DIALOGUE SCENE DETECTION IN MOVIES USING LOW AND MID-LEVEL VISUAL FEATURES

Bart Lehane, Noel O'Connor, Noel Murphy

Centre for Digital Video Processing

Dublin City University

{lehaneb, oconnorn, murphyn}@eeng.dcu.ie

Abstract

This paper describes an approach for detecting dialogue scenes in movies. The approach uses automatically extracted low- and mid-level visual features that characterise the visual content of individual shots, and which are then combined using a state transition machine that models the shot-level temporal characteristics of the scene under investigation. The choice of visual features used is motivated by a consideration of formal film syntax. The system is designed so that the analysis may be applied in order to detect different types of scenes, although in this paper we focus on dialogue sequences as these are the most prevalent scenes in the movies considered to date.

1 Introduction

The decreasing cost of digital storage, coupled with the fact that newer and better video/audio compression formats are being standardised, means that it is now possible for people to build large personal digital video libraries, similar to the way personal digital music libraries have become commonplace. In fact, the use of digital films on the Internet can be compared to the way digital music files were used a few years ago. Back when the standard Internet connection was a dial up modem, and hard drives were small, downloading and storing an entire music album was quite a lengthy task, whereas now, as the emergence of online music stores shows, it is possible to download significant quantities of music quickly and easily. As the bandwidth available to users increases, online film stores will increase in popularity and facilitate consumers in building personal film libraries.

Such video libraries will be extremely difficult to organise and browse unless tools are available to the user to index and organise the content according to the key events and scenes depicted. Clearly such annotation could be performed by the content provider, but this will not happen in the short-term, and even if manual annotation is available, it is desirable to develop automatic tools that make the indexing task as easy as possible. Scene detection is a useful tool as a first pass in order to organise the content into important and meaningful segments. A scene in a movie can be defined a

succession of individual shots that are semantically related. A scene is an important retrieval unit for users – typically users want to search/browse movies based on particular scenes corresponding to e.g. a car chase, a dialogue, a musical montage[1], etc.

There have been many different approaches to scene change detection, which is an important step in scene classification. One approach is to build a *model* of the scene [1], but this requires knowledge of the structure of the scene in advance. This approach has been applied successfully to detect scene changes in TV news, where it is possible to accurately define the structure of news stories (e.g. anchor person followed by report) [2]. A similar approach has also been applied to sports video analysis [3]. However, due to the unpredictable nature of movies, model-based approaches are hard to implement for scene-change detection, although they could be used for scene classification, where the prototypical structure of the scene to be classified is known in advance.

If a model of a scene cannot be generated, more generic approaches to scene detection must be used. These rely on a significant change in the audio-visual features at scene breaks. Yeung et al [4] grouped shots that were visually similar to each other, and constructed a Scene Transition Graph (STG) to map the temporal structure of scenes. Kender et al [5] and Yeo et al [6] use a memory-based approach to scene change detection that measures the visual distance between previous shots and the current shot. Huang et al [7] and Sundaram et al [8] used a combination of both video and audio to assist in the determination of scene breaks. This approach is based on the idea that the audio should change as well as the video in any scene change, so both visual and aural shifts are analysed in order to determine scene changes.

In this paper, we target the detection of dialogue scenes based only on visual features – corresponding to shot frequency and camera motion estimation. Typical approaches to dialogue detection involve looking for the inherent structures that are usually present in dialogue sequences [9, 10]. For example, in a 2-person dialogue there is usually an A-B-A-B-A-B structure of camera angles, so if it is possible to find this characteristic of a sequence of shots then this sequence may be detected as a dialogue scene. This approach can only be applied to 2-person dialogues, however – if there are more than two people present, then an extra shot is introduced which makes finding the inherent structure a complex problem. Indeed, even in a 2-person dialogue, the structure can become quite complex if a shot establishing context is introduced within the dialogue, for example. Our initial approach uses visual features only and makes use of a fundamental rule of film direction.

2 Film Syntax for Dialogue Scenes

We propose to use generally accepted film editing and directorial conventions in order to assist in the analysis of any given movie. One general rule followed by many directors is that the viewer must be able to comfortably view the action in the film [11]. For example, if there is an explosion, the camera should be placed so that there is nothing between it and the explosion to obstruct the view. The camera should also pan and zoom to follow any subsequent action arising that the viewer is required to see. For a dialogue sequence, the camera must remain fixed on the focus of interest (either the person talking, or one of the people he/she is talking to).

[1]This is a rapid succession of very different shots whose meaning usually only becomes apparent when they are viewed as a whole.

Another widely used convention in direction is the concept of a 180° line [12]. This line is set up at the start of the scene, and is typically followed for the remainder of the scene so that viewers can follow the action. Generally, it means that cameras must remain on the same side of the characters. This is illustrated in Figure 1. The 180° line is also used when following a character. If a character begins walking from left to right in the cameras view, then as long as he/she keeps walking in the same direction, he/she must be seen to walk from left to right in the camera. If the camera switches to the other side of the line, viewers will think that the character has switched direction. This means that there will be significant repetition of shots in a dialogue scene, since the camera on the characters will generally remain in the same position for all shots of a character.

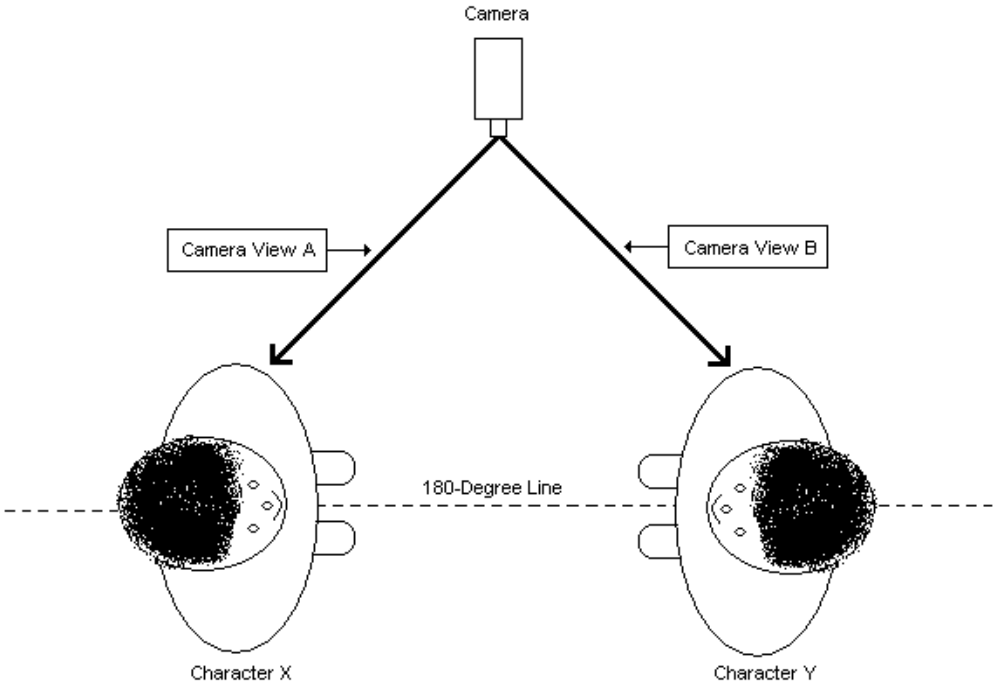

Figure 1: The 180 line

3 Dialogue Detection

3.1.1 System Overview

In order to detect dialogue sequences, we use a three-tier system as shown in Figure 2. The three levels are labelled as low-level, mid-level and high-level according to the level of semantics associated with the features used in each level. The low-level blocks in this system accept raw visual data and process it to produce an output that this is typically not of interest to the user, but that may be useful for subsequent analysis. Mid-level blocks combine the output of the low-level blocks to extract information that may be interesting to the user, but is not yet at the level of a retrieval unit. Finally, mid-level features are combined to extract the retrieval unit – in this case a dialogue scene.

3.1.2 Shot boundary detection

Determining the shot boundaries is a key essential step prior to performing shot-level feature extraction and any subsequent scene-level analysis. An individual camera shot is the atomic unit of our scene analysis engine. To this end, a histogram-based shot boundary detection approach is used in order to detect boundaries and extract keyframes [13].

3.1.3 Motion extraction

The Motion feature extraction block generates statistics about the motion present within each shot in the video sequence. Using motion vectors extracted directly from an MPEG-1 compressed bitstream, it calculates the total length of motion vectors in the frame, the percentage of non-zero blocks in the frame, and the length of short, mid and long runs of zero values in the frame [14].

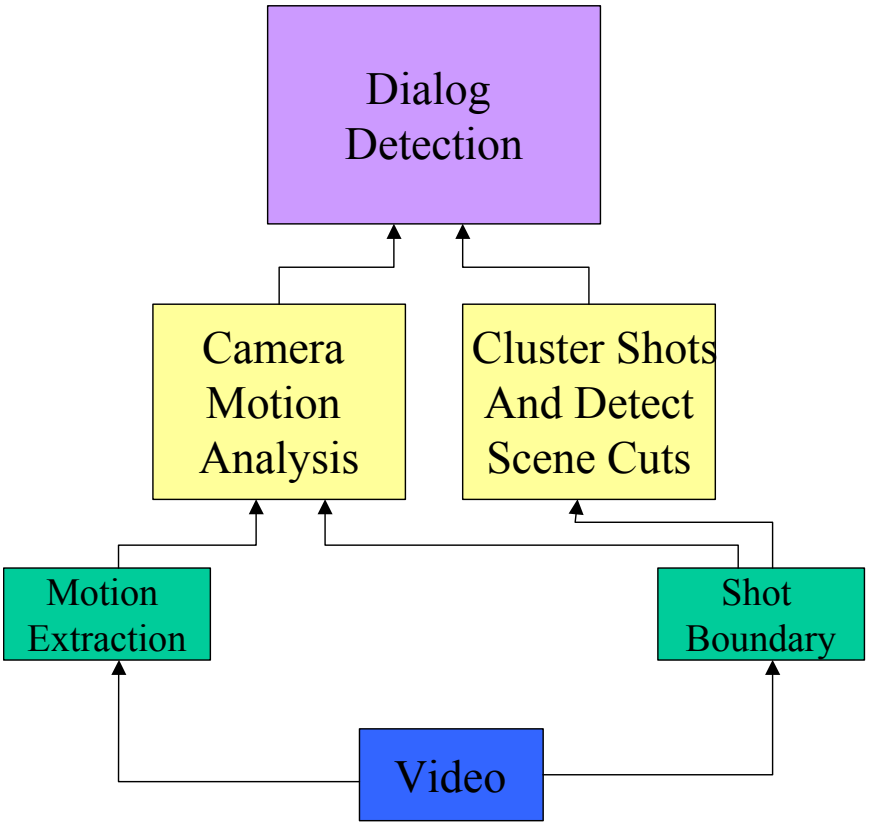

Figure 2. Dialogue detection system

3.1.4 Shot clustering

The Shot Clustering block clusters visually similar shots that are temporally close together. This method is based on the method used by Yeung et al [4]. The distance between the clusters is calculated based on the average color histogram of keyframes in the cluster.

Shot Clustering Algorithm:

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

The time-constraint ensures that only shots that are less than 2000 frames (just over a minute) apart can be merged. A single scene should contain a series of related clusters (i.e. a number of groups of similar shots), so in order to detect scene cuts, we calculate the temporal location when one set of clusters finishes and another starts.

For illustration, consider a scene involving a conversation between three people (see figure 3). This scene will contain a number of clusters, e.g. four (one for each person, and a background shot), that will be intermeshed in time. If cluster A, B and C correspond to shots of the three people in the conversation, and cluster D corresponds to the background or context setting shot, then the sequence of clusters may look like:

A-B-A-C-D-B-D-C-A-B-D-C-D-E-F-E-G-F

From this it can be concluded that there is a new scene at 'E' since this is unrelated to any previous clusters i.e. the *earliest* shot of cluster E (14) occurs later than the *latest* shot in any previous cluster.

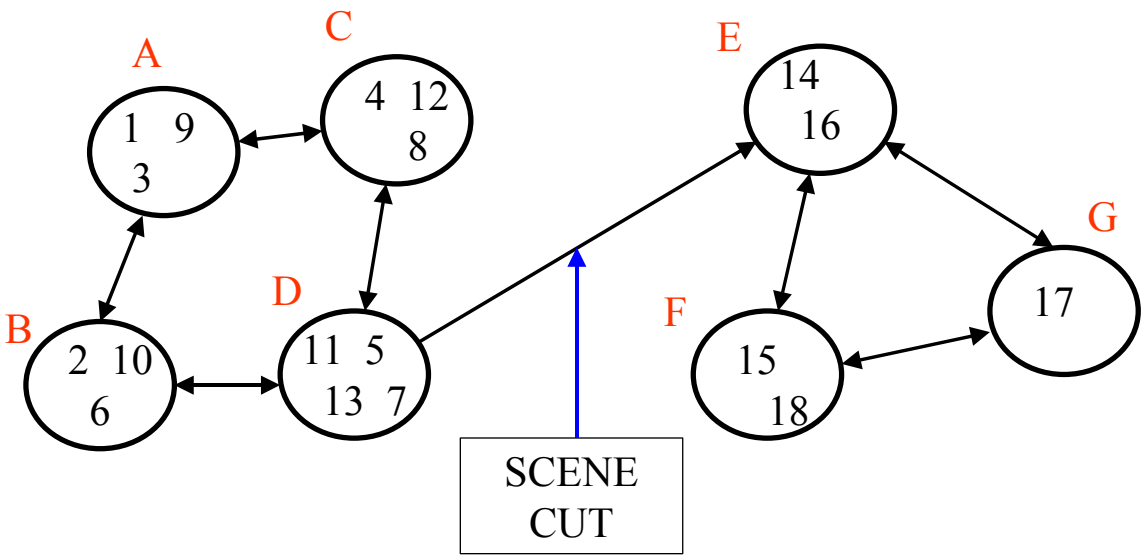

Figure 3: Clustering and Scene Change Detection

3.1.5 Camera Motion Analysis

Motion statistics, as well as the shot cut information, are input to the Camera motion analysis block. This uses the motion statistics to determine whether or not there is significant motion present in a shot. Generally if there is camera motion, there should be very few runs of zero motion vectors since

the whole frame will be moving. A 'Shot with motion' is defined as a shot in which more than 20% of the P-Frames contain camera motion.

3.1.6 Dialogue Detection

The Dialogue detection block accepts the output from both mid-level blocks and makes the final classification. It uses a two pass method. First of all potential dialogue sequences are identified based on the camera activity, and then these sequences are either accepted or rejected based on the clustering results. Potential Dialogue Sequences (PDS) are chosen solely from the output of the camera motion analysis block.

One of the main jobs of the director is to ensure that viewers can comfortably see the events on screen. Thus for a dialogue sequence, the camera will mostly remain static. Thus, the dialogue detection block looks for sequences that contain predominantly static shots. It uses a state machine to determine the start and end points of these sequences.

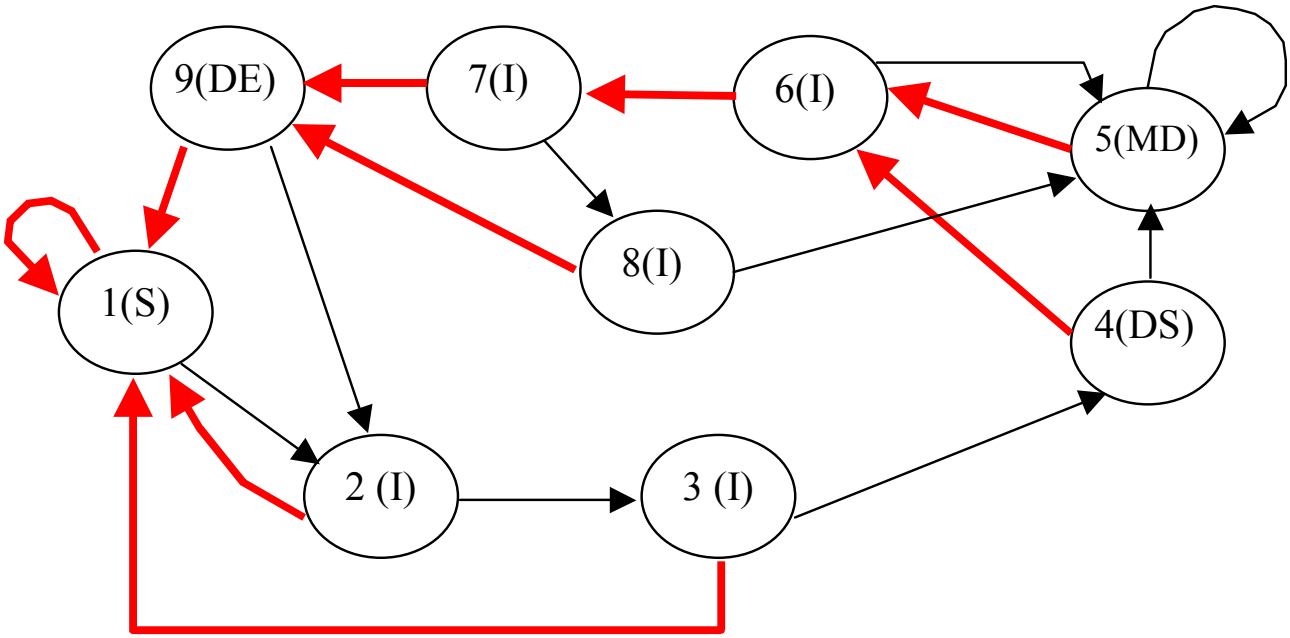

Figure 4: State Machine for Dialogue Detector

Symbol	Full Name
S	Start
I	Intermediate
DS	Dialogue Start
MD	Maintain Dialogue
DE	Dialogue End

Table 1: State Machine Symbols

If the state machine encounters a number of static shots in a row, then it declares this to be the start of a PDS. As long as there are static shots in the sequence, the state machine will remain in a PDS state. It will then take a number of non-static shots in order for the state machine to declare an end to the PDS. This does not necessarily mean that the non-static shots have to be consecutive, just that the non-static shots have to begin to dominate over the static ones. The state machine is built so that it will accept a certain amount of non-static shots before declaring an end to the PDS. This is so that isolated non-static shots within a dialogue sequence (i.e. a shot tracking a character walking across a room in the middle of a conversation) do not terminate the PDS. The state machine is shown in figure 4. The thin black arrows correspond to the action each state takes when the input is a static shot, while the thicker red arrows show the action for a non-static shot input. For a succession of static shots, the state machine will end up in state 5 (via state 4) i.e. dialogue started. For successive non-static shots the state machine will end up in state 1 (via state 9 if a dialogue was previously taking place, directly from state 2 or 3 otherwise) i.e. dialogue sequence finished, or no dialogue. The intermediate states are states in which the state machine has yet to decide whether a conversation has started, is maintained, or is finished. For example the intermediate states 6,7 and 8 are there to allow a certain amount of shots with motion within a conversation. If we are in state 5 (dialogue), and we encounter a single shot with motion followed by static shots, then the state machine goes into state 6, but then back to state 5 where it stays as long as the shots are static. One way of terminating a dialogue sequence would be if the input were three non-static shots in a row. Thus the state machine would go from state 5 (MD) to state 6(I) to state 7(I) to state 9(DE) where the dialogue would be declared finished.

After all PDSs have been detected, a further decision is made as to whether they are actual dialogue sequences or other sequences that happen to contain static shots. In order to make this decision, we calculate the ratio of clusters to shots, termed the C:S ratio, in the PDS. A low C:S ratio means that there is a number of repetitive shots in the sequence, which is consistent with a dialogue sequence. A high C:S ratio means that the shots are visually unrelated and therefore not likely to be part of a dialogue sequence. The number of clusters is simply the number of clusters that have shots within the PDS. An empirically chosen maximum C:S ratio of 0.67 is allowed for dialogue sequences. If the ratio is higher, then the PDS is rejected as a dialogue sequence, if the ratio is lower, then the PDS is marked as a Correct Dialogue Sequence (CDS).

4 Results/Experiments

The system was tested on a number of movies with the results presented in Table 2. In order to generate a ground truth the dialogue scenes were manually identified using the guiding principle that a dialogue sequence corresponds to: ‘A sequence of five or more shots containing at least two people conversing, where the main focus of the sequence is the conversation’. In other words, there needs to be significant interaction between protagonists as appropriate to short conversations (e.g. a passing ‘hello’ between two characters does not qualify). Also, people conversing in the middle of a car chase would not count, as the main focus of the sequence is the car chase. The accuracy of the detection system was measured against the manually marked-up dialogue sequences. If the CDS

corresponds with the marked-up start and end shots then the sequence was deemed to have been detected. Recall was calculated as the number of manually marked-up dialogues divided by the number of correctly detected dialogues, precision as the number of correctly detected dialogues divided by the total number of detected dialogues.

The system was tested on six movies. In choosing movies, we attempted to represent as wide a range of genres as possible, as well as varying directorial styles. The first film, "American Beauty", is a drama featuring a significant amount of dialogue. This resulted in very high precision and recall. Only two conversations were missed, one due to characters moving while conversing, and the other due to a C:S ratio above the manual threshold. The second film, "Dumb and Dumber" is a comedy in which the two central characters travel across America, thus a lot of the dialogue sequences take place inside a moving car, which is deemed to contain camera movement and so these are usually missed by the detector. Despite this, 72% of the conversations are still detected. In the film "Shaft", which is an action/crime film, only two of the conversations are missed – one due to a C:S ratio above the threshold, and the other due to motion. The film "High Fidelity" (drama) again achieves high recall. The missed conversations are typically due to movement of the characters.

The final two films are particularly challenging as the director of both of them uses original and extremely innovative direction techniques throughout the films. Generally his dialogue sequences contain longer, drawn out shots of the conversation taking place. Nevertheless, in "Reservoir Dogs" (thriller) 88% of the conversations are detected and only two are missed. The first missed one occurs while one character is dancing around the room, and is thus missed due to motion, and the second one is missed due to a high C:S ratio. Finally, in "Kill Bill: Volume 1", (action), two conversations are missed. Both are due to high C:S ratio as the director chose to shoot the conversation from varying angles. Note in this movie, the precision is unusually low; this is largely due to short conversations (10-20 seconds) being detected in the middle of non-conversation scenes, i.e. in the middle of a fight two characters talk to each other for a moment.

	Number of Dialogues	***Number Detected***	***Number Missed***	***Precision(%)***	***Recall(%)***
American Beauty	*48*	*46*	*2*	*87*	***96***
Dumb and Dumber	*32*	*23*	*9*	*87*	***72***
Shaft	*18*	*16*	*2*	*76*	***89***
High Fidelity	*44*	*38*	*6*	*75*	***86***
Reservoir Dogs	*16*	*14*	*2*	*94*	***88***
Kill Bill (V1)	*13*	*11*	*2*	*48*	***85***
Total	***171***	***148***	***23***	***77.8%***	***86%***

Table 2. Results of dialogue detector.

The average precision of 77.8%, and average recall of 86% indicate good initial performance. However, we expect this to be improved when other forms of analysis are applied to the video (see conclusions). Clearly, the manually chosen C:S-Ratio threshold adversely affects the performance of the initial version of the system. In the future we plan to address this by investigating approaches to automatically choosing an appropriate threshold. In addition, the dependence on the threshold will be alleviated when the classifier is superseded when new features are available (see conclusions).

5 Conclusion and future work

Although preliminary results are encouraging, more work is required in order to determine the exact start and end points of the conversations taking place. This could be achieved by further analysing the structure of the shots within the clusters.

We plan to add extra mid-level features to the system in order to generate more information for the high-level dialogue detection block. This will allow the high-level decision process to be as informed as possible. Face detection methods will be employed, as usually dialogue sequences would contain shots of the people conversing. Clearly, audio analysis targeting features such as speech/music/silence classification as well as significant audio event detection would be a valuable addition to the system. All of these extra inputs will create a bottleneck of information leading into the dialogue detection block, so more sophisticated machine learning methods will be implemented to relieve this bottleneck.

Finally, in the future we aim to target different media other than movies. Fictional television programmes generally follow by the same directing and editing rules, so they are an obvious target for future versions of our system. Due to the nature of our system, only the high level classifier needs to be adjusted in order to target a different media.

6 Acknowledgment

The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged. This material is based upon work supported by the IST programme of the EU in the project IST-2000-32795 SCHEMA.

7 References

[1] Zhang, H.J., Tan, S.Y., Smoliar, S.W., and Hong, G.Y., “Automatic parsing and indexing of news video”, *Multimedia Systems 1995.*

[2] O'Hare N, Smeaton A, Czirjek C, O'Connor N, and Murphy N. A generic news story segmentation system and its evaluation. *ICASSP 2004 - IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Quebec, Canada, 17-21 May 2004

[3] Sadlier D, O'Connor N, Marlow S, and Murphy N. A Combined Audio-Visual Contribution to Event Detection in Field Sports Broadcast Video. Case Study: Gaelic Football *ISSPIT'03 - IEEE International Symposium on Signal Processing and Information Technology*, Darmstadt, Germany, 14-17 December 2003.

[4] Yeung, M., and Yeo, B.-L., "Video visualization for compact presentation and fast browsing of pictorial content", *IEEE Trans. Circuits Syst. Video Technol. 7*, 5 (Oct. 1997), 771–785

[5] Kender, John R., Yeo, Boon-Lock, "Video Scene Segmentation Via Continuous Video Coherence", Proc. CVPR '98, pp 367-373, June 1998.

[6] Yeo, B.-L., and Liu, B., "Rapid scene analysis on compressed videos", *IEEE Trans. Circuits Syst. Video Technol. 5*, 6 (Dec. 1995), 533–544.

[7] Huang, Jincheng; Liu, Zhu; Wang, Yao, "Integration of audio and visual information for content-based video segmentation", IEEE Int'l Conf. Image Processing (ICIP98), Special SessioSession on "Content-Based Video Search and Retrieval". Oct. 1998. Chicago.

[8] Sundaram, Hari; and Chang, Shih-Fu. "Determining Computable Scenes in Films and their Structures using Audio-Visual Memory Models", ACM Multimedia 2000, Oct 30 - Nov 3, Los Angeles, CA.

[9] Hari Sundaram and Shih-Fu Chang - Condensing computable scenes using visual complexity and film syntax analysis. IEEE Conference on Multimedia and Exhibition, Tokyo, Japan, Aug. 22-25, 2001

[10] Video content analysis using multimodal information. Kluwer Academic Publishers 2003.

[11] Directing. Michael Rabiger. Focal press 1997.

[12] David Bordwell, Kristin Thompson. Film Art: An Introduction. McGraw-Hill.

[13] Browne, Paul; Smeaton, Alan F.; Murphy, N; O`Connor, N; Marlow, S; Berrut, C; Evaluation and Combining Digital Video Shot Boundary Detection Algorithms. *IMVIP 2000 - Irish Machine Vision and Image Processing Conference*, Belfast, Northern Ireland, 31 August - 2 September 2000.

[14] Xinding Sun, Ajay Divakaran, B.S. Manjunath, "A motion activity descriptor and its extraction in compressed domain". IEEE Pacific-Rim Conf. Multimedia (PCM), pp. 450-453, October 2001.