Automatic Indexing of Video Content via the Detection of Semantic Events

by

Bart Lehane, B.Eng.

A dissertation submitted in fulfilment of the requirements

for the award of Doctor of Philosophy

Dublin City University

School of Electronic Engineering

Supervisor: Dr. Noel E. O'Connor

June, 2006

Declaration

I hereby certify that this material, which I now submit for assessment in the programme of study leading to the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my own work.

Signed : Berta Lohae
ID No. : 98606069
Date : 20/06/06

Abstract

The number, and size, of digital video databases is continuously growing. Unfortunately, most, if not all, of the video content in these databases is stored without any sort of indexing or analysis and without any associated metadata. If any of the videos do have metadata, then it is usually the result of some manual annotation process rather than any automatic indexing. Thus, locating clips and browsing content is difficult, time consuming and generally inefficient. The task of automatically indexing movies is particularly difficult given their innovative creation process and the individual style of many film makers. However, there are a number of underlying film grammar conventions that are universally followed, from a Hollywood blockbuster to an underground movie with a limited budget. These conventions dictate many elements of film making such as camera placement and editing. By examining the use of these conventions it is possible to extract information about the events in a movie. This research aims to provide an approach that creates an indexed version of a movie to facilitate ease of browsing and efficient retrieval. In order to achieve this aim, all of the relevant events contained within a movie are detected and classified into a predefined index. The event detection process involves examining the underlying structure of a movie and utilising audiovisual analysis techniques, supported by machine learning algorithms, to extract information based on this structure. The result is an indexed movie that can be presented to users for browsing/retrieval of relevant events, as well as supporting user specified searching. Extensive evaluation of the indexing approach is carried out. This evaluation indicates efficient performance of the event detection and retrieval system, and also highlights the subjective nature of video content.

Acknowledgements

It is a pleasure to thank the many people that have made this thesis possible. This Ph.D. has been such a large part of my life for the past number of years, and so many people have assisted me on this journey, it is quite difficult to know where to begin.

I guess the best place to start is to thank everybody at the CDVP for all their help. It has been an absolute privilege to work alongside such a talented group of people. There always seemed to be somebody available to lend a helping hand and offer support when it was most needed. It is impossible to name each of you individually, so a big thank you to you all.

This thesis would not have been possible without assistance from the higher echelons of the CDVP. I would sincerely like to thank Prof. Alan Smeaton, Dr. Noel Murphy and Dr. Sean Marlow for all their help.

It is impossible to imagine how this thesis would have turned out, or indeed if it would have existed, without my supervisor, Dr. Noel O'Connor. His input throughout the course of my Ph.D. cannot be understated. Despite his hectic schedule, he was always at hand to offer excellent advice and guidance. Noel also had the unenviable job of trawling through the early drafts of this thesis, red pen in hand, to correct the many mistakes present. In some countries, correcting my grammar is considered a form of torture, and I am forever indebted to Noel for his patience throughout this process.

I would also like to thank the people closest to me, my family. My parents, Don and Ans, have always been at hand to give much needed advice, usually when I need it most. My family have been hugely supportive of the path I have chosen, despite the fact that I've been a student for almost eight years now!

Finally, I must thank Claire for putting up with my bad humour when things didn't go according to plan. Your unbelievable patience, constant encouragement, and knowledge of the land of pixels helped me more than you realise. Táim i ngrá leat.

Publications Arising From This Research

Related Publications

Searching Movies Based on User Defined Semantic Events - Bart Lehane, Noel O'Connor and Hyowon Lee. International Conference of Signal Processing and Multimedia Applications (SIGMAP), Setúbal, Portugal, 7-10 August 2006.

Movie Indexing via Event Detection - Bart Lehane and Noel O'Connor. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Incheon, Korea, 19-21 April 2006.

Dialogue Sequence Detection in Movies - Bart Lehane, Noel O'Connor and Noel Murphy. International Conference on Image and Video Retrieval (CIVR), Singapore, 20-22 July 2005.

Action Sequence Detection in Motion Pictures - Bart Lehane, Noel O'Connor and Noel Murphy. European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (EWIMT), London, U.K., 25-26 November 2004.

Dialogue Scene Detection in Movies using Low and Mid-Level Visual Features - Bart Lehane, Noel O'Connor and Noel Murphy. International Workshop on Image, Video, and Audio Retrieval and Mining, Quebec, Canada, 25-26 October 2004.

Contents

List of Figures

List of Tables

Glossary of Commonly used Acronyms

1-D	One Dimensional
2-D	Two Dimensional
3-D	Three Dimensional
ADR	Automatic Dialogue Replacement
BMA	Block Matching Algorithm
CAT	Computed Axial Tomography
CD	Compact Disk
DCT	Discreet Cosine Transform
DSP	Digital Signal Processing
DVD	Digital Versatile Disk
FPS	Frames Per Second
FSM	Finite State Machine
GOP	Group Of Pictures
HMM	Hidden Markov Model
HMSS	High Motion Short Shot
HZCRR	High Zero Crossing Rate Ratio
JPEG	Joint Photographic Experts Group
KKT Conditions	Karush-Kuhn-Tucker Conditions
MP3	MPEG-1 Layer Three Audio
MPEG	Moving Picture Experts Group
RGB	Red Green Blue
RMS	Root Mean Square
SR	Silence Ratio
SVM	Support Vector Machine
TREC	Text REtrieval Conference

TRECVID	Text REtrieval Conference VIDeo Track
VC Confidence	Vapnik Chervonenkis Confidence
VC Dimension	Vapnik Chervonenkis Dimension
VHS	Video Home System
VLC	Variable Length Coding
VO	Video Object
YUV	Y = Luminance Component. U,V = Colour Components
ZCR	Zero Crossing Rate

CHAPTER 1

Introduction

1.1 The Growth of Digital Content

The world of video creation has seen steady growth since its inception in the late 1800's. The initial fascination of seeing moving images on a screen soon gave way to the realisation that, as a medium, video has a multitude of applications. Currently, video has found applications in diverse areas such as personal entertainment, security and communications. Until recently, most of this video data was in analogue form, stored on reels or tapes and transmitted through projectors, radio waves or analogue cables. The advent of digital media breathed new life into the world of video and opened up a whole new range of applications, as well as improving on many of the previous ones. In most of the areas where analogue video proved to be so useful, digital video is now used. It has revolutionised the way in which video data is viewed, stored and transported.

Where previously analogue tapes were used to store video, it is now stored on digital disks or a hard drive. Similarly, the old coaxial cables used for television broadcasting are now being replaced by digital cables. Sales of DVDs and players now far outstrips their analogue counterpart, the VHS (or Betamax) video. In 2003, Video Store Magazine reported that DVD-Video sales were $12.1 Billion, compared to $2.4 billion sales for videos. Similarly, although not as widespread as DVD use, many movie theatres are beginning to convert from reel-based to digital-based projection. Terrestrial television broadcasting has also benefited greatly from the advent of digital video, as it is possible to transmit more reliable, higher quality, video and audio. Another area that has benefited highly from somewhat different advances in digital video is Internet communications. It is possible to communicate via an Internet video conference to virtually any part of the

world. Whereas in a DVD, the main attraction is an unrivalled level of video quality, in a low bitrate application such as web-based communications it is important for the video size (bitrate) to be as small as possible, while still ensuring that the video quality is sufficiently high for user viewing. Similarly, many television networks (news stations in particular) now stream digital video via their websites. It is also possible to watch live 'webcasts' through many websites, allowing any computer with an Internet connection to mimic the functionality of a television set. Also, as can be seen from many news stations, it is possible to send a reporter to a remote location and transmit live video of broadcastable quality to a central studio. Advances in video coding have paved the way for these very different examples of video applications to become a reality.

But why has digital video taken over from its analogue counterpart? Perhaps the most obvious reason is the quality of the video reproduction. Digital video can be copied extremely reliably over and over again, while the quality of analogue video tends to degrade each time it is transferred from one tape to another. Another advantage of digital video is the methods of storage. DVDs, for example, can hold more video data at a higher quality than a VHS or Betamax tape. Also, extra features can be added to a DVD such as optional subtitles, language selection, adjustable video ratio etc., which can not be added to a video tape. The cost of distributing a movie in digital form to cinemas is considerably less than distributing one on film. Digital video data can also, of course, be stored on a hard drive. While in the early days of digital video one main problem was the cost of storage, in recent times the price of this memory/storage has consistently fallen, while the compression rates achieved by standardisation bodies, such as MPEG, means that the amount of memory digital video consumes is also falling. Both of these factors mean that storing video in digital format has become far more cost-effective, and there are many large collections of digital video in existence. Most television companies now store all of their produced content in digital form, and many others are in the process of transferring their analogue archives into a digital format. Most, if not all, applications that use analogue video are being converted to digital video, and new uses of digital video are being dreamt up every day.

Virtually all video content now produced is available in digital format, whether directly filmed using digital equipment, or transmitted and consequently stored digitally (for example played via digital television). Just as digital video now facilitates efficient storage and transmission of content, it has also made the creation of movies and television programs easier and cheaper. Previously, many movies and television programs were shot using expensive cameras that used reels, which also required expensive film. Now, however, the cost of high quality digital cameras has reduced significantly, and unused video can simply be overwritten, saving the money that unused reels cost. This has led to a large increase in the amount of video being created. For example, the number of films created

in 1991 was just under six thousand, while the number created in 2001 was well over ten thousand[1]. This increase can largely be attributed to film creation becoming more cost effective, which resulted in an increase in the number of independent films. Also, editing equipment is now compatible with home computers which makes cheap post-production of movies possible. For example, the film 'Tarnation' by Jonathan Caouette which appeared at the Sundance film festival 2004, was made for a total of $218.32.

Video is not the only benefactor of the digital revolution. The way in which audio is captured, stored and transmitted has also changed significantly in the past number of years. The distribution of commercial music in particular has shifted almost entirely toward digital audio. Although many music shops still sell vinyl records, their most popular sales medium are CDs, which contain high quality digital recordings. The sale of analogue cassette tapes has all but disappeared. As with DVDs for video distribution, the main reasons for this digital takeover were an increase in the quality of the audio, and a reduction in the amount of degradation over time.

Although the digital media revolution was perhaps driven by efficient storage and transmission, an advantageous by-product of using digital, as opposed to analogue, video and audio, is that it is possible to analyse the data. For example, where previously, video was merely stored on a reel of tape, digital video is represented by a set of pixel values that represent the colour of the video. Thus, using digital signal processing (DSP) theory, it is possible to extract information from the video data and use it to gain knowledge about the content. This created the broad field of video analysis which now includes many diverse areas such as surveillance video analysis, face recognition, fingerprint recognition, object detection and object tracking. Also, there is much active research in areas that examine video and image management principles such as digital watermarking, compression techniques and video error protection.

Implications of Digital Video Growth

It is clear that there is a large amount of video in existence which is stored in digital format throughout the world. Unfortunately, most, if not all, of this content is simply stored without any sort of indexing or analysis and without any associated metadata. If any of the videos have metadata, then it is due to some manual annotation rather than any automatic indexing. Thus, locating relevant portions of video or browsing content is difficult, time consuming and generally inefficient. Automatically indexing these videos to facilitate their presentation to the user would significantly ease this process.

There are many video content types that are in need of indexing. Sporting programs, news programs, or documentaries are just some of the content genres created each day. However, fictional video content, specifically movies, are a medium particularly in need

[1]Source = www.imdb.com

of summarisation for a number of reasons. Firstly, their temporally long nature means that it is difficult to manually locate particular portions of a movie, as opposed to, say, a thirty minute news program. Most movies are at least one and a half hours long, with many as long as three hours. Although other fictional content, i.e. television series' (dramas, soap operas, comedies etc.), may be as short as thirty minutes, many episodes of television series' are an hour long, so are also difficult to manage without indexing.

Summarisation of fictional video is also hindered due to its challenging nature. Each television series or movie is created differently, using a different mix of directors, editors, cast, crew, plots etc., which results in varying styles. Also, it may take a number of months to shoot a two hour film. Due to this length of time, filmmakers are given ample opportunity to be creative in how they shoot each scene, which results in diverse and innovative video styles. This is in direct contrast to the way most news and sports programs are created, where a rigid broadcasting technique must be followed as the program makers have an extremely short time span with which to make their program. Thus, summarisation of fictional video content (particularly movies) is particularly challenging and is the topic under study throughout this research.

1.2 Objectives of this Research

The primary aim of the research in this thesis is to develop an approach to automatically index movies and fictional television content by examining the underlying structure of the video, and extracting knowledge based on this structure. This should allow for efficient location of relevant portions of a movie or fictional television program. This research also aims to make this indexing entirely automatic. The indexing process should not involve any human interaction, and no manual annotation should be required.

Although discussed more thoroughly in Chapter 2, it is necessary to introduce the concept of an event at this point. Essentially, an event is something which progresses the story onward. Events are the portions of a movie which viewers remember as a semantic unit after the movie has finished. A conversation between a group of characters, for example, would be remembered as a semantic unit ahead of a single shot of a person talking in the conversation. Similarly, a car chase would be remembered as 'a car chase', not as 50 single shots of moving cars. A single shot of a car chase carries little meaning when viewed independently, it may not even be possible to deduce that a car chase is taking place from a single shot, however, when viewed in context of the surrounding shots in the event, its meaning becomes apparent. Events are components of a single scene, and a scene may contain a number of different events. For example, a scene may contain a conversation, followed by a car chase, which would be two distinct events. Similarly, there may be three different conversations (between three sets of people) in the

same scene, corresponding to three different events. Thus, the detection and presentation of relevant events in movies forms the crux this research toward achieving the objectives stated above, and indeed the first step required in order to create an event-based index is to create an ontology of events that is capable of describing the activities in movies.

In order to evaluate the event detection techniques, a representative test set should be defined and a rigorous testing process undertaken. This test set should contain a wide range of fictional video content representing many different styles and genres. In order to ascertaining the reliability of the event detection system it's results should be compared to a manual annotation of the test set.

Although a comprehensive review of existing approaches will be undertaken prior to designing an event detection system, in order to further ascertain it's reliability, an existing state of the art solution should be implemented and benchmarked against the proposed event detection techniques.

Another objective of this research is to facilitate searching of movie content. Although the event-based index aims to contain all relevant events in a movie, there may be occasions when a different set of events are sought. The searching should facilitate more specific, user defined browsing through a movie, so that a user can select the features most likely to appear in the desired event. The addition of searching allows for tailored retrieval of events using audiovisual information.

This research also aims to build a system that can present an indexed version of a movie to a user. This will allow users to browse an event based index of a movie, and locate specific events. This system should also facilitate searching of movies, by allowing users to retrieve events based on a set of selected features. It should be possible to compare various approaches to video retrieval using this system.

In summary, the main aims of the research are as follows:

- Create a relevant event-based ontology
- Create a system capable of automatically indexing movies and fictional television programs
- Place an event-based structure on the video data by detecting all of the relevant events in a movie/fictional television program
- Gather a representative test set to evaluate the system
- Compare the event detection approach with the existing state of the art
- Implement a system that facilitates user-specified event-based searching of video
- Build an interface that presents the detected events, and also allows searching of the video

Although much of the work in this thesis is concerned with creating a system that facilitates event based video retrieval, it is also envisaged that the correlation between film making conventions and audiovisual features be investigated. This may indicate to which extent film makers utilise audiovisual effects, and lead to a more comprehensive understanding of the film making process. This investigation will show to which extent film grammar principles can be used to automatically extract information about movies.

1.3 Potential Applications

There are numerous application areas where an indexed version of fictional video content could be used. As mentioned earlier, there are a huge amount of movies created every year, and the rate of film creation is continuously rising. Thus, there are large databases of existing movies (and fictional television programs) that are growing at a rapid rate. By associating metadata with these videos, the task of retrieval of relevant clips is eased. Another scenario where movie indexing is advantageous may be in an educational environment, e.g. where students are examining particular films as part of a media communications course. Each film on the course could be presented to the students in the form of an event based index, with a set of representative keyframes for each event. This would make locating clips for their studies easier. Yet another scenario could be in a movie rental application. Currently, in order to rent a movie on-line, a movie is selected from a website and it is then posted out to the customer (who posts it back after the movie has been viewed). As download speed increases, it will soon be possible to directly download a movie on demand and view it instantly. In this environment, a user may decide to examine a film before renting it. By presenting a indexed version of a movie, and allowing the customer to view a small number of customer-chosen events, he/she can decide whether he/she would like the movie or not. Currently, most movie previews are in the form of trailers. These are effectively advertisements made to promote a movie and it is quite difficult for a viewer to ascertain whether he/she would like a movie based purely on a trailer. However, by viewing a summary and watching a portion of the movie, a viewer can actually see the activities on screen and thus garner a better impression of the film. This not only has advantages to the viewer, but also to filmmakers as the cost of creating a trailer decreases significantly. Although many big budget films have sufficient funds to create a trailer, many independent filmmakers do not, so this may be a cost effective method of advertising their work.

Video searching would allow for many different usage scenarios. This could be generic (i.e. find all of the events in this film that contain high amounts of motion, as the searcher is looking for a clip with a moving camera), or specific (i.e. a searcher looking for the conversation between Jim and Paul that takes place in a car as they drive to

the party). A search based approach, where a user of a movie database (e.g. a student, a journalist, a researcher, a documentary maker) can actively seek portions of a movie where a particular feature occurs would not only allow them to locate clips, but also to examine how a particular director uses a feature (music, for example) in his/her films.

1.4 Video Summarisation: A Review of Existing Approaches

1.4.1 Introduction

As indicated in Section 1.1, much of the video content viewed is now created in digital form. Management of video databases has become an increasingly important task as the amount of video produced continues to grow. Due to the high labour costs involved in manual annotation, in recent years a large amount of research has been carried out on automatically extracting semantics and applying structure to digital video. Visual analysis techniques are being applied to consumable video such as sports, news, general television content, and movies. The aims of the analysis are to provide efficient methods of video abstraction, browsing and retrieval. This section provides a review of the most popular techniques in video analysis and summarisation that have emerged in recent times. This review is created from an audiovisual analysis point of view, and studies into the narrative structure of video are not considered (for an in depth discussion on film creation techniques, see Chapter 2).

Existing video summarisation approaches can be broadly classed into three categories. As illustrated in Section 2.2, each category corresponds to one level in the video hierarchy. The first of which, shot-based summarisation, attempts to create an index of a movie by detecting all of the shot boundaries. Shot boundaries are used in this work as a baseline for higher level analysis, thus, existing methods of shot-boundary detection are presented in Section 1.4.2. Although not directly applicable to this research, a shot-based index can also be used as a basis for creating what is known as a video skim. For completeness, a review of video skimming techniques is presented in Section 1.4.2.

Detecting shot boundaries is a fundamental step in analysing video content. Although a shot-based structure may be desirable in some cases, typically video content consists of a large number of shots. It is usually advantageous to impose some other, higher level, structure on the video data. Many video summarisation approaches build upon a shot boundary detection system to detect a higher level component of the video. One such component, usually present in television programs and movies is a *scene*. Thus, the second category of video summarisation is based on detecting the scene boundaries in a video. Existing approaches to scene boundary detection are presented in Section 1.4.3.

The final video summarisation category aims to detect relevant events in a video and therefore create an event-based index. These approaches have the significant advantage of

providing semantics to the video index, as the class of any detected events will be known. This is in contrast to the shot- and scene-based approaches where no knowledge about the content can be inferred from the index. As this research aims to create an event-based index of video content, existing event-based summarisation approaches are presented in Section 1.4.4. Following this, the advantages and disadvantages of each of the video summarisation categories is discussed in Section 1.4.5.

1.4.2 Shot Boundary Detection and Keyframe Selection

The first step in most video summarisation techniques is shot boundary detection. A representative selection of the leading approaches in this area are presented here. As it is a fundamental problem, much of the early work in video analysis concentrated on this task. As 'hard' cuts (i.e. when the video moves from one shot to another in successive frames) are the most common shot boundary type, most techniques focus specifically on detecting this type of shot cut. Boreczky et al. [3] identify the main techniques in shot boundary detection as pixel difference, statistical difference, histogram comparison, edge difference, compression difference and motion vector based. Each of these techniques were implemented, and extensively tested on a dataset consisting of general television programs, news, movies and commercials. In general, it was found that the the simpler algorithms (specifically colour histogram based algorithms) performed best. Browne et al. [4], also implemented three different shot boundary detection methods, A colour histogram based approach, an edge histogram based approach, and an approach using the motion information encoded in the macroblocks of MPEG-1 data. The three approaches were tested on thirteen different videos, ranging from news to cookery programs. Overall, it was found that a colour histogram approach outperformed the edge detection and macroblock based approaches.

The TRECVID series of workshops evaluates different approaches in video retrieval by providing a common video dataset, as well as a set of tasks, to participants. Participants complete the required tasks, and submit their results. As there is a common video data set, a direct comparison of different techniques can be made. For the past number of years, TRECVID has run a shot boundary detection task [5]. In this task, a common set of video (most recently, news content in 2005) is supplied to participants, and they are required to submit a set of automatically detected shot boundary times. Many of the techniques in this shot boundary task are now evolving to detect long fades or dissolves, as the established methods of detecting hard cuts achieve very accurate results. The approach of Petersohn et al [6] was one of the more successful techniques in TRECVID. This technique focuses on detecting three types of shot transitions, hard cuts, dissolves/fades and wipes. Each of the shot transitions are detected independently using colour, motion and edge data, and the results are combined to produce a final set of shot boundaries. Similarly, Volkmer

et al. [7] use a moving window throughout the frames in a video, and detect gradual transitions by comparing the colour information of all frames in the window. The use of a sliding window means that it is possible to map the changes between frames over a long period of time, rather than just from frame to frame. As gradual transitions typically take a large number of frames, the authors claim that this approach is more suitable.

For many applications the selection of a representative frame for each shot (known as a keyframe) is straightforward. Any of the frames contained within the shot could be used as a keyframe. Many applications choose the keyframes temporally, by picking the first, last, or middle frames of a shot. However, there are more sophisticated approaches to keyframe selection. Vermaak et al. [8] aim to select keyframes that are distinct from every other frame in a shot, and therefore, they claim, carry the most information. Colour information is used to analyse frame dissimilarity. Cooper et al [9] extract keyframes with the aim of only extracting keyframes that are dissimilar to other keyframes, thus ensuring that much information is presented in a relatively low amount of images. DCT coefficients are used to discriminate between frames in their work. Girgensohn et al [10] extract keyframes from entire videos, rather than individual shots, by examining the temporal length of the shots, as well as clustering information based on colour similarity.

Video Skimming

One way of utilising the shot boundaries to create a summary of the video content is by simply presenting one keyframe from each shot to the user. However, as there are typically quite a large amount of shots in a video, this is usually not the best approach. Another way of presenting a summary to the user is to use a *video skim*. This concept presents a user with a subset of the relevant shots or frames in a video, which the user can then browse. In some cases video playback speed may increase over the less important portions of video. Despite the reduced amount of content presented to the user, the skim should still contain a high amount of information, thus, skimming techniques aim to remove the redundant areas of the video, while retaining the relevant areas. Smith et al [11] extract the significant audio and visual information of a video using audio keywords, specific objects and by examining the video structure. The presence of significant words in the audio track is used as an indicator as to the relevance of a portion of the video, as is the presence of faces, or the lack of motion in a shot. Christel et al [12] surveyed a number of different skimming techniques, comparing simple incremental shot selection with more sophisticated selection techniques. After a set of user tests, it was found that incorporating speech, language and image processing into skim video creation produced skims more satisfactory to users. Di Lecce et al [13] also evaluated a number of video skimming techniques, and found that colour histograms and Hough transform based approaches created the best video skim.

1.4.3 Scene Boundary Detection

Detecting scene boundaries enhances the known structure of a video, as rather than a *video - shot* representation, a more representative *video - scene - shot* structure is available. Yeung and Yeo [14, 15] propose a technique in which time constrained clustering of shots is used to build a scene transition graph. This involves grouping shots that have a strong visual similarity and are temporally close, and based on these clusters identify the scene transitions. The clustering is threshold-based and any number of clusters can be created. Scene boundaries are located by examining the structure of the clusters and detecting points where one set of clusters ends, and another begins. The approach in [16] (Rui et al.) also aims to group related shots together based on colour and activity histograms, and then detect the scene boundaries based on the shot groupings.

Kender et al [17] utilise the idea of shot coherence in order to find scene boundaries. Instead of clustering similar shots together, the coherence is used as a measure of the similarity of a set of shots with previous shots. The coherence is based on colour similarity using histograms. When there is 'good coherence', many of the current shots are related to the previous shots and therefore judged to be part of the same scene, when there is 'bad coherence', most of the current shots are unrelated to the previous shots and a scene transition is declared. Rasheed et al. [18] also uses the concept of shot coherence to find scene boundaries. A two pass approach is implemented, in the first pass, the coherence of colour features for shots is used to create potential scene boundaries. The second pass involves using motion features, as well as shot lengths to merge scenes and produce a final set of scene boundaries. The ShotWeave system [19], extracts features about each shot using a region based approach, and uses these features to cluster shots and detect scene breaks. The regions used are chosen based on film grammar rules. Troung et al [20] create their own definition of a scene, and implement both edge-based and colour-coherence based approaches to scene boundary detection. Film grammar rules were used to refine the results of both systems, and it was found that the colour-coherence approach yielded better results. Liu et al [21] use hidden Markov models, trained on visual information, to extract a scene-like structure on documentaries.

While each of the scene detection methods described thus far only use visual features, many techniques also utilise the audio track. Sundaram et al [22] define a computable scene as one which exhibits long term consistency of chromaticity, lighting and ambient sound. A set of audio features are used to determine fundamental changes in the audio, while a coherence-based model is used to locate changes in the visual features. These are then combined to produce a set of scene breaks. Cao et al [23] use the same definition of a scene as in [22], and also use audio to assist in scene boundary detection. However, the assumption is made that most scene boundaries are determined by visual properties. So, in this approach, scene breaks are firstly detected using visual features, and then false

boundaries are removed based on the divergence of the audio features. Huang et al [24] note that typically a change in image or motion characteristics is associated with a shot break, while a scene break contains a simultaneous change of image, motion and audio characteristics. Three sets of features are then used to locate breaks in audio, colour and motion and scene and shot breaks are then detected based on these features. Li and Kuo [25] implement a window-based sweep algorithm to locate shot sinks[1], which are then used to classify scene breaks. The consistency of the audio is then examined to filter the detected scene boundaries.

Whilst the shot boundary techniques are typically applied to all types of video content, most of the scene boundary detection approaches listed so far have been concerned with detecting scene boundaries in movies or fictional television shows. There has also been significant work in the area of scene boundary detection in news content, although it is typically referred to as a story boundary rather than a scene boundary. The approach of Merlino and Boykin [26] uses hidden Markov models to detect the beginning and end of stories within news programs. Audio, visual and text features are used to generate a set of observable states, and then the hidden Markov model finds the presence of story breaks. O'Hare et al. [27], use a support vector machine based approach to story segmentation. A set of four features (anchorperson shot identification, face detection, motion analysis and shot length analysis) are extracted for each shot, and the support vector machine locates the anchorperson shots which signal the boundary between news stories. One of the tasks in the aforementioned TRECVID workshop is to find story boundaries for news programs. The system developed by Quenot et al [28], is based on detecting changes in the audio associated with a scene break. Changes included a long pause, a change in speaker, as well as the detection of jingles and common phrases commonly spoken by presenters at the end of a news story. The approach implemented by Hoashi et al [29], used a combination of audio, motion, colour and temporal features to assist in story boundary detection. A support vector machine is then used to discriminate between story boundary shots and all other shots. In the summary of the results of the story boundary task in TRECVID, [30] note that a combination of audio, visual, and speech recognition features gives the best overall performance in this task.

1.4.4 Detection of Semantically Relevant Events

There are advantages to placing a scene-based structure on video as it usually results in a video that is easier to navigate, and in some cases can result in a reduction in the amount of keyframes that need to be presented without substantially reducing the amount of information conveyed to the user. However, a scene-based structure cannot be imposed on many types of video content. Most sports programs, for example, do not have any

[1] Shot sinks contain a pool of shots that are visually similar, and temporally close to each other.

specific 'scenes', but a continuous stream of action. Thus, many of the analysis techniques for sports content focus on the detection of the important events, or highlights, of the particular sports broadcast. Assfalg et al [31] detect free kicks, penalties and corner kicks in soccer matches primarily using camera motion and a hidden Markov model. Similarly, Yow et al [32] create image mosaics from soccer matches. Goal post detection as well as camera motion parameters are used to detect highlights. Cabasson and Divakaran [33] look for high motion portions of soccer videos as these typically indicate an important event. Audio analysis is also undertaken, and high energy segments of audio (corresponding to crowd noise or commentator audio) are sought. A rule based approach of combining the audio and motion information is used to detect highlights. In extracting highlights in basketball videos, many approaches, e.g. [34, 35], analyse the motion information to determine which team is currently in the attacking part of the court, and use this to detect scores. Similarly, the approach to cricket highlight detection proposed by Lazarescu et al. [36] is heavily dependent on camera motion properties. The angle, and amount of motion are used to create descriptors for the cricket video. There have been similar approaches in other sports such as tennis [37], American football [38], motor sports [39] and baseball [40].

Although detecting events in individual sports is desirable, a more efficient event detection system should be able to detect events over a range of different sports. Li et al. [41] locate the relevant 'plays' in a sports program. The technique has been applied to baseball, football and sumo wrestling. Sadlier et al. [42] implemented a generic method of detecting highlights in all field sports such as soccer, rugby, Gaelic football and hockey. A number of audio and video features, such as increased audio activity, increased near field visual activity, close up detection, scoreboard activity and crowd detection, are extracted as they constitute the common characteristics that occur in highlights across all field sports. A support vector machine is then used to detect relevant events.

For sports video, event based feature extraction techniques are necessary due to the lack of a scene-based structure. However, event based systems have many advantages over shot or scene based browsing systems for other video types, as indicated in Section 1.3. Many event detection techniques in movie analysis focus on detecting individual event types from the video. Lienhart et al. [43] detect dialogues in video based on the common shot/reverse shot shooting technique. Typically, in a conversation involving two people, the camera will focus on person A, cut to person B, then cut back to person A and so on. This results in an ABABAB.... shot/reverse shot structure. Thus, if repeating shots are detected, a dialogue event is declared. This approach however, is only applicable to dialogues involving two people, since if three or more people are involved the shooting structure will become unpredictable. Also, there are many other event types in a movie or television program apart from dialogues. Kuo et al [25, 44] expand on this idea to

detect three types of events, 2-person dialogues, multi-person dialogues and hybrid events (where a hybrid event is everything that isn't a dialogue). However, again, only dialogues are treated as meaningful events and everything else is declared as a hybrid event. Chen et al. [45] aim to detect both dialogue and action events in a movie, however the same approach is used to detect both types of events, and the type of action events that are detected is restricted. Alatan et al. [46] use hidden Markov models to detect dialogue events. Audio, face and colour features are used by the hidden Markov model to classify portions of a movie as either dialogue or non-dialogue.

There has been relatively few approaches to finding events other than dialogue events in movies. Nam et al. [47] detect violent events in a movie by searching for visual cues such as flames or blood pixels, or audio cues such as explosions or screaming. This approach, however, is quite restricted as there may be violent events that do not contain their chosen features. Kang [48] extract a set of colour and motion features from a video, and then use relevance feedback from individual users to class events into 'fear', 'sadness', or 'joy'. Zhai et al. [49] generate colour, motion and audio features for a video, and then use finite state machines to class scenes into either conversation scenes, suspense scenes or action scenes. A high classification rate is achieved. This approach relies on the presence of known scene breaks, and classifies a whole scene into one of the categories, while in reality an entire scene may contain a number of important events.

1.4.5 Discussion

Section 1.4.2 introduced existing shot boundary detection techniques as well as discussing video skimming. In terms of indexing video, the most common use for shot boundary detection techniques is as a first step in creating a higher-level (i.e. scene/event-based) video index rather than as a stand-alone indexing solution. This is largely due to the temporal length of many video types which mean that there are an excessive amount of shots present. Whilst skimming at least presents a reduced version of a video to a viewer, it is nevertheless quite time consuming to view an accelerated version of an entire movie.

Many of the movie summarisation approaches illustrated in this chapter aim to either place structure on the content by detecting scene breaks [14, 15, 17, 18, 19, 20, 22, 23, 24, 25], or detect a number of events within a movie [25, 43, 44, 45, 46, 47, 49]. There are a number of disadvantages in simply imposing a scene-based structure when compared to an event-based structure. The main drawback is that no knowledge is gained about the content. Knowing where the scene breaks occur has little or no benefit to a user aiming to retrieve particular clips of a movie. In terms of browsing the video, the viewer may as well be using a shot (keyframe) based system as the scene-based system, since all shots will need to be perused. Within the event-based approaches listed in this chapter, many focus on individual events, such as dialogue [25, 43, 44, 46] or action events [45, 47].

Also, some of the approaches that search for dialogue events often constrain the dialogues they are searching for by only counting two-person dialogues [43, 45]. Similarly, some approaches to action event detection focus on a constrained set of action event types [45]. Typically, summarisation approaches were tested on a constrained set of movies [14, 15, 19, 23, 24, 25, 43, 44, 45]. This not only fails to thoroughly assess the proposed system, but also implies that the approach has limited use in a real scenario. As wide a variation of movies as possible should be used in order to test any event detection approach so that its effectiveness can be accurately evaluated.

Although some approaches to movie summarisation utilise well known film grammar rules, in many cases, a lack of understanding of the process of film creation hindered the summarisation techniques. A deep understanding of the process of film creation is used throughout this research in order to assist in the analysis of movies. Films and television programs are creatively made, so an understanding of the underlying methods used in film creation is essential in order to successfully analyse the content. Very little other research has taken place to index an entire movie by detecting all of the relevant events. Even less research has taken place on the concept of searching for individual events within a movie based on dynamic user requirements.

1.5 Approach Employed

The event detection system developed in this thesis is based on film creation techniques. The methods used by directors and editors throughout the world to create films were examined, and the resultant observations were utilised in order to aid in the design of the event detection techniques. Although film creation is a highly creative process, the underlying concepts which were initially devised by the earliest filmmakers have become conventions which audiences have now come to expect in each movie they see. If these conventions are not followed, then the audience may become confused or disoriented. When making a movie, filmmakers have a set of tools with which to project their vision on screen, much like a baker has a set of ingredients with which to make bread. Of course, creativity is required to make each movie interesting and innovative, but the basic filmmaking 'ingredients' will always be present. Chapter 2 illustrates many of these basic tools.

Movies were chosen as the main focus for research in this thesis over other types of fictional video. In general, movies are longer and more diverse than fictional television content, and thus it is more challenging to summarise and index them. However, fictional television content is created in a very similar way to movies. On many occasions, factors which seem specific to movie creation are discussed (for example: film grammar, shooting techniques, the roles of the various filmmakers etc.), however each of these factors are

equally applicable to creating a television series. Thus, techniques that summarise movies are equally applicable to summarising a fictional television program. Many of the movie summarisation techniques presented in this thesis are based on well defined directing and editing principles that filmmakers use. However, directors making fictional television content also use these principles. In fact, given that a television series will usually have a more constrained budget and timescale, the directors and editors are typically more reliant on these principles, as their creative planning is somewhat restricted. Although the main focus of the research is on movies, as they are deemed to be a more challenging dataset, the application of the summarisation techniques to fictional video is also investigated.

In order to automatically index movies, an event is treated as the fundamental unit of retrieval. If each event in a movie is detected and classified, then the movie is deemed to have been completely indexed. Thus, in order to fulfil the aim of indexing a movie, each meaningful event should be detected. These events should be classified according to a relevant ontology. This corresponds to automatically placing an event based index on an entire movie. Having an event based structure associated with a movie has considerable advantages. Firstly, a great deal can be learnt about the content by classifying each event. In terms of retrieving relevant clips, having knowledge about the event classes significantly aids the search process. Secondly, it is possible to browse a reduced portion of the movie in order to locate a desired clip. As such, not only does a searcher know what is transpiring in a movie, he/she also has a reduced search space in which to find a desired clip.

The importance of having knowledge of an event class when searching for a clip cannot be understated. To illustrate this, consider searching for a clip of a movie corresponding to a conversation involving three characters. The three methods of summarising video mentioned in Section 1.4 (shot-based, scene-based, and event-based) are compared here, and how they assist a browser in finding the conversation is noted. The first approach results in a shot-based display. Each shot is presented to the browser, and he/she must navigate through all shots to find the desired conversation. After browsing through the (1000+) keyframe images in a movie, the clip is located (assuming it only takes the browser one pass to find the clip, which may not always be a valid assumption, as in order to find the clip the user must infer what is taking place based only on the keyframes). The second approach results in a scene based presentation. Using this system, the browser will still need to incrementally search through each scene in order to find the clip. The user may be able to browse faster, as shots at the end of an irrelevant scene can be ignored, but still, the need to search through the entire movie exists. Both of these systems have two main drawbacks. Firstly, the entire movie must be perused, which for a movie usually means at least an hour and a half of video. Secondly, there is no context associated with the presented keyframe images, the user cannot ascertain what is transpiring at each point

in the movie from looking at the keyframes.

The final approach results in providing the browser with a list of all conversations in the movie. Instead of every shot in the conversation being displayed, only shots of the characters involved in the conversation are presented. Thus, the browser knows 1) that the images he/she is looking at are part of a conversation, and 2) that the conversation involves the people in the images. As well as a significantly reduced set of shots being presented to the browser, a higher level of knowledge about the shots is obtained. It is not much use knowing that a particular shot belongs to scene 8 (unless the user knows that scene 8 contains the desired conversation, which is unlikely!), but it is useful to know that the shot is part of a conversation between the presented characters. Of course, over the course of a movie, there may be a number of conversations involving the same three characters, but even if there are, say, five such conversations, all that is required to find the specific clip is to browse through five sets of keyframes. Thus, an event based approach is deemed most suitable in order to index movies, and this is the approach followed in this work.

Automatic indexing of a movie is achieved by examining the underlying structure of the movie and extracting knowledge pertaining to events. By studying film grammar, it is possible to deduce why a director places a particular structure on an event. By knowing the tools a director uses to influence an audience's reaction, and by detecting when the director uses these tools, it is possible to infer knowledge about the event, and indeed the film. Automatic audiovisual analysis is used to detect the presence of these filmmaking 'tools', and then based on this extracted knowledge, index a movie by imposing an event based structure on it. Audiovisual analysis involves selecting and implementing a set of features in such a way so that the maximum amount of information about a movie can be extracted. These features are then used by a set of data classification algorithms in order to detect and classify events in movies. As movies are creatively made, each film-maker will shoot an event differently. Thus, when detecting events, the underlying components of the event are targeted (of which at least some must be present), rather than the event as a whole.

It is important to ensure that the correct set of event classes are used. There is an almost infinite range of possible events that could occur in any movie. Creating a set of detectors for each event, is both impractical and unrealistic. Thus, there should be enough event classes to cover all of the relevant parts in a movie, but also, the number of event classes should be limited so that each event class can be well defined, and browsing of a movie is straightforward. Three event classes are defined to achieve these goals: *exciting* events, *dialogue* events, and a superset of different event classes known as *montage* events. A full justification as to why these particular event classes were chosen, as well as an explanation as to the types of events these classes contain is given in Chapter 2.

There is a certain amount of user interpretation involved with each specific event. Although most events can be classified directly into one of the three classes mentioned above, there are cases where multiple interpretations are appropriate. For example, an argument could be classified by one user as a dialogue event, and by another user as an exciting event. Similarly, there are many montages that contain excitement and could be classed as exciting events. When designing a event-based indexing system, it is important to take into consideration different user interpretations. In practise, this means automatically classifying an exciting argument as both a dialogue event and an exciting event so that no matter what the human interpretation, a user can find the sought event.

A technique that facilitates the searching of video for specific events is also implemented. The searching technique is based on many of the same principles as the event detection system, and utilises audiovisual features combined with data classification algorithms.

In order to assess the success of the approach, the summarisation techniques is subjected to a thorough evaluation process. This not only examines the areas where the technique is successful or unsuccessful at detecting events, but also demonstrates the usefulness of an event based summary of a movie by directly comparing an event based summary with a non-event based summary. Also, a comparison of the developed technique with another event detection technique is undertaken with the aim of examining how the created system compares to an existing state of the art summarisation solution.

1.6 Thesis Structure

The remainder of this thesis explains how movies are automatically indexed, facilitated by the detection of relevant events. Firstly, Chapter 2 examines how movies are created and in turn focuses on the three most influential people involved in the movie making process. Based on the findings in this first part of this chapter, a set of system-based decisions are made and are also presented. The complete event detection system design based on these decisions is also described in Chapter 2. For convenience, the description of the components of this system is split across two chapters. The first, Chapter 3, explains the extraction of low-level features and the generation of a set of features for each shot in a movie (termed the shot-level feature vector). The second system description chapter (Chapter 4) explains the high-level event detection process, which makes use of the shot-level feature vector. A method of searching movies using the same event detection principles is also presented. A second approach to event detection, created with the intention of benchmarking the main approach, is also presented in Chapter 4. Following this, Chapter 5 thoroughly examines the results of the event detection process when applied to a diverse a set of movies, and also to a set of television programs. The benchmark-

ing approach to event detection is also examined in this chapter. A system, named the MovieBrowser, that takes the event detection results and presents them to a user is explained in Chapter 6. The search based system is also implemented in the movie browser. A set of experiments using this system, undertaken by a set of volunteers, are presented in Chapter 6. These experiments aim to directly compare different methods of browsing movie content. Finally, Chapter 7 draws a number of conclusions from the research, and indicates potential future work that could build upon the presented research.

CHAPTER 2

Designing an Event-Detection System Based on Film Grammar

2.1 Introduction

This chapter explains the principles of film-making and illustrates how these principles can be leveraged to design a system for event detection. Clearly, creating a movie is a highly creative process, yet there are many rules and conventions that dictate the actions of film-makers, many of which quite subtle to a viewer. These conventions were initially employed by early film-makers, and although they have evolved since, are still in place today. Section 2.2 firstly examines the film-making process, and then illustrates the roles of the most influential people involved in making a movie. Following this, Section 2.3 provides a number of examples illustrating varying film-making techniques. In Section 2.4, a number of decisions on how to analyse movies are drawn, based primarily on the conventions used in the movie creation process. A general structure of the movie index is proposed in which the event classes are chosen. Also, the features required in order to automatically create this index are selected, and an appropriate data classification tool is chosen. Finally in Section 2.5, a complete event detection system is proposed. This system is based on the film creation techniques discussed in Sections 2.2 and 2.4, leveraging video coding methods (explained in Appendix A).

Figure 2.1: Still from the first moving picture taken by Louis-Augustin Le Prince

2.2 General Film Grammar

The world's very first moving picture was thought to have been filmed by Louis-Augustin Le Prince, a Frenchman living in England, in early October 1888. The 'movie' was only a few seconds long, and showed his mother and mother-in-law in his father-in-law's garden (see Figure 2.1). Later on in the same month, he went on to create more films, one of his son playing the accordion, and another of moving traffic on a busy street in Leeds. However he disappeared shortly afterwards, leaving a number of other inventors to create similar machinery. The Edison corporation invented the Kinetoscope and the Lumiére brothers created the Cinématograph in the years following Le Princes' disappearance. In 1903, the Edison corporation created *The Great Train Robbery*, a 12 minute short film containing only 12 shots [50]. From then on, the creation of movies gained in popularity.

Early directors like D.W. Griffith, Douglas Fairbanks, and Sergei Eisenstein pioneered film creation and began to explore the myriad of associated possibilities. They set conventions that laid the foundations of many film-making methods that are still used today. These conventions continued to evolve through the forties, fifties and sixties as film-makers like Orson Welles, John Ford and, later, Alfred Hitchcock begin to exploit the new technology available to them to make more dynamic and interesting movies. In the seventies and eighties, almost all of the film-making tools seen today were well established, and, although directors continued to explore new methods of exciting audiences, convention dictated many aspects of movies. This period introduced the concept of a 'blockbuster' movie, as many movies became more commercial. However, a strong independent film-making tradition continued. Finally, the period from the early nineties to today has seen an increase in the range of possibilities for a movie-maker with the advent of computer generated films. It is now possible to create whole non-existent worlds using computer generated models. Also, continuing on from the eighties, there has also been an increased amount of remakes, re-releases and sequels.

A *frame* is the lowest possible unit in a film. Typically movies are shot and projected at a rate of 24 frames per second. This frame rate is high enough to ensure that viewers don't notice a flickering between frames. A *shot* is defined as "one uninterrupted run of the camera to expose a series of frames"[51], or, a sequence of frames shot continuously from a single camera. Conventionally, the next level in a film is the *scene*. This is made up of a number of connected shots. It is somewhat harder to define a scene as it is a more abstract concept, but [51] labels a scene as "A segment in a narrative film that takes place in one time and space, or that uses crosscutting to show two or more simultaneous actions". Crosscutting occurs when two related activities are taking place and both are shown on screen. For example, in the film *Home Alone* there is a scene in which the parents (on a plane) realise they have left their son behind, and the son (at home) realises that he has been left on his own. The scene contains shots from both locations that are interwoven to illustrate the fact that the shots are occurring at the same time.

An *event*, as used in this research, is defined as a sub-division of a scene that contains something of interest to a viewer. Essentially an event is a semantic chunk of a movie. It could be a conversation, for example, or a series of shots showing people dancing. An event contains a number of shots and has a maximum length of one scene. Usually a single scene will contain a number of different events. For example, a single scene could begin with ten shots of people talking (dialogue event), in the following fifteen shots a fight could break out between the people (exciting event), and finally, end with eight shots of the people conversing again (dialogue event). In Figure 2.2 the structure of a movie is presented. Each movie contains a number of scenes, each scene is made up of a number of events, each event contains a number of shots, and each shot contains a number of frames.

There are three main stages that a film goes through before it is presented to a general audience. The first of which, *Pre-production*, involves writing the script, raising finance, casting, producing a shooting schedule etc. A lot of the responsibility for this phase rests on the shoulders of the producer. The second stage is called *Production*. This involves the actual shooting of the film. The director usually has most input in this stage, and he/she will have a large *director's crew* containing, among others, an art director, a set director, a costume designer, make up artist, special effects unit, sound unit, electrician (gaffer), cinematographer (who the director consults on lighting and camera placement), camera crew, and a cast of actors. As can be seen from the amount of people involved in production, every detail of a movie is carefully considered, from the position of an ornament behind a lead character, to the location of the film shoot itself. Using his/her crew, the director creates a movie, one shot at a time. Most shots will have been planned in preproduction, but there are many opportunities for innovation whilst shooting. A director will often shoot several *takes*, or varying versions, of the same shot. Only one of these

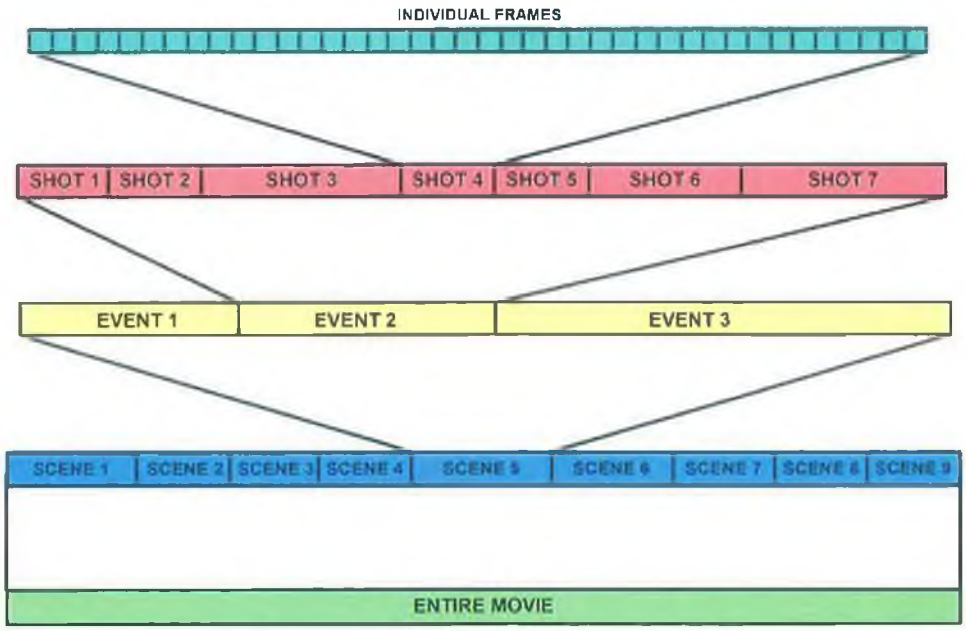

Figure 2.2: Structure of a Movie

takes can eventually make it into the final film. The director will often use a number of different cameras to film a scene from different angles, and later decide on which footage to use [51].

The final stage in creating a movie is *Post-production*. All of the footage shot in the production stage must be refined and filtered to generate a finished movie. There are occasions when production and post-production overlap, if a certain scene needs to be re-shot, for example, but usually they are separate processes. The director and editor must decide which takes are to be chosen for any shot. Also, many shots will be cut from the film and will end up 'on the cutting room floor' (it is every actor's nightmare to only appear on the cutting room floor of a film!). Once the director and editor have worked together to produce a close to final version of the film, a sound editor begins to create the soundtrack. Typically, very few of the noises heard in a film are recorded during the production stage and must be added in by a sound engineer. Similarly, much of the dialogue heard is inserted in the post-processing stage using a process known as automatic dialogue replacement (ADR), which essentially involves re-recording the dialogue. This yields improved quality of sound. Also, special effects may be added at this stage (although they would typically have been generated earlier). Once all aspects of the movie have been created, *release prints* are made for distribution and shown in theatres (different countries may have different release prints). Finally, a movie will usually be promoted by the cast and crew prior to release [51].

Although movie-making is a creative process, there exists a set of well defined conventions, and these *must* be followed [51]. These conventions were established by early

film-makers, and have evolved and adjusted somewhat since then, but they are now so well established that the audience expects them to be followed or else they will become confused. These are not only conventions for the film-makers, but perhaps more importantly, they are conventions for the film viewers. Subconsciously or not, the audience has a set of expectations for things like camera positioning, lighting, movement of characters etc., built up over previous viewings of movies. These expectations must be met, and can be classed as film-making *rules*. Much of this research aims to extract information about a film by examining the use of these rules. In particular, by noting the shooting conventions present at any given time in a movie, it is proposed that it is possible to understand the intentions of a film-maker.

To a certain extent, the film-maker is constrained by convention, as it will dictate many aspects of a scene, but there is still a large array of tools available to film-makers that can be used to allow them to express themselves. Each person, from the lighting engineer to the director, can influence the mood, tone and emotions of the audience. Obviously the skill of the cast have a large impact on the audience, but, in a less obvious way, a dark, murky setting will have a different effect on the audience than a bright, sunny one. Similarly, a camera that hangs on a character for an extra second and gauges a reaction, may inform the audience as to the character's intentions. This information may have been missed had the shot cut away a moment previously. As with any craft, these tools can be used to make a creative and interesting product.

There are often hundreds of people involved in making a single movie. Each person has their own unique role and input into a movie. However, the three people that have the most influence on the production and post production of a movie were identified for the purposes of this work. They are: the *director*, the *editor*, and the *sound engineer*. The roles that each of them play is explained in the following sections, and examples of how they combine to create a finished movie are provided.

Although not all of the film-making techniques presented in the following sections are directly used in this research, for completeness a comprehensive introduction to film-creation is provided. Following this, Section 2.4 illustrates the film-creation techniques directly applicable to this research.

2.2.1 The Role of the Director

The director has perhaps the most influence of any person involved in the making of a movie. It is a director's vision that is portrayed on screen, and ultimately they are responsible for the film, and the viewers' interpretation of it. Viewers need to make up their mind about a shot instantaneously, they shouldn't have to think about what is happening, as the interpretation should be natural. This natural interpretation can usually be achieved by following the conventions alluded to above.

The director must decide what he/she wants to say with the film. It is possible to make a viewer feel a certain way about a scene by using subtle changes in focus. For example, take a scene in which two parents are arguing in front of their son. If the camera shows a shot of both the mother and father shouting at each other in the same frame, followed by a shot of the son, this suggests separation of the boy from the parents. If, however, one shot contains the mother and son, and the other contains the father on his own, this suggests a different configuration of alliances. This small change will influence how an audience feels about the film. Similarly, the director must choose which shots to show. Usually in a conversation, the character talking will be on screen, as he/she is the most important person during the shot. However, this may not always be the case, for example, in a doctor's surgery where a doctor is explaining a prognosis to a patient, it may be more interesting to view the reaction of the patient than the doctor speaking [52].

As mentioned previously, there are a number of rules that a director must follow. One such rule dictates the placement of the camera. It is commonly known as the 180^0 line rule. It was first established by early directors, and has been followed ever since. It is a good example of a rule that, if broken, will confuse an audience. Figure 2.3 shows a possible configuration of a conversation. In this particular dialogue, there are two characters, X and Y. The first character shown is X, and the director decides to shoot him from a camera position A. As soon as the position of camera A is chosen as the first camera position, the 180^0 line is set up. This is an imaginary line that goes between characters X and Y. Any camera shooting subsequent shots must remain on the same side of the line as camera A. When deciding where to position the camera to see character Y, the director is limited to a smaller space, i.e. above the 180^0 line, and in front of character Y. Position B is one possible location. This placement of cameras must then follow throughout the conversation, unless there is a visible movement of characters or camera (in which case a new 180^0 line is immediately set up). This ensures that the characters are facing the same way throughout the scene, i.e. character X is looking right to left, and character Y is looking left to right. If, for example, the director decided to shoot character Y from position C in Figure 2.3, then both characters would be looking from right to left on screen and it would appear that they are both looking the same direction, thereby breaking the 180^0 line rule.

Another example of how this rule dictates camera placement, is if a sequence of shots shows a character walking. If initially he appears to be walking from left to right, then he must continue walking left to right for any following shots, or else the audience will think that the character has turned around and is walking in the opposite direction. Of course, if the camera moves to the other side while the character is in shot, then, as the audience has seen the movement in camera angle, the 180^0 line moves, and the audience won't become confused.

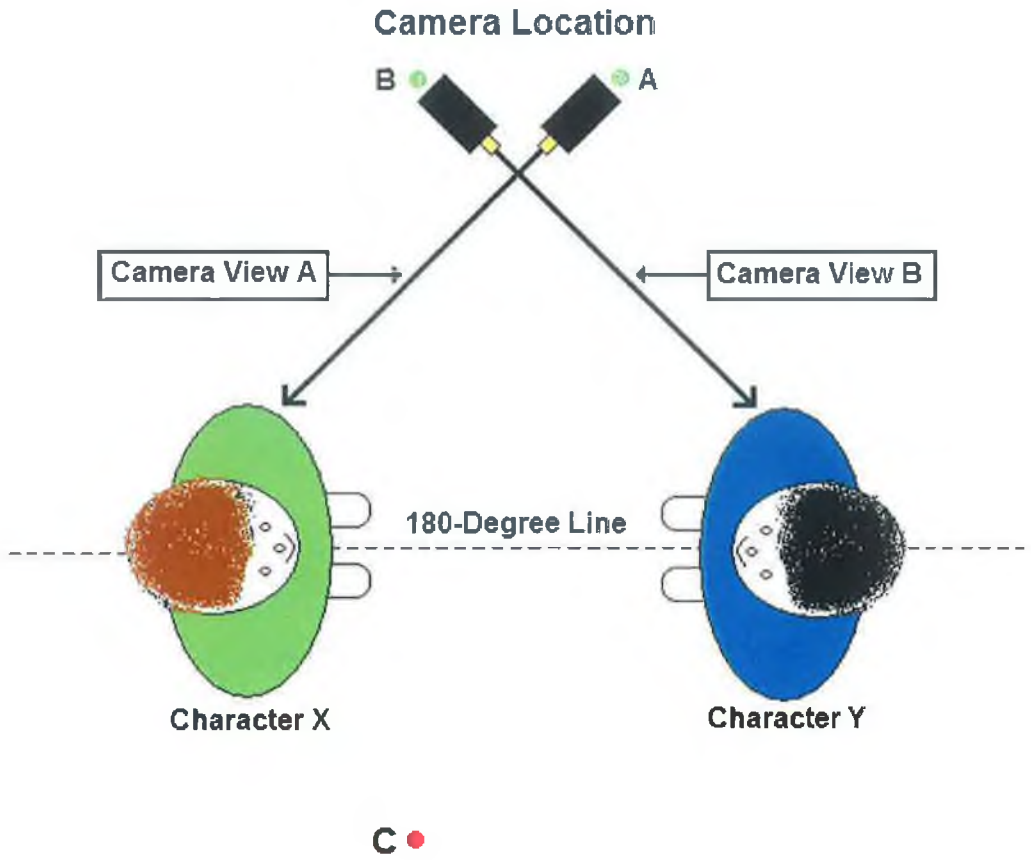

Figure 2.3: Example of 180^0 line rule

The 180^0 line rule is so well established that a viewer will notice any variation in camera placement. This can be exploited by the director. For example, in the film *The Lord Of The Rings*, there is a schizophrenic character called Gollum. He has two personalities, one good and one bad, and he is frequently seen conversing between the two personalities. In order to illustrate which personality was talking, the director filmed the 'conversation' as if there were two people talking, one representing each personality. This can be seen in Figure 2.4. Ordinarily, all shots of the same character should remain on one side of the line, however in this case, camera view A was used for one personality, and camera view B was used for the other. This gives the impression of two people talking as they both appear to be facing different directions. This illustration shows how deeply rooted the 180^0 line rule is as, by breaking the rule, the director enforces the concept of two people talking.

The reason the 180^0 rule became so popular is because it allows the audience to comfortably and naturally view a scene. It is important that viewers are relaxed whilst watching a dialogue in order to take in all of the conversation. The 180^0 line rule is instrumental in ensuring this relaxation is possible [51]. As well as not confusing viewers, the 180^0

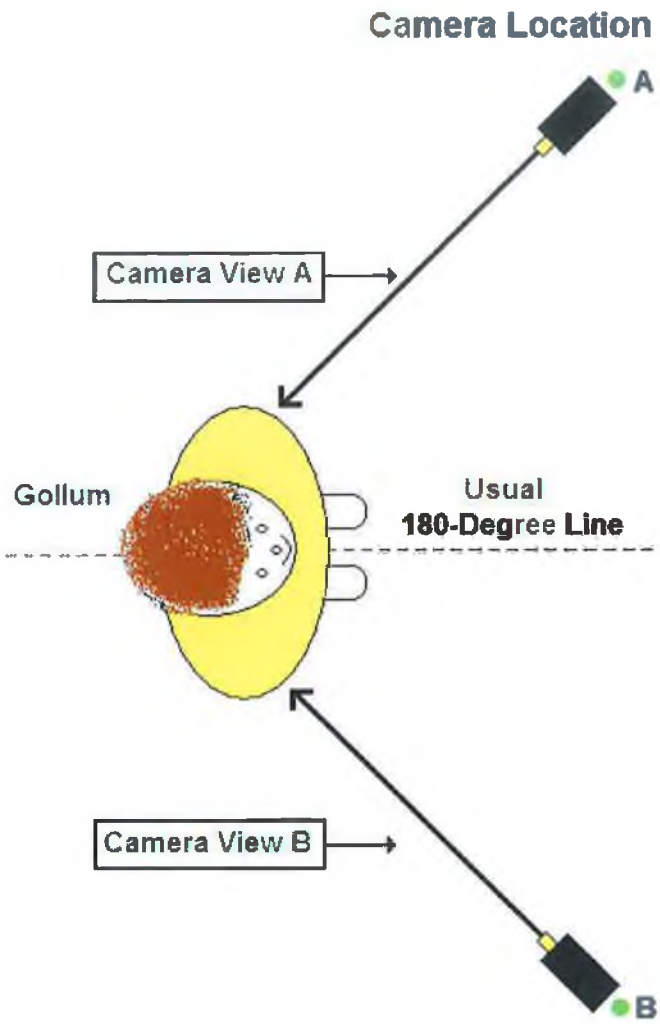

Figure 2.4: Example of breaking the 180^0 line rule

line also ensures that there is a high amount of shot repetition in a dialogue event. This is essential in maintaining viewers' concentration in the dialogue, as if the camera angle changed in subsequent shots, then a new background would be presented to the audience in each shot. This means that the viewers have new information to assimilate for every shot and may become distracted. In general, the less peripheral information shown to a viewer during a dialogue, the more they can concentrate on the words being spoken. It is proposed that the universal use of this rule can be taken advantage of when detecting dialogue events in movies. The presence of large amounts of shot repetition is usually a good indicator that some sort of interaction between characters is taking place. See Chapter 4 for a more in depth discussion on how this concept is leveraged in the proposed system.

The director must also choose how near/far the camera is from the object on screen. The range of possible distances generally ranges from the *extreme close-up* to the *extreme long shot*. The extreme close up typically only has a portion of a persons face on screen, for example a close up on a mouth. An extreme long shot shows a whole person from a distance. In between these two extremes there a number of intermediate steps, such as *close-up* (just the head), *medium shot* (from the waist up), and *long shot* (just about a whole person). The camera distance is chosen based on a number of parameters. If there is movement in the shot then a close up may be unsuitable, as any movement will be hard to follow by a camera. In another scene, a close camera may be required in order to garner an actors reaction etc.

A director may also adjust the height of the camera slightly during a shot. This may occur if one actor is significantly taller than another, for example. Also, the camera may be placed above or below a character at an angle. Having a camera that is looking up to a character gives the appearance of dominance, while if the camera is looking down on a character they appear subordinate.

Much of the director's work comes under the title of "Mise-en-scène". This corresponds to the lighting, setting, camera use and composition that produce the dramatic image on a film. In controlling the Mise-en-scène the director stages the event for the camera [51]. The director must carefully plan every aspect of Mise-en-scène to ensure that the vision in his head becomes the one created in the studio. The use of colour and lighting in cinema is a fundamental tool that every director uses. Incorrect use of colour and lighting can lead to confusion and lack of interest from viewers, but when used correctly the director can influence the way people interpret a scene. For example, it is possible to create a depressing mood by using dark, drab colours as backgrounds and costumes. In the film world, prisons tend to be dark, harsh places and therefore must have dark, harsh colours associated with them. It makes no difference if real prisons have similar colour schemes or not, the viewers interest is far more important to the director than whether the colour he uses is completely accurate. Also lighter and darker areas within

a frame help to create the overall composition of each shot and thus guide the viewers attention to certain objects and actions. A brightly illuminated patch may draw attention to a key gesture, while a shadow may hide something to add suspense [51]. The brightly illuminated patch is not there by coincidence; the director must plan the shot so that the lighting guides the viewers' attention to the desired location. Throughout a single event, there must be consistency of lighting and setting. If an event is being shot outdoors, the shooting will often stop if the light fades as this will result in non-continuous lighting. This characteristic is used in the proposed system to detect changes in the focus of the movie, as can be seen in Chapter 3.

Classical Hollywood has developed the custom of using three light sources per shot (three-point lighting). The three sources are key light, back light and fill light. This means that the main focus of attention should be well illuminated (this can be seen in Figure 2.5). The *Colour* refers to the colour of the light source. Generally film-makers use as pure a white light as possible, and add filters to colour it. E.g. if the candle in the shot is (as perceived by the viewer) the main source of light, then the fill/key/back lights may have an orange tint. Similarly at night a purplish-blue lighting may be used. All of this should mean that the main object in a shot (or at least the object the director wants the audience to focus on) should be well lit and clear for the audience to see. Of course the director may use different lighting to take the viewers attention away from the main object if he/she desires [51]. This use of mise-en-scène helps the audience focus on certain parts of the screen. Typically the colour scheme and setting will remain constant while the movie is in the same location. This ensures that shots in the same events usually have a consistent colour scheme. This fact is exploited when looking for areas where the movie moves from one location/event to another. This is further explained in Chapter 4.

The director plans the camera and character movement in a scene. It was previously mentioned that in order to relax viewers, a slow, repetitive shooting style can be utilised, containing temporally long shots, repeating backgrounds and little movement of cameras. However, in order to create excitement the director does the opposite. Generally the more the viewer is bombarded with short, high motion shots that are visually different, the more excited he/she becomes. Movement of characters and camera in a shot causes the viewer to increase their level of concentration to follow the activities on screen. If the viewer is bombarded with new fast-paced video, and there is too much of this activity to interpret at the speed in which it is being presented, then excitement is generated. For example, in a fight event the camera will move about following the characters as they move around the set. There will also be a large number of shot cuts, each bringing the camera to a different angle with a new background, and therefore new information, being presented to the viewer. The camera may also zoom in and out. This is too much information for any person to follow, and so results in excitement and tension. Examples of how this may

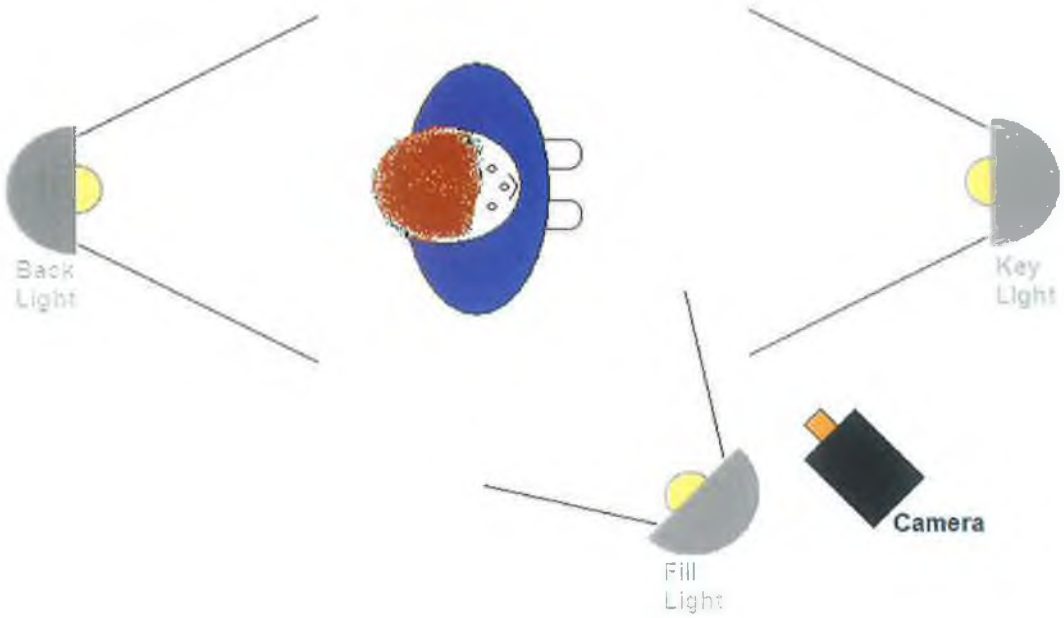

Figure 2.5: Three-Point Lighting

be utilised when shooting an exciting event are given in Section 2.3. If the presence of fast paced editing and high amounts of motion is detected, then it gives a very good indication as to the intentions of the film maker.

2.2.2 The Role of the Editor

Camera work and editing collectively attempt to mimic the way an observer's ears, eyes, and psychological focus migrate within an environment. Whilst the director has an obligation to the viewer, he is also expected to put his own 'stamp' on the film. The best editors are invisible to the casual observer. However, a poorly edited film will lack structure, pace and continuity. Generally, editors work toward the end of the film creation process and have minimal say as to how the scenes are shot, although often directors will seek an editor's input before finishing shooting. Editors also have to make the film as watchable for the viewer as possible; this is achieved through continuity of shots and by creating a natural flow to the film.

Generally speaking, editors have less freedom of expression than directors. This means that they must follow conventions more so than directors. However, there is still considerable room for an editor to be creative. There are a number of qualities that an edit (e.g. a cut/dissolve) must have. First of all there must be *motivation* for the edit. For example, if an actor hears a noise and looks around there is sufficient motivation to cut to see what he is looking at. Secondly, each edit should convey *information* to the viewer. A new shot should mean there is new information for the audience to digest (e.g. if the

actor hears a noise and looks around, the next shot will contain new information if the source of the noise is shown). The more visual information a viewer has and understands, the more informed and involved he/she becomes. (e.g. an editor would seldom cut from a middle shot of a person to a close up of the same person as there is no new information in the second shot). Sound is also essential in maintaining flow between shots. For example, in an office setting, the sound of typewriters clacking or phones ringing must be present over all shots. If, for example an editor went from a long shot of a character with the background noises audible to a close up of the character without the noises, this would startle and confuse audiences [53].

Each time an edit is made, the editor must choose how he/she is going to do so. The hard cut (or simply, the cut) is by far the most common edit used, but an editor may also choose to put in a dissolve, fade or wipe. A dissolve or fade typically indicates that time has passed. Within a scene the edit of choice tends to be the cut, e.g. in a conversation between two people, when the camera changes to go from one person to the other there is usually a cut. If an audience saw for example, a dissolve between characters, this would unsettle them and possibly cause confusion (has time elapsed? Are they in a different location?). There is no logical reason why a dissolve should mean some time has passed, but it has become convention to interpret it this way. Generally, due to common editing practises, audiences have come to accept that a dissolve indicates a brief time lapse or a change in location, while a fadeout indicates a longer time lapse. To summarise, a hard cut is used when the action is continuous, where there needs to be a change of view, or where there is a change of information or location. A dissolve is used where there is a change in time, when time needs to be slowed down, and where there is a change in location. The fade in/out is used at the start/end of a film, at the start/end of a scene, at a change in time, or at a change in location. When looking for shot transitions, hard shot cuts are by far the most prominent and therefore the most important. For this reason hard cuts are targeted as the source of shot boundaries. See Section 3.4.1 for a description of the shot cut method used in this research, as well as a discussion on other methods.

For editing to look smooth and seamless the best place to cut is always on strong physical movement. Editors use any pronounced movement in the scene as a cutting point between angles. Generally, a little movement in the first shot (just before the cut), followed by the bulk of the movement in the incoming shot aids the seamless transition between shots. For example a common editing point in a dinner scene is when somebody raises their glass to propose a toast. Editors will often ask actors to make some movement purely for the purpose of making a smoother edit (fixing their tie, scratching their ear, tilting their head etc.). Also, if there is fast action in the scenes, it may be necessary to actually repeat some of the movement of the shot just before the shot cut in the incoming shot. This is due to the fact that the eye doesn't register the first three or four frames of an

incoming image.

Pace is a very important aspect of editing. Pace is the rate of shot cuts at any particular time in the movie. Although there are no 'rules' regarding the use of pace, the pace of the action dictates the viewers' attention to it. In an action scene, the pace would quicken to tell the viewers that it is important. As pace is usually quite fast during action sequences, it is therefore more noticeable, but it should be present in all sequences. For example, in a conversation that intensifies toward the end, the pace would quicken to illustrate the increase in excitement. Faster pacing suggests intensity, while slower pacing suggests the opposite. The pace of a sequence of shots is used to assist in the detection of events, as described in Chapter 4.

The duration of the shot should be determined by how much attention it demands. This is not to be confused with creating short shots for action scenes. For example, if the reaction of character A is more important than that of character B, then this should be represented by showing character A on screen for a longer duration. Similarly, a shot of a location should be kept for longer on the screen to ensure the audience has enough time to take all the information in.

Through the years, audiences have come to expect films to be edited in a certain way. This is largely due to the fact that previous editors used these techniques and simply passed them on. Current editors do not have the luxury of picking their own conventions as they have inherited the methods of their predecessors. For example, in a three person dialogue, convention dictates that editors should never cut from a two person shot to another two person shot. Also, when cutting a 'rise' (somebody standing up), an editor should try to keep the subject's eyes in frame as long as possible. There are many more subtle editing rules such as these, which again illustrate how much the film-making process is dictated by convention.

The fact that there are so many editing rules and conventions suggests that an editor is more restricted than a director in creating a movie. However, there is still room for innovation in the editing of any movie. In Section 2.3, there are some examples of how the correct and skillful use of editing, when combined with directing and sound production, create a coherent movie.

2.2.3 The Role of the Sound Engineer

It seems to be a matter of debate how important sound is in the overall make up of a film. Some film-makers argue that sound is merely there to accompany the visuals; after all we talk of "watching" films and refer to ourselves as "viewers" [51]. Although these film-makers don't deny that sound has an effect on the movie-watching experience, they simply prefer to focus more of their attention on the visual elements of the film. Another school of thought puts sound on a par with the visuals, seeing it as an essential tool in

creating excitement and setting tone throughout a movie. Sound is also one of the main ways of conveying information to the viewer. It is usually possible to obtain a good idea of what is taking place in a film with just the sound and no visuals, although the reverse is not always true. However much importance a film-maker places on the audio track, there are still many elements to sound that must be brought into consideration in order to make a coherent film.

In many ways sound can be similar to editing - frequently it is called upon to do its job yet remain unnoticeable. The sound editor's job is to make the sound in a movie realistic, or at least meet the expectations of the viewer. For example if a viewer sees a car door slam shut, then he/she expects to hear the corresponding sound synchronised with the visuals. If there is no sound, the viewer may become confused. This sound may be recorded live or possibly added after the filming has been completed by taking the required sound from a sound effects library. There is a loose correlation between the visuals seen and the sounds heard. For example, in a two way conversation using shot/reverse shot camera angles, there often is a dialogue overlap between shots, i.e. a shot of person A while person B is talking. This is used to smooth the abrupt visual change of the shot.

Sound in the cinema can be grouped into three categories, *Speech*, *Music* and *Sound effects*. Narration, as in a voice over, is sometimes given its own category, but is more usually classified as speech [51]. Any speech that is recorded must be synchronised with the visuals. Usually speech is recorded with the visuals to ensure this lip-sync occurs, but it may be recorded at a later date. Usually speech is given priority over the other forms of sound as this is deemed to give the most information and thus should not have to compete for the viewer's attention. If there are sound effects or music at the same time as speech, then they should be at a low enough level so that the viewer can hear the speech clearly. To do this, sound editors may sometimes have to 'cheat'. For example, in a noisy factory the sounds of the machines, that would normally drown out any speech, could be lowered to an acceptable level. A common example of this sound manipulation is in a conversation set beside a busy road. A very long shot may show the road and the actors with loud sounds of cars and trucks passing by. When the camera moves to a middle shot of one of the actors the sound of the cars is reduced, even though logically, since it is nearer, the volume should increase. Conversely, if a character speaking was being filmed, and the camera moves from a close up shot to a long shot, the sound level of the actors dialogue would remain constant, even though this is totally unrealistic [51]. Where speech is present, and is important to the viewer, it should be clearly audible. Thus, when analysing the audio track the detection of speech is quite important, as it gives an insight as to what is taking place in the movie. The detection of speech forms an important part of the analysis work in this research.

Music in films is usually used to set the scene, and also to arouse certain emotions

in the viewers. The musical score tells the audience what they should be feeling. In fact, in many Hollywood studios they have musical libraries catalogued by emotion, so when creating a soundtrack for say, a funeral, a sound engineer will look at the 'sad' music library. Sound effects are usually central to action sequences, while music usually dominates dance scenes, transitional sequences, montages, and emotion laden moments without dialogue [51]. This categorisation of the sounds in movies is quite important in this research, as much of the audio analysis is based on this. The effect of music on the viewer is usually quite subtle, and may go unnoticed in some respects, but a sound editor knows it is vital to have a good musical score to accompany the film. Picture a romantic scene, say a number of shots showing a couple having a romantic dinner. They sit at a candle lit table in an otherwise darkened room in a nice restaurant, a waiter is seen pouring them wine, they are talking closely, the conversation cannot be heard but they are both deeply interested in each others conversation. Now try to think what kind of music would be suitable - the choice is fairly limited. Possibly a musical piece with slow violins or piano, or even a slow lounge track. A fast tempo jazz track would certainly not fit the setting. An unsuitable choice of music could potentially give the scene a different meaning. A badly chosen track can sometimes give a viewer the incorrect impression, as viewers expect to hear certain music with certain situations. Music in film has a very direct role in influencing the viewer. Thus, knowledge about the presence or lack of music in a movie is important for extracting knowledge about the activities on screen. For this reason, the detection of the presence of music forms an important part of this research.

In the world of sound effects creation, viewer interest and understanding is the main goal, and realism may suffer if this aim is jeopardised. Sound effects should usually reflect what the viewer expects to hear, even if it is blatantly unrealistic. For example in a shot of people watching fireworks, the sound of the 'BANG' may occur at the same time as the visual explosion is shown on screen, even though there would realistically be a sound delay.

The *fidelity* of the sound is used to define how realistic it is. For example if a dog is seen barking on screen, and a dog barking sound is heard, the sound is said to be of good fidelity. In the examples given above of fireworks and people talking, the fidelity is somewhat lower, but it is still what the viewer expects to hear and, again, the viewers interpretation is more important than realism. Sometimes, the fidelity will be deliberately lowered by using incorrect sound effects, usually for comic effect. This is widespread in children's films, especially cartoons. For example if two people run into each other there may be a 'boiiing' sound. For the most part though, sound effects need to be a realistic representation of what is expected on the screen.

When the various sound categories come together, they make a complete sound track for a film. They must fuse together to create an audio track that conveys information to

the viewer, keeps him/her interested, maintains realism, and subtly suggests emotions and feelings. Of course sound doesn't just have to be present to ensure the viewers hear what they think they should hear, it is a creative medium that can be exploited for powerful effects. Horror films for example, make good use of sound effects and music to startle and shock viewers - a groan heard from another room, the sound of footsteps, a sudden loud burst of music, etc. Sound can also direct viewer's attention to certain parts of the screen, for example if a creaking door is heard a viewer will immediately look to the door. This is especially true since surround sound has become commonplace in cinemas, as it is possible to hear sounds coming from the left/right/behind.

Many sound editors use a sound overlap between scenes to ease the viewers' attention into the next scene. This is usually done with speech or music. It basically involves overlapping the audio from the end of one scene to play in the first few seconds of the next. This guides and maintains the viewers interest to the next scene. Thus, it is not ideal to use a change in sound as a good indication of a change in scene etc. Overall, it is the sound editor's job to ensure that the audience feels comfortable with the audio, and can easily interpret the sounds played to them. The analysis of the audio track is essential to this research, and there are many occasions in the following chapters where this becomes apparent.

2.3 Examples of Film Creation Techniques

A film needs input from many people in order to be successful. It was illustrated how the three main aspects (directing, editing and sound engineering) of movie creation work individually. Some illustrations of how they combine to produce an effect greater than the sum of each individual part is provided in this section.

The first example is a conversation from the film 'American Beauty'. The first ten shots of the conversation are analysed (shown in Figure 2.6). In this event, the character Lester (shots 2, 4, 6, 8,and 10), is being told by his boss, Brad (shots 1, 3, 5, 7, and 9), to write a report about what his job entails. Lester fears that he might lose his job as the company is downsizing. Notice in the first shot of Brad sitting behind his desk (shot 1) that the camera is placed slightly below him. This indicates that he is in a powerful position, as he looks dominant on the screen. Also notice the way in which Lester is shown in his first shot (shot 2). Firstly, the camera is much further away from him than the camera shooting Brad. This emphasises the distance between the pair (in terms of opinion). Also, the camera is slightly above Lester, emphasising his powerlessness. In Shot 3, Brad stands up, again reiterating his position of power. Toward the end of shot 4 Lester becomes confrontational and begins to argue with Brad. Notice the difference in camera placement between shot 6 and shot 4. As Lester becomes more animated and

Figure 2.6: Dialogue Example from *American Beauty*

hostile the camera cuts to become closer to him. Moving a camera closer to people as a conversation intensifies is a common shooting technique. In shot 7, Brad is shown to sit down. This shows how he is being reduced in stature by Lesters argument. Now that the camera has been moved closer to Lester, it is possible to see his reactions more closely. Lester does not say anything in shot 8, he is only seen to shrug his shoulders in disgust at Brad. Had the camera been in the same position as it was in shot 2 his reaction may not have been noticed. Finally in shot 10, as the conversation reaches its climax, Lester has leaned forward to come even closer to the camera. This again, illustrates the increased intensity of the dialogue.

The editing style of this dialogue event is typical of the way many dialogues are shot. The cut type is always a hard cut. Typically the cuts are made in conjunction with some small amount of movement of the characters. For example, many cuts occur as the actor on screen moves his hands, shrugs his shoulders, stands up etc. In many shot cuts, there is an overlap between the speech and the cut. For example, after shouting at Brad, Lester's reaction remains on screen as Brad speaks, trying to justify his position. In this case, the editor deemed Lesters reaction to be of more importance than Brad's speech. Also, in another reaction shot (shot 8) Brad speaks while Lester sits in his seat shrugging his shoulders.

Figure 2.7: The Initial Dialogue Event From the *Reservoir Dogs* Example. 7 Shots, Taking Almost 4 Minutes

There is quite a lot of shot repetition in this conversation, as shots 1,3,5,7 and 9 (all of Brad) are shot from the same camera. The shots of Lester are split into two groups, shots 2 and 4 are shot with one camera angle, while shots 6, 8, and 10 are shot with a different angle. In terms of how the sound is used in this conversation, there are no other sounds competing with the speech and there is no music in the background. This is to ensure that viewers can fully concentrate on the dialogue.

The next example comes from the film Reservoir Dogs. This example shows how a director and editor can combine to excite and surprise the viewer. In this example, set in a warehouse, there is a two person conversation followed by a fight, which is in turn followed by a three person conversation. This is an example of how a single scene can contain multiple events (two conversations and a fight). In the first conversation (which lasts 7 shots), there is a very relaxed editing style despite the heated nature of the conversation. The 7 shots take up a total time of almost four minutes, and the average shot length is 33 seconds. Although in general this film contains a lot of temporally long shots, this is considered very slow paced. These shots can be seen Figure 2.7. The camera moves around during some of these shots, panning and tracking to follow the characters as they move around the warehouse, although the camera movement is quite gradual.

The two characters then begin physically fighting. Immediately the pace of the scene changes as shot length decreases, and movement of both characters and cameras increase. Altogether there are 9 shots in this event, yet the fight only lasts for 15 seconds, giving an average shot length of 1.7 seconds (this can be seen in the first 9 shots in Figure 2.8). This is in sharp contrast with the preceding dialogue event. There is a vastly increased amount of camera movement throughout the fight. Many of the shots are close ups of the action, which amplifies the movement of the actors (shots 1, 2, 4, 5 and 6 all show

Figure 2.8: The Exciting Event From the *Reservoir Dogs* Example and the First Two Shots of the Following Conversation. There are 9 Shots in the Fight, Taking a Total of 15 Seconds

intense movement with a camera close to the actor). Due to the decreased shot length and increased motion, a lot of information is displayed on screen in a very short space of time. This is more information than it is possible to comprehend, so the viewer becomes over-awed, leading to excitement. Increased pace and movement is a common characteristic of exciting/action events. As the shots immediately before the start of the fight are long and drawn out, the sudden burst of activity is even more noticeable, and the impact on the viewers is maximised.

After the fight has finished, another conversation takes place. This involves the two people that had previously been fighting, and a new character. After the intensity of the fight, the pace slows somewhat to a more comfortable rate for the viewer. The first fifteen shots of this dialogue event have an average length of 6.4 seconds. This is a typical shot length for many conversations. The camera is still for most of these shots, only moving to track character movement. All of these elements work together to calm the viewer and re-introduce them to the conversation. Shots 10 and 11 of Figure 2.8 are the first two shots of this conversation. This example shows how pace/movement can be used to effect the film and the viewers. By detecting areas of high motion it is possible to deduce meaning

Figure 2.9: An Example of Sound Usage from *Amores Perros*

of the film-makers intentions (see Chapter 4).

The final example chosen to illustrate film production techniques comes from a Mexican made film entitled *Amores Perros*. Figure 2.9 shows a number of selected shots taken from one particular event in the film[1]. In this sequence of shots, the woman (Valeria) is driving to the shops to buy some food. She is in a happy, upbeat mood as she has just moved into a new apartment. Music can be heard, and it seems to be coming from the car radio. It is fast and upbeat music, designed to show the mood of Valeria. Interestingly, the music can also be heard in the previous shots (not shown) of her saying goodbye to her boyfriend in their apartment. The fact that the music is coming from the car is emphasised as it becomes louder while she is driving. Thus, there is a lack of fidelity in the music, as, logically, one would not expect the music in the car to continue from the music in the apartment. This demonstrates how sound engineers usually keep the audio constant for a single event. This can be exploited when analysing the audio for event detection by searching for portions of a film with constant audio (Section 4.2.2). The first number of shots (from image 1, 2, 3 and 4) show Valeria and her dog driving happily in the car.

[1]Although they are chronologically displayed, a number of shots were skipped (i.e the length of the event is greater than eleven shots)

Valeria then encounters a red light and slows down. In image 6 she is seen looking in the mirror and reaching for her lipstick. Next the camera changes position to be placed outside the car, looking through the windscreen, as Valeria applies make-up (image 7). The sound volume drops while the camera is placed outside the car to further re-emphasise that the source of the sound is the inside of the car.

As Valeria finished applying her lipstick, the traffic light turns green. In image 9 her car can be seen driving off. However, later on in the same shot, as she is driving through the junction, her car is hit by another car (image 10). The up-beat music that was playing stops the instant the car is hit, and a loud crashing noise is heard. This is immediately followed by the noise of breaking glass, tyres screeching, a car horn (which keeps sounding) and nearby dogs barking. The contrast between the up-beat music and the 'thud' of the two cars colliding is striking, and has the intention of startling the audience. The picture then fades to black (image 11) as the next scene starts, but the sound of the horn and of the burning cars continues despite the lack of visuals. The audio of the next scene (set in a hospital, with the sound of an intercom), is gradually faded in while the black screen remains.

This example illustrates a number of interesting audio concepts. The fidelity of the audio varies throughout the example. Initially it is quite low, as the music from the previous scene is continued despite the fact that it is implied that the music is coming from inside the car (which did not appear in the previous scene). However the fidelity increases to the point where the sound editor lowered the music volume when the camera was placed outside the car. The music stops as soon as the car is hit, possibly explained by the radio breaking, and is replaced by the noise of the accident. This is necessary to further emphasise the point that the happiness is over. The accident noises would have been put in after filming by the sound engineer, and would be taken from a large library of sound effects. The sound throughout the example was chosen by the film-makers to firstly, create a happy atmosphere and show the upbeat mood of Valeria, and secondly, startle the viewer as the horrific car crash takes place. This illustrates how, when used correctly, sound can have a strong influence on the audience.

Each of the three examples chosen in this section illustrate how directing, editing and sound production combine to commit a film-makers vision to film. The techniques mentioned, although only a subset of the available techniques, demonstrate that film-makers have specific tools with which to influence the audience. Some of these tools can be used very creatively, film-making is, after all, a creative medium. However, a great deal of these tools are restrained by common practise and convention. It is by exploiting these conventions, while allowing for creativity, that movie data is analysed.

2.3.1 Discussion

It is clear that film-making is a highly creative process with a large amount of possibilities available to directors, editors and sound engineers. Indeed, the aforementioned film-making methods are only a subset of the possible range of film-making techniques. Within the set of principles outlined, there are a number of film-making techniques that, for various reasons, are not directly used in this research. For example, it was previously mentioned that a director may position a camera slightly below/above a character in order to show dominance or suppression. However, this is quite difficult to detect using visual analysis techniques and also, in terms of event detection, knowledge of character dominance is not of much benefit. Similarly, although directors may use bright colours to indicate a happy environment and drab colours to indicate the opposite, this is also not useful in the context of event detection (although it may be useful in order to further classify events once they are detected).

Numerous editing principles were presented in Section 2.2.2 that are not used in this research. It was stated that there must be motivation for each edit, and that each edit should convey information to the viewer. These are quite abstract concepts and are extremely difficult to infer using audio-visual analysis. Also, the fact that an editor may prefer movement in frame directly before an edit in order to make the transition seamless is not of much concern in this work.

There are also many sound engineering principles mentioned in Section 2.2.3 that cannot be used in this research. For example, it is extremely difficult to ascertain the fidelity of the sound in a movie from the audio track and again, in the context of detecting events knowledge of the fidelity has limited use. Also, due to the near infinite amount of possible sound effects in a movie, detecting anything but a small subset of these sounds is quite difficult.

However, a number of these film-making principles can be leveraged in order to assist in the event detection process. Knowledge about camera placement (and specifically the 180^0 line rule) can be used to infer which shots belong together as they form part of the same event. Repeating shots, again due to the 180^0 line rule, can also indicate that some form of interaction is taking place between multiple characters. Also, the fact that lighting and colour typically remain consistent throughout an event can be utilised, as when this colour changes it is a strong indication that an new event (in a different location) has begun.

The use of camera movement can also indicate the intentions of the film-makers. Generally, low amounts of camera movement indicate relaxed activities on screen. Conversely, high amounts of camera movement indicate that something exciting is occurring. This also applies to movement within the screen, as a high amount of object movement may indicate some sort of exciting event. Thus, when detection events, the amount of mo-

tion present is examined. Another factor that can be used to differentiate between relaxed and exciting parts of a movie is the editing pace. Faster pacing suggests intensity, while slower pacing suggests the opposite. Again, the shot lengths are used as an indication of a film-makers intent.

Finally, the type of audio present at any point in a movie is used in order to assist in the event detection process. For example, the presence of speech is a reliable indicator not only that there is a person talking on-screen, but also that that person's speech warrants the audience's attention. Similarly, the presence of music and/or silence could indicate that some sort of musical, or emotional, event is taking place.

2.4 Leveraging Film Grammar in Designing an Approach to Movie Summarisation

2.4.1 Choice of Event Classes

The advantages of an event based browsing/summarisation system were previously mentioned. However, the event classes need to be carefully chosen. The event classes should be plentiful enough to cover all of the meaningful parts in a movie, yet be generic enough so that only a low amount of event classes are required. This is to ensure that the index is as compact and user efficient as possible. It may be possible to define a large number of event classes, and attempt to implement detectors for each event class. However, due to the near infinite range of possible events in movies this would soon become an impossible task. It is proposed to create a reasonable number of event classes, some of which may encompass a number of different events, in order to create a reliable index. Each of the events in any event class have a common semantic thread that link the events together. This allows intuitive navigation through a movie. However, the selection, and amount, of event classes is dictated by how the films themselves are created.

It is proposed here that three classes are sufficient, corresponding to dialogue, exciting and montage events. Dialogue constitutes a major part of any film, and the viewer can often get the most information about the plot, story, background etc. of the film from the dialogue. Dialogue events should not be constrained to a set number of characters (i.e. 2-person dialogues), so a conversation between any amount of characters is classed as a *dialogue event*. Dialogue events also include events such as a person addressing a crowd, or a teacher addressing a class.

The second event class is *Exciting* events (sometimes referred to as *Action* events). These typically occur less frequently than dialogue events, but are central to many movies. Examples of exciting events include fights, car chases, battles etc. Whilst a dialogue event can be clearly defined as there are people talking, an exciting event is far more

subjective. Most exciting events are easily declared (a fight for example, would be labelled as 'exciting' by almost anyone watching), but others are more open to user interpretation. Should a heated debate be classed as a dialogue event or an exciting event? As mentioned in Section 2.2, film-makers have a set of tools available to create excitement. It can be assumed that if the director wants the user to be excited, then he/she will use these tools. Thus, it is impossible to say that every heated debate should be labelled as 'dialogue' or as 'exciting', as this largely depends on the aims of the director. The only way to class these events is to actually view them. Thus, a large emphasis was placed on detecting exciting events that users specified as exciting, rather than a generic list of scenarios. Thus, there can be no clear definition of an exciting event, other than a sequence of shots that makes a viewer excited.

The final event class is a superset of a number of different events, and they are labelled as *Montage* events. The first type of event in this superset are montage events themselves. A montage in a film is a juxtaposition of shots that typically spans both space and time. A montage usually leads a viewer to infer meaning from it based on the context of the shots. As a montage brings a number of unrelated shots together, typically there is a musical accompaniment that spans all of the shots. For example, a montage could aim to illustrate the futility of war by showing a number of shots of a dying soldier, followed by shots of the dying soldiers widow receiving the news that her husband was killed in battle, and finally a number of shots of a young soldier nervously preparing for battle. Although when viewed individually the three parts of this montage may not convey much meaning to a viewer, when all three parts are viewed together, the negative effect of war is illustrated. It shows that not only are the soldiers effected, but also their families. Also, as the young soldier is about to go into battle, he could meet a similar fate as the dying soldier, which again may effect the audiences view. The second event type labelled in the montage superset is an *emotional* event. Examples of this are shots of somebody crying, or a romantic sequence of shots. Emotional events and montages are strongly linked as many montages have strong emotional subtexts. The final event type in the montage class are *Musical* events. A live song, or a musician playing at a funeral are examples of musical events. These typically occur quite infrequently in most movies. These three event types are linked by the common thread of having a strong musical background, or a least a non-speech audio track. All of the montage, emotional and musical events come under the common umbrella of the 'Montage' event class. Any future reference to montage events in this thesis is referring to the entire set of events labelled as montages.

There are other occurrences throughout a movie, however, the three event classes explained above aim to cover all meaningful parts of a movie. Although part of a movie, the beginning and ending credits are not considered relevant when creating an index of the movie (unless, of course, an event is taking place whilst the credits are displayed).

Firstly, on their own, the credits do not contain any meaningful information in terms of advancing the story of the movie onward. Secondly, the location of the credits is known at either the beginning or ending of the movie, and therefore can be easily found whether they are placed in an index or not. One other element of a movie that is not considered for an event based index is a voiceover. Voiceovers occur when there is some narration over the movie. Most movies do not contain voiceovers, and in the movies that do, they are typically only present in order to set the context of the following event. Thus, they are not deemed to be meaningful events in this work.

The distinction between the three event classes is quite subjective. One person may feel that a particular event belongs to the a certain class, while another feels it belongs to a different class. An emotional conversation, for example, could be labelled as as dialogue event by one person or a montage event by another, as it contains elements belonging to both event classes. Indeed, there are many other types of events that could be classed differently depending on the user. An argument, for example, could be labelled as an exciting event by one user, and as a dialogue event by another. Many emotional (montage) events also aim to excite the viewer, and therefore could also be classed as exciting events or montage events depending on the user. Thus, when detecting events, a level of flexibility is required so that users of any system employing the event based index with differing opinions can still locate their sought events. This means classifying events into more than one event class, so that each user can easily locate the event. For example, classifying an emotional conversation into both the dialogue event class, and the montage event class. Clearly, if manageable content is required, the events should be placed into as few event classes as possible, however there is a fuzzy boundary between each class, and therefore dual classification of some events is necessary. This aspect is explained further in Chapter 5 where the degree of overlap of detected events with different user interpretations is examined.

In movies, meaningful events span a number of shots. Any event that a director wants the user to digest is given sufficient time on screen to do so. Therefore, it is defined that any meaningful event must be at least a number of shots long. This minimum length is to discount the unimportant and meaningless parts of a movie. There will always be unimportant events within a movie, shots that do not fall into the class of an 'event', a passing 'Hello' between two people on a corridor does not qualify as a dialogue event for example. In this work five shots was chosen as the minimum event length after examining events from a number of movies. In the case of dialogues and montages, it is straightforward to expect a director to allow enough time for a viewer to digest the sequence of shots, and to extract some understanding from them. Thus, if dialogue or montage events are in any way important to the viewer, the director will allocate a sufficient amount of shots to them. Many exciting events also contain a high amount of shots with which the

director uses to sustain excitement. However, sometimes the director will want to shock or surprise the viewer by showing a short burst of high tempo shots in rapid succession. However, these short bursts will still contain a relatively high amount of shots (i.e. more than five shots) as the shot cut rate will need to increase in order to create the excitement for the viewer. Therefore, even if the exciting event is temporally short, the minimum five shot rule will still hold.

2.4.2 Feature Selection

In this work, audiovisual analysis is utilised in order to extract information about the occurrences in a movie, and ultimately to detect the events. It is proposed in this thesis that there are numerous occasions where the presence of certain features can indicate a film maker's intent. The detection and analysis of these features should allow for extraction of knowledge about the events in the movie. A colour representation of the video is desirable as this allows for shot matching and comparison. For example, knowing which shots have a highly similar colour value, and are therefore likely to be shot using the same camera angle, is highly beneficial as it indicates that the shots are related. Thus, the motivation in selecting the colour features used arose out of a need to accurately describe the colour values, and therefore allow direct comparison between different frames.

Another reliable indicator of the activities in a movie is the amount and type of motion present. High amounts of motion are usually an indication that the director is trying to create excitement, especially if combined with an increased editing pace. Similarly, a still camera with little or no movement within the frame may indicate that the director is trying to relax viewers. Thus, the features chosen aim to accurately detect and describe the amount, and type, of motion present throughout a film. Also, the shot length (which indicates the editing pace) is an important feature as it is another indicator as to the intentions of the editor.

Similarly, the type of audio in the sound track can give insight into the events taking place. It was noted in Section 2.2.3 that the sound engineer goes to great lengths to ensure that the audience can clearly hear each relevant part of a movie. If speech is present, then it should be clearly audible, and not drowned out by music or sound effects. When speech is present, it can be deduced that it is relevant to the movie. Similarly, the presence of silence or music could indicate that some sort of emotional event is taking place. Accurate classification of the audio track is therefore important, if the film-makers intent is to be inferred. The audio features used were chosen to facilitate this classification prior to event detection.

The features were chosen in order to garner specific information about the movie. As detailed above, each feature was chosen with a particular application in mind. There are, of course, many other methods of extracting features from audiovisual data. Much work

is being done in the field of object matching and recognition, for example. Approaches such as [54] or [55] could recognise sample objects like cars, aeroplanes, or horses in an image. Although it would be beneficial to know when certain objects are present (for example detecting the presence of a gun could indicate an exciting event), movie data is extremely challenging and it is not thought that the existing state of the art would accurately detect the required objects in a given movie. There are also a number of visual analysis techniques that aim to classify a setting. It may be possible to visually analyse the movie data and determine whether it is set indoors, outdoors, in a city, in a natural setting etc. However, knowledge of the location of an event would be of limited use in an event *detection* scenario. Once events are detected, it may be useful to further classify them based on these characteristics (for example, "this exciting event is set in a park").

Face detection may prove to be particularly useful, as it would indicate the presence of a person talking on screen. However, given the challenging nature of movie visual data, it is thought that current audio classification techniques are more accurate at detecting speech than face detection approaches are at reliably detecting faces. Typically, face detection algorithms have difficulty in detecting faces in side view, or partially occluded faces, which are commonly present in movies. Also, in animated movies the face detection techniques may not be able to detect any faces. The presence of speech is a reliable indication that somebody is talking on screen.

Text data may also be useful in certain scenarios. There have been approaches reported that aim to extract knowledge about the content based on the text data present in DVD releases of a movie (the subtitles, or audio descriptions for the blind). For example, Salway et al. [56, 57] define the emotion at particular points in a movie by detecting relevant keywords. Similarly, many video retrieval algorithms are extensively based on text queries and results. Many of the systems in the TRECVid conference[2] [5] use text as their primary source of returning results. Typically, when comparing a text-only based system with a system that incorporates both text and visual features, the text and visual system only marginally improves performance, which indicates the strong reliance on text of many systems. In this research, however, the presence of speech is more important than the actual words spoken. As the detection of the presence of speech is used primarily for dialogue event detection, text information is not required. However, as indicated in the future work section, text information could be used to further classify dialogue events by subject (i.e. dialogue event 1 is about a robbery, dialogue event 2 is about a computer, etc.). This extra layer of information could also prove useful in a search environment, where users could search for particular keywords.

[2]The TRECVid conference sets a number of retrieval tasks on a common set of video data. Results from different video retrieval systems are submitted and can be directly compared

2.4.3 Data Classification Method Selection and Application

Given a set of features, it is necessary to select a data classification tool in order to detect events in a movie. Machine learning and pattern classification are areas of artificial intelligence which develop techniques that allow computers to 'learn'. This involves writing algorithms that can analyse large data sets. This analysis usually leads to some classification of data. Machine learning techniques have been applied to a broad range of fields, from genetics to speech recognition. Typically pattern recognition aims to classify new data based on either a priori knowledge or statistical information generated from other data.

The classifier used in this work receives a set of features that describe the activities in a movie and, based on these inputted features, detects the relevant events. Many data classification techniques build a model based on a provided set of training information in order to make judgements about the current data. Although in any data classification environment there are differences between the training data and data to be classified, due to the varying nature of movies it is particularly difficult to create a reliable training set. The data classifier should be able to detect events with any set of input data, even in the presence of original film-making techniques. Thus, it must be possible to configure the classifier based on the film grammar principles discussed earlier in this chapter, so as not to rely on the film-making style of the examples in the training set.

There have been numerous approaches to video summarisation that utilise binary classifiers, such as support vector machines (SVMs) [42, 58, 27]. Although SVMs are quite successful at separating two classes of data, there are a number of inherent problems in applying them to event detection in this work. Firstly, SVMs have no memory. The current classification is based purely on the current data and not on any previous classification. This is in contrast to how a movie is made, where events typically take up a number of successive shots. Secondly, SVMs rely on training data. This means that each decision an SVM makes is based on a previous set of examples. While this may be desirable in many applications, due to the creativity with which movies are made, a new film-making style may result in incorrect classification (or conversely, a unique film-making style in the training set may result in a poor SVM).

A number of other video summarisation approaches that utilise data classification techniques use a hidden Markov model (HMM) [26, 21, 31, 37]. In the context of this work, HMMs have an advantage over SVMs as they take into account the previous classification state. However, the problem remains that HMMs rely on training data, and if a unique film-making style is present in a movie this may lead to classification errors. Thus HMMs are not deemed suitable.

Finite State Machines (FSMs) were chosen as a data classification technique as they can be configured based on a priori knowledge about the data (in this case movies), do

not require training, and can be used in detecting the presence of events based on the underlying features. This ensures that the data classification method can be tailored for use in movie video data. FSMs are explained further in Section 4.2.1. Although FSMs and HMMs are quite similar in structure, the primary difference is that HMMs require training while FSMs are user designed. However, for completeness, a HMM-based event detection approach is implemented and the results of the two systems are compared in Chapter 5.

Application of Finite State Machines

Although each movie is individually made, there are a number of underlying conventions that must be followed in order to shoot coherent events. It is proposed that the data classification method targets the individual components in order to detect events, rather than targeting the event as a whole. To illustrate why this is deemed necessary, three dialogue events will be compared. It was previously noted that there are a number of conventions used by film-makers when shooting dialogue events. Although not all of these conventions are required to be present in each dialogue, a number of them must be present or else the audience may become confused. Typically dialogue events should contain a relaxed editing and shooting style, with a non-moving camera focused on the people involved in the conversation, clearly audible speech, and a high amount of shot repetition (as the camera alternates between the characters). However, as these three examples show, the presence of all of these conventions cannot be guaranteed.

The first sample dialogue event contains a conversation involving three people sitting in a meeting room. The camera focuses on the people as they speak, and each person speaks a number of times. Thus, in this conversation there is a non-moving camera, audible speech and a high amount of shot repetition. The second dialogue event involves two people, and takes place in a moving car. The camera focuses on the person talking and alternates between the two characters. This conversation contains a moving camera (as the car is moving), audible speech and a high amount of shot repetition. The final conversation again involves two people, in this case standing in a nightclub. Again, the camera focuses on the person talking and alternates between the two characters. However, subtitles are required as the music is considerably louder than the speech. Thus, in this conversation there is a non-moving camera, music and a high amount of shot repetition. When viewed as a whole, each of these dialogue events are quite different. However, they contain a number of common components. In all cases there is shot repetition, for example. Also, two of the events contain speech, and two contain a non-moving camera. Thus, it is proposed when using a data classification method to detect events in movies, it is important to ensure that the *individual components* are used as a basis for classification, rather than the complete event. Thus, the FSMs must detect the presence of the individ-

ual components that are typically present in an event, and then apply some heuristics to determine if enough of these components are present to declare an event.

2.5 Event Detection System Overview

2.5.1 System Requirements

The aim of the system presented in this thesis is to automatically detect events in a movie. In order to detect events, audiovisual analysis, based on film creation techniques, is utilised. This audiovisual analysis involves generating appropriate features, and then applying data classification methods to extract knowledge about the video. As described in Section 2.4.1, there are three event classes used for summarisation: *dialogue* events, *exciting* events, and *montage* events. Although there is some variation between the events within each class (i.e. not all dialogue events are shot in the same way), it has been observed that there are a number of elements that occur frequently within the events in each class. For clarity, the main elements of each event class (as indicated in the previous sections) are re-stated here.

Dialogue events tend to contain high amounts of clearly audible speech. A static camera, fixed on the people talking, is often used by a director when shooting a conversation. Due to this static camera and the 180^0 line rule, dialogue events usually contain high amounts of shot repetition, as the camera focuses on each person repeatedly talking. Thus, the features sought in order to detect dialogue events are: a measure of the camera movement, a measure of the amount of speech, and a measure of the amount of shot repetition. In contrast with dialogue events, *exciting* events tend to contain high amounts of movement. A low amount of shot repetition is also usually present as the camera moves from one location to another. Also, an increased editing pace (a decreased shot length) is typically present in exciting events. Thus, the features sought in order to detect exciting events are: a measure of motion in the video, a measure of the amount of shot repetition, and a measure of the editing pace. Finally, the events in the *montage* class[3] contain non-speech audio (usually music, but occasionally silence), with low amounts of camera motion, and relatively slow paced editing. Thus the features required in order to detect montage events are: a measure of the amount of non-speech audio, a measure of the amount of camera movement, and a measure of the editing pace.

Although not a feature of a particular event class, it is important to detect the boundaries between events, in other words where the focus of the movie changes from one event to another. This could be between two classes (e.g. the point where a dialogue event finishes and a montage event begins), or in the same class (e.g. the point where one

[3]Note that, in this context, the *montage* class contains montage events, emotional events, and musical events.

dialogue event finishes and another dialogue event begins). Thus, a method of detecting these 'changes-of-focus' is also required.

In summary, the features required in order to detect the three event classes in a movie are: a description of the audio content (where the audio is placed into a specific class; speech, music etc.), a measure of the amount of camera movement, a measure of amount of motion in the frame (regardless of camera movement), a measure of the editing pace, and a change-of-focus detector.

2.5.2 System Design

The system used to detect events in movies is shown in Figure 2.10. As can be seen from the diagram, there are three modules in the system. The low-level feature extraction module, the shot-level feature vector generation module, and the high-level event detection module. The 'Low-Level Feature Extraction' module extracts the features necessary for the summarisation process, while the 'Shot-Level Feature Vector Generation' module evaluates these low-level features at a common data unit (i.e. a shot), so that a direct comparison between these data units can be made. Both of these feature extraction modules are explained in Chapter 3. Finally, the 'High-Level Event Detection' module applies data classification techniques to the generated features and creates an index of the dialogue, exciting, and montage events in a movie. This is further explained in Chapter 4.

2.6 Summary

The first portion of this chapter examined a number of film creation techniques. The roles of three key people in the film-making process (the director, editor and sound engineer) were examined, and general film-making principles were described. Sample applications of these principles from three different films were then presented. It was noted that, although film-making is a highly creative medium, many common conventions exist that must be followed as they are now expected by the audience. It is proposed that by detecting the presence of these conventions throughout a movie, it is possible to deduce the intentions of the film-makers, and therefore detect events in a movie. Based on the film grammar principles, a number of decisions about the system were made and presented in Section 2.4. Firstly, an ontology for the class of events present in movies was created. This is generic enough to include all of the events in a movie, yet the classes are well defined so that it is possible to easily recognise which class any event belongs too. Also, a set of features were proposed that are used to detect these events. Finally, a strategy using finite state machines in order to detect these events was presented. Based on the decisions made in Section 2.4, an overall system structure was proposed in Section 2.5.

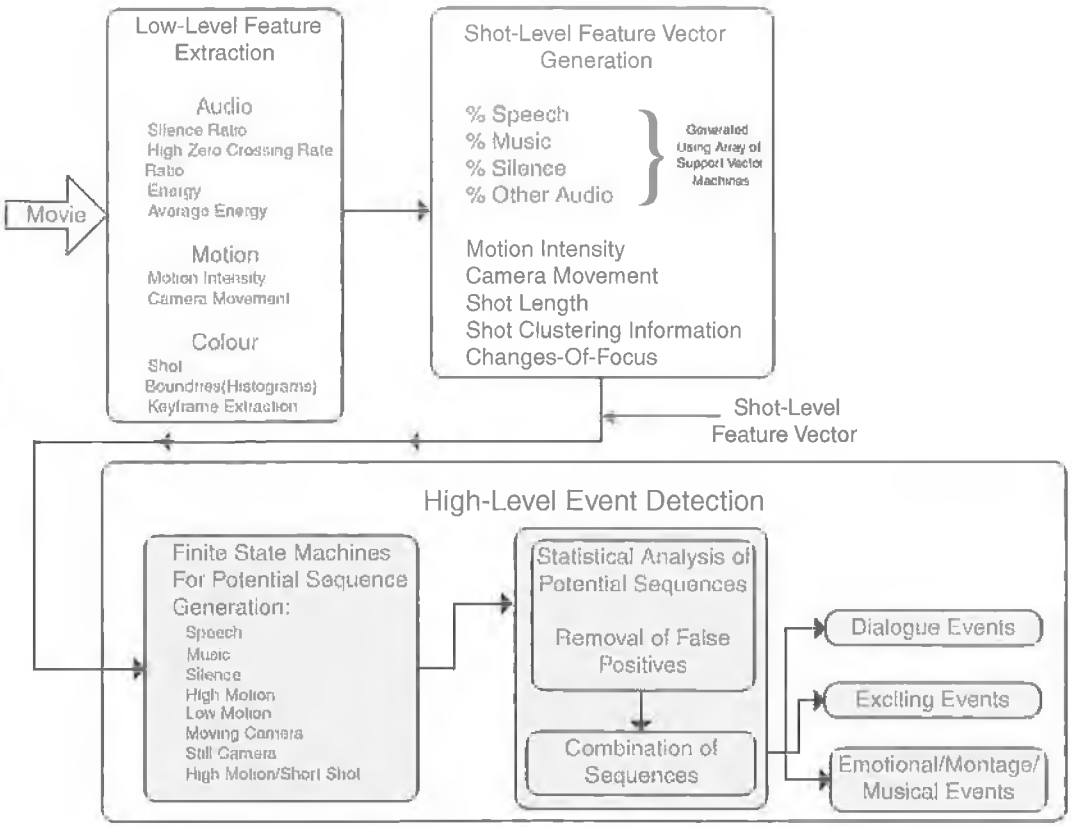

Figure 2.10: System Overview

This system utilises audiovisual analysis techniques to extract information about movies, and using this information creates an event-based index of a movie.

CHAPTER 3

Low-Level Feature Extraction and Shot-Level Feature Vector Generation

3.1 Introduction

Chapter 2 discussed the common techniques used to create movies and proposed a system based on these techniques. Three event classes (dialogue, exciting, and montage events) were proposed that aim to cover all meaningful parts of a movie. Thus, the primary aim of the movie summarisation system is to detect each of the events in these three event classes. Previously, Section 2.5 described the overall event detection system structure. This chapter explains the feature extraction components of the system and illustrates how a set of features for each shot in a movie are generated. Firstly, Section 3.2 discusses the audiovisual format selection process. Secondly, Section 3.3 explains low-level feature extraction, and a set of features used to extract information about the video content are defined. Following this, Section 3.4 explains how these features are utilised in order to create a shot-level feature vector for each shot. Subsequently, Chapter 4 explains the remaining component, the high-level event detection module, of the system.

3.2 Video and Audio Format Selection

There are numerous digital video standards in existence. Although there are differences between each of them, they share many common elements. Typically, the structure of the video coder-decoder (codec) is quite similar for different video formats as many different codecs implement similar compression techniques. Thus, the selection of a video standard

is not a critical design issue in this work since, in general, analysis undertaken on one video format is equally applicable to any other format.

Compressed video was chosen for analysis over uncompressed video due to the large memory requirements of uncompressed video, and also due to the fact that compressed video typically contains inherent motion information in the form of motion vectors[1]. Although they share many common elements with other video formats, video compression standards such as H.261 and H.263 target low bit-rate applications (such as videoconferencing) and were therefore not considered in this work. The moving picture experts group (MPEG) have defined three video standards, MPEG-1, MPEG-2 and MPEG-4, which can be used for many different video applications. The video quality can be adjusted for each of these standards (for example, by adjusting the bit-rate or frame-rate), however, in broad terms, MPEG-1 is designed for consumable video such as a video CD, MPEG-2 is designed for high bit-rate applications (such as DVDs), and MPEG-4 is designed for both high and low bit-rate video applications[2].

For this work, MPEG-1 is used as the video format on which all analysis is undertaken as it is more widely used than the other formats and there are many MPEG-1 editing and transcoding tools in existence. However, any analysis undertaken on MPEG-1 video could also be undertaken on other video formats, and is therefore not reliant on MPEG-1 video. Appendix A.1 provides an introduction to general video coding before focusing on the MPEG-1 video standard.

As mentioned in Chapter 1, recent times have seen a large increase in the consumption of compressed digital audio. Audio formats such as MP3 and AAC can provide high quality audio, while efficient compression algorithms reduce the file size significantly. Indeed, some approaches to audio analysis work in the compressed domain and extract features directly from the compressed audio (for example, in [59] the scale-factor values from MP3 audio are used in order to discriminate between speech and music). However, in order to extract many audio features a full decode is required. It has previously been shown that features extracted from uncompressed audio, such as zero-crossing rate, silence ratio, energy estimation, fundamental frequency, cepstral coefficients etc., can be used for reliable audio classification [25, 24, 23, 22, 47, 60]. Thus, in this work, all audio is in PCM-encoded WAV audio format which allows direct access to the uncompressed bit-stream. An introduction to digital audio is also presented in Appendix A

[1] see Appendix A for an introduction to video coding

[2] MPEG-4 video also supports, among other things, transmission of video objects for object-based video

3.3 Low-Level Feature Generation

3.3.1 Introduction

The first module in the system in Figure 2.10 generates the set of low-level features targeted in Section 2.4.2. The generated low-level features can be classified into three categories, *Colour*, *Motion*, and *Audio*. Each of the features in these categories are explained in Sections 3.3.2, 3.3.3, and 3.3.4 respectively.

The term 'Low-level features' can be interpreted in a number of ways depending on the research context, so it is important to clearly define the meaning of the term in the context of this work. The aim of these features is to create a shot-level feature vector (explained in Section 3.4). Thus, the features must be extracted at a rate of at least one value per shot, and preferably at a higher rate. This is to allow for an accurate feature representation of each shot. If there was less than one value per shot, down-sampling of the features would be required to generate a value for a single shot, which may lead to sparse feature representation. The average shot cut rate is approximately 6 seconds[3], so in this work, a low-level feature is a feature which is extracted at a rate of at least four per shot, using the the average shot length as a standard value. As a result of examining a number of features extracted at various rates this is deemed a high enough frequency to allow for accurate representation of the characteristics of a shot. Therefore, in this work all low-level features are extracted at a rate of at least one per 1.25 seconds.

3.3.2 Low-Level Colour Feature Extraction

Colour information is often used when analysing digital video. A video frame is, after all, made up of a grid of colours in the form of pixels. The data firstly needs to be decompressed from the MPEG-1 video in order to access the colour values for each individual pixel. Once the decode is complete, a number of features can be extracted from the pixel values. The aim of the colour analysis is to use a feature that represents a frame of video so that it can be compared to other frames (much like a human would compare two images), both for frame matching and for shot boundary detection.

One possible way of comparing images is by directly comparing pixel values, i.e. finding the sum of pixel differences between two frames. This approach has a number of limitations. Firstly, it is quite computationally expensive, as each pixel value needs to be compared, and secondly, small changes in object location in an image can result in large pixel differences. To illustrate why using the pixel values directly often results in incorrect values, note the two pictures in Figure 3.1. The two pictures are quite similar (actually they are horizontally rotated versions of each other), so they should result in a

[3]Based on a sample of ten movies

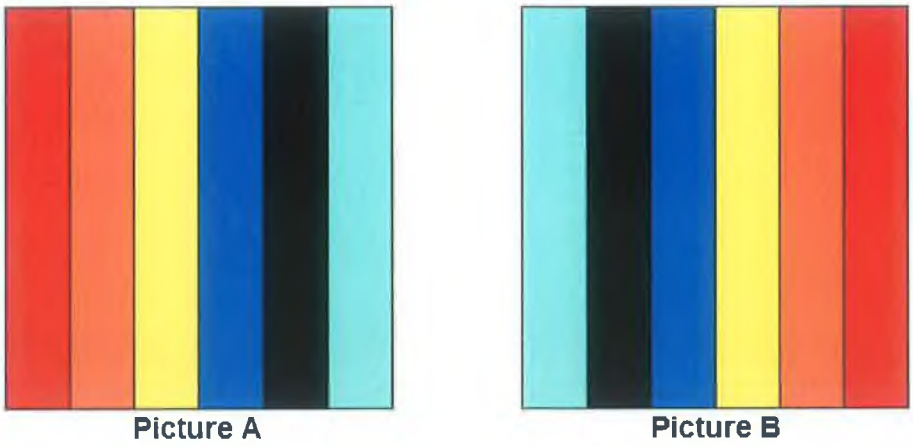

Figure 3.1: Two images

high similarity value if compared. Using absolute pixel difference will however, result in a very high pixel difference value between the two pictures, even though they are quite similar.

Another method of comparing single images is by detecting edges in a set of frames and comparing them. This involves firstly, filtering the images in order to detect edges, and then searching for similar edges in different frames. If a high number of the edges in both frames are identical then the images can be said to be similar. This is often used in shot boundary detection [16, 4] as it can be useful in detecting fades and dissolves. However, a number of studies have indicated that colour histograms are more adept in detecting shot boundaries [4, 3].

In order to achieve more efficiency, there have also been numerous approaches that compare frames in the compressed domain. By extracting the DC images of a set of frames, direct comparisons between frames can be made [61]. Although this means that only a fraction of the colour data is examined, it eliminates the need for full-frame decompression. However, since speed is not a critical issue, comparing DC images is not deemed suitable in this work.

Dominant colour has been used for retrieval of images in image databases [62]. This feature attempts to find the dominant colour(s) in an image by detecting the amount of pixels each of the dominant colours take up. This can then be used to find similar images. One advantage of this feature is that is is possible for it to be generated regardless of the colour space used, therefore for a large collection of images, direct comparison is possible.

Colour histograms have been demonstrated as a highly accurate and efficient method of comparing images and detecting shot boundaries [4, 3] and are used in this work. Colour histograms are a representation of colour values throughout the entire frame, as

opposed to localised pixel values. As the name suggests, a histogram of the individual colour channels is built for each frame. MPEG-1 uses the YUV colour space for pixel values, so three histograms are generated for each frame. The number of *bins* in a histogram represents the amount of values in the histogram. For example, the maximum amount of bins is 256, since there are 256 possible Y, U and V values for 8-bit video. Reducing the number of bins has the effect of increasing computational efficiency while still generating an accurate representation of the colour in the image. A 64 bin histogram, where each bin contains four pixel values, reduces the size of the histogram by a factor of four, without significantly reducing the quality of the colour representation. Using colour histograms to examine the similarity of the two pictures in Figure 3.1 will result in an exact match. This shows the benefits of the colour histogram approach. Figure 3.2 shows three 64 bin histograms for the Y channel of three frames of video. Note that image 1 and 2 are quite similar, and have similar histograms. Image 3, however, is quite different to the other two images and this is reflected in the histogram.

Another advantage in using a colour histogram as a representation of a frame is its reliability in finding shot boundaries. There are a number of ways to detect shot boundaries. However, as mentioned in Section 1.4, numerous studies [4, 3, 5], have indicated the high performance of colour histogram based approaches. The extracted colour histograms are used for shot boundary detection, key frame selection and shot clustering. This is explained in more detail in Section 3.4. In order to generate a colour representation of a frame in this work, a 64 bin Y histogram is extracted for each frame of video and used as the sole colour representation.

3.3.3 Low-Level Motion Feature Extraction

As outlined previously, MPEG-1 video is used for the analysis, thus the motion vectors created during encoding video can be used. Although the MPEG-1 motion vectors are designed for efficient compression rather than accurate motion representation, they are a reasonably efficient gauge of the movement in the video [62]. Also, due to the fact that they are compressed in the MPEG-1 data stream a full decode of the video is not required, which means that analysis can be extremely fast. The I-frames in an MPEG-1 video stream do not contain any motion information, and B-frames may contain backward as well as forward information. It was decided to only use the motion information from P-frames in the analysis since this does not significantly effect the accuracy, as P-frames typically occur every 3/4 frames in MPEG-1 video (i.e up to ten times per second) and any movement present in a movie will take place over a number of successive P-frames.

There are a number of methods of motion estimation and analysis. For example, much work has taken place on determining the direction of camera movement [62, 63]. Although knowledge of the direction of movement/zoom is useful in many scenarios, as

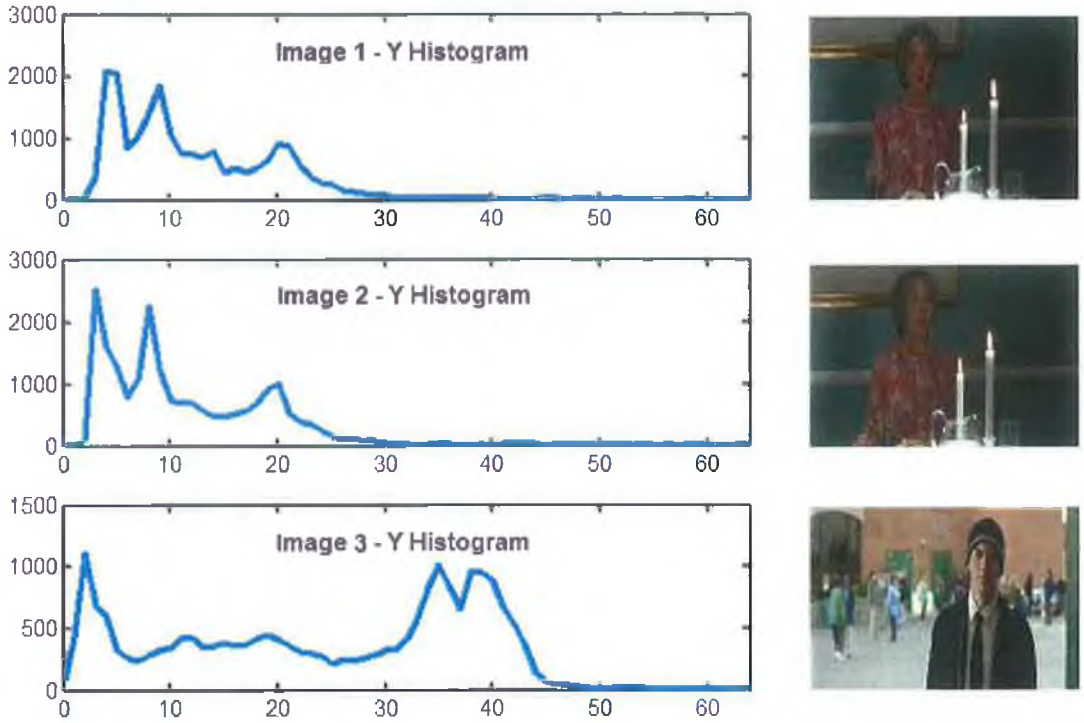

Figure 3.2: 64-bin Y histograms for 3 different images.

referred to in Chapter 2, for the purposes of event detection it is not required as this gives a limited amount of information regarding the directors' intentions. For example, knowledge that a camera pans left has no significant advantage over the knowledge that a camera is panning. For this reason, in this work all camera movement is treated equally. Similarly, knowledge of an object's trajectory within a frame (e.g. the direction of movement of a character) has limited use, as this typically has no bearing on the event type taking place.

Thus, for each P-Frame, two sets of motion features are required. The first motion feature, the standard deviation of the motion vectors, is an efficient way of measuring the *motion intensity* in a particular frame. The second motion feature, the zero motion vector runs, is used to detect the presence of *camera movement* in a particular frame. Both of these features combine to give an accurate representation of the movement in a video.

There are two important types of motion that need to be extracted in order to garner motion information from the video. The first is the motion within the frame. This can be seen as a measure of the amount of movement of the objects in the video, where the objects can be people, cars, planes, etc. Knowing how much of this movement is taking place can give insights as to the activities on screen, i.e. low amounts of movement may indicate a relaxed atmosphere in the movie, while a high amount of object movement may indicate an exciting event. Thus, the motion intensity aims to find the amount of

Intensity Value	Range of σ
1	$0 \leq \sigma < 3.9$
2	$3.9 \leq \sigma < 10.7$
3	$10.7 \leq \sigma < 17.1$
4	$17.1 \leq \sigma < 32$
5	$32 \leq \sigma$

Table 3.1: Thresholds for MPEG-1 Video

motion within a frame of video. This feature is defined by MPEG-7 [62]. Motion-vector magnitude indicates the magnitude of the motion itself, and is therefore used to give a value of the motion intensity. Both the standard deviation, and the average of the motion vectors could be used for this purpose, however MPEG-7 found that standard deviation gave a slightly better approximation of the motion intensity. The higher the standard deviation, the higher the motion intensity in the frame. In order to generate the standard deviation, firstly the mean motion vector value is obtained:

$$\bar{x} = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij}$$

where the frame contains $N \times M$ motion blocks, and x_{ij} is the motion vector at location (i, j) in the frame. The standard deviation can then be evaluated as

$$\sigma = \sqrt{\frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (x_{ij} - \bar{x})^2}$$

MPEG-7 define a five point scale comparing the standard deviation of the motion vectors to human observed values, the scale is shown in Table 3.1. Figure 3.3 shows the standard deviation values for a number of successive frames. In the first 81 values there is relatively little motion in the video. However after this point, there is a high amount of camera and character movement in most frames. This is reflected in the increase in the standard deviation values. There are a number of short pockets of frames without much motion (around the 190th sample for example), but overall the standard deviation is significantly higher.

The second type of motion required to complete the motion description of a video is the presence of camera movement. There will be significant overlap in the the results of both techniques, as a frame with high amounts of camera movement will also have a high motion intensity value. However, occasions where the camera movement may be quite small (a slow pan for example), will result in a low intensity value. On these occasions, knowledge of the presence of camera movement is still desired, so a separate feature is required. One possible method of extracting camera movement involves measuring the

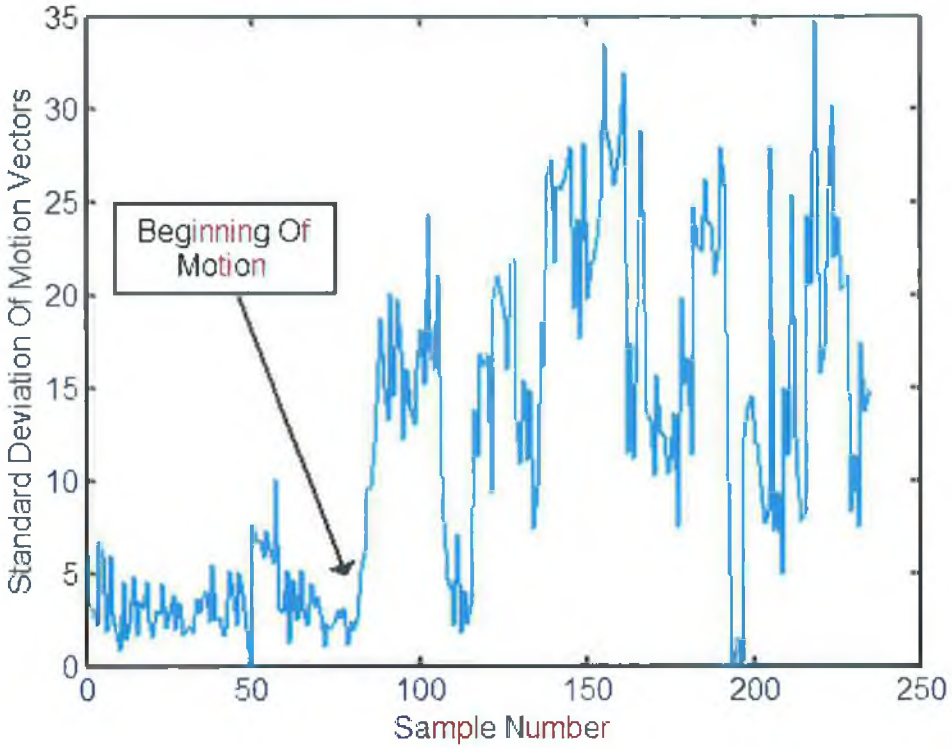

Figure 3.3: Standard deviation of motion vectors

optical flow. The optical flow of a set of motion vectors gives the velocity distribution observed by a viewer [25]. However, this approach is computationally intensive, and a more efficient method for detecting camera movement was sought. Other approaches that analyse camera movement are intent on detecting the direction of camera movement [64, 65], however, in this work only knowledge of the presence of camera movement is desired, and the direction of camera movement is not required. Therefore, the emphasis is placed on efficiently and accurately detecting any type of camera movement.

A novel camera motion detection method is proposed. In this approach, the motion is examined across the entire frame, thus, complete motion vector rows are examined. If there is no movement in a motion block from frame to frame, then the motion vector for that particular block will be zero. In a frame with no camera movement, there will be a large number of zero motion vectors present. Furthermore, these motion vectors should appear across the frame, not just centred in a particular area. Thus, the *runs* of zero motion vectors for each row are calculated, where a run is a number of zero motion vectors in a row. Three run types are created: short, middle and long. A short run will detect small areas with little motion. A middle run is intended to find medium areas with low amounts of motion. The long runs are the most important in terms of detecting camera movement and represent motion over the entire row. In order to select optimal values for the lengths

of the short, middle and long runs, a number of values were examined by comparing frames with and without camera movement. Based on these tests, a short run is defined a run of zero motion vectors up to $\frac{1}{3}$ the width of the frame, a middle run is between $\frac{1}{3}$ and $\frac{2}{3}$ the width of the frame, and a long run is greater than $\frac{2}{3}$ the width of the frame. Figure 3.4 displays the number of short, middle and long run values for a portion of video. There is relatively little camera movement in the first 81 frames, which results in high amounts of zero motion vectors across the frame, and therefore a high amount of zero motion vector runs. However, after this point the camera begins to move, following characters around the screen. As with Figure 3.3 there are small areas with no camera movement, but overall most of the frames after frame 81 contain camera movement. This has a significant effect on the number of runs, as the frames with camera motion contain little or no middle/long runs of zero motion vectors. The amount of short runs is reduced slightly, although not to the extent of the longer runs. Although, as can be seen from Figure 3.4, these runs accurately describe the motion within the frame, the selection of the run lengths is a non-critical design decision as there is a further thresholding step before camera movement is declared.

The amount of short, middle and long runs are counted for each frame. In a frame with camera movement there should be very few, if any, long runs of zero motion vectors, as the whole frame moves when there is camera movement. In order to find the optimal minimum amount of runs permitted before camera movement is declared, a sample of 200 P-frames was used. Each frame was manually annotated as being a motion/non-motion frame. Following this, various values for the minimum amount of runs for a non-camera motion shot were examined, and the accuracy of each set of values against the manual annotation was obtained. The values that resulted in the highest accuracy were used. As a result of this process, if there are less than 17 short zero motion vector runs, less than 2 middle zero motion vector runs, and less than 2 long zero motion vector runs in a P-frame, then the frame is labelled as containing camera movement. Otherwise is is labelled as a static camera frame. The results for the chosen thresholds can be seen in Section 3.4.3. The reliability of the other motion feature (the motion intensity) was previously tested by MPEG-7 in a set of user trials [62] and so was not investigated in this work.

3.3.4 Low-Level Audio Feature Extraction

Low-level audio features are used in order to classify the audio into different states such as speech, music, silence etc. The selection of these states is motivated by observations on the typical audio content in a movie. Thus, features are chosen that can be subsequently used to differentiate between these audio states. Many successful approaches to audio classification combine a number of features [66, 67, 59, 25, 45], and a combination of features is used here. There is quite a large variation in an audio signal, even if it remains

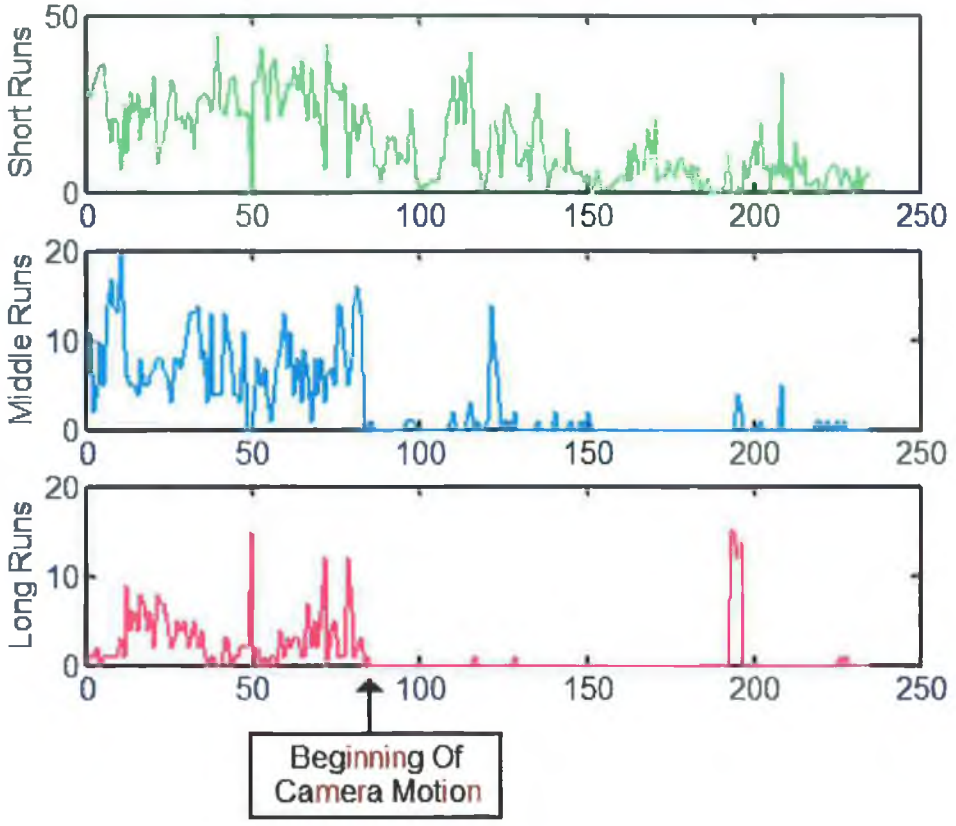

Figure 3.4: Short, middle and long runs of zero motion vectors

in the same state, so a combination of features is required in order to extract as much knowledge about the audio as possible before a decision as to its class is made. Initially, a number of audio features were extracted and their reliability in differentiating between the aforementioned classes of audio was assessed. Features such as fundamental frequency, Mel-frequency cepstral coefficients, and pitch tracking methods were extracted and examined. However, implementation of a reliable audio classification tool based on these features is deemed beyond the scope of this thesis. Estimating the fundamental frequency, for example, is not considered to be a solved problem in audio analysis and many algorithms exist that attempt to estimate it, yet none of them are satisfactory for a wide range of audio signals [25]. Also, in many cases it can be computationally expensive to extract these features. It was decided to implement a reliable and efficient set of features in order to classify the audio. These features, or variants of them, have been used to classify audio successfully in many different approaches. In [25], energy, zero crossing rate, silence ratio and cepstral coefficients are used to class audio speakers and music segments with high accuracy. In [45], silence ratio, energy values and the high zero crossing rate ratio are used to classify audio which is in turn used for extracting high-level semantics. This approach yielded high audio classification accuracy, and some of the features chosen for this analysis were based upon this approach. Four audio features were extracted in total, all of the features will be used to generate a single value for each second of audio. Ultimately, the amount of speech, music, quiet music, and silence per second of audio is detected, as described in Section 3.4.4.

High Zero Crossing Rate Ratio

In order to calculate high zero crossing rate ratio (HZCRR), the zero crossing rate (ZCR) must first be calculated. This is defined as the amount of times the audio signal crosses the zero line in a given analysis window, i.e. goes from positive to negative, or negative to positive. The ZCR for a single sample (using a $\frac{1}{100}$ second window, as recommended in [45]) is generated. The average zero crossing rate for a full second is then calculated using all of the zero crossing rate samples. The HZCR (High Zero Crossing Rate) value is defined as $1.5\times$ the average zero crossing rate. The HZCRR (HZCR Ratio) is, for a one second window, the ratio of the amount of ZCR samples over the HZCR to the amount of ZCR samples under the HZCR. This feature is very useful in speech classification, as speech commonly contains short silences in between the spoken words. These silences drive the average down, while the actual speech values will be above the HZCR, resulting in a high HZCRR [45, 25]. To illustrate this, Figure 3.5 shows the HZCRR for a 43 second segment of audio from a film. Initially there are some background noises (a car pulling up beside a pavement, and the sound of the electric window rolling down), after 16 seconds a man and a woman begin to talk to each other. There is a noticeable increase

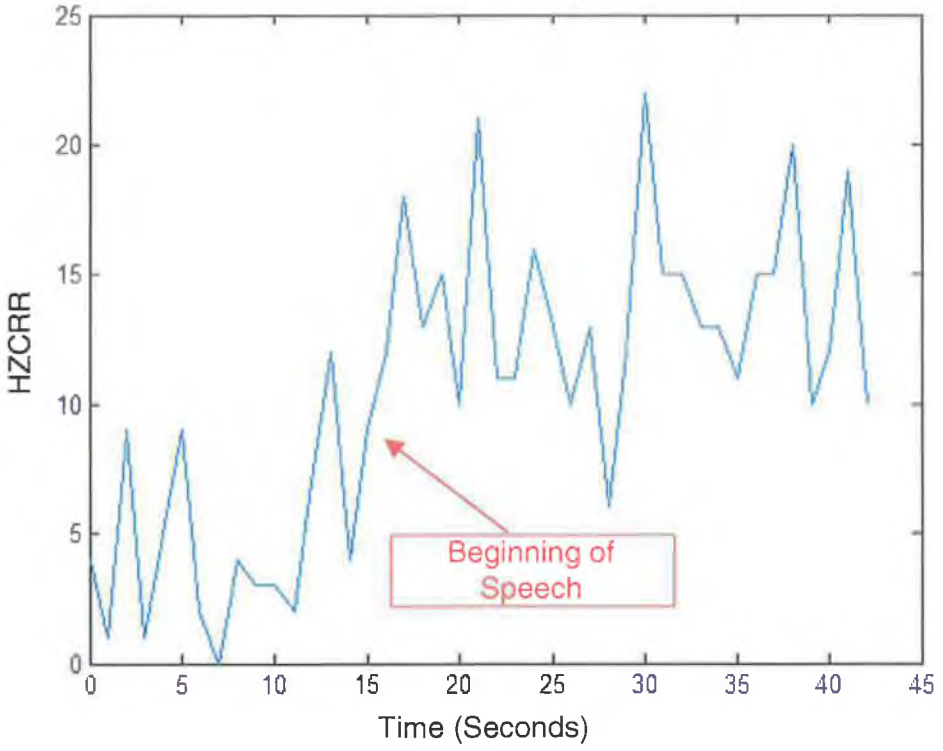

Figure 3.5: A plot of HZCRR values for a 43 second segment of audio.

in the HZCRR during the conversation, which continues until the end of the clip.

Silence Ratio

The silence ratio is a measure of how much silence there is in an audio sample. The Root Mean Squared (RMS) value of a one second clip is first calculated as

$$x_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \ldots + x_N^2}{N}}$$

where N is the number of samples in the clip, and x_i are the audio values. The clip is then split into a number of smaller temporal segments (in this case, as recommended in [45], fifty 20ms segments are used) and the RMS value of each of these segments is calculated. A silence segment is defined as a segment with a RMS value of less than half the RMS of the entire window. The silence ratio is then the ratio of silence segments to the number of segments in the window. This feature is useful for distinguishing between speech and music. Music tends to have constant RMS values throughout the entire second, therefore the silence ratio will be quite low. On the contrary, the gaps present mean that the silence

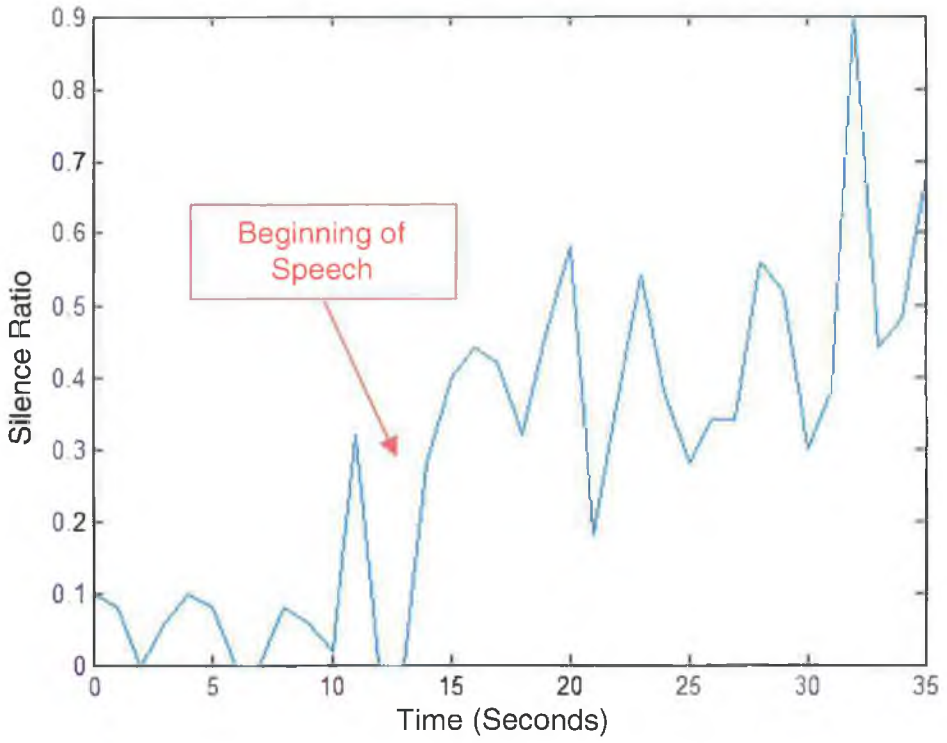

Figure 3.6: A plot of silence ratio values for a 35 second segment of audio.

ratio tends to be higher for speech [45]. Figure 3.6 illustrates the difference between the silence ratio of music and speech. The first 13 seconds of the clip of audio in the graph is music, while the remainder is speech. There is a clear increase in the silence ratio when the speech begins.

Energy Values

Two audio energy values are extracted. The first is the *short-term energy* (or short-time energy), while the second energy feature is a variant of the short term energy. In order to generate these features, firstly a one second window is divided into non-overlapping windows (150 in this case, as recommended in [25]) and the short term energy is calculated for each window as

$$x_{ste} = \sum_{i=0}^{N} x_i^2$$

This provides a convenient representation of the signal's amplitude variations over time [25]. Secondly, the number of samples that have an energy value of less than half of the overall energy for the one second clip are calculated. The ratio of low to high energy values is obtained and used as a final audio feature, known as the *short-term energy variation*. Both of these energy-based audio features can distinguish between silence and speech/music values, as the silence values will have low energy values [25]. Figure 3.7

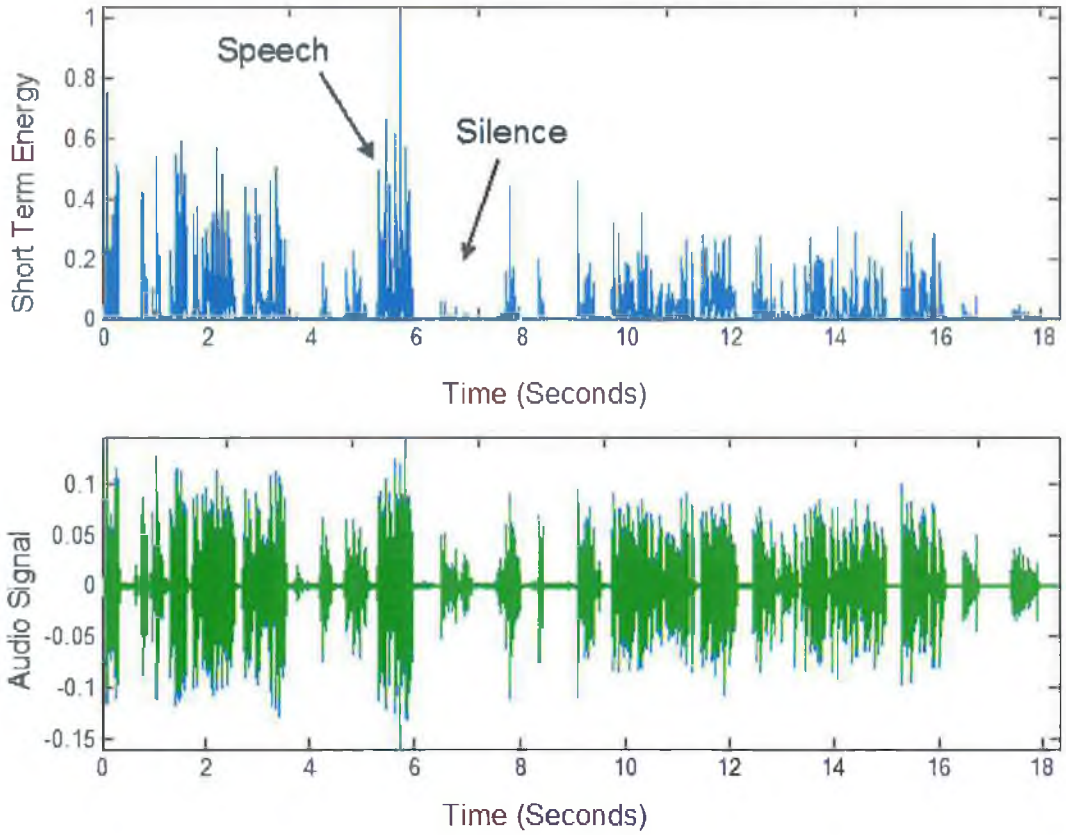

Figure 3.7: An audio signal and a plot of the short-term energy against time

shows the energy values for a 18 second clip, and the accompanying audio signal. This audio signal primarily contains speech, but there is a number of silent segments, one of which is indicated on the figure. Notice how the energy values decrease when there is no audio signal.

3.3.5 Summary

Extraction of the low-level features presented in this section constitutes the first step in the proposed approach to event detection in movies. The features were carefully chosen to maximise the amount of information gained. The features were chosen based on the the requirements of the event detection system, and the techniques commonly used by filmmakers. Of course, extracting information about individual frames has limited use. In order to learn anything meaningful about the content a higher level approach must be taken, where the shots and scenes are examined. Each feature presented is thus used at a later stage for higher level analysis. The next section details how these low-level features are used to generate a feature vector for each shot, which is in turn used to extract high-level semantics from the video content by detecting meaningful events in movies.

3.4 Generation of a Shot-Based Feature Vector

The previous section explained the low-level feature extraction techniques. On their own, these features are of limited value, but they can be used to detect semantic events if a higher level of meaning is inferred from them. There are two main reasons why the low-level features cannot be directly used. Firstly, the frequency of the features is too high. Knowing, for example, the motion intensity of a single P-frame has limited value, it is only when the surrounding P-frames are examined that the true motion information can be estimated. The second, and perhaps most important reason why the low-level features are not directly useful in their current form, is that the timing of each feature type is different. The colour histograms are extracted at a rate of one per frame, the motion features are extracted at a less frequent rate of one per P-frame, and one set of audio features is extracted for every second of data. In order to infer higher level semantics from these low-level features, it is required to create a common data unit, for which all features will be evaluated (i.e. up-sampled or down-sampled).

There are many levels of the video structure (as described in Section 2.2) on which this common unit could be based. It would be possible, for example, to reduce the features to the lowest possible unit, and down-sample the motion and audio data to have a set of features for each frame of video. However, this would still result in a frequency of features that is too high (as there will be a set of features for every $\frac{1}{30}$ of a second[4], and it is difficult to extract knowledge about the movie based on a sample as short as this as very little happens in this period of time), and does not solve the first problem listed above. A second approach could be based on timing, producing a feature at a specified time increment, e.g. 1, 2, or 5 seconds. While this frequency level may be more suitable than 30 per second, it is difficult to decide on which time unit to choose. If it is one per second, this may be too frequent, as it may be difficult to make a decision on the movie based on one second. Conversely, one per, say, ten seconds, may be too infrequent, as a number of significant activities may take place within the ten second window. Also, the fact that shot length of films varies quite substantially from film to film makes this choice even harder. For some movies with a fast editing pace a short time increment may be suitable, while for movies with many long shots a longer time increment may be preferential. Due to the fact that features usually vary from shot to shot, a timing based system is deemed unsuitable.

In order to ensure that the frequency of features is optimal, it needs to be individual to each film and each editing style. In order to ensure this, a set of features are generated for each *shot* in a film. This means that the frequency of feature set generation is tailored to each film (and film-making style). This solves the timing problem of choosing an increment, and also since shots are filmed using the same camera, the features should be

[4]Assuming a frame rate of 30Hz

relatively consistent throughout the entire shot. Thus, in this work, a shot is considered to be the atomic unit of a film. This means that all high-level analysis assumes that a shot is the basic unit of a film.

Clearly, shot cuts firstly need to be detected. Section 3.4.1 explains how this is done in the author's system. Once a set of shot boundaries have been obtained, the methods of up-sampling the low-level features to a shot-level are described. The aim of shifting the low-level features to a shot-level is to produce a *feature vector* for each shot. This contains all of the audiovisual information about each shot. This feature vector is subsequently used in a classification strategy to detect high-level events.

3.4.1 Shot-Cut Detection

As defined in Chapter 2 a shot is a continuous set of frames filmed using a single camera. Shot cuts are an integral part of any digital video analysis method, and usually form the basis of a larger system. Generally shot cut detectors work by taking a single frame, comparing it to the next frame, and using some content-based measurement quantifying the difference. If the difference between frames is large, then this is a strong indication that a shot cut has occurred.

Section 1.4.2 introduced a number of approaches to shot-boundary detection. Numerous studies indicated that colour histogram based approaches outperform other approaches to hard-cut detection (such as edge-based, motion vector-based, pixel difference-based) [3, 4]. Thus, in order to detect shot boundaries, a colour histogram approach is implemented. Although the vast majority of edits in a movie are in the form of hard-cuts, there are occasions where fades, dissolves and, to a lesser extent, wipes, may be used. However, in contrast to hard-cuts, detection of these shot boundary types is not deemed to be a solved problem. Preliminary investigation into a fade/dissolve detection algorithm (based on [7]) resulted in an unacceptable increase in the amount of false shot boundaries detected. In order to partially alleviate this problem, the colour histogram based shot boundary detection method is designed to also detect short fades and dissolves.

As mentioned in Section 3.3.2, colour histograms give an accurate representation of the content of an image (or frame) and also can be used to detect shot boundaries. A colour histogram is generated for each frame. In order to compare frames, a measure of the difference between two histograms must be generated. The sum of absolute difference is used for this purpose as it gives a reliable difference measure for two histograms:

$$Diff_{xy} = \sum_{i=1}^{M} |h_x(i) - h_y(i)|$$

Where $Diff_{xy}$ is the histogram difference between frame x and frame y. h_x and h_y are the histograms for frame x and y respectively, and each contain M bins. If the

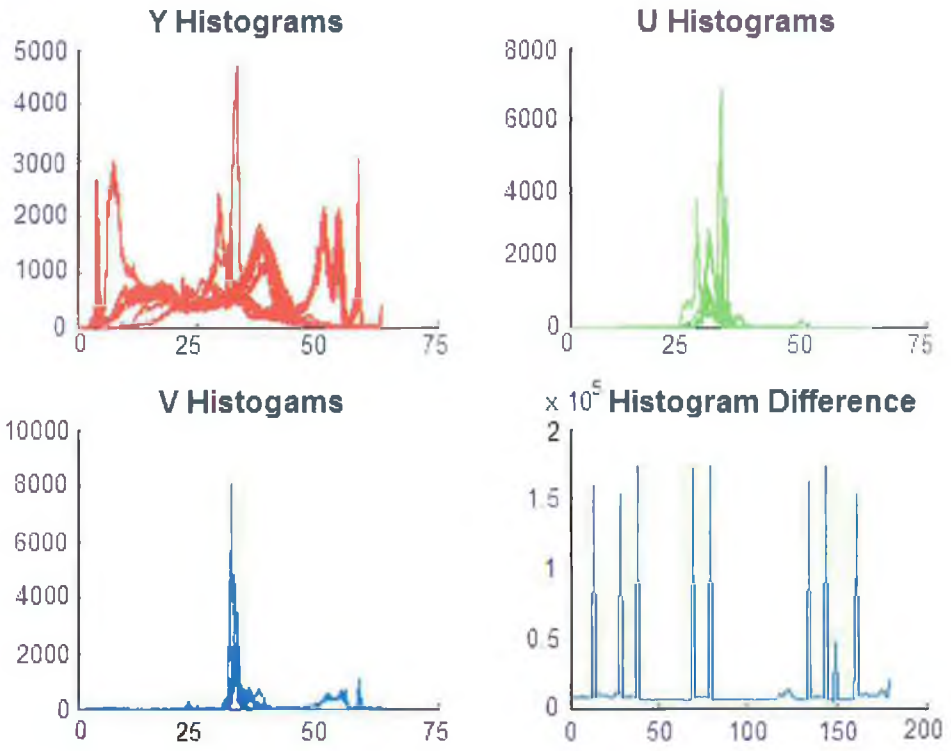

Figure 3.8: 64-bin Y, U and V histograms for successive frames of MPEG-1 video, as well as a graph of the histogram differences

difference between two successive colour histograms is greater than a defined threshold a shot cut is declared. This threshold was chosen based on a one hour sample of video data which contained a number of hard cuts, fades and dissolves. The threshold which achieved the highest overall results was selected. Figure 3.8 shows the inter frame colour histogram difference for a number of successive frames, as well as the respective colour (Y, U and V) histograms. The shot boundaries can be clearly seen in the bottom right graph as large spikes. As can be seen from this graph, the histogram difference at the shot boundaries is significantly higher than that of the non-shot boundary points. This meant that when selecting the optimal threshold for shot boundary detection, numerous thresholds achieved similar results.

The proposed implementation of a shot boundary detection system uses a two step approach. Although similar approaches to shot boundary detection have been proposed previously (see discussion in Section1.4.2), there are a number of small novel elements, and thus the approach will be presented here. Firstly, the colour histograms of successive frames are compared, and if the difference is above the predefined threshold the frame is marked as a *potential* shot cut. The second step involves tidying up these potential shot boundaries. This second step's main purpose is to remove multiple shot cut detections due to high movement or fades. It was generally found that a short fade would result in

Num. Detected Shot Cuts	Missed	False Positives	Recall	Precision
378	11	21	97%	94.7%

Table 3.2: Results of shot boundary detection system

multiple shot cuts being detected for a single 'real' shot cut. To counter this, a minimum shot length was set at 25 frames (just under 1 second). Whenever two shots cuts are detected as being less than 25 frames apart, they are merged and the final shot cut is declared as the true boundary. For example, if there are shot cuts detected at frame number 102, 103, 104 and 105, a single shot cut is declared at frame 105.

A thirty five minute section of video was selected to test the approach. The video contains samples from three different films. The results are shown in Table 3.2. As can be seen, the shot cut method achieves high results for both precision and recall. Many of the missed shot cuts occur due to long fades in the video. Many short fades are detected, as the inter-frame difference is above the threshold, but for the longer fades the difference is quite low and they are missed. Typically, the false positives occur due to high amounts of motion in the video, for example a fast pan or zoom. The results of this approach demonstrate its versatility. Although any incorrect results are undesirable, the number of missed cuts and false positives are kept to a minimum. The shot cut information is not only used as a basis for further analysis, but is also used to find the length of each shot, which is included in the shot-level feature vector.

3.4.2 Keyframe Selection and Shot Clustering

Using the set of generated shot boundaries, a representative frame is selected for each shot. This can be used both for further analysis, and also as a reference image in any system built as a result of this analysis. Any further colour analysis assumes that the keyframe is the sole representation of a shot, so rather than simply selecting a predefined frame from each shot (such as the first, last, or middle frame), a more accurate representation is desired.

In this approach to selecting keyframes, the average frame is sought, as this gives a good overall representation of the shot. The first step involves extracting a number of potential keyframes from each shot (i.e. if the shot is 90 frames long, and ten frames are selected, then the 0th, 9th, 18th, 27th, 36th, 45th, 54th, 63rd, 72nd, and 81st frames are potential keyframes) and computing the average colour histogram of these. The potential keyframe with the colour histogram closest to the average is selected as the keyframe for the shot.

The final part of colour analysis involves clustering similar shots together. The aim of this step is to cluster shots that are filmed using the same camera in the same location as this gives insights into the activities on screen. For example, in a conversation, typically

repeating shots of the same character are shot using the same camera in the same position. Thus, all shots of this character should have similar colour characteristics and be clustered together. The clustering method is based on the technique first proposed in [14], although variants of the algorithm have been used in other approaches since [15, 16]. The algorithm can be described as:

1. Make N clusters, one for each shot.
2. Stop when the histogram difference between 2 clusters is greater than a predefined threshold.
3. Find the most similar pair of clusters, R and S, within a specified time constraint.
4. Merge R and S (more specifically, merge the second cluster into the first one).
5. Go to step 2.

The time constraint in step 3 ensures that only shots that are temporally close together can be merged. The optimal value for this was chosen based on an hour long sample of video data. A cluster value is represented by the average colour histogram of all shots in the cluster, and differences between clusters are evaluated based on the average histograms. When two clusters are merged (step 4), the shots from the second cluster are added to the first cluster, and a new average cluster value is created based on all shots in the cluster. This results in a set of clusters for a film, each containing a number of visually similar shots.

Once shots are clustered, it is possible to detect changes in the focus of on-screen activity by detecting when one group of clusters ends and another begins. This gives a strong indication that an event is finishing. For example, in Figure 3.9 shots 1,3 and 9 belong to cluster A (i.e. shots 1, 3 and 9 are all shot from the same camera angle and contain visually similar information), whilst cluster B contains shots 2, 6 and 10, and similarly for the other clusters. The four clusters on the left hand side (A, B, C and D) are all related as the shots in these clusters are interconnected (e.g. cluster B is interconnected with clusters A, C, and D, as they all contain shots both before and after shots in cluster B), while the three clusters on the right (E, F, and G) are also interconnected. A change of focus is declared between groups of unrelated clusters. In order to find a change of focus, a cluster that is only related to previous clusters needs to be found. For example, if there exists a cluster that's highest value is 13 (shot 13), in order for a change of focus to be declared at the next shot, any cluster that contains a value higher than 13 must not contain any value less than 13. To find this, the maximum and minimum shot values for each cluster are found, and the change of focus is detected when a cluster maximum value is less than any following clusters minimum value. Thus, the transition between cluster D

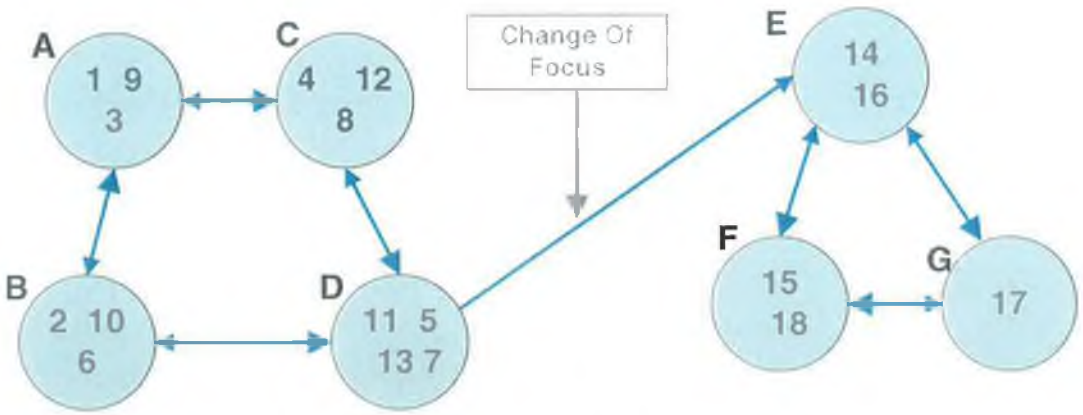

Figure 3.9: An example of shot clustering

and E marks a change in focus, as the earliest shot in cluster E occurs later than the latest shot in cluster D.

An alternate illustration of a change of focus is presented in Figure 3.10. Each cluster is represented as a bar from the first shot in the cluster to the last shot in the cluster. Thus, cluster A is represented by a bar from shot 1 to shot 9. If there is an overlap between the clusters, then they are deemed to be related (as the last shot in one cluster occurs before the first shot in a different cluster). Thus, clusters A, B, C, and D are related as they all overlap with each other. However, no overlap occurs between these four clusters and clusters E, F, and G. Therefore a change of focus is deemed to occur after shot 13 at the point indicated in the figure.

The change of focus is used to determine the point at which one event ends and another begins. For example, clusters A, B, C, and D, could contain all the shots of one dialogue event, and clusters E, F, and G, the shots of a montage event in a different location. In this example, each cluster may contain all shots of a particular character or location (as the camera repeatedly returns to a character/location, resulting in a set of visually similar shots which are clustered together). The change of focus (at the end of the dialogue event) can be declared after shot 13, which is the last shot in cluster D, as no shot after shot 13 is related to any of the shots before shot 13, and therefore the camera never returns to any of the characters in clusters A, B, C, or D.

There are many clusters that only contain only one shot. These occur when a shot is visually dissimilar to any other. There may be (and quite often are) changes-of-focus that occur at these single value clusters. This may lead to inaccurate results if there are a number of single value clusters in a row (as any one of these may be the true change of focus). In order to find the correct point, the forward and backward differences are obtained (that is, the difference between the single valued cluster and the last and next cluster to contain more than one shot) and used to determine which cluster the single value cluster is closest to. This is then used to pick a more precise value of the change

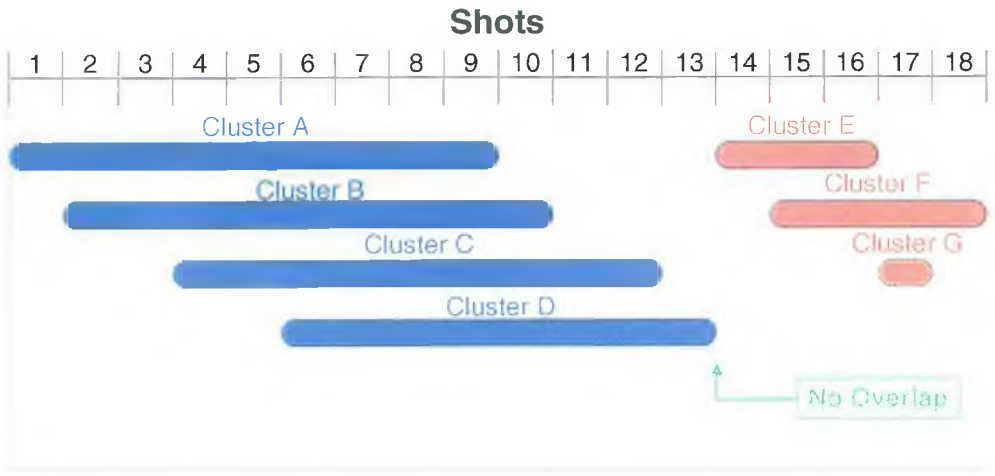

Figure 3.10: Alternate illustration for detecting a change of focus

of focus. For example if there were a group of clusters as shown in Figure 3.11. It can be seen that there is a change of focus somewhere between the end of shot 9 and the start of shot 13 (i.e. at the end of one of the clusters C, D, E or F), but the exact shot where the change of focus occurs is unknown. In order to find the exact location, the backward and forward difference is calculated. For cluster D, the difference between it and cluster C is defined as the backward difference, and the difference between it and cluster G, the forward difference (the same difference values are calculated for clusters E and F). For any single cluster, if the backward difference is smaller than the forward difference, then it is assumed that it occurs before the change of focus, if the forward difference is the smaller, then it is assumed that it occurs after the change of focus. In this case, it is found whether cluster D is visually closer to cluster C or cluster G. This process is repeated for the remaining single value clusters until the change of focus is found. So, for example, if the colour histogram from clusters D and E were visually closer to the average of the histograms in cluster C than the average histogram of cluster G, then they are deemed to be before the change of focus. If, however, the histogram from cluster F is closer to the average histogram of cluster G, than to the histogram from cluster C, cluster F is deemed to occur after the change of focus. Therefore the change of focus is marked as occurring between E and F. An alternate representation of this scenario, where clusters D and E are closer to cluster C than cluster G, and cluster F is closer to cluster G than cluster C is shown in Figure 3.12.

The clustering information is central in the event detection process, as it is used in order to locate boundaries between different events. A change of focus gives a visual indication that the movie has moved away from one event and on to another. Also, the clustering information can be used to measure the amount of shot repetition present in an event (as described in Section 4.2.3). The following sections explain in more detail the

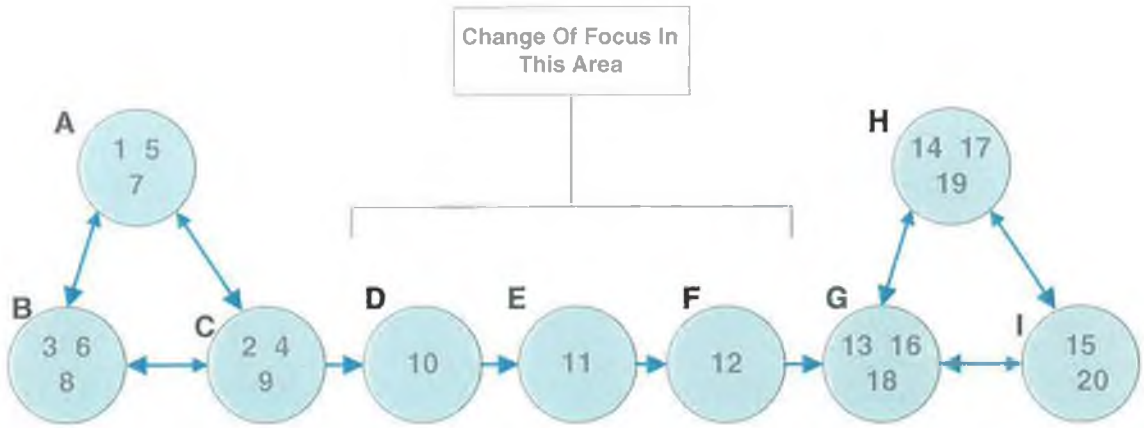

Figure 3.11: Using the forward and backward difference to find changes of focus when there are a number of single value clusters

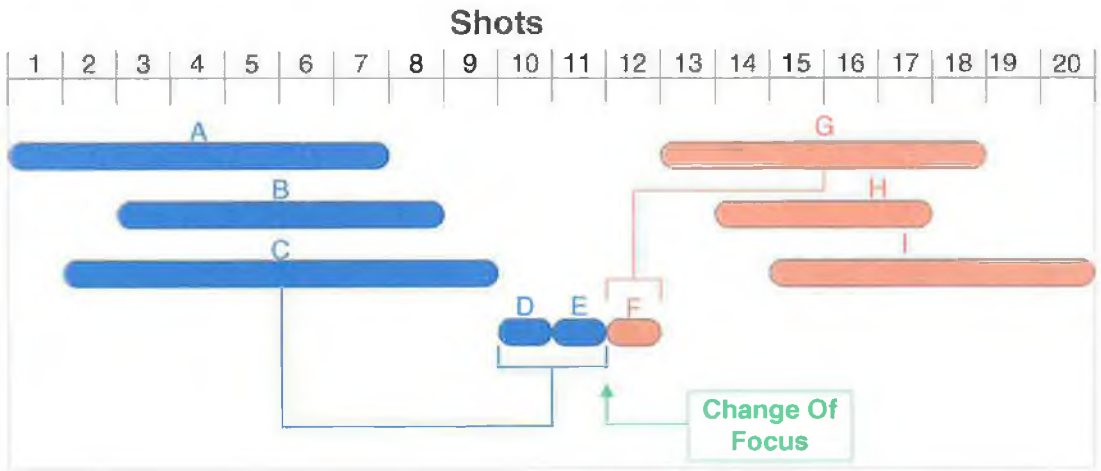

Figure 3.12: Alternate representation of using forward and backward difference to find a change of focus with single value clusters

components of the shot feature vector.

3.4.3 Shot-Level Motion Features

Two low-level motion features are described in Section 3.3.2; the motion intensity, and the presence of camera movement. Each of these features was evaluated at each P-frame of video. As defined previously, the fundamental unit of a film is a shot, therefore the motion values need to be up-sampled from one per P-frame, to one per shot. This is straightforward to do for both features. Firstly, the motion intensity value for a shot can be calculated as the average of the motion intensities for the P-frames within the shot. i.e.

$$X_{Mint} = \frac{1}{N}\sum_{i=1}^{N} P_i$$

where X_{Mint} is the motion intensity value for the entire shot, N is the number of P-frames in a shot and P_i is the motion intensity value for each P-frame.

Secondly, in order to generate a shot-based value for camera movement, the percentage of P-frames with camera movement is obtained. So, the shot camera movement value can be calculated as

$$X_{Cmot} = \frac{P_{Cmot}}{N}$$

where X_{Cmot} is the camera movement value for the shot, P_{Cmot} is the number of P-frames with camera movement and N is the number of P-frames in the shot. Thus, X_{Cmot} is the percentage of frames in a shot that contain camera movement.

The reliability of the standard deviation as a measure of motion intensity has been demonstrated previously by MPEG-7 [62]. In order to examine both the accuracy of the camera motion method, and to determine optimal thresholds that differentiate between shots with high/low amounts of camera motion, a set of manually annotated shots was generated. In total, just over 65 minutes of video data, containing 689 shots, was used. Firstly, each shot was manually labelled as being either a static shot (no camera motion) or a non-static shot (with some camera movement). For each shot the percentage of P-frames that contain camera movement is also calculated. Using both of these values it is possible to obtain the optimal amount of non-static P-frames present in a shot that can be accepted before a non-static shot is declared. A number of thresholds were examined, and the threshold with the highest accuracy on the test set was obtained. As a result of this, shots that contain greater than 20% non-static frames are labeled as non-static shots. This value is consequently used as a threshold in further analysis (see Section 4.2.2). Based on these tests and the 20% threshold, an accuracy of 85% was obtained (584 shots out of 689 marked correctly). Incorrect classifications are usually due to a lot of movement within a frame (such as a passing truck) which is detected as camera

movement. As the camera motion technique aims to detect global motion in a video, these false classifications cannot be avoided. Fortunately, in film grammar terms, movement of a camera and global movement on screen are often used to achieve the same effect (i.e. excite viewers) [51]. As detailed in Section 4.2.3, the motion intensity value (as opposed to the camera movement value) is used as a measure when looking for exciting events, so the false detections in global camera movement, when there is other movement, does not significantly affect the resultant analysis. Comparitavely few of the false detections were due to the system labelling a moving camera shot as a non-moving camera shot. This is encouraging, as the camera movement is used to detect sequences of static camera shots (see Section 4.2.2).

A similar process was undertaken in order to find the optimal threshold to differentiate between high/low motion intensity shots. Based on a sample of video with varying thresholds, it was observed that a motion intensity value of 9 is optimal in distinguishing between high-motion intensity shots and low-motion intensity shots. This is used as a threshold throughout this thesis. Both of the motion values are added to the shot feature vector, which now contains three values in total. The next section explains how the remaining values of the feature vector are generated.

3.4.4 Shot-Level Audio Features

This section describes how the low-level audio features defined in Section 3.3.4 are used to categorise the audio into a number of classes. After low-level audio feature extraction, there is a set of audio features for each second of audio. The features are: high zero crossing rate ratio (HZCRR), silence ratio, short term energy, and short term energy variation. These features cannot directly be used in order to extract information about the movie as they carry little meaning independently, however they can be combined and used to detect the presence of meaningful features. Thus, in order to accurately classify all of the audio types encountered in a movie (as discussed in Section 2.2.3), a set of audio classes are proposed which correspond to: *speech*, *music*, *quiet music*, *silence*, and *other*. All areas where people are talking will be placed into the speech class. Similarly, areas of clear music will be placed into the music class. Areas where there is either background music with no clear foreground audio, or quiet music (e.g. a piano playing softly) are placed into the quiet music class. Areas with no detected audio are placed into the silence class. Finally, all other audio (sound effects for example), are placed into the remaining class.

Clearly, there are areas in a movie which contain more than one audio type. For example, a character may be talking whilst there is music playing in the background. However, as indicated in Section 2.2.3, a sound engineer ensures that the relevant audio is clearly audible at all times. Thus, if speech is the relevant audio, while the music is merely part of the background audio, the speech must be made sufficiently clear for an

audience to hear. Therefore, this audio should be placed in the speech class. If, for a piece of audio, the music more relevant than any other audio, the sound engineer will make it sufficiently loud to ensure that the audience knows all other audio is less relevant.

The first step taken in classifying the audio is to implement a *silence filter*. This marks all frames as either silent or non-silent. After examining the features it was noted that using a combination of silence ratio, and short term energy variation gives the best indication as to the presence of silence. A ground truth of silent/non silent audio data was created, and used as a test bed to determine the optimal thresholds for the silence ratio and short term energy variation. This ground truth is just over 10 minutes long, and contains 605 one-second samples from three different films. A number of thresholds were tested, and the threshold that resulted in the highest accuracy was used. Once these thresholds were found, the silence filter was used to mark each second of audio as silent/non-silent.

The threshold-based system can be easily implemented for the silence filter, as, whenever there is silence there is a lack of an (or at least a very low) audio signal. However, the next step of the audio classification involves classing the audio that is present as speech/music/quiet music/other. This is a far more complicated task. Within each audio class, there is often large variations in the signal. For example, there is a large difference between low-tempo jazz music and rock music. Therefore, a threshold based solution, as used for the silence filter, will not suffice. A support vector machine (SVM) based approach is proposed that will detect the presence of speech or music. This is just one possible approach to speech/music discrimination. As mentioned in Section 3.3.4 there are numerous others that use more sophisticated analysis. However, a number of factors make the SVM based approach desirable in this application. Firstly, as there is a relatively low amount of computation involved, the analysis can be done quite efficiently. More importantly, as indicated later in this section, quite a high classification accuracy is achieved and there is minimal room for an increase in accuracy that may result from using a more complex approach. The trade off from garnering an minimal increase in accuracy by significantly increasing the complexity is not deemed worthwhile.

An introduction to SVMs is presented in Appendix B. As SVMs are a binary classification tool, two are required here. One is used to classify the audio into speech/non-speech, and another to classify the audio into music/non-music. The results of these are combined at a later stage. The implementations of the speech/non-speech SVM and the music/non-music SVM are comparable, and they both use the same features. Both SVMs are unrelated, and run in parallel. Only the implementation of the speech SVM is described here, but the implementation of the music SVM follows directly.

As mentioned in Appendix B, the first stage in generating a SVM is the training stage. This involves taking a number of positive/negative samples, and based on these samples finding the hyperplane that can separate the positive and negative samples as well as

possible. In this case, it means classifying an audio signal as speech/non-speech. A 15 minute ground truth of audio samples was created where the audio for each second was either labelled as speech (positive) or non-speech (negative). This ground truth spanned a number of different movies. One half the ground truth was used for training, and the remainder for testing.

Both the best combination of audio features to use for speech/non-speech discrimination, as well as the best SVM settings (cost factor, kernel type etc.) require investigation. To this end, the SVM was trained and tested multiple times, with the training parameters varied each time. Figure 3.13 shows precision and recall values on the test set for SVM models trained with varying parameters. The results of the two highest scoring feature combinations is shown. As can be seen from the graph, the highest performing feature combination results in using silence ratio with the HZCRR. This combination consistently achieves higher precision and recall values than using the same features with the addition of the energy values (which was the second highest performing feature combination). All other combinations of features (e.g. Energy and HZCRR etc.) performed worse than those shown in Figure 3.13. Using these results, the optimal set of parameters for the final SVM (which is then used to classify all audio) was chosen. This process resulted in choosing a radial basis function kernel, a cost factor of three, and a trade off between training error and margin of zero. The results on the test set using these optimal parameters is also shown in Figure 3.13. This resulted in a recall value of 95% and a precision value of 64%. Misclassification occurs if elements of more than one audio class are present in the sample. For example, musical sequences with a strong vocal element may be incorrectly classed as speech. Also, primarily silent segments of audio with some background noise are occasionally labelled as speech. The same SVM based process was undertaken for music classification, which resulted in an SVM that used the same features as the speech SVM (i.e. silence ratio and HZCRR), the same kernel, but it was found that a cost factor of two, and a trade off between training error and margin of three resulted in the optimal performance.

After the audio is processed by the silence filter and the two SVMs, three values are available for each second of audio: the output of the silence filter (1 or -1), the output of the speech SVM (positive for speech, negative for non-speech), and the output of the music SVM (positive for music, and negative for non-music). These features are all independently generated, i.e. the silence filter, speech SVM and music SVM are run in parallel, and their outputs do not depend on each other. There are two steps remaining, firstly, to categorise the audio for each second into an audio class, and secondly to create the audio value for each shot.

Each second of audio is classified into one of the five categories using a straightforward technique. If one value is positive, and all other values are negative, then that second

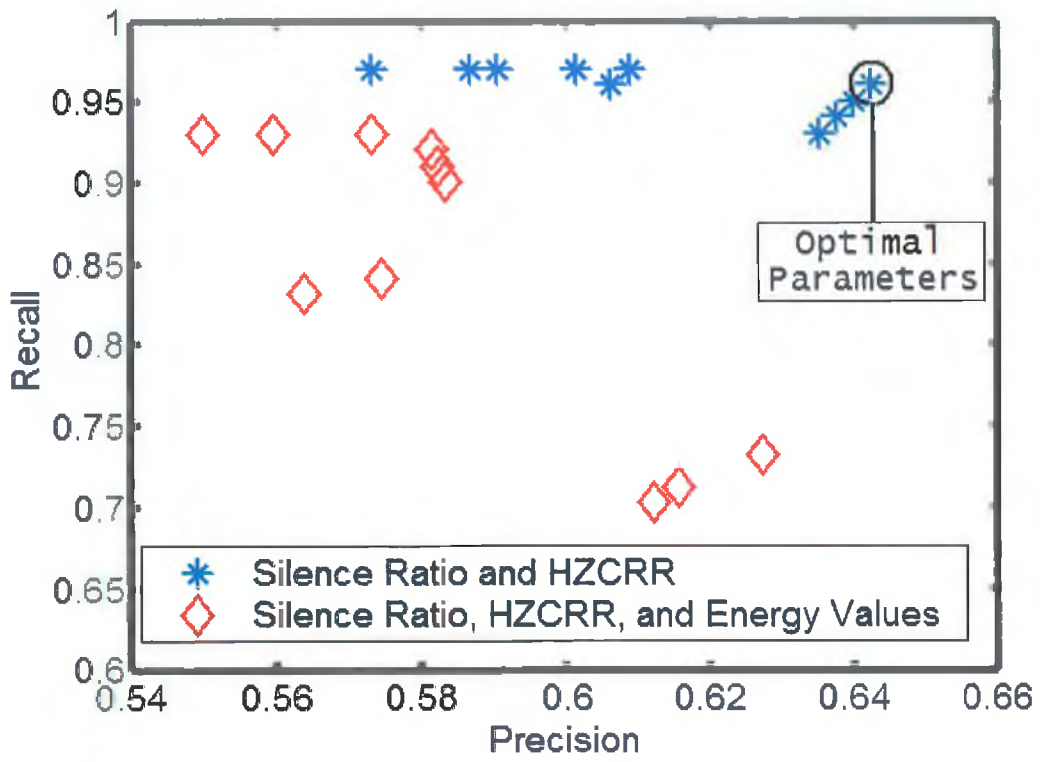

Figure 3.13: Results for speech/non-speech discrimination using an SVM with varying parameters and feature combinations

of audio is labelled directly into the corresponding category. For example, if the speech value is positive, and both the silence and music values are negative, then that second is labelled as speech. If both silence and music are positive, then the audio is placed into the 'quiet music' class. On the rare occasions where speech and music are both positive, then the audio is placed into whichever category has the highest value (as outputted by the SVM). All other possible combinations result in the audio being placed into the 'other' class. This is usually as a result of obtaining three negative values.

Some post processing to remove potentially false observations is then undertaken. This involves re-classifying audio that has a high probability of being incorrect. The nature of films means that the audio tends to be similar for long periods of time and if there is a change in audio it usually lasts for a number of seconds. This is because constantly changing audio will unsettle an audience. For example, in an event with music, typically there is music throughout the event, and there are no short bursts of speech present. Similarly, in a sequence of speech, if a character pauses between sentences, this will result in a second of silence being detected. Although this is a correct classification, in order for further, high-level analysis to be as reliable as possible it should be classed as speech. So, if the situation arises where audio frame X is classified as audio type A, and frames $(X-2), (X-1), (X+1)$, and $(X+2)$ are classified as audio type B where $A! = B$ then frame X is re-classified as audio type B. A two shot buffer on either side is deemed sufficient. If the buffer was set to be longer than two shots, two or more false classifications that occur near each other (e.g. three shots apart) may not be correctly reclassified.

The final step in audio classification takes the results for each second of audio and up-samples them to a shot-based value. This involves finding the percentage of each shot taken up by each audio class. For example in a shot that lasts ten seconds, if eight of the seconds are classed as speech, one second is classed as music, and the remaining second classed as silence, then the audio values for the shot are: (speech = 80%, silence = 10%, music = 10%, quiet music = 0%, other = 0%). These audio class percentages are inserted into the feature vector.

In order to test the accuracy of the audio shot classifier, a large number of samples from different films were extracted and the shots from each of these samples were manually classed into one of the audio categories. In total, samples from four films were used. The total length of all samples was 65 minutes and contained 675 shots. The manual classifications were then compared with the automatically generated results. As the dominant audio type is sought, if the shot contains 50% or more of a particular audio type, then that audio type is deemed to be the audio for the shot. Therefore a shot is deemed to have been classed correctly if the automatically generated audio value for a shot is 50% or above, and it corresponds with the manual annotation. For example, if a shot is labelled as 70%

speech and 30% music, then if it was manually marked up as speech it is a correct classification, but is incorrect if the manual annotation is music. The audio classifier achieved an accuracy rate of 90%, as 604 of the 675 shots were labelled correctly. One main cause for misclassification was where speech and music are both present, i.e. two people talking in a disco. Another reason for misclassification occurs when two audio classes are difficult to seperate, like a capella singing, or rap music (where the singing is often very similar to speech). The high accuracy rate indicates the versatility and reliability of the approach.

3.5 Summary

Section 3.2 discussed the selection of the digital video and audio formats used in this thesis. Following this, Section 3.3, explained the methods used for extracting low-level features. Each of the low-level features were chosen with the aim of extracting as much useful information about the movie as possible. Thus, the colour features that accurately represent the colour in each frame were selected, motion features that model the motion intensity, as well as detect camera motion, and audio features that can be used for audio classification were selected. Section 3.4 detailed the creation of a shot-level feature vector which is a set of features extracted for each shot in a movie. The complete feature vector for each shot contains: [% of speech, % of music, % of silence, % of quiet music, % of other audio, % of static camera frames, % of non-static camera frames, motion intensity, shot length]. In addition to this, shot clustering information is available, and a list of points in the film where a change-of-focus occurs is known. A brief summary of these features can be seen in Table 3.3.

Once all features are generated, semantic meaning must be extracted from the feature vector in the next step. Although the features chosen are tailored for movies (and therefore also for fictional television content), the generation of the feature vector isn't necessarily specific to movies. A similar set of features could be used to analyse news or sports content. If, in the future, a system to analyse news content was required, the system up to this point would be quite similar, as the same (or a similar) set of features could be used. However, the analysis does deviate from this point on, as the movie specific high-level semantics are extracted. While the feature vector generated may be common to many systems, how the features are used in the remainder of this thesis is tailored toward movie analysis, more specifically toward high-level event detection.

Feature	Discussion
Speech	The amount of speech present in a shot is used in order to determine whether characters are speaking during a movie. This is important in the detection of dialogue events. Speech is detected using SVMs and is detected with an accuracy of 90%.
Music	The amount of music present in a shot is primarily used in order to determine areas in a movie where the audio track is musically driven. This is of primary importance when detecting montage events. Music is detected using SVMs and is detected with an accuracy of 90%.
Quiet Music	The amount of quiet music present is also used in order to determine areas in a movie where the audio track is musically driven. Again, this is of primary importance when detecting montage events. It is detected using SVMs with an accuracy of 90%.
Silence	The amount of silence present in a shot can be used to locate areas where there is very little activity in the audio track of a movie. It is detected using a filter with an accuracy of 90%.
Other Audio	The amount of non speech/music audio present in a shot is used to locate areas where there is undetermined audio present in the audio track. It is detected using SVMs and is detected with an accuracy of 90%.
Static Camera	The amount of static camera frames in a shot is used to locate areas where the director uses a still camera. Still cameras are often used in order to allow an audience to concentrate fully on some activity on the screen and thus can give insight into the aims of the director at certain points in the movie. It is detected by examining the motion vectors of each P-frame with an accuracy of 85%.
Non-Static Camera	The amount of non-static camera frames in a shot is used to locate areas where the director uses a moving camera. Moving cameras are often used in order to create excitement so this features is primarily used in the detection of exciting events. It is detected by examining the motion vectors of each P-frame with an accuracy of 85%.
Motion Intensity	Motion intensity measures the amount on movement within a P-frame of video. Shots with high amounts of motion intensity may be used by film-makers in order to excite an audience. The standard deviation of the motion vectors is used in order to calculate the motion intensity.
Shot Length	The length of each shot is calculated by examining the shot boundary information. A number of short shots in succession may indicate that an editor is trying to create excitement, while long shots indicate that the editor is trying to relax the audience.

Table 3.3: Summary of extracted features

CHAPTER 4

Event Detection

4.1 Introduction

All of the audiovisual analysis described thus far has had the aim of generating a shot-level feature vector. Thus, the movie summarisation system up till this point has features in common with many other summarisation systems. It is at the stage of using features to extract knowledge of the content that the potential for innovation is at a maximum. At this point, an in depth knowledge of the content to be summarised becomes essential as a deep understanding of the video creation methods is required in order to extract meaning from the content. For convenience, the system overview diagram first shown in Section 2.5.2 is reproduced in Figure 4.1. This chapter explains the 'High-Level Event Detection' module of the system diagram.

As high-level event detection utilises finite state machines (FSMs), an introduction to FSMs is firstly presented in Section 4.2.1. This outlines the general principles of FSMs as well as illustrating a number of sample applications. Following the generation of the shot-level feature vector in Chapter 3, the method of detecting events in movies involves two more steps. Firstly, a set of *potential sequences* are generated. The potential sequences are sequences of shots with a common characteristic. An array of finite state machines (FSMs) is used to locate these sequences using the features from the shot-level feature vector. Potential sequence generation is explained in Section 4.2.2. Once this step is complete, the second step involves analysing the potential sequences, automatically removing any sequences deemed to be false positives, and finally, combining the potential sequences in order to produce a list of dialogue, exciting and montage events. These steps are outlined in Section 4.2.3. The process undertaken in order to place the detected list of

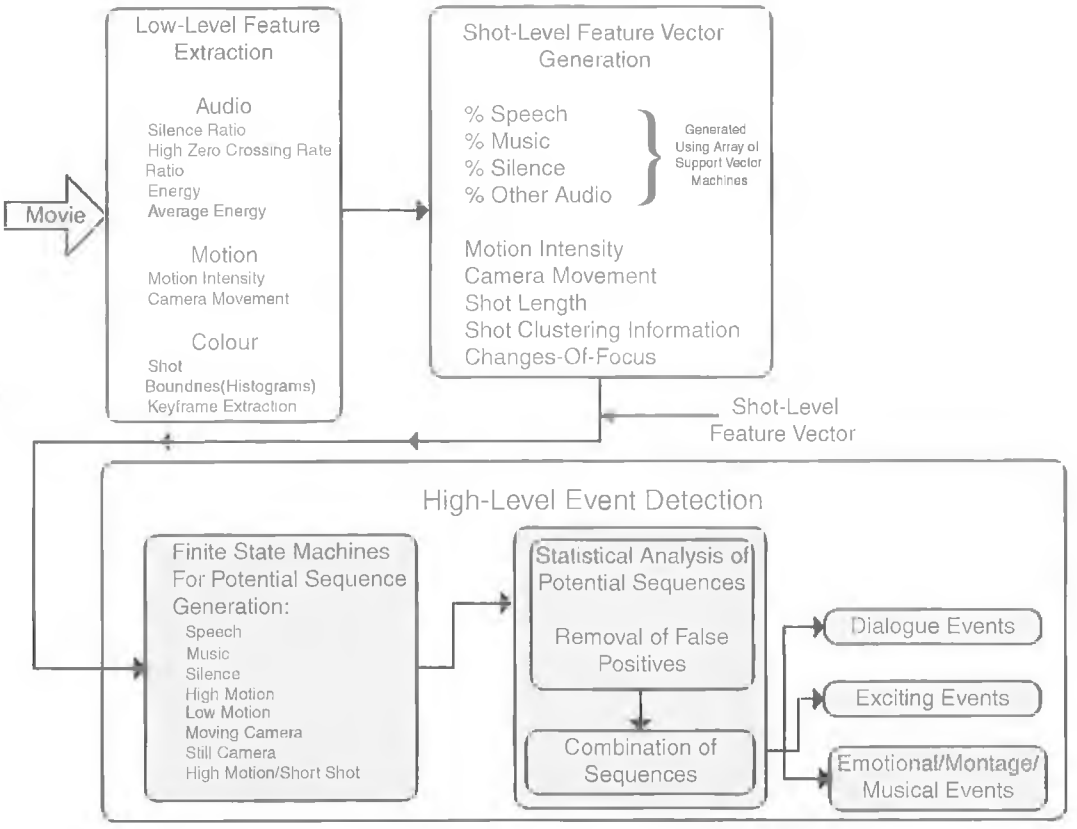

Figure 4.1: System Overview

events into a format that can be easily perused is presented in Section 4.2.4.

The need for searchable video databases is described in Chapter 1. A searching method is thus proposed which is based on the event detection methods presented in Sections 4.2.2 and 4.2.3. This allows a searcher to select both the potential sequences and the filtering methods, and therefore create a customised event list. Section 4.2.5 presents this event-based searching solution.

Finally, a second event detection method is presented in Section 4.3. This approach uses a hidden Markov model in order to classify each shot into one of the event classes. This system is described in order to illustrate an alternative event detection method and to provide a benchmark for the FSM based approach.

4.2 Film Grammar Based Event Detection

4.2.1 Introduction to Finite State Machines

Finite state machines (FSM) are models of behaviour composed of states, transitions and actions. A state stores the current condition of the model. The current state of the FSM changes periodically via a set of transitions, which transport the FSM from one state to

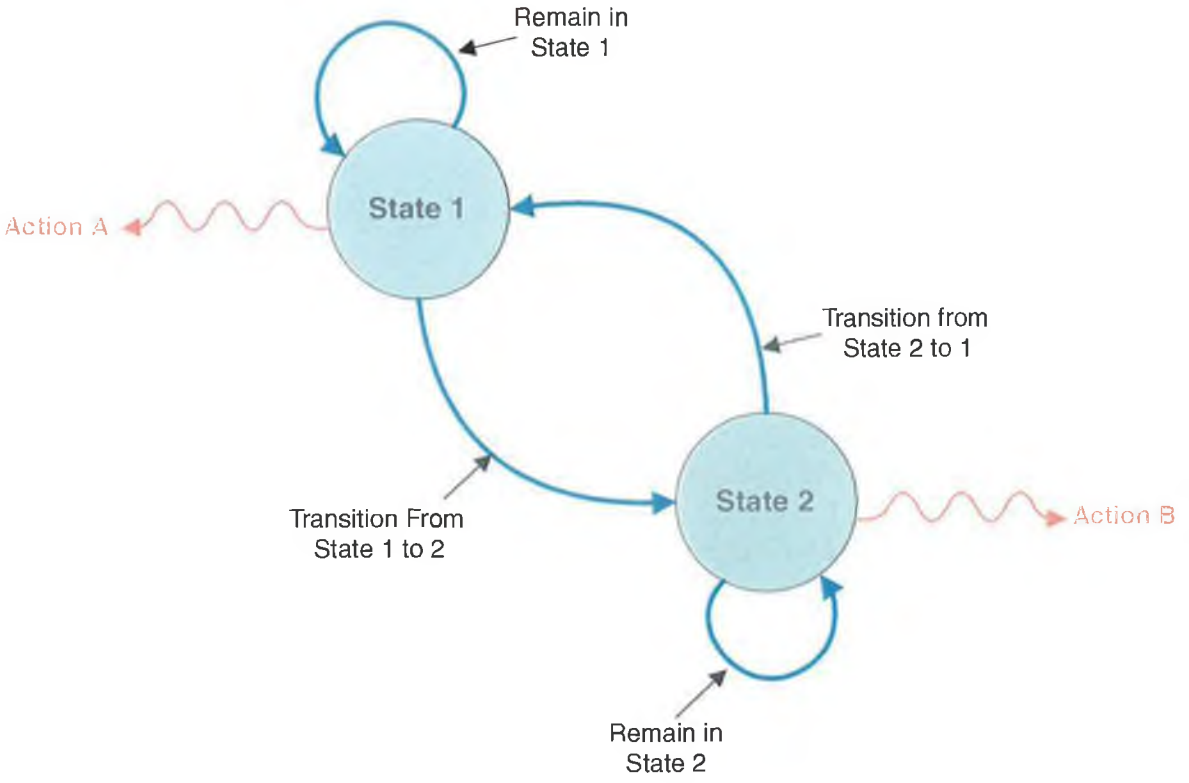

Figure 4.2: A 2-State Finite State Machine

another. Actions occur based on the current state and the transitions.

Figure 4.2 illustrates a general FSM. This FSM has two states. Therefore there are four possible transitions. The FSM can stay in its current state (1 or 2), or it can go from one state to another (either traverse from state 1 to state 2, or from state 2 to 1). The actions (A and B in fig 4.2) can be triggered by the FSM. The actions are not necessarily triggered each time the FSM is in a state. For example, action A may be triggered when the FSM enters state 1 from state 2, however it may not be triggered each time the FSM remains in state 1. Typically, the current state is evaluated periodically. For example, the FSM may check for transitional conditions every 2 μseconds. So after 2 μseconds has passed, one of the transitions will be traversed.

Contrary to many other data classification techniques, FSMs do not require any training. The transitions, actions and states are all defined by the user when the FSM is created. The advantage of this is that it gives the creator of a FSM complete control over its actions, transitions etc. This is useful when a specific set of actions is required by a system. However, the lack of training also necessitates that designers have a knowledge of the target application, and can specify exactly the actions, transitions and states of the FSM. This knowledge may not always be readily available.

Figure 4.3 illustrates a sample operation of a FSM. This is a possible configuration for a FSM that controls a set of traffic lights. This particular set of traffic lights also contains

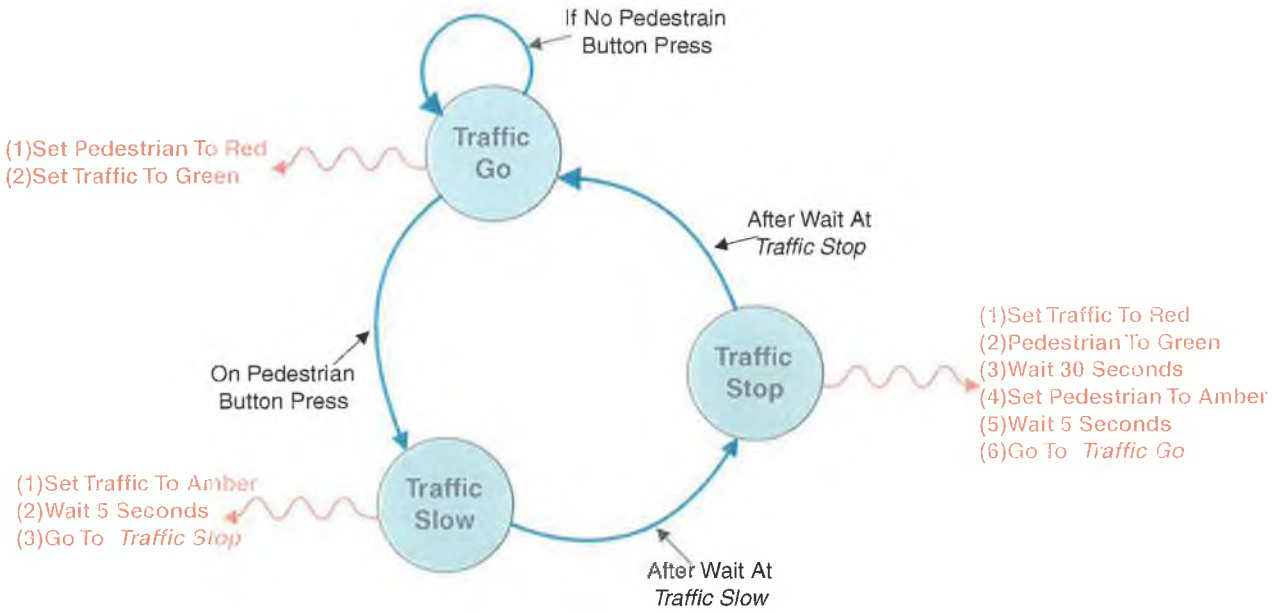

Figure 4.3: A Traffic Light Configuration of a FSM

a pedestrian crossing and an accompanying set of pedestrian lights. The pedestrian lights are triggered when someone waiting to cross the road pushes a button. There are three states that control the FSM, *Traffic Go*, *Traffic Slow* and *Traffic Stop*. The FSM begins in the *Traffic Go* state, where the status of the state is evaluated periodically to check for a button press. If there is no button press while in this state, the pedestrian lights are set to red, and the traffic lights are set to green. While there is no button press, the FSM remains in the *Traffic Go* state.

If a button is pressed while in the *Traffic Go* state then the FSM moves to the *Traffic Slow* state. When this state is entered, the traffic lights are set to amber to warn oncoming motorists that the traffic lights will soon be set to red. The FSM remains in the *Traffic Slow* state for 5 seconds, and then goes to the *Traffic Stop* state.

In the *Traffic Stop* state the traffic lights are set to red, and the pedestrian lights to green. The FSM then waits 30 seconds to allow the pedestrians to cross the road. Following this, the pedestrian lights are set to amber to warn pedestrians of the impending red pedestrian light. The pedestrian light remains amber for 5 seconds, and then the FSM goes back to the original state, *Traffic Go*, where the pedestrian light is set to red, and the traffic light set to green. The FSM remains in the *Traffic Go* state until there is another button press, and the process is then repeated. Note that in this system, the transitions are concrete, i.e. there is a 100% probability of going from *Traffic Slow* to *Traffic Stop*. This is because the desired next state at any given time is known by the designer, i.e. there is no desired probability of transition.

It is straightforward to add functionality to any FSM by adding extra states (and the

accompanying transitions and actions). For example it would be straightforward to add the functionality of a flashing amber light between the red and green lights, by adding in a new state between the *Traffic Stop* and *Traffic Go* states (or alternately by adding a flashing light as part of the actions of the *Traffic Stop* state). It is this flexibility, as well as the ease of designing FSM diagrams, that have ensured that FSMs have found use in many engineering systems, from microprocessor chips to computer games. The following section explains how FSMs are used in order to generate potential sequences for event detection.

4.2.2 Potential Sequence Generation

Chapter 2 explains how there are many film-making conventions in existence. Although movies are inherently creative, some of these conventions must be present in order to shoot a coherent event that the audience will understand. Following on from this assertion, Section 2.4.3 illustrates the need to detect events by examining the individual components of events rather than the events as a whole. By examining the individual components (e.g. speech or camera movement etc.) as opposed to a complete event, it is possible to build up a picture of what the filmmaker is trying to achieve, and from this deduce if an event type (i.e. dialogue, exciting or montage event) is taking place. Even if an event is being filmed in a particularly creative manner, some of the conventions must still be present, and thus it should still be possible to detect the event.

A potential sequence is described as a sequence of successive shots in which a specified feature occurs frequently. For example, a sequence of shots in a row that contains speech is termed a 'speech potential sequence'. As illustrated in Section 2.4.3, meaningful events contain a number of shots with common characteristics throughout. Typically, all of the shots in a dialogue event will contain speech for example, similarly, most of the shots in an exciting event will contain high amounts of motion. An array of FSMs is used to detect potential sequences. Each potential sequence type has one associated FSM.

The aim of the FSMs is to find sets of shots where particular features are dominant and label them as potential sequences. This does not necessarily mean that the feature is present in every shot, just that it occurs frequently. In any event, there may be times where a dominant feature is not present in a shot. Returning to a dialogue event, it would be expected that speech would occur in every shot, however, there may be shots of silence present within the event, for example if one of the speakers pauses for a moment. This pause is part of the overall speech potential sequence, so the detection of the whole sequence of shots is desired, including the pauses in between speech. Similarly, there could be moments of speech, in an otherwise predominantly non-speech sequence of shots which, again, should be detected as part of an overall non-speech potential sequence. Thus, the FSM cannot be too rigid in detecting potential sequence and some flexibility is

desired.

All of the FSMs used to generate the potential sequences are run independently, and in parallel. This means that the FSMs do not have any influence on each other. Many of the generated set of potential sequences will overlap, and therefore many shots will be present in more than one potential sequence. For example, if a sequence of shots contains both speech shots and static camera shots, then these shots would be detected by both the speech and static camera FSM, and therefore placed into both a speech potential sequence and a static camera potential sequence. This is in fact desirable given the subjectivity associated with the event classes as discussed in Section 2.4.1.

If a change of focus (as explained in Section 3.4.2) occurs while generating a potential sequence, the FSM must automatically declare the end of the potential sequence as the last shot before the change of focus. This is so that potential sequences are constrained to one event, and they do not span different events.

General FSM for Potential Sequence Generation

Recall that the shot feature vector contains the following values for each shot: [% speech, % music, % silence, % silence/quiet music, % other audio, % static camera frames, % non-static camera frames, motion intensity, shot length]. The thresholds detailed in Section 3.4 are utilised when constructing the FSMs in order to distinguish between speech/non-speech shots, static/non-static shots etc.

All of the FSMs used to generate the potential sequences are shown in Appendix D. As their design takes into account data that is introduced in Chapter 5, the individual FSM design process is explained fully in Section 5.3. The general design principles are presented here. There is some variation between the FSMs as each of them is tailored to a specific feature. However, they all follow the same design principle. There are three types of states in all FSMs created in this work: the initial (Start) state, the state which declares a potential sequence is occurring, and the intermediate (I) states which are placed in between the other two states. The state the FSM is in is evaluated after each shot. This general FSM structure is shown in Figure 4.4.

The FSM always begins in the Start state. Whenever a shot that contains the desired feature occurs (indicated by the blue arrows in Figure 4.4), the FSM moves toward the state that declares that a potential sequence is occurring (the state furthest on the right in all FSM diagrams). Whenever an undesired shot occurs (the green arrows in Figure 4.4), the FSM moves toward the Start state, where it is reset. If the FSM had previously declared that a potential sequence was occurring, then returning to the Start state will result in the end of the potential sequence being declared as the last shot before the FSM left the 'Potential Sequence Occurring' state.

The primary variation in the designs of the different FSMs is the configuration of the

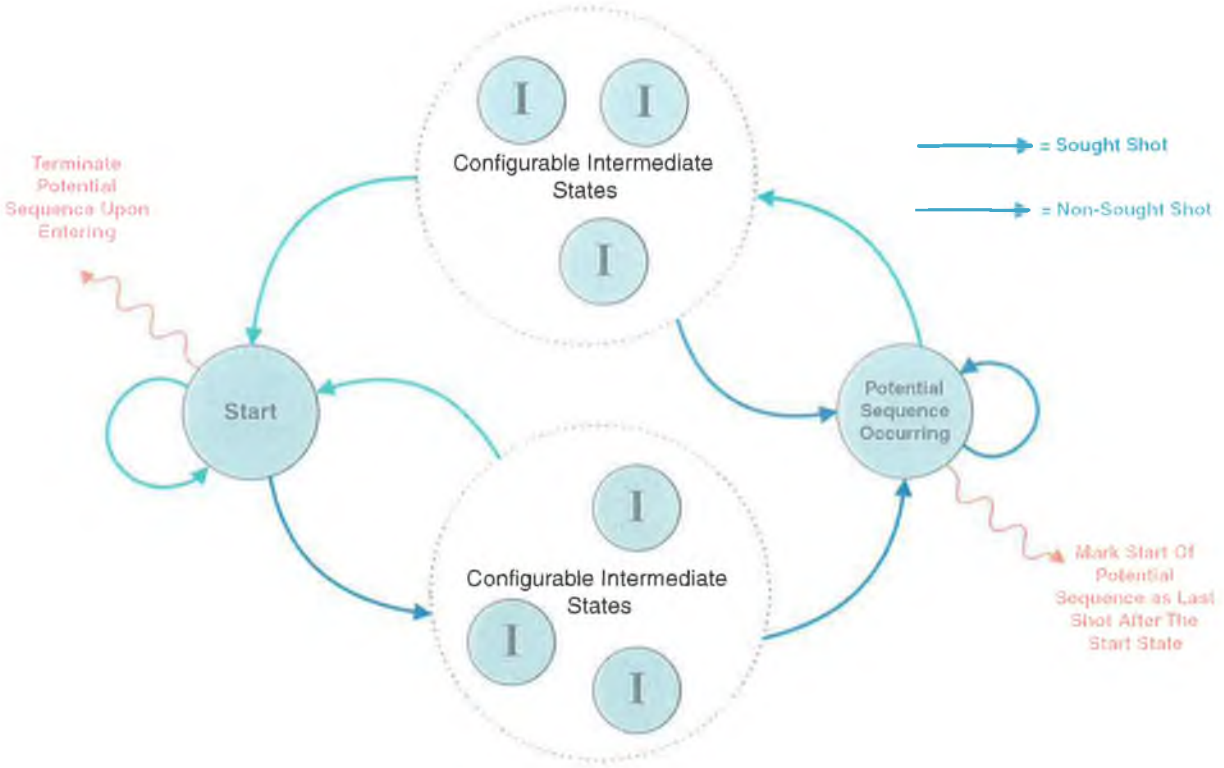

Figure 4.4: General FSM Structure

intermediate (I) states. In all FSM figures, the bottom set of I-states dictate how difficult it is for the start of a potential sequence to be declared, as they determine the path from the 'Start' state to the 'Potential sequence occurring' state. The top set of I-states dictate how difficult it is for the end of a potential sequence to be declared, as they determine the path from 'Potential sequence occurring' back to the 'Start' state (where the potential sequence is terminated). The design approach used when creating the individual FSMs is explained in Section 5.3. This process determines the optimal configuration of I-states for each individual FSM.

Illustration of Potential Sequence Generation

To illustrate the construction of a FSM, the *speech* FSM is described. This locates areas in the movie where speech shots occur frequently. Again, this does not mean that every shot needs to contain speech, simply that speech is dominant over non-speech. This state machine is shown in Figure 4.5 and is typical of the design of the other state machines used[1]. There is an initial (start) state on the left, and on the right there is a speech state. When in the speech state, speech should be the dominant shot type, and the shots should be placed into a speech potential sequence. When back in the initial state, speech shots should not be prevalent. The intermediate states (I-states) effectively act as buffers, for

[1]This FSM was designed after carrying out the analysis in Section 5.3

when the FSM is unsure whether the movie is in a state of speech or not. The state machine enters these states at the start/end of a speech segment, or during a predominantly speech segment where non-speech shots are present. When speech shots occur, the FSM will drift toward the 'speech' state, when non-speech shots occur the FSM will move toward the 'start' state. Upon entering the speech state, the FSM declares that the beginning of a speech potential sequence occurred the last time the FSM left the start state (as it takes two speech shots to get from the start state to the speech state, the first of these is the beginning of the speech potential sequence). Similarly, when the FSM leaves the speech state and, through the top I-states, arrives back at the start state, an end to the potential sequence is declared as the last time the FSM left the speech state.

For example, if the FSM begins in the start state, and ten speech shots in a row are encountered, the FSM will go to the bottom I-state on the first speech shot, then to the speech state on the next shot, where the beginning of a potential sequence is declared as occurring at the first speech shot. The FSM will remain in the speech state for the next eight shots. Following this, as non-speech shots occur, the FSM will begin to go back toward the start state via the top two I-states. When it reaches the start state, the last time the FSM was in the speech state is declared as the last shot in this particular speech potential sequence. At this point, the FSM waits to encounter more speech shots so that more speech potential sequences can be generated.

As can be seen, it takes at least two consecutive speech shots in order for the start of speech to be declared, this ensures that sparse speech shots are not considered. However, the fact that only one I-state is present between the 'Start' and 'Speech' states makes it easy for a speech potential sequence to begin. There are two I-states on the top part of the FSM. Their presence ensures that a non-speech shot (e.g. a pause) in an area otherwise dominated by speech shots does not result in an end to a speech potential sequence being declared. As with all FSMs, if a change of focus is detected via the clustering algorithm, then the state machine returns to the start state, and an end to the potential sequence is declared. For example, if there were two dialogue events in a row, there will likely be a continual flow of speech shots from the first dialogue event to the second, which, ordinarily, would result in a single potential sequence that would span both dialogue events. However, as the change of focus will result in the FSM declaring an end to the potential sequence at the end of the first dialogue event, there will be two distinct speech potential sequences (one for each event).

As the speech FSM runs through a movie, it generates a list of areas where the speech values of the speech-level feature vector is continuously high. This list is the set of speech potential sequences. Similarly, there are other FSMs that detect areas where other features are dominant. There are also FSMs for detecting *Music*, *Non-Speech*, *Static Camera*, *Non-Static Camera* and *High Motion Intensity/Short Shot* potential sequences. As can be seen,

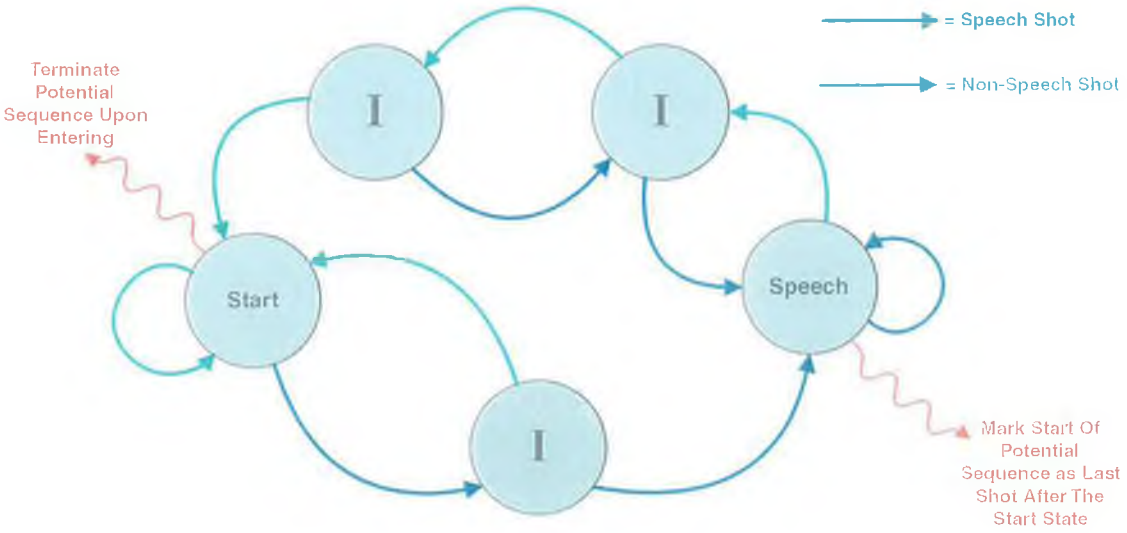

Figure 4.5: Sample FSM for detecting sequences where speech is the dominant shot type.

all of the FSMs contain one input feature, with the exception of the high motion/short shots (HMSS) FSM. Management of an FSM is simplified when only one input feature is used, as it allows for specifically tailored states and transitions suited to the feature in question. However, an exception was made for the HMSS FSM. This FSM combines two features that are closely linked. These two features were chosen due to film makers reliance on them for creating excitement on screen. The ultimate aim of the HMSS potential sequences, is to assist in exciting event detection. As mentioned in Chapter 2, a director has a set of tools to excite viewers, two main tools that are universally used are fast shot cuts, and high motion. These two features are explicitly linked, therefore a combined FSM is most beneficial in the ultimate aim of finding exciting events.

The HMSS FSM can be seen in Figure 4.6. The black arrow indicates the action that the state machine takes when the shot under examination has both high motion intensity and a short length. The green arrow shows the action that the state machine takes if the shot has both low motion and a long temporal length. The blue arrow indicates what happens when the shot has either a short length or high motion, but not both. The FSM begins in the start state. If it encounters shots that are consistent with those of an exciting events, then it tends toward the HMSS state where it stays until it encounters shots that are dissimilar to exciting event shots, where it begins to travel back toward the start state. Of course, in any exciting event, there will be shots that do not exactly follow the high motion and short length pattern, so again, the top states allow for these shots without terminating the potential sequence. Similarly, there will be shots in other parts of the movie that happen to have a short length and/or high motion activity, but are not part of an exciting event. The intermediate states (I-1 to I-5) ensure that the state machine doesn't classify these false positives as high motion/short shot potential sequences.

Again, the structure of the FSM follows from the way editing pace and motion are used in movies. Although motion may appear during other events (character movement in a conversation for example), the combination of both high motion and fast editing pace is strong evidence that the filmmaker is trying to create excitement. Thus, the number, and positioning, of the lower I-states (I-1 and I-2) make it easy to declare the beginning of a HMSS potential sequence when both features are present, but more difficult when only one feature is present, ensuring that only valid HMSS potential sequences are detected. The top I-states (I-3, I-4 and I-5) make it more difficult to declare an end to a HMSS potential sequence when low motion shots with slower editing occur. This is because these types of shots occur frequently during exciting events, and the potential sequence should not be ended due to these shots.

In order to demonstrate the functionality of this state machine, a short walk through is presented. For convenience sake, shots with both high motion and short length are labelled as '11'. Shots with low motion and long length are represented as '00', and shots with either short length or high motion but not both are represented as '10', or alternately '01'. An input of '10' is treated the same as an input of '01'. In this example, the state machine begins in the start state. As long as it receives shots with '00' it remains in that state. If it encounters a shot with '11', then it attempts to take the most direct route to the HMSS state and enters state I-1, upon which if it receives a '11' or a '10'/'01' it decides that it has enough evidence to declare the start of a HMSS potential sequence and enters the HMSS state. If, while in the start state, it receives a '10'/'01' it enters state I-2. This is an intermediate state that is necessary to ensure that '10'/'01' shots are not neglected (as they could be part of a potential sequence). If state I-2 receives a '00' input, then the state machine is reset, and goes back to the start state. However, if it receives a '11', then a potential sequence is deemed to have started, and the FSM goes to the HMSS state . The state machine remains in the HMSS state as long as there are '11' inputs.

In order for the potential sequence to be declared as being finished, long shots with low motion need to become the dominant shot type. If this occurs then the FSM will traverse states I-3, I-4, and I-5 and return to the start state (where the potential sequence is declared finished). In order to go through I-3, I-4, and I-5, there needs to be a number of '00' and '10'/'01' shots in a row. If, for example, there is a sequence 00, 01, 00, 00 then the state machine will go from state HMSS -> I-3 -> I-4 -> I-5, and finally to the start state where the potential sequence is declared over and the FSM is reset. If, however there is an exciting sequence that happens to contain non-action like shots, then the state machine may go into states I-3, I-4, or I-5, but will then return to the HMSS state once more exciting shots are encountered. This FSM is far more elaborate than the other single input FSM, as there are two inputs instead of one. If the number of inputs grew to more than two then the transitions would become increasingly difficult to map. The output of

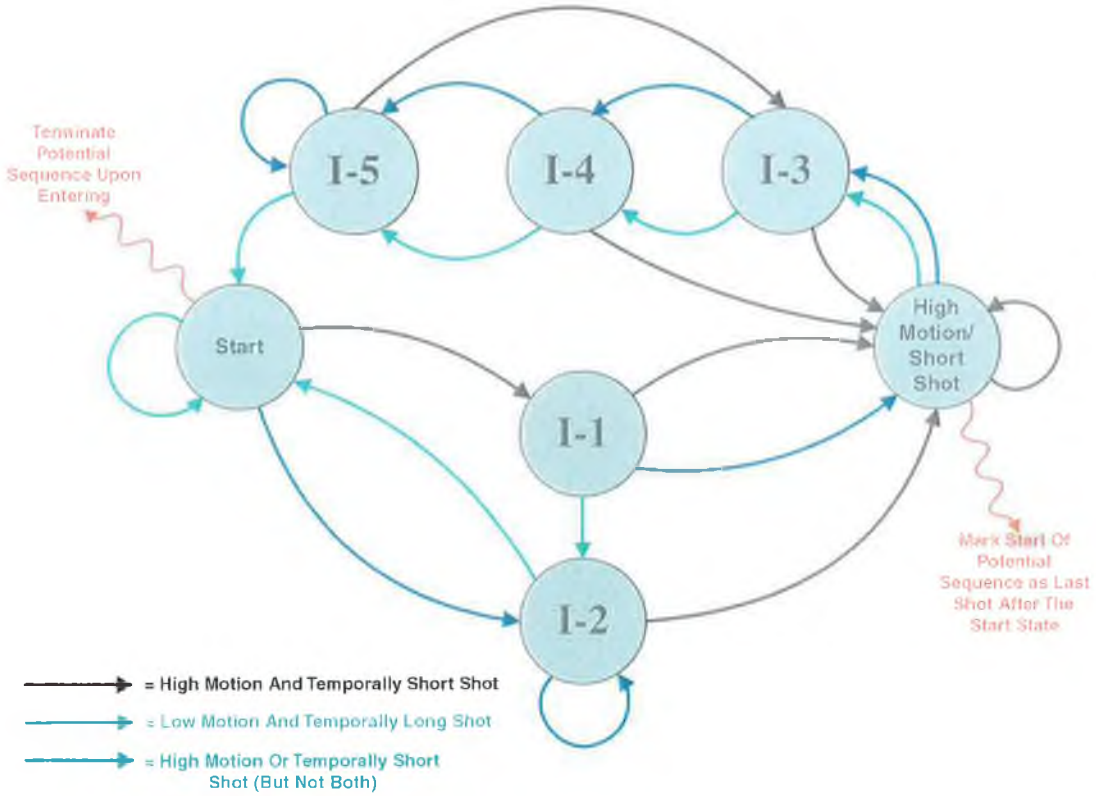

Figure 4.6: FSM for detecting high motion/temporally short shots.

this FSM is a list of all areas that contain high motion and short shots.

Resultant Output

The array of FSMs produce a set of potential sequences. All of these use features from the shot feature vector to find areas where particular features dominate. In total there are six sets of potential sequences, each of which are generated independently. There is, of course, overlap between many of the potential sequences. The music potential sequences will overlap considerably with the non-speech potential sequences. Similarly, the speech potential sequences may overlap with the static camera potential sequences (or indeed the non-static camera potential sequences) if both features are present in the feature vector. These potential sequences target the individual components of an event, and are subsequently used in order to detect events.

4.2.3 Detection of Events

The potential sequences generated by the FSMs cannot be declared as events even though they have the basic attributes synonymous with the desired events. For example, the conclusion could be drawn that any areas in the movie that contain a high amount of speech

shots are dialogue events, as whenever there is speech present, there must be a conversation. This, however is a premature conclusion. Of course, the presence of speech is a very good indication that a dialogue event is taking place, however, there could be speech present in a non-dialogue event as well. For example, parts of a movie with a voice-over will contain speech, but are not conversations. Thus, further evidence is required to move from a potential sequence to an event. This further evidence can be leveraged from the shot feature vector by examining how many of the shots in a potential sequence contain a different feature. For example, more evidence of a dialogue event is provided if there are also a large number of static camera shots present in a speech potential sequence, as a non-moving camera is a common feature of a conversation. Therefore, each speech potential sequence can be examined and the amount of static camera shots in the sequence can be calculated. If there is a high amount of static camera shots present, it adds further evidence that the speech potential sequence in question is indeed a dialogue event. Thus, in order to detect events, it is necessary to utilise heuristics which are generated based on the film-creation tools described in Chapter 2

One feature that has not been discussed, but is essential to the detection of dialogue (and other) events is the *cluster to shot ratio*, or *CS ratio*. This was not discussed previously, as this feature is extracted only for a potential sequence. The CS ratio is a measure of shot repetition for a sequence of shots. As implied by its name, it is the ratio of the number of clusters in a sequence of shots, to the amount of shots in the sequence. So, for example, given a sequence of fifteen shots, each of the fifteen shots belongs to a cluster, and each cluster is made by grouping shots with similar colour (as explained in Section 3.4.2). A low number of clusters indicates a high amount of shot repetition, as many similar shots are clustered together. A high amount of clusters indicates that shots are not repeating too frequently. Figure 4.7 shows two possible clustering scenarios for a set of fifteen shots. In the low CS ratio example on the left, the fifteen shots are all contained within three clusters, indicating a high amount of shot repetition. For example, the shots 4, 7, 11, and 13 are visually similar, most likely as they were shot with the camera in the same position with the same subject in frame. As there are three clusters, this sequence of fifteen shots has a CS ratio of $\frac{3}{15} = 0.2$. Contrast that with the shots on the right, where there are twelve clusters for the fifteen shots. This indicates that most shots are visually dissimilar to the all of the other shots in the sequence, as there are many clusters with only one or two shots. This gives a higher CS ratio of $\frac{12}{15} = 0.8$.

The following three subsections explain how the transition from potential sequence to event is performed, and describes how dialogue, exciting and montage events are declared.

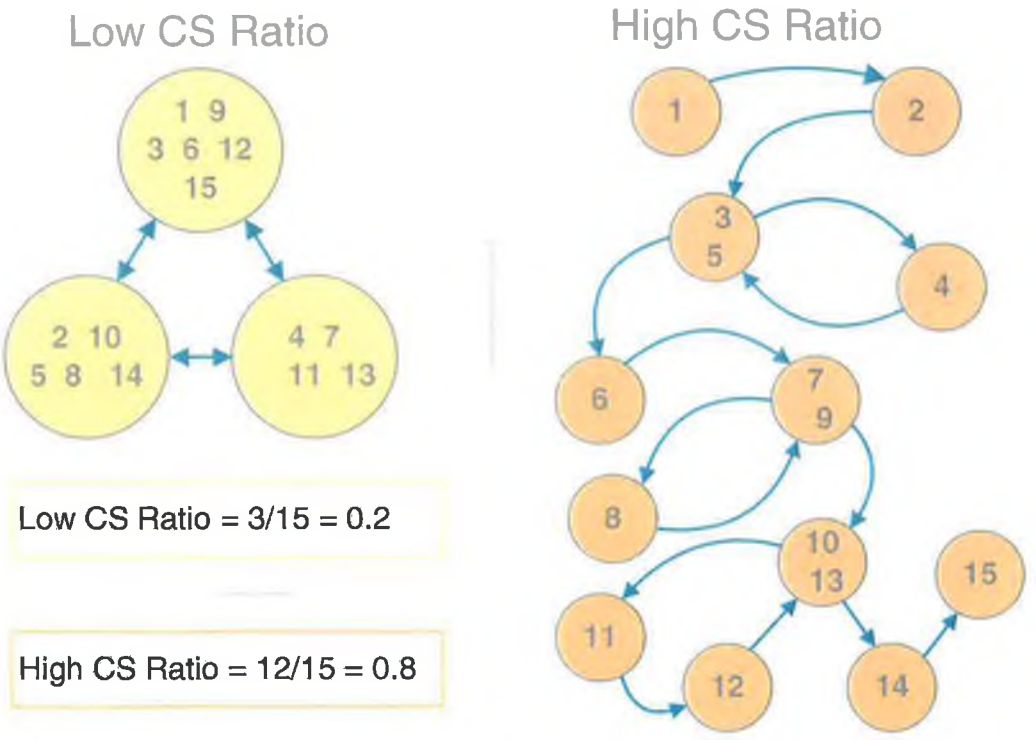

Figure 4.7: An example of shot clustering, and calculating the cluster to shot ratio

Dialogue Events

In order to detect dialogue events, the film making conventions universally used by directors, editors and sound engineers are leveraged. As described in Chapter 2, a typical conversation will contain a number of repeating shots with a non-moving camera that maintains the same position for successive shots of the same person, and will also contain clearly audible speech. However, not all conversations are shot in the same manner, and therefore the presence of all of these features cannot be guaranteed. In order to detect dialogues (and other types of events), the common fundamental components are sought, and then used as a basis for event detection.

As the presence of speech and a static camera are good indicators of the presence of a dialogue event, the potential sequences generated from the speech FSM and static camera FSM are used. The process used to ascertain if the potential sequences are dialogue events is as follows:

- The CS ratio is generated for both static camera, and speech potential sequences;
- For potential sequences detected using the speech-based FSM, the percentage of shots that contain a *static camera* is calculated;
- For the potential sequences detected by the static camera-based FSM, the percentage of shots containing *speech* in the sequence is calculated.

For any potential sequence detected using the speech FSM to be declared as a dialogue event, it must have either a low CS ratio *or* a be dominated by static shots (i.e. at least 50% static shots). Similarly for a potential sequence detected by the static camera FSM to be declared a dialogue event, it must have either a low CS ratio *or* be dominated by speech shots. The CS ratio threshold for distinguishing between low and high amounts of shot repetition is 0.67. If the CS ratio is below this figure, then the sequence is deemed to contain high shot repetition. This means that for any potential sequence, at least one third of the shots must repeat. If the potential sequence is relatively short, then this CS ratio will allow for an establishing[2] shot, as well as the repeating shots of a conversation. For example, if a potential sequence is 6 shots long, then there can be at most 4 clusters, this will allow for some isolated clusters with establishing shots or non-character shots, and some clusters containing shots of the characters talking.

The final step merges the retained sequences using a Boolean OR operation to generate a final list of dialogue events. This process ensures that different dialogue events shot in various ways, can all be detected, as they should have at least some features consistent with convention.

Exciting Events

As discussed in Chapter 2, based on filmmaking convention, the director has a set of tools available to excite the viewer. Although the effect of context cannot be understated, typically a directors intention can be interpreted based on the tools he/she chooses to use. In the case of creating excitement, the two main tools used by directors are fast paced editing and high amounts of motion. This has the effect of startling and disorientating the viewer, creating a sense of unease and excitement. So, in order to detect exciting events, the high motion/short shot potential sequences are used, and combined with a number of heuristics.

The first filtering step is based on the premise that exciting events should have a high CS ratio, as there should be very little shot repetition present. This is due to the camera moving both during and between shots. Typically, no camera angle is repeated, so each keyframe will be visually different. Secondly, short sequences of shots that last less than 5 shots are removed. This is so that short, insignificant moments of action are not misclassified as exciting events. These short bursts of activity are usually due to some movement in between events, for example a number of cars passing in front of the camera. It is also possible to utilise the audio track to detect exciting events by locating high-tempo musical sequences. This is detailed further (along with montage event detection) in the following section.

[2]Establishing shots are frequently used to set the scene in a movie. For example, in a conversation that takes place in a park, the establishing shot may pan around the park, before zooming in and focusing on the people talking

Montage Events

Emotional events usually have a musical accompaniment. Sound effects are usually central to action events, while music can dominate dance scenes, transitional sequences, or emotion-laden moments without dialogue [51]. Thus, the audio FSMs are essential in detecting montage[3] events. Notice that either the music FSM or the non-speech FSM could be used to generate a set of potential sequences. Although emotional events usually contain music, it is possible that these events may contain silence, thus the non-speech FSM potential sequences are used, as these will also contain all music potential sequences. The following statistical features are then generated for each potential sequence:

- The CS Ratio of the potential sequence.
- The percentage of long shots in the potential sequence.
- The percentage of low motion intensity shots in the potential sequence.
- The percentage of static camera shots in the potential sequence.

Potential sequences with very low CS ratios are rejected. This is so that potential sequences with very high amounts of shot repetition are rejected. This is to remove dialogue events that take place with a strong musical backgrounds. Montage events should contain high percentages of the remaining three features. Usually, in a montage event the director aims to relax the viewer, therefore he/she will relax the editing pace and have a high number of temporally long shots. Similarly, the amount of moving cameras, and movement within the frame will be kept to a minimum. A montage may contain some movement (e.g. if the camera is panning etc), or it may contain some short shots, however, the presence of both high amounts of motion and fast paced editing is generally avoided when filming a montage. Thus, if the potential sequence contains high values for one or more of these features, it is declared a montage event.

As mentioned in the exciting event section, the non-speech potential sequences can be used to detect exciting events. Distinguishing between exciting events and montages is difficult, as sometimes a montage also aims to excite the viewer. Ultimately, it has to be assumed, that if a director wants the viewer to be excited, he/she will use the tools available to him/her, and thus will use motion and short shots in any sequence that excitement is required. If, for a non-speech potential sequence, the last three features (% long shots, %low motion shots and %static camera shots) all yield low percentages, then the potential sequence is labeled as an exciting event.

[3]Note that, in this context, the term *montage* refers to montage events, emotional events, and musical events.

4.2.4 Post Processing and Keyframe Selection

Once the events have been detected, they need to be tailored into a form that can be presented to end users of a browsing system. In some cases, the start and end times of the detected events may be incorrect. For example, in a dialogue event running from 2:30 minutes till 3:45 minutes, the detected dialogue might run from 2:50 minutes to 3:35 minutes (if, for example, the speech didn't begin until late in the conversation and therefore wasn't detected by the speech FSM). For dialogue events, generally the start and end points are incorrect due to the presence of motion in some of the earlier/later shots in the event. It is quite common for a set of establishing shots to contain motion.

In order to further refine the boundaries, the start of a dialogue event is declared as the first shot of the cluster to which the first detected shot belongs. For example if the earliest shot in a detected dialogue event is shot 15, and the cluster that 15 belongs to contains shots {11, 13, 15, 18}, then the dialogue event is deemed to start at shot 11. Similarly the conversations are deemed to end at the last shot of the cluster to which the last detected shot belongs. This means that all previous and following shots of the characters in the conversation are included, as the shots in the same cluster are taken from the same camera angle and therefore are likely to contain the same person.

In order to present the dialogue events for browsing, keyframes needed to be selected. As the events in this case are conversations, the main requirement for the keyframes is that they show the people that are involved in the conversation (a secondary requirement could be that a keyframe shows the location of the conversation, but this information is usually inherent in a shot of a character so the emphasis here is placed on finding character shots as opposed to setting shots). In order to select these keyframes the structure of the dialogue events is examined, and the shots that repeat most often are sought. This is because a director will usually maintain the same camera angle on subsequent shots, thus, these are more likely to be shots of the characters. Also, shots that occur late on in the sequence are more likely to be clear shots of the characters, since early shots may be in an establishing context and may contain a moving camera. It was observed that in most cases three keyframes is enough to give a browser enough information about the event, yet be efficient enough to enable fast browsing. Typically, if a conversation involves a high number of people, more than one person is shown at a time on screen. Thus, three keyframes will still show many of the characters present.

The keyframe selection technique for dialogue events examines the clustering structure of the shots in a detected dialogue event. Firstly the largest three clusters in the event are found. These will contain the three most frequent shots in the event. As the event is a dialogue, these should be shots of the characters involved in the dialogue. Secondly, the last shot in each of these chosen clusters is chosen as the keyframe, as later shots have a higher probability of being a shot with a fixed view of a character. If there are only

two clusters in a dialogue event (i.e. a two person conversation that only contains shots of the two people talking) then only two keyframes are selected. The technique used to select keyframes could be adapted to find more or less keyframes (i.e. five keyframes per dialogue event) if required. This keyframe selection technique was tested on three movies to ascertain its reliability. It was found that 90% of the characters involved in a dialogue event were present in at least one of the extracted keyframes.

As there are two ways of detecting exciting events (using the method in the exciting events section, and the method from the rejected montage events), these two sets of exciting events need to be merged. This is done using a simple Boolean OR operation. As exciting events rely heavily on increased editing pace and high motion, the first occurrence of these features is deemed to be an accurate representation of when the exciting event starts, thus no further post processing to refine the boundaries of exciting events is undertaken. After analysing the results from this method of exciting events detection (as reported in Chapter 5), it was noted that there were a high amount of false positives. To alleviate this problem, a preliminary extra layer of filtering is introduced. Many of the false positives are actually parts of a different event, i.e. a dialogue event, that may contain a small number of fast paced shots at some stage in the conversation, or a montage event with a small amount of action (but not enough to be labelled an exciting event) in the middle. Thus, after all of the events have been detected, the overlap between the exciting events and the dialogue/montage events is examined. Note that this is a preliminary investigation to the effect of filtering on the exciting events, so the thresholds chosen were not subjected to the same evaluation process as the other thresholds in this thesis. If there is a high amount of overlap between events, then more stringent filtering is exerted on the exciting event. If there is high overlap between an exciting event and a dialogue event, the amount of speech shots in the exciting event is calculated, and if above 50%, the exciting event is removed. Similarly, if the overlap between an exciting event and a montage event is high, and the exciting event is less than eight shots the event is rejected. This is because it was noted that a number of montages contain short clusters of shots with camera movement, which were incorrectly labeled as exciting events. Although the additional filtering of the exciting events reduces the amount of false positives, there is a trade off involved as some of the true positives are also rejected. The results of the additional filtering are discussed in the results chapter (Section 5.2.4).

As montage events rely heavily on the presence of music, no further processing is required on the generated montage events to detect the correct boundaries. This is because the music occurs throughout the event, so the start/end of the music corresponds to the start/end of the event. In order to select representative keyframes for both the exciting and montage events, a number of temporally spaced keyframes are chosen. Due to their varying nature, it is difficult to ascertain what exactly is transpiring during excit-

ing and montage events. For example, exciting events in movies range from one-on-one fist fights, to battles involving thousands of people, to car chases through busy streets. Thus, a straightforward temporally based keyframe selection method is deemed suitable. Although any amount of representative frames could be chosen, in order to allow for efficient browsing, three equally spaced keyframes are chosen to represent the event.

The boundary detection and keyframe selection step is the final step in the event detection process. A list of dialogue, exciting and montage events are available, as well as a set of representative keyframes for each event. Note that this approach requires no manual interaction in order to detect events. Once the movie is in the correct format, all that is required for the user is to input the movie into the system, and a list of start and end times for dialogue, exciting, and montage events is generated. The event detection techniques are not specific to any particular format, and could be applied to any digital video type. Also, note that the principles that the event detection techniques are built on are equally applicable to fictional television content as well as to movie data. Chapter 5 extensively tests the approach to event detection described in this chapter.

4.2.5 Searching Videos

Although the three event classes that are detected aim to constitute all meaningful events in a movie, they are, in effect, three possible implementations of the same movie-summarisation framework. Within this framework, potential sequences were selected, and some post filtering was undertaken in order to generate a particular event list. The three event classes were chosen to directly target movie browsing. However, it is desirable for the event detection techniques to be applicable to other media and event classes.

In order to allow a user to control a set of returned events, a search based variant of the system is also proposed. In this system, a user can specify a set of features that he/she believes belongs to the desired event. The system then uses the FSMs and applies some user defined filtering to find events that contain the specified features. Note how searching involves two steps. The first step in detecting events uses the FSM to find areas where particular shots dominate. For example, in order to find the dialogue events, the speech FSM is used to find areas where speech shots occur frequently. The second step takes these potential sequences, and applies some user-defined filtering to remove unwanted events. Both steps in event detection utilise features that are already generated, and use them to find a particular event class. When generating a final list of events some further processing is undertaken, but this is usually only used to remove a number of false positives, and to specifically refine the event detection framework to a particular event. Much of the work in detecting events is achieved by, firstly, using a FSM to find potential sequences, and secondly, by filtering the potential sequences by only accepting the potential sequences with other features common to the desired event.

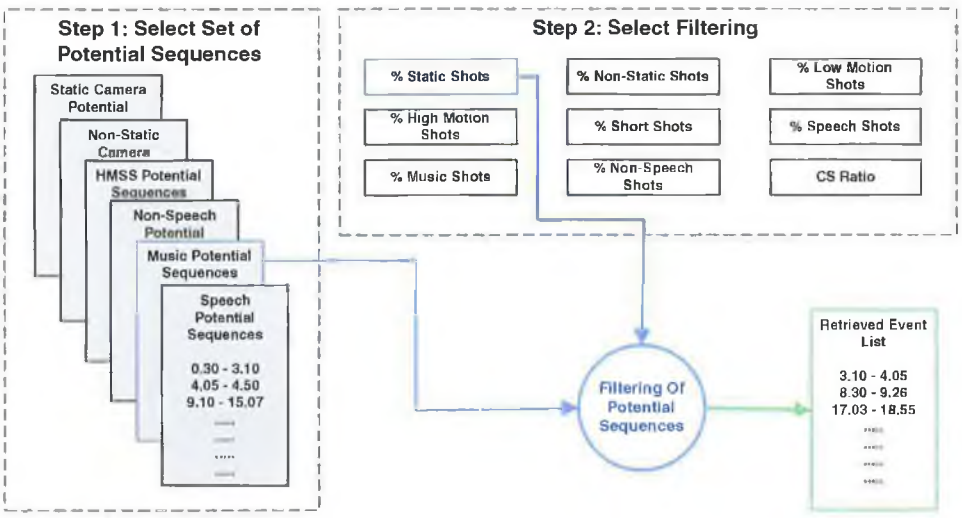

Figure 4.8: The process involved in user defined searching.

The search based system allows users to control the two steps after the shot-level feature vector has been generated. This means choosing a desired FSM, and then deciding on how much (if any) filtering to undertake on the potential sequences. So, for example if a searcher wanted to find a particular event, say a conversation that takes place in a moving car, he/she could use the speech FSM to find all the speech potential sequences, and then filter the results by only accepting the potential sequences with high amounts of camera motion. In this way, a number of events will be returned, all of which will contain high amounts of speech and high amounts of moving camera shots. The user can then browse the returned events and find the desired conversation. Note that another way of retrieving the same event would be to use the moving camera FSM (i.e. the non-static FSM) and then filter the returned potential sequences by only allowing the ones with high amounts of speech through.

Similarly, a searcher could be looking for an event in a nightclub, where two people's eyes meet and they stare at each other across the dance floor. To find this event, the potential sequences generated by the music FSM could be used, and then filtered by only accepting the sequences with a high amount of shot repetition, or only accepting sequences with a high amount of static camera shots. Figure 4.8 illustrates this two-step approach. In the first step, the set of music potential sequences is selected. Secondly, these potential sequences are filtered by only retaining potential sequences with a user defined amount of static camera shots. This results in a retrieved event list as indicated in the figure.

In some cases no filtering may be desired, and the potential sequences may be the desired events. For example, a film student may be interested in analysing how music is used in a particular film. The student could use the music FSM to find all of the areas where music is present. Thus, for this particular case, no filtering is required.

This search based system gives control of the analysis to users to allow them to find a required set of events. As the results in Chapter 5 show, the three main event classes typically cover all meaningful areas of a movie. There may, however, be occasions where the set of dialogue, exciting and montage events are not as relevant as a set of user specified events. Also, although the retrieval rate for the three event classes is very high, as detailed in Chapter 5 there are occasions where events may be missed, usually due to the event in question containing features not usually synonymous with that event class. In these cases, a search based retrieval technique is desirable. Details of how the search based system is implemented with a graphical user interface are given in Chapter 6.

4.3 A Hidden Markov Model Approach to Event Detection

Due to the fact that it is difficult to locate other approaches that attempt to create a complete movie index in existence, direct comparison between the approach in Section 4.2 and another movie summarisation approach is problematic. However, it is possible to create a different implementation and directly compare the results. It was previously noted that many video summarisation systems may have a similar approach, in that firstly, a set of features is generated for each shot, and secondly, some higher level semantics are inferred from the shot-level features. The main approach in this thesis uses finite state machines and filtering to infer these semantics.

In order to benchmark the finite state machine approach, an alternative supervised machine learning approach corresponding to a hidden Markov model (HMM) approach was implemented and is presented here. HMMs have been applied to summarise/detect important events in a wide range of video applications. [21] applied a HMM to find the structure of documentaries, for example, while [31, 37] used HMMs to detect events in sports videos. The approach presented here uses the same shot feature vector as the finite state machines, so direct comparison is possible. The shot feature vector contains the values for each shot: [% speech, % music, % silence, % silence/quiet music, % other audio, % static camera frames, % non-static camera frames, motion intensity, shot length]. The HMM approach is not as heavily based on film grammar since, rather than targeting the individual components of events, the events as a whole are examined. Also, a HMM requires training and it's reliability in classifying new data is dependent on the quality of the training data. [26] have a similar approach to the structure of video content, and aim to segment the events in a news programme using a HMM. The technique presented here is partly based on their approach.

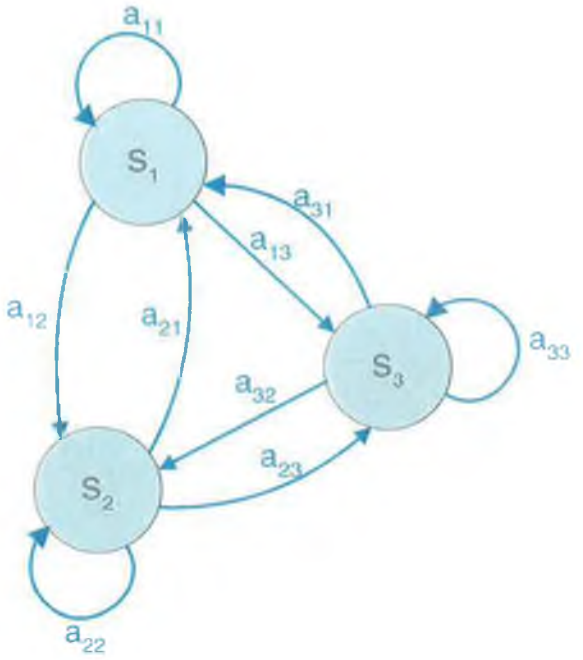

Figure 4.9: A three-state Markov model with associated transitions

4.3.1 A Brief Introduction to Hidden Markov Models

Hidden Markov Models (HMMs) have become increasingly popular in the last number of years. One area where they are prominently found is in speech/gesture recognition. It is partly due to their popularity in these area that they have begun to appear in many different applications. At the core of HMM functionality lies a set of states and transitions, rather like a finite state machine. However, unlike a FSM, HMM transitions and states have probabilities associated with them. HMMs can be used when the state at time (t) is influenced by the state at time $(t-1)$. These are *first order* HMMs, as the current state only depends on the previous state [68]. Before HMMs are discussed, an introduction to general Markov models is necessary.

A Markov model is a system which can be described as being in one of a set of N states. Periodically, the Markov model evaluates its next state. It can go from any one state to another (including remaining in the same state). This is illustrated in Figure 4.9 with a three state Markov model. The states are represented as $(S_{statenumber})$, while the notation for the probability of transition is $(a_{source,destination})$. At each time interval $(t = 1, 2, 3....)$ one of the transitions is traversed. As there is a finite amount of states, a probability matrix, A, can be generated which contains all possible transitions. So, for Figure 4.9, the A matrix could be

$$A = \{a_{i,j}\} = \begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0.6 & 0.3 & 0.1 \\ 0.1 & 0.4 & 0.5 \end{bmatrix}$$

Thus, given the Markov model is in state 2 at time t, the probability of it remaining in state 2 is 0.3, while the probability of it moving to state 3 is 0.1. Note that the rows all sum to 1, since it must enter one of the states. A set of probabilities that the Markov model will begin in a particular state are also required, this is denoted as

$$\Pi = \begin{bmatrix} 0.33 & 0.33 & 0.33 \end{bmatrix}$$

if the probability of beginning in any state is equal. The A and π matrices collectively form the model, λ. This is an *observable Markov model* since the set of states at any particular time can be evaluated, where each state is an observable event. For example, it is possible to find the probability of observing a sequence of states beginning at S_1, then going to S_2 to S_3 and back to S_2 (this is known as an *observation sequence*, $O = \{S_1, S_2, S_3, S_2\}$) by evaluating the probability:

$$\begin{aligned} P(O, A) &= P[S_1, S_2, S_3, S_2|\lambda] \\ &= \pi_1 \cdot a_{12} \cdot a_{23} \cdot a_{32} \\ &= 0.33 \cdot 0.2 \cdot 0.3 \cdot 0.4 \\ &= 0.00528 \end{aligned}$$

This model is useful for calculating likelihoods as above, however it is not possible to apply it to many real problems. In order to extend these Markov models, *hidden states* need to be introduced. These hidden states cannot be viewed directly, rather the HMM relies on the observable states to deduce the hidden state. This can be used to explain a set of observations. An example of creating a HMM based on one presented in [69] is given in Appendix C.2. Before the HMM is discussed, the necessary elements of a HMM are defined. [70] states that there are five elements of a HMM.

1. N, the number of hidden states in the model, where the hidden states are denoted as $S = S_1, S_2, ...S_N$
2. M, the number of observable states, denoted as $V = v_1, v_2, ...v_M$
3. A, the state probability distribution. The probability of being in hidden state S_i at time $(t + 1)$ given that it is in hidden state S_j at time (t).
4. B, the observation probability distribution. If it is in hidden state S_i, what is the probability of observing v_k, for $1 \leq k \leq M$
5. Π, the initial state probability distribution.

These five values form the HMM model, λ. Given λ, there are three central problems that HMMs are used to solve [68]:

1. The Evaluation Problem. Given a HMM, with all five features listed above, determine the probability of a given observation sequence. The solution to this problem may not be too important if there is only one HMM, but if there are many competing HMMs (as is often the case in speech recognition systems for example), this can be used to calculate the most likely HMM. As only one HMM is used in this work, this problem is not of great significance, however for completeness, Section C.1 of the Appendix presents a solution to the evaluation problem.

2. The Decoding Problem. Given a HMM, as well as a set of observations, determine the most likely set of hidden states that produced these observations. This is akin to asking "Given that the following observation sequence is observed, $O = (O_1, O_2, O_3, O_4, O_5, O_6)$, what is the most likely hidden sequence to occur over the six observations?". The solution of this problem uncovers the *hidden* sequence of states. In Section 4.3.2 a HMM is proposed that uses the solution to this problem in order to assist in movie summarisation.

Solution to decoding problem. There are several ways of solving this problem, however the most commonly used solution is the Viterbi algorithm [70]. This aims to find the single best state sequence (or path) which has the maximum probability of occurring given a set of observations and a model, i.e. $P(Q, O|\lambda)$, where Q is the path of hidden states $Q = \{q_1, q_2, ...q_T\}$, and O is the observation sequence $O = \{O_1, O_2, ...O_T\}$. The path with the highest probability of occurring is sought, so

$$\delta_t(i) = \max_{q_1, q_2, ..., q_{t-1}} P[q_1 q_2 ... q_t = i, O_1 O_2 ... O_t|\lambda]$$

can be defined where $\delta_t(i)$ is the best score (highest probability) along a single path, at time t, which accounts for the first t observations and ends at state S_i [70]. Essentially, $\delta_t(i)$ is the state at time t, that has the highest probability of occurring, and is therefore the state that best fits the path of highest probability.

The next state along the correct path is therefore

$$\delta_{t+1}(j) = \max_i [\delta_t(i) a_{ij} b_i(O_{t+1})]$$

which finds the next state with the highest probability, given the current state and the next observation. To retrieve the final state sequence, a backtracking step is required, this is simply the argument that maximises the previous equation (denoted as $\psi_t(j)$). There are four steps involved in solving the decoding problem, *Induction, Recursion, Termination* and *Path Backtracking*.

Induction simply finds the initial state with the highest probability of occurring

$$\delta_1(i) = \pi_i b_i(O_i), \quad 1 \leq i \leq N$$

$$\psi_1(i) = 0$$

The *recursion* step finds the best path, while keeping note of the maximum of the argument

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_t - 1(i) a_{ij}] b_j(O_t), \quad 2 \leq t \leq T$$

$$\psi_t(j) = arg \max_{1 \leq i \leq N} [\delta_t - 1(i) a_{ij}], \quad i \leq j \leq N$$

The *Termination* step finds the last state of the path

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

Finally, the path *backtracking* step goes back through the hidden states and finds the correct path:

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

It is important to note, that the Viterbi algorithm does not simply accept the most likely state for a given time instant, but it takes a decision based on the whole sequence. Therefore if there are unlikely events midway through the data, this will not result in incorrect classification, provided the whole context of what is seen is reasonable [69].

3. The Learning Problem. Given the number of visible and invisible states, but not the transitional probabilities, how can the matrices holding a_{ij} and b_{ij} be determined. This is used to *train* a HMM.

Solution to learning problem. There is no known method for finding the optimal set of parameters for finding a_{ij} and b_{ij} from a set of training data, but it is possible to determine a good solution using the forward-backward algorithm (also known as the Baum-Welch algorithm). As the solution to this problem of not of concern here, the solution will not be explained. Interested readers are advised to consult [68] or [70] for a complete solution.

HMM Conclusion

By using the solutions to the three problems listed in this section, HMMs can be used in a range of applications. HMMs and FSMs are quite similar in many ways. However, the effectiveness of HMMs relies on the quality of the training data (which may not always be perfect), while the FSMs effectiveness relies heavily on the expertise of the designer.

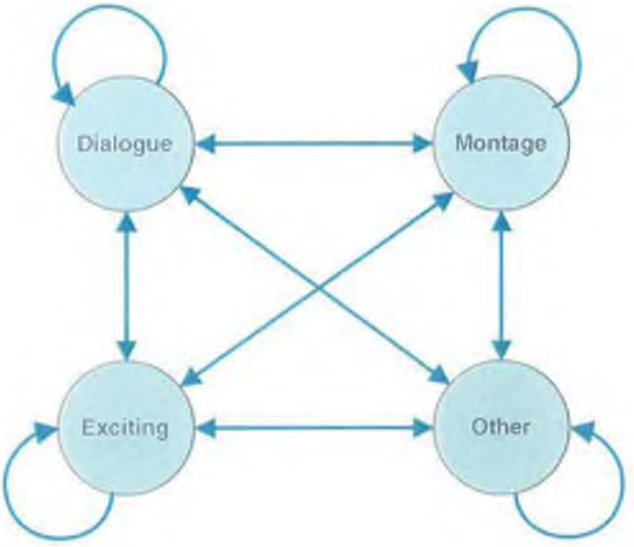

Figure 4.10: The HMM for detecting dialogue, exciting, montage and other events

4.3.2 HMM Approach

A high-level approach to movie summarisation, based in part on previous HMM summarisation methods applied to news data [26], is employed. In this system, the hidden states are the events themselves. Following the FSM based approach, four hidden states are deemed necessary; dialogue, exciting, montage and other. The 'other' state is necessary as there are occasions in a film where it is not in any of the other states (voiceovers, credits, short parts of the movie etc.). The HMM will always be in one of these four hidden states. The HMM is shown in Figure 4.10.

The observations are the extracted features that form the shot feature vector. A number of observable states are generated that reflect the characteristic features of the events to be detected. Each event class has a number of observable states assigned to it. In Chapter 3 the features used to detect events in the three event classes were described, and a shot-level feature vector was created. In order to detect dialogue events, there are three features from the shot-level feature vector that can be used, the speech value, the camera movement value and the shot-length value. Thus, the speech-like observable states are made up of all possible combinations in which at least 2 of these features are present. As can be seen below, states 1-4 contain all of these variations. Note that these features are the same as the ones used to detect dialogue events using FSMs. As with the FSM approach, in order to detect exciting events short shots with high motion are sought. Thus, states 5, 6, and 7 correspond to the combinations of shot types that typically occur during exciting events. Finally, as montages contain music or silence and usually contain a relaxed editing style, states 8 and 9 reflect these features. State 10 contains all other observations. Note that a shot can only be in one observable state, and it is put in the state with which is has the most features in common. So, for example, while a shot that contains the features

of observable state 1 could also be placed in observable state 2, it has more features in common with observable state 1 (3 features) than observable state 2 (2 features), so it is placed in the first observable state. Each observable state, and its associated features, is presented below. Each feature threshold value is the same as for the FSM based approach.

EVENT = OBSERVABLE STATE

Dialogue = (1)High speech, Low camera movement, Long shot length
(2)High speech, Low camera movement
(3)Low camera movement, Long shot length
(4)High speech, Long shot length

Exciting = (5)High motion, Short shot length
(6)High motion
(7)Short shot length

Montage = (8)High non-speech, Long shot length - i.e. un-action type shots.
(9)High non-speech

Other = (10)All other shots

4.3.3 Training and Implementation of HMM

As described previously, there are two stages in using a HMM; a training stage and a classification stage. As stated in Section 4.3.1, the generation of three sets (matrices) of probabilities is required in the training phase. The probability matrices are created by examining a training set of known data. These matrices are then used in the classification stage on a set of unknown data. The three necessary matrices are as follows:

1) The probability of starting in hidden state X. i.e. probability of the first shot in the movie being a dialogue/exciting/montage/other shot. These values form the π *Matrix.*

2) Given that model is in a particular hidden state at shot T, what is the probability of the HMM being in a different hidden state at shot T+1. i.e. if the model is in the dialogue state at shot t, for shot $t + 1$, what is the probability of it (a) staying the dialogue state (b) going to the exciting state (c) going to the montage state (d) going to the other state? Typically the probability of (a) will be quite high, while (b), (c), and (d) will be smaller. These probabilities make up the *A Matrix.*

3) Given a particular observation, what is the probability of the HMM being in a particular hidden state. i.e. in shot X, if state 5 is observed, there should be a high probability that the HMM is in the hidden action state, while there will be a lower probability that the observation belongs to a dialogue or montage event. These probabilities make up the *B Matrix.*

Generating the first set of probabilities is straightforward. An even distribution of events at the start of a movie was observed in a representative test corpus, so the probability of beginning in any of the four hidden states is set to be equal. For the second set of probabilities, a subset of the manually marked up observations (as described in Section 5.2) are used. In this marking, each dialogue, exciting and montage event in a movie is manually annotated. If a shot is not marked into one of these events, then it is labelled as other. Therefore each shot in a movie is assigned into one of the four hidden states. The fundamental timing unit of the HMM is the shot, so each time there is a new shot in a movie, the states are re-evaluated. Thus, it is possible to deduce these probabilities by examining an annotated movie and noting the pattern of hidden states. For example, in the movie *Snatch* it was noted that the movie was in the dialogue state for 668 shots. Of the 668 times it was in a dialogue state, 638 times the next state was the dialogue state also. Therefore, given that the movie is in a dialogue state at shot x, there is a 95.5% chance of the shot $x+1$ being dialogue. A transition from a dialogue state to an exciting state only occured 4 times, so there is only a 0.61% probability of the HMM treading that path. Similarly, there is a 0.59% probability of a transition from dialogue to montage, and a 3.3% probability of a transition from the dialogue state to the other state. These results were calculated for each of the other hidden states and the A matrix is generated for that particular film. The final A matrix is generated by taking a representation from a number of films of different genres including a drama, an action film, and a thriller. Four different movies (*American Beauty, Goodfellas, Reservoir Dogs, and Snatch*) are used when generating the A matrix which is shown below.

A Matrix:	Dialogue	Exciting	Montage	Other	
Dialogue	[95.4%	0.5%	0.6%	3.5%	]
Exciting	[1.5%	90.2%	2.4%	6%	]
Montage	[1.8%	1.1%	93.7%	3.5%	]
Other	[13.4%	2%	1.3%	83.3%	]

The values on the left represent the current state, and the values on the top represent the probability of the HMM going into that state in the next shot. So there is a 1.8% chance of transition from a montage state to a dialogue state. As can be seen, the probability of staying in the same state is quite large for all hidden states. This is to be expected, since typically events last for a large number of shots, and only change to a different hidden state once (at the end of the event).

In order to generate the B matrix, a similar approach is taken. Given that the movie is in one of the hidden states, the probability of observing each of the observable states is calculated. So, again taking the film *Snatch* as an example, there were 282 exciting shots

Observable State	Num. Exciting Shots	Percentage
1	3	1.1%
2	17	6%
3	0	0%
4	9	3.2%
5	126	44.7%
6	66	23.4%
7	45	16%
8	5	1.8%
9	1	0.4%
10	10	3.6%

Table 4.1: Exciting Observable States For the Film Snatch. These are used to generate the B Matrix.

in total (i.e. 282 shots were manually annotated as being part of an exciting event). The observable state and number of exciting shots is shown in Table 4.1: So, in Snatch, 126 times out of the 282 (44.7%) times the movie is in an exciting state, the observable state is state 5 (high motion with short shot length). Also, only 1.1% of the exciting shots has the characteristics of observable state 1 (i.e. only 1.1% of shots in an exciting event have high speech, low camera movement and a long shot length) A set of probabilities is again generated for four films, and the final B matrix is generated, as shown below.

B Matrix:

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Dialogue	[30.5%	31.7%	0.01%	6.6%	3.5%	4.4%	7.6%	1.5%	0.7%	13.6%]
Exciting	[3.7%	14.4%	0.01%	7.9%	31.1%	28.2%	8.5%	1.3%	0.9%	4%]
Montage	[3.2%	4.7%	0.01%	4.5%	16.2%	18.3%	12.3%	14.7%	7.3%	18.6%]
Other	[19.9%	16.3%	0.01%	16.6%	7.5%	12.5%	4.8%	2.7%	2.2%	17.6%]

Note that the highest probabilities are contained in the observable states that reflect the common characteristics of the hidden events. For example, the observable states 5, 6 and 7 were designed to be exciting-like states, and 67.8% of the time when an exciting event is taking place in the training movie, the observable state is either 5, 6 or 7 (67.8% = 31.1% from state 5, 28.2% from state 6, and 8.5% from state 7). Of course, in any exciting event there will be shots that are in different observable states, but the probability of these occuring is lower. Note, for example, how the probability of a dialogue shot, belonging to observable state 5 (an exciting like state) is quite low at 3.5%.

Now that the three required matrices have been found, the HMM training stage has been completed. The viterbi algorithm is now used to find the most likely set of hidden states for a set of new observable data. For each movie, a set of input values is created and used to detect the dialogue, exciting, montage, and other events. Results for this approach can be seen in Chapter 5.

4.4 Summary

This chapter described the high-level event detection module of the overall event detection system. This module uses the shot-level feature vector as input and then, using these features, creates a list of dialogue, exciting and montage events. The proposed approach firstly creates a set of potential sequences, generated using an array of FSMs, and secondly, combines and filters the potential sequences in order to create an event list. Based on this structure, a method of searching video was also proposed. In order to benchmark the FSM approach, a HMM based approach to event detection was also presented in this chapter. The results of both of these high-level event detection methods are investigated in Chapter 5.

CHAPTER 5

Event Detection Results & Analysis

5.1 Introduction

This chapter contains a thorough investigation of the effectiveness of the event detection system. The main focus of this chapter is in evaluating the finite state machine based approach to event detection. Section 5.2 examines the reliability of the finite state machine approach in detecting each of the three event classes in a movie. This is achieved by comparing the automatically generated results against a manual annotation. As the design process of the FSMs presented in Chapter 4 involved examining the manual annotation presented in Section 5.2, the full design process could not be explained in Chapter 4. Thus, Section 5.3 describes how the FSMs were designed. In order to remove any unintentional bias that may exist in the annotation of Section 5.2, Section 5.4 presents a set of results for movies marked up by a set of volunteers. Given the generated event lists, the fourth Section (5.5) examines the overlap between detected events (i.e. cases where shots were labelled as being part of more than one event). The shots that are not labelled as being part of any event are also examined. The fifth Section (5.6) examines how the finite state machine summarisation approach can be applied to video data other than movies. Section 5.7 contains the results of the hidden Markov model approach to event detection, and comparisons between the two systems are drawn. Finally, conclusions drawn from the results are presented in Section 5.8.

5.2 Reliability of Event Detection

5.2.1 Generation of the Ground Truth

A set of event start and end times for the three event classes are automatically generated using the techniques described in Chapter 4. In order to test the reliability and validity of the approach, an extensive set of experiments and tests were undertaken. These experiments aim to evaluate the effectiveness of the event detection system. A number of movies from various genres were chosen as a test set. In order to asses the generic nature of the event detection techniques the movies were carefully chosen to represent a broad range of styles and genres. Within the test set, there are a number of comedies, dramas, thrillers, art house films, animated and action films. Many of the films have vastly different audiences, ranging from animations suitable for children, to violent action movies only suitable for adult viewing. Although the fundamental properties of movies are universal to all movie types, there may be differing styles depending on cultural influences, thus, the movies in the test set were chosen to represent a broad range of origins, and span different geographical locations including America, Australia, Japan, England and Mexico. The test data in total consists of ten movies corresponding to over eighteen hours of video.

In order to create a ground truth with which to compare the automatically generated results, each movie was manually annotated, and the start and end time for each dialogue, exciting, and montage[1] event was located. All other parts of the film were labelled 'Other'. As mentioned in Section 2.4.1, some events can be interpreted in more than one way (i.e. an emotional conversation could be classified as a dialogue event or a montage event). In cases such as this, the class which the annotator felt the event was most appropriate was chosen. Dual classification of events is further examined in Section 5.4.

As was previously mentioned when defining an event (Chapter 2), when annotating the ground truth, one guideline followed is that events must be at least 5 shots long. This is so that short, insignificant sequences of shots are not marked up (e.g. 2 characters saying a passing 'hello' does not constitute a dialogue event).

In accordance with its definition, any conversation involving two or more characters was annotated as a dialogue event. This includes a person addressing a crowd, group conversations, as well as conversations involving only two people. Exciting and montage events however, cannot be defined as strictly, so a certain amount of discretion was required in order to annotate these events. The guideline followed was that areas of the film in which the annotator felt excited were marked as exciting events. These included, among others, fights, chases, robberies, and verbal arguments. For the montages, as there are three different types of events in this class, there are more possible scenarios. All areas

[1] Where musical, montage and emotional events are all labelled under the umbrella of 'montage' events

where the annotator noted a highly emotional context were marked as montages. For example romantic, sad, scared, happy, and distressful events within a movie were included in this set. Although excitement is also an emotion, these events were deemed more aligned with the exciting event class. Also marked as montages were areas where music is central to the event, such as dances, or musical performances. Finally, all montages were marked in the montage class. Note that each event in the movie was only marked into one event class. There were, however, many instances where a particular event had characteristics of more than one event class. In these cases, the event was placed in the class with which the annotator felt it has most in common with.

5.2.2 Evaluation Against the Ground Truth

Given a set of manually annotated events for each film, it is possible to compare this manual annotation to the automatically generated event list. In order to quantify performance, the following guidelines were adopted: if the majority of shots in a manually generated event were detected by the event detection system, then it is labelled as a true positive. For example, if an event lasts three minutes, and only two and a half minutes of the event is detected by the system, it is labelled as a true positive. This is because the main purpose of the event detection system is to aid in retrieval of events. Even if part of an event is missed, the detected portion will still accurately represent the event, as the same characters and activities will be present. Therefore, users of a retrieval system can still locate the desired event, even if part of the event is missed (this is demonstrated experimentally in Section 6.3).

Precision and recall are used in order to measure of the performance of the system. Recall is defined as the (*number of correct events detected*) / (*number of events in ground truth*), while precision is defined as (*number of correct events detected*) / (*total number of events detected*). Although precision and recall are common metrics for measuring performance of a retrieval system, and are an accurate measure of the relationship between the detected events and the ground truth, there are some cases where they are not entirely suitable metrics for this application. The event detection system is used to create an event-based index of a movie, with the ultimate aim of allowing efficient user retrieval of events. A high recall value is always desired, as a user should always be able to find a desired event in the returned set of events. As is explained in the following sections, there are occasions where the precision value for certain movies is quite low, as there are more detected events than relevant ones. However, in some movies there may be very few events in any particular event class. For example, some movies may only contain two exciting events, so if, say, eight exciting events are detected, a precision value of 37.5% will result. Although this precision value is quite low, in terms of an indexed movie, browsing eight events is efficient. Also, as illustrated in Section 5.4, different users often

Film Name	Dialogue			Exciting			Montage		
	Num. Events	Prec.	Recall	Num. Events	Prec.	Recall	Num. Events	Prec.	Recall
American Beauty	44	86%	96%	2	17(29)%	100(100)%	19	71%	95%
Amores Perros	49	56%	84%	19	56(67)%	95(82)%	24	55%	96%
Battle Royal	35	62%	94%	33	71(75)%	91(82)%	10	72%	90%
Chopper	36	90%	94%	6	22(43)%	83(80)%	2	50%	100%
Dumb & Dumber	36	74%	91%	14	55(79)%	100(79)%	15	68%	86%
Goodfellas	57	67%	95%	10	46(53)%	90(89)%	7	60%	86%
High Fidelity	44	80%	100%	3	17(29)%	100(100)%	6	56%	83%
Reservoir Dogs	14	89%	94%	5	50(100)%	80(60)%	3	100%	100%
Shrek	32	73%	97%	14	58(80)%	100(80)%	12	67%	75%
Snatch	62	84%	97%	21	71(85)%	100(84)%	6	67%	83%
Average	**41**	**76%**	**94%**	**13**	**57(67)%**	**95(82)%**	**10**	**73%**	**90%**

Table 5.1: Results of event detection using the authors ground truth.

have differing opinions on a movie, which results in different interpretations of events, this means that some browsers may consider a particular event to be a dialogue event, while others may consider it to be an exciting event (an arguement, for example). Thus, in order to facilitate both interpretations, events such as this should be detected by both the exciting event detector, and the dialogue event detector, which will again decrease the precision value for any one interpretation. Clearly, if the precision value becomes too low then this would mean results that are difficult to browse, however, a number of 'false positives' are actually desirable for any one interpretation.

If a movie contains a large amount of events in a particular event class, then increased precision is required. For example, if a movie contains 70 dialogue events, then a precision value of 37.5% would be unacceptable in a browsing scenario, as this would mean 186 returned dialogue events, 116 of which are incorrect, which would not allow for efficient retrieval. For this reason, the event detection process was geared toward an efficient indexing solution, corresponding to high recall with slightly lower precision. The results of this process are presented in Table 5.1. In the following sections, the results for each event class are analysed.

5.2.3 Dialogue Events

As can be seen from Table 5.1, the dialogue events have both high recall and precision values. In all movies except one, the recall is at least 90%, and the average recall is just over 94%. Thus, it can be concluded that the system obtains consistantly high recall accross all movie types. For example, the recall values for Shrek (a children's movie made in Hollywood), and Battle Royal (a violent action film made in Japan), are both high. Whilst the variation in the precision from movie to movie is greater, an average

value of 76% indicates efficient performance.

There are a number of reasons why the system may miss a dialogue event. As explained below, the events that are not detected usually have characteristics that are not common to dialogues e.g. some events have a high CS ratio (i.e. low amount of shot repetition) and therefore are rejected. Other dialogue events contain low amounts of speech, for example somebody crying during the conversation, and the sequence of shots is therefore not detected by the speech FSM. Alternately, some dialogue events may contain excessive motion, and will therefore be rejected. In total for all films, there are 409 manually marked up dialogues, 386 of which were detected by the system. Of the 23 missed dialogues, 11 were missed due to the presence of a moving camera with little shot repetition, 6 were missed due to a moving camera with little speech present (either due to audio misclassification or long gaps of silence), 5 were missed due to little speech present with a low amount of shot repetition, and 1 was missed due to a combination of a moving camera, incorrectly classified speech, and a low amount of shot repetition.

The film Amores Perros proved to be the most challenging film overall due to its unique directorial style. The director chose to shoot the film using a free-hand camera style. This shooting style results in a significant amount of camera movement throughout the film, as the camera moves in a similar way to a personal hand-held camera. This means that very few dialogues contain a static camera. Also, there was a low amount of shot repetition as the camera moved freely around the set. A number of conversations were missed due to these effects. The lack of static camera shots in particular results in over reliance on the speech FSM to detect all of the events. Despite this, 84% of the dialogue events for this movie were detected. This illustrates the versatility of the approach, as even in a movie with a unique approach to shooting dialogues, the vast majority of these dialogues are still detected.

In the film American Beauty, one of the missed dialogues is an emotional conversation between two of the main characters. There is music playing in the background, and the amount of speech is limited, which resulted in the shots being labelled as music. Also, the director aims to intensify the conversation by changing the camera angle a number of times, focusing closer and closer on the speaking characters. There is no doubt that this is an emotional conversation, so it could possibly be labelled as a montage (and indeed it is detected by the montage system), however there is enough conversational elements for it to be deemed an undetected dialogue event, and therefore should have been detected by the dialogue event detection system.

Most of the other films follow in the vein of American Beauty, with very few dialogue events not detected. In the film Chopper, there are two missed dialogues, both are quite short and occur in a nightclub with music and flashing lights. In Dumb and Dumber a number of dialogues take place in a moving car (which contains a moving camera).

Although many of the in-car dialogues are detected using the speech FSM, in some cases there is music playing in the background. This combined with a low amount of shot repetition (as the background changes), means that two of these dialogues were missed. However, since many other conversations take place in a moving car, and are detected, this still indicates high performance overall.

There are also a number of occasions where the system detects a dialogue event that is not present in the manually marked up annotation and this results in lower precision values. Many of these false detections are due to an event occurring that may have elements from more than one event class, e.g. an emotional conversation could be a montage or a dialogue event. The film American Beauty has many such occurrences. In the film Snatch, on one occasion a bare knuckle fight is about to commence. The two fighters briefly talk to each other for a number of shots. This was manually marked as an exciting event, but also contains elements of dialogue, and was detected by the dialogue system. Later in the same film, two thieves are robbing a betting office. During the robbery one of them shouts demands at the cashier. Although, in the setting it is part of an exciting event, again there are elements of conversation, and these parts are detected as dialogue events.

Finally, a dialogue event might be shot in a similar way to another event type. In the film High Fidelity for example, there is a sequence of shots that contains speech and a static camera, but is part of a voice over, and is not a conversation. Also in the same film, there are a few occasions where the main character talks directly to the camera. These are falsely detected as dialogue events. Typically these monologues occur quite infrequently in a movie, and in most movies do not occur at all. Therefore these are treated as false positives. It could be possible to extend the definition of a dialogue to include monologues, but as they rarely occur, this was deemed unnecessary. Similarly, in Dumb and Dumber, one sequence of shots, that actually makes up a montage, contains a static camera with a high amount of shot repetition and is detected as a dialogue event.

5.2.4 Exciting Events

Note that there are two values given for exciting events. The first values are the results without post-processing, while the values in brackets are the results after additional filtering (as detailed in Section 4.2.4). The filtering step aims to reduce the number of false positives. Typically the films with low precision values are films with a very low amount of action, thus, any false positives impact heavily in reducing the precision.

Overall, the detection rate for the pre-filtered results is quite high, resulting in very high recall, with all but 2 being above 90% and a number having 100% detection rate. In total for all films, there are 127 manually marked up exciting events, 120 of which are successfully detected. Of the 7 missed exciting events, 4 were missed as they were quite short (6/7 shots long) and were not detected by the FSM, 2 were missed due to slow paced

editing, and 1 was missed as it had high amounts of shot repetition.

The lowest recall rates are obtained by Reservoir Dogs and Chopper. In both of these films however, only one exciting event is missed, as there are only 5 (in Reservoir Dogs), or 6 (in Chopper) exciting events in total. Of the missed events, some of them are shot with a static camera, which leads to a low motion intensity value. A few exciting events are shown in slow motion which resulted in longer shots and low motion. Also, in Battle Royal, one of the missed events is a basketball match played in slow motion, which does not contain a high amount of motion intensity. In Amores Perros, the only exciting event not detected is quite short, and therefore rejected as it is detected as being less than five shots long. Similarly, in Goodfellas, the only exciting event not detected is a short sequence of shots that only lasts seven seconds and is rejected as being too short. As can be seen from the results, these short exciting events occur quite infrequently. Any attempt to detect them would result in a large decrease in the amount of false positives as many non-exciting portions of the movie would also be detected.

The precision values for these events vary quite substantially. In some movies the precision value is over 70%, while in others it is lower than 20%. The low values can be explained by low amounts of exciting events in some movies, where a few false positives can lead to very low precision (such as in American Beauty where there are only two manually annotated exciting events). Similarly, in many slow paced films, directors may shoot parts of the film in an exciting style in order to keep the attention of the viewers. For example a dialogue may be shot with elements of motion and with a fast shot cut rate. Many of the false positives in American Beauty, High Fidelity and Chopper are due to this. Although they may not fit in with the annotator's definition of an exciting event, they usually constitute the most exciting events in the film. In Reservoir Dogs, all of the falsely detected exciting events occur during intense conversations, or during conversations with a moving camera.

Some exciting events that are detected contain exciting elements, but were not manually annotated. In Snatch, for example, a number of events manually labelled as montage events could just as fairly have been labelled as exciting events. Some of these were detected as exciting events which reduced the precision. Also, many of the false detections in the original results are due to short sequences of shots that contain exciting elements. These are usually part of a larger event, and do not contain enough excitement to be classed as an exciting event. Examples include character movement at the beginning of a conversation and a panning camera during a montage. This is a common cause of false positives across all films.

There is a clear trade off involved if an attempt is made to improve precision. In order to remove false positives, a more stringent filtering process is undertaken (as explained in Section 4.2.4). This results in the removal of many false positives. However, it also

unintentionally removes a small number of true positives. This can be seen in the second set of results for the exciting events. After this filtering, the average precision rises from 57% to 67%, while the average recall drops from 95% to 82%. Many of the filtered events are short sequences of shots that have increased editing pace, but are not exciting events, for example the end of a conversation where the pace is increased to emphasise intensity in the conversation. Also, a number of the filtered events are ambiguous events that could be labeled into more than one event class. Although the increased precision is desireable, typically only a small number of exciting events are detected and therefore presented to a user, so a number of false positives can be tolerated. It is deemed more important that all possible events are shown to a user, rather than a reduced set of correct ones. Thus the pre-filtered set of results are deemed more appropriate.

5.2.5 Montage Events

The recall of the montage event detection system is quite high, with all of the recall rates except one above 80%. This high recall is largely due to the filmmakers reliance on the use of music in these situations. In total for all films, there were 105 manually marked up montage events, 94 of which were automatically detected by the system. Of the 11 missed montage events, 8 were missed due to incorrect audio classification, and 3 were missed due to the presence of fast paced editing and camera movement (and were classified as exciting events). For example, in the film High Fidelity, the only montage event missed is a musical event in which a musician performs a song a capella. The relevant shots are detected as speech instead of music, so the event is missed. In the film Goodfellas, the only missed montage event occurs during a wedding, where a series of envelopes are passed to the bride. As each envelope is passed, a shot cut and some movement occurs, which leads to the montage event being detected as an exciting event. Similarly in Battle Royal the only missed montage event, takes place at a high tempo and is detected as an exciting event.

The most common reason for false positives is speech being classified as music. This usually only occurs if there is music playing in the background. Also, some dialogue events or voice overs have very slow paced audio where there may be only a few words spoken in each shot, this leads to shots being classified as silent (or music if there is music in the background), which leads to incorrect montage events being detected. As mentioned previously, many events labelled as montages by the author may be labelled as exciting/dialogue events by another annotator (and indeed are by the annotators used in Section 5.4). Many of the montage false positives contain elements of action, which lead the annotator to label them as exciting events. In Dumb and Dumber for example, one particular event occurs as loud music plays and one of the characters collapses. This was labelled as an exciting event, but could have been interpreted as a musical event.

In general, for all event types, when the minimum event length is set to be shorter than five shots (i.e. when the minimum event length is three or four shots), this results in an increased amount of false positives. For example, shots of a character walking across a room during a conversation may be detected as an exciting event due to the small amount of increased editing pace and camera movement. Conversely, when the minimum event length is increased to be longer than five shots, this results in a number of correctly detected events being rejected as they are deemed too short. Overall, the results for all three event detection techniques are quite high. The average recall for all event classes is above 90% (not including the filtered exciting events) and the precision level varies from 57% to 76%. This precision level is actually a benefit overall, as explained in Section 5.4.

5.3 FSM Design Justification

This section describes how the FSMs, which were initially presented in Section 4.2.2, are designed. The FSM design process was not previously fully explained as it relies on manually annotated data only present in this chapter. However, as the generation of this data has now been described, the FSM configuration process is now presented. As mentioned in Chapter 4, the chief variant in the design of the FSMs is the amount and placement of the intermediate states (I-states). In order to find the optimal number of I-states in each FSM, varying configurations of the I-states were examined, and the configuration which resulted in the highest overall performance was chosen. To illustrate this approach, the design process of the non-speech FSM is explained here. The design of the other FSMs follows in the same manner.

Firstly, a representative set of four movies was chosen. The movies are: American Beauty, Dumb and Dumber, Reservoir Dogs and Shrek. The four movies are selected in order to represent a broad range of genres. Secondly, a number of non-speech FSMs were created, each with a different configuration of I-states. Each configuration of I-states shows how the FSM is effected when I-states are added or taken away. For example, two of the non-speech FSMs that were examined are shown in Figures 5.1 and 5.2. Recall from Section 4.2.2 that the bottom I-states make it difficult for the beginning on a potential sequence to be declared, while the top I-states make it difficult for the end of a potential sequence to be declared. Thus, the FSM in Figure 5.1 makes it easy to declare the beginning and end of a non-speech potential sequence as there is only one I-state at the bottom and the top of the FSM. The FSM in Figure 5.2 has more I-states, and thus it is more difficult to declare the beginning/end of a non-speech potential sequence.

By comparing the potential sequences that are generated by each FSM to a manual annotation, it is possible to determine the best configuration of I-states. As the non-speech FSM is used in order to locate montage events, the manually annotated montage events

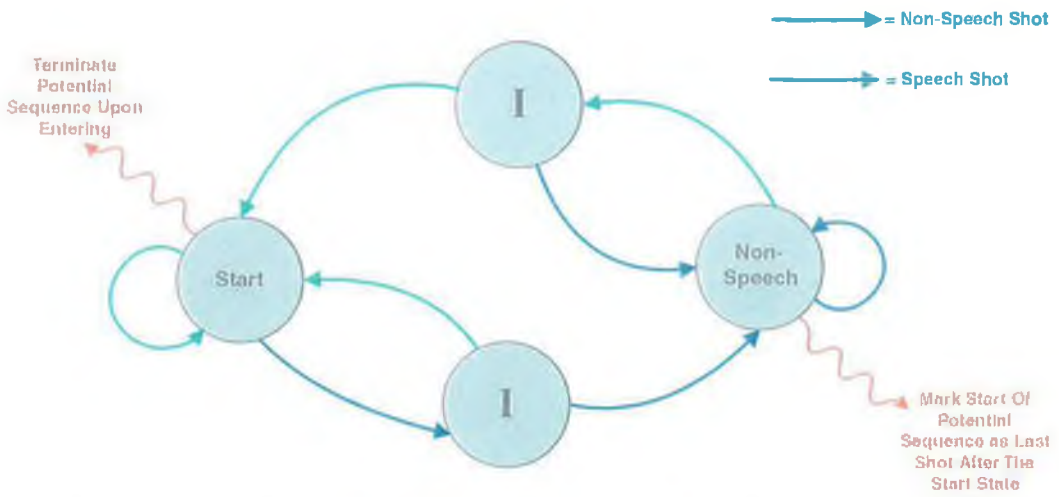

Figure 5.1: Possible configuration of the I-states for non-speech FSM design

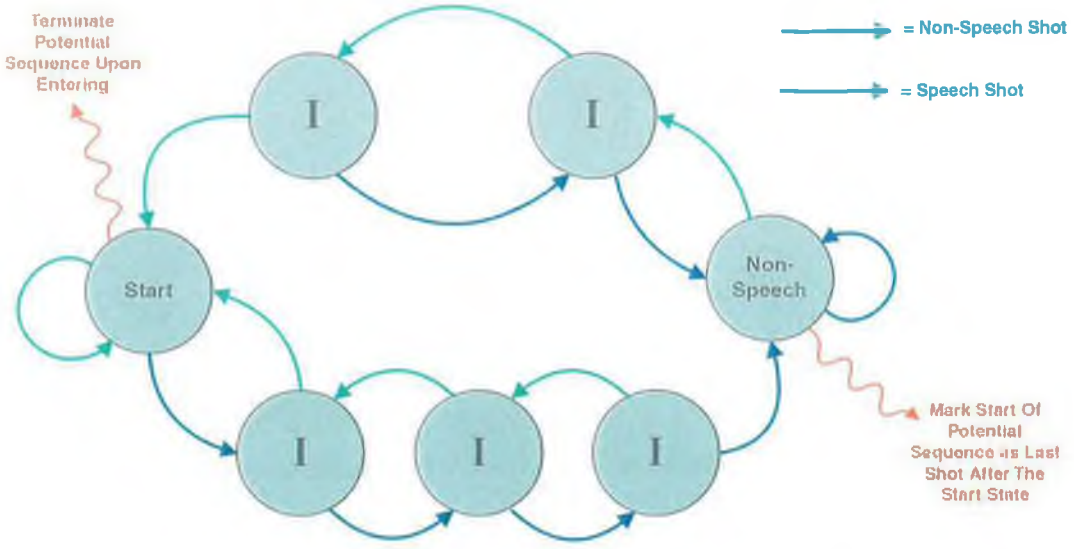

Figure 5.2: Possible configuration of the I-states for non-speech FSM design

Film Name	1		2		3		4	
	P%	R%	P%	R%	P%	R%	P%	R%
American Beauty	61	70	71	95	41	90	61	80
Dumb and Dumber	65	73	68	87	42	87	61	87
Reservoir Dogs	100	100	100	100	60	100	75	100
Shrek	67	67	67	75	58	83	61	75
Average	66	72	**71**	**88**	47	88	62	82

Table 5.2: Results of four non-speech FSMs evaluated against the montage ground truth

(as presented in Section 5.2) are used in order to assess each non-speech FSM. In Section 5.2, the method of comparing automatically generated results with manual annotation was described, and that process is also utilised here. In this process, the potential sequences generated by the non-speech FSM are compared with the manually annotated montage events, and precision and recall values are obtained. The FSM that results in the highest overall values is then used as the non-speech FSM. The results for the different FSMs can be seen in Table 5.2. Note that this process only examines the non-speech FSMs, and thus the values are different to the ones of Section 5.2, where the montage events were detected using a combination of features and filtering. As mentioned previously, more importance is placed on the recall value as it is imperative that as many events as possible are detected. Based on this, FSM number 2 was chosen as the non-speech FSM as it achieved the highest results. In fact, this is the FSM in Figure 5.2. In total four different non-speech FSMs were created for examination. Although this clearly does not represent all possible configurations of I-states, enough configurations are examined so that the behaviour of the FSM with different configurations can be inferred. For example, the FSM in Figure 5.1 is system 3 in Table 5.2. Due to the fact that there is only one I-state on the bottom of this FSM, it is quite easy for the beginning of a non-speech potential sequence to be declared. For this particular feature, this results in a low precision value (47%). If this I-state was removed to leave a direct path between the 'Start' and 'Non-Speech' states, it can be seen that this would result in even lower precision. Thus, examining this configuration is not deemed necessary. Many other possible configurations were not examined for this reason.

All of the other FSMs used in the event detection process were designed in a similar manner. Tables 5.3, 5.4 and 5.5 show the results for the other FSMs used in the event detection process, the speech FSM, the static camera FSM and the high motion/short shot FSM. In all cases, the values in bold represent the FSM that obtained the highest results and was thus used to generate the potential sequences (as used in Section 4.2.2). In most cases the final FSM reflects film grammar theory presented in Chapter 2. For example, the highest performing speech FSM (system 3) contains only one I-state on the bottom which makes it easy to declare the beginning of a speech potential sequence. This reflects the fact that the presence of speech is usually a good indicator that a dialogue is taking place.

Film Name	1		2		3		4		5		6	
	P%	R%	P%	R%	P%	R%	P%	R%	P%	R%	P%	R%
American Beauty	97	57	97	52	94	76	94	76	93	72	93	78
Dumb and Dumber	84	69	87	69	92	81	79	78	87	78	84	80
Reservoir Dogs	95	83	95	83	96	83	90	78	91	83	91	83
Shrek	85	64	84	64	79	77	72	69	81	72	81	72
Average	90	66	90	64	**89**	**79**	83	75	88	75	87	78

Table 5.3: Results of six speech FSMs evaluated against the dialogue ground truth

Film Name	1		2		3		4	
	P%	R%	P%	R%	P%	R%	P%	R%
American Beauty	88	81	91	67	83	81	85	81
Dumb and Dumber	92	61	100	44	100	64	92	58
Reservoir Dogs	100	72	100	67	96	72	100	72
Shrek	80	69	77	62	76	64	71	64
Average	**88**	**72**	86	60	84	71	87	71

Table 5.4: Results of four static camera FSMs evaluated against the dialogue ground truth

Whenever there is an area of speech present in a movie, it is usually quite significant, as a small amount of speech shots surrounded by non-speech shots usually has no purpose in movie and are indeed quite uncommon. Therefore, a small number of speech shots in a row will be classed as the beginning of a speech potential sequence. When a large number of I-states are placed on the bottom of the speech FSM (system 2 for example), it makes it more difficult for a speech potential sequence to be declared, which in turn means that these FSMs miss a number of dialogue events. As can be seen, this results in a lower recall value. Similarly, when detecting the end of a speech potential sequence, the feature characteristics dictate how many I states are optimal in the top of the FSM. Any pause in a conversation will only last a very small number of shots, so whenever any significant number of non-speech shots occur it is a strong indication that this particular speech sequence is at an end. So only a small number of I states are required in the top half of the speech FSM.

Each of the FSMs used in Section 4.2.2 was generated in this manner. There are also

Film Name	1		2		3		4	
	P%	R%	P%	R%	P%	R%	P%	R%
American Beauty	11	100	11	100	7	100	10	100
Dumb and Dumber	56	100	54	100	47	100	61	100
Reservoir Dogs	35	80	36	80	24	80	27	80
Shrek	47	93	41	93	39	93	36	93
Average	**43**	**94**	38	91	31	94	40	94

Table 5.5: Results of four HMSS FSMs evaluated against the exciting ground truth

two FSMs that are not used in the event detection process, but only for searching video (the music FSM, the non-static camera FSM). As there is no manual annotation available for the potential sequences generated by these FSMs, the same design process could not be undertaken. However, some empirical analysis was carried out, and a number of different configurations of the I-states were examined before the final music and non-static FSMs were chosen.

5.4 Alternate Interpretation of Events

5.4.1 Alternate Annotation

As discussed previously in Section 2.4.1, there is significant subjective user interpretation involved in terms of determining what constitutes a dialogue, exciting or montage event in the generation of the ground truth. The results presented in Table 5.1 are one interpretation of the ground truth for the films under test. Clearly, for a true indication of performance, implicit bias in the generation of the ground truth (if any) should be eliminated. To this end, user tests were conducted on a subset of the movies whereby six volunteers were asked to manually mark up all of the dialogue, exciting and montage events that he/she deemed worthy of retrieval, i.e. events that they would expect to be either included in a summary of the movie and/or be useful in browsing the content. The volunteers were presented with a set of guidelines in order to annotate the movie. The minimum event length was set at five shots. Dialogue events were defined as conversations between two or more people. As exciting and montage events are more subjective, user discretion was advised. The volunteers were asked to label the portions of a movie in which they felt excited as exciting events. Similarly, areas in which the volunteer felt there was a high emotional context, a montage, or singing/dancing, were marked as montage events. Each volunteer was assigned one movie to annotate. Results obtained against the ground truth created in this manner are shown in Table 5.6.

The results of the dialogue event detection system are quite high, with a recall of 94% and a precicion of 75%. These results are quite similar to the results in Table 5.1. Similarly, for exciting and montage events, the recall value is quite high (91% and 78% respectively). In the cases of the exciting and montage events, although the precision is quite low, the number of falsely detected events is not excessively high. Take the lowest precision value for example, which was 11% for the exciting events in American Beauty. Overall for this film, nine exciting events were detected by the system, only one of which was manually marked as an exciting event by the user. This gives a very low precision value, but in a browsing scenario nine events would only take a matter of seconds to examine. It is important to take into account different user interpretations of events, so that different users will be able to use the same index to locate events in a movie, even

Film Name	D		E		M	
	Precision	Recall	Precision	Recall	Precision	Recall
American Beauty	66%	94%	11%	100%	33%	100%
Dumb and Dumber	70%	88%	23%	100%	38%	89%
Goodfellas	67%	92%	29%	71%	36%	67%
High Fidelity	68%	98%	29%	100%	43%	80%
Reservoir Dogs	100%	94%	50%	100%	67%	67%
Shrek	75%	100%	32%	100%	83%	73%
Average	**75%**	**94%**	**26%**	**91%**	**50%**	**78%**

Table 5.6: Results of event detection using ground truths created by multiple users. Legend: D - dialogue events, E - exciting events, M - montage events

if they have different interpretations of the events. In other words, the recall rate is far more important, as the presentation of all events is necessary. This will result in lower precision throughout the movies, but should result in consistantly high recall for different annotations.

5.4.2 Comparison of Different User Interpretations

Although the user interpretation differed on many occasions to the author's, the detection rate is still quite high. It is interesting to note the differences in opinion between the author's annotation and that of the other users. In general, the mark up of dialogue events was quite similar, this is largely due to the straightforward definition of a dialogue. Both precision and recall values are therefore quite similar. Many of the users marked up fewer exciting and montage events compared to the author. This decreased the precision for both event classes, as there were more unmarked events. This is due to a lack of a strong definition of exciting/montage events, which resulted in the users only marking up the most prominent exciting/montage events. Note, however the high recall in both cases. Overall, there is a strong overlap between both sets of annotations, although there are still a number of differences, due to different interpretations of the activities on screen.

A comprehensive study of the results was undertaken in order to establish how much of an overlap occurred between the two different users' interpretation of the same film and how this related to the results obtained by the event detection system. The results of this analysis can be found in Tables 5.7, 5.8 and 5.9. These tables represent the difference in opinion between two different users in their definition of each of the three event classes. User one is the author, and user two is the volunteer who annotated that particular movie.

In the three overlap tables; the first column represents the total number of events manually marked up by *either* user. This includes events marked up only by one user and events marked up by both users. The 'User One Total' and 'User Two Total' columns display two numbers each. The first number is the number of events marked up by the

Film Name	Total Dialogues	User One Total	User Two Total	Only One Annotation	No. Detected
American Beauty	45	44(13)	32(1)	14	13
Dumb and Dumber	44	36(3)	41(8)	11	8
Goodfellas	68	57(7)	61(11)	18	15
High Fidelity	50	44(2)	48(4)	6	5
Reservoir Dogs	22	14(0)	22(8)	8	7
Shrek	35	32(2)	33(5)	7	6
Total	**264**	**227(27)**	**237(37)**	**64**	**54 (84%)**

Table 5.7: Results of overlap between different users in manual mark up of dialogue events

Film Name	Total Exciting	User One Total	User Two Total	Only One Annotation	No. Detected
American Beauty	2	2(0)	2(0)	0	-
Dumb and Dumber	14	14(11)	3(0)	11	11
Goodfellas	12	10(5)	7(2)	7	5
High Fidelity	3	3(0)	3(0)	0	-
Reservoir Dogs	5	5(3)	2(0)	3	3
Shrek	14	14(7)	7(0)	7	7
Total	**50**	**48(26)**	**24(2)**	**28**	**26(93%)**

Table 5.8: Results of overlap between different users in manual mark up of exciting events

Film Name	Total Montage	User One Total	User Two Total	Only One Annotation	No. Detected
American Beauty	19	19(4)	5(0)	14	13
Dumb and Dumber	15	15(6)	9(0)	6	5
Goodfellas	10	7(4)	6(3)	7	6
High Fidelity	6	6(2)	4(0)	2	1
Reservoir Dogs	4	3(1)	3(1)	2	1
Shrek	18	12(0)	18(6)	6	4
Total	**72**	**62(27)**	**45(10)**	**37**	**30 (81%)**

Table 5.9: Results of overlap between different users in manual mark up of montage events

user, and the second number (the one in brackets) represents the number of unique events marked up by that particular user (i.e. that were not marked up by the other user). For example, in the entry for American Beauty in Table 5.7, user one marked up 44 dialogues, 31 of which were also marked up by user two. The remaining 13 were not marked up by user 2. Similarly, user two marked up 32 dialogue events, 31 of which were also marked up by user one while 1 dialogue event was only marked up by user two, and not user one. Of the remaining two columns, 'Only One Annotation' represents the total number of events only marked up by one user (in this case, the 13 from user one and the 1 from user two, giving a total of 14). The final column, No. Detected, displays the number of events only marked up by one user, and detected by the system. So in this case, of the 14 events that were only marked up by one of the users, 13 of them were automatically detected. This indicates that the event detection system is capable of facilitating different interpretations of the same movie. This is important, as different people will invariably have differing opinions on what constitutes an event. It is important to have this flexibility inherent in the system, so that many different people can make use of the results.

Throughout Tables 5.7, 5.8 and 5.9, there is significant overlap between the different users. The level of agreement can be seen from the **Total** row. In Table 5.7 for all films and all users, 264 dialogue events were marked up. 64 of these were only marked up by one user, while 200 dialogue events (264 - 64) were marked up by both users. In the mark up for exciting events and montage events, there was less agreement between the two ground truths. This can largely be attributed to the lack of an exact definition of these events. Although it is straightforward to recognise a conversation, as there will be a number of people interacting with each other, it is quite hard to define 'exciting' or 'emotional'. These are abstract concepts, and are open to interpretation from different users. As can be seen from the total value in the **No. Detected** column, a large percentage of the events that the two users disagreed on (i.e events that were marked up by *only* one person) were detected (84% for dialogues, 93% for exciting, and 81% for montage events). This indicates that different user interpretations are accommodated by the detection system.

5.5 Analysis of Overlapping Events and Unused Shots

5.5.1 Overlap Between Detected Events

A level of overlap between the results of the three event detection techniques is present, i.e. some portions of a movie are labelled by the *system* as being part of more than one event. Since there is a certain amount of leeway required in the presentation of events, a level of overlap is desired. This, again, is largely due to the fact that different users will have different interpretations of the same event in a movie.

Although not addressed in this work, it may be possible to extend the event ontol-

ogy to create further sub-classes of events. For example one sub-class could contain all emotional conversation events, another could contain all exciting dialogue (e.g. argument) events etc. This could lead to more efficient retrieval of events. This idea is further discussed in the future work section of Chapter 7.

One common reason for overlapping occurs when one event is finishing and another is starting. For example if there is a dialogue event followed by an action event within the same scene. This usually leads to a small overlap where the distinction between the end point of the first event (dialogue) and the second event (exciting) is blurred, so a number of shots are detected as belonging to both the end of the dialogue and the beginning of the exciting event. These overlaps are typically quite small, and do not effect the indexing process. It is more interesting to look at the more significant parts of the movie that overlap, where large parts of events, or complete events, are detected by more than one event detection method. As there are three events being detected, there are three possible overlap scenarios between two event classes, each of which is analysed below. The overlap between all three events is not extensively discussed. In all cases this three-way overlap occurs when one event finishes, while another that belongs to two event classes begins, or vice versa. So for example, in the film American Beauty, in one scene two people are having a fight (detected as an exciting event), and immediately following this the same characters have an emotional conversation (detected as both a dialogue and a montage event). There are a number of shots that belong to both the fight and the emotional conversation, so were detected as part of all three event classes. This type of overlap is extremely rare, and only occurs in 0.5% of the shots.

Overlap Scenario 1: Dialogue and Exciting

Overall, the most common type of overlap occurs between dialogue and exciting events. Of the 11162 shots, there were 969 shots that were classified as both dialogue and action events. This corresponds to 8.7% of the total shots for all films. The causes of overlap between these events are similar to those of the dialogue and montage overlap. Again, most overlapping events have elements of both types. One such example occurs in *Dumb and Dumber*. In this sequence of shots, one character is talking to another beside a car. A comical situation ensues, whereby one character's foot accidentally gets set on fire. He then tries to continue the conversation, without the other character realising that his foot is on fire. This sequence of shots contains elements of action and dialogue. The increased shot pace and movement are consistent with an exciting event, thus it is detected by the exciting event system, but there is also speech and shot repetition, which is detected by the dialogue system. Similarly, in the film *Chopper* the lead character drags his girlfriend through a crowded nightclub (exciting) whilst arguing with her (dialogue). This is an example of the most common reason for this overlap, where there is an argument between characters that involves them talking to each other.

There are also occasions where one of the classifications is a false positive. For example, a conversation that takes place in a car may also be incorrectly detected as action due to the moving camera (this happens in *American Beauty* and *Dumb and Dumber*). The directorial style of the film *Amores Perros* ensures that many dialogue events are incorrectly detected as action events. In this film, the director chose to film in a hand-held camera style, which means there is constant camera movement throughout the film. This leads to incorrect classifications if combined with a fast shot cut rate.

Overlap Scenario 2: Dialogue and Montage

For the ten films tested, in total there were 11162 shots. 454 shots were detected as both dialogue and montage (an average of 45 per film). This corresponds to 4% of the total number of shots. In general the detection overlap between these two events contains elements of both events. In the film *American Beauty*, there are a number of occasions where an emotional conversation takes place. For example, one particular overlap occurs when two characters kiss for the first time. Both before and after they kiss they converse in an emotional manner. This is an example of an event that can be justifiably labelled as both dialogue and montage (emotional). Other examples of situations where the dual classification of these two events is justifiable occur in *High Fidelity*. Much of the film takes place in a music shop, and on a number of occasions loud music is prominently playing while characters are talking to each other. If the music was quietly playing in the background while people were conversing, then this would be a false detection of a musical event, however, the music is quite prominent and is a fundamental part of the event, as the characters are talking about the music. Therefore it is justifiable to label this event as being a musical event. In the film *Battle Royal* two characters, who are about to commit suicide by jumping off a cliff, bid a tearful goodbye to each other. This has elements of dialogue as well as montage, and is thus detected by both techniques. There is also a dream-like sequence of shots that contains dialogue between two characters, and is detected by both systems.

Another cause of overlap between dialogue and montage events occurs when one of the events is detected incorrectly, for example, a real dialogue event that is incorrectly labelled as a montage. These are obviously undesirable occurrences. Typically, if this occurs, the event coincidentally contains elements similar to both types of events. For example, in a part of *American Beauty*, one of the characters (standing at his bedroom window) is filming his neighbour (also standing in her bedroom window) directly across from him. The non-moving camera changes a number of times between showing the bedroom window of the character doing the filming, and the neighbours bedroom window. Thus, although the event does not contain speech, it does contain a static camera and a high amount of shot repetition, features usually associated with dialogue events. This leads to the sequence of shots erroneously being detected as a dialogue. This is an

example of a non-dialogue interaction between characters.

Overlap Scenario 3: Action and Montage

The final possible overlap between events is the overlap between action events and montage events. This is the least common occurrence of overlap, and indeed in three of the ten films there is no overlap between action and montage events. One example of a justifiable overlap occurs in *Dumb and Dumber*. This particular sequence of shots begins as a montage showing two people enjoying a day out playing in the snow. However, after a number of shots showing them building snowmen etc., they begin to have a fight. Although this fight is part of the overall montage event, it is also correctly detected as an exciting event. Dual detection may also occur in an action event with an accompanying musical score that is incorrectly labelled as a montage. For example a fight with music playing in the background. Overall there are 270 shots detected as both action and montage shots, corresponding to 2.4% of the total shots.

5.5.2 Unused Shots

Although the aim of the event detection system is to classify all relevant parts of a movie into one of the three event classes, there is also a number of shots that are not classified as any type. There are a number of reasons why a shot might not be detected as being part of any event. One common example is a short, insignificant, occurrence such as a passing 'hello' between two characters. Although there are elements of dialogue involved, it is not significant enough to be labelled as a dialogue *event*. Similarly, there may be a short sequence of action lasting one or two shots. Again, given the small amount of shots present, the exclusion of this type of occurrence is desirable in order to avoid over segmentation.

There are also occasions where there are a number of shots at the start or end of an event that are missed. This may occur in events in which an establishing shot is used. For example, before an event, the director may *establish* the scene by showing a shot of the outside location. This effectively lets the viewer know where the event is taking place. This shot occurs before the beginning of the event, so, although its detection is useful for setting context, it is not a fundamental part of the event. Possible future work could specifically detect these establishing shots and incorporate them into the movie index. Also, as mentioned in Section 2.4.1, the end credits are unclassified as the rolling list of credits does not belong to any event class. Although the end credits are part of a movie, no attempt is made to include them in the indexing process as their location is always known (i.e. at the end of the movie). As they do not advance the story in any way, they are not meaningful parts of the movie. Also, much of the information present in the credits is available from other sources, such as Internet movie databases. Although they are not desired, the addition of the ending credits to an index would be straightforward as they

Film Name	Total Shots	Shots Not Classified	Percentage Not Classified
American Beauty	1001	98	9.8%
Amores Perros	1779	217	12.2%
Battle Royal	1292	78	6%
Chopper	813	74	9.1%
Dumb and Dumber	932	153	16.4%
Goodfellas	1134	91	8%
High Fidelity	1048	74	7%
Reservoir Dogs	626	70	11.2%
Shrek	1221	52	4.64%
Snatch	1316	75	5.7%
Total	**11162**	**982**	**8.8%**

Table 5.10: The unused shots in the test corpus

are usually contained in a single long shot at the very end of a movie

There are, of course, missed events that should be classified into one of the three event classes. These are summarised at the start of the results section so are not scrutinised further here. A common cause of missed shots occurs when the event detection system misses part of an event. For example, an action event may last 2 minutes, but only 1 minute 45 seconds is detected. This usually occurs either due to the state machine prematurely detecting the end of an event, or missing part of the beginning. For example, there could be an action event where the action slows down toward the end of the event, resulting in the state machine perceiving this as an end to the action. The total amount of unclassified shots can be seen in Table 5.10. As can be seen, over 91% of all shots are detected as belonging to one of the three event classes.

5.6 Non-Movie Data

The finite state machine approach was also subjected to a number of experiments on non-movie data. Although the main theme of this thesis is concerned with indexing movies, the directing, editing, and sound creation principles the FSM system is based upon are common to many other types of video. Therefore, the FSM based event detection system can also be utilised to summarise and index this content.

The widespread use of filmmaking conventions, which is guided by directing, editing, and sound engineering principles, has been reported on numerous occasions throughout this thesis. Due to a more constrained budget and timescale, many programs made for television are more reliant on convention than the relatively creative film world. Shooting a one and a half hour movie may take a number of months to complete, so there is ample time to be creative when planning and shooting. Conversely, many television shows may

Program Name	Dialogue		Exciting		Montage	
	Precision	Recall	Precision	Recall	Precision	Recall
Sopranos 1	97%	100%	67%	100%	25%	33%
Sopranos 2	100%	96%	60%	75%	100%	100%
Sopranos 3	77%	100%	38%	75%	75%	100%
Simpsons 1	96%	100%	-	-	100%	100%
Simpsons 2	89%	100%	100%	100%	-	-
Simpsons 3	97%	100%	67%	100%	50%	100%
Lost 1	78%	81%	79%	100%	80%	100%
Lost 2	77%	94%	69%	100%	67%	100%
Lost 3	84%	78%	54%	100%	83%	100%
Average	**88%**	**93%**	**64%**	**96%**	**72%**	**92%**

Table 5.11: Results of event detection system on a set of television programs

have the same amount of time to complete a whole series. Many shows are produced at a rate of one hour of content per week. Often in programs, such as soap operas, a complete half hour program may be shot in a single day. Thus, the makers of the program do not have as much time to plan original shots and edits, and rely substantially on convention.

In order to test this assertion, a number of television shows were selected and analysed by the event detection system. Three different television series were selected and three programs from each were analysed. The three series selected were: *The Sopranos*, *The Simpsons*, and *Lost*. They were selected due to their diverse nature, as they all have contrasting settings and themes. The Sopranos is a crime-based drama, based on a mafia boss and his family in New Jersey. The Simpsons is a long running animated television series set in an American suburb, while Lost is set on a deserted island, and follows the lives of a group of survivors of a plane crash.

The nine programs, in total just over five hours of content, were subjected to the same analysis techniques as the movies in Section 5.2. Firstly, a manual annotation of each program was created by marking each of the exciting, dialogue and montage events. Secondly, the videos underwent the same automatic event detection process as the movies. Finally, the automatically generated results were compared to the manual annotation. The results are presented in Table 5.11. As can be seen, there is consistently high precision and recall across all events for each of the programs. The results are comparable, slightly higher even, than the results for the movies for the reasons outlined above.

Generally, the reasons for missed events and misclassification are similar to those for the movies. In the program Lost, many conversations are shot with a moving camera which results in a small number of missed dialogue events. These conversations take place when the characters are walking through a jungle, and thus there is a moving camera with very little shot repetition. Apart from Lost, only one other dialogue event is not detected for all of the videos. This takes place in the second Sopranos episode and is missed due

Film Name	D		E		M	
	Precision	Recall	Precision	Recall	Precision	Recall
American Beauty	88%	84%	25%	100%	57%	80%
Amores Perros	86%	16%	41%	84%	64%	88%
Battle Royal	86%	54%	87%	70%	21%	70%
Chopper	89%	89%	50%	50%	43%	100%
Dumb and Dumber	82%	72%	40%	43%	56%	67%
Goodfellas	70%	84%	12%	20%	29%	29%
High Fidelity	79%	96%	100%	50%	50%	83%
Reservoir Dogs	78%	94%	100%	71%	100%	67%
Shrek	82%	92%	80%	79%	56%	24%
Snatch	93%	86%	95%	71%	30%	50%
Average	**83%**	**76%**	**57%**	**65%**	**49%**	**68%**

Table 5.12: Results of event detection using a hidden Markov model for the authors' ground truth. Legend: D - dialogue events, E - exciting events, M - montage events.

to a lack of speech (as there is much silence present) or repetitive camera shots (as the characters are moving). In total, only two exciting events, and two montage events are not detected. The two missed exciting events are filtered by the system, as they are deemed to be too short, while the two montages are both missed due to errors in the audio, where the audio is misclassified as speech instead of music.

5.7 Hidden Markov Model System

5.7.1 HMM Results

The hidden Markov model based event detection method is used as a benchmark to assess the finite state machine based approach. This approach is tested on the same set of films as the finite state machine approach (although four of the films are also used in order to train the HMM), and the results are presented in Table 5.12. The movies used as training data are in bold. The ground truth used is the same as for the finite state machine approach, so the results are directly comparable.

Note firstly, that overall, the precision and recall values are lower than those for the FSM based approach. The average precision value for the dialogue events is 83%, while the average recall value is 76%. These values are quite high and indicate good performance of the system. The precision value is 7% higher than the FSM approach, while the recall value is 19% lower. In the film American Beauty, two conversations that take place in a moving car are not detected by the system. In both cases, they are labeled as exciting events due to the large amount of camera motion, despite the fact that there is a high amount of speech present. In another example, an emotional conversation is labeled as a montage event due to the relatively high amount of music in each shot. The film Amores

Perros resulted in the lowest overall recall value. A large amount of the conversations are labeled as exciting events. This is due to the free-hand moving camera style employed by the director, which essentially ensures that there is a moving camera in most shots. This camera style is quite unconventional, and is quite dissimilar to the way most films are created, resulting in low recall. Note how this low recall for the dialogue events in Amores Perros impacts on the precision value for the exciting events, as most of the missed conversations are labeled as exciting events. This illustrates one of the key advantages of the FSM approach since it targets the individial components of the dialogues (speech, static camera and shot repetition), rather than the event as a whole. Although one of the components is not present (static camera) the other components are, and therefore more of the dialogue events are detected by the FSM approach. A number of dialogues in Shrek are incorrectly labelled as exciting events. Although there are some exciting elements to the conversations, there is insufficient evidence to conclude that it is an exciting event. Also, in the same film, at one point a conversation occurs just after a fight. The HMM labels both events as being part of the same exciting event, and therefore ignores the dialogue.

The precision rate for the exciting events is comparable with that of the FSM approach at 57%. However, the recall rate is 30% lower. In the film Battle Royal, for example, a number of exciting events have an untypically long shot length and music in the background. Thus, they are erroneously labeled as montage events. Also, some exciting events contain people shouting at each other and a still camera, so are classified as dialogue events. Similarly, for Dumb and Dumber, a number of arguments and fights are classed as dialogue events due to the presence of speech. Also, on one occasion in Dumb and Dumber, a long montage occurs which contains a fight in the middle. The system labels the whole event as a montage, including the fight.

Finally, the montage event detection method produces a 24% decrease in precision compared to the FSM approach, with a 22% drop in recall. The lower precision can be attributed to the incorrect labeling of many other events as montages. The films Battle Royal and Amores Perros contain a number of montage events that are incorrectly labeled as exciting events due to the presence of small amounts of camera movement. Dumb and Dumber contains one montage event that contains portions of speech. This results in the montage event being classed as a dialogue. Also in American Beauty, one montage occurs in between two conversations, and all three events are labeled as dialogue events. Finally in Shrek, one montage event is immediately followed by an exciting event, however both are classified as exciting events.

5.7.2 Comparison of HMM and FSM Approaches

The choice of the features used as an input to the HMM system can be a limiting factor in its accuracy. However due to the high performance of the FSM based approach, it is

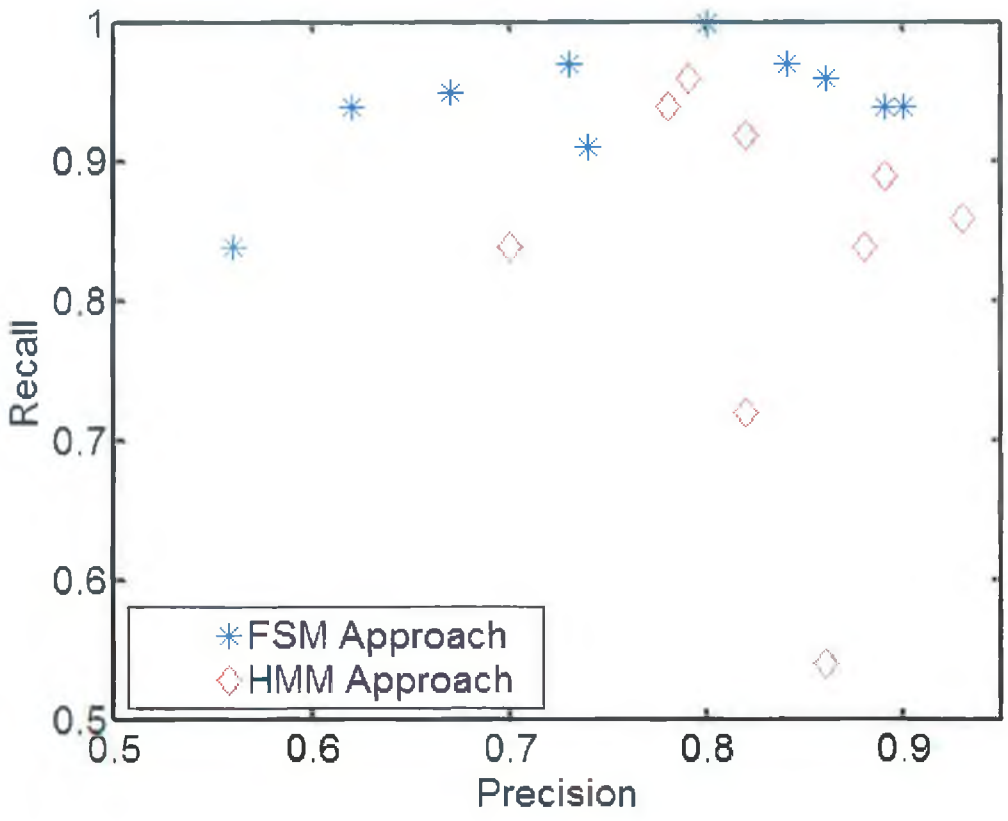

Figure 5.3: FSM and HMM dialogue event detection results

not thought that the quality of the shot-level feature vector is a large performance restriction. Of course, as with the FSM approach, occasionally the shot features are incorrect (labelling speech as music for example), but overall they seem appropriate. There are occasions where a system may produce an incorrect classification as a direct result of noise in the extracted features, however these occasions are similar for both approaches, and will not hinder one approach over another. The FSM approach is more suited to cope with noise in the generated data of the shot-level feature vector, and it produces better results using the same data. Figures 5.3, 5.4 and 5.5 present graphs of precision and recall values for both the FSM and HMM systems for all three event classes. These are the same results presented in the various tables in Sections 5.2 and 5.7. As noted previously, 100% recall is desired, however a slightly lower precision value (around 80%) is preferred in order to allow for different user interpretations. As can be seen from these graphs, the FSM values are consistantly closer to these ideal values.

Although the HMM approach uses the same features as the FSM approach, it is not directly based on film making techniques. By detecting the individual 'ingredients' of each event, and using these as a basis for classification, the FSM approach is more suitable to make a judgement as to the film-makers intentions. By grouping features together and looking for the overall event, the HMM approach assumes that the features present in one

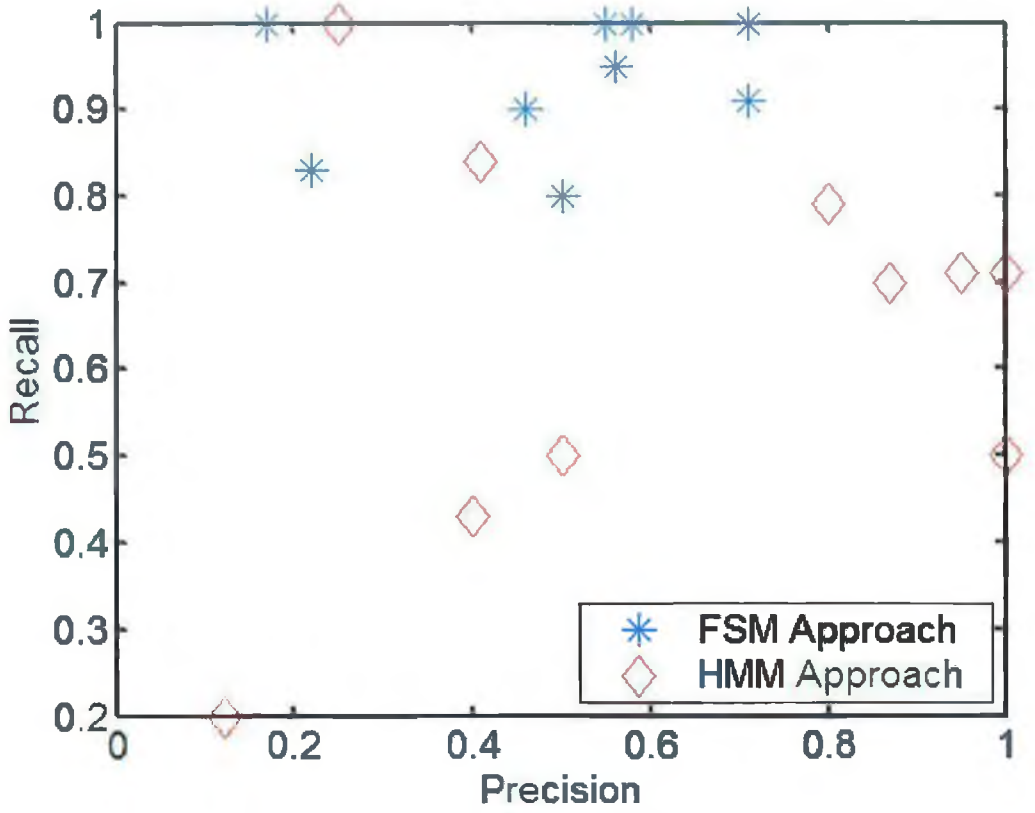

Figure 5.4: FSM and HMM exciting event detection results

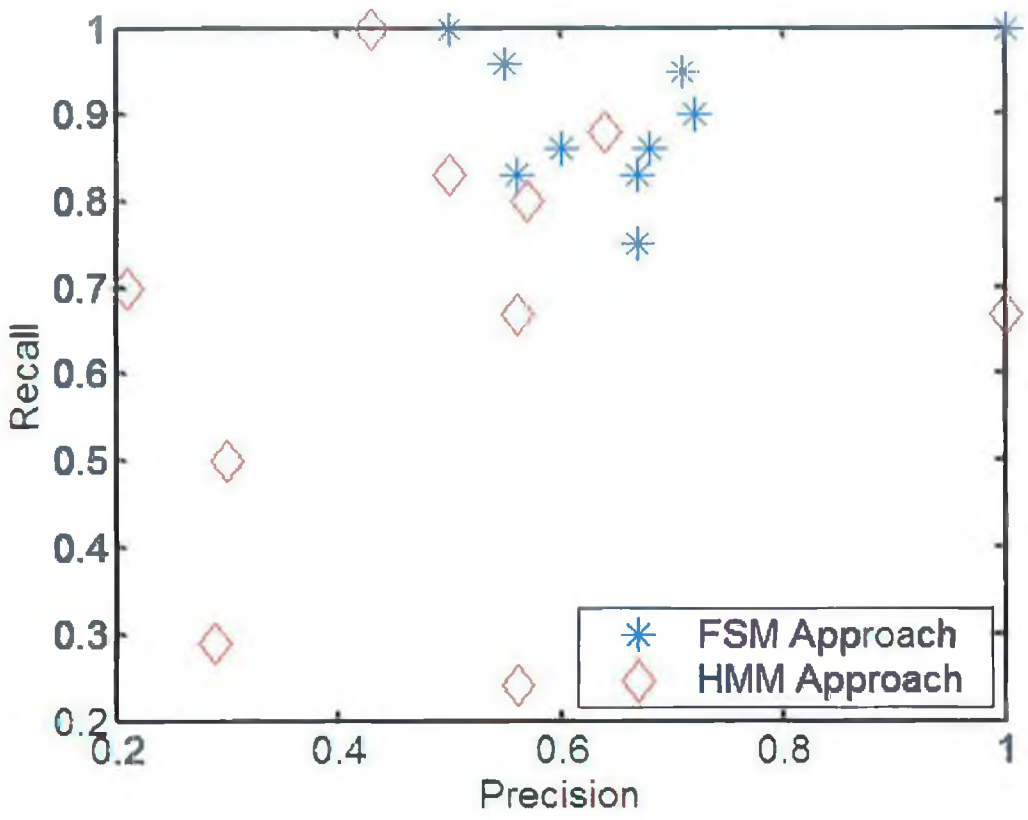

Figure 5.5: FSM and HMM montage event detection results

event will also be present in another, albeit with different weights. In Amores Perros, for example, the original 'hand-held' camera style adopted was previously mentioned. Very few other films use this camera style. While the dialogue events in most movies contain a static camera, the hand held style ensured the presence of camera movement throughout the dialogue events in Amores Peros. As the HMM was not trained on this particular film, when it encountered camera movement it was strongly associated with exciting events. The dialogue events in Amores Perros did, however, have other elements necessary for a director to present a conversation to a viewer, such as speech and shot repetition. The FSM approach searches for the individual components of the events, so even if one component is missing, or has changed, the remaining components should still be in place. Ultimately, filmmaking is a creative medium, however there are a number of essential components that, if detected, can be exploited to infer some meaning from the content. The FSM approach is far superior at exploiting this. Despite the atypical method of shooting dialogue events in Amores Perros, the FSM based approach managed to find 84% of all dialogues, while the HMM approach could only manage 16%. This indicates that the HMM approach is not as generally applicable as the FSM approach.

Another drawback of the HMM system is that shots can only be classified as part of one event, as the HMM can only be in one hidden state at a time. Although, for any one interpretation of a movie, this may be desirable, ultimately, each user will have a distinct interpretation. Therefore, as discussed in Section 2.4.1, a successful system needs to accommodate as many user interpretations as possible, while still maintaining a reasonably high precision value. As detailed in Section 5.5, there are many occasions when a classification overlap of events occurs, and is justified. Also, due to the high probabilities of staying in the same hidden state, if there is a short event in between two longer events, it may be overlooked as the HMM only detects the larger events. Many montage events contain exciting portions, for example. It is desirable that both events can be detected, so independent event detection systems are preferred.

5.8 Discussion

This chapter investigated the performance of the film grammar based event detection system. Extensive experiments were carried out, each of which explored a different aspect of the system. The first set of experiments proved the reliability of the system in detecting dialogue, exciting and montage events. An explanation of the FSM design process was presented. Secondly, using a set of impartial users, the system's efficiency in accommodating differing interpretations of events was demonstrated. This also proved that a slight drop in the precision value was actually preferred, as it allowed for flexibility in user interpretations. The desired presence of this flexibility of interpretation was re-enforced

by examining the occasions where the automatically detected events overlap, indicating many events that could be classified into multiple classes. The FSM approach was also tested on a test set of non-movie data to investigate how adaptable it is to other content. The results of this test are extremely encouraging, and enforce the view that many forms of non-movie video are more constrained by convention than movies themselves. This also demonstrates the versatility of the approach, as, although it was built for movies, the system performs very well on previously unseen sets of video data. Following this, the FSM-based system was evaluated against a different approach to event detection using a different machine learning method, a hidden Markov model. As the HMM system was not as heavily based on film grammar, or the inherent structure of movies, its performance is significantly lower than the FSM approach.

The various results in this chapter indicate that there is significant correlation between the film grammar principles presented in Chapter 2 and the extracted audiovisual features. The FSM approach is largely based on the premise that this correlation can be exploited in order to extract knowledge about the content. It was previously noted that dialogue events typically contain speech, low amounts of camera movement, and high amounts of shot repetition. It was also noted that exciting events typically contain low amounts of shot repetition, and high amounts of both motion intensity and short shots. Finally, it was noted that the montage events typically contain high amounts of music and/or silence, and low amounts of shot repetition and motion. The high results of the event detection process indicates that these notations are valid. A more precise measure of this correlation can be viewed in the design of the B matrix of the hidden Markov model (Section 4.3). The high amount of overlap between the observable states based on each event class, and the respective hidden states further indicates the reliance of film makers on the features mentioned above.

CHAPTER 6

User Evaluation

6.1 Introduction

Having developed a system for detecting all of the dialogue, exciting and montage events in a movie and selecting a set of key-frames, a presentation mechanism to demonstrate the operation of the system is required. To this end, a film event summarisation interface was created that allows users to browse and play all of the detected events in a film. The search based method of locating events described in Section 4.2.5 is also incorporated into the system. The purpose of this system is to demonstrate one potential use of the event detection and movie summarisation research, as well as providing a testbed to assess the validity of event detection in a user experiment.

This chapter firstly explains the movie browser interface, and then describes a set of experiments carried out using the movie browser. These experiments aim to assess how useful event-based browsing and retrieval is in a real application. To this end, comparisons between locating clips using an event based approach, a keyframe based approach, and a search based approach are drawn.

6.2 The Movie Browser

6.2.1 Interface

In the movie browsing interface there are three ways to browse a movie. The first is to browse by keyframe, the second to browse by event, and the third is to search for events. There are a number of reasons why users may prefer to use one system over

another. The keyframe based method for example, may be useful for somebody who knows exactly when a particular event in a movie takes place. Thus, by immediately navigating to a particular temporal location, he/she can locate the clip efficiently. Also, as all keyframes are presented, a user can be sure that he/she is viewing everything that happens in a movie. In the other two methods, as is described later, keyframes are not shown for portions of a movie that do not belong to an event class. Also, each event is only allocated five keyframes, so all information about the event must be garnered from this reduced set of images. The advantages of these approaches have previously been described (smaller search space, higher semantics etc.) and are not discussed further here, but the experiments in Section 6.3 indicate cases where one retrieval method is preferred over another. As both the event and search based methods have similar structure, a direct comparison of these methods of retrieval can be performed.

The opening screen of the interface developed is shown in Figure 6.1. The list on the left shows each film in the database. In this case, the database consists of ten films. The film Shrek is selected as the current film, and its name is highlighted. A filter box is placed above the list of films, which allows users to constrain the list of films by specifying an actor or a character. For example, by selecting the name 'Brad Pitt' from the actors filter, only films in which he appears will be presented. Although for a database of ten films this has limited use, this becomes a useful feature as the amount of films in the database increases.

As can be seen in Figure 6.1, to the right of the film list is the main display pane. This displays a set of images from the currently selected movie. In total there are 36 images, chosen at $\frac{1}{36}$ temporal increments throughout the film. To the right of the display images, is a small button that says 'See All'. When this button is pressed, all keyframes from the film are displayed in the display pane.

Metadata for the selected film is presented above the display pane. This information states the film name, the director, the writer, the year it was released, the films genre, as well as a list of the main cast and characters. This information is freely available from many sources on the web for a given movie. Also displayed is an event list which states the amount of events detected for each event class. In this film, 15 exciting events, 64 dialogue events and 17 montage events were detected by the FSM-based system proposed in Chapters 3 and 4. Just to the left of this, the three event classes are listed in large letters, each with its own button. By clicking one of these buttons, each detected event in the chosen event class is presented. Finally, just below and to the right of where 'Montage' is written is a small arrow. By selecting this arrow, a search pane is opened which allows users to search for particular events. The following sections discuss the three browsing methods.

Figure 6.1: Opening screen of the movie browsing system. The film *Shrek* is selected

6.2.2 Keyframe Browsing

Keyframe browsing is a rather simplistic way of browsing a movie. If a user is searching for a particular clip, browsing by keyframe is certainly more efficient than actually watching the movie, as it is possible to see a larger portion of the movie at any given time. However, there are a number of drawbacks to using keyframes to browse movies. Firstly, the volume of keyframes makes it difficult to navigate through an entire movie. For the ten movies in the database used in the MovieBrowser system, the average amount of shots per movie was 1116. Thus, if the portion of the movie sought is at the end of the film, a user would have to examine over a thousand images before he/she finds the correct ones.

Figure 6.2 shows the interface when the 'See All' button is pressed. This displays all keyframes in the movie for a user to browse. The keyframes are displayed five per row in chronological order. Users can navigate forward or backward through the keyframes. On the left of each row of keyframes, a start time is displayed. This indicates the start time of the first keyframe in that row. Just under the start time is a 'Play' button which opens an external video player and plays the movie from the start time. This is illustrated in Figure 6.3.

Figure 6.2: Browsing by keyframe using the movie browsing system

6.2.3 Browsing By Event

The second way of using the movie browser is to browse a movie by event class. A user selects a particular event class and a pre-detected list of events as well as a number of representative keyframes are presented. In order to browse a particular event class, users click on the relevant button, and the selected events are presented. In Figure 6.4, the 'Exciting' button was pressed, and all (15) exciting events from the movie Shrek are displayed.

The coloured bar at the top of the display panel shows the areas in the film where exciting events are detected. For each event, five keyframes are selected. The first and last keyframes are the first and last keyframes in the event, while the three keyframes in the middle are the selected ones as specified in Section 4.2.4. To the left of the five keyframes, some information about the event is displayed. This includes the event number, the start and end times of the event, as well as the amount of shots in the event. There is also a 'See All' button which allows users to display all of the keyframes from all of the shots in that particular event. Notice that in Figure 6.5 every keyframe for exciting event 5 is displayed.

It is also possible to play the event in an external video player by clicking on the 'Play' button. All of this functionality is available for each event class. Figure 6.6 shows

Figure 6.3: Playing a movie clip selected based on keyframe browsing

Figure 6.4: Displaying the exciting events in the movie Shrek

a conversation being played in the external player after being selected from the dialogue event list. Similarly, Figure 6.7 shows a montage event playing after being selected from the list of detected montages.

Displaying events has a number of advantages. Firstly, browsers know the keyframes they are looking at are part of a particular event. By looking at the keyframes from the dialogue events, for example, a user can quickly establish who is involved in the conversation. Also, there is a hugely reduced set of keyframes displayed. If a browser is looking for an exciting event in the film Shrek, only 75 keyframes are presented to the user (as there are 15 detected exciting events, and there are 5 keyframes presented per event). This is a significant reduction from the total number of 1221 keyframes that occur in the film.

6.2.4 Searching for Events

The final way of locating events in movies is to search for them using a set of query criteria. As mentioned in Section 4.2.5, there are two steps involved in searching for events. The first step involves selecting a FSM that finds a set of potential sequences. The second step involves taking this set of potential sequences, and filtering unwanted sequences (i.e. potential sequences that don't have the desired features).

Figure 6.5: Displaying the exciting events in the movie Shrek. Note in event 'Exciting 5', all keyframes are displayed.

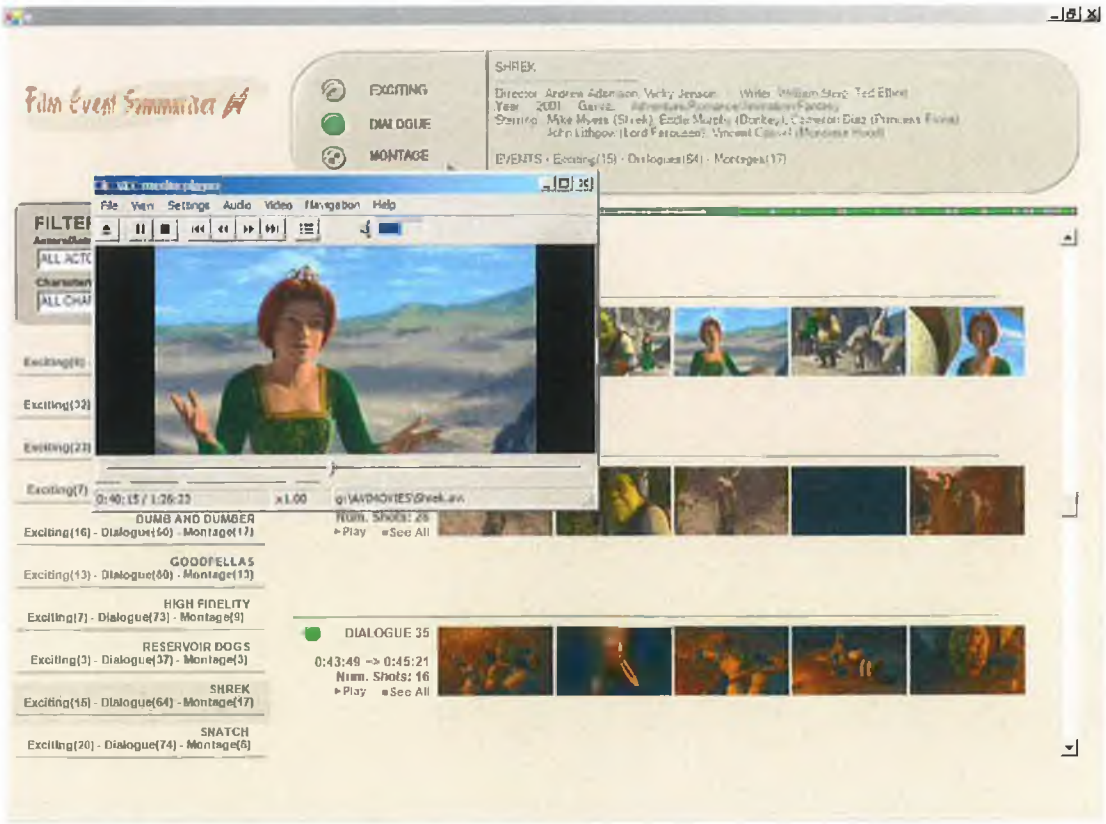

Figure 6.6: Displaying the dialogue events in the movie Shrek, with an external player showing one of the dialogue events.

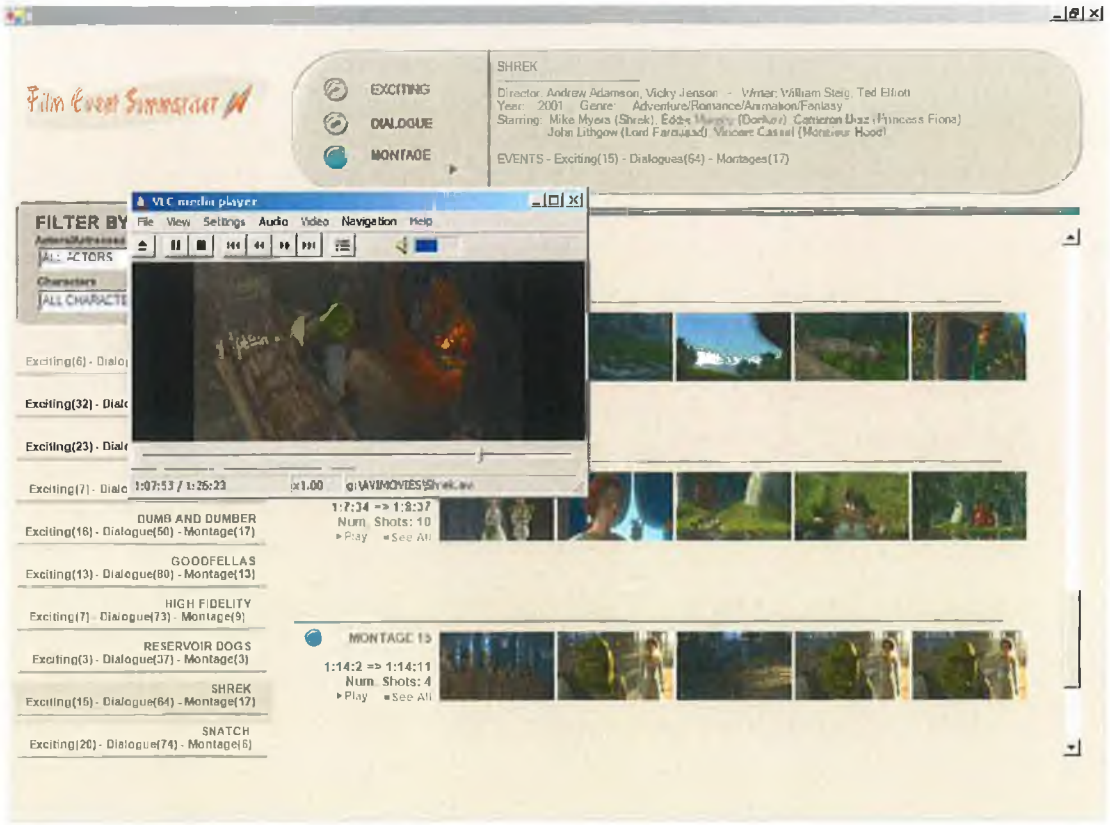

Figure 6.7: Displaying the montage events in the movie Shrek

Figure 6.8: The search panel used to find events with a set of specified features

By clicking on the small arrow, a search panel is revealed at the top of the movie browser, as can be seen in Figure 6.8. The search panel has two sections, one for selecting the relevant FSM (on the left), and the other for filtering the potential sequences generated by the FSM (on the right). There is also a button on the right to initiate the search once the FSM and filtering options have been set. Selecting the FSM involves selecting one of the six displayed options, however the filtering step requires more explanation. There are eight slider bars in total in the filtering section, each of which has an assigned feature. The value of the feature for each event is calculated, and this is compared to the user specified criterion. If it meets the criterion it is accepted, otherwise it is discarded. To the right and left of each slider bar is a check box. Only one of these check boxes can be selected at a time. The check box to the right can be interpreted as an *at least* check box, while the one on the left is an *at most* check box. So, if the slider bar is set to, say, 30%, and the right checkbox is selected this can be interpreted as 'If at least 30% of the shots in this potential sequence contain feature X, then accept it, otherwise discard it'. For example, if the slider bar for the 'Static Camera' is set as shown in Figure 6.9 - (A), where the bar is at 50%, then that means only potential sequences in which at least 50% of the shots have a static camera are accepted. While, if the slider bar is set as in Figure 6.9 - (B), then only potential sequences in which less than 50% of the shots contain a static camera

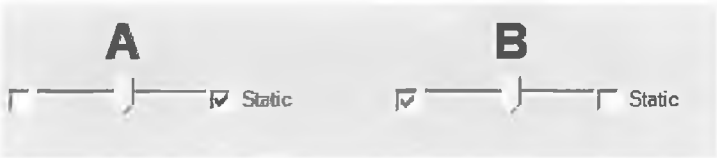

Figure 6.9: Two examples of slider bars for filtering

are accepted. Users can also specify the minimum number of shots in any retrieved event. So, for example, if a user knows that a certain event lasts quite a long time, he/she may not want to display the events that only last 2/3 shots. Therefore he/she can specify the minimum number of shots in order to remove the short events.

To illustrate this consider the following example. A user that is interested in finding events which contain music and high amounts of camera motion (for example a dance, where the camera follows the dancers around a room) can use the music FSM, set the high camera motion slider bar to only retain potential sequences with high amounts of camera motion, and click on the search button. The results of a search using the music FSM, with at least 60% of the shots in each potential sequence containing camera motion are shown in Figure 6.10. In total there are eight events returned. Note that all of the features in the event browsing system are present; the colour bar indicating the location of the events, the play button to open the external player at the start of the event, the start/end times, the number of shots in the event, the ability to display all keyframes from the event etc. As with the predetermined events, there are five keyframes presented. In this case, they are chosen at equal time increments throughout the event.

If more than one filtering slider bar is checked, then the results are OR'ed. Thus, if two filters are activated, then this is equivalent to the statement 'If the potential sequence satisfies the criterion from slider bar 1 OR the criterion from slider bar 2, then accept the potential sequence, otherwise discard it'. So, by setting, say, the 'short shots' slider bar to be at least 60%, then the search in Figure 6.10 would become - 'Find the music potential sequences, for any single potential sequence, if at least 60% of the shots contain high amounts of camera movement OR if 60% of the shots are short shots, then accept the potential sequence, otherwise discard it'. This search would return an increased amount of events as the criterion for accepting potential sequences as events has been relaxed (it would actually return 21 events for the film Shrek). Although not implemented in this interface, the 'AND' operation could be incorporated into a future filtering system. In general, ANDing a set of potential sequences considerably reduces the set of events after filtering.

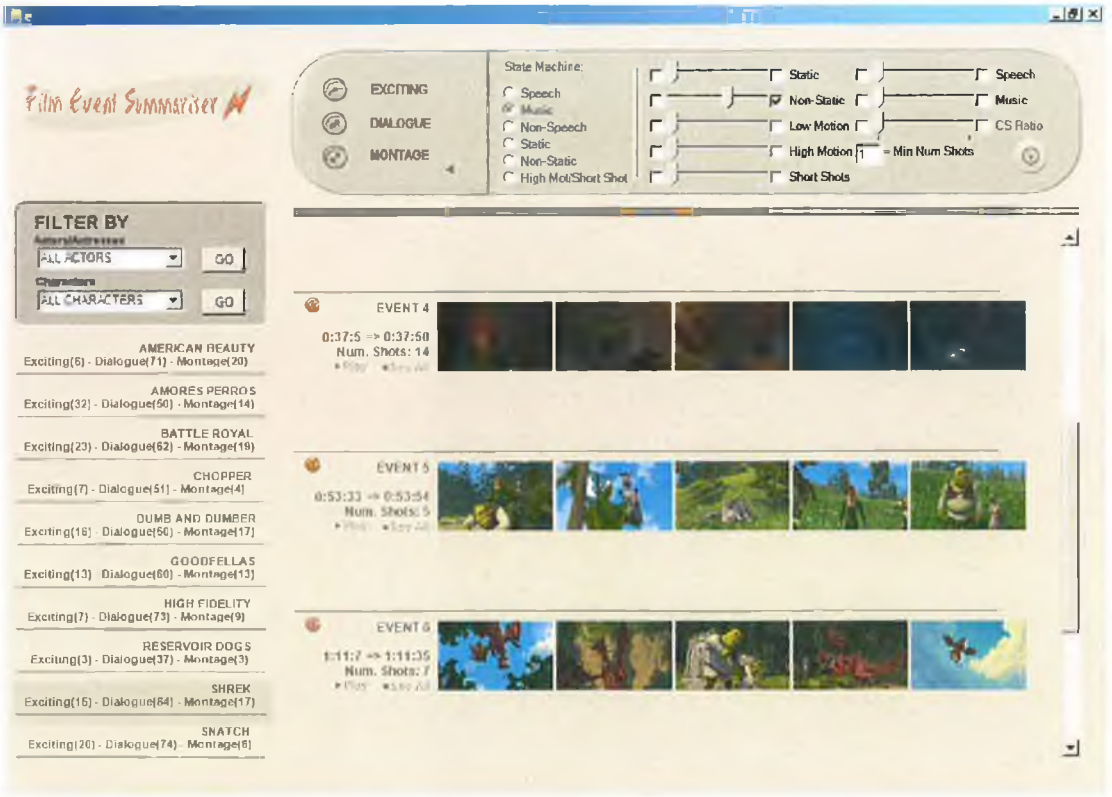

Figure 6.10: Retrieved events after searching for events that contain high amounts of music and moving camera

6.3 User Tests

6.3.1 Motivation

In order to assess the effectiveness of detecting events in a movie and presenting them to a user as an indexing solution, a set of browsing experiments using the movie browser were devised. The purpose of the experiments is to investigate which method of browsing users find most useful. The process involves a number of users completing a set of tasks, which involve retrieving particular clips using the three different browsing methods. The aim of the event based method is to detect and class all meaningful events in a movie, thus, if it is successful, users should be able to complete the experiment without encountering a task that cannot be completed due to shortcomings in the system. Similarly, as the search based system allows users to directly select features that they deem suitable for a task, it should return events that will assist the user in completing the task.

6.3.2 Experimental Set Up

A set of thirty tasks were created, where each task involves a user using one of the systems to locate a clip from a movie. An example is the task: *In the film High Fidelity, find the part where Barry sings 'Lets get it on' with his band.* The complete list of tasks is presented in Appendix E. The tasks were chosen in order to assess how well the respective browsing and retrieval methods can be used in a movie database management scenario. In this scenario, retrieval of specific portions of a movie is essential, and thus the tasks were chosen based on this requirement. The complete task list is quite diverse as it incorporates many different occurrences in a wide range of movies. The tasks were created in order to challenge each of the three retrieval methods. They also aim to simulate real use cases in a video retrieval environment. Each task corresponds to a specific event in a movie. Although this may seem unduly biased toward the event based system, any meaningful part of the movie can be selected for retrieval as part of a task.

There are some possible cases where one system will clearly outperform another. For example, finding shot number 25 is easier to achieve using the keyframe based system than the other two systems, as the keyframes from each shot are presented to the browser, who simply has to count 25 keyframes to locate the desired shot. Similarly, if a task involves finding as many areas of music within a movie as possible, the search based system (and to a lesser extent the event based system) will perform significantly better than the keyframe based system as the audio information can be utilised. Thus, when selecting tasks, it was important to ensure that each task could potentially be completed by each system. This ensures that bias was not placed on one system over another. It is important to note that each system only has the *potential* to complete each task, failings of

each system may result in a task that is difficult to complete. For example, if a particular clip is sought, say an emotional part of the movie with music in the background, then if music is not detected by the audio classifier, this will clearly hinder both the event based system and the search based system in locating the event, as firstly, the event may not be detected in the montage event class, and secondly, if the music feature is used as part of a search, the relevant portion of the movie will not be retrieved.

The start and end times in the movie of each of the events used as tasks were noted. In total there were twelve volunteers, each one completing fifteen tasks, five for each method of browsing. Each task was completed by six volunteers, twice for each system. Each volunteer was given a brief introduction to the interface, and a sample task was completed under guidance for each of the three retrieval methods. Although a brief definition of the three event classes was given to each user, no insight into the event detection methods employed in this system was provided. For example, when describing the exciting events, a number of examples were provided to each volunteer, but no insight into the features used in order to detect exciting events were given. Two users used the keyframe based method to complete each task, two users used the event based method, and two used the search method. For a complete task assignment list see Table 6.1. As can be seen, the first five tasks were completed by user 1 and 2 using the keyframe system, by user 3 and 4 using the event system, and user 5 and 6 using the searching system. This structure was repeated for the rest of the tasks. Three separate versions of the movie browsing system explained in Section 6.2 were created. Each version only facilitated one method of browsing, and all other functionality was removed. For example, in the event based version, the 'see all keyframes' button and the search panel are removed so that users could only complete tasks by browsing the exciting, dialogue and montage events.

An automatic timing program was implemented that recorded how long it takes a user to complete each task, and also to check whether users have located the correct event. Once a user has located a clip in the movie that he/she considers to be correct, they enter the time of the event into the system (which compares this time with the correct start and end times of the tasks). If the supplied time is correct (i.e. between the start and end time of the task) the time taken to complete the task is automatically recorded. If a user supplies an incorrect time, he/she is instructed to continue browsing in order to find the correct time of the event. If a user cannot complete a task, there is an option to give up browsing. If this happens, a completion time of ten minutes is assigned for the task. This heavily penalises non-completion of tasks. In order to compare results for different users, a pre-test questionnaire was created in which the volunteers were required to state which films they had seen before, and how long ago they had seen them. Also, for equality, each user was supplied with thumbnail pictures of the main characters in each movie. Each user was briefly trained on the three retrieval methods, and then asked to begin their

User Name	Total Tasks	Tasks Using Keyframe	Tasks Using Event	Tasks Using Search
User 1	0 → 14	0 → 4	5 → 9	10 → 14
User 2	0 → 14	0 → 4	5 → 9	10 → 14
User 3	0 → 14	10 → 14	0 → 4	5 → 9
User 4	0 → 14	10 → 14	0 → 4	5 → 9
User 5	0 → 14	5 → 9	10 → 14	0 → 4
User 6	0 → 14	5 → 9	10 → 14	0 → 4
User 7	15 → 29	15 → 19	20 → 24	24 → 29
User 8	15 → 29	15 → 19	20 → 24	24 → 29
User 9	15 → 29	25 → 29	15 → 19	20 → 24
User 10	15 → 29	25 → 29	15 → 19	20 → 24
User 11	15 → 29	20 → 24	25 → 29	15 → 19
User 12	15 → 29	20 → 24	25 → 29	15 → 19

Table 6.1: User assignment for tasks

fifteen tasks. Including training, the average time taken to complete the experiment was under an hour.

Upon completion of the experiment, each volunteer was asked to complete a post-test questionnaire in which a number of questions were asked about the movie browser system. Volunteers were asked to specify which method of browsing they preferred, and also to provide any positive or negative comments regarding each method of browsing, and the system in general.

6.3.3 Results

As can be seen from Table 6.1, the first six users were assigned to complete the first fifteen tasks (although using different systems). The results for these users is presented in Figure 6.11. The vertical axis is the time in seconds taken to complete the task, and the horizontal axis is the task index. There are two results for each method for each task (i.e. six results for each task). So, taking task 0 as an example, the two users took 127 seconds and 123 seconds respectively to find the clip using the keyframe based method. Using the event based method, the two users took 33 seconds and 102 seconds respectively. The times for the two users of the search based method were 65 seconds and 25 seconds. As can be seen from the graph, the keyframe based system has many large spikes corresponding to longer search times. For these fifteen tasks, the longest task completion time was for task 10 using the keyframe method, which took 579 seconds. The shortest task time was 22 seconds, for task 5 using the event based system.

The second set of results (for the second set of six users on tasks 15 to 29) are presented in Figure 6.12. On two occasions users gave up whilst using the search based method, thus the maximum time taken was 600 seconds. These were the only two tasks that were not completed. In both cases, the users were not familiar with the movie, and

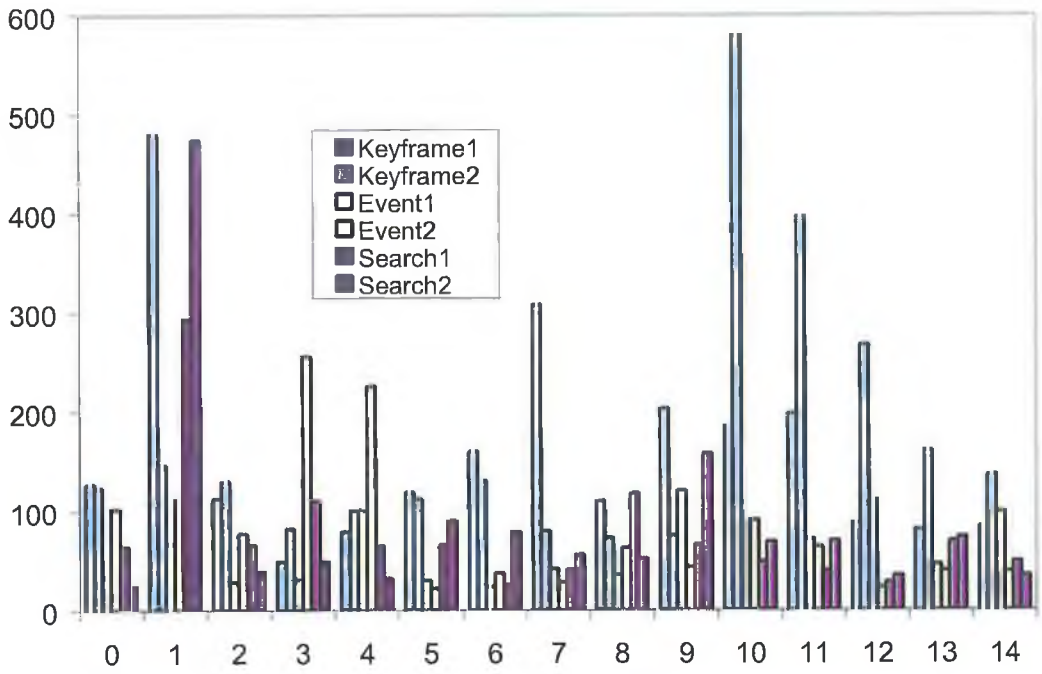

Figure 6.11: Plot of time taken to locate clip for the first fifteen tasks using three browsing methods

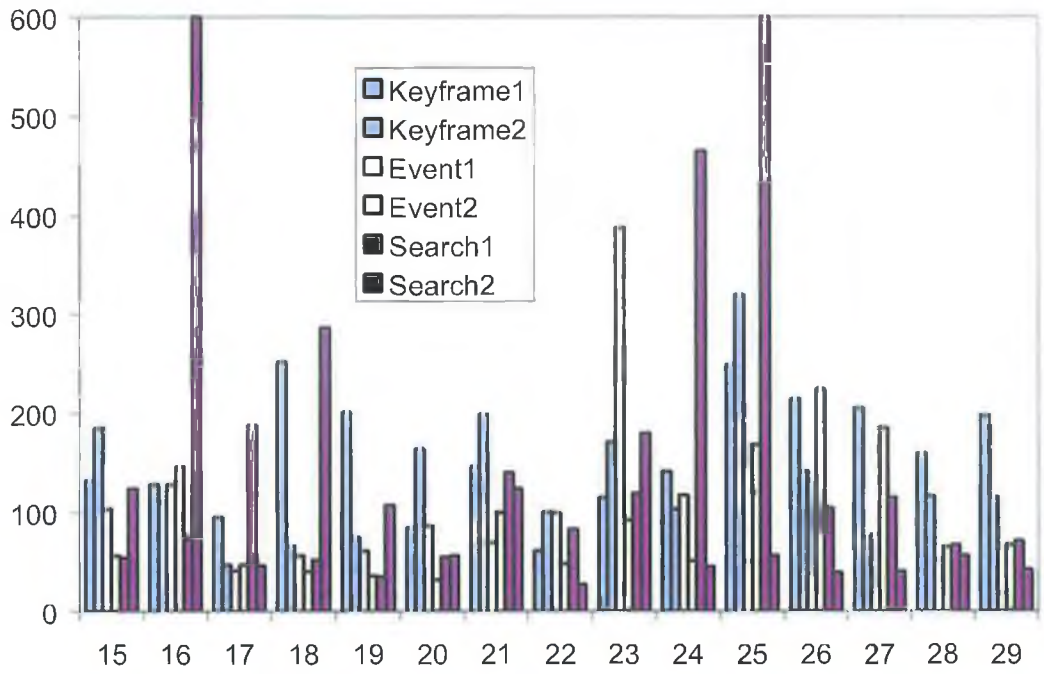

Figure 6.12: Plot of time taken to locate clip for the second fifteen tasks using three browsing methods

although the results returned from searching contained the sought event, these were not recognised. The fastest time to complete a task was 18 seconds for task 29, again using the event based method.

Overall, for all tasks using all methods, the average time to complete a task was 115.3 seconds. The average time for users of the keyframe based method to complete a task was 155.7 seconds. The average time for users of the event based method to complete a task was 81.3 seconds (just over half the average time of the keyframe method). The average time for users of the search based method to complete a task was 109 seconds (just over two thirds of the average time for the keyframe based method). These results are presented in Table 6.2. The search method is heavily penalised by the policy of giving a time of ten minutes for an uncompleted task. If the two uncompleted tasks are ignored, the average time for the search based method falls to 92 seconds.

Clearly is it possible that the results could be biased depending on which films users had previously seen. This effects the results in a number of ways. Firstly, if a user is familiar with a film, he/she may know immediately where in the movie a particular event occurs. Secondly, a user familiar with a film is more likely to recognise a particular event from the keyframes than a user not familiar with a film. As such, analysis of the results

Average time for all tasks =	115.3 S
Average time for keyframe based method =	155.7 S
Average time for event based method =	81.3 S
Average time for search based method =	109.0 S

Table 6.2: Average time in seconds taken for each browsing method

Method Used	**Average Time For Unseen Movies (s)**	**Average Time For Seen Movies (s)**
Keyframe method	187.5	145.11
Event method	111.47	71.27
Search method	174.3	92.7

Table 6.3: Average task completion times by browsing method, where the average results are shown for users that had, and had not seen the film previously

based on whether a person had seen the film was also undertaken. In total, 180 tasks were completed. For 42 of these tasks, users had not seen the film before. Table 6.3 presents the average task completion time for tasks in which the film had, and had not, been previously viewed. As would be expected, the task completion time for each method is higher for users that had not previously viewed the movie. However, note that the average time for the event based method is lowest in both cases. Also, the average time for viewers using the event based method that had not seen the movie before is lower than the average time using the keyframe based method even if the user has seen the movie previously.

From the post-test questionnaires, of the twelve volunteers that took part in the experiment, none thought that the keyframe based system was the best overall. Nine volunteers felt the event based system was the best, and three preferred the search based system. The most common negative point made about the keyframe based system was that it took too long to locate an event, even if the user knew what he/she was looking for. A number of volunteers commented that it was only useful if he/she already had a good idea of what the sought event looks like. The positive comments about the keyframe based system mostly noted that it was straightforward to use, and that all shots were shown to the browser.

Of the nine volunteers that preferred the event based method, the most common reasons given were that it intuitively classified the desired event, and presented the events in an easy to navigate manner. Most users stated that they found it straightforward to categorise any event into one of the three event classes, and then quickly retrieve it. Although there were relatively few negative comments, some users felt that if the desired event was not detected by the event detection techniques, then it would not be possible to find.

Each of the three volunteers that chose the search based system as their favourite retrieval method liked the greater flexibility that this method allows. They also liked that it is possible to refine the search to be quite specific to a particular task. Some volunteers also liked the user driven nature of the search. Many of the other users (those

that preferred the event based system) also liked the flexibility of the search based method, but found it hard to select the correct features for each task, and were worried that if the incorrect feature was chosen, a number of searches would be required. A number of volunteers also noted that more extensive training with the search based method may lead to more efficient searches.

6.3.4 Observations

As expected, the keyframe based method performed poorly, both in terms of the task completion times, and the volunteers' opinions. This is due to the presence of an increased amount of images, a lack of structure, and a lack of semantic meaning. Its performance was worse than the other two retrieval methods by all measurements. Its shortcomings over the other methods of locating events have been illustrated at length previously, and are not discussed further here, other than to say that the predicted disadvantages of keyframe based browsing were observed. Whilst observing the volunteers using the keyframe method, it was noted that users often had to play many parts of the movie to see if they had located the correct event. This resulted in a further lengthening of the retrieval time. A number of times, where knowledge of the film was known, users skipped large portions of the movie, which accelerated the retrieval process. Typically, if a clip was missed on the first pass through the movie, most users simply returned back to the start of the movie and began browsing again, which is time consuming.

By all measurements, the event based method performed best. The task completion times for this method of browsing are consistently lower than for either of the other two methods. On average it is 48% faster than the keyframe based method, and 25% faster than the search based method. For users who had previously seen the movie, the retrieval time was particularly low using the event based system. This indicates that the events detected by the system correspond to the users interpretation of the events, and are located in the correct event class. In comparison, the average retrieval time for the event based method where volunteers had not seen the film before is only 56% greater (40 seconds longer), and in some cases this can be as low as 24 seconds. From observing the volunteers it was noted that typically users did not have any trouble in classifying the sought event into one of the three event classes, even if they had not seen the movie before. In some cases users incorrectly searched in one event class for an event that was detected in a different class. But when this happened, users simply browsed the other event class next and then retrieved the event. Typically, if an event has elements belonging to two event classes it is detected by both systems, however occasionally users misinterpreted the task and browsed the wrong event class. For example, one task involved finding a conversation between two characters where one character is playing a guitar. While the guitar is not central to the event, and in fact is played quite sparingly, the browser incorrectly assumed

that it was a musical event and browsed through the montage events. When the conversation was not found, the dialogue events were perused, and the task was completed. On most occasions users did not have to play the event to confirm its identity. This suggests that the five selected event keyframes accurately summarise the activities in each event.

The search-based method also performed well in most cases. When the users chose features appropriately it provided for efficient retrieval. Some of the tasks suited the search based method more than others. For example locating a song is straightforward, as the music FSM, with little or no filtering, can be used. However, in some cases the search based method can cause difficulty. For example, when searching for a particular conversation, many users chose to use the speech FSM. This typically returns a large amount of events, as speech is a very common feature in a movie. If filtering of these results is undertaken, for example removing all events that do not contain very high amounts of static camera shots, then the searcher may unintentionally filter out the desired event. Similarly, if the incorrect FSM is chosen, the desired event may not be present in the returned events.

In general, the users that performed best using the search based method were the ones that did little or no filtering, and relied heavily on the FSMs. This usually meant a larger amount of events to browse through, but, as the results show, the returned events can be navigated quickly. It was observed that the best search method involved selecting a feature that the user knows is prevalent in the desired event, and then browsing all results. This may in part be due to the user interface of the search based method, as some volunteers noted that it could be made more user friendly. Also, as the search based system requires the most user input, a larger amount of training time may help users familiarise themselves with the system and result in better search queries.

6.3.5 Discussion

A number of conclusions can be drawn from these experiments. Firstly, imposing an event based structure on a movie is highly beneficial in locating specific parts of the movie. This is demonstrated in the higher performance of both the event and search based methods over the keyframe based method. The tasks were chosen so that they could be completed by all systems (as all tasks involved finding a specific part of the movie). In fact the keyframe based system benefits most from this experimental condition, as it means tasks that would be impossible to complete with this system are excluded from the investigation. For example, a task like *Find all locations in the movie where music is used*, would be quite difficult to complete using the keyframe system as users cannot tell from the keyframes whether music is playing or not. This would result in users being forced to play the entire movie. This task would, however, be ideal for the search or event based system. The keyframes chosen to represent the events in general gave users an accurate summary of the shots in the event. Also, the method in which the events

were displayed was informative to users. With very little training, users could easily use the system. However, a number of system improvements, such as altering the searching interface, could be made which may result in improved results.

6.4 Summary

The movie browser system was introduced in this chapter. Three methods of browsing movies are implemented in this system. The first presents one keyframe from each shot to the user. The other two browsing methods were based upon the event detection techniques presented in Chapter 4. These allow users to peruse all detected events for each of the event classes, as well as initiate searches based on a chosen set of features. The movie browser system was then used in order to assess the various browsing methods. A set of tasks was created, and each task was completed a number of times by a set of volunteers using each of the three retrieval methods. After the tasks had been completed, a comprehensive review of the results was undertaken. The event based method achieved the best results both in terms of task completion time and user opinion of the system. The search based method did not achieve results as high as the event based method, but performed significantly better than the keyframe based method. These results illustrate the advantages of having knowledge about the content when browsing keyframes, as well as the benefit in creating an event based summary. A number of areas of potential future work were also discovered based on the user comments regarding the systems.

CHAPTER 7

Conclusions

7.1 Thesis Summary

The first chapter in this thesis introduced the motivation for this work. It was noted that, while in general, automatic indexing of digital video is a difficult problem, the indexing process of movies and fictional television content is particularly difficult due to the originality and creativity with which many of these videos are made. Movies were chosen as the basis for the research in the thesis, although the techniques for indexing movies are equally applicable to the indexing of fictional television content. It was noted that many other forms of video content, such as news or sports, are created and broadcast almost immediately (and in many cases live), so there is a more constrained creative process. With these forms of video, the aim of the broadcast is typically to inform the viewer (news), or show activities as clearly as possible (sports). However, when creating movies the aims are somewhat different, as film-makers strive to make creative, elaborate and entertaining movies. Also in the first chapter, a number of research objectives were outlined. These involved: creating an system that can automatically index movies, placing an event-based structure on video content, implementing a method for searching movies for events, generating a representative test set with which to evaluate the event detection and searching systems, comparing the event detection system, with the state of the art, and finally, creating an user interface that can display the events and facilitate user searching. A set of potential applications were also discussed in this chapter. Following this, a review of the existing state of the art in video summarisation was provided detailing existing approaches to shot-based, scene-based and event-based video summarisation. Finally, the general approach to movie summarisation used in this thesis was presented.

An introduction to film grammar was presented in Chapter 2. Firstly, the general structure of a movie was illustrated and the position of events in this structure was noted. Following this, the roles of three influential people involved in the movie-making process (the director, the editor and the sound engineer) were examined and a number of examples illustrating how they combine when making a movie were presented. Based on film grammar rules a number of system requirements for an effective movie summarisation system were stated. This not only included what structure a movie index should use (an event-based structure was proposed and an ontology of events was chosen), but also how an automatic system could detect these events (the features were selected and requirements for a data classification method were stated). Finally in Chapter 2, based on all of the findings throughout the chapter, an event detection system was proposed, and an overall system structure was proposed.

Chapter 3 presented the audio-visual feature extraction techniques utilised in the event detection process. These features were chosen in order to garner as much information about the intentions of the film makers as possible. Through a combination of colour, motion and audio features, knowledge of the film-makers intent during the movie can be extracted. This chapter contained two main sections, each one corresponding to a module on the system diagram. Section 3.3 described the low-level feature extraction techniques, while Section 3.4 used the low-level features in order to create a shot-level feature vector. The colour features extracted in these sections were used to detect shot boundaries and to create a representative frame (a keyframe) for each shot. The keyframe was later used in order to cluster similar shots together and detect changes-of-focus. The motion features were used in order to accurately describe the movement taking place on screen. This not only involves a description of the movement of objects in the screen, but also a description of the amount of camera movement taking place. The audio features were extracted in order to classify the audio into a number of predefined categories. By using a set of low-level audio features, a number of filters and support vector machines, the audio was labelled as either speech, music, quiet music, or silence. The output of this processing is a shot-level feature vector.

In Chapter 4, the final component of the system, the high-level event detection module, was described. Firstly, a set of potential sequences were generated using an array of finite state machines. Each FSM created a set of potential sequences using specific input features. By examining the potential sequences, filtering them, and combining them, a list of dialogue, exciting and montage events were created. By detecting these events, a complete event based index was created for the movie. A search based system was also implemented based on the underlying event detection system, which allows users to retrieve events based on a set of specified features. Finally, in order to later benchmark the FSM approach to event detection, a hidden Markov model approach that does not

leverage film grammar was also presented.

A comprehensive analysis of the FSM event detection approach was presented in Chapter 5. There were a number of steps involved in fully assessing the system. Firstly, the results of the FSM system were compared to a manual annotation. Secondly, different user annotations were compared, and the systems ability to cope with different interpretations was investigated. Thirdly, the shots that were detected as belonging to more than one event class were compared, and the unused shots were located and described. Fourthly, the event detection techniques were applied to a set of fictional television programmes. Finally in Chapter 5, the results of the hidden Markov model approach were presented and compared with the FSM approach.

The comparison of the FSM results with a manual annotation proved that a high recall rate is achieved across all event types. By examining different interpretations of the same film, it was noted that a slightly lower precision value was necessary in order to accommodate different interpretations of events. Also, an event classification overlap was noted, and indeed desired, in order to allow different users to have different interpretations of the same event. The FSM approach was deemed to have reliably indexed the events in the movies tested as the vast majority of the shots in each movie were classified into one of the three event classes. When the event detection techniques were applied to fictional television content, higher values were obtained for precision and recall. This not only indicates the versatility of the event detection approach, but also that television content is produced in a more constrained manner than movies. By comparing the FSM and HMM systems, the benefits of basing a film summarisation system on film grammar principles were illustrated.

In Chapter 6, the movie browser system was presented. This system allows users to browse movies by either browsing all keyframes, browsing the detected events, or by searching the movie using a set of selected features. Firstly, this system was intended to show one possible application of this research in a real world environment and, secondly, it was created in order to conduct user experiments that assess different methods of movie browsing. User experiments were undertaken in which a set of volunteers were asked to complete a number of tasks using each of the three systems. The time to complete each task was recorded so that a direct comparison between different methods of browsing could be made. Also, each user completed a post experiment survey in which reasons for system preference were stated. The results of these experiments indicate that creating an event based index is an effective method of summarising movies.

7.2 Objectives Achieved

The primary aim of this research was to create a system that is capable of automatically indexing movies and fictional television content. In order to achieve this aim, it was proposed to use an event-based structure that would detect the meaningful events in a movie according to a formulated ontology. To this end, an event detection approach that utilises audio-visual analysis based on film creation techniques was designed and implemented. Chapter 2 described how an event-based ontology was formulated, while the following chapters detailed how a system that is capable of detecting the events in this ontology was designed and implemented. A representative set of movies of differing styles, origins and genres was created and used in order to examine the systems operation. As can be seen from the results of Chapter 5, the event detection technique itself is successful. A high detection rate was reported for all event types, with each event detection method achieving over 90% recall. As the results of Section 5.5.2 show, there is only a small amount of shots that are not classed into one of the event classes (less than 9% of shots). This indicates that indexing by event is an efficient method of structuring a movie and also that the event classes selected are broad enough to index an entire movie.

Although the high values of precision and recall proves that the finite state machine approach to event detection is successful, it does not justify the concept of indexing a movie by event (as opposed to a different indexing method). However, as can be seen from the results of Chapter 6, the event detection technique results in fast and reliable user retrieval of events. The experiments using the movie browser system, where users were asked to complete a set of tasks using keyframe-based, event-based and search-based retrieval methods, assessed which retrieval method is most efficient. By all measures the event-based method outperformed the other methods. The average task completion time was significantly lower when using the event-based system than for the keyframe-based system and, to a lesser extent, the search-based system. Also, most users were more comfortable with the event-based system than with any other system. This proves the effectiveness of event-based indexing as a general approach. Users could locate events quickly, which indicates that the event classes chosen were intuitive to use and it is possible to class a sought event into one of the three event classes.

Also, as detailed in Chapter 4, an alternate approach to event detection was created in order to benchmark the FSM approach. This system used a hidden Markov model in order to detect the events in a movie. The results of both systems were compared and explicitly illustrated the benefits of the FSM approach over the HMM approach.

Another research objective involved creating a system capable of searching movies and retrieving events. This system was also implemented successfully and incorporated into the movie browser interface. During the experiments with the movie browser, this system often resulted in an extremely short search time, especially in cases where users

chose features that accurately represent the sought events.

Although this research was primarily focused on movies, another aim was to investigate the use of the event detection methods on non-movie fictional video content. Section 5.6 presented the results obtained when the event detection techniques were applied to television programs. These results were slightly higher than those for the movie data. This illustrated the use of more constrained filming methods when creating television programs, but also the relative creativity with which movies are made.

Thus, the objectives of the research as specified in the introduction chapter have been met. An event-based ontology was created, a system for automatically indexing movies was created, an event-based structure that can accurately detect the relevant events in a movie/fictional television program was created, a representative test-set of movies was created, the approach was compared with the existing state of the art, a user-specific search-based system was implemented, and finally, an interface that allows users to browse movies by event as well as initiate event-based searching of movies was produced.

7.3 General Conclusions

There are a number of implications of this work. Firstly, the general concept of an event-based approach to movie indexing has been validated. The event detection technique classifies the vast majority of any movie into one of the defined classes and in general, based on the results of the user experiments, users find it straightforward to associate an unseen event with one of the classes and locate it. Thus, an event based index allows for efficient retrieval in movies.

In terms of analysing movie content, this research demonstrated that an understanding of movie creation techniques is very important. A number of the system design decisions were made primarily based on how movies are created. For example, by examining how events in each of the three event classes are shot, it was possible to select appropriate features for each event class. The values of these features during a movie gave strong indications as to the film-makers intent and, from this, knowledge about the event class could be inferred. Similarly, the fact that film-makers shoot events differently depending on the context, means that detecting the individual components of the events, rather than the event as a whole, results in more accurate results. This indicates that an in depth knowledge about video creation techniques (be it for movies, sports, news etc.) significantly assists analysis.

By contrasting the film grammar based FSM approach with the HMM approach it is possible to directly see the effect of failing to take into account the content creation process when creating an analysis system. Although the same feature set was used for both systems, the HMM approach failed to take into account the fact that events can be

shot in different ways. The HMM event detection system was based on a set of training examples, so when detecting events on new data it assumed that they would be shot in a similar manner, which is not always the case. Also, as the HMM approach examined the events based on a combination of features, it could not handle cases where a number of the features were not present, or where the features varied. In contrast, based on the film-making conventions presented in the film grammar chapter, the FSM approach targets the individual components of each event. Similarly, FSMs are not reliant on any training data, as their designs are tailored based on each feature and how it is used in a movie. The combination of these factors resulted in significantly higher results.

The results of Section 5.4 illustrate the need to facilitate different user interpretations in any video summarisation system. Although there is a strong overlap between the annotations of different users, a number of differences exist. This highlights the strong subjective nature of movie interpretation. When events are labelled according to an ontology which contains terms such 'exciting' or 'emotional', interpretation is clearly subjective. Thus, it is important to ensure that different user interpretations are facilitated without over-classifying the data. In the context of this work, this meant that some events are classified into more than one event class.

Another outcome from the experiments with the movie browser system was the encouraging results of the search based system. Although relatively little training was provided, a number of volunteers preferred this system to the other two browsing methods and it often resulted in short search times. This indicates that some users prefer controlling the type of events that are presented and also, once some familiarity with the features is obtained, users can initiate searches that result in efficient retrieval.

7.4 Future Work

As mentioned in the introduction chapter, there are significant benefits to a student of film in having an indexed, searchable movie, as opposed to a movie file found on a DVD. In general, more efficient analysis can take place as events can be quickly located. Also, user specific searches can be undertaken that allow browsers to locate events that contain specified features. The School of Communications in Dublin City University expressed an interest in using the developed movie browser system for teaching purposes. A set of movies used for teaching were automatically indexed and ingested into the movie browser system. This system was installed into a student laboratory, and is currently in use by both students and staff. As the movie browser system was not created with teaching specifically in mind, future work will alter the system to make it more applicable to film studies students. For example, one user commented that in order to analyse editing pace in a movie, currently many students manually locate the shot cuts in a movie and time

each individual shot length. The simple addition of an editing-pace graph using the shot boundary information would considerably reduce the effort required for this. Further adjustments will be made to the system, which may involve tweaking the event detection and presentation techniques based on the feedback from users. Also, as indicated by the feedback from Chapter 6, adjustments to the searching interface may yield improved results.

Although it is not envisaged that additional features would improve the event detection process, by adding data to the detected events a number of applications could be built. For example, by adding in the subtitle information (available on virtually all DVDs) or data from a movie script to the dialogue events, extra knowledge could be garnered about the dialogue events. If this was incorporated into the movie browser system presented in Section 6, it could allow users to not only browse all dialogue events, but to browse all dialogue events that contain a user specified keyword. In terms of video summarisation, many techniques have previously shown the advantage of text based retrieval when compared to visual based retrieval. In the TRECVID conference, where a set of retrieval tasks are set on a large video dataset, it has been continuously shown that text based approaches perform better than visual only approaches. Thus, it is envisaged that the addition of text information to the movie browser application will result in even more efficient retrieval.

Another feature that could be built upon the event based indexing involves face recognition. If a technique that reliably recognises actors faces from the keyframe image could be implemented, this again would add more information to the event index. For each event, a list of actors that appear could be presented and users could, for example, search for exciting events that contain a particular actor. This again would make the retrieval process more efficient. Note in both of these cases (i.e. the addition of text and face recognition), that additional features do not improve the event detection process, but do add relevant metadata to the events once they are detected to allow for a more descriptive index.

It may be possible to extend the event class ontology to include a set of sub-classes. For example, many events were detected as belonging to two event classes as they contained elements of both classes. By creating a set of sub-classes, that further classify the events, a richer description of a movie could be generated. These sub-classes would contain exciting dialogues (e.g. an argument), montage dialogues (e.g. an emotional conversation) and exciting montages (e.g. a musical event with high amounts of action).

Using the current state of the art, it may be possible to achieve many of the aims listed above, however, if the current audiovisual analysis limitations were removed, a much higher level of semantic knowledge could be extracted from a movie. For example, much of the audience's interpretation of a movie comes from the actors actions and reactions. A sinister facial expression of an actor gives the audience an insight as to the characters

villainous intentions. Contrarily, if a character is seen to be smiling, this indicates happiness. If a character's facial expression could be detected and classified, this would lead to the extraction of emotional information that could further enhance the event based index.

Similarly, if the actor's speech could be further analysed in order to extract the tone of the actors voice, insights into the nature of the spoken words could be gained. For example, distinguishing between when an actor is telling a joke and when he/she is giving an order. This, again, would lead to a richer semantic understanding of the occurrences in a movie, which would in turn assist in more efficient retrieval.

It is an assertion of the work in this thesis that an event-based index of video content are superior to a structural index (i.e. shot or scene based indices). The various experiments of Chapters 5 and 6 demonstrate that this assertion is true. Thus, it is the authors opinion that in order to create more efficient video summaries, a shift from the conventional structural index is required and an increase in the amount of research on event-based summarisation is necessary.

Much of the work in this thesis involved selecting and extracting a set of audiovisual features for use in inferring semantics about the movie content. Although this feature selection process is tailored for the research in this thesis, a common set of feature extraction algorithms would assist in the creation of video retrieval systems. If an open-source software tool were available that facilitates the extraction of audiovisual features, researchers could concentrate their attention on extracting a higher level of meaning from the video content. MPEG-7 supports low-level feature extraction, however, a set of features specifically selected for movie analysis would allow research to focus on obtaining semantics from the content rather on video processing.

All too often when creating video retrieval systems, the research does not take into account the eventual benefactors of the work, i.e. the end users. This may lead to blinkered research that does not reflect real user requirements. An increase in the level of collaboration between video retrieval experts and the eventual users may avoid this. For example, in order to create an ontology of events classes in this work, the process of creating a film was examined. If a set of experts in the field of film-making were to create a definitive ontology for movies, this would allow video retrieval experts to focus on creating algorithms to detect the events in this ontology.

Also, in order to compare different event-based approaches, a more co-ordinated effort is required. In order to compare approaches to shot boundary detection, feature extraction, and searching of news content, the TRECVID workshop supplies participants with a common video data set, and a common set of tasks. Thus, each approach can be evaluated and results can be compared. Creation of a TRECVID style forum with a common set of video, containing movies and fictional television content, would allow researchers in this field of video retrieval to compare and contrast different approaches, and to jointly move

forward in creating superior video retrieval methods.

APPENDIX A

Video and Audio Coding

A.1 Overview

This section discusses how digital video and audio is created, and also the various techniques and algorithms employed when compressing data. Although not all of the video and audio coding elements are specific to the research presented in this thesis, for completeness general video coding concepts are also discussed.

Although some degradation of quality will occur when compressing video, the huge reduction in size of the content, combined with the comparably small decrease in quality (for high bit-rate applications), make this trade off very desirable. The underlying representation of video is discussed as this is important in the context of designing the analysis techniques used in Chapters 3 and 4.

A.2 General Principles

A digital image is created when an analogue signal is generated by a camera scanning a 2-D scene and converting it to an electrical signal. In order to create video, successive scans are taken and simply presented sequentially to the viewer. A digital image is composed of a large number of elements called pixels. These pixels are assigned colour values (e.g. each pixel may be made up of a combination of red, green and blue (RGB)) and displayed on the screen. Clearly, the more pixels that can fit into an image the better the resolution and the sharper the image will look. However, since each pixel contains three values, the more pixels the bigger the size of the image file. In general, video and audio coding techniques aim to reduce the size of the content using an array of compression techniques.

Once the files are compressed, in order to display them they must be decompressed. A compressor-decompressor (codec) is used for this purpose.

A.2.1 Lossy and Lossless Compression

Typically when compressing images or video, some degradation of the picture quality can be tolerated. Although the degradation should not result in unviewable video, it is usually necessary in order to substantially reduce the size of the file. Lossy compression allows some degradation in order to compress images and video. In some applications however, no degradation of the picture can be tolerated, and lossless compression must be used (in this case no visual information is removed). For example, a hospital may require that a CAT scan image be delivered to a doctor with as much detail as possible. Although lossless images are essential in many applications, by far the most popular way of encoding images/video is by using lossy compression. Typically, lossy encoders aim to discard the least important (to a human observer) visual data and keep as much of the important data as possible. In lossy compression, the expected reduction in the quality of the picture is justified by a large reduction in file size. All of the encoding principals in the following sections are used in the process of lossy encoding.

A.2.2 RGB to YUV Conversion

The first step in reducing the size of the video usually involves transforming data from the RGB colour space to the YUV colour space. Like RGB, the YUV colour space has three values for each pixel. The Y component is the luminance (brightness) of the pixel, and the U and V components represent two colour values. The Y value on its own is a greyscale version of the video. A straight conversion from RGB to YUV without any loss of information can be achieved using the formulae A.1, A.2 and A.3 [62].

$$Y = 0.299R + 0.587G + 0.114B \tag{A.1}$$

$$U = -0.169R - 0.331G + 0.500B \tag{A.2}$$

$$V = 0.500R - 0.419G - 0.081B \tag{A.3}$$

This conversion does not lose any information, as it is possible to convert back to RGB using a different set of formulae. However, due to the fact that the human visual system is much more sensitive to the brightness (luminance) changes in an image than to colour changes, it is possible to discard much of the U and V colour components. Although this obviously reduces the resolution of the image, it is not noticeable to a human observer. For example, often a video encoder will subsample the U and V components by a factor of two, while maintaining the Y component at full resolution. This will mean that the YUV

Figure A.1: RGB to YUV conversion with the U and V components sub-sampled by a factor of two.

data is half the size of its RGB counterpart. This can be seen in Figure A.1. In Section 3.3.2, the YUV values are used to generate a colour histogram value of each frame, which is used as the sole colour representation for a frame of video. As the video is represented in YUV format it is possible to place more emphasis on the Y (luminance) values than U or V (colour) values when analysing the visual component of the video.

A.2.3 Spatial Redundancy

One way to exploit the nature of video in order to reduce its size, is to exploit the amount of spatial redundancy present. Typically, regions that are close together in an image/video contain similar pixel values, so one way to exploit this spatial redundancy is to map the pixels into the frequency domain prior to data reduction. As the image energy of most natural scenes is mainly concentrated in the low frequency region, there is a reduced amount of relevant transform coefficients. This means that the coefficients can be quantised, which removes insignificant coefficients without greatly affecting the original pixel values. This is a lossy method of data reduction, since, once quantised, the original pixel values cannot be reconstructed exactly [2].

The Discrete Cosine Transform (DCT) is the most commonly used transform for spatial redundancy exploitation. Typically, the frame is split into a number of 8×8 regions and the two dimensional DCT algorithm is implemented on these regions [2]. Essentially, the DCT algorithm shifts the more important low frequency components of the image into the top left of the 8×8 region. Thus, it is possible to filter the less important high frequency components, without significantly degrading the quality of the image to a human observer. In order to achieve this filtering, firstly, the pixels are quantised. The image is then passed through a low pass filter, which removes the unrequired high frequency components of the image. These steps can be seen in Figure A.2. Both images are in the frequency domain. As can be seen, most of the information is contained in the top left of the DCT image. The image on the right shows how a filter could remove high frequency information. Clearly, the more high frequency components that are removed, the more degradation (see Figure A.3 for two images after DCT filtering, the one on the right has been subjected to a higher level of filtering).

Source coding is then used to re-arrange the 8x8 block into a one-dimensional array

Figure A.2: DCT domain representation of an image and the resultant filtering.

Figure A.3: The effects of increasing the amount of quantisation and filtering on an image

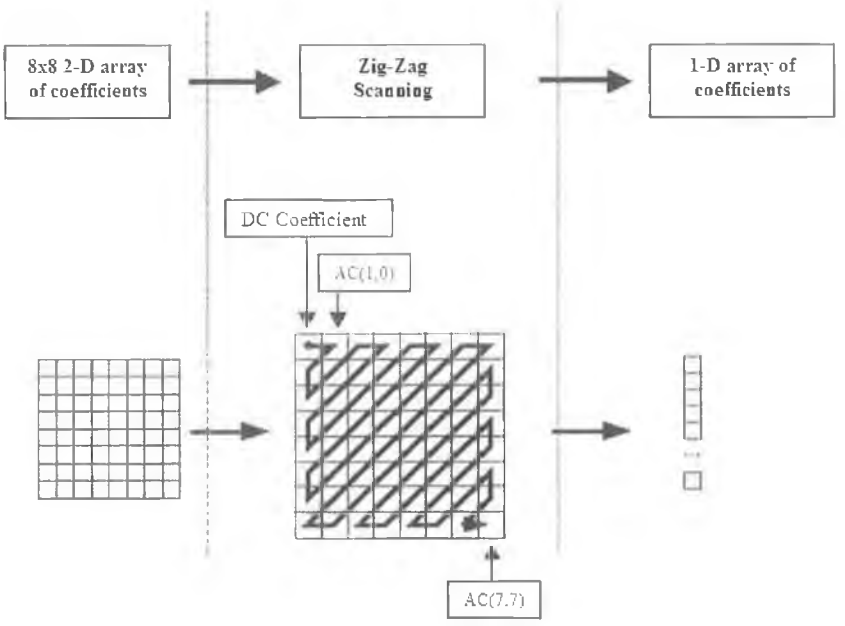

Figure A.4: Source coding [1]

by scanning the blocks in a certain sequence (this is usually zig-zag, as shown in Figure A.4). This rearrangement puts the DC (average) coefficient at the first location of the array and the remaining AC coefficients are arranged from low to high frequency, in both the horizontal and vertical directions. The assumption is that the quantised DCT coefficients at higher frequencies will likely be zero, thereby separating the non-zero and zero parts. The rearranged array is coded into a sequence of run-level pairs. The run is defined as the distance between two non-zero coefficients in the array. The level is the non-zero value immediately following a sequence of zeros. This coding method produces a compact representation of the 8×8 DCT coefficients, since a large number of the coefficients have been already quantised (and filtered) to zero value.

A.2.4 Temporal Redundancy

Video is typically shown at a frame rate greater than 25 frames per second. This means that in successive frames, there is a usually only a small change in the image content. For static parts of the frame, the difference from frame to frame will be close to zero. Therefore there is a significant inter frame temporal redundancy. Instead of encoding each frame in a video independently, it is more efficient to calculate the *difference* between each frame, and encode that. In order to calculate this difference, an estimate of the motion present is calculated, and then a motion compensated image is generated as the prediction [2].

For motion estimation, there must be a reference frame present from which the motion

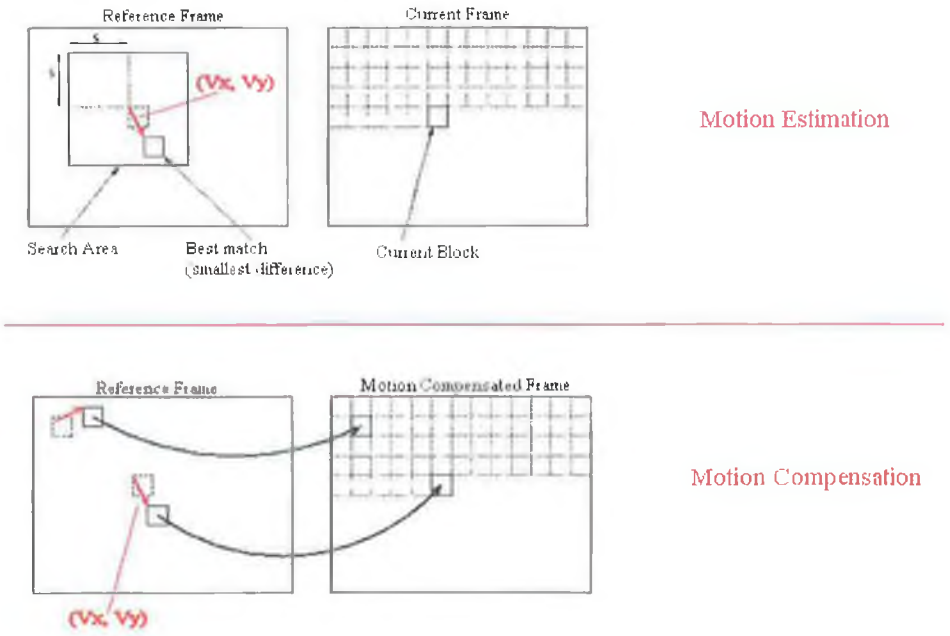

Figure A.5: Motion estimation and compensation [1]

of the current frame is estimated. The most commonly used motion estimation technique is the block matching algorithm. In this algorithm, the reference frame is first divided into square blocks of pixels. The difference between the reference frame and the current frame is assumed to be solely due to the presence of motion. Motion estimation involves taking one of the square pixel blocks in the current frame, and looking for the closest match in the reference frame (block matching). This search is usually restricted to a certain area around the block under inspection. Using the displacement between the location of the current block and that of the matching block, a *motion vector* is generated. Using these motion vectors and the reference image, the current image can be reconstructed. In Figure A.5 the value (V_x, V_y) represents the motion vector for a particular block, and is illustrated by a red arrow. Motion information about the video is extracted from the motion vectors to assist in the visual analysis in the proposed approach. As can be seen in Section 3.3.3, information about the amount, and type, of motion present on screen is extracted.

Many video codecs have two different types of frames. *Intra* frames and *inter* frames. Intra frames are frames that are coded independently without any motion estimation or motion compensation, while inter frames use motion compensation. The Inter frames may generate motion vectors from previous or future frames. As well as being an efficient method of compressing video, motion vectors can also be interpreted as a representation of the motion throughout a frame, and therefore throughout the video. As these motion vectors can be extracted, it is possible to automatically detect areas that contain high/low motion, as well as defining the type of motion, based on the generated motion vectors. The use of motion vectors to analyse the motion of a digital video is quite common. As

well as representing the motion in a video, using the embedded motion vectors has the advantage of not requiring a full decode of the video and is therefore quite fast.

A.3 MPEG

A.3.1 MPEG Family

The Motion Picture Experts Group (MPEG) is a working group of ISO/IEC charged with the development of standards for coded representation of digital audio and video. The first video standard defined by MPEG was MPEG-1. Video CDs and MP3 audio are based on the technologies in this standard. All of the visual analysis carried out in this research uses MPEG-1 video and so it is the main focus in the remainder of this section. The second standard defined by the group, MPEG-2, targeted higher quality video applications. It supports a wide range of possible bitrates and framerates. It is the main technology behind the coding standard on DVDs. MPEG-4 is the most recent video coding standard ratified by MPEG. As usual with MPEG standards, much of the MPEG-4 standard is based on predictions as to where the world of digital video is heading.

Finally, MPEG-7 and MPEG-21 are also standards defined by MPEG. However, they are not video coding standards, they are standards for associating metadata with, gaining information about, and delivering, multimedia content.

A.3.2 MPEG-1 Video Codec

The general principles presented thus far in this section are common to many video standards. All of the research in this thesis is carried out on MPEG-1 video. MPEG-1 was designed to deliver digitised and compressed video signals up to a data transfer rate of 1.5 Mbits per second. One of the problems with defining video standards is that the target usage is unknown. MPEG-1 was created in the early 1990s when the most advanced computers at the time are now dwarfed by even the most basic computers in terms of processing power, speed and storage space. Yet MPEG-1 is still widely used today, albeit with several amendments. One of the main reasons for this is the fact that MPEG-1 supports a wide range of applications and supports flexible picture size and frame rate. Among the other attributes of MPEG-1 is its support for fast-forward and fast reverse, frame based random access, and editing of streams. MPEG-1 is based on a macroblock structuring of an image (Y=16x16, U, V = 8x8).

MPEG-1 has a hierarchically structured bitstream. The lowest level is a *block*. This is made up of 8×8 pixels. A *macroblock* contains 16×16 pixels. These macroblocks are combined together in a horizontal fashion to create *slices*. A *picture* (or frame) contains a configurable number of these slices. The slice layer is essential in maintaining

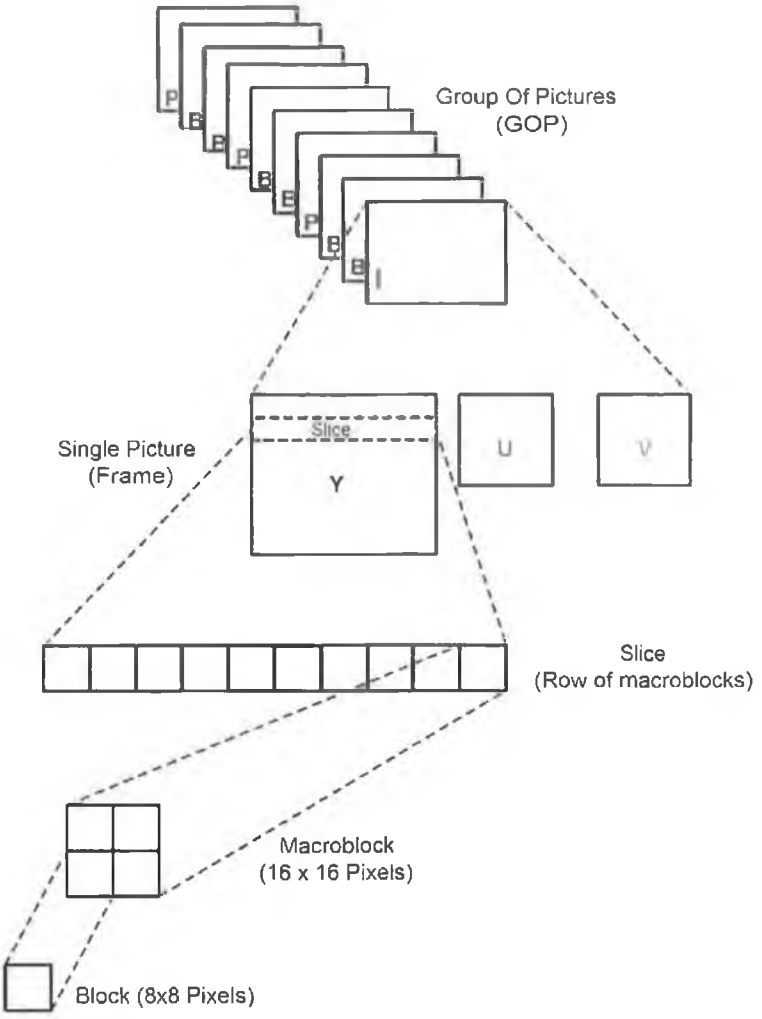

Figure A.6: MPEG-1 Hierarchy Structure (adapted from [2])

synchronicity and enhancing error resilience. For example if some data becomes corrupt or is lost (which is a distinct possibility when streaming data) the decoder can resume decoding at the next slice, instead of losing the whole frame. A picture can either be an intra frame or an inter frame. Finally, the highest level is the *Group Of Picture* (GOP) layer. This contains a number of successive pictures. The structure can be seen in Figure A.6 [2].

MPEG-1 uses both intra and inter frames. There are three frame types in MPEG-1:

- 1) I-Frames - These are intra coded frames, i.e. are coded independently of other frames.
- 2) P-Frames - These are inter frames, and are coded on the basis of the prediction from the previous I (or P) frame.
- 3) B-Frames - These are inter frames coded on the basis of a combination of the

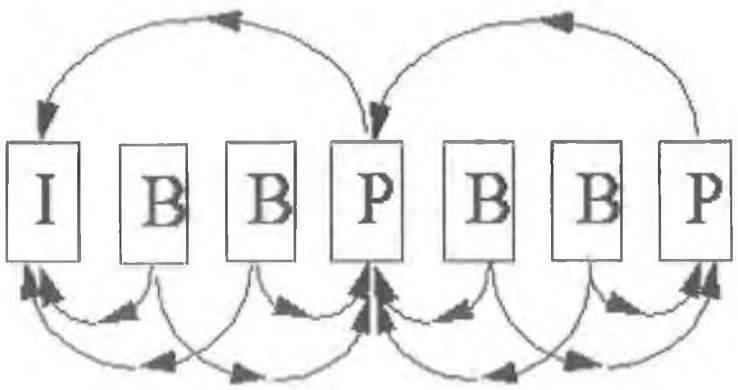

Figure A.7: Possible Repetitive Configuration of I,P and B Frames in an MPEG-1 Video Sequence [1]

previous frame and the next frame in a video sequence.

Within each inter frame, the encoder may decide that individual macroblocks should be intra coded. This may occur due to excessive motion which would make motion estimation unreliable. In Figure A.7 there is an example of a possible layout of I, P and B-Frames in an MPEG-1 video sequence. The arrows show the dependence of the frame. For instance the I-frame doesn't depend on any of the other frames as it is independently coded. The P-frames depend on either the last I-Frame, or the previous P-frame, and the B-Frames depend on both the next I or P-Frame, and the previous I or P-Frame, but never on another B-Frame. As they contain much motion estimation information, P-frames can be used as a representation of the motion in video. Their frequency and accuracy make them ideal representations of the movement in the video. The steps involved in extracting the motion information for analysis can be seen in Section 3.3.3.

A.3.3 MPEG-1 Encoder and Decoder

Figure A.8 shows a block diagram of a complete MPEG-1 encoder. Its use is illustrated by explaining the process of encoding a macroblock. First, the coding mode is chosen, this depends on the picture type (I/B/P) and the effectiveness of the motion compensated prediction. Secondly, depending on the coding mode, a motion compensated prediction of the contents of the block based on past and/or future reference frames is formed. This prediction is subtracted from the actual data in the current macroblock to form an error signal. Thirdly, this error signal is divided into 8×8 blocks and a DCT is performed on each block. The resulting 2-D block of DCT coefficients is quantised and scanned in zigzag order to convert it into a 1-D string of quantised DCT coefficients. Information about the macroblock (type, block pattern etc.) is also coded. Finally, all of this

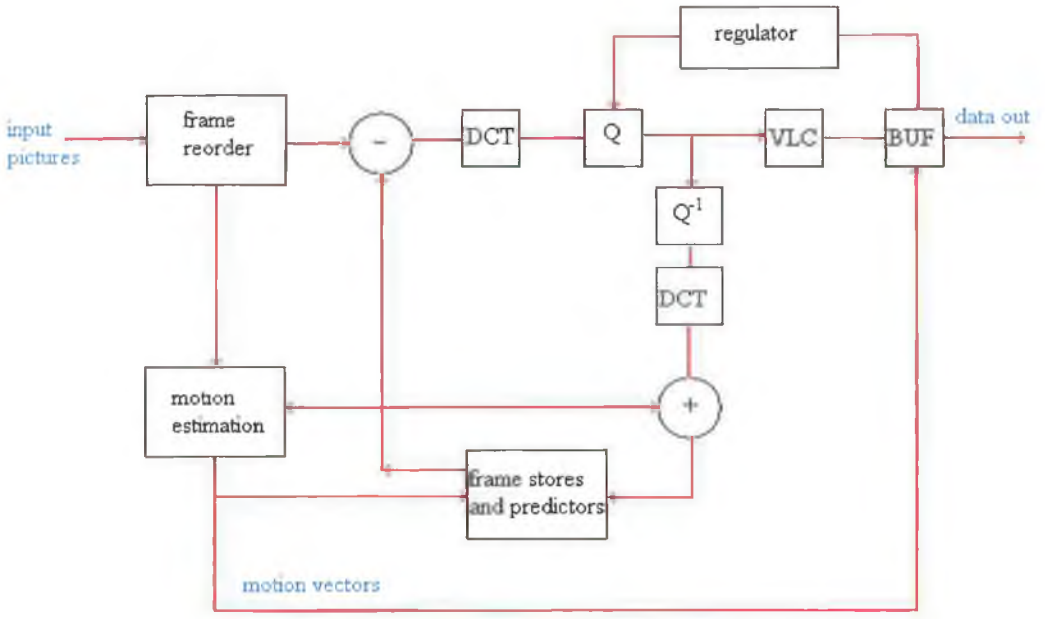

Figure A.8: MPEG-1 Encoder Structure [2]

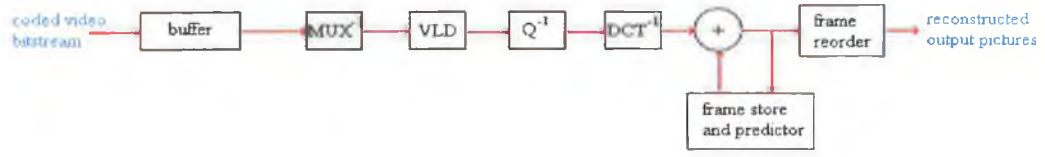

Figure A.9: MPEG-1 Decoder Structure [2]

information is variable length coded[1] for maximum efficiency [2].

An MPEG-1 decoder is shown in Figure A.9. The incoming bitstream is stored in the buffer, and is demultiplexed. These values are variable length decoded using tables known to the decoder. The values are then inverse Quantised and the inverse DCT is also applied. These values are added to the motion compensated prediction (if required) in order to reconstruct the pictures. The frame store then holds the decoded I or P-frame (there is no need to store a B frame since it is never used again). Finally, the frames are reordered and are now presented to the user [2].

As can be seen from the decoder structure, in order to extract the motion vectors, a full decode is not required. Therefore, examining the motion vectors is an efficient, as well as an accurate, method of extracting the motion information from the video. In order to extract information about the colour of a video frame, a full decode is required. However,

[1]Variable Length Coding (VLC) is a process in which short code words are assigned to the highly probably values, and long code words to the less probable values

once this is done each individual pixel value is known. This facilitates in depth analysis of the colour values. Details of how an array of colour and motion features are extracted are presented in Sections 3.3.2 and 3.3.3.

A.4 Audio Coding

Similarly to digital video, many parameters can be altered when creating digital audio. It is possible to create low bit-rate audio when memory or transmission speed is at a premium. Conversely, when creating consumable audio, the encoding parameters can be set to ensure higher quality. Audio CDs are encoded at a high sample rate to ensure a very high quality level of sound, however, audio data can also be compressed significantly using a array of techniques. Many efficient compression standards have facilitated the creation of extremely high quality yet compact audio files. These audio compression techniques have much in common with digital video compression, in that they both aim to remove as much irrelevant information as possible, while retaining the important (for the viewer/listener) data. The transmission of compressed audio files is now widespread, with a number of on-line music stores now in existence. Also, many websites, radio stations for example, stream live audio over the Internet. All of the audio analysis in this thesis was carried out on WAV audio, so only this digital audio format is discussed further here.

WAV is a Microsoft and IBM audio file format widely used for storing and recording digital audio. Although the WAV format can handle audio created by any codec, the most common format, and also the one used in this research, is pulse-code modulation (PCM) audio data. PCM coded audio was chosen as it maximises the audio quality. PCM is an uncompressed, lossless method of storing audio. This means that all samples on the audio track are retained. The quality of the PCM audio is dependent on the amount of bits allocated per sample and its sampling rate. The more bits that are allocated per sample, the more accurately the input signal is represented. The sampling rate is simply the amount of samples per second taken from a continuous (analogue) signal to create a digital signal. The more samples taken per second, the higher the quality. The unit of measurement is Hertz (Hz). For example, a sampling rate of 8,000 Hz is commonly used for telephone communications, as it is deemed adequate for human speech. For applications that require higher audio quality, the sampling rate can be increased. For example, the sampling rate used for audio CDs (which are also coded using PCM) is 44,100Hz, while for digital television broadcasts and DVDs, a sampling rate of 48,000 Hz is used. PCM based WAV audio was chosen as the format for all audio analysis due to its accurate representation of sound. As no data is thrown away, it is possible to extract a number of reliable audio features which are in turn used for audio classification, see

Section 3.3.4 for further details.

APPENDIX B

Introduction to Support Vector Machines

A Support Vector Machine (SVM) is a supervised-learning data classification tool used primarily for binary classification. It is a versatile solution to the data classification problem that has found applications in many areas such as object recognition, face detection, text characterisation, and speech detection. This section describes the approach of SVMs and then explains how they are trained and tested for binary classification.

B.1 General Approach

An SVM tries to find a functional relationship between the input values and the output labels, this is called a *decision function*, since it makes a decision based on the inputs, as to the class of the output. A SVM uses a set of training data in order to come up with this decision function. The ability of a SVM (or any other learning machine) to learn a training set is called its *capacity*. SVMs with high capacity can efficiently learn a training set without error, no matter how diverse. As shall be seen in a moment, high capacity does not necessarily equate to high performance, a better way of measuring the performance of a SVM is by its *generalisation*, which is the ability of the SVM to classify new data not in the training set [58].

If the capacity of a SVM is too high, then it is in danger of *overfitting* the data. This occurs when the decision function becomes too closely related to the training data and will only correctly classify data points that appear in the training set (or that are very closely related to data points in the training set). This results in poor generalisation performance. [71] compares a learning machine with too much capacity as a botanist with a photographic memory who, when presented with a new tree, concludes that it is not a

tree as it has a different number of leaves than any tree he has seen before. A machine with low capacity is like a lazy botanist, who declares that if it is green, then it is a tree. Machines with either too high or too low capacity generalise poorly. An SVM will aim to find the correct balance between the accuracy attained on a particular training set, and the capacity of the SVM. In theory, this will result in the best generalisation [71].

B.2 Generalisation Theory

A SVM accepts inputs and outputs in the form (x_i, y_i), where $x \in R^N$, and $y \in \{1, -1\}$. When training, the aim is to learn the mapping between a set of observation inputs/outputs and create the decision function[1]. The task is to choose, from all possible hypotheses, the one that minimises the risk of error in the classification of test data, as this will lead to the best generalisation performance [58].

In order to chose the best solution, an SVM attempts to find the model that will result in the least amount of classification errors. α is defined as the set of adjustable parameters in the trained learning machine. The actual risk of error, $R(\alpha)$ is the probability that a test point will be classified incorrectly by the SVM. $R(\alpha)$ cannot be calculated directly, as is depends on an unknown probability distribution. However, an empirical risk of false detection $R_{emp}(\alpha)$ can be calculated using the training observations. The following inequality holds with probability 1-η [71]:

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{(h(\log(2l/h) + 1) - \log(\eta/4))}{l}}$$

This shows that the actual risk of error is bounded (within a certain probability) by the empirical risk plus a term known as Vapnik Chervonenkis (VC) confidence (the right hand side of the inequality). By minimising the VC confidence, the risk of error can be minimised.

The VC confidence is dominated by the term h/l. l is the number of training points, while h is known as the VC dimension of a set of functions. The lower this ratio h/l, the lower the VC confidence becomes, therefore reducing the actual risk. For a given set of l points, which can be labelled in 2^l possible ways, if for each labelling a member of the set of functions $\{f(\alpha)\}$ can be assigned which correctly assigns the labels, then the set of points is said to be *shattered* by that set of functions. For example, see Figure B.1, for a set of $l = 3$ points that can be shattered by oriented lines. The dividing lines are known as *hyperplanes* in higher dimensions. There are $2^3 = 8$ ways that the three training points can be classed into two categories, and each of these can be correctly separated by

[1] Actually, the decision function is an estimate of the *target function*, which is function that represents the best possible mapping between inputs and outputs

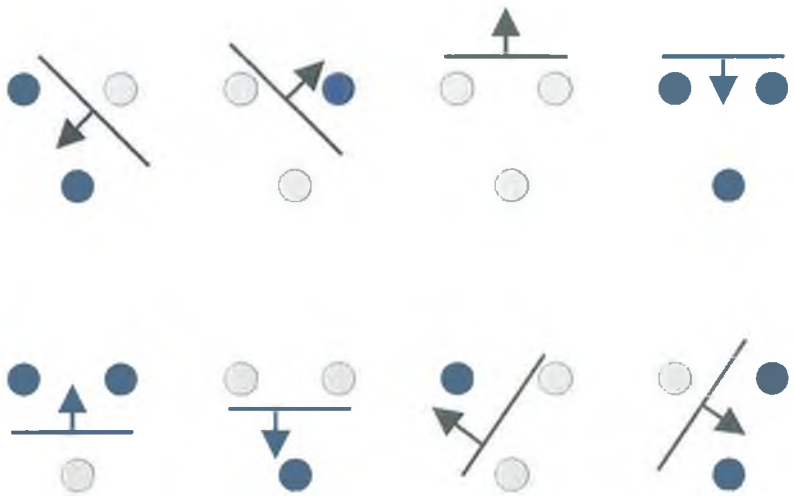

Figure B.1: All 8 possible combinations of three points, shattered by oriented straight lines. Dark circles are positive data points.

Figure B.2: One possible configuration of 4 points that cannot be separated by a straight line.

a straight line. Note that if there were $l = 4$ points there would be possible combinations of classes that could not be divided by a straight line (Figure B.2 shows an example of one such configuration). The VC dimension of a set of functions is defined as the maximum number of training points that can be shattered by that set of functions. As can be seen from Figures B.1 and B.2, the VC dimension of the set of oriented hyperplanes in R^2 is 3. In fact, the VC dimensions of the set of hyperplanes in R^n will always be $n + 1$. The VC dimensions can be seen as a measure of capacity, since a set of functions with infinite VC dimensions can learn any set of training points correctly [58, 71].

B.3 Linear SVMs

For a set of linearly separable data, suppose there exists some hyperplane, H, which separated the positive from the negative. The points, x which lie on the hyperplane satisfy

$$w \cdot x + b = 0$$

where w is the normal to the hyperplane, and $|b|/||w||$ is the perpendicular distance from the hyperplane to the origin. Two other oriented hyperplanes can also be defined, H_1 and

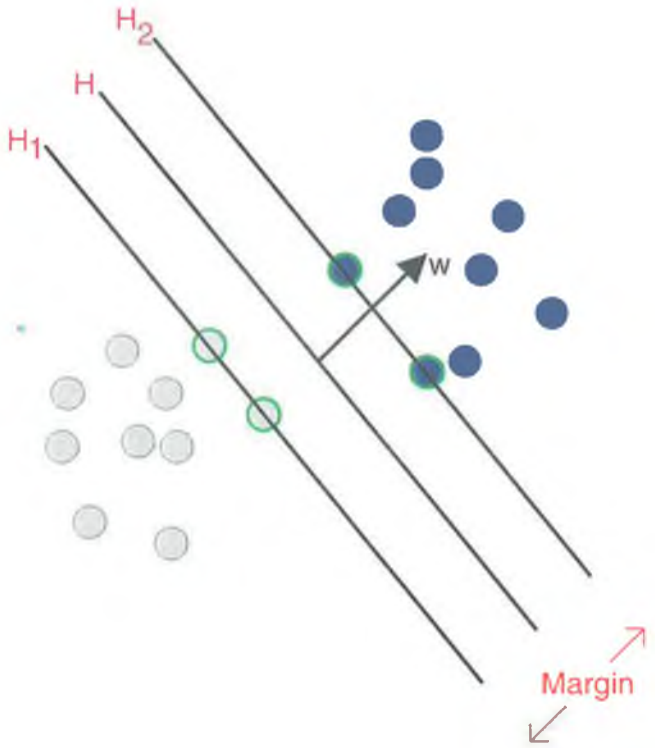

Figure B.3: A set of positive and negative data samples, and a possible set of hyperplanes, H, H_1, and H_2. The points with the green circles are the support vectors.

H_2 which are parallel to H, and lie on the decision boundary (i.e. the data points closest to H as seen in Figure B.3):

$$w \cdot x + b \geq 1, \quad for\ y_i = 1$$

$$w \cdot x + b \leq -1, \quad for\ y_i = -1$$

By combining the above two equations,

$$y_i(w \cdot x + b) - 1 \geq 0, \quad for\ all\ i$$

The value $y_i(w \cdot x + b) - 1 = 0$ holds for all training points that lie on H_1 or H_2. These training points are highlighted with green circles in Figure B.3. These values are called *support vectors*, since they have the most direct influence on the positioning of the hyperplane. The distance between the two hyperplanes H_1 and H_2 is called the *margin*. SVMs try and maximise this margin to give the best possible generalisation. It can be shown [71] that maximising the margin, will minimise the VC dimension. Thus a separating hyperplane that gives the largest margin (i.e. maximum distance between H_1 and H_2) will minimise the risk, and therefore give improved generalisation. SVMs attempt to find the hyperplane that gives the maximum margin.

B.4 SVM Training

The problem of maximising the width of the margin $(2/||w||)$ (which involves minimising the value $\frac{||w||^2}{2}$), subject to the constraints $y_i(w \cdot x + b) - 1 \geq 0$ can be solved using Lagrange multipliers. The procedure of solving these equations is not presented here, however interested readers are pointed to [71] for a more complete solution. Essentially, a training point for which the Lagrange multiplier, α_i is greater than zero is labelled as a support vector, as they are the training points that lie closest to the decision boundary on either H_1, or H_2. All other points have $\alpha_i = 0$ and have no effect on the solution. This feature of SVMs is known as *sparseness* since the final solution is dictated by a subset of the training data. It means that points that are far away from the hyperplane do not have any effect on the resultant SVM [58].

In order to assess the solution the hyperplane a set of conditions known as the *Karush-Kuhn-Tucker (KKT) conditions* must be met. These are satisfied at the solution to any optimisation problem in situations where the constraints are linear. For a linear SVM there are five such conditions. If all of these conditions hold then there is necessary and sufficient proof that w, b, and α are the solution to the SVM training problem [71].

If the data is non-separable, i.e. there is noise, there may not be a hyperplane that is capable of separating the data correctly into two classes. Noise like this is quite common in real applications. For example, if trying to classify audio data into speech and non-speech, a sample of music (non-speech) may have speech like characteristics. SVMs deal with non-separable data using *slack variables*. Slack variables relax the set of constraints $y_i(w \cdot x + b) - 1 \geq 1$, when necessary, i.e. allow some points to be classified incorrectly in training. The new set of constraints are:

$$w \cdot x + b \geq 1 - \xi_i, \quad for\ all\ y_i = 1$$

$$w \cdot x + b \geq -1 - \xi_i, \quad for\ all\ y_i = -1$$

$$\xi_i \geq 0$$

Combining as before:

$$y_i(w \cdot x + b) \geq 1 - \xi_i$$

Where ξ_i is the slack variable, and can be seen as a measure of how much a point violates a constraint. If ξ_i is between 0 and 1, then the point will be classified correctly, however it will be inside the margin (i.e. between H_1 and H_2). If ξ_i is greater than 1, the data point will be misclassified. Therefore, $\sum \xi_i$ is the upper bound of the total error of the SVM. In order to train the SVM, again by maximising the margin, it is required to minimise the function $\frac{||w||^2}{2} + C(\sum_i \xi_i)^k$, where C is a user chosen error penalty, and the value for k is usually 1. The C value essentially assigns the penalty given to errors, the larger C

the harsher the penalty. Similarly to the separable case, it is possible to use Lagrange multipliers to find the optimal hyperplane, that satisfy a set of KKT conditions.

In order to test new data (both separable and non separable), all that is required is to see which side of the hyperplane a given test pattern x lies. If it is on the positive part of the hyperplane, then it is labelled as belonging to that class and vice versa. This can be evaluated by $f(x) = sgn(w \cdot x + b)$

B.5 Non-Linear Data

In all of the examples given so far, the target function can be expressed as a linear function of the data. This may not always be possible, as in many cases the target function is non-linear. A non-linear target function will occur if a set of input data cannot be linearly separated. In this case the techniques described in the previous sections cannot apply. In order to find a suitable boundary, SVMs map the input data to a higher dimensional feature space in which the data *can* be separated by a linear function. This involves mapping the set of input vectors (x,y) to a higher dimensional space. One possible mapping could be:

$$\Phi(x, y) = (x, x^2, xy, y^2)$$

where Φ represents the mapping. This mapping, from a 2-D input space, to a 4-D feature space is based on the features of the input vector, and could make the linear separation of the data possible. In order to map the data to a high dimensional feature space, without explicitly creating the feature space, SVMs use a set of functions, known as *kernel functions*. These kernel functions are inserted into the training and test algorithms, to generate linearly separable data in some higher dimensional feature space. They have the additional advantage of having low computational complexity, as the only extra computation is that of computing the kernel function [58]. These kernel functions are chosen by the user when designing the SVM. Two examples of kernel functions are:

$$K(x, y) = s(x \cdot y + c)^p$$

which is a kernel function that results in a classifier that is a polynomial of degree p of the data. Also,

$$K(x, y) = e^{-||x-y||^2/2\sigma^2}$$

gives a Gaussian radial basis function classifier. By using these kernel functions, the same testing techniques used in linear data can be applied.

In general, SVMs are a useful tool for classifying data sets. Given an accurate training set, and a reasonable selection of the error penalty (C) value, and kernel function

$(K(x, y))$, SVMs are versatile classifiers that have many uses.

APPENDIX C

Hidden Markov Models

C.1 Solution to the Evaluation Problem

[70] states that one way of solving this problem is to exhaustively search through every set of possible states, i.e. find every possible future state from the current state, and every possible future state of the future states! However this requires excessive computation. A more efficient way of solving this problem is using the *forward algorithm.* This involves evaluating the probability of the observation sequence recursively. It works under the principle the probability of any state occurring only depends on the previous states, and the A and B matrices. The probability that the HMM is in state S_i at time t, given the observation sequence O is $\alpha_t(i)$ (i.e. the probability that the model is in state S_i at time t and has generated the sequence up to time t). There are three steps in the forward algorithm, *Initialisation, Induction* and *Termination.* Initialisation involves finding the probabilities of the first state using the initial probabilities and the observation probabilities, i.e.

$$\alpha_1(i) = \pi_i b_i(O_i), i \leq i \leq N$$

The induction stage then begins, which calculates the probability of each hidden state from each of the previous hidden states. i.e.

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1$$
$$1 \leq j \leq N.$$

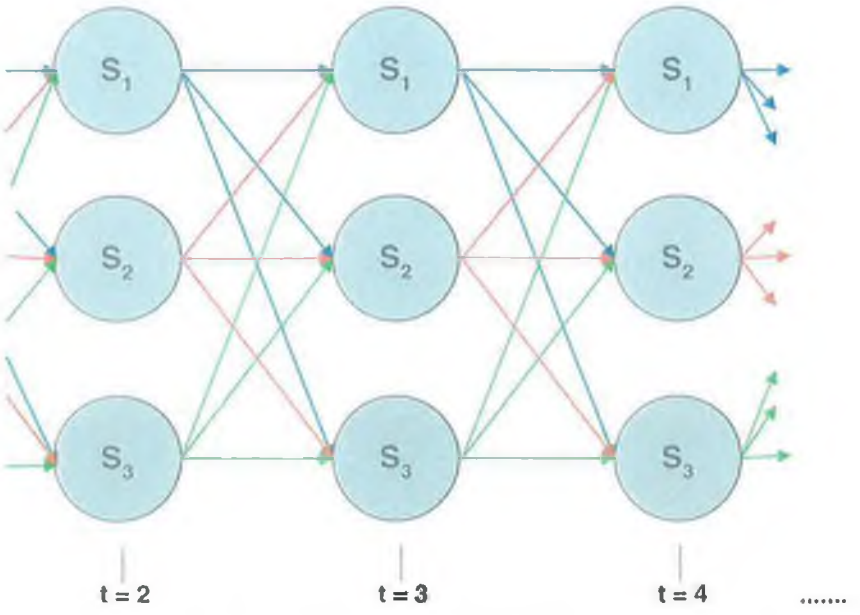

Figure C.1: The forward algorithm in the induction stage.

This stage of the algorithm essentially finds the probability of being in a particular state given that the probabilities of previous states are known. As there are only N states, the computation is significantly decreased. This is shown in Figure C.1. In this picture the α_{ij} lines are shown, i.e. the blue line connecting S_1 to state S_2 is α_{12}. The probability of being in state two at time $t = 3$ is the sum of the probabilities of coming from state one, two and three ($\alpha_3(2)$). The induction stage continues until the end of the observation sequence, where the algorithm terminates, and evaluates the sum of the probabilities of the final states, i.e.:

$$P(O|\lambda) = \sum_{i=0}^{N} \alpha_T(i)$$

or the sum of the probability of being in S_1, S_2, and S_3, at the final time, T. This is shown in Figure C.2. This sum of final probabilities is the probability that the HMM produces the given observation sequence. If there are multiple HMMs, this is used to find the most likely HMM to match the data.

C.2 Hidden Markov Model Example

The theory of hidden Markov models is presented in section 4.3.1. A brief example, adapted from [69] is presented here. In this example, a person in a locked room is attempting to predict the outside weather (the hidden states) based on a sample of seaweed (the observable state). The weather will be classed as either sunny, cloudy, or rainy, while

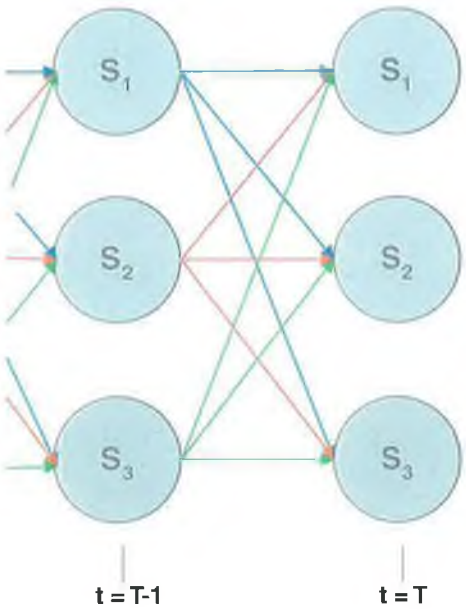

Figure C.2: The forward algorithm in the termination stage.

the seaweed is either dry, damp, or wet. In order to generate a model, three sets of probabilities are required. The initial probabilities, $\Pi = (\pi_i)$, the probability of going from one hidden state to another $A = (a_{ij})$ and the probability of a hidden state occurring given an observation, $B = (b_{ij})$. For simplicity, it is assumed that the probability of being in any hidden state at the beginning of a set of observations is equal, i.e.

$$\Pi = \begin{bmatrix} 0.33 & 0.33 & 0.33 \end{bmatrix}$$

Now, also assume that in the locked room, there is a book of records that shows previous weather. Using these records it is possible to estimate the A matrix. It is possible to look at the weather for all days, and count up how many times the weather goes from one hidden state to another or remains the same the following day. Thus, the probabilities of transition can be generated. The resultant A matrix may look like:

$$A = \begin{array}{c|ccc} & \textit{Sunny} & \textit{Cloudy} & \textit{Rainy} \\ \hline \textit{Sunny} & 0.6 & 0.3 & 0.1 \\ \textit{Cloudy} & 0.3 & 0.4 & 0.3 \\ \textit{Rainy} & 0.1 & 0.3 & 0.6 \end{array}$$

Where the probability if being sunny on day t, and rainy on day $t + 1$ is 0.1. Also in the locked room, by luck, a measure of seaweed wetness for each day classed into dry, damp and wet is also available in this book of records. Using these records it is possible to generate a set of probabilities for a hidden state occurring, given an observation state i.e.

given that the seaweed is wet today, what is the probability of it being sunny/cloudy/rainy outside. This is known as the observation symbol probability distribution, B (sometimes known as a confusion matrix). This may look something like:

$$B = \begin{array}{c|ccc} & Wet & Damp & Dry \\ Sunny & 0.1 & 0.2 & 0.7 \\ Cloudy & 0.3 & 0.4 & 0.3 \\ Rainy & 0.7 & 0.2 & 0.1 \end{array}$$

So, for example, if the hidden state is sunny, then the probability of dry seaweed being passed through the door is 0.7. These three sets of probabilities make up the HMM. This HMM can then be used to take a new piece of seaweed and make a prediction about the weather outside using the Viterbi algorithm.

APPENDIX D

Finite State Machines for Potential Sequence Generation

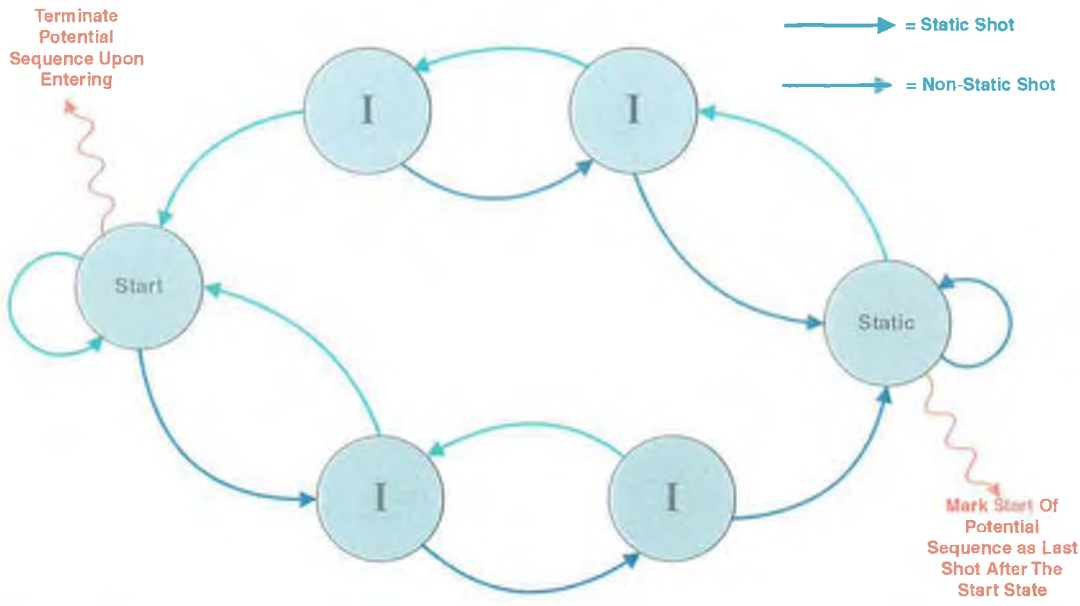

Figure D.1: FSM for detecting potential sequences where static camera shots dominate.

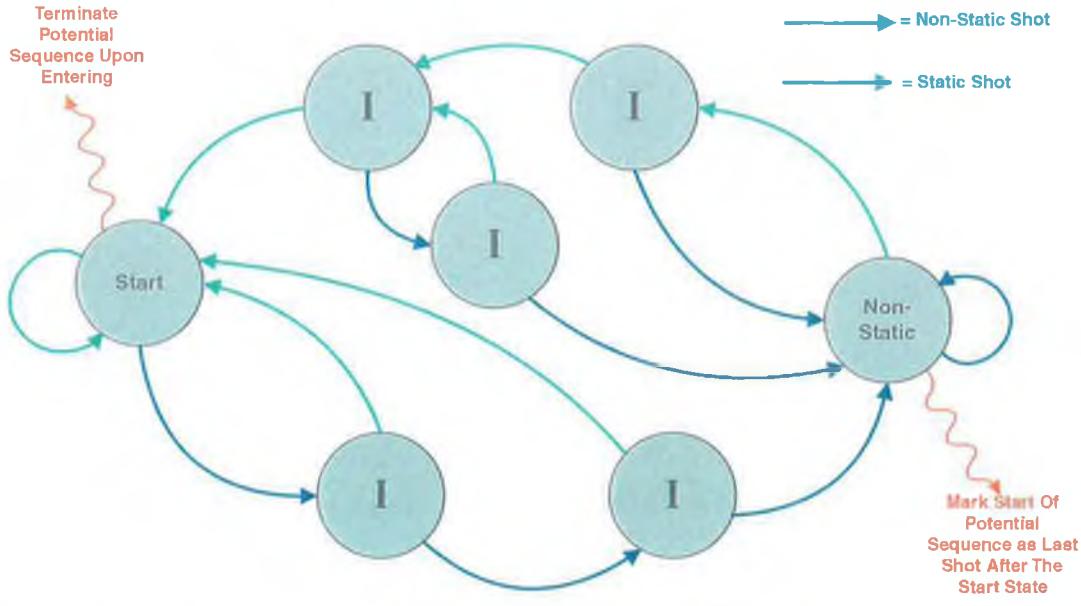

Figure D.2: FSM for detecting potential sequences where non-static camera shots dominate.

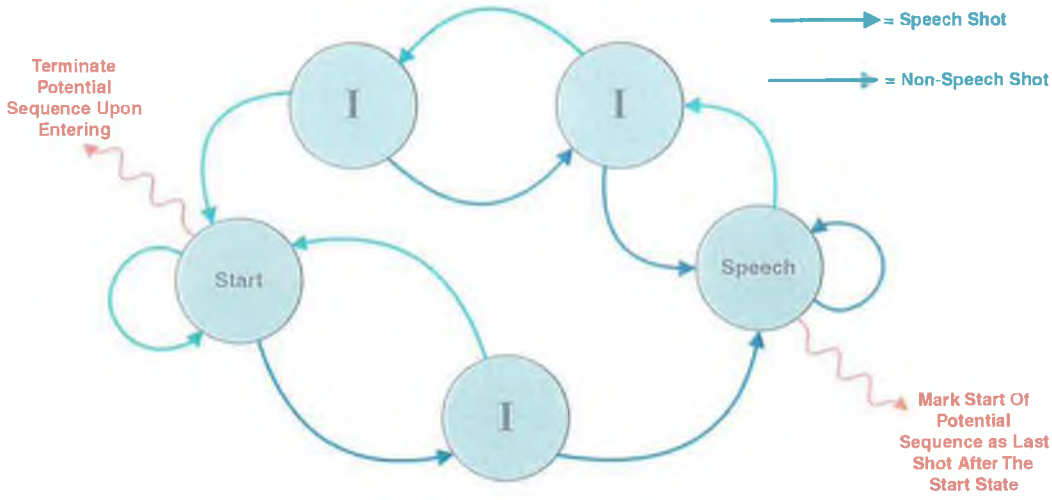

Figure D.3: FSM for detecting potential sequences where speech shots dominate.

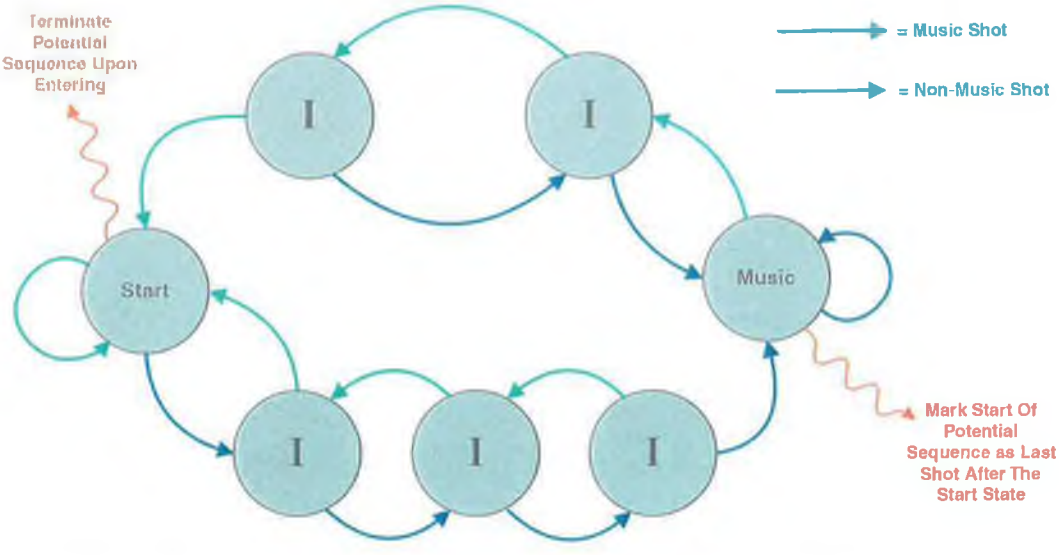

Figure D.4: FSM for detecting potential sequences where music shots dominate.

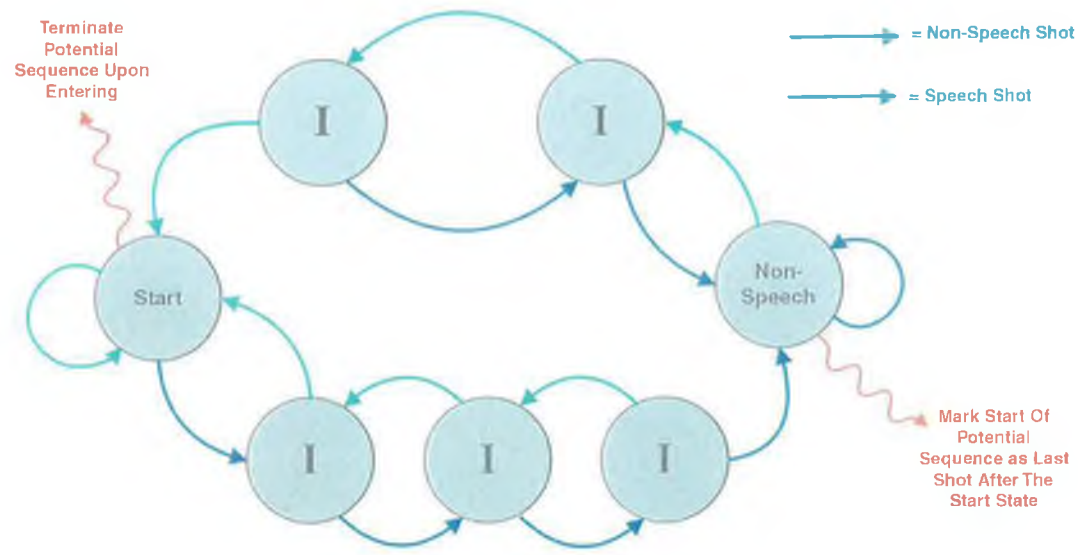

Figure D.5: FSM for detecting potential sequences where non-speech shots dominate.

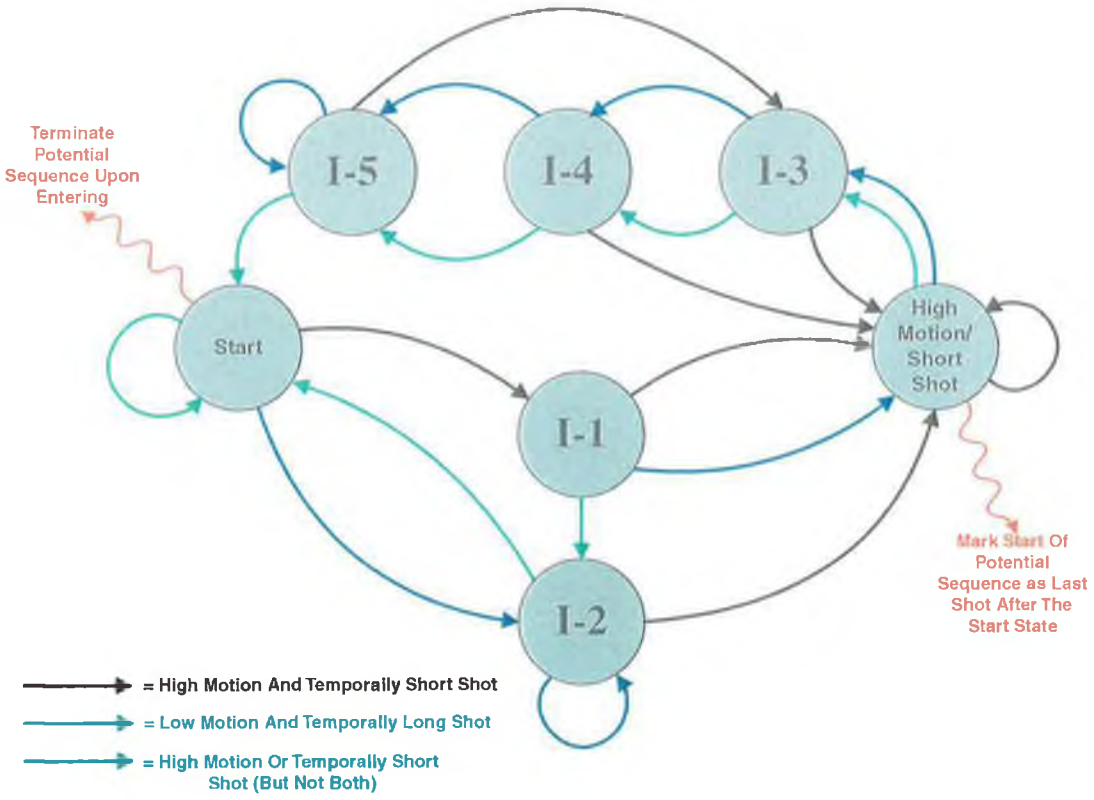

Figure D.6: FSM for detecting potential sequences where high motion/temporally short shots dominate.

APPENDIX E

Task List for User Experiments

- Task 1 = In the film High Fidelity, Find the part where Barry sings 'Lets get it on' with his band.
- Task 2 = In the film American Beauty, Find the conversation between Carolyn (Annette Benning) and 'The Real Estate King/Buddy' in the restaurant, where Buddy tells Carolyn he and his wife are splitting up because he is too driven.
- Task 3 = In the film High Fidelity, Find the part where Rob sees the skateboarders stealing records and chases them out of the store.
- Task 4 = In the film Reservoir Dogs, Find the getaway scene when Mr. Pink is running away from the cops.
- Task 5 = In the film Snatch, Find the first fight in a boxing ring involving Mickey (Brad Pitt), where he knocks the opponent (Bomber Harris) out with one punch.
- Task 6 = In the film Shrek, Find the part where Donkey is being chased by the guards after his owner tries to sell him.
- Task 7 = In the film Chopper, Find the part where Chopper stabs Keithy George in the face.
- Task 8 = In the film, Dumb and Dumber Find the karate fight between Lloyd and the chef.
- Task 9 = In the film American Beauty, Find the dream sequence where Lester is thinking about Angela Hayes in a bath of roses.

- Task 10 = In the film High Fidelity, Find the part where Rob fights Ray after he comes into the store.
- Task 11 = In the film Reservoir Dogs, Find the conversation between Mr. White and Joe, where they are discussing Alabama, and then the heist.
- Task 12 = In the film Snatch, Find the second boxing match involving Mickey, where they are fighting in a boxing ring in bare knuckles.
- Task 13 = In the film Shrek, Find the fight where Shrek and Donkey take on the prince's soldiers, when they first walk into the palace.
- Task 14 = In the film Chopper, Find the conversation where Chopper is talking to the police about the death of Keithy George.
- Task 15 = In the film Dumb and Dumber, Find the snow fight between Harry and Mary (It occurs after they build a snowman together).
- Task 16 = In the film American Beauty, Find the part where the two neighbours (Jim and Jim) deliver a welcome basket to the father (Frank Fitz) of the new family in the neighbourhood.
- Task 17 = In the film High Fidelity, Find the conversation between Rob and Bruce Springsteen, where Bruce is playing guitar.
- Task 18 = In the film Reservoir Dogs, Find the sequence where the gang are walking in slow motion with music in the background.
- Task 19 = In the film Snatch, Find the part where the disguised gang commit armed robbery on the Jewish jewellers.
- Task 20 = In the film Shrek, Find the conversation between the ginger bread man and Prince Farquad. Where the ginger bread man calls the Prince a 'Monster' and then tells him about the 'Muffin Man'.
- Task 21 = In the film Snatch, Find the emotional sequence where Mickey sees his mothers caravan burning whilst she is in it.
- Task 22 = In the film Dumb and Dumber, Find the part where Harry gets shot by Nicholas, but then jumps up and shoots back.
- Task 23 = In the film American Beauty, Find shots of the basketball match before the dance recital.

- Task 24 = In the film High Fidelity. Find the conversation between Rob and Maria De Salle the morning after sleeping together, where they talk on the couch, and then say goodbye outside her apartment
- Task 25 = In the film Reservoir Dogs, Find the part where the whole gang are in the warehouse. Joe is lecturing the gang about joking too much, and then he gives them their code names
- Task 26 = In the film Snatch. Find the conversation where Tommy is buying the gun from Boris the blade.
- Task 27 = In the film Shrek, Find the first meeting of Shrek and Prince Farquad. It occurs in Prince Farquad's palace
- Task 28 = In the film Chopper. Find the court case where Jimmy Loughnan interrogates Chopper.
- Task 29 = In the film Dumb and Dumber, Find where the song 'Pretty Woman' is playing (hint: Harry and Lloyd are getting ready for a ball
- Task 30 = In the film American Beauty, Find the sequence where Lester is singing along to the song 'American Woman' in his car (You can recognise the song title from the lyrics)

Bibliography

[1] N. O'Connor and M. Murphy. *EE554 Lecture Notes on Image and Video Compression*. 2000.

[2] Mohammed Ghanbari. *Standard Codecs: Image Compression to Advanced Video Coding*. The Institution of Electrical Engineers, 2003.

[3] John S. Boreczky and Lawrence A. Rowe. Storage and retrieval for image and video databases (spie). In *Comparison of Video Shots Boundary Detection Techniques*, 1996.

[4] Paul Browne, Alan Smeaton, N. Murphy, N. O'Connor, S. Marlow, and C. Berrut. Evaluating and combining digital video shot boundary detection algorithms. In *Irish Machine Vision and Image Processing Conference*, 2002.

[5] Alan Smeaton, W Kraaij, and P. Over. The TREC video retrieval evaluation (TRECVID): A case study and status report. In *Riao 2004*, 2004.

[6] Christian Petersohn. Franhoffer HHI at TRECVID 2004: Shot boundary detection system. In *Proceedings of TRECVID 2004*, 2004.

[7] H.E. Williams T. Volkmer, S.M.M. Tahaghoghi. RMIT university at TRECVID 2004. In *Proceedings of TRECVID 2004*, 2004.

[8] J. Vermaak, P. Perex, M. Ganget, and A. Blake. Rapid summarisation and browsing of video sequences. In *Proceedings of the British Machine Vision Conference*, 2002.

[9] Matthey Cooper and Jonathan Foote. Discriminative techniques for keyframe selection. In *IEEE International Conference on Multimedia and Expo*, 2005.

[10] Andreas Girgensohn, John Boreczky, and Lynn Wolcox. Keyframe-based user interfaces for digital video. In *IEEE Computer*, volume 34(9), pages 61–67, September, 2001.

[11] M. Smith and T. Kanade. Video skimming for quick browsing based on audio and image characterization. In *Tech Report, School of Computer Science, Carnegie Mellon Univ.*, 1995.

[12] Michael G. Christel, Michael A. Smith, C. Roy Taylor, and David B. Winkler. Evolving video skims into useful multimedia abstractions. In *Proceedings of the ACM CHI'98 Conference on Human Factors in Computing Systems*, 1998.

[13] V. DiLecce, G. Dimauro, A. Guerriero, S. Impedovo, G. Pirlo, and A. Salzo. Image basic features indexing techniques for video skimming. In *Image Analysis and Processing*, 1999.

[14] M. Yeung and B.-L. Yeo. Time constrained clustering for segmentation of video into story units. In *Proceedings of International Conference on Pattern Recognition*, 1996.

[15] M. Yeung and B.-L. Yeo. Video visualisation for compact presentation and fast browsing of pictorial content. In *IEEE Transactions on Circuits and Systems for Video Technology*, pages 771–785, 1997.

[16] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Constructing table-of-content for video. In *ACM Journal of Multimedia Systema*, pages 359–368, 1998.

[17] John R. Kender and Book-Lock Yeo. Video scene segmentation via continuous video coherence. In *Proceedings CVPR*, pages 167–393, 1998.

[18] Zeeshan Rasheed and Mubarak Shah. Scene detection in hollywood movies and tv shows. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[19] Junyu Zhou and Wallapak Tavanapong. Shotweave: A shot clustering technique for story browsing for large video databases. In *Proceedings of the Workshops XMLDM, MDDE, and YRWS on XML-Based Date Management and Multimedia Engineering-Revised Papers*, 2002.

[20] Ba Tu Truong, Svetha Venkatesh, and Dorai Chitra. Scene extraction in motion pictures. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 13, pages 5–15, January 2003.

[21] Tiecheng Liu and John R. Kender. A hidden markov approach to the structure of documentaries. In *Proceedings of the IEEE Workshop on Content-Based Access on Image and Video Libraries*, 2000.

[22] Hari Sundaram and Shih-Fu Chan. Determining computable scenes in films and their structures using audio-visual memory models. In *ACM Multimedia 2000*, 2000.

[23] Yu Cao, Wallapak Tavanapong, Kihwan Kim, and JungHwan Oh. Audio-assisted scene segmentation for story browsing. In *Proceedings of the International Conference on Image and Video Retrieval*, 2003.

[24] Jincheng Huang, Zhu Liu, and Yeo Wang. Integration of audio and visual information for content-based video segmentation. In *IEEE Int'l Conf. Image Processing*, 1998.

[25] Ying Li and C.-C. Jay Kou. *Video Content Analysis using Multimodal Information.* Kluwer Academic Publishers, 2003.

[26] Stanley Boykin and Andrew Merlino. Machine learning of event segmentation for news on demand. In *Communications of the ACM*, February, 2000.

[27] N. O'Hare, A. Smeaton, C. Czirjek, N. O'Connor, and N. Murphy. A generic news story segmentation system and its evaluation. In *IEEE International Conference on Acoustics, Speech, and signal Processing (ICASSP) 2004, Montreal, Quebec*, 2004.

[28] Georges M. Quenot, Daniel Moraru, Stephane Ayache, Mbarek Charad, Mickael Guironnet, Lionel Carminati, Jerome Gensel, Dennis Pullerin, and Laurent Besacier. Clips-lis-lsr-labri experiments at TRECVID 2004. In *TREC Video Retrieval Conference*, 2004.

[29] Keiichiro Hoashi, Masaru Sugano, Kazunori Matsumoto, Fumiaki Sugaya, and Yasuyuki Nakajimi. Shot boundary determination on mpeg compressed domain and story segmentation experiments for trecvid 2004. In *TREC Video Retrieval Conference*, 2004.

[30] Wessel Kraaj and Joaquim Arlandis. Trecvid-2004 story segmentation task: Overview. In *TREC Video Retrieval Conference*, 2004.

[31] J. Assfalg, M. Bertini, A. Del Bimbo, W. Nunziati, and P. Pala. Soccer highlights detection and recognition using hmm's. In *Proc. IEEE ICME 2002*, 2002.

[32] D. Yow, B.L. Yeo, M. Yeung, and B. Liu. Analysis and presentation of soccer highlights from digital video. In *Proc. Asian Conference on Computer Vision 1995*, 1995.

[33] R. Cabasson and A. Divakaran. Automatic extraction of soccer video highlights using a combination of motion and audio features. In *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Media Databases*, volume 5021, pages 272–276, Jan 2002.

[34] D.D. Saur, Y-P. Tan, S.R. Kulkarni, and P.J. Ramadge. Automated analysis and annotation of basketball video. In *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Image and Video Databases*, volume 3022, pages 176–187, Jan 1997.

[35] S. Nepal, U. Srinivasan, and G. Reynolds. Automatic detection of goal segments in basketball videos. In *Proc. of ACM Multimedia*, pages 261–269, 2001.

[36] M. Lazarescu, S. Venkatesh, and G West. On the automatic indexing of cricket using camera motion parameters. In *Proc IEEE ICME 2002*, 2002.

[37] E. Kijak, L. Oisel, and P. Gros. Temporal structure analysis of broadcast tennis video using hidden markov models. In *Electronic Imaging: Science and Technology: Storage and Retrieval for Media Databases*, volume 5021, pages 277–288, 2003.

[38] Li and M.I. Sezan. Event detection and summarization in american football broadcast video. In *ESymp. Electronic Imaging: Science and Technology: Storage and Retrieval for Media Databases*, volume 4676, pages 202–213, 2002.

[39] M. Petkovic, V. Mihajlovic, M. Jonker, and S. Djordjevic-Kajan. Multi-modal extraction of highlights from TV formula 1 programs. In *Proc. IEEE ICME*, 2002.

[40] P. Chang, M. Han, and Y. Gong. Extract highlights from baseball game video with hidden Markov models. In *Proc. IEEE Int. Conf. on Image Proc. (ICIP)*, 2002.

[41] B. Li and M.I. Sezan. Event detection and summarization in sports video. In *IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL)*, 2001.

[42] D. Sadlier and N. O'Connor. Event detection in field sports video using audio-visual features and a support vector machine. In *IEEE Transitions on Circuits Systems and Video Technology*, July, 2005.

[43] R. Leinhart, S. Pfeiffer, and W. Effelsberg. Scene determination based on video and audio features. In *In proceedings of IEEE Conference on Multimedia Computing and Systems*, pages 685–690, 1999.

[44] Ying Li and C.-C. Jay Kou. Movie event detection by using audiovisual information. In *Proceedings of the Second IEEE Pacific Rim Conferences on Multimedia: Advances in Multimedia Information Processing*, 2001.

[45] L. Chen, Shariq J. Rizvi, and M.T. Ötzu. Incorporating audio cues into dialog and action scene detection. In *Proceedings of SPIE Conference on Storage and Retrieval for Media Databases*, pages 252–264, 2003.

[46] A. Aydin Alatan, Ali N. Akansu, and Wayne Wolf. Multi-modal dialogue scene detection using hidden Markov models for content-based multimedia indexing. In *Multimedia Tools and Applications*, volume 14, pages 137–151, 2001.

[47] Jeho Nam, Masoud Alghoniemy, and Ahmed H. Tewfik. Audio-visual content-based violent scene characterization. In *Proceedings of International Conference on Image Processing (ICIP)*, volume 1, pages 351–357, 1998.

[48] Hang-Bong Kang. Emotional event detection using relevance feedback. In *Proceedings of the International Conference on Image Processing*, 2003.

[49] Yun Zhai, Zeeshan Rasheed, and Mubarak Shah. A framework for semantic classification of scenes using finite state machines. In *International Converence on Image and Video Retrieval*, 2004.

[50] www.infoplease.com.

[51] David Bordwell and Kristen Thompson. *Film Art: An Introduction*. McGraw-Hill, 1997.

[52] Ken Dancyger. *The Technique of Film and Video Editing. History, Theory and Practice*. Focal Press, 2002.

[53] Roy Thompson. *Grammar of the Edit*. Focal Press, 1993.

[54] S. Sav, N. O'Connor, A.F. Smeaton, and N. Murphy. Associating low-level features with semantic concepts using video objects and relevance feedback. In *6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAIMIS 2005)*, 2005.

[55] Y. Xu, E Saber, and A.M. Teklap. Object segmentation and labeling by learning from examples. In *IEEE Transactions on Image Processing*, volume 12, pages 627–638, June 2003.

[56] A. Salway, A. Vassiliou, and K. Ahmad. What happens in films? In *International Conference on Multimedia and Expo (ICME)*, 2005.

[57] A. Salway and M. Graham. Extracting information about emotions in films. In *11th ACM Conference on Multimedia*, pages 299–302, 2003.

[58] Neil K. O'Hare. Using support vector machines to segment digital video. Master's thesis, School of Computer Applications, Dublin City University, 2003.

[59] R. Jarina, N. Murphy, N. O'Connor, and S. Marlow. Speech/music discrimination from mpeg-1 bitstream. In *WSES - International Conference on Speech, Signal, and Image Processing, Malta*, 2001.

[60] L. Lu, Stan Z. Li, and Hong-Jiang Zhang. Content based audio segmentation using support vector machines. In *Proc. ICME, Tokyo, Japan, 2001*, pages 956–959, 2001.

[61] B.-L. Yeo and B. Liu. Rapid scene analysis on compressed videos. In *IEEE Transactions on Circuits and Systems for Video Technology*, pages 533–544, 1995.

[62] B.S. Manjunath, Philippe Salember, and Thomas Sikora. *Introduction to MPEG-7, Multimedia content description language*. John Wiley and Sons ltd, 2002.

[63] S. Jeannin and B Mory. Video motion representation for improved content access. In *IEEE Transactions on Consumer Electronics*, volume 46(3), pages 645–655, 2000.

[64] A. Akutsu, Y. Tonomura, H. Hashimoto, and Y. Ohba. Video indexing using motion vectors. In *SPIE Visual Communications and Image Processing*, pages 1522–1530, 1992.

[65] J. Park, N. Yago, K. Enami, and E. Kiyoharu. Estimation of camera motion parameters from image sequence for model-based video coding. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 4(3), pages 593–606, 1994.

[66] G. Lu and T. Hankinson. A technique towards automatic audio classification and retrieval. In *4th International Conference on Signal Processing, Beijing*, 1998.

[67] Gethin Williams and Daniel P.W. Ellis. Speech/music discrimination based on posterior probability features. In *Eurospeech, Budapest*, 1999.

[68] Richard O. Duda, Peter E. Hart, and Stork David G. *Pattern Classification.* John Wiley and Sons, 2001.

[69] www.comp.leeds.ac.uk/roger/HiddenMarkovModels/htmldev/main.html.

[70] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE, VOL. 77, NO. 2, 257-286*, 1989.

[71] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. In *Data Mining and Knowledge Discovery*, volume 2, pages 121–167, 1988.