A System For Event-Based Film Browsing

Bart Lehane, Noel E. O'Connor, Alan F. Smeaton, and Hyowon Lee

Centre for Digital Video Processing and Adaptive Information Cluster,
Dublin City University, Ireland
lehaneb@eeng.dcu.ie
oconnorn@eeng.dcu.ie
asmeaton@computing.dcu.ie
hlee@computing.dcu.ie
http://www.cdvp.dcu.ie

Abstract. The recent past has seen a proliferation in the amount of digital video content being created and consumed. This is perhaps being driven by the increase in audiovisual quality, as well as the ease with which production, reproduction and consumption is now possible. The widespread use of digital video, as opposed its analogue counterpart, has opened up a plethora of previously impossible applications. This paper builds upon previous work that analysed digital video, namely movies, in order to facilitate presentation in an easily navigable manner. A film browsing interface, termed the MovieBrowser, is described, which allows users to easily locate specific portions of movies, as well as to obtain an understanding of the filming being perused. A number of experiments which assess the system's performance are also presented.

Key words: Movie Indexing, Summarisation, Presentation, Interface

1 Introduction

The past number of years has seen a growth in the use of digital video for creating, editing, broadcasting and viewing content. The amount of movies created each year has grown considerably since cost effective digital filming and editing equipment has become available. For example, according to [IMDB, 2006], 10,342 film and video titles were released worldwide in 2001 alone. This digital media revolution was helped by efficient storage and transmission, while an advantageous by-product of using digital, as opposed to analogue, video and audio, is that it is possible to analyse the data automatically. Where previously video was merely stored on a reel of tape, the use of digital video means that it is possible to extract information from the video data and use it to gain knowledge about the content. Unfortunately, most, if not all, of this content is simply stored without any sort of indexing or analysis and without any associated meta-data. For videos with meta-data, then it is usually 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 the retrieval process as well as allowing users to gain some higher knowledge about the video content. Films are particularly in need of indexing as their temporally long nature and varying styles make it is difficult to know where, and indeed how, to locate desired clips.

The research presented here describes a system that allows users to retrieve sought after parts of films, as well as to gain knowledge about their content. Three methods of navigation are presented, *shot-based browsing*, *event-based browsing* and *search-based browsing*. Previous work by the authors, which is summarised later, used audiovisual analysis to automatically create an event-based structure of movies as well as facilitating user initiated searching. The system described here takes the results of this and presents it as a complete system. This allows for the efficient retrieval of sought portions of a movie.

It is necessary to introduce the concept of an *event* at this point. As defined in this work, 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 the 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.

There have been a number of other approaches that aim to 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 (what are termed events in this paper). [Yeung and Yeo, 1996,Yeung and Yeo, 1997] propose a scene boundary detection 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. Scene boundaries are located by examining the structure of the clusters and detecting points where one set of clusters ends, and another begins. Approaches such as [Sundaram and Chan, 2000] and [Cao et al., 2003] define a computable scene as one which exhibits long term consistency of chrominance, lighting and ambient sound and use audiovisual 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 the content is inherent in the index. A user searching for a particular point in the movie must peruse the whole movie in order to locate it, unless significant prior knowledge of the movie is available.

Many event detection techniques in movie analysis focus on detecting individual event types from the video. [Leinhart et al., 1999] detect dialogues in video based on the common shot/reverse shot shooting technique, where 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 apart from dialogues. [Li and Kou, 2003,Li and Kou, 2001] 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., 2003] 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. [Zhai et al., 2004] generate colour, motion and audio features for a video, and then use finite state machines to classify scenes into either conversation scenes, suspense scenes or action scenes. A high classification rate is achieved. However, this approach relies on the presence of known scene breaks, and classifies a whole scene into one of the categories, while in practice an entire scene may contain a number of important events.

In general, event detection approaches are focused on detecting single events, however, previous approaches by the authors created a system which completely indexes a movie by detecting all of the relevant events present. By creating a number of event classes that represent as much of a movie as possible (dialogue events, exciting events and musical events) and then detecting all of the events in these classes, an event-based index of a movie can be created [Lehane et al., 2004,Lehane et al., 2005,Lehane and O'Connor, 2006]. A brief overview of this technique, as well as some results presented in previous papers, are summarised later for context.

The remainder of this paper is organised as follows. Section 2 introduces the requirements for a film browsing system and presents some of the design choices made in the design process. Section 3 describes the underlying technology behind the browsing system, while Section 4 describes the user interface. Section 5 explains the set of experiments undertaken in order to assess the system. Finally, Section 6 draws a number of conclusions and indicates potential future work.

2 System Design

The aim of the system presented in this paper is to allow users to quickly and efficiently locate relevant portions of movies and also to assist in the understanding of a movie. In order to facilitate this, the system has a number of requirements. Firstly, the index should be event based so that some knowledge of the movie is inherent in the index. For example, if a user is looking at a scene-boundary based index, where each scene is presented to a user, it is quite difficult to locate relevant portions of a movie, or indeed garner any information about the movie, without actually viewing each of the scenes. Creating an event-based index, where each event belongs to a particular class, makes it easier to browse and interpret the movie as each event class is known, and therefore the browser has a better idea of what is taking place in the movie simply by viewing keyframes from the event. In order to create event-based browsing, a number of event classes must be defined. The event classes should be plentiful enough to cover all of the meaningful parts in a movie, yet generic enough so that only a low amount of event classes are required. This is to ensure that the index is as compact 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 is an impossible task. It is proposed to create a reasonable number of event classes, some of which may encompass a number of different events. Each of the events in any event class have a common semantic thread that link the events together allowing intuitive navigation through a movie. However, the selection, and amount, of event classes is dictated by how the films themselves are created.

The first event class contains all *Dialogue* events. Dialogue constitutes a major part of any film, and the viewer usually gets 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 number of characters is classified 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 (or *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. A director has a set of tools available to create excitement (such as increased editing pace, on-screen movement etc) and when these tools are used it is a good indication that an exciting event is taking place.

The final event class is a superset of a number of different events, which are all labelled as *Musical* events. The first type of event in this superset are montage events. As a montage brings a number of unrelated shots together, typically with musical accompaniment that spans all of the shots. The second event type labelled in the musical 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 this class are *musical* events themselves. 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 'Musical' event class. Any future reference to musical events in this paper is referring to the entire set of events labelled as 'musical'.

The three event classes described aim to cover all meaningful parts of a movie. The distinction between the three event classes is quite subjective. One person may feel that a particular event belongs to a certain class, while another feels it belongs to a different class. An argument, for example, could be interpreted as an exciting event by one user, and as a dialogue event by another. Many montages events also aim to excite the viewer, and therefore could also be classified as exciting events or musical 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 musical event class counts as a multi-class event. Clearly, if manageable content is required, 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. Browsing a movie based on these three event classes is termed *event-based* browsing.

Although the event-based index aims to incorporate all relevant events in a movie, there may be occasions when a different set of events are sought. For example, a user may be interested in examining how a particular director uses editing throughout a movie and may want to locate all of the areas (or events) where fast paced editing is used. Thus, another requirement for the system is to allow users to initiate event-based searching. This allows for more specific, user defined browsing through a movie, as 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 is termed *search-based browsing.* Also, there may be portions of a movie that are not part of an event-based index, or cannot be located by searching. Thus, as a last resort, a method of examining all of the shots in a movie is facilitated. This is termed *shot-based browsing.*

3 System Description

Section 2 identified three methods of browsing supported by our film browsing system namely *shot-based* browsing, *event-based* browsing (using the three event classes) and *search-based* browsing. Previous work by the authors focused on detecting the events belonging to each event class [Lehane and O'Connor, 2006] and on facilitating searching through a movie [Lehane et al., 2006]. Much of the underlying mechanics of the system has been presented previously, however knowledge of some aspects of the system is required and so a summary of the most important parts is supplied here.

3.1 Shot-Based Analysis

As with many audiovisual analysis techniques, the first step in video analysis involves detecting shot boundaries. The approach employed uses colour histograms to detect large inter-frame variances in colour which can be attributed to a change in camera angle. When the approach was examined against a manually created ground truth, 97% of shot boundaries were detected which indicates that it is quite accurate. Once shot boundaries are detected, a representative frame, or keyframe, is selected for each shot. As this is the sole representation for each shot, the frame that is visually closest to the average frame in the shot is used. Colour histograms are used in order to determine this. By combining the two automatic processes of shot boundary detection and keyframe selection, it is possible to implement shot-based browsing. The implementation of this is presented in Section 4

3.2 Event-Based Analysis

In order to detect events, film creation techniques were examined, and the features commonly used were extracted. For example, when shooting a dialogue event, a director will usually try to relax the audience so that they can interpret the words being spoken. This typically results in a relaxed shooting style which contains little or no camera movement, repetitive shots and clearly audible speech. When shooting an exciting event however, the aims of the filmmaker are different. Typically, fast-paced editing and high amounts of camera movement are used in order to create excitement in the audience. Finally, when filming what we term musical events, there will be a constant musical audio track, usually combined with low amounts of camera movement and slower paced editing [Lehane and O'Connor, 2006].

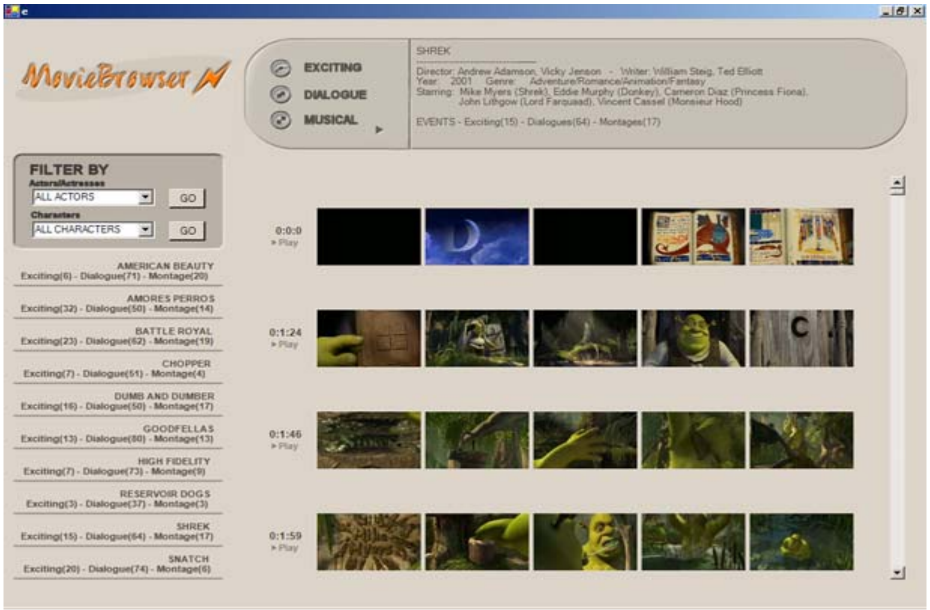
Thus, in order to detect the events contained in these three event classes, the audiovisual features associated with each event class are extracted. The editing pace can be extracted by examining the shot boundary information and using the shot lengths. Two features which describe the motion present in each shot were extracted. The first measures the amount of *camera movement* present, while the second measures the amount of motion present within the frame (i.e. occasions where there is a still camera but an object moving within the frame) and is termed *motion intensity*. Both of these features combine to give a complete description of the movement in each shot. In order to identify the type of audio present in each shot, a support vector machine based audio classifier was implemented which detects the amount of *speech*, *music*, *silence* and *other audio* present in each shot [Lehane et al., 2005]. Finally, a measure of shot repetition is implemented which measures, for a given sequence of shots, how many repeating shots are present [Lehane et al., 2005].

Once all of these features are extracted, the events themselves can be detected. The event detection approach was previously presented in [Lehane and O'Connor, 2006]. Essentially, a set of finite state machines (FSMs) are used in order to detect parts of a movie where particular features are prominent, then some filtering is applied which removes incorrectly detected events. For example, in order to detect dialogue events, FSMs are used in order to detect areas which contain various combinations of speech shots, still cameras and repeating shots. The output of the FSMs is filtered, and a list of dialogue events is created. A similar process is repeated for the exciting events (where fast-paced editing and camera movement are sought) and for musical events (where, among others, shots with silence and music are sought). The output of the event detection process yields a list of dialogue, exciting and musical events present in the movie.

The results of the event detection process, which was tested on a varied collection of ten movies, are also present in [Lehane and O'Connor, 2006]. It was reported that, on average, 95% of dialogue events, 94% of exciting events and 90% of musical events were detected by the system, which indicates the reliability of the event detection system. The detection of these events facilitates the use of an event-based index. Also, on average 91% of all shots in a movie were categorised into one of the event classes by the event detection system, which indicates that the three event classes cover a high percentage of the footage in a movie.

3.3 Search-Based Analysis

In order to allow users to search for particular parts of a movie, a similar approach to the event detection method is used. This can be viewed in [Lehane et al., 2006]. All of the features described in Section 3.2 are also used for searching. As with event detection, searching involves two steps. Firstly, a user selects the desired FSM, and secondly selects the filtering (if any) required. So, for example, a user looking for an event that contains a moving camera and music could use the moving camera FSM to find all of the areas in the movie with a moving camera, then filter the results to only retain the parts that also contain music. Another way of searching for the same event would be to use the music FSM and then filter the results to only retain events with high amounts of moving camera shots. In another situation, a user may want to locate all of the areas



SHREK

Director: Andrew Adamson, Vicky Jenson - Writer: William Steig, Ted Elliott
Year: 2001 Genre: Adventure/Romance/Animation/Fantasy
Starring: Mike Myers (Shrek), Eddie Murphy (Donkey), Cameron Diaz (Princess Fiona),
John Lithgow (Lord Farquaad), Vincent Cassel (Monsieur Hood)

EVENTS - Exciting(15) - Dialogues(64) - Montages(17)

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(15) - Dialogue(64) - Montage(17)

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

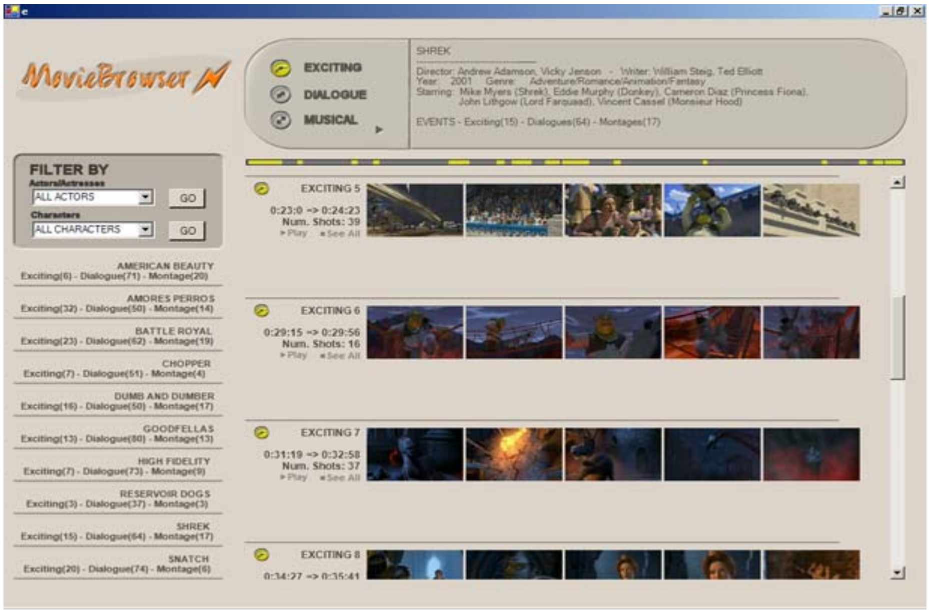
Fig. 1. Shot-based browsing using the MovieBrowser system

with speech present, then he/she can use the speech FSM with no additional filtering and will be returned all of the areas in a movie with speech present. An implementation of this searching technique is presented in Section 4.

4 User Interface: The MovieBrowser

This section presents the user interface to the *MovieBrowser* system which implements the three methods of perusing a movie. Figures 1, 2 and 3 show the respective browsing methods. Firstly, notice that on the left hand side of the interface, each of the movies in the database are listed. By clicking on a movie, the system switches to present information (director, actors, character names, year of release, genre etc.) about the selected movie. Figure 1 shows the shot-based browsing view. In this view, the keyframe from each shot of the movie is presented in temporal order. Each row contains five shots, and a user can play the movie from any point by clicking on the respective icon. Although this is a relatively basic method of browsing a movie, it may be useful in certain scenarios where the location of a clip is known.

The event-based browsing method is shown in Figure 2. By clicking on one of the event icons in the top of the screen (i.e. exciting, dialogue or musical), each of the events in that class for the selected movie are displayed. For example, in Figure 2, all of the detected exciting events for the film 'Shrek' are displayed. Just below the information panel, a visual guide that indicates the location of the events in the movie is shown. Each event is given five keyframes in order to give the user an idea of what is taking place during the event. The event keyframes for the dialogue events are selected based on the most commonly repeating shots, as these usually correspond to the characters speaking in the event and are therefore



SHREK

Director: Andrew Adamson, Vicky Jenson - Writer: William Steig, Ted Elliott
Year: 2001 Genre: Adventure/Romance/Animation/Fantasy
Starring: Mike Myers (Shrek), Eddie Murphy (Donkey), Cameron Diaz (Princess Fiona),
John Lithgow (Lord Farquaad), Vincent Cassel (Monsieur Hood)

EVENTS - Exciting(15) - Dialogues(64) - Montages(17)

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(15) - Dialogue(64) - Montage(17)

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

EXCITING 5
0:23:0 -> 0:24:23
Num. Shots: 39

EXCITING 6
0:29:15 -> 0:29:56
Num. Shots: 16

EXCITING 7
0:31:19 -> 0:32:58
Num. Shots: 37

EXCITING 8
0:34:27 -> 0:35:41

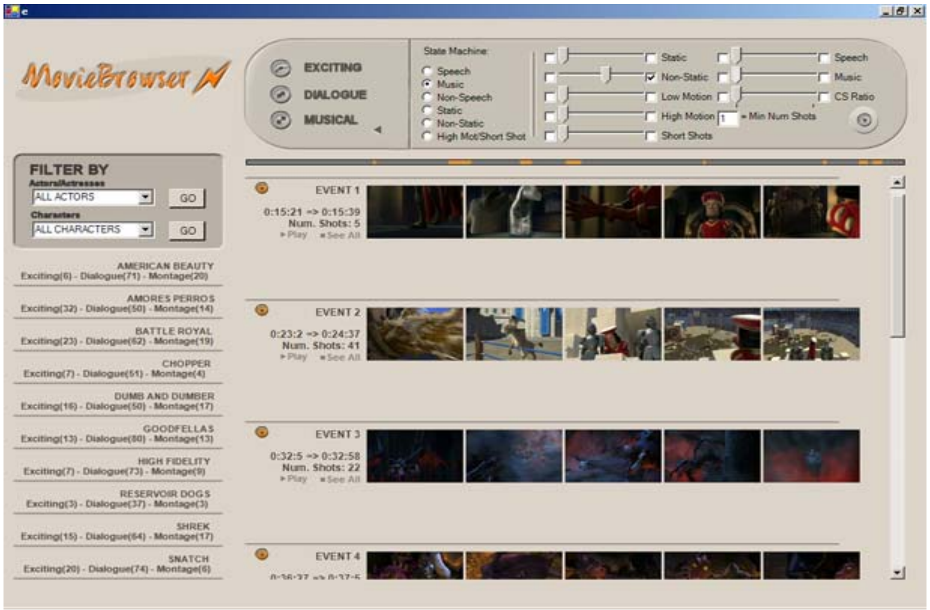
Fig. 2. Event-based browsing displaying the exciting events in the movie Shrek

deemed most appropriate. For the exciting and musical events, it is more difficult to reliably select event keyframes as there are many different possible activities, so they are selected at equal time increments throughout the event. For each event, the start and end time is displayed, as well as the number of shots present. It is possible to play any event from the beginning by clicking on the 'Play' button.

The final method of browsing a movie, the search-based method, is presented in Figure 3. This allows users to initiate a search. By clicking on the arrow icon, the search panel is revealed. Users can then select the desired FSM on the left hand side, and the filtering on the right. For example, in Figure 3 the 'Music' FSM is selected, and only events with high amounts of 'Non-Static' camera shots are retained. This returns a set of events, which are displayed in the same manner as in the event-based approach.

5 Experiments and Testing

In order to assess the effectiveness of this system as an indexing solution, a set of browsing experiments using the MovieBrowser were devised. The purpose of the experiments is to investigate whether the system facilitates efficient retrieval and also, which method of browsing users find most useful. The process involved a number of users completing a set of tasks, which involved retrieving particular clips using the three different browsing methods. A set of thirty tasks were created. For each task, a user used 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 tasks were chosen in order to assess how well the respective browsing and retrieval methods can be used in a movie



State Machine:
○ Speech
● Music
○ Non-Speech
○ Static
○ Non-Static
○ High Mot/Short Shot

□ Static
☑ Non-Static
□ Low Motion
□ High Motion
1 = Min Num Shots
□ Short Shots
□ Speech
□ Music
□ CS Ratio

FILTER BY
Actors/Actresses: ALL ACTORS
Characters: ALL CHARACTERS

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(15) - Dialogue(64) - Montage(17)

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

EVENT 1
0:15:21 -> 0:15:39
Num. Shots: 5

EVENT 2
0:23:2 -> 0:24:37
Num. Shots: 41

EVENT 3
0:32:5 -> 0:32:58
Num. Shots: 22

EVENT 4
0:36:37 -> 0:37:5

Fig. 3. Search-based browsing displaying the retrieved events after searching for events that contain high amounts of music and moving camera shots

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.

In total there were twelve volunteers, each one completing fifteen tasks, five for each method of browsing. So, for example, user 1 completed tasks 0 to 4 using shot-based browsing, 5 to 9 using event-based browsing, and 10 to 14 using search-based 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. When completing a task with one browsing method, the functionality of the other methods was removed. An automatic timing program was implemented that recorded how long it took each user to complete each task, and also to check whether users have located the correct event. If a user could not complete a task, a completion time of ten minutes was assigned for the task. This heavily penalises non-completion of tasks.

The results of these tasks are presented in Figure 4. The vertical axis is the time in seconds taken to complete the task, and the horizontal axis is the task

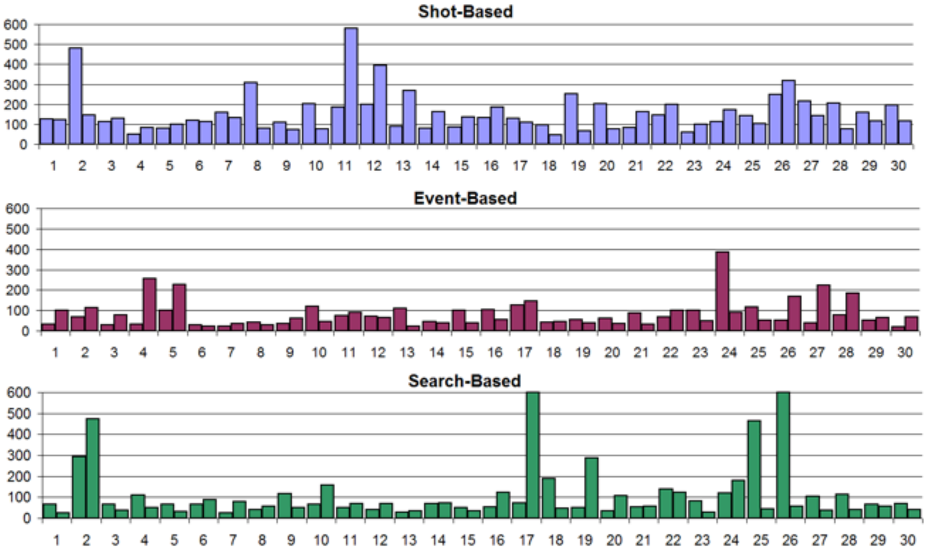


Fig. 4. Plot of time taken to locate clip for the tasks using three browsing methods

index. There are three graphs in the figure, one for each method of browsing. In each graph, there are two results for each task (as two users completed each one). As can be seen from the graph, the shot-based system has many longer completion times than the other methods of browsing. Many of the shot-based searches take over 100 seconds, and a number take considerably longer. In contrast, most of the event-based and search-based completion times are quite low. 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, although the results returned from searching contained the sought event, these were not recognised by the user. The minimum completion time was 18 seconds for user 1 in task 30 using the event-based system.

Clearly it is possible that the results could be biased depending on which films users had previously seen as he/she may know immediately where in the movie a particular event occurs. As such, analysis of the results based on whether a person had seen the film was also undertaken. In total, 180 tasks were completed (i.e. the 30 tasks completed by six people each). For 42 of these tasks, users had not seen the film before. Table 1 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 longer for users who have not previously viewed the movie, however in both cases the event-based method performs best, followed by the search-based method, followed by the shot-based method.

The results presented indicate that the events detected by the system correspond to the users' interpretations of the events, and are located in the correct event class. As these results show, implementing an event-based index has a number of advantages which result in reduced search time. This can be intuitively explained as when an event class is selected, users are significantly reducing the search space of the movie. Also, as they know that the events that they are

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

Table 1. Average task completion times by browsing method, where the average results are shown for users that had, and had not seen the film previously

looking at belong to a particular class, this helps them to understand what is transpiring in the event. For example, looking at the representative keyframes for a dialogue event allows a browser to reach the conclusion that the characters in the keyframes are talking to each other. If the same characters were viewed in the keyframes of an exciting event, the user can infer that they are fighting or arguing. This is far more difficult to achieve when viewing the keyframes on their own with no context.

Initiating searches with the features that users feel are common to the sought events proved to be a reliable method of finding events in movies. Performance for the search-based system may be increased if the user interface of the search based method is improved, 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 Conclusions

This paper described the MovieBrowser system which allows users to browse movies using a number of methods. Shot-based, event-based and search-based methods for browsing were described and the results of a number of experiments comparing their use were presented. The advantages of event-based browsing was illustrated, and proved to be 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 shot based method. The combination of all three browsing methods creates a complete index of a movie. Although the experiments were focused on using the system in an event retrieval scenario, future experiments will aim to assess how much insight into the actual movie can be extracted by using this system.

The system was recently deployed in a learning environment at a University. A set of 12 movies studied as part of a film studies course were analysed and incorporated into the system, which is now used by students in order to assist in their analysis of films. Future work will focus on altering this system based on feedback from the students and faculty using it, so that it is more tailored toward this specific learning environment. 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. Also, the addition of text information (obtained from the subtitle information) to the MovieBrowser may result in even more effective retrieval. Finally, additional visual analysis may yield even more contextual information from the movie. For example, face detection, detection of the camera framing etc, may enhance the system.

7 Acknowledgement

The research leading to this paper was partly supported by Enterprise Ireland and by Science Foundation Ireland under grant number 03/IN.3/I361

References

[Cao et al., 2003] Cao, Y., Tavanapong, W., Kim, K., and Oh, J. (2003). Audio-assisted scene segmentation for story browsing. In *Proceedings of the International Conference on Image and Video Retrieval*.

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

[IMDB, 2006] IMDB (2006). The Internet Movie Database - IMDb. http://www.imdb.com/.

[Lehane and O'Connor, 2006] Lehane, B. and O'Connor, N. (2006). Movie indexing via event detection. In *Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Incheon, Korea*.

[Lehane et al., 2006] Lehane, B., O'Connor, N., and Lee, H. (2006). Searching movies based on user defined semantic events. In *International Conference of Signal Processing and Multimedia Applications (SIGMAP), Setubal, Portugal*.

[Lehane et al., 2004] Lehane, B., O'Connor, N., and Murphy, N. (2004). Action sequence detection in motion pictures. In *The international Workshop on Multidisciplinary Image, Video, and Audio Retrieval and Mining*.

[Lehane et al., 2005] Lehane, B., O'Connor, N., and Murphy, N. (2005). Dialogue scene detection in movies. In *International Conference on Image and Video Retrieval (CIVR), Singapore, 20-22 July 2005*, pages 286–296.

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

[Li and Kou, 2001] Li, Y. and Kou, C.-C. J. (2001). Movie event detection by using audiovisual information. In *Proceedings of the Second IEEE Pacific Rim Conferences on Multimedia: Advances in Multimedia Information PRocessing*.

[Li and Kou, 2003] Li, Y. and Kou, C.-C. J. (2003). *Video Content Analysis using Multimodal Information*. Kluwer Academic Publishers.

[Sundaram and Chan, 2000] Sundaram, H. and Chan, S.-F. (2000). Determining computable scenes in films and their structures using audio-visual memory models. In *ACM Multimedia 2000*.

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

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

[Zhai et al., 2004] Zhai, Y., Rasheed, Z., and Shah, M. (2004). A framework for semantic classification of scenes using finite state machines. In *International Conference on Image and Video Retrieval*.