

# Semantic classification of movie scenes using finite state machines

Y. Zhai, Z. Rasheed and M. Shah

**Abstract:** The problem of classifying scenes from feature films into semantic categories is addressed and a robust framework for this problem is proposed. It is proposed that the finite state machines (FSM) are suitable for detecting and classifying scenes and their usage is demonstrated for three types of movie scenes: conversation, suspense and action. This framework utilises the structural information of the scenes together with the low-level and mid-level features. Low level features of the video including motion and audio energy and a mid-level feature, body, are used in this approach. The transitions of the FSMs are determined by the features from each shot in the scene. The FSMs have been experimented on over 80 clips and convincing results have been achieved.

## 1 Introduction

Recent years have seen a growing interest in the annotation and retrieval of video data. The increasing number of subscribers to digital cable now demands efficient tools so that viewers can browse and search sections of interest of video. Among many genres of video production, feature films are a vital field for the application of such tools. It is a sizeable element of the entertainment industry, easily available, widely watched and therefore, is becoming the focus of researchers in many aspects. For example, applications for content-based video annotation and retrieval have been developed at all levels of the video structure: shot level, scene level, and movie level. A shot is a sequence of images that preserve consistent background settings. It is the basic element of a movie. A scene, which consists of a set of continuous shots, constitutes a portion of the storyline. On the highest level, a movie is composed of a series of related scenes defining a theme. For a user, who may be looking for a particular scene of a feature film, a shot level analysis is insufficient since a shot level analysis fails to capture the semantics of the video content. For example, how does one answer a query for a suspense scene in a feature film based on a single shot content? Any semantic category like suspense or tragedy, cannot be defined over a single shot. These concepts are induced in viewers over time. Indeed, a meaningful result can only be achieved by exploiting the interconnections of shot content.

In this paper, we present a novel framework for classifying scenes, focusing on feature films, into three semantic categories: conversation, suspense and action.

This method analyses the structural information of the scenes based on the low-level and mid-level shot features which are robust and easily computable. The low-level features used in our framework include shot motion and audio energy, and the mid-level feature is body identity based on face detection. To bridge the gap between the low and mid-level features and a high-level semantic category, finite state machines are studied and developed. The transitions are determined based on the statistics of these features for each shot. This paper is organised as follows: related work is discussed in Section 2, Section 3 describes the classification framework, including the features and the finite state machines for detecting conversation, suspense and action scenes. Section 4 shows the experimental results and Section 5 concludes our work.

## 2 Related work

In the area of higher level scene understanding, Adams *et al.* [1] proposed the detection of ‘tempo’ in movies. The camera motion magnitude and the shot length were the two features used to compute a continuous function. Our framework, however, analyses the structure of the movie scene and classify scenes into more specific categories. Sundaram *et al.* [2] used the audio-visual features of the video in the movie scene segmentation. First, two types of scenes, audio scene and video scene, are detected separately. The correspondences between these two types of scenes are then determined using a time-constrained nearest-neighbour algorithm. Yeung *et al.* [3] were among the first to propose a graph-based representation of the video data by constructing a shot connectivity graph. The graph is split into several sub-portions using the complete-link method of hierarchical clustering such that each sub-graph satisfies a colour similarity constraint. Yoshitaka *et al.* [4] also used shot length and visual dynamics to analyse scene type. In their approach, the colour statistics of the frames in the shot were used to calculate the visual dynamics and the similarities between the repeating shots were exploited. Experiments on only one kind of scene were demonstrated and it was not clear how the approach could be extended to other scene categories.

Lienhart *et al.* [5] used face detection in the scenes to link similar shots. A ‘face-based class’ with a group of related frames showing the same actor was constructed by the similarity of the spatial positions and sizes of the detected faces. These ‘face-based classes’ were linked across shots in the video to form the ‘face-based sets’ by using Eigenfaces. The pattern of a dialogue scene was flagged if several conditions were satisfied. In their experiment, face recognition suffered accuracy and the system typically split the same actor into different sets causing over-detection. Liu *et al.* [6] proposed a scene classification framework on TV programmes using hidden Markov models (HMM). Audio information is used as the feature, and the TV programmes are classified into five categories: news report with anchors, weather reports, TV commercials, live basketball games and live football games. Li *et al.* [7], exploited the global structural information of a scene and built ‘shot sinks’ to classify a scene into one of three scenarios including ‘two-speaker dialog’, ‘multi-speaker dialog’, and ‘others’. The overall structure was computed based on the low-level visual features, such as colour of the shots in the scene. In their approach face information, which is an important cue for speaker detection, was not used. We combine both structure and face detection in a finite state machine framework to provide a more general solution for the scene classification task.

### 3 Proposed approach

In this Section, we discuss the extraction of the low-level and mid-level features used in our approach. The activity intensity, which is a function of low-level features, is one of the inputs to the finite state machines. Motion intensity and audio energy are used to compute the activity intensity. Another input, body identity, is extracted from the face detection process. We construct FSMs for three different semantic categories of scenes. These include conversational, suspense and action.

#### 3.1 Activity intensity ( $\Gamma$ )

Motion in the videos has been used by several researchers in detecting and identifying scenes in feature films. Some examples are [1, 8]. In feature films, the camera motion is generally translation, pan and zoom, whereas camera roll and tilt are rare. Furthermore, the play-rate for the movies is relatively high (usually shot at 24 fps), causing change between consecutive frames to be small. Therefore, affine motion model is suitable for capturing the frame-to-frame

global transformation in the video. We exploit the motion vector information embedded in the MPEG compressed video. The approximate motion model is computed based on the  $16 \times 16$  pixel macro-blocks. Two consecutive images and their motion field are shown in Fig. 1. For each macro-block  $[x \ y]^T$ , its motion vector  $[u \ v]^T$  is computed as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

or  $U = AX$ , where  $[b_1 \ b_2]^T$  vector captures the global translation and  $[a_1 \ a_2; a_3 \ a_4]$  interpret the scaling, zooming and sheering. The motion intensity can be formulated by the degree of how well the affine model fits to the consecutive images. Let  $\mu$  denote the motion intensity of an image

$$\mu = \text{mean}(\text{mag}(X - X')) \quad (2)$$

where  $X$  is the original motion field and  $X'$  is the re-projected motion field by applying  $A$  to the image grid. We take the average magnitude of the difference as the motion intensity. This can be thought as the ‘disagreement’ of the motion field to the affine model.

Sound also plays an important role in distinguishing scenes from each other. In conversational scenes, characters speak smoothly and calmly. Alternatively, in action scenes, which often include explosions, collisions, or vehicle chases, the audio energy is very high. Figures 2a and 2b show the plots of audio signals for a conversation scene and an action scene, respectively. Note that the high energy in the audio of the action scene is distinctive from that of the conversational scene. Therefore, the computation of activity intensity also incorporates the mean audio energy  $\theta$ . The overall activity intensity is the combination of two quantities,  $\mu$  and  $\theta$

$$\Gamma = w_\mu \times \mu + w_\theta \times \theta \quad (3)$$

where  $w_\mu$  and  $w_\theta$  are the weights to balance the effects from motion and audio features. From the empirical observation, the magnitude level of the absolute values of motion information usually is around 15 times of the ones for audio energy. Therefore, to achieve the equal importance of these two feature, we set  $w_\mu = 1/16$  and  $w_\theta = 15/16$ .

Figure 3 shows the plots of the activity intensity values for three types of movie shots: conversation, action and suspense.

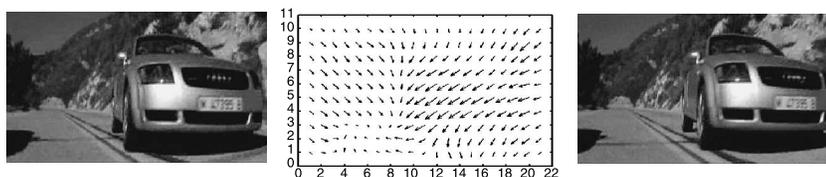


Fig. 1 Two images from movie ‘Mission Impossible 2’ and their motion field

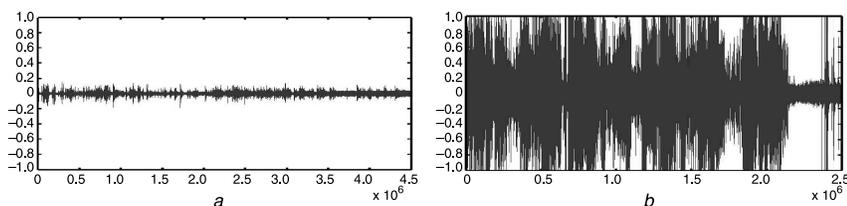
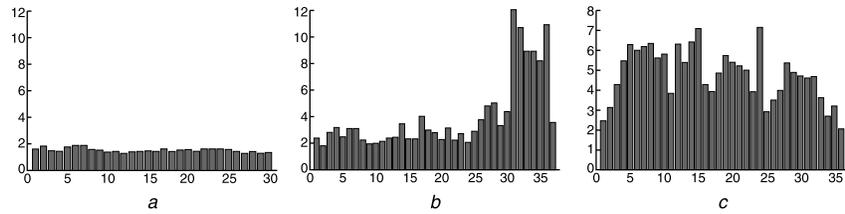


Fig. 2 Plots of audio signals

a Conversation scene  
b Action scene



**Fig. 3** Plots of activity intensity,  $\lambda$ , for three example scenes

a Conversation  
b Suspense  
c Action

The horizontal axis represents the shot number in the scene, while the vertical axis represents the  $\lambda$ .



**Fig. 4** Key-frames with 'bodies' of one example scene from movie 'The Others'

### 3.2 Body identity

Conversational scenes generally have shots where at least two characters are talking. We utilise this cue and detect human faces in the video using the method proposed by Viola *et al.* [9]. We have found that [9] performs well for faces with different scales in the video. The shots with faces detected are clustered into groups, each of which corresponds to a character. Generally, there are two approaches: (a) cluster the shots based on the similarity computed from the global feature of the image, and (b) cluster the shots based on the face correlations. For case (a), the clustering fails if the same character is in different background settings, while for case (b), common face correlation without training creates over-detections.

To overcome these problems, we used the 'body' regions to compute the similarity between shots. The middle frame of each shot  $i$  is selected as the key frame  $k_i$  of that shot. The face detection program [9] then is applied to the key frame  $k_i$ . Owing to the various image sizes of different movies, the detection program is performed on four levels of scales to the original image size, 50%, 100%, 200% and 400%. To extract the 'body' regions, the bounding boxes of the detected faces are extended downward to cover the upper part of the character (Fig. 4). There might be multiple 'bodies' in a single key-frame. Therefore, the body regions in one shot are denoted as a set  $F_i = \{f_i^1, \dots, f_i^n\}$ . For every body patch, a colour histogram in RGB channels is computed. The similarity  $S(i, j)$  between two facial shots  $i$  and  $j$  is the similarity between the sets  $F_i$  and  $F_j$

$$S(i, j) = \max(\text{sim}(f_i^m, f_j^n)) \quad (4)$$

where  $m = 1, 2, \dots, n_i, n = 1, 2, \dots, n_j$ , and  $\text{sim}(f_i^m, f_j^n)$  is the histogram intersection between the image patches for the bodies  $f_i^m$  and  $f_j^n$ . The shots then are clustered based on the body similarity by using K-means algorithm.

### 3.3 Finite state machines (FSM)

A finite state machine is defined as

$$A = (Q, \Sigma, \sigma, q_0, F) \quad (5)$$

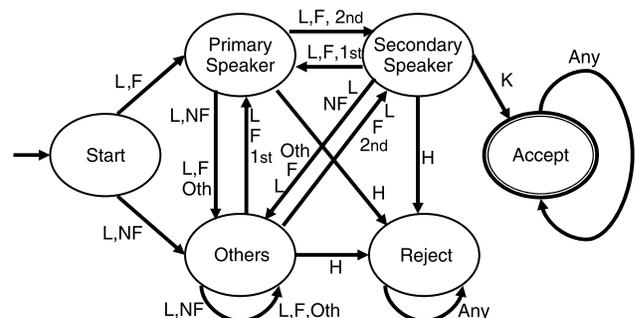
where  $Q$  is a set of states in the FSM and  $\sigma$  is the set of transitions.  $\Sigma$  contains the conditions for the transitions.  $q_0$  is the initial state, and  $F$  is the set of accepting (final) states. In feature films, scenes are generally composed in

accordance with the conventional film grammar. We have observed the following characteristics for three different categories of scenes:

- (i) conversational scenes: low activity intensity, medium audio energy and multiple speakers,
- (ii) suspense scenes: a long period of silence followed by a sudden eruption either in audio track or in activity intensity or both, and
- (iii) action scenes: intensive action activity for a certain number of shots.

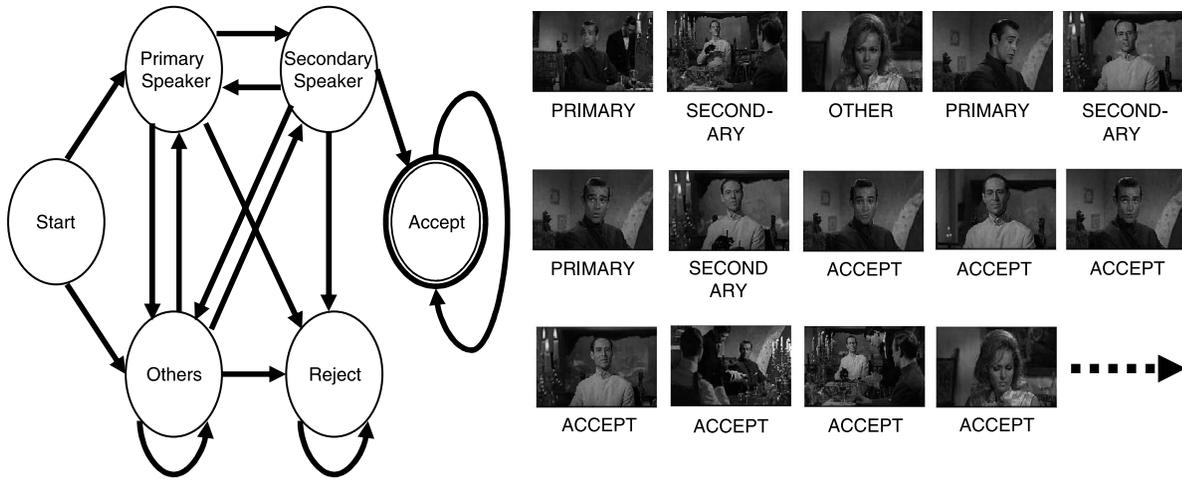
We discuss three different FSMs which detect conversational, suspense and action scenes.

**3.3.1 FSM for conversation scenes:** In conversation scenes, we often see multiple characters speaking in a switching fashion. Based on this pattern, we construct an FSM to control the number of speakers and the times of their appearance. In our system, the accepting condition is there are at least two main speakers, each of whom appears more than three times in the scene. Figure 5 shows a deterministic finite state machine for detecting conversation scenes. The FSM consists of six states: *Start*, *Primary Speaker*, *Secondary Speaker*, *Others*, *Reject* and *Accept*. Shots with high similarity between bodies are clustered together. The state *Primary Speaker* is represented by the largest cluster, and the *Secondary Speaker* is represented by the second largest cluster. The transitions are determined



**Fig. 5** Finite state machine for conversation scene detection

**L** - low activity intensity; **H** - high activity intensity; **F** - facial shot; **NF** - non-facial shot; **1st, 2nd, Oth** - speaker clusters; **K** acceptance condition satisfied; **Any** - any shot



**Fig. 6** Demonstration for dialogue FSM  
It shows the key-frames in the scene and their corresponding states

**Table 1: Transition matrix for conversation detection. Columns represent 'From' states, rows represent 'To' states and '-' indicates no transition from one state to another**

$\sigma$	Start	Primary	Secondary	Others	Reject	Accept
Start	-	L, F	-	L, NF	-	-
Primary	-	-	L, F, 2nd	L, NF/L, F, Oth	H	-
Secondary	-	L, F, 1st	-	L, NF/L, F, Oth	H	K
Others	-	L, F, 1st	L, F, 2nd	L, NF/L, F, Oth	H	-
Reject	-	-	-	-	Any	-
Accept	-	-	-	-	-	Any

based on the feature values of the shots in the scene. If the state *Accept* is reached, the scene is declared as a 'Conversation' scene. Otherwise, it is declared as 'Non-Conversation'. A short demonstration is shown in Fig. 6.

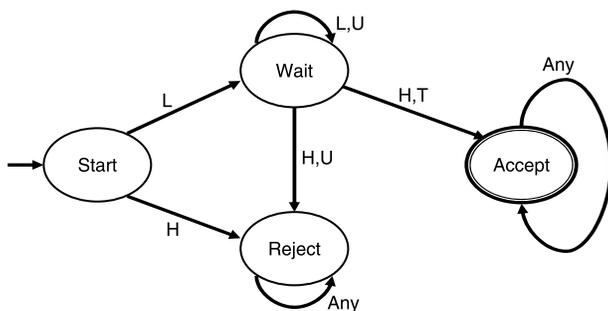
In this FSM,  $Q = \{Start, Primary\ Speaker, Secondary\ Speaker, Others, Reject, Accept\}$ ,  $q_0 = \{Start\}$  is the initial state and  $F = \{Accept\}$  is the final state. The transitions in the FSM and their transition conditions are shown in Fig. 5. The transition matrix for  $\sigma$  is shown in Table 1 (transition values explained in Fig. 5).

**3.3.2 FSM for suspense scenes:** We have observed that suspense scenes often have the following pattern. In the beginning, the scene is relatively silent and is followed by a sudden increase in audio energy. In many cases, it is also accompanied by abrupt camera and actor movements. Based on these observations, the FSM for

detecting the suspense scenes have the following four states: *Start, Wait, Reject* and *Accept*. The state *Wait* represents the pre-action moments. After a period of 'waiting' (1 minute in the experiment), the state is transferred to *Accept* if a sudden action shot is seen. The FSM rejects the scenes in which the sudden action happens before the predefined time interval.

Similarly, the definition of the FSM for the classification of suspense scenes can be written in the general formula for the finite state machines. In this case,  $Q = \{Start, Wait, Reject, Accept\}$  are the states. The initial state is  $q_0 = \{Start\}$ , and the final state is  $F = \{Accept\}$ . The FSM is shown in Fig. 7. Table 2 is the transition matrix (transition values explained in Fig. 7).

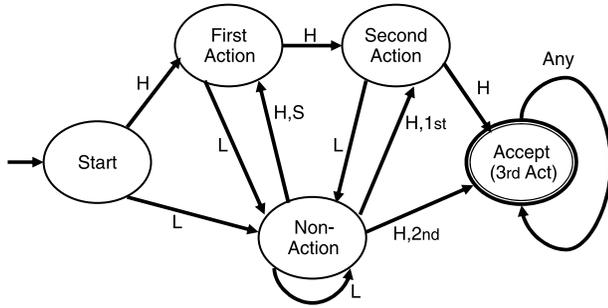
**3.3.3 FSM for action scenes:** Action scenes in movies generally have very high action intensity, such as scenes containing explosions, chasing and fighting. To classify a scene as an action scene, the scene must contain certain number of shots with action intensity higher than defined threshold level.



**Fig. 7** Finite state machine for suspense scene detection  
L - low activity intensity; H - high activity intensity; U - under the required time interval; T - over the required time interval; Any - any shot

**Table 2: Transition matrix for suspense scene detection. Columns represent 'From' states, rows represent 'To' states and '-' indicates no transition from one state to another**

$\sigma$	Start	Wait	Reject	Accept
Start	-	L	H	-
Wait	-	L, U	H, U	H, T
Reject	-	-	Any	-
Accept	-	-	-	Any



**Fig. 8** Finite state machine for action scene detection

L - low activity intensity; H - high activity intensity; S, 1st, 2nd - pre-state values to determine which transition should be taken from stat 'Non-Action'; Any - any shot

**Table 3: Transition matrix for action scene detection. Column represent 'From' states, rows represent 'To' states and '-' indicates no transition from one state to another**

$\sigma$	Start	1st-Act	2nd-Act	Non-Act	Accept
Start	-	H	-	L	-
1st-Act	-	-	H	L	-
2nd-Act	-	-	-	L	H
Non-Act	-	H, S	H, 1st	L	H, 2nd
Accept	-	-	-	-	Any

For action FSM, the state set  $Q$  has  $\{Start, First\ Action, Second\ Action, Non-Action, Accept\ (Third\ Action)\}$ , where the initial state  $q_0$  is  $\{Start\}$ , and the final state  $F$  is  $\{Accept\ (Third\ Action)\}$ . A pre-state attribute for a state  $q_i$  in the FSM is defined as the 'from' state of the immediate transition before reaching state  $q_i$ . This is used for the determination of the outgoing transitions from state *Non-Action*. The detail information of this FSM is shown in Fig. 8. The transition matrix is in Table 3 (transition values explained in Fig. 8).

#### 4 Experimental results

We have experimented with over 80 scenes using the finite state machines for three categories of scenes. These clips are taken from seven Hollywood movies, including 'The Others', 'Jurassic Park III', 'Terminator II', 'Gone in 60 Seconds', 'Mission Impossible 2', 'Dr. No', and 'Scream'. We also included a TV talk show, 'Larry King Live', and a TV news programme, 'CNN Headline News'. The feature movies cover a variety of genres such as horror, drama, and action. Each input scene contains approximately 20–30 shots. The process of detecting shot boundaries and scene boundaries is taken as a prior knowledge in advance of scene classification. We use the method described in [10] for detecting shots, and the shots are further grouped into scenes using the method proposed in [8]. Four human observers were asked to choose the most suitable label from three

categories for each scene. Each scene was given all applicable ground truth label(s) with the category that the most human observers agreed upon. Thus, each scene is considered as a positive member of the category to which it is assigned. Observers were also asked to provide the most unlikely category for each scene. We used this information to label a scene as a non-member (or a negative member) for the unlikely categories.

To evaluate the performance of the proposed approach, two measures of accuracy were computed. These measures are precision and recall and defined as follows

$$P_{pos} = \frac{M_{pos}}{D_{pos}}, \quad R_{pos} = \frac{M_{pos}}{G_{pos}} \quad (6)$$

and

$$P_{neg} = \frac{M_{neg}}{D_{neg}}, \quad R_{neg} = \frac{M_{neg}}{G_{neg}} \quad (7)$$

where  $P_{pos}, R_{pos}, P_{neg}$  and  $R_{neg}$  are the precision and recall for positive and negative member detection.  $G_{pos}$  and  $G_{neg}$  are numbers of the ground truth positive and negative members.  $D_{pos}$  and  $D_{neg}$  are the detected positive and negative members.  $M_{pos}$  and  $M_{neg}$  are the numbers of the correctly matched positive and negative members.

There were 35 conversational scenes in the data set. The results achieved were 97.1% precision and 94.3% recall. For the other 37 non-conversational scenes, the precision was 94.7%, and the recall was 97.3%. The number of positive members of the suspense category in the data set was 16, with 20 non-member scenes. The precision and recall for the member detection was 100.0% and 93.7% respectively, and the precision and recall for the non-member clips was 95.2% and 100.0% respectively. In action scenes, we had 33 member scenes and 37 non-member scenes. The precision and recall for the positive members was 91.4% and 97.0% respectively. The precision and recall for the negative members are 97.1% and 91.9% respectively. The overall performance is summarised in Table 4. These results clearly demonstrate that the finite state machine can detect and classify video scenes into categories. Figure 9 shows some scenes with the key frames of time shots.

Overall, the performance is satisfactory. The face-detector [9] is robust for the faces with full or semi-frontal views. The 'body' feature we used is able to cluster the corresponding characters with >90% accuracy. However, sometimes the character appears in side view, where he or she cannot be detected. Therefore, time current shot would be declared as a non-facial shot amid miss the clustering. Another failure scenario is that sometimes the camera is far away from the high-speed vehicle in a chase scene, such that the global motion can fit to the entire image well. In this case, the activity intensity is reflecting a relatively low value, and the current shot would be declared as non-action shot. However, this is not common in the feature films, since the producer mostly wants the focus of interest (the car in our case) to be the dominant element in the field of view.

**Table 4: Precision and recall for conversation, suspense and action scene classification**

Scene type	Conversation		Suspense		Action	
	Positive	Negative	Positive	Negative	Positive	Negative
Precision	97.1%	94.7%	100.0%	95.2%	91.4%	97.1%
Recall	94.3%	97.3%	93.7%	100.0%	97.0%	91.9%



**Fig. 9** Three testing scenes

Six representative key-frames from each are displayed

*a* Conversation scene: 007–Dr. No

*b* Suspense scene: *Scream*

*c* Action scene: *Terminator II*

## 5 Conclusions

In this paper, we have presented a novel framework for classifying video scenes into high-level semantic categories using deterministic finite state machines (FSM). The transitions in each FSM are based on the low and mid-level shot features. These features are robust and easily computable. We also incorporated face detection to cluster shots and used these clusters to determine the transitions of the FSMs. We demonstrated the usefulness of FSM for this task by experimenting on over 80 movie scenes and achieved high recall and precision scales.

There are other options for the proposed task, e.g., support vector machines (SVM) or hidden Markov models (HMM). One major difference between the proposed method, FSM, and these methods is that FSMs are designed based on the production rules, ‘grammars’, while SVMs or HMMs need to be trained without prior knowledge, which may not be available sometimes. Furthermore, SVMs or HMMs give a confidence score for how likely a certain action/activity is recognised. This is true in the action recognition scenario, since one action can be categorised with multiple classes. Alternatively, transitions in FSMs are deterministic and so is the final output. In the end of the process, a movie scene is declared to be a target category with either a ‘Yes’ or ‘No’. This is realistic to the audiences. For example, it is seldom found that people can hardly distinguish a non-dialogue

scene from a dialogue scene. Therefore, owing to the lack of training data and the ultimate goal, the FSMs have been designed in our framework.

## 6 References

- 1 Adams, B., Dorai, C., and Venkatesh, S.: ‘Novel approach to determining tempo and dramatic story sections in motion pictures’. Int. Conf. on Image Processing, 2000
- 2 Sundaram, H., and Chang, S.F.: ‘Video scene segmentation using video and audio features’. Int. Conf. on Multimedia and Expo, 2000
- 3 Yeung, M., Yeo, B., and Liu, B.: ‘Segmentation of videos by clustering and graph analysis’, *Comput. Vis. Image Underst.*, 1998, **71**, (1), pp. 94–109
- 4 Yoshitaka, A., Ishii, T., Hirakawa, M., and Ichikawa, T.: ‘Content-based retrieval of video data by the grammar of film’. IEEE Symp. on Visual Languages, 1997
- 5 Lienhart, R., Pfeiffer, S., and Effelsberg, W.: ‘Scene determination based on video and audio features’, Proc. IEEE Conf. on Multimedia Computing and Systems, Florence, Italy, 1999
- 6 Liu, Z., Huang, J., and Wang, Y.: ‘Classification of TV programs based on audio information using hidden Markov model’. IEEE Signal Processing Society Workshop on Multimedia Signal Processing, 1998
- 7 Li, Y., Narayanan, S., and Jay Kuo, C.-C.: ‘Movie content analysis indexing, and skimming’ in ‘Video mining’ (Kluwer Academic Publishers, 2003), Chap. 5
- 8 Rasheed, Z., and Shah, M.: ‘Scene detection in Hollywood movies and TV shows’. IEEE Computer Vision and Pattern Recognition Conf., Madison, Wisconsin, 16–22 June 2003
- 9 Viola, P., and Jones, M.: ‘Robust real-time object detection’, *Int. J. Comput. Vis.*, 2001
- 10 Zhai, Y., Rasheed, Z., and Shah, M.: ‘University of Central Florida at TRECVID 2003’. TREC Video Evaluation Forum (TRECVID), 2003