Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2007, Article ID 14615, 15 pages
doi:10.1155/2007/14615

Research Article

Indexing of Fictional Video Content for Event Detection and Summarisation

Bart Lehane,[1] Noel E. O'Connor,[2] Hyowon Lee,[1] and Alan F. Smeaton[2]

[1] *Centre for Digital Video Processing, Dublin City University, Dublin 9, Ireland*
[2] *Adaptive Information Cluster, Dublin City University, Dublin 9, Ireland*

Received 30 September 2006; Revised 22 May 2007; Accepted 2 August 2007

Recommended by Bernard Mérialdo

This paper presents an approach to movie video indexing that utilises audiovisual analysis to detect important and meaningful temporal video segments, that we term *events*. We consider three event classes, corresponding to dialogues, action sequences, and montages, where the latter also includes musical sequences. These three event classes are intuitive for a viewer to understand and recognise whilst accounting for over 90% of the content of most movies. To detect events we leverage traditional filmmaking principles and map these to a set of computable low-level audiovisual features. Finite state machines (FSMs) are used to detect when temporal sequences of specific features occur. A set of heuristics, again inspired by filmmaking conventions, are then applied to the output of multiple FSMs to detect the required events. A movie search system, named *MovieBrowser*, built upon this approach is also described. The overall approach is evaluated against a ground truth of over twenty-three hours of movie content drawn from various genres and consistently obtains high precision and recall for all event classes. A user experiment designed to evaluate the usefulness of an event-based structure for both searching and browsing movie archives is also described and the results indicate the usefulness of the proposed approach.

1. INTRODUCTION

Virtually, all produced video content is now available in digital format, whether directly filmed using digital equipment, or transmitted and stored digitally (e.g., via digital television). This trend means that the creation of video is easier and cheaper than ever before. 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 results in an increase in the number of independent films produced. Also, editing equipment is now compatible with home computers which makes cheap postproduction possible.

Unfortunately, the vast majority of this content is stored without any sort of content-based indexing or analysis and without any associated metadata. If any of the videos have metadata, then this is due to manual annotation rather than an automatic indexing process. 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 a user would significantly ease this process. Fictional video content, particularly movies, is a medium particularly in need of indexing 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 a thirty-minute news program, for example. Most films are at least one and a half hours long, with many as long as three hours. In fact, other forms of fictional content, such as television series (dramas, soap operas, comedies, etc.), may have episodes an hour long, so are also difficult to be managed without indexing.

Indexing 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, and so forth, which results in varying styles. Also, it may take a number of months to shoot a two-hour film. 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 work to very short (sometime real-time) time constraints. The focus of this paper is on summarising fictional video content. At various stages throughout the paper, concepts such as *filmmaking* or *film grammar* are discussed, however each of these factors is equally applicable to creating a television series.

The primary aim of the research reported here is to develop an approach to automatically index movies and fictional television content by examining the underlying structure of the video, and by extracting knowledge based on this structure. By examining the conventions used when fictional video content is created, it is possible to infer meaning as to the activities depicted. Creating a system that takes advantage of the presence of these conventions in order to facilitate retrieval allows for efficient location of relevant portions of a movie or fictional television program. Our approach is designed to make this process completely automatic. The indexing process does not involve any human interaction, and no manual annotation is required. This approach can be applied to any area where a summary of fictional video content is required. For example, an event-based summary of a film and an associated search engine is of significant use to a student studying filmmaking techniques who wishes to quickly gather all dialogues or musical scenes in a particular director's oeuvre to study his/her composition technique. Other applications include generating previews for services such as video-on-demand, movie database websites, or even as additional features on a DVD.

There have been a number of approaches reported that aim to automatically create a browsable index of a movie. These can broadly be split into two groups, those that aim to detect scene breaks and those that aim to detect particular parts of the movie (termed *events* in our work). A scene boundary detection technique is proposed in [2, 3], 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 in order to identify the scene transitions. Scene boundaries are located by examining the structure of the clusters and detecting points where one set of clusters ends and another begins. The concept of shot coherence can also be used in order to find scene boundaries [4, 5]. Instead of clustering similar shots together, the coherence is used as a measure of the similarity of a set of shots with previous shots. 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. Approaches such as [6, 7] define a computable scene as one which exhibits long term consistency of chrominance, lighting, and ambient sound, and use audio-visual detectors to determine when this consistency breaks down. Although scene-based indexes may be useful in certain scenarios, they have the significant drawback that no knowledge about what the content depicts is contained in the index. A user searching for a particular point in the movie must still peruse the whole movie unless significant prior knowledge is available.

Many event-detection techniques in movie analysis focus on detecting individual types of events from the video. Alatan et al. [8] 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 nondialogue. Dialogue events are also detected in [9] based on the common-shot-/reverse-shot-shooting technique, where if repeating shots are detected, a dialogue event is declared. However, this approach is only applicable to dialogues involving two people, since if three or more people are involved the shooting structure will become unpredictable. This general approach is expanded upon in [10, 11] to detect three types of events: 2-person dialogues, multiperson dialogues, and hybrid events (where a hybrid event is everything that is not a dialogue). However, only dialogues are treated as meaningful events and everything else is declared as a hybrid event. The work of [19] aims to detect both dialogue and action events in a movie, but the same approach is used to detect both types of events, and the type of action events that are detected is restricted.

Perhaps the approach most similar to ours is that of [12, 13]. Both approaches are similar in that they extract low-level audio, motion, and colour features, and then utilise finite state machines in order to classify portions of films. In [12], the authors classify clips from a film into three categories, namely conversation, suspense and action as opposed to dialogue, and exciting and montage as in our work. Perhaps the most fundamental difference between the approaches is that they assume the temporal segmentation of the content into scenes as a priori knowledge and focus on classifying these scenes. Whilst many scene boundary approaches exist (e.g., [3–7] mentioned above), obtaining 100% detection accuracy is still difficult, considering the subjective nature of scenes (compared to shots, e.g.). It is not clear how inaccurate scene boundary detection will affect their approach. We, on the other hand, assume no prior knowledge of any temporal structure of the movie. We perform robust shot boundary detection and subsequently classify every shot in the movie into one (or more) of our three event classes. A key tenet of our approach is to argue for another level in the film structure hierarchy below scenes, corresponding to events, where a scene can be made up of a number of events (see Section 2.1). Thus, unlike Zhai, we are not attempting to classify entire scenes, but semantically important subsets of scenes. Another important difference between the two approaches is that we have designed for accommodating the subjective interpretation of viewers in determining what constitutes an event. That is, we facilitate an event being classified into more than one event class simultaneously. This is because flexibility is needed in accommodating the fact that one viewer may deem a heated argument a dialogue, for example, whilst another viewer could deem this an exciting event. Thus, for maximum usability in the resulting search/browse system, the event should be classed as both. This is possible in our system but not in that of Zhai. Our goal is to develop a completely automatic approach for entire movies, or entire TV episodes, that accepts a nonsegmented video as input and *completely* describes the video by detecting all of the relevant events. We believe that this approach leads to a more thorough representation of film content. Building on this representation, we also implement a novel audio-visual-event-based searching system, which we believe to be among the first of its kind.

The rest of this paper is organised as follows: Section 2 examines how fictional video is created, Section 3 describes our overall approach, and based on this approach, two search systems are developed, which are described in Section 4. Section 5 presents a number of experiments carried out to evaluate the systems, while Section 6 draws a number of conclusions and indicates future work.

2. FICTIONAL VIDEO CREATION PRINCIPLES AND THEIR APPLICATION

2.1. *Film structure*

An individual video *frame* is the smallest possible unit in a film and typically occurs at a rate of 24 per second. A *shot* is defined as "one uninterrupted run of the camera to expose a series of frames" [14], or, a sequence of frames shot continuously from a single camera. Conventionally, the next unit in a film's structure is the *scene*, made up of a number of consecutive shots. It is somewhat harder to define a scene as it is a more abstract concept, but is labelled in [14] as "a segment in a narrative film that takes place in one time and space, or that uses crosscutting[1] to show two or more simultaneous actions." However, based on examining the structure of a movie or fictional video, we believe that another structural unit is required. An *event*, as used in this research, is defined as a subdivision of a scene that contains something of interest to a viewer. It is something which progresses the story onward corresponding to 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, and it may not even be possible to deduce that a car chase is taking place from a single shot. Only when viewed in context with the surrounding shots in the event does its meaning becomes apparent. In our definition, 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 1, the movie structure we adopt 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. In this research, an event is considered the optimal unit of the movie to be detected and presented as it contains significant semantic meaning to end-users of a video indexing system.

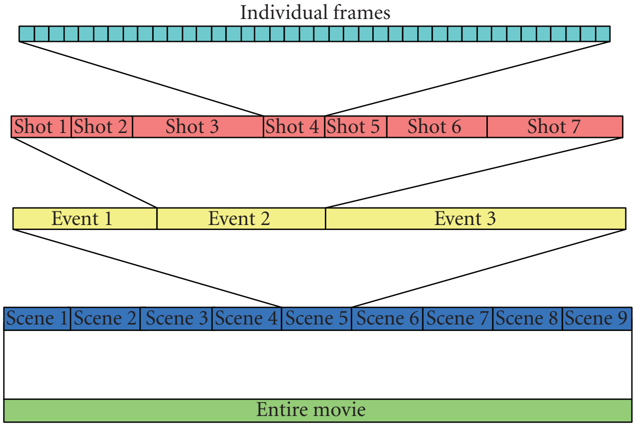


FIGURE 1: Structure of a movie.

2.2. *Fictional video creation principles*

Although movie-making is a creative process, there exists a set of well-defined conventions, that *must* be followed. These conventions were established by early filmmakers, and have evolved and adjusted somewhat since then, but they are so well established that the audience expects them to be followed or else they will become confused. These are not only conventions for the filmmakers, 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, and so forth, built up over previous viewings. These expectations must be met, and can be classed as filmmaking *rules*. Much of our 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 filmmaker and, as a byproduct of this, the activities depicted in the video.

One important rule that dictates the placement of the camera is known as the 180° 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 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° line is set up. This is an imaginary line that joins 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, that is, above the 180° 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° line is immediately set up). This ensures that the characters are facing the same way throughout the scene, that is, character X is looking right to

[1] Crosscutting occurs when two related activities are taking place and both are shown either in a split screen fashion or by alternating shots between the two locations.

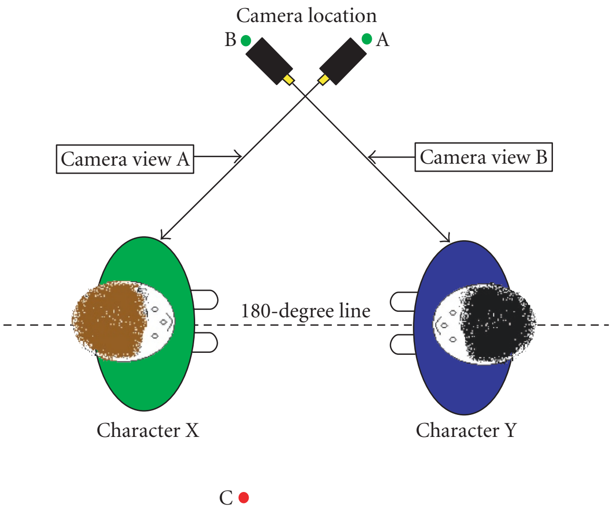


Figure 2: Example of 180° line rule.

left, and character Y is looking left to right (note that this includes shots of characters X and Y together). If, for example, the director decided to shoot character Y from position C in Figure 2, 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° line rule.

The 180° rule allows the audience to comfortably and naturally view an event involving interaction between characters. It is important that viewers are relaxed whilst watching a dialogue in order to fully comprehend the conversation. As well as not confusing viewers, the 180° 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 periphery information shown to a viewer, the more they can concentrate on the words being spoken. Knowledge about camera placement (and specifically the 180° line rule) can be used to infer which shots belong together in an event. Repeating shots, again due to the 180° 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 a new event (in a different location) has begun.

The use of camera movement can also indicate the intentions of the filmmaker. 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, the amount and type of motion present is an important factor in analysing video.

Editing pace is another very important aspect of filmmaking. 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 quickens to tell the viewers that something of import is happening. Pace is usually quite fast during action sequences and 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, thus shot lengths can be used as an indication of a filmmakers intent.

The audio track is an essential tool in creating emotion and setting tone throughout a movie and is a key means of conveying information to the viewer. Sound in films can be grouped into three categories, *Speech, Music,* and *Sound effects*. Usually speech is given priority over the other forms of sound as this is deemed to give the most information and thus not have to compete for the viewer's attention. If there are sound effects or music present 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. Where speech is present, and is important to the viewer, it should be clearly audible. 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 [14]. This categorisation of the sounds in movies is quite important in our research. In our approach, the presence of speech is used as a reliable indicator not only that there is a person talking on-screen, but also that person's speech warrants the audience's attention. Similarly, the presence of music and/or silence indicates that some sort of musical, or emotional, event is taking place.

It is proposed that by detecting the presence of filmmaking techniques, and therefore the intentions of the filmmaker, it is possible to infer meaning about the activities in the video. Thus, the audiovisual features used in our approach (explained in Section 3.2) reflect these film and video making rules.

2.3. *Choice of event classes*

In order to create an event-based index of fictional video content, a number of event classes are required. The event classes should be sufficient to cover all of the meaningful parts in a movie, yet be generic enough so that only a small amount of event classes are required for ease of navigation. Each of the events in an event class should have a common semantic concept. It is proposed here that three classes are sufficient to contain all relevant events that take place in a film or fictional television program. These three classes correspond to *dialogue, exciting,* and *montage*.

Dialogue constitutes a major part of any film, and the viewer usually gets the most information about the plot, story, background, and so forth, 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 number 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.

Exciting events typically occur less frequently than dialogue events, but are central to many movies. Examples of exciting events include fights, car chases, battles, and so forth, Whilst a dialogue event can be clearly defined due to the presence of people talking, an exciting event is far more subjective. Most exciting events are easily declared (a fight, e.g., would be labelled as "exciting" by almost anyone watching), but others are more open to viewer interpretation. Should a heated debate be classed as a dialogue event or an exciting event? As mentioned in Section 2, filmmakers have a set of tools available to create excitement. It can be assumed that if the director wants the viewer 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. Thus, we have 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 subevents that are not explicitly detected but are collected labelled *Montages*. The first type of events in this superset is traditional montage events themselves. A montage 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. 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, and 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 at least a nonspeech audio track. Any future reference to montage events refers to the entire set of events labelled as montages. The three event classes explained above (dialogue, exciting, and montage) aim to cover all meaningful parts of a movie.

3. PROPOSED APPROACH

3.1. Design overview

In order to detect the presence of events, a number of audiovisual features are required. These features are based on the film creation principles outlined in Section 2. The features utilised 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 the amount of motion in the frame (regardless of camera movement), a measure of the editing pace, and a measure of the amount of shot repetition. A method of detecting the boundaries between events is also required. The overall system comprises two stages. The first (detailed in Section 3.2) involves extracting this set of audiovisual features. The second stage (detailed in Section 4) uses these features in order to detect the presence of events.

3.2. Feature extraction

The first step in the analysis involves segmenting the video into individual shots so that each feature is given a single value per shot. In order to detect shot boundaries, a colour-histogram technique, based on the technique proposed in [15], was implemented. In this approach, a 64-bin luminance histogram is extracted for each frame of video and the difference between successive frames is calculated:

$$\text{Diff}_{xy} = \sum_{i=1}^{M} \left| h_x(i) - h_y(i) \right|, \tag{1}$$

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 contains M bins. If the difference between two successive colour histograms is greater than a defined threshold, a shot cut is declared. This threshold was chosen based on a representative sample of video data which contained a number of hard cuts, fades, and dissolves. The threshold which achieved the highest overall results was selected. As fades and dissolves occur over a number of successive frames, this often resulted in a number of successive frames having a high interframe histogram difference, which, in turn, resulted in a number of shot boundaries being declared for one fade/dissolve transition. In order to alleviate this, a postprocessing merging step was implemented. In this step, if a number of shot boundaries were detected in successive frames, only one shot boundary was declared. This was selected at the point of highest interframe difference. This led to significant reduction in the amount of false positives. When tested on a portion of video which contained 378 shots (including fades and dissolves), this method detected shot boundaries with a recall of 97% and a precision of 95%. After shot boundary detection, a single keyframe is selected from each shot by, firstly, computing the values of the average frame in the shot, and then, finding the actual frame which is closest to this average.

The next step involves clustering shots that are filmed using the same camera in the same location. This can be achieved by examining the colour difference between shot keyframes. Shots that have similar colour values and are temporally close together are extremely likely to have been shot from the same camera. Shot clustering has two uses. Firstly it can be used to detect areas where there is shot repetition (e.g., during character interaction), and secondly it can be used to detect boundaries between events. These boundaries occur when the focus of the video (and therefore the clusters) shifts from one location to another, resulting in a clean break between the clusters. The clustering method is based on the technique first proposed in [2], although variants of the algorithm have been used in other approaches since [3, 16]. The algorithm can be described as follows.

(1) Make N clusters, one for each shot.
(2) Find the most similar pair of clusters, R and S, within a specified time constraint.
(3) Stop when the histogram difference between R and S is greater than a predefined threshold.
(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. 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. The clustering information can be used in order to evaluate the amount of shot repetition in a given sequence of shots. The *ratio of clusters to shots (termed CS ratio)* is used for this purpose. The higher the rate of repeating shots, the more shots any given cluster contains and the lower the CS ratio. For example, if there are 20 shots contained in 3 clusters (possibly due to a conversation containing 3 people), the CS ratio is 3/20 = 0.15 [17].

Two motion features are extracted. The first is the *motion intensity*, which aims to find the amount of motion within each frame, and subsequently each shot. This feature is defined by MPEG-7 [18]. The standard deviation of the video-motion vectors is used in order to calculate 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:

$$\overline{x} = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij}, \tag{2}$$

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 (motion intensity) for each frame can then be evaluated as

$$\sigma = \sqrt{\frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(x_{ij} - \overline{x}\right)^2}. \tag{3}$$

The motion intensity for each shot is calculated as the average motion intensity of the frames within that shot. It is then possible to categorise high-/low-motion shots using the scale defined by the MPEG-7 standard [18]. We chose the midpoint of this scale as a threshold, so shots that contain an average standard deviation greater than 3 on this scale are defined as high-motion shots, and others are labelled as low-motion shots.

The second motion feature detects the amount of camera movement in each shot via a novel camera-motion detection method. In this approach, the motion is examined across the entire frame, that is, complete motion vector rows are examined. In a frame with no camera movement, there will be a large number of zero-motion vectors. 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 the number of successive zero-motion vectors. 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 as a run of zero-motion vectors up to 1/3 the width of the frame, a middle run is between 1/3 and 2/3 the width of the frame, and a long run is greater than 2/3 the width of the frame. In order to find the optimal minimum number of runs permitted in a frame before camera movement is declared, a representative sample of 200 P-frames was used. Each frame was manually annotated as being a motion/nonmotion frame. Following this, various values for the minimum amount of runs for a noncamera-motion shot were examined, and the accuracy of each set of values against the manual annotation was calculated. This resulted in a frame with camera motion being defined as a frame that contains 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. When tested, this technique detected whether a shot contained camera movement or not with an accuracy of 85%.

For leveraging the sound track, a set of audio classes are proposed corresponding to *speech, music, quiet music, silence*, and *other*. The music class corresponds to areas where music is the dominant audio type, while quiet music corresponds to areas where music is present, but not the dominant type (such as areas where there is background music). The speech and silence classes contain all areas where that audio type is prominent. The other class corresponds to all other sounds, such as sound effects, and so forth, In total, four audio features are extracted in order to classify the audio track into the above classes. The first is the *high zero crossing rate ratio* (HZCRR). To extract this, for each sample the average zero-crossing rate of the audio signal is found. The high zero crossing rate (HZCR) is defined as 1.5 × the average zero-crossing rate. The HZCRR is the ratio of the amount of values over the HZCR to the amount of values under the HZCR. This feature is very useful in speech classification, as speech commonly contains short silences between spoken words. These silences drive the average down, while the actual speech values will be above the HZCR, resulting in a high HZCRR [10, 19].

The second audio feature is the *silence ratio*. This is a measure of how much silence is present in an audio sample.

The root mean-squared (RMS) value of a one-second clip is first calculated as

$$x_{\text{rms}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \cdots + x_N^2}{N}}, \tag{4}$$

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 and the RMS value of each of these segments is calculated. A silence segment is defined as a segment with an 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, gaps mean that the silence ratio tends to be higher for speech [19].

The third audio feature is the *short-term energy*. In order to generate this, firstly a one-second window is divided into 150 nonoverlapping windows, and the short-term energy is calculated for each window as

$$x_{\text{ste}} = \sum_{i=0}^{N} x_i^2. \tag{5}$$

This provides a convenient representation of the signal's amplitude variations over time [10]. 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.

In order to use these features to recognise specific audio classes, a number of support vector machines (SVMs) are used. Each support vector machine is trained on a specific audio class and each audio sample is assigned to a particular class. The audio class of each shot can then be obtained by finding the dominant audio class of the samples in the shot. Our experiments have shown that, based on a manually annotated sample of 675 shots, the audio classifier labelled the shot in the correct class 90% of the time.

Following audiovisual analysis, each of the extracted features is combined in the form of a feature vector for each shot. Each *shot feature vector* contains [% speech, % music, % silence, % quiet music, % other audio, % static-camera frames per shot, % nonstatic-camera frames per shot, 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. This information can be used in order to detect events and allow searching as described in the following section.

4. INDEXING AND SEARCHING

Two approaches to movie indexing are presented here. The first builds a structured index based on the event classes listed in Section 2.3. This approach is presented in Section 4.2. Building on this, an alternate browsing method is also proposed which allows users to search for specific events in a movie. This is presented in Section 4.3. Both of these approaches are event-based and rely on the same overall approach. Both browsing approaches rely of the detection of segments where particular feature dominate, that we term *potential event sequences.*

4.1. *Sequence detection*

Typically, events in a movie contain consistency of features. For example, if a filmmaker is filming an event which contains excitement, he/she will employ shooting techniques designed to generate excitement, such as fast-paced editing. While fast-paced editing is present, it follows that the excitement is continuing, however, when the fast-paced editing stops, and is replaced by longer shots, then this is a good indication that the exciting event is finished and another event is beginning. The same can be said for all other types of event. Thus, the first step in creating an event-based index for films is to detect sequence of shots which are dominated by the features extracted in Section 3.2, which are representative of the various filmmaking tools. The second step is then to classify these detected sequences.

In order to detect these sequences some data-classification method is required. 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. Finite state machines (FSMs) were chosen as a data-classification technique as they can be configured based on a priori knowledge about the data, do not require training, and can be used in detecting the presence of areas of dominance based on the underlying features. This ensures that the data-classification method can be tailored for use with fictional video data. Although FSMs are quite similar in structure and output to other data-classification techniques such as hidden Markov models (HMMs), the primary difference is that FSMs are user designed and do not require training. Although an HMM-based event-detection approach was also implemented for completeness, it was eventually rejected as it was consistently outperformed by the FSM approach.

In total there are six FSMs to detect six different kinds of sequences: a speech FSM, a music FSM, a nonspeech FSM, a static motion FSM, a nonstatic motion FSM and a high-motion/short-shot FSM. Each of the FSMs contain one feature with the exception of the high-motion/short-shot FSM. This was created due to filmmakers' reliance on these particular features to generate excitement.

The general design of all the FSMs employed is shown in Figure 3. Each selected feature has one FSM assigned to it in order to detect sequences for that feature. So for example, there is a speech FSM that detects areas where speech shots are dominant. There are similar FSMs for the other features which generate other sequences. The FSM always begins on

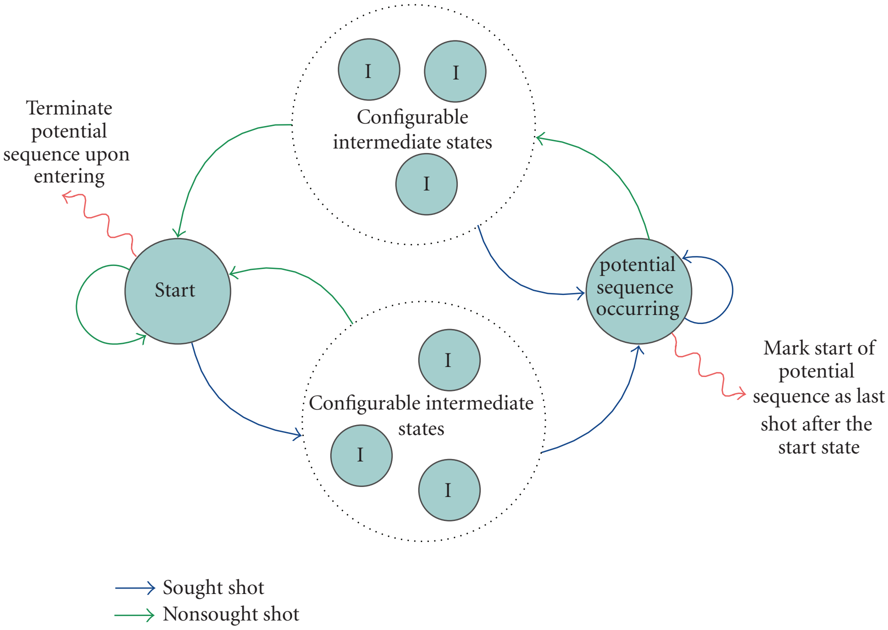

I
I
Configurable intermediate states
I
Terminate potential sequence upon entering
Start
potential sequence occurring
I
Configurable intermediate states
I
I
Mark start of potential sequence as last shot after the start state
Sought shot
Nonsought shot

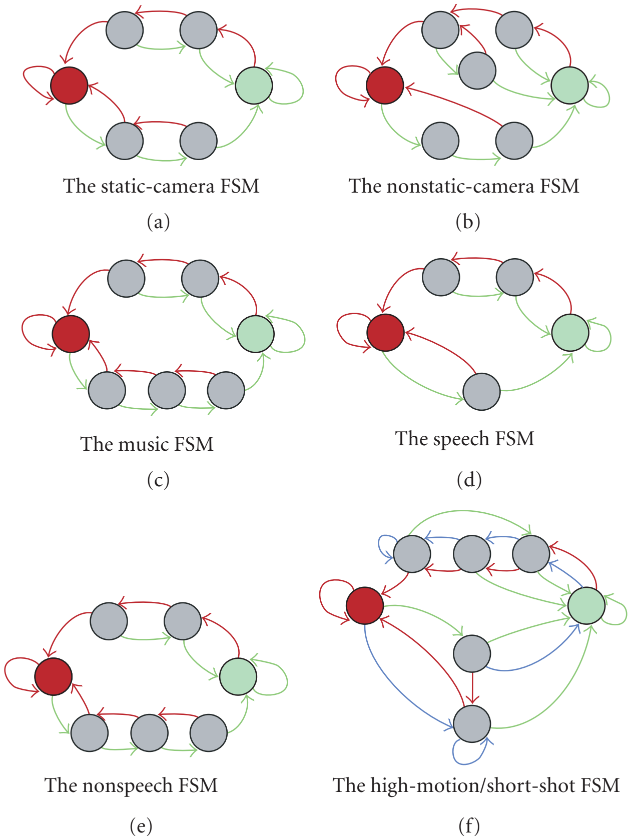
Figure 3: General FSM structure.

the left, in the "start" state. Whenever a shot that contains the desired feature occurs (indicated by the darker, blue arrows in Figure 3), the FSM moves toward the state that declares that a sequence has begun (the state furthest on the right in all FSM diagrams). Whenever an undesired shot occurs (the lighter, green arrows in Figure 3), the FSM moves toward the start state, where it is reset. If the FSM had previously declared that a sequence was occurring, then returning to the Start state will result in the end of the 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 used is the configuration of the intermediate (I) states. Figure 4 illustrates all FSMs employed. In all FSM figures, the bottom set of I-states dictate how difficult it is for the start of a 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 sequence to be declared, as they determine the path from "potential event sequence occurring" back to the "start" state (where the sequence is terminated). In order to find the optimal number of I-states in each individual FSM, varying configurations of the I-states were examined, and compared with a manually created ground truth. The configuration which resulted in the highest overall performance was chosen as the optimal configuration. In all cases, the (lighter) green arrows indicate shots of the type that the FSM is looking for, and the (darker) red arrows indicate all other shots. For example, the green arrows in the "static camera" FSM, indicate shots that predominantly contain static camera frames, and the red arrows indicate all other shots. The only exception to this is in the "high-motion/short-shot" FSM in which there are three arrow types. In this case, the green arrow indicates shots that contain high motion and are short in length. The red arrow indicates shots that contain low motion and are not short, and the blue arrows indicate shots that either contain high motion or are short, but not both.

Due to space restrictions, all of the FSMs cannot be explained in detail here, however the speech FSM is described, and the operation of all other FSMs can be inferred from this. The speech FSM locates areas in the movie where speech shots occur frequently. This does not mean that every shot needs to contain speech, but simply that speech is dominant over nonspeech during any given temporal period. 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 sequence. When back in the initial state, speech shots should not be prevalent. The intermediate states (I-states) effectively act as buffers, for 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 nonspeech shots are present. When speech shots occur, the FSM will drift toward the "speech" state, when nonspeech shots occur the FSM will move toward the "start" state. Upon entering the speech state, the FSM declares that the beginning of a speech 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 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 sequence is declared as the last time the FSM left the speech state.

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


The static-camera FSM

(a)

The nonstatic-camera FSM

(b)

The music FSM

(c)

The speech FSM

(d)

The nonspeech FSM

(e)

The high-motion/short-shot FSM

(f)

Figure 4: All FSMs used in detecting temporal segments where individual features are dominant.

“start” and “speech” states makes it easy for a speech 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 a premature end to a speech sequence being declared.

In all FSMs, if a change of focus is detected via the clustering algorithm described in Section 3.2, then the state machine returns to the start state, and an end to the potential sequence is declared immediately. For example, if there were two dialogue events in a row, there is 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, the change of focus will result in the FSM declaring an end to the potential sequence at the end of the first dialogue event, thereby ensuring detection of two distinct events.

4.2. *Event detection*

In order to detect each of the dialogue, exciting, and montage events, the potential event sequences are used in combination with a number of postprocessing steps as outlined in the following.

4.2.1. *Dialogue events*

As the presence of speech and a static camera are reliable indicators of the occurence of a dialogue event, the sequences detected the speech FSM and static-camera FSM are used. The process used to ascertain if the sequences are dialogue events is as follows.

(a) The CS ratio is generated for both static camera, and speech sequences to determine the amount of shot repetition present.

(b) For sequences detected using the speech-based FSM, the percentage of shots that contain a *static camera* is calculated.

(c) For the sequences detected by the static-camera-based FSM, the percentage of shots containing *speech* in the sequence is calculated.

For any sequence detected using the speech FSM to be declared as a dialogue event, it must have either a low CS ratio *or* a high amount of static shots. Similarly for a sequence detected by the static-camera FSM to be declared a dialogue event, it must have either a low CS ratio *or* a high amount of speech shots. The clustering information from each sequence is also examined in order to further refine the start and end times. As the clusters contain shots of a single character, the first and last shots of the clusters will contain the first and last shots of the people involved in the dialogue. Therefore, these shots are detected and the boundaries of the detected sequences are redefined. 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 must have at least some features consistent with convention.

4.2.2. *Exciting events*

In the case of creating excitement, the two main tools used by directors are fastpaced 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 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.

4.2.3. 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 [14]. Thus, the audio FSMs are essential in detecting montage[2] events. Notice that either the music FSM or the non-speech FSM could be used to generate a set of sequences. Although emotional events usually contain music, it is possible that these events may contain silence, thus the non-speech FSM sequences are used, as these will also contain all music sequences. The following statistical features are then generated for each sequence

(a) The CS Ratio of the sequence.
(b) The percentage of long shots in the sequence.
(c) The percentage of low motion intensity shots in the sequence.
(d) The percentage of static-camera shots in the sequence.

Sequences with very low CS ratios are rejected. This is because sequences with very high amounts of shot repetition are rejected in order to discount dialogue events that take place with a strong musical background. 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 large 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 fastpaced editing is generally avoided when filming a montage. Thus, if there is an absence of these features, the sequence is declared a montage event.

As mentioned in Section 4.2.2, the nonspeech 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, we assume 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 where excitement is required. If, for a non-speech sequence, the last three features (% long shots, % low-motion shots and % static-camera shots) all yield low percentages, then the detected sequence is labelled as an exciting event.

4.3. Searching for events

Although the three event classes that are detected aim to constitute all meaningful events in a movie, in effect they constitute three possible implementations of the same movie-indexing framework. The three event classes targeted were chosen to facilitate fictional video browsing, however, it is desirable that the event-detection techniques can be applied to user-defined searching as well. Thus, the search-based system we propose allows users to control the two steps in event detection 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 sequences detected. 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 sequences, and then filter the results by only accepting the 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 sequences based on the presence of high amounts of speech.

Figure 5 illustrates this two-step approach. In the first step, a FSM is selected (in this case the music FSM). Secondly, the sequences detected are filtered by only retaining those with a user defined amount of (in this case) static camera shots. This results in a retrieved event list as indicated in the figure.

5. RESULTS AND ANALYSIS

In order to assess the performance of the proposed system, over twenty three hours of videos and movies from various genres were chosen as a test set. 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 videos. Many of the videos target vastly different audiences, ranging from animations aimed at young viewers, to violent action movies only suitable for adult viewing. As there may be differing styles depending on cultural influences, the movies in the test set were chosen to represent a broad range of origins, and span different geographical locations including The United States, Australia, Japan, England, and Mexico. The test data in total consists of ten movies corresponding to over eighteen hours of video and a further nine television programs corresponding to over five hours of video. Each of the following subsections examines different aspects of the performance of the system.

5.1. Event detection

For evaluating automatic event detection, each of the videos was manually annotated and the start and end times of each dialogue, exciting and montage event were noted. This manual annotation was then compared with the automatically generated results. Precision and recall values were generated and are presented in Table 1.

It should be noted that in these experiments, 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. There are occasions where the precision value for certain movies is quite low, as there are more detected events than relevant

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

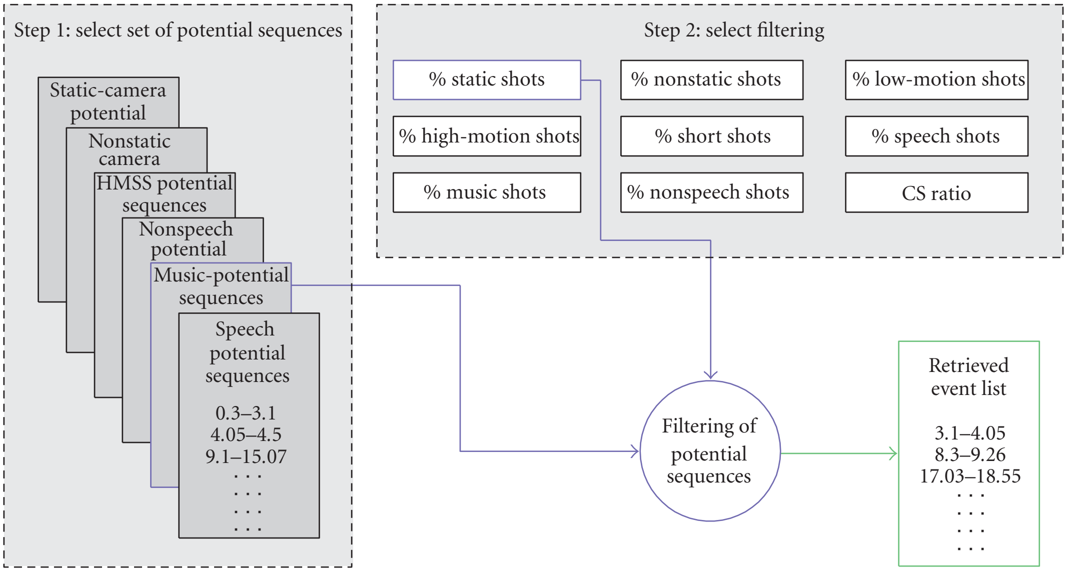


Figure 5: The process involved in user defined searching.

Table 1: Results of event detection using the author's ground truth.

Film name	Dialogue		Exciting		Montage	
	Prec.	Recall	Prec.	Recall	Prec.	Recall
American Beauty	86%	96%	17%	100%	71%	95%
Amores Perros	56%	84%	56%	95%	55%	96%
Battle Royal	62%	94%	71%	91%	72%	90%
Chopper	90%	94%	22%	83%	50%	100%
Dumb & Dumber	74%	91%	55%	100%	68%	86%
Goodfellas	67%	95%	46%	90%	60%	86%
High Fidelity	80%	100%	17%	100%	56%	83%
Reservoir Dogs	89%	94%	50%	80%	100%	100%
Shrek	73%	97%	58%	100%	67%	75%
Snatch	84%	97%	71%	100%	67%	83%
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	**81**%	**94**%	**59**%	**95**%	**73**%	**91**%

ones. However, this scenario is actually beneficial as different viewers often have differing interpretations of events in a movie. This means that some viewers may consider a particular event to be a dialogue event, while others may consider it to be an exciting event (an argument, e.g.). 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 typically decrease the precision value for any one interpretation. This is further explained in Section 5.2. Also, 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 still very efficient.

As can be seen, on average 94% of all dialogue events across all videos are detected by the system. This indicates extremely high performance but there are a number of reasons why the system may miss a dialogue event. The events that are not detected usually have characteristics that are not common to dialogues, for example, 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. However, the high recall rate indicates efficient retrieval.

The recall rates for the exciting events is similarly high, with an average value of 95%. In general, the missed exciting events are short bursts of action that are rejected as being too short. The precision rate is somewhat lower, which is primarily due to the small number 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). Also, 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.

The high recall of the montage events is largely due to filmmakers' reliance on the use of music when filming this type of event. In general the events that are not detected are due to incorrect audio classification where the audio is not correctly labelled as music. Also, most of the false positives are areas that contain speech being labelled as music, primarily due to some background music.

Some events in a movie are detected by the system as belonging to more than one event class. Since there is a certain amount of leeway required in the presentation of events, this dual classification is in fact desirable. This is largely due to the fact that different users will have different interpretations of the same event in a movie. Overall, the most common type of overlap occurs between dialogue and exciting events. An 8.7% of the total shots for all videos were labelled as belonging to both a dialogue event and an exciting event. In general, these occur when there is an element of excitement in a conversation. 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 excitement 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.

Table 2: Results of overlap between different users in manual mark up of events.

Event class	Total events	Combined annotation	Single annotation	No. detected
Dialogue	264	200	64	54(84%)
Exciting	50	22	28	26(93%)
Montage	72	35	37	30(81%)

In total, 4% of the shots were labelled as belonging to both a dialogue event and a montage event. For example, one particular overlap occurs in the film *American Beauty* 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). There is a similarly small dual classification rate between exciting events and montage events (2.4% of shots common to both classes). In this case, dual detection typically occurs 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.

In total, 91.2% of the shots in any given video are placed into at least one of the three event classes. Thus, 8.8% of each video is left unclassified. A common cause of unclassified 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. Also, there are a number of parts of the movie (such as ending credits, etc.) that are intentionally not detected by our indexing system. Finally, although the recall rates for each class are quite high, they are not 100%, so some unclassified shots are due to missed events.

5.2. Accomodating different viewer interpretations

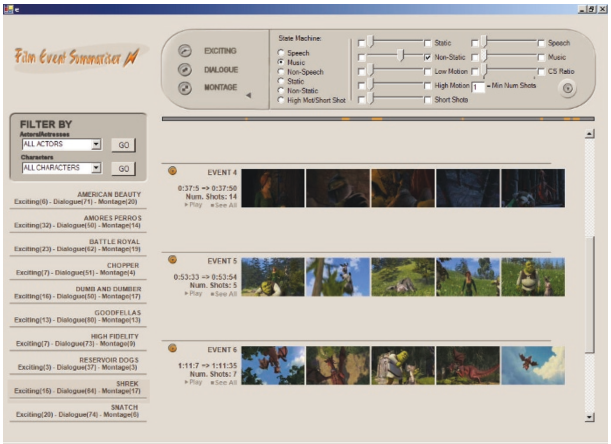
There is significant subjective viewer interpretation involved in terms of determining what constitutes a dialogue, exciting or montage event in the generation of the ground truth used for testing. In order to test our system response to this phenomenon, a number of user trials were conducted. In these trials, two users were asked to independently view the same movie and mark the start and end points of each dialogue, exciting and montage event. Their annotations were firstly compared to each other, and secondly with the results of the automatic system. In total, six films were used and the results are presented in Table 2.

In the table, the first column represents the total number of events manually marked up by *either* viewer. The "Combined annotation" column displays the number of events that both annotators marked in that event class, while the "Single annotation" column gives the number of events that only one person annotated. Finally, the "No. detected" column gives the number of these singly annotated events that were correctly detected by the system. For example, in total there were 264 dialogue events annotated between the two viewers. Twohundreds of these dialogue events were annotated by both, which means they both agreed that a particular part of the movie should be labelled as dialogue. They disagreed on 64 occasions, that is one declared a dialogue event, while the other labelled it belonging to a different event class. Of the 64 occasions on which only one person annotated a dialogue event, the system correctly detected that dialogue event 84% of the time. 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 annotators. 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 our approach. 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. The fact that different viewers can have different interpretations of the same part of the movie indicates that a lower precision value is nessessary for each individual interpretation so that consistantly high recall can be achieved and users can locate the sought events.

5.3. User trials

Having developed a system for detecting all of the dialogue, exciting, and montage events in a movie, as well as facilitating event-based searching, a presentation mechanism to assess the indexing solutions was required. To this end, a user interface, named the *MovieBrowser*, 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 4.3 is also incorporated into the system. This allows a direct comparison between searching for events and browsing a predefined index, as well as demonstrating one potential application of our research.

The basic display unit of the MovieBrowser is an event. When displaying each event, 5 representative keyframes are displayed as well as some additional information about the event (start/end times, number of frames, etc.). Users can play the event in an external video player by clicking on the "Play" button. It is possible to browse the movie using either the event-based index or by searching. In order to browse the event-based index, users can click on the corresponding event-class (either dialogue, exciting, or montage). Each detected event is then displayed in temporal order. In order to search, users can input queries (by selecting an FSM and some filtering), and are presented by the detected events. Figure 6 shows the MovieBrowser displaying the results of one such search. Further details of this system can be seen in [20].



AMERICAN BEAUTY
Exciting(6) - Dialogue(71) - Montage(20)

AMORES PERROS
Exciting(32) - Dialogue(50) - Montage(14)

BATTLE ROYAL
Exciting(23) - Dialogue(62) - Montage(19)

CHOPPER
Exciting(7) - Dialogue(51) - Montage(4)

DUMB AND DUMBER
Exciting(16) - Dialogue(50) - Montage(17)

GOODFELLAS
Exciting(13) - Dialogue(80) - Montage(13)

HIGH FIDELITY
Exciting(7) - Dialogue(73) - Montage(9)

RESERVOIR DOGS
Exciting(3) - Dialogue(37) - Montage(3)

SHREK
Exciting(16) - Dialogue(64) - Montage(17)

SNATCH
Exciting(20) - Dialogue(74) - Montage(6)

EVENT 4
0:37:5 => 0:37:50
Num. Shots: 14

EVENT 5
0:53:33 => 0:53:54
Num. Shots: 5

EVENT 6
1:11:7 => 1:11:35
Num. Shots: 7

Figure 6: Retrieved events after searching for events that contain high amounts of music and moving camera in MovieBrowser.

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 user 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 two different browsing methods (event-based and search-based).

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. Each clip corresponds to a known item of retrieval where it is known (although not to the searcher) that one and only one clip will satisfy the search request. An example of a task is: *In the film High Fidelity, find the part where Barry sings "Lets get it on" with his band*. 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 task list was generated by asking viewers who had previously seen the film to name the most memorable events. The complete task list is quite diverse as it incorporates many different occurrences in a wide range of movies.

An automatic timing program was implemented that recorded how long it took a user to complete each task, and also to check whether users located the correct event. Once a user located a clip in the movie that he/she considers to be correct, they entered the time of the event into the system (which compared this time with the correct start and end times of the tasks). If the supplied time was correct (i.e., between the start and end time of the task), the time taken to complete the task was automatically recorded. If a user supplied an incorrect time, he/she was instructed to continue browsing in order to find the correct time of the event. If a user could not complete a task, there was an option to give up browsing. If this happened, a completion time of ten minutes was assigned for the task. This heavily penalised

Table 3: Average time in seconds taken to complete tasks using each browsing method.

Method used	*All movies*	*Unseen movies*	*Seen movies*
Event based	81.3	111.5	71.3
Search based	98.9	124.3	92.7

noncompletion of tasks. In order to compare results for different users, a pretest questionnaire was created in which the volunteers were required to state which films they had seen before.

The average time for users of the event-based method to complete a task was 81.3 seconds. The average time for users of the search-based method to complete a task was 98.9 seconds. Predictably, when people had seen the movie previously their retrieval time was reduced, while the opposite can be said for people who had not seen the movie. These results are presented in Table 3.

The task completion times for the event-based method of browsing are consistently lower than for the searching system. On average, it is approximately 20% 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. 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 browsed 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.

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.

The results of the MovieBrowser experiments indicate that imposing an event-based structure on a movie is highly beneficial in locating specific parts of the movie. This is demonstrated in the high performance of both the event and search-based methods.

6. CONCLUSION

The primary aim of this research was to create a system that is capable of indexing entire movies and entire episodes of fictional television content completely automatically. In order to achieve this aim, we implemented two browsing methods. The first was an event-based structure that detects the meaningful events in a movie according to a predefined index. To this end, an event detection approach that utilises audio-visual analysis based on film-creation techniques was designed and implemented. The second browsing method facilitated user-driven searching of video content in order to retrieve events.

As can be seen from the experiments reported in Section 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. Also, there is only a small amount of shots in any given movie that are not classed into one of the event classes. 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. These results are significant as they demonstrate that an overall event-based summary of a film is possible. Upon analysing different peoples' interpretation of the same movies it can be concluded that consistently high recall of events is desired, however a lower precision value is necessary in order to facilitate differing opinions. As the results of Section 5.3 show, searching can also result in a short retrieval time, especially in cases where users chose features that accurately represent the sought events. The results of the searching technique are particularly encouraging, as they indicate that general users can easily relate to an event-based film representation. Clearly this should be reflected in the structure of future video search systems.

In considering the end-user applications of this work, we can envisage Video-on-Demand websites that contain preview of their movie collections in which the users can jump to dialogue/exciting/montage events before paying for full-streaming, or a "scene access" feature (similar to those seen in many commercial DVD movie menus) which is automatically generated and that highlights dialogue/exciting/montage events when a user downloads or records a movie on his/her set-top box. This is especially relevant given the recent shift toward video on demand technologies in the set-top box market. The playback interface (on the Web, TV, or media centre) could focus on the selection of movies and keyframe presentation (as has been done in our MovieBrowser (Figure 6)), or focus on various preview techniques, community-based commenting, voting, or even annotating different parts of the movies by the viewers for content sharing.

Future work in this area will involve incorporating additional features into the system framework. This may include textual information, possibly taken from subtitle information, which could improve retrieval efficiency, or face detection, which would provide additional information about the content. Speech recognition software may also be utilised in order to improve the system's audio analysis performance.

ACKNOWLEDGMENT

The research leading to this paper was partly supported by Enterprise Ireland and by Science Foundation Ireland under Grant no. 03/IN.3/I361.

REFERENCES

[1] "The Internet movie database," http://www.imdb.com/, September 2006.

[2] M. Yeung and B.-L. Yeo, "Time constrained clustering for segmentation of video into story units," in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 3, pp. 375–380, Vienna, Austria, August 1996.

[3] M. Yeung and B.-L. Yeo, "Video visualisation for compact presentation and fast browsing of pictorial content," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 5, pp. 771–785, 1997.

[4] Z. Rasheed and M. Shah, "Scene detection in Hollywood movies and TV shows," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 2, pp. 343–348, Madison, Wis, USA, June 2003.

[5] J. R. Kender and B.-L. Yeo, "Video scene segmentation via continuous video coherence," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '98)*, pp. 367–373, Santa Barbara, Calif, USA, June 1998.

[6] H. Sundaram and S.-F. Chan, "Determining computable scenes in films and their structures using audio-visual memory models," in *Proceedings of the 8th ACM International Conference on Multimedia*, pp. 95–104, Los Angeles, Calif, USA, October-November 2000.

[7] Y. Cao, W. Tavanapong, K. Kim, and J. Oh, "Audio-assisted scene segmentation for story browsing," in *Proceedings of the 2nd International Conference Image and Video Retrieval (CIVR '03)*, pp. 446–455, Urbana-Champaign, Ill, USA, July 2003.

[8] A. A. Alatan, A. N. Akansu, and W. Wolf, "Multi-modal dialogue scene detection using hidden Markov models for content-based multimedia indexing," *Multimedia Tools and Applications*, vol. 14, no. 2, pp. 137–151, 2001.

[9] R. Leinhart, S. Pfeiffer, and W. Effelsberg, "Scene determination based on video and audio features," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, vol. 1, pp. 685–690, Florence, Italy, June 1999.

[10] Y. Li and C. C. Jay Kou, *Video Content Analysis Using Multimodal Information*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.

[11] Y. Li and C. C. Jay Kou, "Movie event detection by using audio visual information," in *Proceedings of the 2nd IEEE Pacific Rim Conference on Advances in Multimedia Information Processing*, pp. 198–205, Beijing, China, October 2001.

[12] Y. Zhai, Z. Rasheed, and M. Shah, "A framework for semantic classification of scenes using finite state machines," in *Proceedings of the International Conference on Image and Video Retrieval (CIVR '04)*, pp. 279–288, Dublin, Ireland, July 2004.

[13] Y. Zhai, Z. Rasheed, and M. Shah, "Semantic classification of movie scenes using finite state machines," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 152, no. 6, pp. 896–901, 2005.

[14] D. Bordwell and K. Thompson, *Film Art: An Introduction*, McGraw-Hill, New York, NY, USA, 1997.

[15] P. Browne, A. F. Smeaton, N. Murphy, N. E. O'Connor, S. Marlow, and C. Berrut, "Evaluating and combining digital video shot boundary detection algorithms," in *Proceedings of Irish Machine Vision and Image Processing Conference (IMVIP '02)*, North Ireland, UK, August-September 2002.

[16] Y. Rui, T. S. Huang, and S. Mehrotra, "Constructing table-of-content for video," *Journal of Multimedia System*, vol. 7, no. 5, pp. 359–368, 1999.

[17] B. Lehane, N. E. O'Connor, and N. Murphy, "Dialogue sequence detection in movies," in *Proceedings of the 4th International Conference on Image and Video Retrieval (CIVR '05)*, pp. 286–296, Singapore, July 2005.

[18] B. Manjunath, P. Salember, and T. Sikora, *Introduction to MPEG-7, Multimedia Content Description Language*, John Wiley & Sons, New York, NY, USA, 2002.

[19] L. Chen, S. J. Rizvi, and M. T. Özsu, "Incorporating audio cues into dialog and action scene extraction," in *Storage and Retrieval for Media Databases*, vol. 5021 of *Proceedings of SPIE*, pp. 252–263, Santa Clara, Calif, USA, January 2003.

[20] B. Lehane, N. E. O'Connor, A. F. Smeaton, and H. Lee, "A system for event-based film browsing," in *The 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE '06)*, pp. 334–345, Darmstadt, Germany, December 2006.