entire activity set respectively. The areas of the squares are proportional to the numerical entries of the confusion matrix.

A.2.2 Example 2: HumanID data

We also verified our model using the MoBo Database [45] available from the Robotics Institute, Carnegie Mellon University. The database consists of 25 subjects performing 4 different walking activities on a treadmill. Each sequence is 11 seconds long and was recorded at 30 fps. We used the data corresponding to one of the view angles (vr03_7 of [45]). Sample frames of some of the activities are shown in Figure A.7.

Figure A.7 Sample frames showing activities from the CMU MoBo Database [45].

The activities in this database (Slow walk, Fast walk, Incline walk, Walking with a ball) have subtle differences, which make the recognition task challenging. On an average, the activities have been correctly identified 81% of the times.

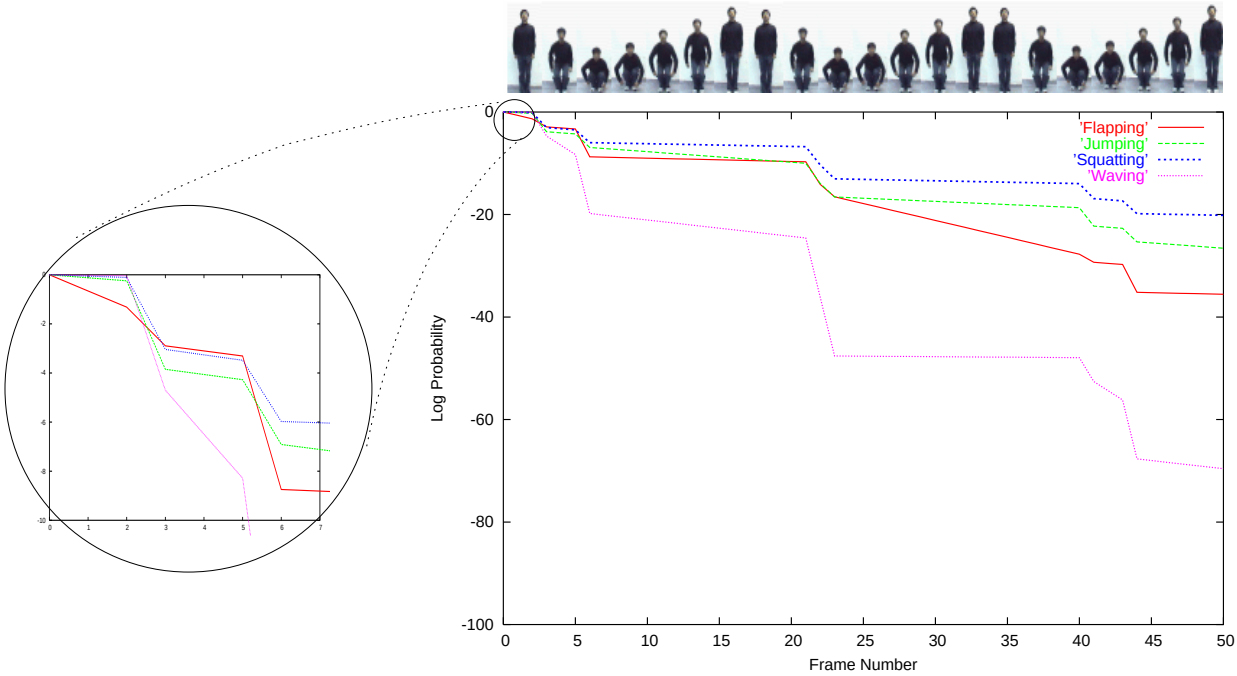


Figure A.8 Cumulative sequence probabilities for the activity Squatting. Sample frames of this activity (performed 3 times) are shown above the graph. The horizontal axis represents the frame number and the vertical axis represents the logarithm of the sequence probability. The uppermost plot (blue dotted line) corresponds to Squatting. A closer view of the graph (shown in inset) indicates that the activity is recognized after observing a few frames – 5 in this case.

A.2.3 Discussions

One of the significant advantages of the model presented is that it frees us from the task of feature extraction. Instead, the features are automatically chosen so as to *best* explain the observed activity in an economical manner. The preprocessing on raw video data is quite minimal. In addition, the model does not incur the computational overhead of subject (agent) tracking since precise spatiotemporal localization is not a primary requirement. The probabilistic framework allows for a coarse localization while leveraging the power of Bayesian inference for learning the actions and subspace representation. Since actions can be learned individually from each activity, the training sequences need not be aligned to actions or possess equal length. This is significant across the example categories also, considering that the activity durations for the recorded data and HumanID data are quite different ($6 - 8$ seconds and 11 seconds respectively). Another feature of the model is that the learned representations are intuitive – they are based on the actions that occur when an activity is performed. This is clearly demonstrated by the representative appearances of actions (as shown in Figure A.1) and also the predominant actions in the activities (Figure A.9). Training for these activities requires limited amount of data (in our case, $3 - 4$ examples per activity were found to be sufficient). Moreover, the advantage of learning a low-dimensional representation such as ours, lies in the accurate recognition of activities in *real-time*. For

real-time applications, the activities are to be identified after observing a few video frames. In Figure A.8 we observe that the activity Squatting is identified in the initial few frames (indicated by the corresponding probability value in the graph). Thus, our model is suitable for such applications. The framework also ensures spatiotemporal capture of all the activities without further constructs such as modelling an activity grammar or the transitions between them etc.

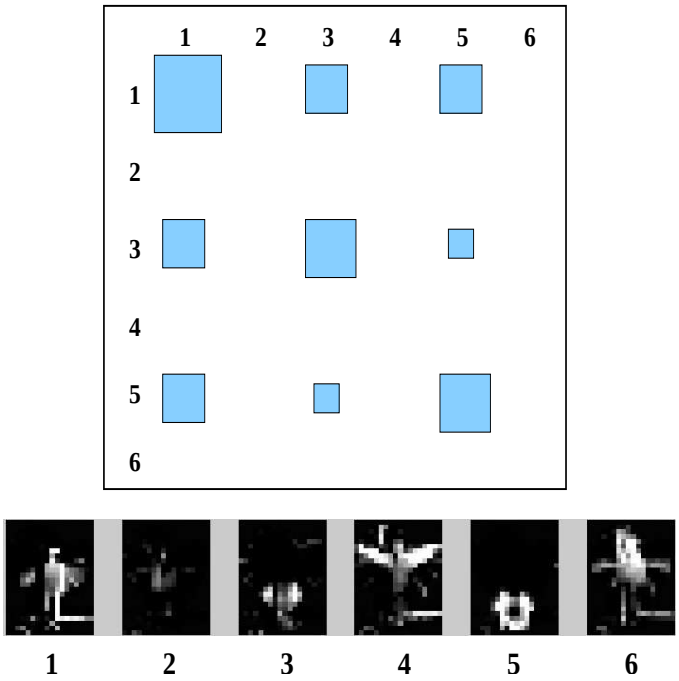


Figure A.9 Cluster transition matrix for the activity Squatting. The rows and columns correspond to the actions learned by the model. The shaded areas are proportional to the numerical probability entries in the transition matrix. Here, squatting is represented by the transitions among clusters $1, 3, 5$. Note the constituent actions – Standing and Sitting – represented by these cluster means.

We illustrate some of the features of the model using the activity Squatting (see Figure A.2) as an example. Our representation needs only 40 (fixed *apriori*) floating point numbers to explain a 320×240 frame, a reduction of nearly 99.94%. This drastic reduction in the size of representation makes the model extremely favourable for applications involving real-time recognition. The recognition process over frames is displayed in Figure A.8, as a plot of the likelihood for each possible activity. The correct activity Squatting – the uppermost plot in the figure – is clearly disambiguated within the first few frames (around 5), which shows the ability of the model to obtain all the aspects of the activity quickly and accurately. The fact that the actions of each activity are properly represented is demonstrated by Figure A.9, which shows the transition matrix for Squatting. The rows and columns correspond to the actions learned by the model. The areas of the squares indicate the transition probabilities between

these actions. Notice that the predominant entries correspond to Standing and Sitting - the main actions present in Squatting.

A.3 Summary

The purpose of this appendix is to a give the details of the Mixture of Factor Analyzer based model proposed earlier. We discussed a framework for modelling various kinds of human activities using MFA. The application of this model for recognizing activities was demonstrated. A low-dimensional representation of the activities is learnt, which captures both the spatial and temporal aspects of activities. This is ideal for applications involving real-time activity recognition. The model has potential for application in continuous video analysis – representation and summarization, for instance.

Bibliography

[1] *Proc. IEEE Workshop on Event Mining: Detection and Recognition of Events in Video*, Madison, Wisconsin, June 18-20 2003.

[2] Special Issue on Video Retrieval and Summarization. In N. Sebe, M. S. Lew, and A. W. M. Smeulders, editors, *Computer Vision and Image Understanding*, volume 92, pages 141–306. 2003.

[3] *Proc. 2nd Workshop on Statistical Methods in Video Processing*, Prague, Czech Republic, May 2004.

[4] A. D. Wilson and A. F. Bobick. Hidden Markov models for modeling and recognizing gesture under variation. In *Hidden Markov models: Applications in Computer Vision*, pages 123–160. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2001.

[5] K. Alahari, S. L. Putrevu, and C. V. Jawahar. Discriminant Substrokes for Online Handwriting Recognition. In *Proc. International Conf. on Document Analysis and Recognition (to appear)*, 2005.

[6] K. Alahari and S. S. Ravi Kiran and C. V. Jawahar. A Spatiotemporal Model for Recognizing Human Activities from Constituent Actions. *submitted to Pattern Recognition (under review)*, 2005.

[7] A. Ali and J. K. Aggarwal. Segmentation and Recognition of Continuous Human Activity. In *Proc. IEEE Workshop on Detection and Recognition of Events in Video (EVENT'01)*, pages 28–35, 2001.

[8] C. Anderson, P. Burt, and G. van der Wal. Change detection and tracking using pyramid transformation techniques. In *Proc. SPIE - Intelligent Robots and Computer Vision*, volume 579, pages 72–78, 1985.

[9] S. Antani, R. Kasturi, and R. Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, 35(4):945–965, 2002.

[10] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[11] J. A. Bilmes. A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical Report TR-97-021, University of California, Berkeley, CA, 1998.

[12] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[13] A. F. Bobick and A. D. Wilson. A state-based technique for the summarization and recognition of gesture. In *Proc. International Conf. on Computer Vision*, pages 382–388, 1995.

[14] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 994–999, 1997.

[15] C. Bregler. Learning and Recongizing Human Dynmaics in Video Sequences. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 568–574, 1997.

[16] L. Campbell and A. Bobick. Recognition of human body motion using phase space constraints. In *Proc. International Conference on Computer Vision*, pages 624–630, 1995.

[17] C. Cédras and M. Shah. Motion-based Recognition: A Survey. *Image and Vision Computing*, 13(2):129–155, 1995.

[18] M. Christel, T. Kanade, M. Mauldin, R. Reddy, M. Sirbu, S. Stevens, and H. Wactlar. Informedia Digital Video Library. *Communications of the ACM*, 38(4):57–58, 1995.

[19] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. A System for Video Surveillance and Monitoring. Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2000.

[20] J. Crowley and F. Bernard. Finger tracking as an input device for augmented reality. *International Workshop on Automatic Face and Gesture Recognition*, pages 195–200, 1995.

[21] J. E. Cutting. Six tenets for event perception. *Cognition*, pages 71–78, 1981.

[22] J. W. Davis. Appearance-Based Motion Recognition of Human Actions. Master's thesis, MIT Media Lab, Cambridge, MA, 1996.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[24] H. Denman, N. Rea, and A. Kokaram. Content-based analysis for video from snooker broadcasts. *Computer Vision and Image Understanding*, 92(2-3):176–195, 2003.

[25] N. Dimitrova, H.-J. Zhang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor. Applications of Video-Content Analysis and Retrieval. *IEEE Multimedia*, pages 42–55, 2002.

[26] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classifi cation*. John Wiley and Sons, New York, 2001.

[27] I. A. Essa. Computers Seeing People. *AI Magazine*, 20(2):69–82, 1999.

[28] B. S. Everitt. *An Introduction to latent variable models*. Chapman and Hall, London, 1984.

[29] F. De la Torre and M. J. Black. Robust Principal Component Analysis for Computer Vision. In *Proc. International Conf. on Computer Vision*, volume I, pages 362–369, 2001.

[30] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36(1):259–275, 2003.

[31] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[32] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[33] W. Freeman and M. Roth. Orientation histogram for hand gesture recognition. In *Proc. International Workshop on Automatic Face- and Gesture-Recognition*, 1995.

[34] H. Fujiyoshi and T. Kanade. Layered Detection for Multiple Overlapping Objects. In *Proc. International Conf. on Pattern Recognition*, volume 4, pages 156–161, 2002.

[35] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, New York, second edition, 1990.

[36] A. Galata, N. Johnson, and D. Hogg. Learning Variable Length Markov Models of behaviour. *Computer Vision and Image Understanding*, 81(3):398–413, 2001.

[37] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[38] D. M. Gavrila and L. S. Davis. 3-D model-based tracking of humans in action: A multi-view approach. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[39] Z. Ghahramani and G. E. Hinton. The EM Algorithm for Mixtures of Factor Analyzers. Technical Report CRG-TR-96-1, University of Toronto, Canada, 1996.

[40] N. H. Goddard. Representing and recognizing event sequences. In *Proc. AAAI Workshop on Neural Architectures for Computer Vision*, 1989.

[41] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison Wesley Longman, 1992.

[42] K. Gould and M. Shah. The trajectory primal sketch: A multi-scale scheme for representing motion characteristics. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 79–85, 1999.

[43] H. Greenspan, J. Goldberger, and A. Mayer. A probabilistic framework for spatio-temporal video representation and indexing. In *Proc. European Conference on Computer Vision*, volume 4, pages 461–475, 2002.

[44] A. Gritai, Y. Sheikh, and M. Shah. On the use of anthropometry in the invariant analysis of human actions. In *Proc. International Conf. on Pattern Recognition*, volume 2, pages 923–926, 2004.

[45] R. Gross and J. Shi. The CMU Motion of Body (MoBo) Database. Technical Report CMU-RI-TR-01-18, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, June 2001.

[46] B. Gunsel, A. M. Tekalp, and P. J. L. van Beek. Content-based access to video objects: Temporal segmentation, visual summarization, and feature extraction. *Signal Processing*, 66:261–280, 1998.

[47] G. Halevy and D. Weinshall. Motion of disturbances: Detection and tracking of multi-body non-rigid motion. *Machine Vision and Applications*, 11(3):122–137, 1999.

[48] R. Hamid, Y. Huang, and I. Essa. ARGMode - Activity Recognition using Graphical Models. In *IEEE Workshop on Event Mining: Detection and Recognition of Events in Video*, 2003.

[49] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-Time Surveillance of People and Their Activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[50] D. Hogg. A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[51] http://homepages.inf.ed.ac.uk/rbf/CVonline/applic.htm.

[52] http://i21www.ira.uka.de/image_sequences/.

[53] http://www.ginmiller.com/gmf04/gmfstore/vids/waltz.htm.

[54] http://www.wisdom.weizmann.ac.il/mathusers/vision/EventDetection.html.

[55] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[56] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[57] C. V. Jawahar, P. K. Biswas, and A. K. Ray. Detection of clusters of distinct geometry: A step towards generalised fuzzy clustering. *Pattern Recognition Letters*, 16:1119–1123, 1995.

[58] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.

[59] N. Jojic and B. J. Frey. Learning Flexible Sprites in Video Layers. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 199–206, 2001.

[60] B. H. Juang and L. R. Rabiner. Mixture autoregressive hidden markov models for speech signals. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 6:1404–1413, 1985.

[61] A. Kapoor. Automatic Facial Action Analysis. Master's thesis, MIT Media Lab, Cambridge, MA, June 2002.

[62] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis. A survey of video processing techniques for traffi c applications. *Image and Vision Computing*, 21(4):359–381, April 2003.

[63] K. Knight and D. Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artifi cial Intelligence*, 139(1):91–107, 2002.

[64] A. Komlodi and G. Marchionini. Keyframe preview techniques for video browsing. In *Proc. ACM International Conf. on Digital Libraries*, pages 118–125, 1998.

[65] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning Layered Motion Segmentations of Video. In *Proc. IEEE International Conf. on Computer Vision*, 2005.

[66] K. C. Lee, J. Ho, M.-H. Yang, and D. Kriegman. Video-Based Face Recognition Using Probabilistic Appearance Manifolds. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 313–320, 2003.

[67] A. Levy and M. Lindenbaum. Sequential Karhunen-Loeve basis extraction and its application to image. *IEEE Transactions on Image Processing*, 8(9):1371–1374, 2000.

[68] R.-S. Lin, M.-H. Yang, and S. E. Levinson. Object Tracking Using Incremental Fisher Discriminant Analysis. In *Proc. International Conf. on Pattern Recognition*, volume 2, pages 757–760, 2004.

[69] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li. A user attention model for video summarization. In *Proc. Tenth ACM International Conf. on Multimedia*, pages 533–542, 2002.

[70] I. Mani and M. T. Maybury, editors. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA, 1999.

[71] S. Marcel. Hand Posture Recognition in a Body-Face centered space. In *Proc. Conf. on Human Factors in Computer Systems*, 1999.

[72] O. Masoud and N. Papanikolopoulos. Recognizing Human Activities. In *Proc. IEEE Conf. on Advanced Signal and Video and Signal Based Surveillance (AVSS)*, pages 157–162, 2003.

[73] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York, 1992.

[74] D. McNeill. *Hand and Mind—What Gestures Reveal about Thought*. The University of Chicago Press, Chicago/London, 1992.

[75] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher Discriminant Analysis with Kernels. In *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[76] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[77] H. Murase and S. Nayar. Visual learning and recognition of 3D objects from appearance. *IJCV*, 1(14):5–24, 1995.

[78] S. A. Niyogi and E. H. Adelson. Analyzing and recognizing walking fi gures in xyt. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 469–474, 1994.

[79] J. Oh and K. A. Hua. An Effi cient Technique for Summarizing Videos using Visual Contents. In *Proc. IEEE International Conf. on Multimedia and Expo*, volume II, pages 1167–1170, 2000.

[80] N. Oliver and E. Horvitz. Selective perception policies for guiding sensing and computation in multimodal systems: A comparative analysis. *Computer Vision and Image Understanding*, Available online, 2005.

[81] N. Oliver, E. Horvitz, and A. Garg. Layered Representations for Human Activity Recognition. In *Proc. International Conference on Multimodal Interfaces*, pages 3–8, 2002.

[82] S. C. W. Ong and S. Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):873–891, June 2005.

[83] J. O'Rourke and N. I. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):522–536, November 1980.

[84] R. Plamondon and S. N. Srihari. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.

[85] R. Polana and R. C. Nelson. Detecting activities. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2–7, 1993.

[86] D. A. Pomerleau. Knowledge-based training of artifi cial neural networks for autonomous robot driving. In J. Connell and S. Mahadevan, editors, *Robot Learning*, pages 19–43. Kluwer Academic Publishers, Boston, 1993.

[87] F. Quek. Unencumbered gestural interaction. *IEEE Multimedia*, 3:36–47, 1997.

[88] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE*, 77(2):257–286, February 1989.

[89] J. M. Regh and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proc. International Conf. on Computer Vision*, pages 612–617, 1995.

[90] P. L. Rosin and T. Ellis. Image difference threshold strategies and shadow detection. In *Proc. British Machine Vision Conference*, pages 347–356, 1995.

[91] S. Roweis. EM Algorithms for PCA and SPCA. *Neural Information Processing Systems*, pages 626–632, 1997.

[92] Y. Rui, S. X. Zhong, and T. S. Huang. Efficient access to video content in a unified framework. In *IEEE International Conf. on Multimedia Computing and Systems*, volume 2, pages 735–740, 1999.

[93] S. S. Ravi Kiran, K. Alahari, and C. V. Jawahar. Recognizing Human Activities from Constituent Actions. In *Proc. National Conf. on Communications*, pages 351–355, 2005.

[94] H. Sakoe and S. Chiba. Dynamic Programming Optimization for Spoken Word Recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26:43–49, 1978.

[95] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, New York, 1983.

[96] M.-K. Shan and S.-Y. Lee. Content-based retrieval via motion trajectories. *SPIE*, 3561:52–61, 1998.

[97] M. A. Smith and T. Kanade. Video Skimming and Characterization through the Combination of Image and Language Understanding. In *Proc. IEEE International Workshop on Content-Based Access of Image and Video Database*, 1998.

[98] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, 1999.

[99] X. Sun, C. C. Chen, and B. S. Manjunath. Probabilistic Motion Parameter Models for Human Activity Recognition. In *Proc. International Conference on Pattern Recognition*, volume 1, pages 443–446, 2002.

[100] X. Sun, C. C. Chen, and B. S. Manjunath. Probabilistic Motion Parameter Models for Human Activity Recognition. In *Proc. International Conf. on Pattern Recognition*, volume 1, pages 443–446, 2002.

[101] K.-W. Sze, K.-M. Lam, and G. Qiu. Scene cut detection using the colored pattern appearance model. In *Proc. IEEE International Conf. on Image Processing*, volume 2, pages 1017–1020, 2003.

[102] H. Tao, R. Kumar, and H. S. Sawhney. Dynamic layer representation with applications to tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 134–141, 2000.

[103] A. M. Tekalp. *Digital Video Processing*. Prentice Hall, New Jersey, 1995.

[104] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 23(3):297–303, 2001.

[105] M. Turk and A. Pentland. Eigenfaces for recognition. *Cognitive Neuroscience*, 3(1):71–86, 1991.

[106] J. Y. A. Wang and E. H. Adelson. Layered Representation for Motion Analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 361–366, 1993.

[107] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003.

[108] C. R. Wren, A. Azarbayejani, T. J. Darrell, and A. P. Pentland. PFINDER: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[109] Q.-Z. Wu, I.-C. Jou, and S.-Y. Lee. On-Line Signature Verification using LPC Cepstrum and Neural Networks. *IEEE Transactions on on Systems, Man and Cybernetics - Part B*, 27(1):148–153, 1997.

[110] Y. Wu, L. Jiao, G. Wu, E. Chang, and Y. F. Wang. Invariant Feature Extraction and Biased Statistical Inference for Video Surveillance. In *Proc. IEEE Workshop on Advanced Video and Signal-based Surveillance*, pages 284–289, 2003.

[111] T. Xiang, S. Gong, and D. Parkinson. Autonomous Visual Events Detection and Classifi cation without Explicit Object-Centered Segmentation and Tracking. In *Proc. British Machine Vision Conference*, pages 233–242, 2002.

[112] Y. Yacoob and M. J. Black. Parameterized modelling and recognition of activities. *Computer Vision and Image Understanding*, 73:232–247, 1999.

[113] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden Markov model. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 379–385, 1992.

[114] M.-H. Yang, N. Ahuja, and D. Kriegman. Face Detection Using a Mixture of Factor Analyzers. In *Proc. IEEE International Conference on Image Processing*, volume III, pages 612–616, 1999.

[115] M.-H. Yang, N. Ahuja, and D. Kriegman. Face recognition using kernel eigenfaces. *Proc. IEEE International Conf. on Image Processing*, 1:37–40, 2000.

[116] H.-S. Yoon, J. Soh, Y. J. Bae, and H. S. Yang. Hand gesture recognition using combined features of location, angle and velocity. *Pattern Recognition*, 34(7):1491–1501, 2001.

[117] Y. Yusoff, W. Christmas, and J. Kittler. Video Shot Cut Detection Using Adaptive Thresholding. In *Proc. British Machine Vision Conference*, 2000.

[118] L. Zelnik-Manor and M. Irani. Event-Based Analysis of Video. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 123–130, 2001.

[119] H. Zhong, J. Shi, and M. Visontai. Detecting Unusual Activity in Video. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[120] A. Zisserman and R. Hartley. *Multiple View Geometry in computer vision*. Cambridge University Press, 2000.

[121] Z. Zivkovic. Improved Adaptive Gaussian Mixture Model for Background Subtraction. In *Proc. International Conf. on Pattern Recognition*, volume 2, pages 28–31, 2004.