VIDEO CATEGORIZATION USING SEMANTICS AND SEMIOTICS

by

ZEESHAN RASHEED
B.E. NED University of Engineering and Technology, Karachi, 1998

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2003

Major Professor:
Mubarak Shah

ABSTRACT

There is a great need to automatically segment, categorize, and annotate video data, and to develop efficient tools for browsing and searching. We believe that the categorization of videos can be achieved by exploring the concepts and meanings of the videos. This task requires bridging the gap between low-level content and high-level concepts (or semantics). Once a relationship is established between the low-level computable features of the video and its semantics, the user would be able to navigate through videos through the use of concepts and ideas (for example, a user could extract only those scenes in an action film that actually contain fights) rather than sequentially browsing the whole video. However, this relationship must follow the norms of human perception and abide by the rules that are most often followed by the creators (directors) of these videos. These rules are called film grammar in video production literature. Like any natural language, this grammar has several dialects, but it has been acknowledged to be universal. Therefore, the knowledge of film grammar can be exploited effectively for the understanding of films. To interpret an idea using the grammar, we need to first understand the symbols, as in natural languages, and second, understand the rules of combination of these symbols to represent concepts. In order to develop algorithms that exploit this film grammar, it is necessary to relate the symbols of the grammar to computable video features.

In this dissertation, we have identified a set of computable features of videos and have developed methods to estimate them. A computable feature of audio-visual data is defined as any statistic of available data that can be automatically extracted using image/signal processing and computer vision techniques. These features are global in nature and are extracted using whole images, therefore, they do not require any object detection, tracking and classification. These features include video shots, shot length, shot motion content, color distribution, key-lighting, and audio energy. We use these features and exploit the knowledge of ubiquitous film grammar to solve three related problems: segmentation and categorization of talk and game shows; classification of movie genres based on the previews; and segmentation and representation of full-length Hollywood movies and sitcoms.

We have developed a method for organizing videos of talk and game shows by automatically separating the program segments from the commercials and then classifying each shot as the host's or guest's shot. In our approach, we rely primarily on information contained in shot transitions and utilize the inherent difference in the scene structure (grammar) of commercials and talk shows. A data structure called a shot connectivity graph is constructed, which links shots over time using temporal proximity and color similarity constraints. Analysis of the shot connectivity graph helps us to separate commercials from program segments. This is done by first detecting stories, and then assigning a weight to each story based on its likelihood of being a commercial or a program segment. We further analyze stories to distinguish shots of the hosts from those of the guests. We have performed extensive experiments on eight full-length talk shows (e.g. Larry King Live, Meet the Press, News Night) and game shows (Who Wants To Be A Millionaire), and have obtained excellent classification with 96% recall and 99% precision. http://www.cs.ucf.edu/∼vision/projects/LarryKing/LarryKing.html

Secondly, we have developed a novel method for genre classification of films using film previews. In our approach, we classify previews into four broad categories: comedies, action, dramas or horror films. Computable video features are combined in a framework with cinematic principles to provide a mapping to these four high-level semantic classes. We have developed two methods for genre classification; (a) a hierarchical method and (b) an unsupervised classification method. In the hierarchical method, we first classify movies into action and non-action categories based on the average shot length and motion content in the previews. Next, non-action movies are sub-classified into comedy, horror or drama categories by examining their lighting key. Finally, action movies are ranked on the basis of number of explosions/gunfire events. In the unsupervised method for classifying movies, a mean shift classifier is used to discover the structure of the mapping between the computable features and each film genre. We have conducted extensive experiments on over a hundred film previews and demonstrated that low-level features can be efficiently utilized for movie classification. We achieved about 87% successful classification. http://www.cs.ucf.edu/~vision/projects/movieClassification/movieClassification.html

Finally, we have addressed the problem of detecting scene boundaries in full-length feature movies. We have developed two novel approaches to automatically find scenes in the videos. Our first approach is a two-pass algorithm. In the first pass, shots are clustered by computing backward shot coherence; a shot color similarity measure that detects potential scene boundaries (PSBs) in the videos. In the second pass we compute scene dynamics for each scene as a function of shot length and the motion content in the potential scenes. In this pass, a scene-merging criterion is used to remove weak PSBs in order to reduce over-segmentation. In our second approach, we cluster shots into scenes by transforming this task into a graph-partitioning problem. This is achieved by constructing a weighted undirected graph called a shot similarity graph (SSG), where each node represents a shot and the edges between the shots are weighted by their similarities (color and motion). The SSG is then split into sub-graphs by applying the normalized cut technique for graph partitioning. The partitions obtained represent individual scenes in the video. We further extend the framework to automatically detect the best representative key frames of identified scenes. With this approach, we are able to obtain a compact representation of huge videos in a small number of key frames. We have performed experiments on five Hollywood films (Terminator II, Top Gun, Gone In 60 Seconds, Golden Eye, and A Beautiful Mind) and one TV sitcom (Seinfeld) that demonstrate the effectiveness of our approach. We achieved about 80% recall and 63% precision in our experiments. http://www.cs.ucf.edu/~vision/projects/sceneSeg/sceneSeg.html

This work is dedicated to my parents; any virtue in me is due to them...

ACKNOWLEDGMENTS

I would like to acknowledge my advisor, Dr. Mubarak Shah for his continued guidance throughout my endeavor. Several times when I lost my focus, he helped me reaffirm my commitments. His advice and suggestions made this dissertation possible. I would like to thank my friend Omar Javed as well. Many of our discussions resulted in innovative ideas; working with him has been both fruitful and enjoyable. Lisa Spencer and Yaser Sheikh have been of great help in compiling my work. Their useful remarks made this manuscript pleasurable to read. I would also like to thank the entire Vision Lab for being supportive throughout my stay.

Last but not least, I must thank Sohaib Khan who has been an inspiration and motivation from my first day at UCF.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

SEMANTIC INTERPRETATION OF PRODUCED VIDEOS

1.1 INTRODUCTION

There is a staggering amount of multimedia data available today and new documents, presentations, home videos, motion pictures and television programs augment this ever-expanding pool of information daily. Recently, the Berkeley "How Much Information?" project ([37]) found that 4,500 motion pictures are produced annually, amounting to almost 9,000 hours or half a terabyte of data *every year*. They further found that 33,000 television stations broadcast for twenty-four hours a day and produce eight million hours per year, amounting to 24,000 terabytes of data! With digital technology becoming inexpensive and popular, there has been a tremendous increase in the availability of this audio-visual information through cable and the Internet. In particular, services such as *video on demand* allow the end users to interactively search for content of interest. However, to be useful, such a service requires an intuitive organization of the available data. Although some data is labelled at production time, an enormous portion remains unindexed. Furthermore, the label provided may not contain sufficient context for locating the data of interest in a large database. For practical access to such huge amounts of data, there is a great need to organize and develop efficient tools for browsing and retrieving contents of interest. Annotation of audio-video sequences is also required so that users can quickly locate clips of interest without having to go through entire databases. With appropriate indexing, the user can be provided with a superior method of extracting relevant content and can therefore effectively navigate through large amounts of data. On the other hand, sequentially browsing for a specific section of video is very time consuming and frustrating. Thus, there is great incentive for developing automated techniques for indexing and organizing audio-visual data.

Digital video is a rich medium compared to text material. It is usually accompanied by other information sources, such as speech, music and closed captions. Therefore, it is important to fuse this heterogenous information intelligently to fulfill the users' search queries. Conventionally, the data is often indexed and retrieved by directly matching homogeneous types of data. Multimedia data, however, also contains important information *between* heterogenous types of data, such as video and sound, a fact confirmed through human experience. We often observe that a scene may not evoke the same response, like horror or sympathy, if the accompanying sound is muted. Conventional methods fail to utilize these relationships since heterogenous data types cannot be compared directly. Thus, there is a need to develop sophisticated techniques to fully utilize the rich sources of information contained in multimedia data.

1.2 OUR APPROACH

We believe that the categorization of videos can be achieved by exploring the concepts and meanings of the videos. This task requires bridging the gap between low-level contents and high-level concepts. Once a relationship is developed between the computable features of the video and its semantics, the user would be allowed to navigate through videos by ideas instead of the rigid approach of content matching. However, this relationship must follow the norms of human perception and abide by the rules that are most often followed by the creators (directors) of these videos. Like any natural language, this grammar has several dialects, but is fortunately more or less universal. For example, most television game shows share a common pattern of transitions among the shots of host and guests, governed by the grammar of the show. Similarly, a different set of rules may be used to film dialogue between two actors than an action scene in a feature movie. This fact in film-making (as compared to arbitrary video data) suggests that knowledge of cinematic principles can be exploited effectively for the understanding of films. To interpret an idea using the grammar, we need to first understand the symbols, as in natural languages, and second, understand the rules of combination of these symbols to represent concepts. Daniel Arijon, a famous name in film literature, writes, *"All the rules of film grammar have been on the screen for a long time. They are used by filmmakers as far apart geographically and in style as Kurosawa in Japan, Bergman in Sweden, Fellini in Italy and Ray in India. For them, and countless others this common set of rules is used to solve specific problems presented by the visual narration of a story"*, [4], page 4.

Therefore, the interpretation of concepts using this grammar first requires the extraction of appropriate features. Secondly, these features or *symbols* need to be semiotically (symbolic as opposed to semantic) explored as in natural languages. However, the interpretation of these symbols must comply with the governing production rules for a particular genre. An important aspect of this approach is to find a suitable mapping between low-level video features and their bottom-line semantics. These steps are summarized as:

- Learn the video making techniques used by the directors. These techniques are also called *Film Grammar.*
- Learn the theories and practices of film aesthetics, such as the effect of color on the mood, the effect of music on the scene situation and the effect of postprocessing of the audio and video on human perception.
- Develop a model to integrate the aforementioned information to explore concepts.
- Provide users with a facility to navigate through the audio-visual data in terms of concepts and ideas.

This framework is represented in Figure 1.1. In the next chapter, we will define a set of computable features and methods to compute them. Later, we will demonstrate that by combining these features with the semantic structure of talk and game shows, interview segments can be separated from commercials. Moreover, the video can be indexed as *host-shots* and *guest-shots* (Chapter 3). We will also show that by employing cinematic principles, Hollywood movies can be classified into different genres such as *action*, *comedy*, *horror* and *drama* based on their previews (Chapter 4). In Chapter 5, we will discuss a novel approach to segment a variety of videos into scenes and to select only one key frame to represent the content of an entire scene. In each chapter, we present experimental results which will demonstrate the appropriateness of our methodology.

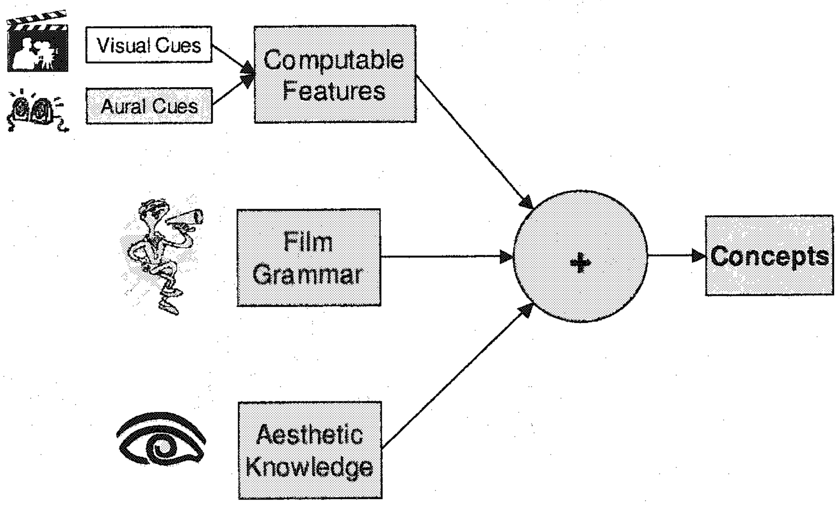


Figure 1.1: Our approach.

1.3 RELATED WORK

There have been several studies on indexing and retrieval for image databases, for example [64, 57, 35]. A large portion of the research work in this field is done by content extraction and matching. Features such as edges, shape, texture and GLCM (gray level consistency matrix) are extracted for all images in the database and indexed on the basis of similarity. Although these techniques work well for single images, they cannot be applied directly to video databases. The reason is that in the audio-visual data, the contents change with time. Even though videos are collections of still images, meaning is derived from the change in these images over time, which cannot be ignored in the indexing and retrieval task.

A large amount of work has also been reported in structuring videos, resulting in several interactive tools to provide navigation capabilities to the viewers. "Virage Video Engine" [18], VideoZoom [59], [60] and [12], are some examples. Similarly, content-based video indexing also constitutes a significant portion of the work in this area. Chang et al. [9] developed an interactive system for video retrieval. Several attributes of video, such as color, texture, shape and motion, are computed for each video in the database. To search for a video of interest, the user provided a set of parameters for attributes of video. These parameters were compared with those in the database using a weighted distance formula for the retrieval. A similar approach was reported by Deng and Manjunath in [13].

The use of Hidden Markov Models has been very popular in the research community for video categorization and retrieval. Naphade and Huang [39] proposed a probabilistic framework for video indexing and retrieval. Low-level features were mapped to high-level semantics as probabilistic multimedia objects called *multijects*. A Bayesian belief network, called *multinet*, was developed to perform the semantic indexing using Hidden Markov Models. Huang and Chang [21], Wolf [68], Dimitrova et al. [14] and Boreczky and Wilcox [8] are some other examples that make use of probabilistic approaches and classify videos into news, sports, and cartoons. The weakness of these approaches is due to the training on specific kinds of videos. That is, if the design and production styles are changed (for example, the position of the logo of the broadcasting channel), the HMM will fail to recognize the video. Similarly, a new set of training data will be required for videos of different production style. Haering et al. [16], on the other hand, suggested a semantic framework for video indexing and detection of *events*. They presented an example of *hunt* detection in videos in [43].

Much research work on video categorization has been done in the compressed domain using MPEG-1 and MPEG-2. The work in this area utilizes the features extractable from compressed video and audio. The compressed information may not be very precise, however, it avoids the overhead of computing features in the pixel domain. Kobla et al. [33] used the DCT coefficients, macroblock and motion vector information of MPEG videos for indexing and retrieval. Their proposed method was based on *query by example*. The methods proposed by Yeo and Liu in [70] and Patel and Sethi in [41], are few more examples which exploit compressed audio and video information. Lu et al. [36] applied the HMM approach in the compressed domain and promising results were presented on the classification of clips of news, commercials and sports events such as basketball and football. Recently, MPEG-7 has been employed for video indexing by using the embedded semantic descriptors [7]. However, the standardization of MPEG-7 is currently under development and the content-to-semantic interpretation for retrieval of videos is still an open question for the research community.

1.4 SUMMARY

Due to the tremendous increase in the availability of digital video in the last decade, there has been a need for automatic categorization and indexing of video. Provided with an appropriate indexing and tools to extract semantics of videos, users can quickly locate the clip of interest or retrieve a video of particular genre from large video databases. This task, however, requires methods to relate low-level video features to high-level semantics in order to bridge the gap between the two. In this chapter, we outlined the need for using film grammar for video categorization. We pointed out the fact that professionally created videos share a common set of rules or *grammar* pertaining to their genres. This grammar may have several dialects but is generally universal. We also provided a four-step approach to exploit this grammar for video categorization. In the next chapter, we will provide a list of low-level computable audio-visual features and methods to extract them. In the following chapters, we will present a framework to categorize TV talk and game shows into program segments and commercials. We will also demonstrate the use of computable features to identify the genre of feature films from their previews. Finally, we will construct a framework for temporal segmentation of produced videos, such that the video can be parsed in small story units or scenes. A novel approach of finding one representative image for the entire scene will also be discussed.

CHAPTER 2

COMPUTABLE FEATURES OF AUDIO-VISUAL DATA

2.1 FILM STRUCTURE

In this chapter, we first discuss the structure of a film, which is an example of audio-visual information, and then define the associated *computable features.* There is a strong analogy between a film and a novel. A *shot*, which is a collection of coherent (and usually adjacent) image frames, is similar to a *word.* A number of words make up a *sentence* as shots make a visual thought, called a *beat.* Beats are the representation of a subject and are collectively referred to as a *scene* in the same way that sentences collectively constitute a *paragraph.* Scenes create *sequences* like paragraphs make *chapters.* Finally, sequences produce a *film* when combined together as the chapters make a *novel* (see Figure2.1). This final audio-visual product, i.e. the film, is our input and the task is to extract the concepts within its small segments in a bottom-up fashion. Here, the ultimate goal is to decipher the meaning as it is perceived by the audience.

2.2 COMPUTABLE FEATURES

We define computable features of audio-visual data as a set of attributes that can be extracted using image/signal processing and computer vision techniques. In order to exploit knowledge of ubiquitous film grammar, it is necessary to be able to relate the symbols of film grammar to *computable video features.* Computable video features, as the name suggests, are defined as any statistic of the available video data. Since the relationship between film grammar symbols and high-level film semantics is known, if we are able to find computable representations of these symbols, the problem of video understanding can be favorably posed. Unfortunately, not all the symbols of film grammar can be well-represented in terms of a statistic. For instance, how does one compute the irony in a scene? It is immediately evident that *high-level* symbols like emotion, irony, or gestures are difficult to represent as statistics. On the other hand, *low-level* symbols like lighting, shot length and background music are far easier to represent. It should also be noted that low-level symbols correspond to the implicit communication that the director uses, and incidently are also the type of symbols that have the most established techniques. Audiences also become *trained* to interpret low-level symbols in a certain way, as is evidenced by feelings of expectation associated with silence, or feelings of fear associated with dim lighting. These ideas are investigated in depth in [51, 4]. In this chapter, we discuss these features and present methods to compute them.

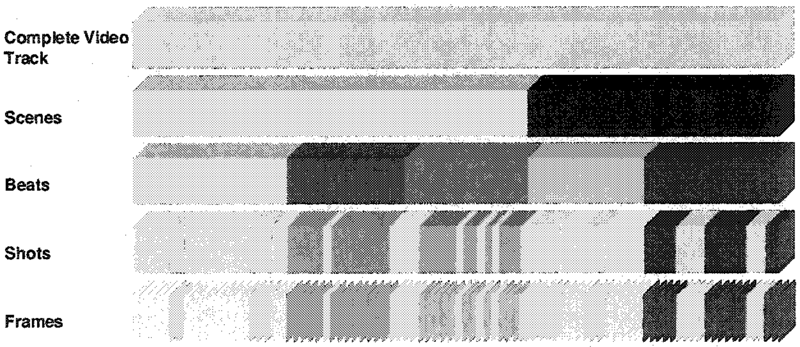


Figure 2.1: A film structure; frames are the smallest unit of the video. Many frames constitute a shot. Similar shots make scenes. The complete film is the collection of several scenes presenting an idea or concept.

2.2.1 SHOT DETECTION

A shot is defined as a sequence of frames taken by a single camera with no major changes in the visual content. We have used a modified version of the color histogram intersection method proposed by [17]. For each frame, a 16-bin HSV normalized color histogram is estimated with 8 bins for hue and 4 bins each for saturation and value. Let $D(i,j)$ represent the histogram intersection of two frames i and j that is:

$$D(i,j) = \sum_{k \in bins} min(H_i(k), H_j(k)), \tag{2.1}$$

where H_i and H_j are the histograms of frames i and j respectively and $D(i,j)$ represents their maximum color similarity. The similarity function, $S(i)$ for two consecutive frames i and $j = i - 1$ is then represented as:

$$S(i) = D(i, i-1).$$

Figures 2.2 and 2.3 show the histogram intersection of two pairs of frames. Generally, a fixed threshold is chosen empirically to detect the shot change. This approach works quite well (see [17]) if the shot change is abrupt without any shot transition effect. However, a variety of shot transitions occur in videos, for example *wipes* and *dissolves*. Applying a fixed threshold to $S(i)$ when the shot transition occurs with a *dissolve* generates several outliers because consecutive frames differ from each other until the shot transition is completed. To improve the accuracy, an iterative smoothing of the one dimensional function S is performed first. We have adapted the algorithm proposed by [42], based on anisotropic diffusion. This is done in the context of scale-space. S is smoothed iteratively using a Gaussian kernel such that the variance of the Gaussian function varies with the signal gradient. Mathematically:

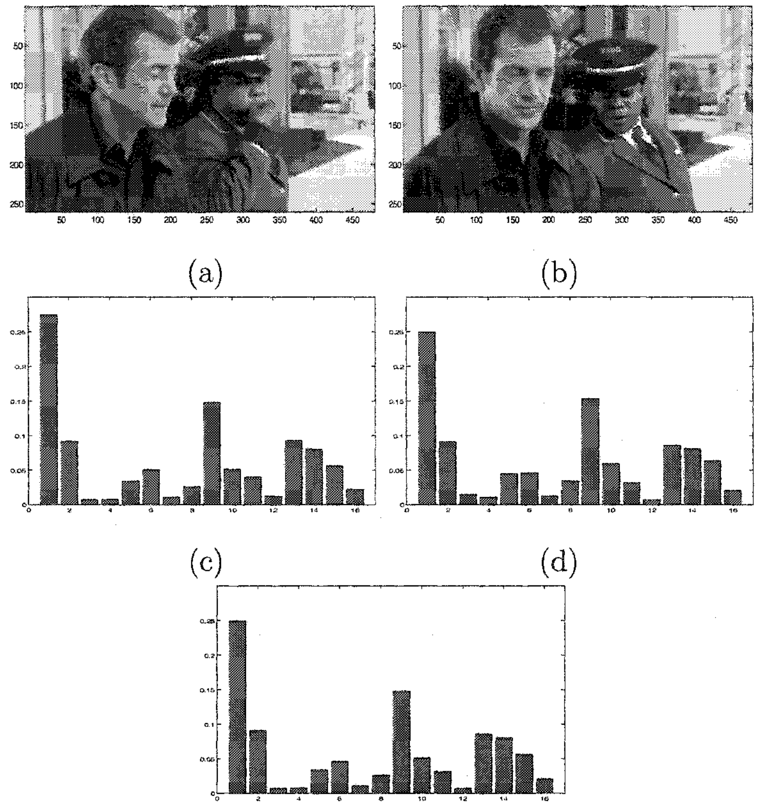
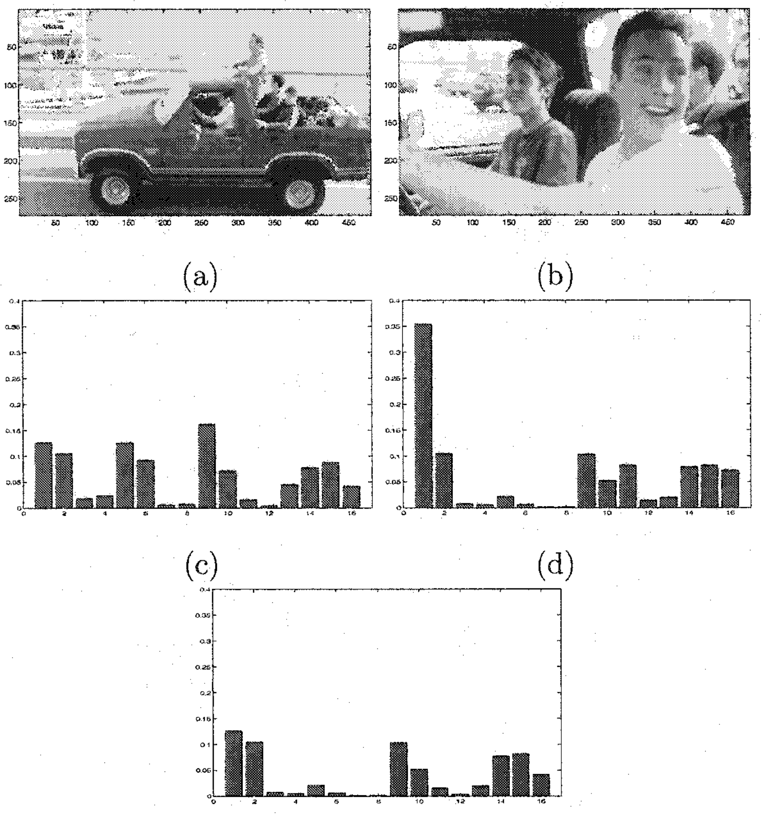


Figure 2.2: Histogram intersection for two consecutive frames of a shot. (a) and b) are two consecutive frames. (c) and (d) show the respective histograms in HSV space. (e) shows the intersection of (c) and (d). The intersection value is 0.95

$$S^{t+1}(i) = S^{t}(i) + \lambda \left[c_E \cdot \nabla_E S^t(i) + c_W \cdot \nabla_W S^t(i) \right], \tag{2.2}$$

where t is the iteration number and $0 < \lambda < 1/4$ with:

$$\begin{aligned} \nabla_E S(i) &\equiv S(i+1) - S(i), \\ \nabla_W S(i) &\equiv S(i-1) - S(i). \end{aligned} \tag{2.3}$$



(a) (b)

(c) (d)

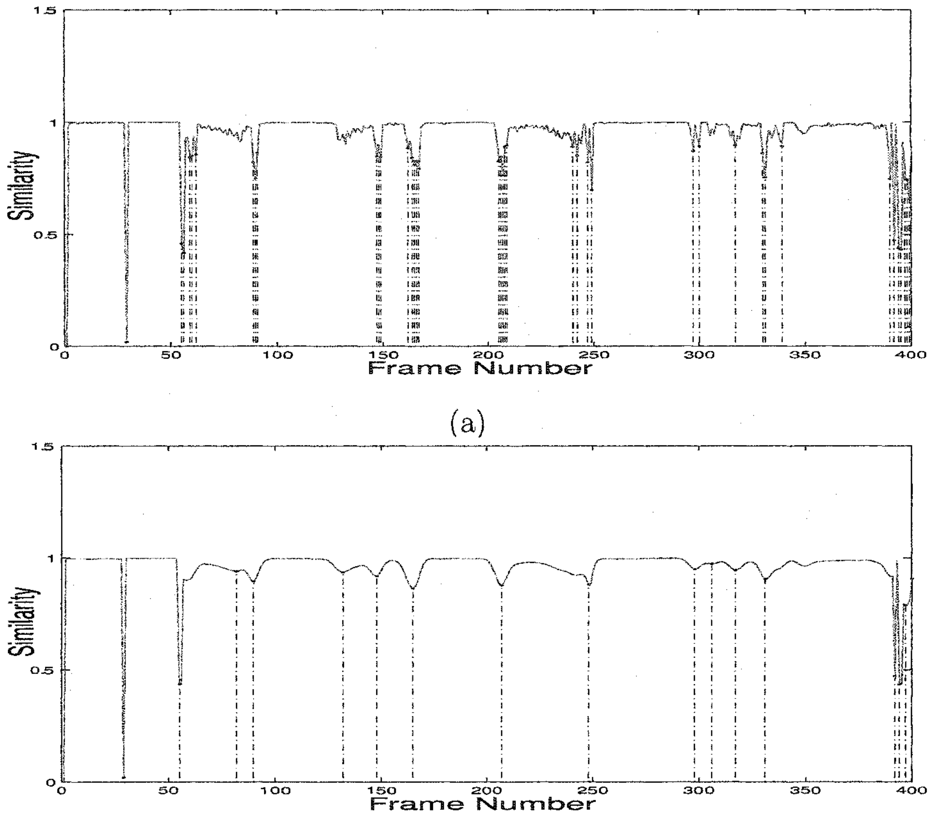
(e)

Figure 2.3: Histogram intersection for two shots. (a) and (b) are the last and the first frames of two consecutive shots respectively. (c) and (d) show the respective histograms in HSV space. (e) shows the intersection of (c) and (d). The intersection value is 0.66

The conduction coefficients are a function of gradients and are updated for every iteration as:

$$
\begin{aligned}
c_E^t &= g\left(\mid \nabla_E S^t(i) \mid\right), \\
c_W^t &= g\left(\mid \nabla_W S^t(i) \mid\right),
\end{aligned}
\tag{2.4}
$$

where $g(\nabla S) = e^{-(\frac{|\nabla_E|}{k})^2}$. In our experiments the constants were set to $\lambda = 0.1$ and $k = 0.1$. Finally, the shot boundaries are detected by finding the local minima in the smoothed

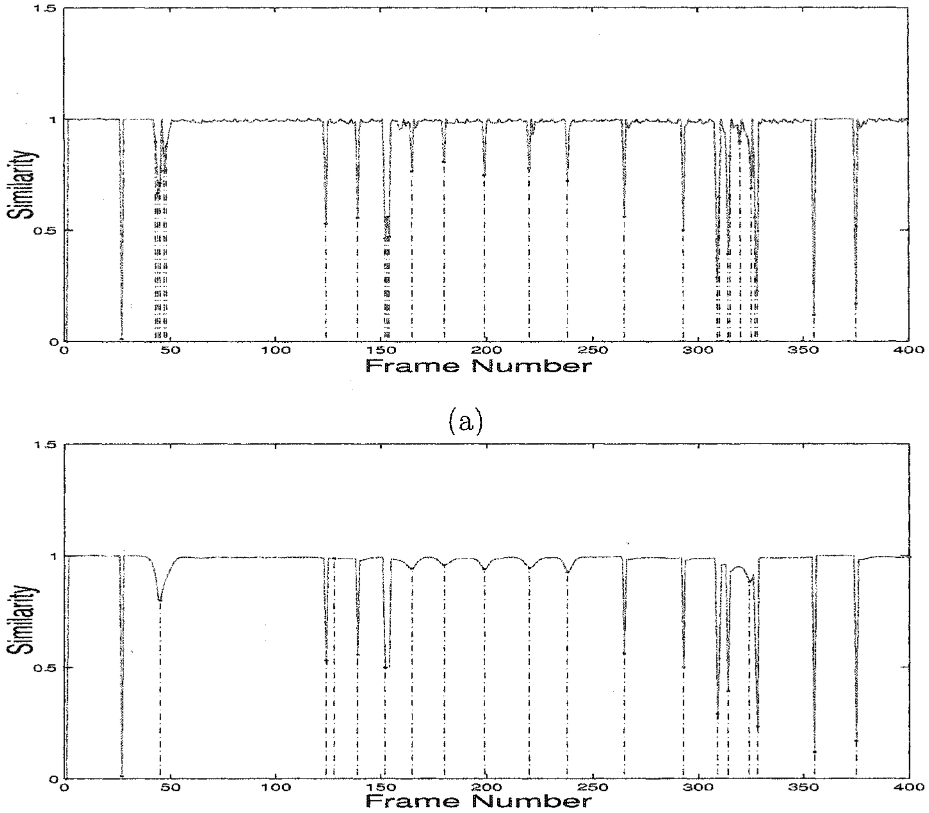


(a)

(b)

Figure 2.4: Shot detection results for the movie preview of "Red Dragon". There are 17 shots identified by a human observer. (a) Fixed threshold method. Vertical lines indicate the detection of shots. Number of shots detected: 40, Correct: 15, False positive: 25, False negative: 2 (b) Proposed method. Number of shots detected: 18, Correct: 16, False positive: 2, False negative: 1.

similarity function S. Thus, a shot boundary will be detected where two consecutive frames have minimum color similarity. This approach reduces the false alarms produced by the fixed threshold method. Figure 2.4 presents a comparison of the two methods. The similarity function S is plotted against the frame numbers. Only 400 frames are shown for convenient visualization. There are several outliers in (a) because gradually changing visual contents from frame to frame (*dissolve* effect) are detected as shot changes. For example,



(a)

(b)

Figure 2.5: Shot detection results for the movie preview of "Road Trip". There are 19 shots identified by a human observer. (a) Fixed threshold method. Vertical lines indicate the detection of shots. Number of shots detected: 28, Correct: 19, False positive: 9, False negative: 0. (b) Proposed method. Number of shots detected: 19, Correct: 19, False positive: 0, False negative: 0.

there are multiple shots detected around frame numbers 50, 150 and 200. However, in (b), a shot is detected when the similarity between consecutive frames is minimum. Compare the detection of shots with (a). Figure 2.5 also shows improved shot detection for the preview of the movie "Road Trip". In our experiments, we achieved about 90% accuracy for shot detection in most cases. See Table 2.1 that lists precision and recall of shot detection for some of the videos used in our experiments.

Table 2.1: Examples of shot detection results in some movie previews used in our experiments.

Shot Detection Results		
Movie	**Recall**	**Precision**
24 Hours Party People	0.96	0.84
Ali	0.85	0.91
American Pie	0.99	0.98
Americas Sweethearts	0.95	0.92
Big Trouble	0.89	0.92
Dracula 2000	0.95	0.96
The Fast and the Furious	0.93	0.87
Hannibal	0.94	0.86
The Hours	0.88	0.99
Jackpot	0.96	0.93
Kiss Of The Dragon	0.97	0.90
Legally Blonde	0.98	0.96
Mandolin	0.91	0.95
Red Dragon	0.96	0.91
Road Trip	1.00	0.99
Rush Hour	0.94	0.91
Sleepy Hollow	0.96	0.89
Stealing Harvard	0.98	0.95
The One	0.95	0.86
The Others	0.91	0.95
The Princess Diaries	0.90	0.88
The World Is Not Enough	0.96	0.83
The Tuxedo	0.98	0.91
What Lies Beneath	0.97	0.97
What Women Want	0.96	0.94

2.2.2 SHOT LENGTH

Once the shot boundaries are known, each shot S_i is represented by a set of frames, that is:

$$S_i = \left\{ f^a, f^{a+1}, ..., f^b \right\}, \tag{2.5}$$

where a and b are the indices of the first and the last frames of the i^{th} shot respectively. We associate a computable feature *shot length* with each shot which is the number of frames contained within the shot boundaries, that is

$$L_i = b - a + 1. \tag{2.6}$$

where L_i is the shot length of the i^{th} shot. Typically, dialogue shots are longer and span a large number of frames. On the other hand, shots of fight and chase scenes change rapidly and last for fewer frames ([4]).

2.2.3 SHOT REPRESENTATION

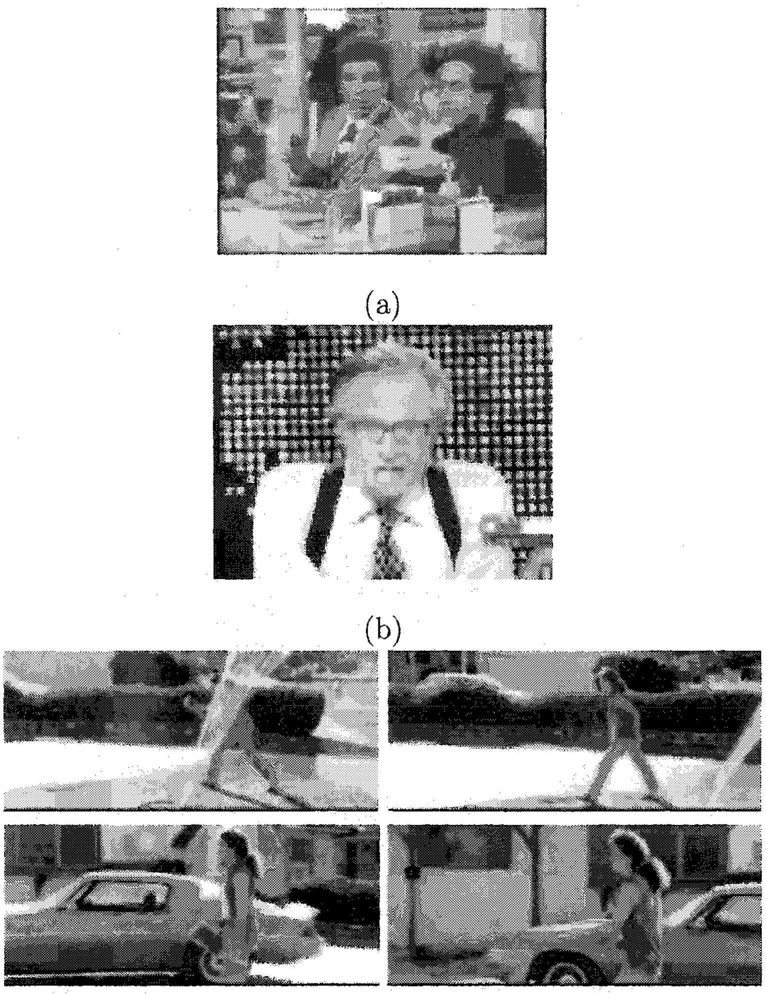
A shot may span a few to several hundreds frames. To compute further attributes of shots, such as color distribution, key-lighting, it is required to process all frames within the shot boundaries. However, the visual content in a shot may not vary a lot. Therefore, to reduce the computational complexity, often only one frame is processed and the results are generalized for the entire shot. We consider a variety of videos including feature movies, sitcoms, interview shows, which contain both action and non-action scenes. Selecting one key frame (for example the middle frame) may represent a static shot (a shot with little actors/camera motion) quite well, however, a dynamic shot (a shot with higher actors/camera motion) may not be represented adequately. Therefore, we have developed a method to select multiple key frames depending upon the shot activity. This is discussed in the following section.

2.2.3.1 Key Frame Selection

In this section we propose a method to select multiple key frames for each shot. Each shot, S_i, is represented by a set of key frames, K_i, such that all frames are distinct. Initially, the middle frame of the shot is selected and added to an empty set K_i as the first key frame. The reason for taking the middle frame instead of the first frame is to make sure that the frame is free from shot transition effects, for instance a diffusion effect. Next, each frame within a shot is compared to every frame in the set K_i. If the frame differs from all previously chosen key frames by more than a fixed threshold, it is added in the key frame set, otherwise it is ignored. This algorithm of key frame detection can be summarized as:

```
STEP 1: Select middle frame as the first key frame
  $K_i \leftarrow \{f^{\lfloor (a+b)/2 \rfloor}\}$
STEP 2: for $j = a$ to $b$
  if    $max\left(D\left(f^j, f^k\right)\right) < Th \qquad \forall f^k \in K_i$
     then $K_i \leftarrow K_i \cup \{f^j\}$
```

where Th is the minimum frame similarity threshold that declares two frames to be similar. With this approach, multiple frames are selected for the shots which have higher dynamics and temporally changing visual contents. For less dynamic shots, fewer frames are chosen. This method assures that every key frame is distinct and, therefore, prevents redundancy. Figure 2.6 shows key frames selected for shots from (a) a sitcom, (b) a talk show and (c) a feature movie.



(a)

(b)

(c)

Figure 2.6: Key frame detection: (a)-(b) One key frame selected for each shot from a sitcom and a talk show. (c) Multiple frames selected for a shot from the feature movie, *Terminator II*, due to greater shot activity.

2.2.4 SHOT MOTION CONTENT

Shot length and *shot motion content* are two interrelated features. These features provide cues to the nature of the scene. Typically, the motion content of shots also depends on the nature of the shot. The dialogue shots are relatively calm (neither actors nor the camera exhibit large motion) ([4]). Although camera pans, tilts and zooms are common in dialogue shots, they are generally smooth. In fight and chase shots, the camera motion is jerky and haphazard with larger actor movements. For a given scene, these two attributes are generally consistent over time to maintain the pace of the movie.

2.2.4.1 Computation of Shot Motion Content

Motion in shots can be divided into two classes; *global motion* and *local motion.* Global motion in a shot occurs due to the movement of the camera. This may include *pan* shots, *tilt* shots, *dolly/truck* shots and *zoom in/out* shots ([51]). On the other hand, local motion is the relative movement of objects with respect to the camera, for example, an actor walking or running. We define shot motion content as the amount of local motion in a shot and exploit the information encoded in MPEG-1 compressed video to compute it. The horizontal and vertical velocities of each block are encoded in the MPEG stream. These velocity vectors may indicate global or local motion. We estimate the global affine motion using a least squares method. The goodness of the fit is measured by examining the difference between the actual and reprojected velocities of the blocks. The magnitude of this error is used as a measure of shot motion content. An affine motion model with six parameters is represented as follows:

$$\begin{aligned} u &= a_1 \cdot x + a_2 \cdot y + b_1 \\ v &= a_3 \cdot x + a_4 \cdot y + b_2, \end{aligned} \tag{2.7}$$

where u and v are horizontal and vertical velocities obtained from the MPEG file, a_1 through a_4 capture the camera rotation, shear and scaling, b_1 and b_2 represent the global translation in the horizontal and vertical directions respectively, and $\{x, y\}$ are the coordinates of block's centroid. Let u_k and v_k be the encoded velocities and u'_k and v'_k be the reprojected velocities of the k^{th} block in the j^{th} frame using the affine motion model, then the error ϵ_j in the fit is measured as:

$$\epsilon_j = \sum_{k \in motionblocks} \sqrt{(u'_k - u_k)^2 + (v'_k - v_k)^2}. \tag{2.8}$$

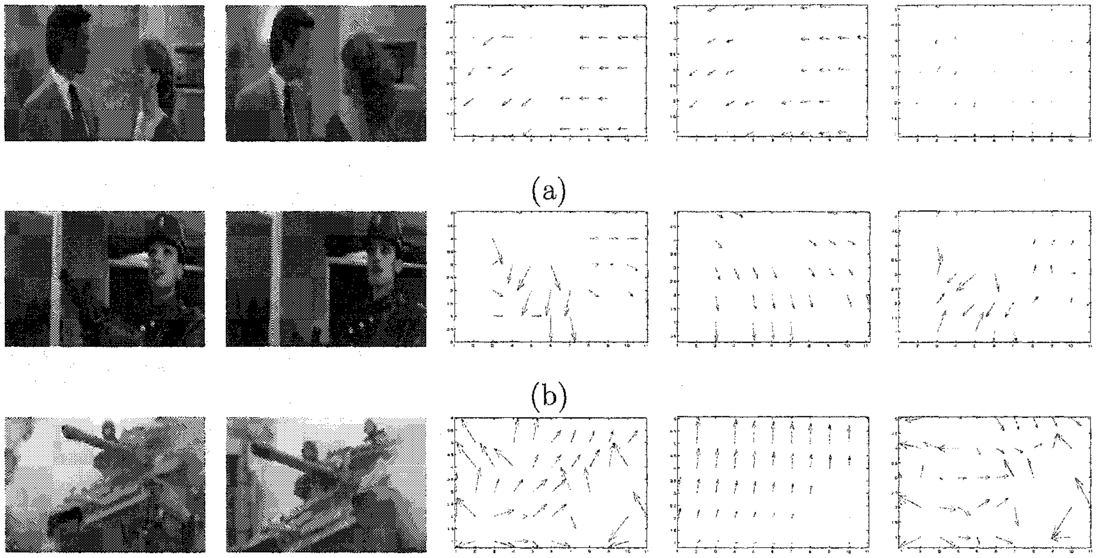
The shot motion content of shot i is the aggregation of ϵ of all P frames in the shot:

$$SMC_i = \sum_{j \in S_i} \epsilon_j, \tag{2.9}$$

where SMC is the *shot motion content.* Figure 2.7 shows the shot motion content for three different cases. The SMC in the shot is normalized by the total number of P frames in the shot.

2.2.5 COLOR VARIANCE

Zettl observes in [72], "*The expressive quality of color is, like music, an excellent vehicle for establishing or intensifying the mood of an event.*" In Chapter 4, we will demonstrate that the variance of color in a clip *as a whole* can be exploited to discriminate between genres of a film. Intuitively, the variance of color has a strong correlational structure with respect to genres, for instance, comedies tend to have a large variety of bright colors, whereas horror films often adopt only darker hues. Thus, in order to define a computable feature, two requirements have to be met. First, the features has to be defined that is *global* in nature, and second, distances in the color space employed should be perceptually uniform. We employ the CIE *Luv* space, which was designed to approach a perceptually uniform color space. To represent the variety of color used in the video we employ the generalized



(a)

(b)

(c)

Figure 2.7: Estimation of shot motion content using motion vectors. The first two columns show frames from each shot. Encoded motion vectors from the MPEG file for each frame are shown in the third column. The fourth column shows the reprojected flow vectors after a least squares fit using an affine model. The rightmost column shows the difference between the actual and the reprojected flow vectors. The SMC error by our algorithm for the three examples is (a) 9.8, (b) 46.64 and (c) 107.03. These values are proportional to the shot activity.

variance of the *Luv* color space of each clip. The covariance matrix of the multivariate vector is defined as

$$\rho = \begin{bmatrix} \sigma_L^2 & \sigma_{Lu}^2 & \sigma_{Lv}^2 \\ \sigma_{Lu}^2 & \sigma_u^2 & \sigma_{uv}^2 \\ \sigma_{Lv}^2 & \sigma_{uv}^2 & \sigma_v^2 \end{bmatrix}, \tag{2.10}$$

where σ_L^2 is the variance of L component of color, σ_{Lu}^2 is the covariance of L and u components of color, and so on. The *generalized variance* is obtained by finding the determinant of ρ,

$$\sum = \det(\rho) = \sigma_L^2\sigma_u^2\sigma_v^2 - \sigma_L^2\sigma_{uv}^4 - \sigma_v^2\sigma_{Lu}^4 - \sigma_u^2\sigma_{Lv}^4 + 2\sigma_{Lu}^2\sigma_{Lv}^2\sigma_{uv}^2. \tag{2.11}$$

This feature is used as a representation of the color variance.

2.2.6 LIGHTING KEY

In the hands of an able director, lighting is an important dramatic agent. Generations of filmmakers have exploited luminance to evoke emotions, using techniques that are well studied and documented in cinematography circles [72]. A deliberate relationship exists, therefore, between the lighting and the genre of a film.

In practice, movie directors use multiple light sources to balance the amount and direction of light while shooting a scene. The purpose of using several light sources is to enable a specific portrayal of a scene. For example, how and where shadows appear on the screen is controlled by maintaining a suitable proportion of intensity and direction of light sources. Lighting can also be used to direct the attention of the viewer to a certain area of importance in the scene. It can also affect viewers' emotions directly, regardless of the actual content of the scene. Reynertson comments on this issue: *"The amount and distribution of light in relation to shadow and darkness and the relative tonal value of the scene is a primary visual means of setting mood."* [51], p. 107. In other words, lighting is an issue not only of enough light in the scene to provide good exposure, but of light and shade to create a dramatic effect, consistent with the scene. In a similar vein, Wolf Rilla says *"All lighting, to be effective, must match both mood and purpose. Clearly, heavy contrasts, powerful light and shade, are inappropriate to a light-hearted scene, and conversely a flat, front-lit subject lacks the mystery which back-lighting can give it."* [52] p. 96.

There are numerous ways to illuminate a scene. One of the commonly used methods in the film industry is called *Three Point Lighting*. As the name implies, this style uses three main light sources:

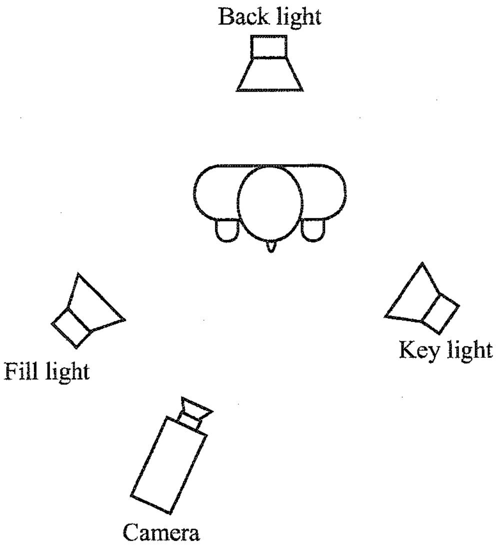


Figure 2.8: Positioning of lights in a three-point lighting setup.

Key light: This is the main source of light on the subject. It is the source of greatest illumination.

Back light: This source of light helps emphasize the contour of the object. It also separates it from a dark background.

Fill light: This is a secondary illumination source which helps to soften some of the shadows thrown by the key light and back light.

Figure 2.8 shows how the light sources are placed with respect to the camera and the subject. With different proportions of intensity for each source, movie directors *paint* the scene with light and typify the situation of the scene. Thus, within the design phase of a scene there is deliberate correlation between scene context and the lighting of the scene. In film literature, two major lighting methods are used to establish such a relation between the context and the mood of the viewer, called *low key lighting* and *high key lighting*:

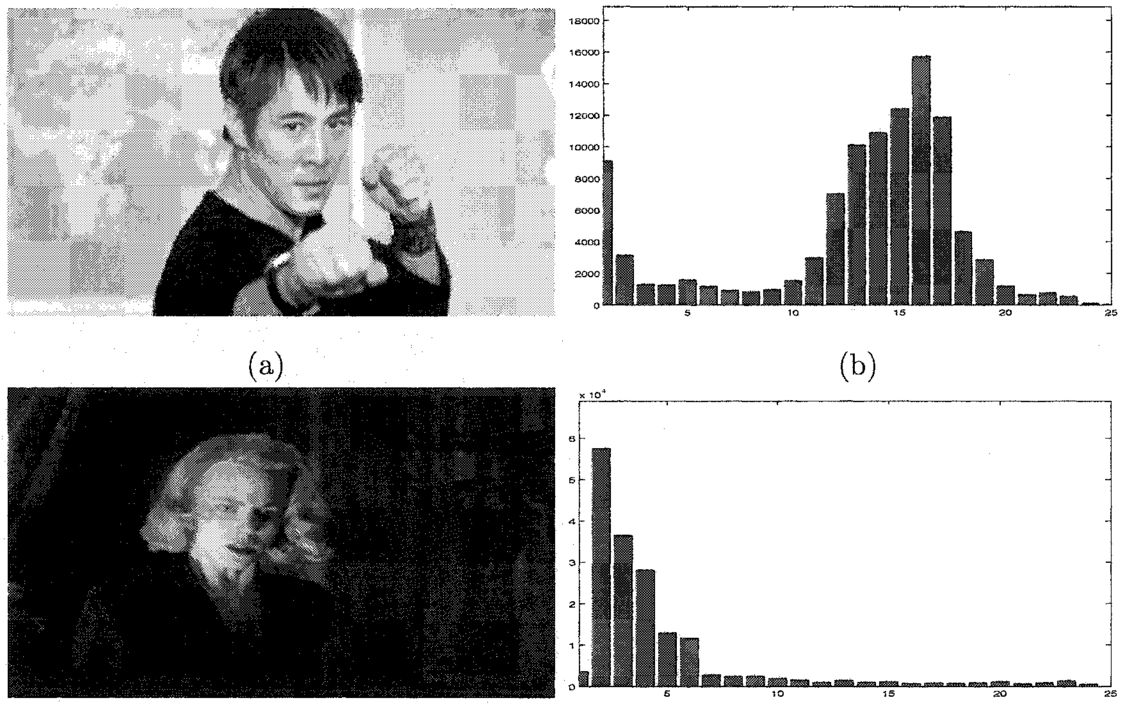
High key lighting: *High key* lighting means that the scene has an abundance of bright light. It usually has less contrast, and the difference between the brightest light and the dimmest light is small. Practically, this configuration is achieved by maintaining a low *key-to-fill* ratio i.e. a low contrast between dark and light. *High key* scenes are usually action scenes or are less dramatic. As a result, comedies and action films typically have high key lighting [72], p. 32.

Low key lighting: In this type, the background of the scene is generally predominantly dark. In *low key* scenes, the contrast ratio is high. *Low key* lighting is more dramatic and is often used in *film noir* or *horror* films.

2.2.6.1 Computing the Lighting Key

Many algorithms exist that compute the position of a light source in a given image. If the direction and intensity of the light sources are known, the *key* of the image can be easily deduced, and some higher-level interpretation of the situation can be elicited. Unfortunately, for general scenes, of the nature usually encountered in films, assumptions typically made in existing algorithms are violated, for example, single light source or uniform Lambertian surface. However, it is still possible to compute the *key* of the lighting using a simple computation. The brightness value of pixels in an image vary proportionally with the scene illumination and the surface properties of the observed object. Hence, a *high key* shot, which is more illuminated than a *low key* shot, contains a higher proportion of bright pixels. On the other hand, a *low key* frame contains more pixels of lower brightness. This simple property has been exploited here to distinguish between these two categories. Figure 2.9 shows the distribution of brightness values of *high* and *low* key shots. It can be roughly observed from the figure that for low key frames, both the mean and the variance are low, whereas for high key frames the mean and variance are both higher. Thus, for a given key frame, i, with $m \times n$ pixels in it, we find the mean, μ, and standard deviation, σ, of the value component in the HSV space. The value component is known to correspond to brightness. A scene lighting quantity $\zeta_i(\mu, \sigma)$ is then defined as a measure of the lighting key of a frame,

$$\zeta_i = \mu_i \cdot \sigma_i. \tag{2.12}$$



(a) (b)

(c) (d)

Figure 2.9: Distribution of gray scale pixel values in (a) *high key* shot (b) histogram and (c) *low key* shot (d) histogram.

In *high key* frames, the light is well distributed which results in larger values for the standard deviation and the mean. Whereas, in *low key* shots, both μ and σ are small. This enables us to formally interpret a higher-level concept from low-level information, namely the key of the frame. This computable feature is exploited in Chapter 4 to classify movies into genres from their previews.

2.2.7 AUDIO FEATURES

Music and nonliteral sounds are often used to provide additional energy to a scene. They can quite easily describe a condition; such as whether a situation is stable or unstable. In movies, the audio is often correlated with the scene. For example, shots of fighting and explosions are usually accompanied by a sudden change in the audio level. Therefore, the energy in the audio track can be used as a cue to detect such events when the peak in the audio energy is relatively high. The energy of an audio signal is computed as:

$$E = \sum_{i \in interval} (A_i)^2 , \qquad (2.13)$$

where A_i is the audio sample indexed by time i and *interval* is a small window which is set to 50 ms for our experiments. Since our interest is in the instances where the energy in audio changes abruptly, a peakiness test is performed on the energy plot. A peak is good if it is sharp and deep. That is:

$$Peakiness_i = \left(1 - \frac{V_a + V_b}{2P}\right) \cdot \left(1 - \frac{N}{W \times P}\right) , \qquad (2.14)$$

where P is the heights of the peak, V_a and V_b are the height of the valleys on either sides of the peak. W is the width of the peak and N denotes the area under the valley, (see Figure 2.10). Figure 2.11 shows plots of the audio signal and its energy for the audio track of the movie preview of “The World Is Not Enough”.

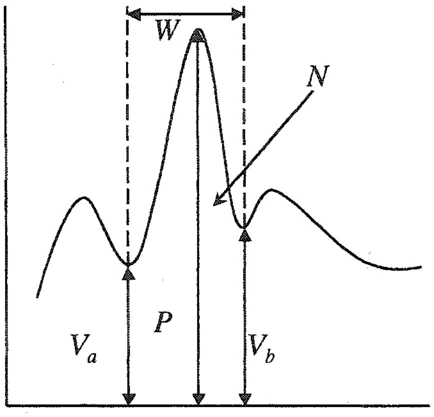
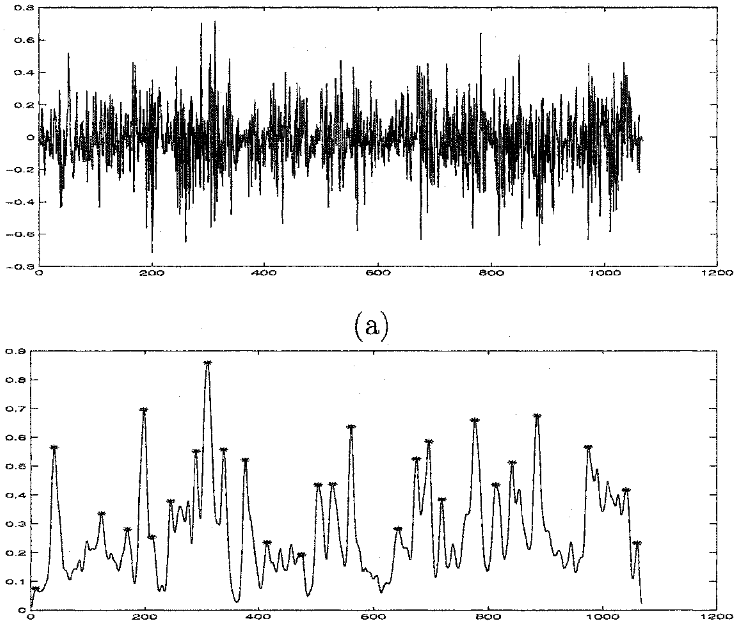


Figure 2.10: Peakiness test.



(a)

(b)

Figure 2.11: Audio processing: (a) the audio waveform of the movie "The World Is Not Enough", (b) Energy plot of the audio: Good peaks are indicated by '*' after the peakiness test.

2.3 SUMMARY

In this chapter, we defined a set of computable features for audio-visual data that can be combined with filmmaking rules in order to map low-level features to high level semantics. So far, we have focused on visual information, however, the size of the feature set can be extended by incorporating other sources of information. For example, closed captions, which accompany many commercially created videos as text, can be used to extract context. Also, speech processing techniques can be used to identify the speaker as male or female or to distinguish between speech and music. In the next chapter, we will exploit the semantics of TV talk and game shows to automatically remove the commercials from the video and to further classify shots into host and guest shots.

CHAPTER 3

CATEGORIZATION OF TALK AND GAME SHOWS USING VISUAL CUES

3.1 INTRODUCTION

Talk show videos are an important segment of televised programs. Several popular prime-time programs are based on the host and guests concept, for example "Crossfire", "The Larry King Live", "Who Wants To Be A Millionaire", "Jeopardy" and "Hollywood Squares". In this section, we address the problem of organizing such video shows. We assume that the user might be interested in looking only at interview segments without the commercials. Perhaps the user wants to view only clips that contain the questions asked during the show or only the clips which contain the answers of the interviewee. For example, the user might be motivated to watch only the questions in order to get a summary of the topics discussed in a particular interview. We exploit the *Film Grammar* of such shows and extract interview segments by separating commercials. We further classify interview segments as shots of the host or the guests.

In our approach, we rely primarily on information contained in shot transitions, rather than analyzing the scene content of individual frames. We utilize the inherent difference in the scene structure (grammar) of commercials and talk shows to differentiate between them. The entire show is first parsed into shots. Then, we construct a data structure called a *shot connectivity graph*, which links similar shots over time. Analysis of the shot connectivity graph helps us automatically separate commercials from program segments.

This is done by first detecting stories, and then assigning a weight to each story based on its likelihood of being a commercial or a program segment. We further analyze stories to distinguish shots of the hosts from shots of the guests. The results of extensive experiments on eight full length talk and game shows are provided at the end of the chapter.

3.2 RELATED WORK

The Informedia Project ([22]) at Carnegie Mellon University is one of the earliest works that focused on the segmentation of news videos in particular. It spearheaded the effort to segment and automatically generate a database of news broadcasts every night. The overall system relied on multiple cues, like video, speech, close-captioned text and other cues. Haupmann et al. [20], however, presented a heuristic approach to segment commercials and individual news stories. They relied heavily on the fact that commercials have more rapidly changing shots than programs and are separated by blank frames. The overall error reported is high. We, on the other hand, exploit scene structure rather than multiple heuristics based on shot change rate.

There are a few approaches that deal with higher-level semantics, instead of using low-level feature matching as the primary indexing criteria. Work by Fischer et al. in [15] and a similar approach by Truong et al. in [63], distinguished between newscasts, commercials, sports, music videos and cartoons. The feature set consisted of scene length, camera motion, object motion and illumination. These approaches were based on training the system using examples and then employing a decision tree to identify the genre of the video. Colombo et al. [10] exploited the colors, editing effects, rhythms, and object motion patterns of commercials and extracted an 18-dimensional feature vector for each commercial shot. The commercials were retrieved from the database by the degree to which the video conformed to the indices of the semantics. The commercials could also be detected by selecting one of the videos of the database as an example and performing a query for similar videos. They, however, did not provide any discussion on differentiating commercials from other televised programs, like sitcoms and talk shows.

Yeo et al. [71] used a scene transition graph to extract scene structure of sitcoms. We employ a similar data structure in our computations. However, our work differs from their work in some important respects. In [71], all cut edges were treated as story boundaries. This paradigm would result in a high number of stories for non-repetitive scenes, like commercials. Their approach, therefore, would not work well in separating commercials from programs. In addition, we employ a novel weighing scheme for each story to distinguish commercials from programs in a logical way, as apposed to the training based approaches adopted in [21, 68, 14, 15, 63] and [8]. We also analyze the story for its content, rather than simply finding its boundaries.

3.3 THE GRAMMAR OF TALK AND GAME SHOWS

The programs belonging to the genre of videos in which a host interacts with guests share a common grammar. This grammar can be summarized as:

- The camera switches back and forth between the host and the guests.
- Frequent repetitions of shots.
- *Guests'* shots are longer than *Hosts* shots.

On the other hand, commercials are characterized by the following grammar:

- More colorful shots than talk and game shows.
- Fewer repetitions of shots.
- Rapid shot transitions and small shot durations.

In the next section we describe a data structure for videos which will lead to the extraction of program sections and detection of program host and guests.

3.3.1 SHOT CONNECTIVITY GRAPH

We first find the shot boundaries and organize the video into a data-structure, called a *Shot Connectivity Graph, G.* This graph links similar shots over time. The vertices V represent the shots and edges represent the relationship between the nodes. Each vertex is assigned a label indicating the serial number of the shot in time and a weight w equal to the shot's length. In order to connect a node with another node we test the key frames of respective shots for three conditions:

- *Shot similarity constraint*: Key frames of two shots should have similar distribution of HSV color values.
- *Shot proximity constraint*: A shot may be linked with a recent shot (within the last T_{mem} shots).
- *Blank shot constraint*: Shots may not be linked across a blank in the shot connectivity graph. Significant story boundaries (for example, between the show and the commercials) are often separated by a short blank sequence.

Two nodes in the *Shot Connectivity Graph* are linked if the following condition is satisfied:

$$\sum_{j \in bins} min(H_q(j), H_{q-k}(j)) \geq T_{color} \quad for some\ k \leq T_{mem}, \tag{3.1}$$

where T_{color} is a threshold on the intersection of histograms. Thus two vertices v_p and v_q, where $v_p, v_q \in V$ and $p < q$, are adjacent, that is, they have an edge between them, if and only if:

- v_p and v_q represent consecutive shots or
- v_p and v_q satisfy the shot similarity, shot proximity and blank shot constraints.

The shot connectivity graph exploits the structure of the video selected by the directors in the editing room. Interview videos are produced using multiple cameras running simultaneously, recording the host and the guest. The directors switch back and forth between them to fit these parallel events on a sequential tape. Examples of shot connectivity graphs automatically computed by our method are shown in Figure 3.1, 3.2 and 3.3.

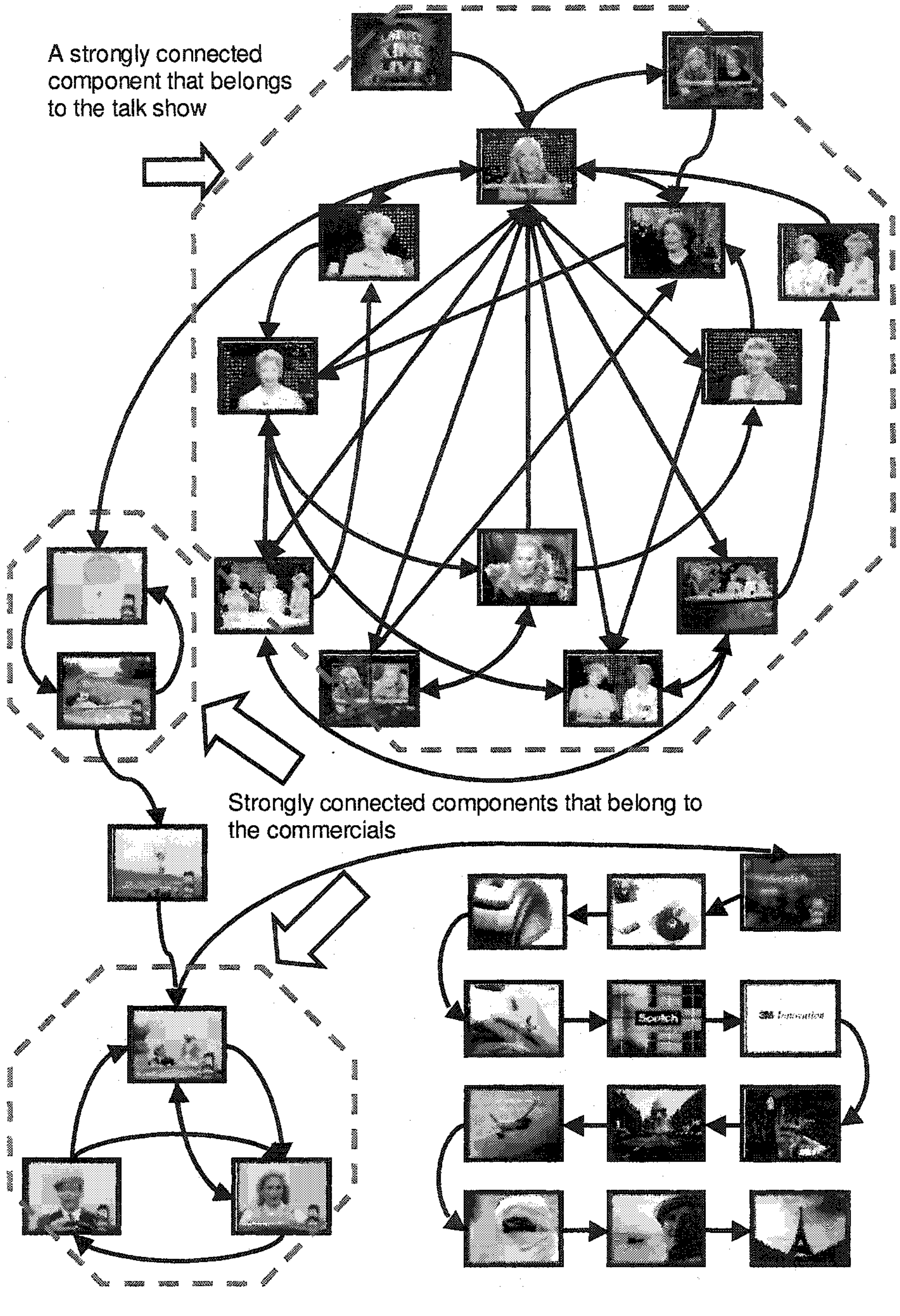
3.4 STORY SEGMENTATION AND REMOVAL OF COMMERCIALS

Shots in talk shows have strong visual correlation, both backwards and forwards in time, and this repeating structure can be used as a key cue in segmenting them from commercials, which are non-repetitive and rapidly changing. There may still be repetitive shots in a commercial sequence, which appear as cycles in the shot connectivity graph. However, these shots are not nearly as frequent, or as long in duration, as those in the interview. Moreover, since our threshold of linking shots back in time is based on the $numberofshots$, and not on the total time elapsed, commercial segments will have less *time memory* than talk shows.

To extract a coherent set of shots, or *stories*, from the shot connectivity graph G, we find all strongly connected components in G. A strongly connected component $G'(V', E')$ of G has the following properties:

- $G' \subseteq G$
- There is a path from any vertex $v_p \in G'$ to any other vertex $v_q \in G'$.
- There is no $v_z \in (G - G')$ such that adding v_z to G' will form a strongly connected component.

Each strongly connected component $G' \in G$ represents a story. We compute the likelihood of all such stories being part of a program segment. Each story is assigned a weight based on two factors; the *number of frames in a story* and the *ratio of the number of repetitive shots* to the *total number of shots in a story.* The first factor follows from the observation that long stories are more likely to be program segments than commercials.



A strongly connected component that belongs to the talk show
Strongly connected components that belong to the commercials

Figure 3.1: A Shot Connectivity Graph of the "Larry King Live" show hosted by Leeza Gibbons. Strongly connected components are circumscribed by octagons. Note that the interview sections appear with more connected components. Commercials, on the other hand, have smaller cycles and fewer repetitions.

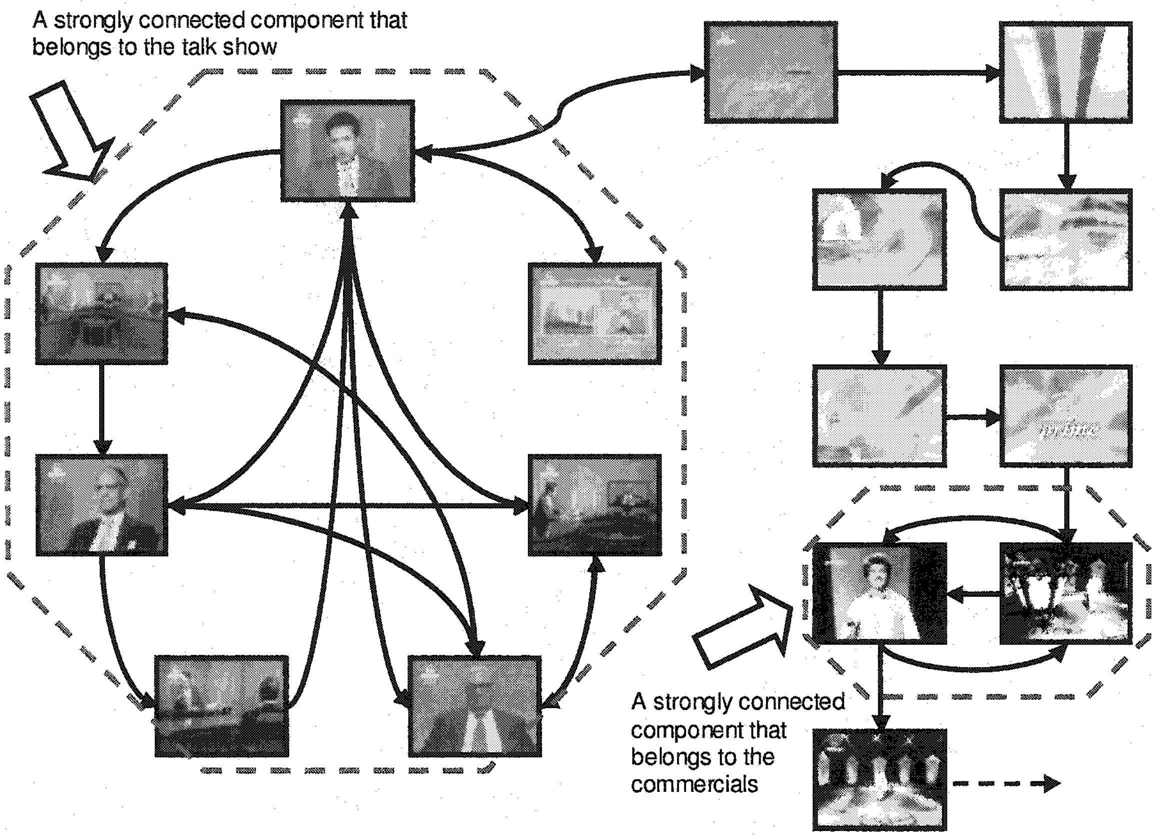

A strongly connected component that belongs to the talk show
A strongly connected component that belongs to the commercials

Figure 3.2: A Shot Connectivity Graph of a Pakistani talk show "News Night" followed by commercials. Strongly connected components are circumscribed by octagons. Note that the segment of the talk show forms a strongly connected component.

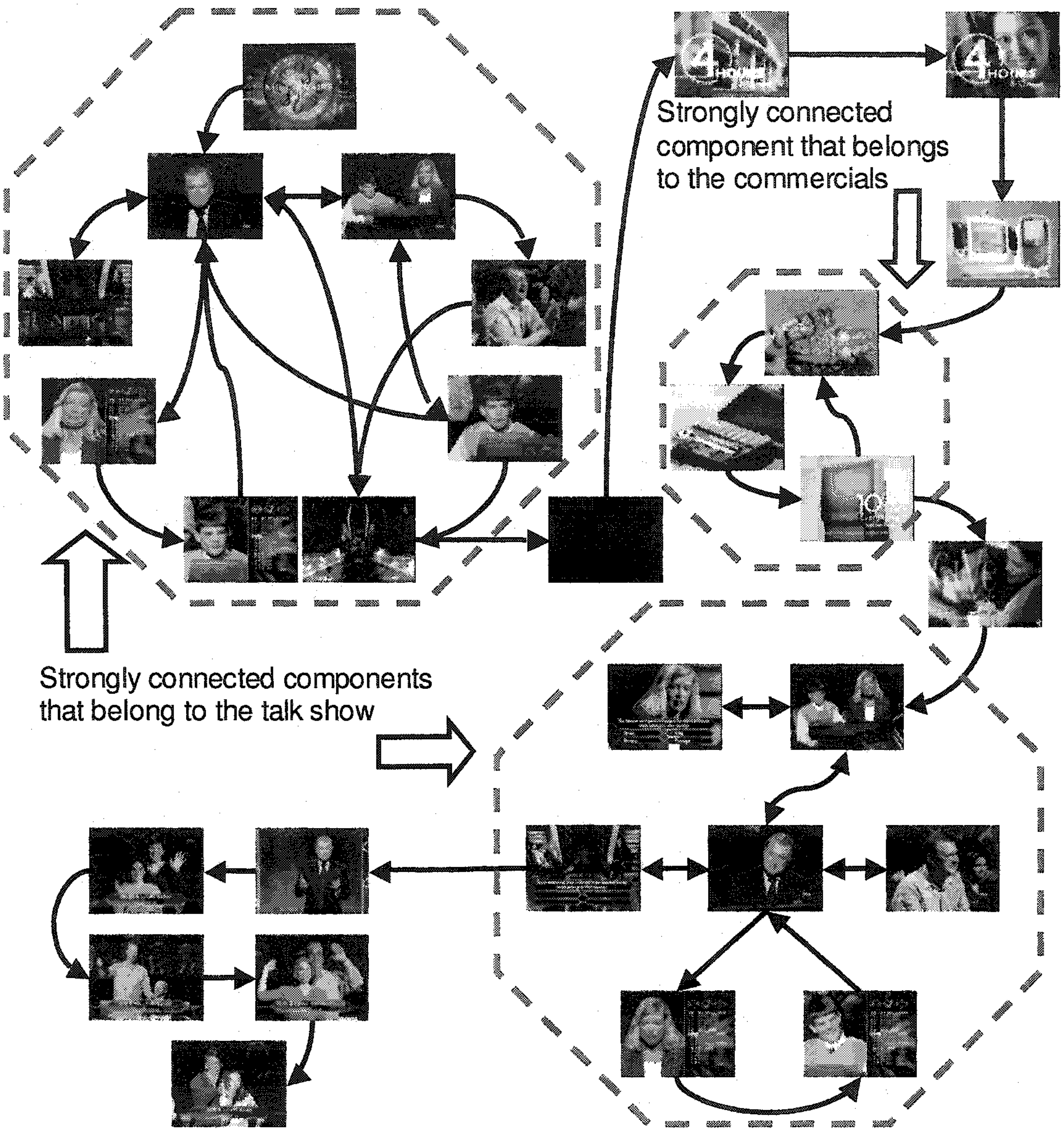

Strongly connected component that belongs to the commercials
Strongly connected components that belong to the talk show

Figure 3.3: A Shot Connectivity Graph of a game show, "Who Wants to be a Millionoire" followed by commercials. Strongly connected components are circumscribed by octagons. Note that the segment of the talk show forms a strongly connected component.

Stories are determined from strongly connected components in the shot connectivity graph. Therefore, a long story means that we have observed multiple overlapping cycles within the story since the length of each cycle is limited by T_{mem}. The second factor stems from the observation that programs have a large number of repetitive shots in proportion to the total number of shots. Commercials, on the other hand, have a high shot transition rate. Even though commercials may have repetitive shots, this repetition is small compared to the total number of shots. Thus, program segments will have more repetition than commercials, relative to total number of shots. Both of these factors are combined in the following likelihood of a story being a program segment:

$$L(G') = \sum_{\forall j \in G'} w_j \cdot \frac{\sum_{\forall E'_{ji} \in G' | j > i} 1}{\sum_{\forall v_j \in G'} 1} \cdot \Delta t, \tag{3.2}$$

where G' is the strongly connected component representing the story, w_j is weight of the j^{th} vertex, i.e. the number of frames in the shot. E' are the edges in G'. Δt is the time interval between consecutive frames. Note that the denominator represents the total number of shots in the story. This likelihood forms a weight for each story, which is used to determine the label for the story. Stories with $L(story)$ higher than a certain threshold are labelled as program stories, whereas those that fall below the threshold are labelled as commercials. This scheme is robust and yields accurate results, as shown in Section 3.5.

3.4.1 HOST DETECTION: ANALYSIS OF SHOTS WITHIN AN INTERVIEW STORY

We perform further analysis of program stories to differentiate host shots from those of guests. Note that in most talk shows a single person is the host for the duration of the program, but the guests keep changing. Also, the host asks questions which are typically shorter than answers. These observations can be utilized for successful segmentation. Note that no specific training is used to detect the hosts. Instead, the host is detected from the pattern of shot transitions, exploiting the semantics of scene structure.

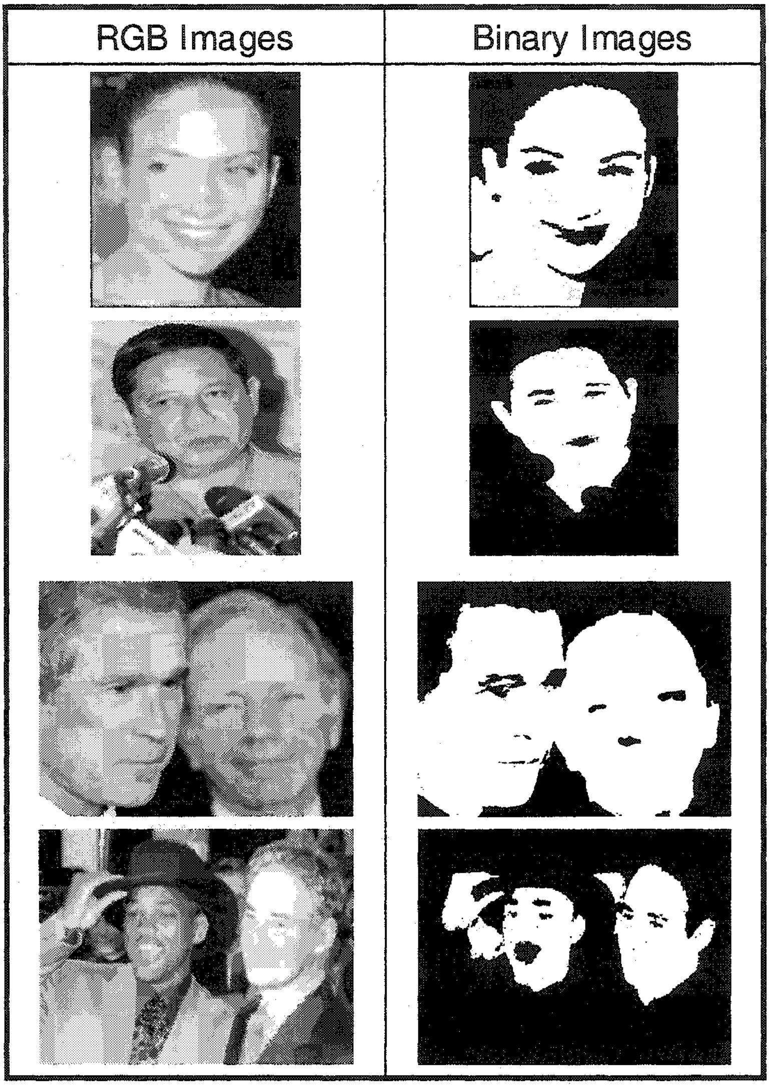


Figure 3.4: Example images and their binary masks used to train the system for skin detection. Portions of the images containing skin are manually marked in the binary images.

For a given show, we first find the N shortest shots in the show containing only one person. To determine whether a shot has one or more persons, we use the skin detection algorithm presented by [32], using RGB color space. A skin color predicate is first trained on a few training images, by manually marking skin regions and building a 3D color histogram of these frames. Figure 3.4 shows some of the training images used to train the system for skin detection. A binary mask is made for each image marking the presence of skin. For each positive example, the histogram is incremented by a 3D Gaussian distribution, so that colors similar to the marked skin color also get selected. For each negative training example, the histogram is decremented by a narrower Gaussian. After incorporating information from all training images, the color predicate is thresholded to a small positive value, and thus essentially forms a color lookup table. Including persons of various ethnic backgrounds in training images makes this color predicate robust for a variety of skin tones. For detection, the color of each pixel is looked up in the color predicate to be labelled as skin or non-skin. If the image contains only one significant skin colored component, then it is assumed to have one person in it. Figure 3.5 shows some results of skin detection.

The key frames of the N shortest shots containing only one person are correlated with each other to find the most repetitive shot. Since questions are typically much shorter than answers, host shots are typically shorter than guest shots. Thus it is highly likely that most of the N shots selected will be host shots. An $N \times N$ correlation matrix C is computed such that each term of C is given by:

$$C_{ij} = \frac{\sum_{r \in rows} \sum_{c \in cols} (I_i(r,c) - \mu_i).(I_j(r,c) - \mu_j)}{\sqrt{\left(\sum_{r \in rows} \sum_{c \in cols} (I_i(r,c))^2\right) \left(\sum_{r \in rows} \sum_{c \in cols} (I_j(r,c))^2\right)}}, \tag{3.3}$$

where I_k is the gray-level intensity image of frame k and μ_k is its mean. Notice that all the diagonal terms in this matrix are 1 (and therefore do not need to be actually computed). Also, C is symmetric, and therefore only half of the non-diagonal elements need to be computed. The frame which returns the highest sum for a column is selected as the key frame representing the host. That is,

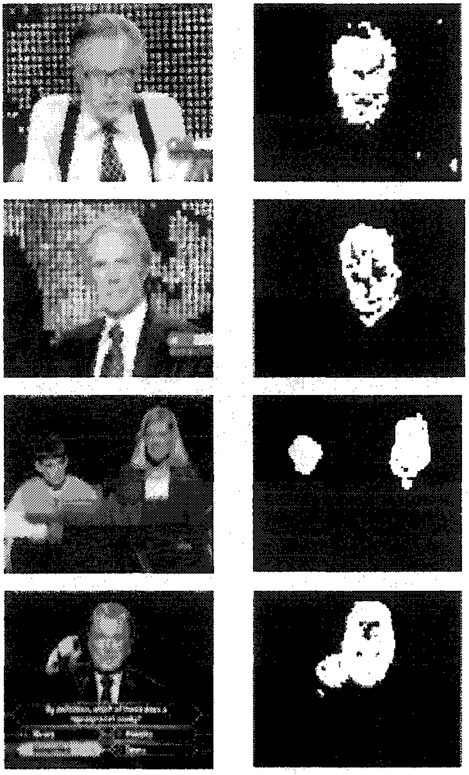
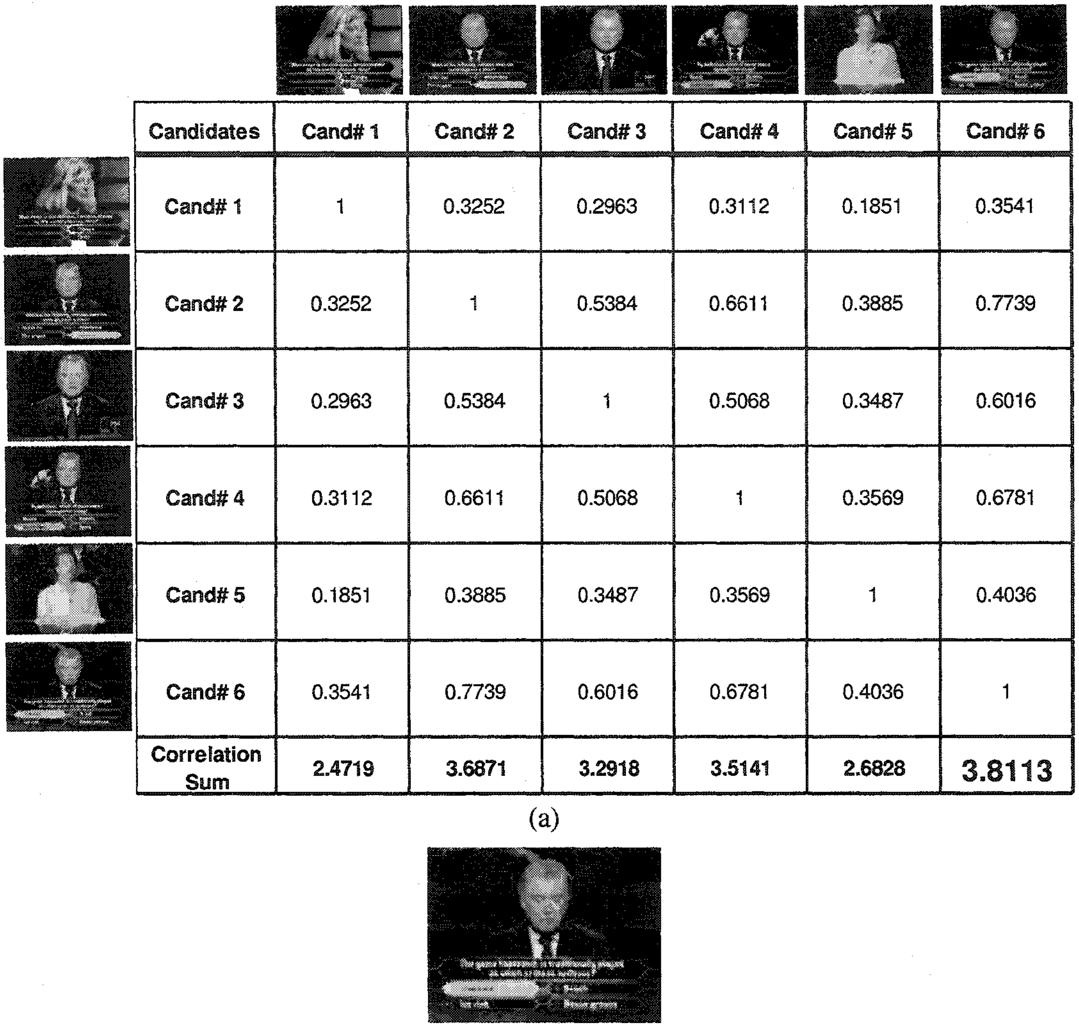


Figure 3.5: Some results of skin detection. White areas in the images show regions where skin is detected.

$$HostID = argmax_r \sum_{c \in allcols} C_{rc} \quad \forall r. \tag{3.4}$$

Figure 3.6 demonstrates the detection of the host for one game show, "Who Wants To Be A Millionaire". Six candidates are picked for the host. Note that of the six candidates, four are shots of the host. The last row shows the summation of correlation values for each candidate. The sixth candidate has the highest correlation sum and is automatically selected as the host. Figure 3.7 shows key host frames extracted for our test videos. Guests are the shots which are non-host. The key host frame is then correlated against key frames of all shots to find all shots of the host. Figure 3.8 shows some guests' key frames found by our algorithm.



Candidates	Cand# 1	Cand# 2	Cand# 3	Cand# 4	Cand# 5	Cand# 6
Cand# 1	1	0.3252	0.2963	0.3112	0.1851	0.3541
Cand# 2	0.3252	1	0.5384	0.6611	0.3885	0.7739
Cand# 3	0.2963	0.5384	1	0.5068	0.3487	0.6016
Cand# 4	0.3112	0.6611	0.5068	1	0.3569	0.6781
Cand# 5	0.1851	0.3885	0.3487	0.3569	1	0.4036
Cand# 6	0.3541	0.7739	0.6016	0.6781	0.4036	1
Correlation Sum	2.4719	3.6871	3.2918	3.5141	2.6828	**3.8113**

(a)

(b)

Figure 3.6: Detection of the host in a game show, "Who Wants To Be A Millionaire". (a) Six candidate shots (b) The shot of the host is correctly identified.

3.5 EXPERIMENTAL RESULTS

The test suite was four full-length "Larry King Live" shows, two complete "Who Wants To Be A Millionaire" episodes, one episode of "Meet The Press", one Pakistani talk show, "News Night" and one Taiwanese show, "News Express". The results were compared with

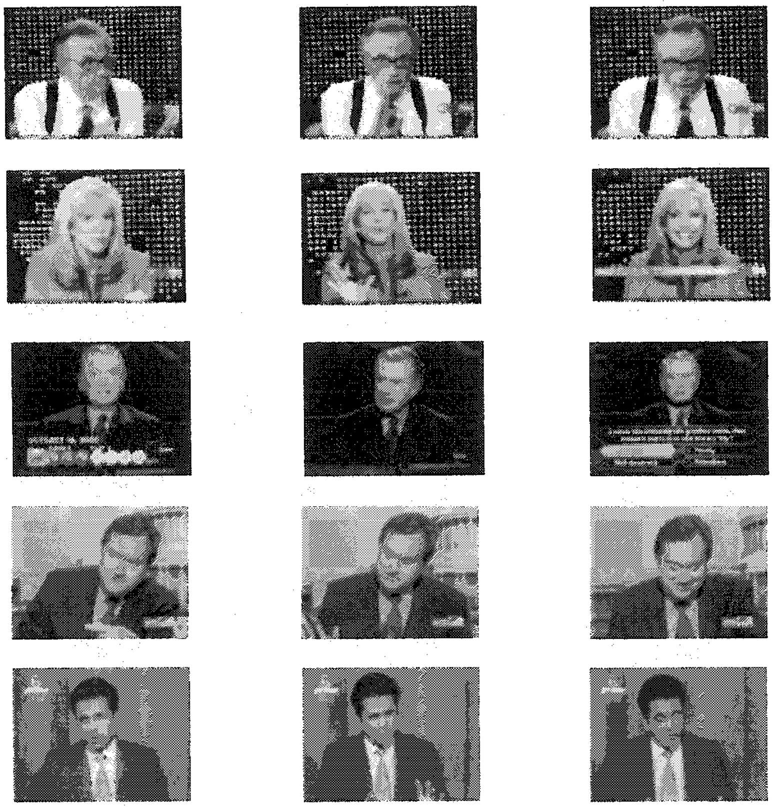


Figure 3.7: Hosts detected for talk and game shows.

the ground truth determined by a human observer, i.e. classifying frames as either belonging to a commercial or a talk show. Table 3.1 shows that the correct automatic classification rate is over 95% for most of the videos. The classification results for "Larry King 3" are not as good as others. This particular show contained a large number of outdoor video clips that did not conform to the assumptions of the talk show model. The overall accuracy of talk show classification results is about the same for all programs, even though these shows have quite different layout and production styles. Figure 3.7 presents some shots that are detected as the hosts. Note that the host of each show is correctly identified. Table 3.2 contains detailed host detection results compared to the ground truth established by a human observer. The second column shows whether the host identity was correctly established. The last column shows the overall rate of misclassification of host shots. Note

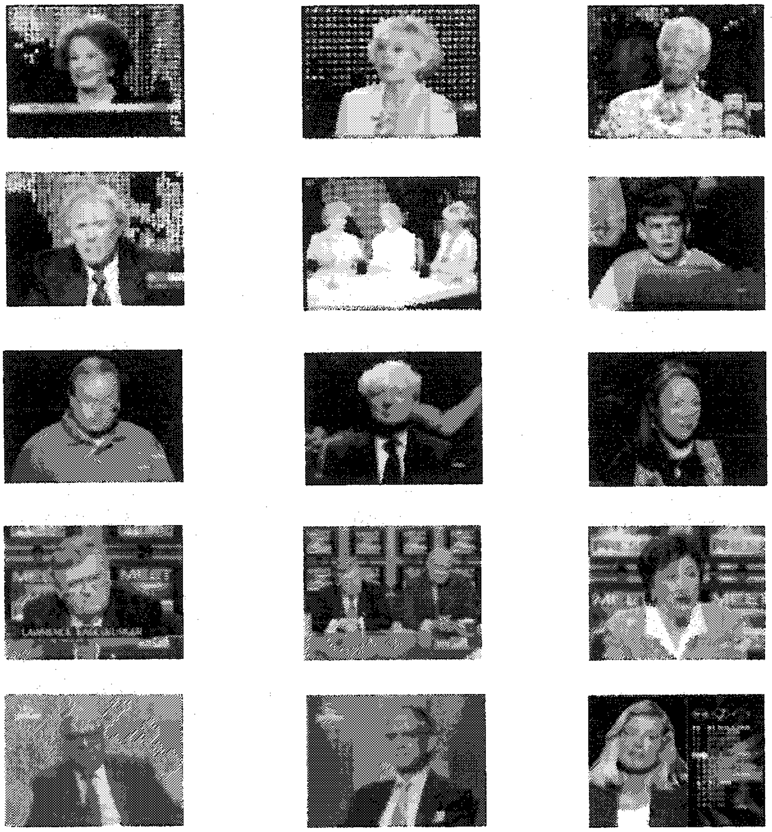


Figure 3.8: Guests detected for talk and game shows.

that for all videos, very high accuracy and precision is achieved by the algorithm. Figure 3.8 are the results of guest shot detection. These are shots which are detected as *non-host* shots.

3.6 SUMMARY

In this chapter, we discussed a framework for automatic characterization of talk and game shows by exploiting the scene structure of these videos. Instead of analyzing the scene content of frames, we relied on the semantics of shot transitions by constructing a shot

Table 3.1: Results of story detection in a variety of videos. Precision and recall values are also listed. Video 1 was digitized at 10 fps. All other videos were digitized at 5 fps.

Show	**Frames**	**Shots**	**Story Segments**		**Recall**	**Precision**
			Ground Truth	**Found**		
Larry King 1	34,611	733	8	8	0.96	0.99
Larry King 2	12,144	446	6	6	0.99	0.99
Larry King 3	17,157	1,101	8	9	0.86	0.99
Larry King 4	13,778	754	6	6	0.97	0.99
Millionaire 1	19,700	1,496	7	7	0.92	0.99
Millionaire 2	17,442	1,672	7	7	0.99	0.99
Meet The Press	32,142	561	2	2	0.99	1.00
News Night (Pakistani)	9,729	501	1	1	1.00	1.00
News Express (Taiwanese)	16,472	726	4	4	1.00	0.92

similarity graph. We segmented the videos into stories and characterized stories into talk and game shows and commercials. We further presented a method to detect the host of each program by analyzing the structure of the SCG. Our work presented in this chapter has been resulted in following publications: [48], [25], [27] and [24]. In the next chapter, we will discuss the problem of classifying feature films into genres by examining their audio-visual features. In our framework, we will combine the computable video features with cinematic principles to provide a mapping to the four broad high-level semantic classes including action, comedy, horror and drama.

Table 3.2: Host detection results. All hosts are detected correctly.

Show	**Correct Host ID ?**	**Host Shot Detection Accuracy**
Larry King 1	Yes	99.32%
Larry King 2	Yes	94.87%
Larry King 3	Yes	96.20%
Larry King 4	Yes	96.85%
Millionaire 1	Yes	89.25%
Millionaire 2	Yes	95.18%
Meet the Press	Yes	87.7%
News Night	Yes	62.5 %

CHAPTER 4

CLASSIFYING FILM GENRES USING COMPUTABLE VIDEO FEATURES

4.1 INTRODUCTION

Films constitute a large portion of the entertainment industry. Every year about 4500 films are released around the world, which correspond to approximately 9,000 hours of video [66]. In this chapter, a framework is presented to identify feature films genres based on visual cues which can be computed from previews. Our work is a step towards high-level semantic film interpretation, using low-level video features and knowledge of ubiquitous cinematic practices.

Films are a means of expression. Directors, actors, and cinematographers use this medium as a way to communicate a precisely crafted storyline. This communication operates at several levels; explicitly, with the delivery of lines by the actors, and implicitly, with the background music, lighting, camera movements and so on. Our current domain of study is, however, the movie preview; the commercial advertisements created primarily to attract audiences. A preview often emphasizes the theme of a film and hence provides suitable information for classification. In our approach, we classify previews into four broad categories: comedies, action, dramas or horror films. Computable video features are combined in a framework with cinematic principles to provide a mapping to these four high-level semantic classes. We have developed two methods for genre classification: (a) a hierarchical method and (b) an unsupervised method. In the hierarchical method, we first classify movies into action and non-action categories based on the average shot length and motion content in the previews. Next, non-action movies are subclassified into comedy, horror or drama categories by examining their lighting key. Finally, action movies are ranked on the basis of number of explosions/gunfire events. In the unsupervised method for classifying movies, a mean shift classifier is used to discover the structure of the mapping between the computed features and each film genre. We have conducted extensive experiments on a large number of film previews and demonstrate that low-level features may be utilized for movie classification. Our approach can also be broadened for many potential applications including scene understanding, the building and updating of video databases with minimal human intervention, browsing and retrieval of videos on the Internet (video-on-demand) and video libraries.

4.2 RELATED WORK

Barnard and Forsyth [5] proposed the idea of a statistical model for organizing image collections that integrated semantic information (provided by associated text) and visual information (provided by image features). The model was demonstrated for information retrieval tasks such as database browsing and searching for images based on text and/or image features. The model learned the relationships between text and image features in an unsupervised fashion for object recognition. Li et al. [34] also statistically modelled the concepts by training on categorized images. Images of any given concept category were regarded as instances of a stochastic process that characterized the category. To measure the extent of association between an image and the textual description of a category of images, the likelihood of the occurrence of the image based on the stochastic process derived from the category was computed. Although the aforementioned works were limited to image analysis only, they provide a foundation for semantic interpretation of videos.

Specific to film classification, Vasconcelos et al. proposed a feature-space based approach in [65]. In this work, two features of the previews, average shot length and shot activity, were used. In order to categorize movies they used a linear classifier in the two-dimensional feature space. An extension of their approach was presented by Nam et al. [38], which identified violence in previews. They attempted to detect violence using audio and color matching criteria. One problem with these existing approaches in film classification is the crude structure that is imposed while classifying data (in the form of the linear classifier). In our work, we adopt a non-parametric approach, using mean shift clustering. Mean shift clustering has been shown to have excellent properties for clustering real data. Furthermore, we exploit knowledge of cinematic principles, presenting four computable features for the purposes of classification. We believe that the *extendibility* of the proposed framework to include new, possibly higher-level features is an important aspect of the work. Since the approach discovers the structure of the mapping between features and classes autonomously, there is no longer a need to handcraft rules of classification.

4.2.1 THE NEED FOR SEMANTIC LABELLING

While it is feasible to classify films at the time of production, classification at finer levels, for instance classification of individual scenes, would be a tedious and sizable task. Currently, there is a need for systems to extract the genre of scenes in films. Application of such scene-level classification would allow departure from the prevalent system of *movie* ratings to a more flexible system of *scene* ratings. For instance, a child would be able to watch movies containing a few scenes with excessive violence, if a pre-filtering system can prune out scenes that have been rated as violent. Such semantic labelling of scenes would also allow far more flexibility while searching movie databases. For example, automatic recommendation of movies based on personal preferences could help a person choose a movie, by executing a scene level analysis of previously viewed movies. While the proposed method does not actually achieve scene classification, it provides a suitable framework for such work.

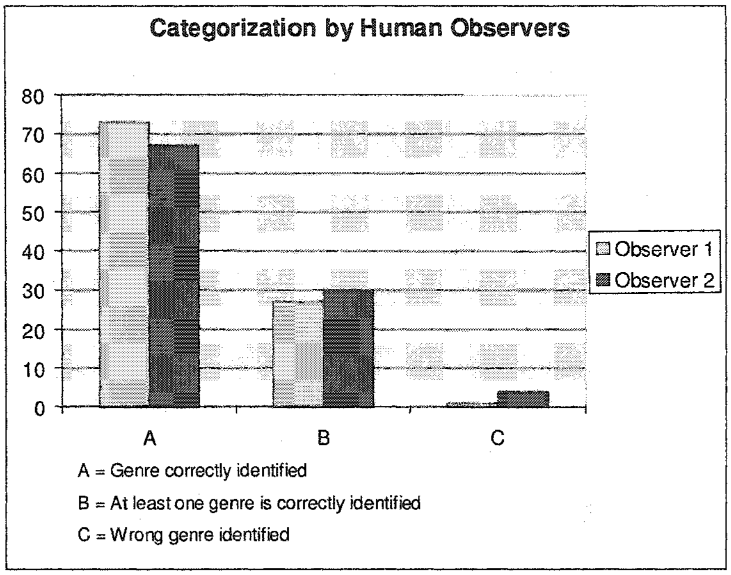


Figure 4.1: Genre classification by human observers with respect to the ground truth obtained from the IMDB and Apple websites. Observers were asked to categorize films into four genres based on their previews.

Some justification must be given for the use of previews for the classification of movies. Since movie previews are primarily commercial advertisements, they tend to emphasize the theme of the movie, making them particularly suited for the task of genre classification. For example, previews of action movies inevitably contain shots of fights, chases and sometimes crashes, explosions and gunfire. Exceptions exist, of course, and in order to strengthen the claim that high-level semantic classification based on previews is *possible*, we conducted human evaluations of our data set consisting of over a hundred film previews and compared the results with the ground truth obtained from IMDB (Internet Movie Database [6]). Two observers were asked to watch the previews and classify each movie into the four genres. Both observers managed to identify at least one of the genres in the ground truth, for practically all the movies. What the experiment suggests then, is that classification based on movie previews is, at the very least, *possible.* The results of the evaluation are displayed in Figure 4.1. In conclusion, we present a framework for genre classification based on computed features from film previews. Since both previews and scenes are composed of several shots, this framework can be suitably extended for applications of scene classification.

4.3 SUITABLE FEATURES FOR GENRE CLASSIFICATION

We present the problem of semantic classification of films within the feature-space paradigm. In this paradigm, the input is described through a set of features that are likely to *minimize* variance of points within a class and *maximize* variance of points across different classes. A parametric representation of each feature is computed and is mapped to a point in the multidimensional space of the features. Of course, the performance depends heavily on the selection of appropriate features. In our approach, we exploit four computable features that provide good discrimination between genres. These four features that are employed for classification are:

- Average shot length
- Average shot motion content
- Color
- Lighting key

Table 4.1 presents a comprehensive list of genres (http://us.imdb.com/Sections/Genres/). From this list we identified four *major* genres, namely action, comedy, horror and drama. There are two reasons for such a classification. Firstly, these genres represent the majority of movies currently produced, and most movies can be classified, albeit loosely, into at least one of these major genres. Secondly, we have selected these four genres since low-level discriminant analysis is most likely to succeed at distinguishing between these genres. However, the data set itself was not pre-screened to fit specifically into one of these genres. As the subsequent results will show, many movies fit more than one of the categories. Rather than espouse individual genre classification, we acknowledge the fact that a film may correctly be classified into *multiple* genres. For instance, many Hollywood action films produced these days have a strong element of comedy as well. In the next section, we present the hierarchical classification of feature films. In a later section, we will demonstrate the clustering of previews in the feature space which will be able to exploit the inherent similarities within the genres in the feature space.

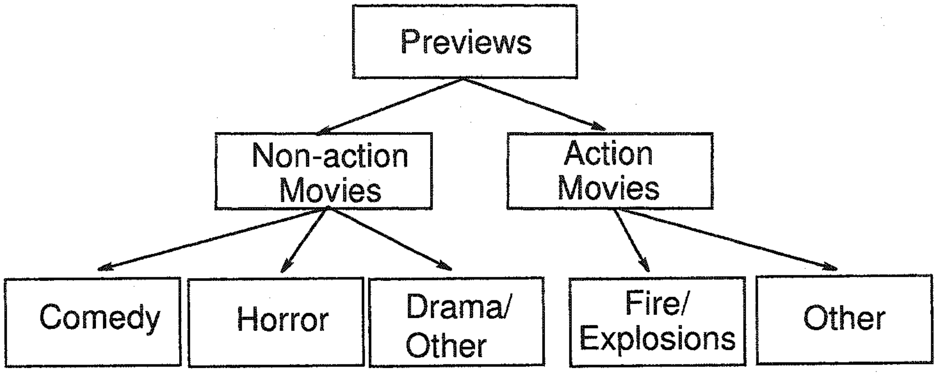


Figure 4.2: Flow chart showing the hierarchy of movie genres with our proposed approach.

4.4 HIERARCHICAL METHOD OF CLASSIFICATION

In the hierarchical method of movie genre classification, the movies are initially divided into two categories, namely *action* and *non-action.* Two computable features are analyzed to obtain this classification; average shot length and average shot motion content. A linear classifier is used to separate the movies in the feature space. In the next step, the key frames of all non-action movies are examined and movies are subclassified into comedy, horror and drama. In the final step, action movies are subclassified into *explosion/fire* and *other-action* categories. This is done by first analyzing the audio information to locate

Table 4.1: Some popular Hollywood genres. A feature movie may belong to one or more genres.

Hollywood Genres	
(i)	Action
(ii)	Adventure
(iii)	Animation
(iv)	Comedy
(v)	Crime
(vi)	Documentary
(vii)	Drama
(viii)	Family
(ix)	Fantasy
(x)	Film Noir
(xi)	Horror
(xii)	Musical
(xiii)	Mystery
(xiv)	Romance
(xv)	Science Fiction
(xvi)	Short
(xvii)	Thriller
(xviii)	War
(xix)	Western

possible explosion events and then processing the corresponding video frames to detect such events and to remove outliers. This genre hierarchy is shown in Figure 4.2.

4.4.1 INITIAL CLASSIFICATION

Non-action movies, such as *drama*, contain mostly dialogue shots. Actors as well as cameras undergo relatively smaller movements in this kind of scenario. On the other hand, previews of action movies have several shots of fights and chases with higher dynamics. In addition, camera motion is often very large and haphazard, which results in higher shot motion content. Another distinctive property of these two classes is that in *action* movies shots change more rapidly than the other class. This is due to the fact that action scenes are fast paced and camera cuts are frequent. However, dialogue shots are longer and, therefore, the *average shot length* is greater. For the initial classification, the average shot motion content is plotted against average shot length as shown in Figure 4.3. Please note that the distribution of movies used in the data set also satisfies this trend. For example, the movie "The Princess Diaries" (21), has the smallest average motion content since most of the shots are static and with many frames in them. On the other hand, "Rush Hour" (15), with the highest motion content, appeared on the rightmost side of the x-axis. It should also be noted that the spread of movies on the 2D grid forms three distinct clusters; the first in the upper left corner, the second is in the bottom right portion of the space and the third is in the middle of the plot. We used the k-means method to cluster the points in 2D space. It is found that movies in cluster 3 (see Figure 4.3) are strictly action movies; that is the dominant genre of these movies. On the other hand, clusters 2 and 3 contain movies with mixed genres. With this observation, a linear classifier is used which separates action and non-action movies.

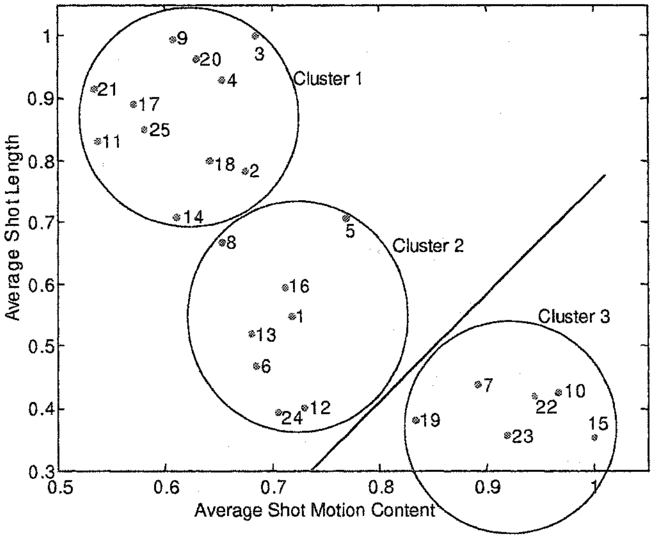


Figure 4.3: The distribution of movies on the basis of average motion content and average shot length. Three clusters obtained by the k-means method of data clustering. Cluster 3 contains movies that belong to the action genre. Clusters 1 and 2 represent movies with mixed genres. A linear classifier separates movies into action and non-action categories.

4.4.1.1 Subclassification of Non-action Movies

As discussed in Section 2.2.6, the perceptual effects of color on human emotions are well known and we usually form specific associations with particular colors. In this section, we use this information for subclassification of non-action movies. In general, previews contain many important scenes from the movie. Directors pick the shots that emphasize the theme and put them together to make an interesting preview. For example, in *horror* movies, the shots are mostly *low key* to induce fear or suspense. On the other hand, *comedy* movies tend to have a greater number of *high key* shots, since they are less dramatic in nature. To exploit this information we consider all key frames of the preview in the gray scale space and analyze the computable feature, lighting key ζ, of the entire video clip (see Section 2.2.6 for details).

Our experiments show that we can distinguish between different genres of non-action movies using lighting as explained below.

Comedy: Movies belonging to this category have a gray scale mean near the center of the gray-scale axis, with a large standard deviation, indicating a rich mix of colors in the movie. This results in higher ζ.

Horror: Movies of this type have a mean gray scale value towards the dark end of the axis, and have low standard deviation. This is because of the frequent use of dark tones and colors by the director resulting in smaller ζ.

Drama/other: Generally, these types of movies do not have any of the above distinguishing features.

Based on these observations, we define a scheme to classify an unknown movie to one of these three types by selecting two thresholds, τ_c and τ_h. A category is assigned to each movie i based on the following criterion:

$$L(i) = \begin{cases} Comedy & \zeta_i \geq \tau_c \\ Horror & \zeta_i \leq \tau_h \\ Drama/Other & \tau_h < \zeta_i < \tau_c \end{cases} \tag{4.1}$$

4.4.2 SUB-CLASSIFICATION WITHIN ACTION MOVIES

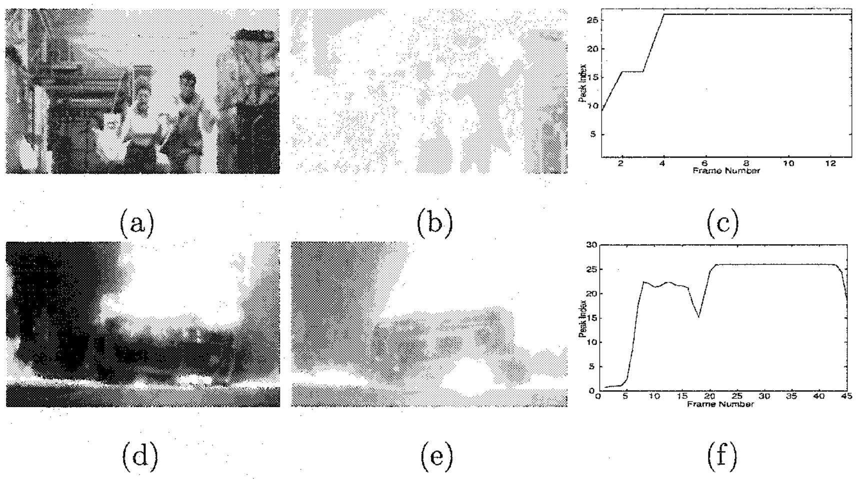
Action movies can be subclassified as martial art, war, sci-fi or violent depending on the second dominating genre. In this section, action movies are further rated on the number of fire/explosions present in the previews. Several other attributes are also crucial for rating a movie. For example, offensive language and/or profanity. Here, our focus is to identify the explosions as one of the criteria to discriminate among action movies. Although this attribute may not map directly to a semantic concept (violence being one of them), however, it can be used as one of the discriminating features. For example, a movie with many gunfire/explosions may not be suitable for young kids. The next section explains the detection process that utilizes both audio and color information from the previews.

4.4.2.1 Audio Analysis

Music and non-literal sounds are often used to provide additional energy to the scene. It can quite easily describe a condition, for example, whether a situation is stable or unstable. In movies, the audio is usually correlated with the interesting events. For example, the shots of fighting, explosions, etc. are mostly accompanied with a sudden change in the audio level. Therefore, the energy in the audio track is computed to locate a possible occurrence of such events (see Section 2.2.7). Since our interest is in the instances where the energy in audio changes abruptly, a peakiness test is performed on the energy plot. A peak is good if it is sharp and deep. Video frames corresponding to the peaks more than a threshold are selected to process the corresponding video.

4.4.2.2 Fire/explosion Detection

Once the occurrence of *events* in the movie are detected, the corresponding frames of video are analyzed to detect fire and/or explosions. In such cases, there is a change in the intensity of the images in the video from low to high. The gray level histogram of each frame within the shot boundary of shot as identified by the peakiness test is computed. Each histogram is 26 bins wide. The index of the bin with the maximum number of votes is then plotted against time. During an explosion, the shot shows an increase in the intensity and the gray level of the pixels changes from lower intensity to higher intensity values and the peak of the histogram shifts from a lower index to a higher index. A shot that fulfils this criteria is labelled as an *explosion/gunfire* shot. A camera flash, which does not last for more than a few frames, might be considered as an explosion. Therefore, shots that show low stability in the plot are excluded. Figure 4.4 shows the detection in two candidate shots that are successfully identified as *explosion* shots. Figure 4.5 shows two shots that are successfully identified as *non explosion shots.* Although several video clips had abrupt changes in audio in our experiments, our algorithm successfully differentiated between explosions and non-explosion shots. See Table 4.2 for the recall/precision of the algorithm on the movies in the database.



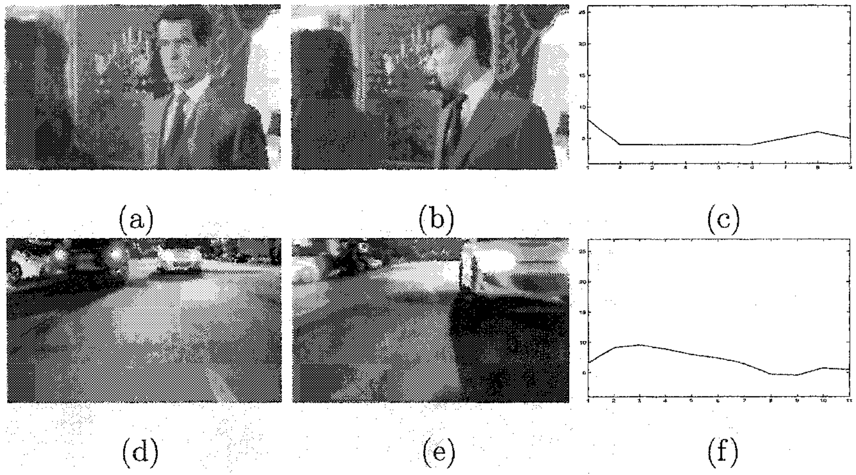
(a) (b) (c)

(d) (e) (f)

Figure 4.4: Detection of *fire/explosion* in two shots. (a) and (b) are two frames of one shot. (c) the plot of the index of the histogram peak against time. (d) and (e) are two frames of another shot (f) the plot of the index of the histogram peak against time. Both shots were successfully identified as *fire/explosion.*

4.4.3 EXPERIMENTAL RESULTS

Experiments with previews of 25 Hollywood movies were conducted to test the algorithm. These previews were obtained from Apple's website [3]. For each preview, video tracks were analyzed at a frame rate of 24 Hz and a resolution of 120x68. The audio tracks were processed at 22 KHz using a 16-bit precision.



(a) (b) (c)

(d) (e) (f)

Figure 4.5: Detection of *fire/explosion* in two shots. (a) and (b) are two frames of one shot. (c) the plot of the index of the histogram peak against time. (d) and (e) are two frames of another shot (f) the plot of the index of the histogram peak against time. Both shots were successfully identified as *non fire/explosion.*

Figure 4.3 shows the distribution of movies on the feature plane by plotting the average motion content against the average shot length. Movies with more action tend to have shorter average shot length. On the other hand, comedy/drama movies have less action content and longer shots. A linear classifier separates the two aforementioned classes of movies. This linear classifier works pretty well for most of the titles present in the data set. However, its accuracy is sensitive to the way a preview is edited (called montage). For example, the movies "Rush Hour" and "The Tuxedo" are action/comedy. However, the previews contain many shots from fights and few dialogue shots. Another counterexample of this trend is the movie "Ali", which is being labelled as action/drama on the official website and is characterized as a non-action movie by this method. This movie is about the life of the boxer, Muhammad Ali. A couple of shots have been shown while he is in the ring, however, a greater portion of the preview emphasizes on his personal life. As a result, this movie stands together with slow paced movies in the feature space.

Table 4.2: Results of fire detection algorithm.

Fire/Explosion Detection						
Movie	Gnd. Truth	Found	False+ Positive	False-ve Negative	Recall	Precision
The World Is Not Enough	10	12	4	2	0.8	0.67
Fast and Furious	3	5	2	0	1	0.6
The One	4	5	2	1	0.75	0.6
Kiss Of The Dragon	2	4	2	0	1	0.5
Tuxedo	2	3	1	0	1	0.67
Rush Hour	2	1	0	1	0.5	1

In the next step of subclassification of movies in the non-action group, the intensity distribution of key frames of previews is used. Two thresholds were set by observation; $\tau_c = 0.55$ and $\tau_h = 0.4$. The classification task performed quite well for 19 movies classified as non-action in the first step (see Table 4.3 and Figure 4.6). Readers would appreciate that this criterion, although very simple, works quite well as far as the dominating genre of each movie is concerned. For example, "Dracula", "Red Dragon", "Sleepy Hollow", "The Others" and "What Lies Beneath" were correctly classified as horror movies. Previews of these movies contain several low key shots to express the horror and fearfulness of the movies. Some of these movies are also labelled as thriller, however, the feeling of thrill is built up in the audience by a sequence of shots over time. It is very difficult to identify this attribute by a few incomplete shots joined together in the previews. We believe that the semantic interpretation of this attribute requires more information than the previews provide, for example, a sequence of shots from the movie.

The experimental data set also contained several comedy movies including "American Pie", "Big Trouble", "Legally Blonde", "Road Trip", "Stealing Harvard" and "The Princess Diaries". With many high key shots in the previews, all of these movies were successfully identified as comedy. There is one misclassification, the movie "Mandolin", which is also marked as a comedy. In fact, this movie is a drama/romance/war according to its official

Table 4.3: Sub classification of non-action movies. C = Comedy, H = Horror and D = Drama/Other. Thresholds used for classification are $\tau_c = 0.55$ and $\tau_h = 0.4$.

Subclassification of Non-action Movies		
Movie	$\zeta(\mu, \sigma)$	**Genre Found**
24 Hours Party People	0.84	C
American Pie	0.72	C
Ali	0.43	D
Americas Sweethearts	0.76	C
Big Trouble	0.62	C
Dracula	0.23	H
Hannibal	0.53	D
Jackpot	0.44	D
Legally Blonde	0.82	C
Mandolin	0.69	C
Red Dragon	0.38	H
Road Trip	0.60	C
Sleepy Hollow	0.17	H
Stealing Harvard	0.67	C
The Hours	0.44	D
The Others	0.36	H
The Princess Diaries	0.57	C
What Lies Beneath	0.27	H
What Women Want	0.42	D

website. The only cue used here is the intensity images of key frames and several high key shots deceived our algorithm. We expect that by incorporating further information, such as the audio, the accuracy can be improved.

Action movies are sorted on the basis of the number of fire/explosion shots. Table 4.2 lists the accuracy of our algorithm. The number of alarms (the correct ones plus false positives) are used to sort movies. Figure 4.6 shows the movies sorted in decreasing order based on the number of fire/explosions detected in their previews. This figure indicates that the movie "The World Is Not Enough" contains the highest number of explosions/gunfire, whereas the movie "Rush Hour" contains the least number of detected events. Hence, a conclusion can be reached that the former movie might not be suitable for young children. With the hierarchical classification, each movie is classified with only one genre. However, it is not unusual to find a feature movie with more than one semantic label. For example, movies labelled as action/comedy or comedy/drama are very common in Hollywood film industry (see Figure 4.7 which shows a pie-chart of genre membership of over a hundred Hollywood movies). Thus, there is a need to develop classifiers that can exploit correlation in the feature space within and over the genres. In the next section, we propose the use of a mean shift classifier that captures the inherent similarities in low level features of movie previews.

4.5 UNSUPERVISED METHOD USING MEAN SHIFT CLASSIFICATION

In the previous section, we discussed the classification of movies into genres in a hierarchical fashion. However, the relevance of various low-level features of video data to the high-level semantics suggests a formulation based on feature-space analysis. The analysis of the feature-space itself is a critical step that determines both the effectiveness and the practicality of the method. Even with a highly discriminating feature space, if the anal-

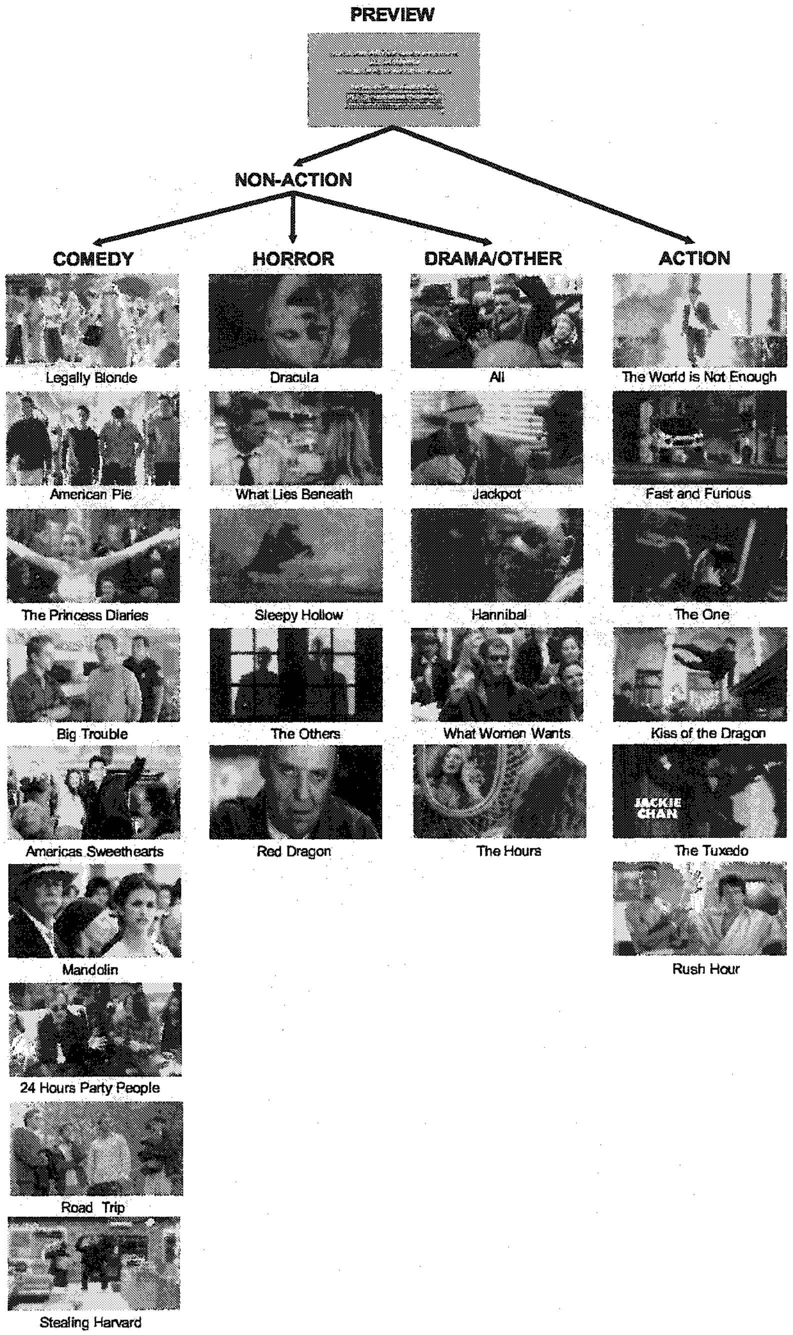


Figure 4.6: Classification of Movies. Note that action movies are sorted according to the fire/explosion content in the previews.

ysis is rule-based or imposes an unwarranted structure on the data (e.g. linear classifiers discussed earlier, elliptical shape, etc.) the possibility of extending or deploying the work becomes suspect. Extendibility, in particular, is a central aspect of this work, as an interdependent, low-to-high level analysis towards semantic understanding. Although a multitude of techniques exist for the analysis of feature spaces (see [23] for a recent survey), most are unsuited for the analysis of real data. In contrast, the mean shift procedure has been shown to have excellent properties for clustering and mode-detection with real data. An in-depth treatment of the mean shift procedure can be found in [11]. Two salient aspects of mean shift based clustering that make it suited to this application is its ability to automatically detect the number of clusters, and the fact that it is non-parametric in nature (and as a result does not impose regular structure during estimation). Since the four-dimensional feature space is composed of the lighting key, average shot length, motion content and color variance, we employ a joint domain representation. To allow separate bandwidth parameters for each domain, the product of four univariate kernels define the multivariate kernel, that is

$$K(\mathbf{x}) = \frac{C}{h_1 h_2 h_3 h_4} \prod_{i=1}^{4} k\left(\frac{x_i^2}{h_i}\right), \tag{4.2}$$

where x_i, $i = 1$ to 4, corresponds to the average shot length, color variance, motion content and lighting key respectively, h_1 to h_4 are the bandwidth parameters for each feature and C is the volume of unit *four-dimensional* sphere. A normal kernel is used, giving a mean shift vector of

$$\mathbf{m}_{n,N}(\mathbf{y}_j) = \mathbf{y}_{j+1} - \mathbf{y}_j = \frac{\sum_{i=1}^{n} \mathbf{x}_i exp\left(\|\frac{x-x_i}{h}\|^2\right)}{\sum_{i=1}^{n} exp\left(\|\frac{x-x_i}{h}\|^2\right)} - \mathbf{y}_j. \tag{4.3}$$

Mean shift clustering provides a means to analyze the feature space without making arbitrary assumptions, and lets the data define the probabilities of membership, so to speak. This formulation enables us to examine how well the computable features discriminate between the high-level labels known *a priori*. As a result, an exemplar based labelling system is also facilitated, since if there are consistent clusters and the label of one within the cluster is known, labels can be assigned to the rest.

4.6 EXPERIMENTAL RESULTS

We have conducted extensive experiments on just over a hundred film previews. These previews were obtained from the Apple website [3]. Our experiments show interesting structure within the feature space, implying that a mapping does indeed exist between high-level classification and low-level computable features. We identified four major genres, namely action, comedy, horror and drama. We will first present our data set and the associated ground truth, followed by experimental results and discussion.

To investigate the structure of our proposed low-level feature space, we collected a data set of 101 film previews, the ground truth of which is graphically displayed in Figure 4.7. As mentioned earlier, classifying movies into binary genres is unintuitive, since modern cinema often produces films with more than one theme (presumably for both aesthetic and commercial reasons). Thus, we study multiple memberships both within the ground truth and the output of the proposed method. We performed mean shift classification over all the data points in the feature space, and studied the statistics of each cluster that formed. In the following discussion, we refer to the ground truth genres as *labels* and the cluster genres as *classes.*

The data formed 6 clusters in the four-dimensional feature space, the analysis of which is displayed in Figure 4.8. Each cluster was assigned the label of the 'dominating genres' in the cluster. We analyzed each cluster formed, counting number of films (1) with all genres correctly identified (2) at least one genre correctly identified and (3) no genre correctly identified. The first (and largest) cluster that was identified was the *action-drama* cluster, with 38 members. Although, only five movies were labelled *action-Dramas* in the ground truth, *all* five appeared within this cluster. Moreover, the remaining points within this

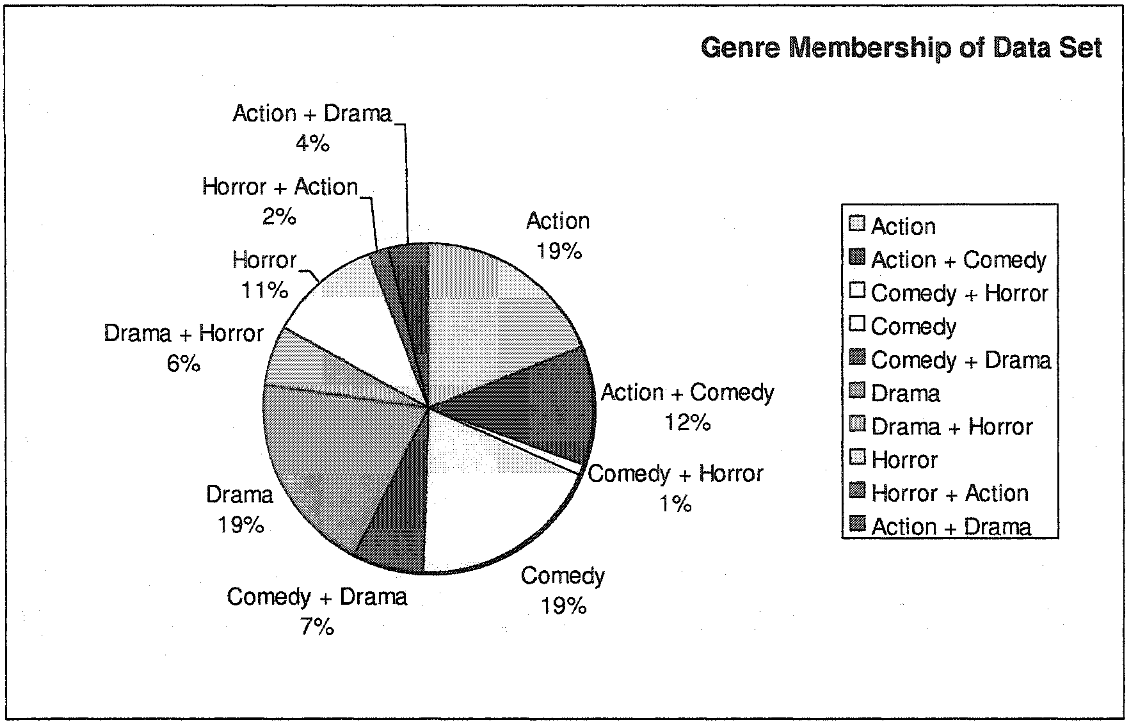


Figure 4.7: Genre membership of data set. It should be noted that some films have a second genre, as well.

cluster were composed of ten films labelled as action films, and six films labelled as dramas. Eleven films with at least one genre labelled as drama or action were also observed in this cluster. The majority of the outliers (five out of six) came from the horror genre. The dominating genre in the second cluster was drama, with nine members. Nineteen films were labelled dramas in the ground truth, and eight of them were classified in this cluster. Only one outlier was observed within this cluster, "Darkness Falls", which was labelled horror in the ground truth. The third cluster was classified as *comedy-drama*, with nineteen members. Seven films were initially labelled as comedic dramas in the ground truth, and four of these seven were classified in this cluster. The cluster contained eight films labelled as comedies and two films labelled as dramas. The only outlier was the horror film, "Session 9". The fourth cluster, classified as comedy, contained the highest

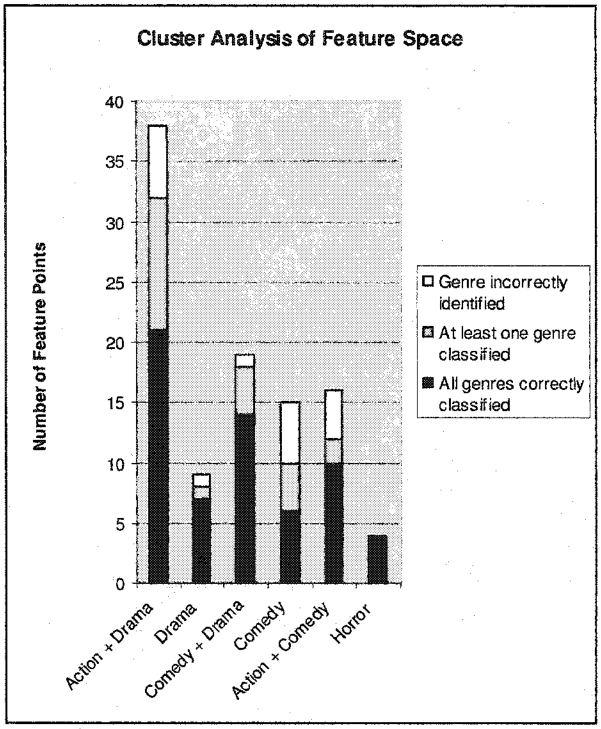


Figure 4.8: Cluster Analysis of Feature Space. Six clusters are observed in the data and each cluster is classified by its dominating genre.

percentage of outliers. Of the fifteen films in the cluster, six were labelled comedies, four had at least one genre labelled as comedy, and five were incorrectly identified. The fifth cluster was classified as Action and Comedy and had a population of sixteen. Four films in this cluster were labelled as *action-comedies*, five were action movies, and one was a comedy. In the last cluster, classified as *horror*, we had four horror films grouped together. This small cluster can be seen as the only successful cluster of horror films, showing that while our features are not sufficiently discriminating for *all* horror films, it captures *some* of the structure that exists. Since our feature space is four dimensional, we cannot visually display the clustering. In order to give the reader some feeling of the results, Figure 4.9 displays the *profile* of each feature. The films are indexed according to their association with each cluster. Figure 4.10 shows some movie previews with their associated cluster.

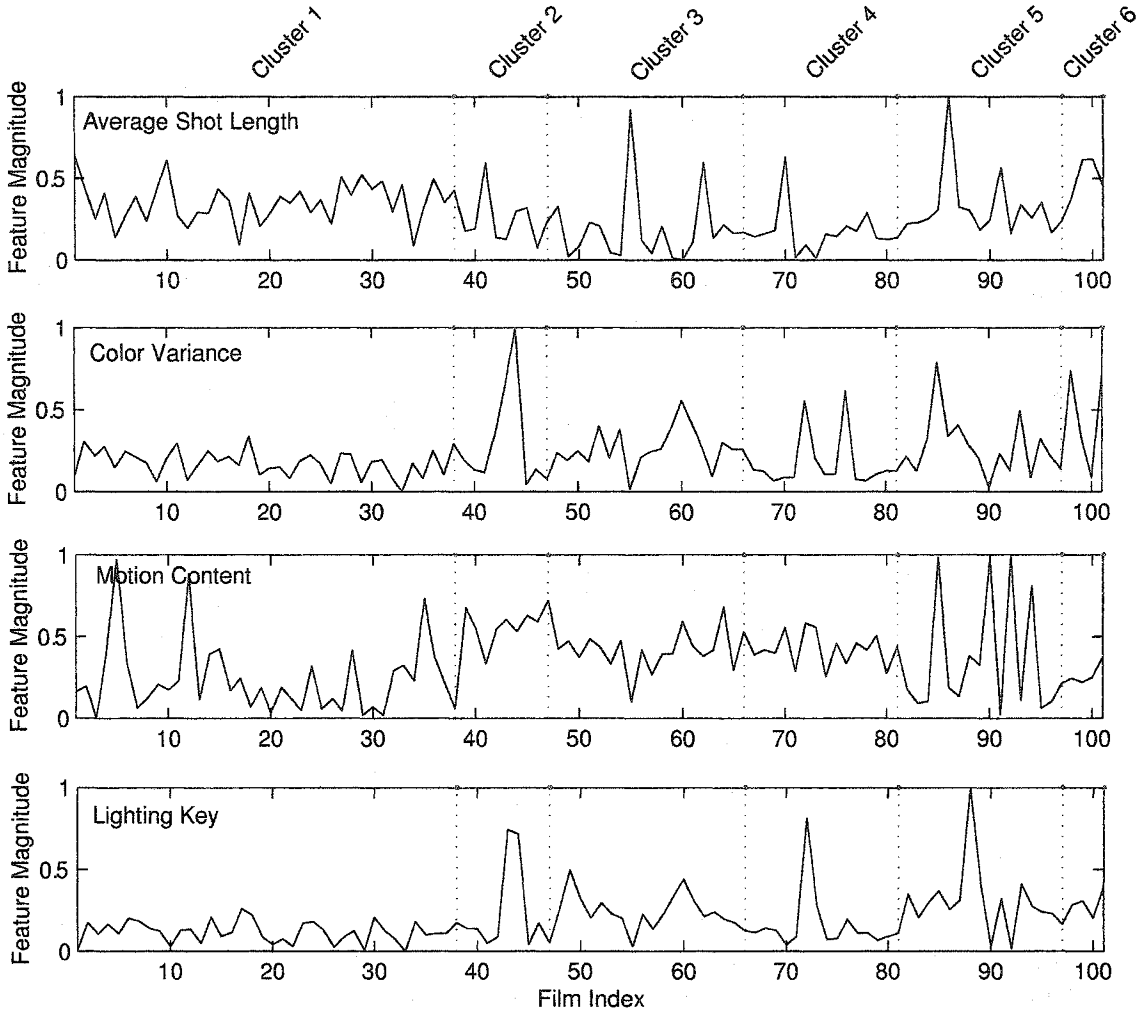


Figure 4.9: Profiles of each feature. The films are indexed according to their association with each cluster.

The total number of outliers in the final classification was 17 (out of 101). While this number cannot be interpreted as an 83% genre classification accuracy, it strongly supports the claim that a mapping exists between low-level video features and high-level film classes, as predicted by film literature. Thus, this domain provides a rich area of study, from the extension and application of this framework to scene classification, to the exploration of higher-level features. And as the entertainment industry continues to burgeon, the need for

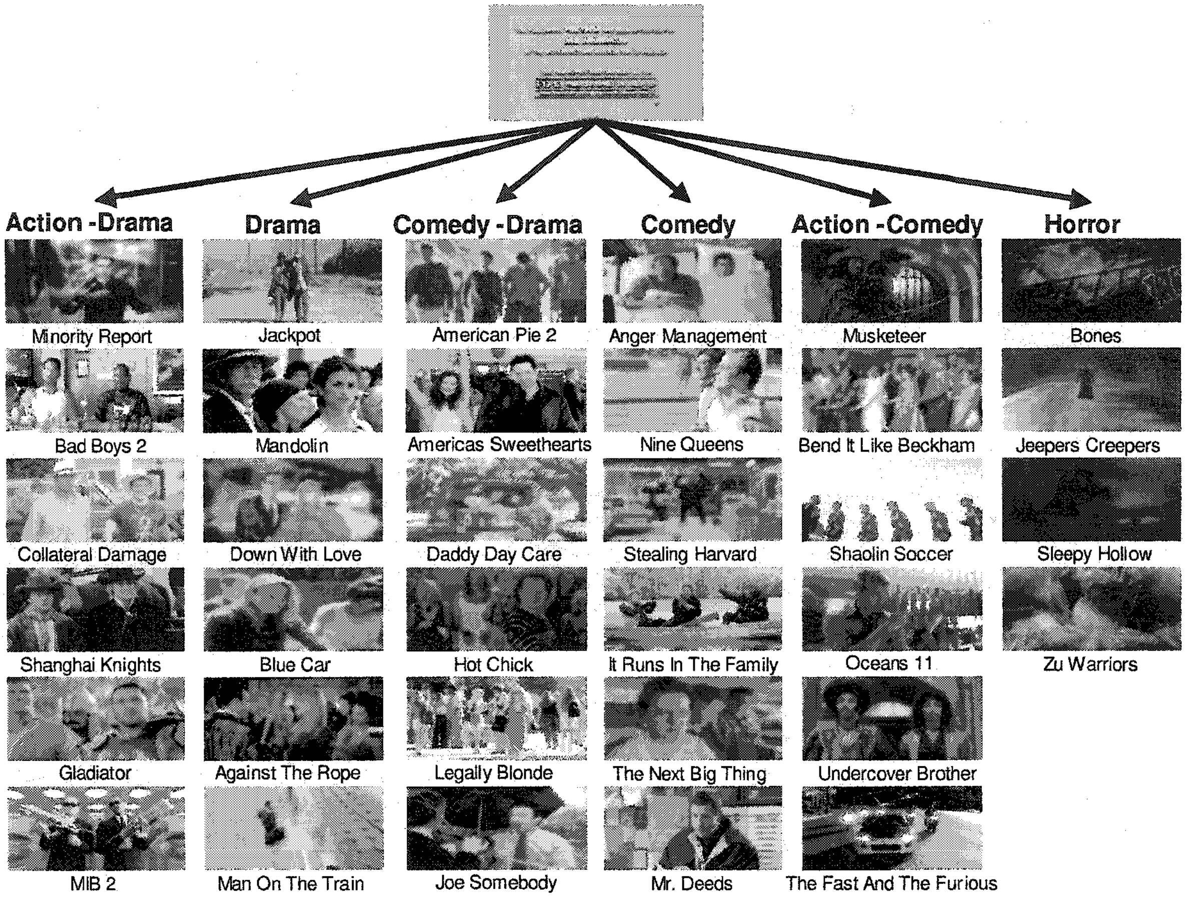


Figure 4.10: Classification of movie previews using Mean shift. Only some of the movies from each group are shown.

efficient classification techniques is likely to become more important, making automated film understanding a necessity of the future.

4.7 SUMMARY

In this chapter, we proposed two methods to perform high-level classification of previews into genres using low-level computable features. We demonstrated that the combination of visual cues and cinematic principles provides powerful tools for genre categorization. Classification is performed in the four-dimensional feature space of average shot length, color variance, average shot motion content and lighting key. We discussed the clustering thus obtained and its implication. We plan to extend this work to analyze complete movies and to explore the semantics from the shot level to the scene level. We also plan to utilize the grammar of movie making to discover the higher level description of the entire stories. Furthermore, we are interested in developing computable features for mid-level and high-level information, as an interdependent multi-level analysis is envisaged. The ultimate goal is to construct an autonomous system capable of understanding the semantics and structure of films, paving the way for many *intelligent* indexing and post-processing applications. The research work in this chapter has been published in [45], [50] and [49](under review process). In the next chapter, we discuss methods for temporal segmentation of produced videos. We will exploit the structure of produced videos in terms of low-level features (color and motion) and cluster shots to make scenes. We will also provide a framework to detect the frames that best represent the content of segmented scenes.

CHAPTER 5

SCENE SEGMENTATION AND REPRESENTATION OF PRODUCED VIDEOS

5.1 INTRODUCTION

There are several possible ways to segment and represent videos. A basic approach is to detect the shots and use a set of key frames to represent the shot content. The second and higher level of abstraction is to combine similar shots together to form *scenes* or *story units.* The organization of videos in this fashion is more meaningful than presenting the shots alone. Recently, DVDs are being made with options to view a particular scene in the movie. To obtain such a representation, a human observer is required to watch the video sequentially and locate the important boundaries or *scene edges.* However, a manual content analysis is not feasible for large amount of data as it is slow as well as expensive.

In this chapter we present two novel approaches for identifying scene boundaries in a variety of videos, including Hollywood movies, sitcoms and talk shows. The first approach is a two-pass algorithm. In the first pass, shots are clustered by computing *Backward Shot Coherence* (BSC); a shot color similarity measure that detects *Potential Scene Boundaries* (PSBs) in the videos. In the second pass we compute *Scene Dynamics* (SD) for each scene, a function of shot length and the motion content in the potential scenes. In this pass, a scene merging criterion is used to remove weak PSBs in order to reduce oversegmentation.

In the second approach, we cluster shots into scenes by transforming this task into a graph partitioning problem. This is achieved by constructing a weighted undirected graph called a *shot similarity graph* (SSG), where each node represents a shot and the edges between the shots are weighted by their similarities. Both color and motion information are utilized to compute shot similarities. The SSG is then split into subgraphs by applying the *normalized cut* technique for graph partitioning. The partitions so obtained represent individual scenes in the video. The use of normalized cuts ensures the maximization of intra-subgraph similarities within a scene and minimization of inter-subgraph similarities across the scenes. While clustering the shots, we consider the global similarities of shots rather than the individual shot pairs.

We further extend the framework to automatically detect the best representative shot of identified scenes. With this approach, we are able to obtain a compact representation of huge videos in a small number of key frames. The segmentation of video data into scenes also facilitates improved browsing of videos in electronic form, such as video on demand, digital libraries, and the Internet.

5.2 RELATED WORK

A large amount of work has been reported to structure videos resulting in several interactive tools to provide navigation privileges to the viewers. Some examples are "Virage Video Engine" [18], "Video Zoom" [59], [60] and [12]. Yeung et al. [71] proposed a graphical representation of videos by constructing a *scene transition graph*, STG. Each node in an STG represented a shot, and the edges represented the transitions between the shots based on the visual similarity and temporal locality. The STG was then split into several subgraphs using the *complete-link* method of hierarchical clustering. Each subgraph satisfied a similarity constraint based on color and represented a *scene*. Hanjalic et al. [19] used a similar approach for shot clustering using a graph and found *logical story units*. The linking of shots was done by defining an *inter-shot dissimilarity measure* among the shots. Their method used MPEG compressed video sequences and utilized DCT images. The number of key frames for each shot varied with the dynamics of that shot. Each shot was represented by combining all key frames within the shot. An average distance between the shots was computed in the $L^*u^*v^*$ color space and thresholded empirically. This determined whether or not two shots were part of one logical story unit. If two shots were found to be similar, all the shots in between them were also merged to construct one logical story unit. Javed et al. [27] proposed a framework for the segmentation of interview and game show videos. Their method automatically removed commercials and detected hosts and guests by analyzing the structure of the video and constructing a *shot connectivity graph* of videos.

Ngo et al. [40] proposed a motion based approach to represent shots and to cluster similar shots to form scenes. Spatio-temporal slices of video sequences were constructed and local orientation of pixels were computed using structure tensors. Using a non-linear histogram, the dominant pixel motion was estimated and pixels were identified as either background or foreground pixels. The motion patterns of spatio-temporal slices were further analyzed to determine if a mosaic construction was possible for shot representation. Each shot was then represented by key frames or by constructing mosaics. Finally, shots were clustered together by analyzing the shot similarities computed by using color histogram intersection of key frames and/or mosaics. A time constrained grouping of shots was conducted to find the scene boundaries. In their approach, the motion information was used to separate camera motion from local motion identifying the foreground, however, the motion information was not used as a cue for finding similarities among the shots. Aner et al. [2] also proposed a mosaic based approach for shot representation and scene clustering. Mosaics were created by registering frames of each shot and by eliminating the moving objects. These mosaics with only the background of the scenes were used to cluster scenes into physical settings within an episode of a video program as well as across episodes of the same program. A similarity test was performed among representative mosaics using the *rubber-sheet* matching algorithm. Their experiments were conducted on sitcoms for a hierarchical representation of videos by clustering scenes according to the physical location. For sports videos, they applied the same method to classify shots and detected repeating characteristic events. Many genres of videos, such as feature movies, are not limited to fixed physical settings. Proper mosaic construction and matching is therefore difficult to achieve.

Rui et al. [55] proposed the construction of a table-of-contents for videos. A time-adaptive grouping of shots was done by finding the visual similarities between them. This similarity was a function of color and activity features of the shots, weighted by their temporal locality. Shots were merged together to form groups by defining a *group threshold* and groups were merged together to form scenes by defining a *scene threshold.* They also suggested a method to determine these thresholds automatically. Recently, Zhou et al. [73] exploited film editing techniques and discovered that certain regions in the video frame are more robust to noise for computing shot similarities. The clustering method was, however, similar to [19]. The improvement made in their work was the use of different frame regions for establishing links between the shots. While [73] reported slightly better performance than [55], both approaches fail to capture the global similarities of shots in scenes. They find the scene boundaries by detecting the strongly connected components in the graphs, which is based on a one-to-one shot similarity criterion. We, on the other hand, exploit the scene structure on the whole and avoid boundary detection based on a single shot-pair criterion.

Adams et al. [1] parsed videos by computing the *tempo* in feature movies. Their approach was inspired by the existing cinematic conventions known as film grammar. Camera motion parameters and the shot lengths in the video were used to construct a tempo plot. The proposed method detected edges in the tempo function and identified instances where the tempo of the movie changed with time. This information was further used as a cue for detecting story sections and events. However, their method did not detect logical boundaries between the scenes in which the tempo was consistent. We believe that the visual similarities of shots can be combined with the motion and shot length features to improve the temporal segmentation of videos.

5.2.1 LIMITATIONS OF EXISTING METHODS

The idea behind the graphical representation is to detect the scene boundaries by first splitting the complete video into shots and then finding the similar shots by comparing the color properties of key frames ([71, 19, 27, 40]). A threshold is determined and a binary decision is made whether or not two shots belong to one scene. In many videos the shots within one scene have similar color contents, for example, videos made inside a studio by stationary cameras. This is true for news videos, talk shows, game shows, interview programs, sitcoms and many others. Feature movies, however, are often filmed in open and dynamic environments using moving cameras and have continuously changing contents. Furthermore, directors use different camera techniques and effects, which make it difficult to create suitable *shot transition graphs* for scenes. These scenes may include non-action scenes such as dialogue and conversation scenes, as well as action such as fighting and chasing scenes. In action scenes, the shot transition rate is very high and the visual contents of shots change rapidly. Therefore, the color similarity criterion is not suitable for this situation and results in an oversegmented graph. On the other hand, if an incorrect match is found between two different scenes, both scenes are merged together and an undersegmented result is generated. Therefore, two major problems are encountered:

- A false color match between shots of two different scenes may wrongly combine the scenes (and the intermediate scenes) into one segment causing an undersegmentation.
- Action scenes may be broken into many scenes for not satisfying the color matching criterion producing an oversegmentation.

The first problem is graphically shown in Figure 5.1. Consider two scenes, 1 and 2. Scene 1 consists of shots A and B, and scene 2 consists of shots C, D and A'. If shot A happens to be similar to any shot in scene 2, such as A', an erroneous link will be formed between scene 1 and 2 (indicated by a dotted link). This will create a strongly connected component in the graph consisting of all the shots from both scenes and will result in undersegmentation of the video. On the other hand, if the scene contains shots with continuously changing contents as shown in Figure 5.2, which is typical of scenes in many action movies, an oversegmented result will likely be obtained. We believe that the detection of scenes should not be based on one-to-one shot similarity. Therefore, in our approach, we consider the shot similarities on the whole rather than individual shot pairs. We incorporate the full effect of surrounding shot contexts which maximizes the intra-scene similarity between the shots of one scene and minimizes the inter-scene similarity between the shots of two different scenes. As a result, it is not affected by any mismatch between shots of different scenes. It incorporates motion information together with the color information of the video, which makes it robust to the aforementioned problems. Another advantage of our method is that it does not require any specific threshold, and experiments show that it performs equally well for a variety of film genres.

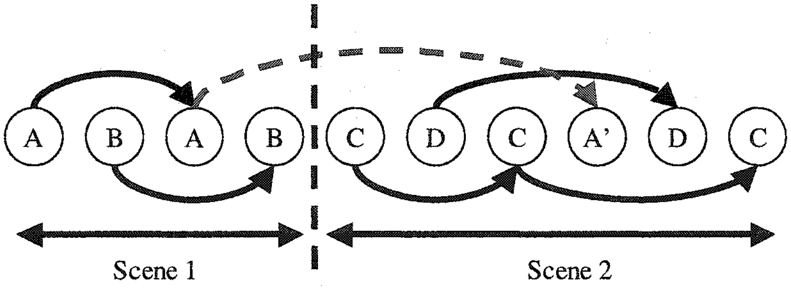


Figure 5.1: Erroneous shot linking indicated by the dotted link. A valid scene boundary will be missed resulting in undersegmentation of the video.

5.3 FIRST METHOD: TWO PASS SCENE DETECTION ALGORITHM

In Webster's dictionary, a scene is defined as follows [67]:

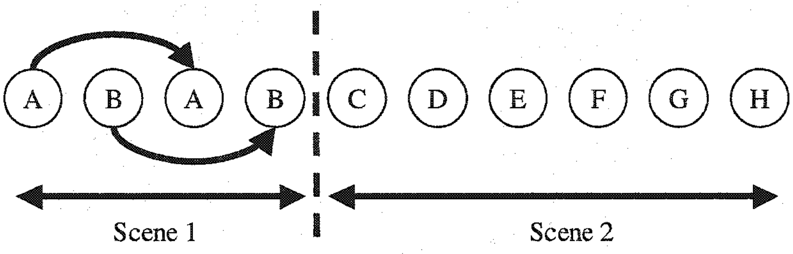


Figure 5.2: Oversegmentation due to continuously changing visual content of shots, which is typical of shots in many action scenes. Scene 2 is oversegmented since links cannot be established between the shots.

- A subdivision of an act in a dramatic presentation in which the setting is fixed and the time continuous OR
- One of the subdivisions of a play; as a division of an act presenting continuous action in one place.

The first definition of the *scene* emphasizes the fact that shots belonging to one scene are often taken with fixed physical settings. Several cameras capture the video with different angles while the background remains the same. A scene filmed inside a studio is an example, where physical settings do not change with time. A characteristic of this category of scenes is their repetitive structure due to switching between cameras with fixed views. However, this definition may not hold for all kinds of scenes. One example is an outdoor scene, where the background may change. Other examples are scenes that are shot with the cameras mounted on trucks or trolleys. In this case, a scene may be defined by the continuity of ongoing actions performed by the actor(s).

Movie directors, while filming a scene, control the pace of the film in order to sustain the viewer's interest. Two important factors which have been known to influence the pace of a movie are the Montage (or editing, i.e. the shot change rate) and the motion [61]. For a given scene, these factors are kept consistent so that the the viewer's attention is always engaged. We use this knowledge to develop a two-pass algorithm for scene boundary detection suitable for feature movies. The video is initially parsed into shots by camera break detection. Each shot is represented by one or more key frames depending upon the shot activity. For each shot, its length and motion contents are also estimated as shot features. In pass 1 of our algorithm, motivated by the first definition of a *scene*, a color similarity measure of shots is computed. This measure is called *Backward Shot Coherence* (BSC) and describes how well a shot matches with the previously seen shots. We find valleys in the BSC and detect several *Potential Scene Boundaries* (PSB). As a result of this step, *action* scenes may be split into many scenes for not satisfying the first definition of a scene. To improve the segmentation, we merge scenes during pass 2 by deleting weak PSBs. This is achieved by computing the *Shot Dynamics* (SD) of each scene detected in pass one. SD is a function of shot length and motion contents of the shots in a scene. Our observation is that *non-action* scenes, such as dialogue scenes, have longer shot length and little activity. On the other hand, action scenes, such as fights, are composed of rapidly changing shots (shorter shot lengths) and higher motion content. We exploit this information in the second pass of the algorithm. Figure 5.3 shows the flowchart of complete algorithm. These steps are discussed in Sections 5.3.1 and 5.3.2 respectively. Section 5.4 discusses results of our algorithm obtained by processing five Hollywood movies, one episode of a sitcom and one hour of a talk show.

5.3.1 PASS ONE: COLOR SIMILARITY ANALYSIS

The first pass of the algorithm deals with the detection of *Potential Scene Boundaries* (PSBs). A PSB is a possible instance of the beginning and/or ending of a scene in a movie. This is achieved by estimating a feature for each shot, called *Backward Shot Coherence* (BSC); a similarity measure of a given shot with respect to the previous shots. First we compute the *shot coherence* (SC) between shot i and the N previous shots. SC is defined as the color similarity between two shots. Let SC_i^j express the shot coherence of shot i and shot j, where shot i and j contain n and m key frames respectively. Then:

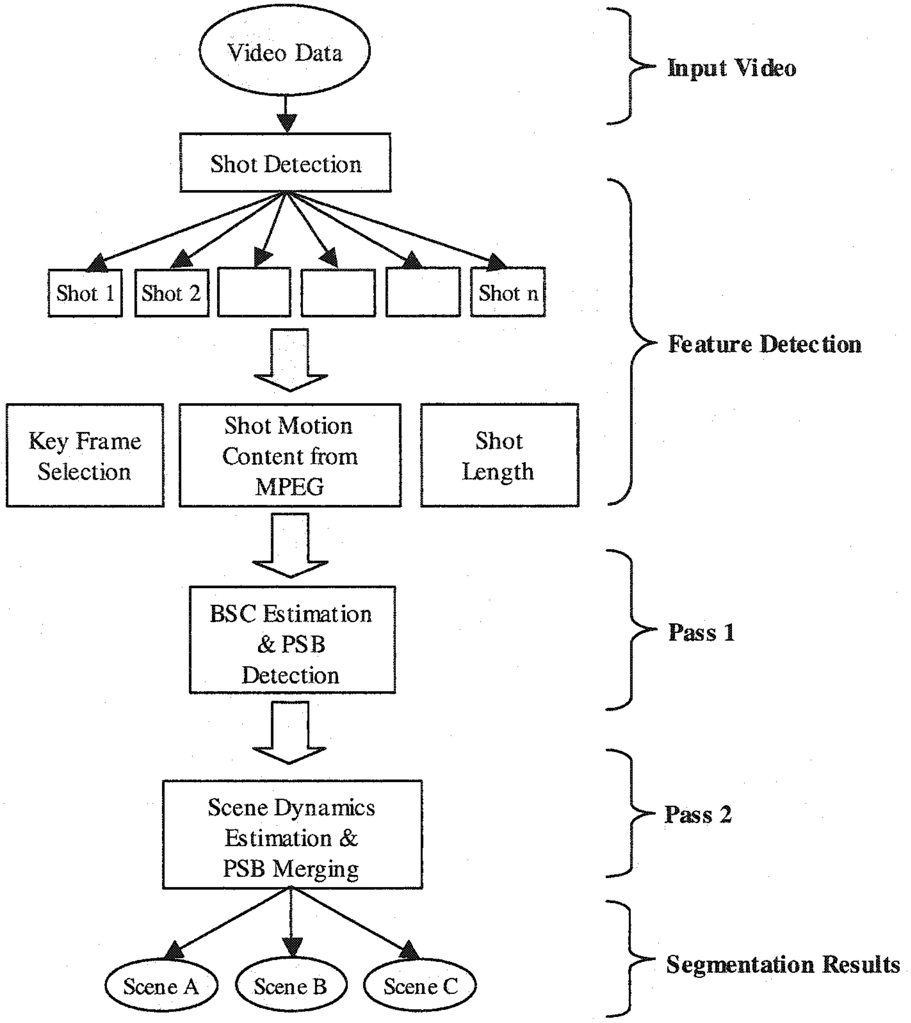


Figure 5.3: Flow chart showing the stages of Scene Boundary Detection algorithm.

$$SC_i^j = \max_{f^x \in K_i, f^y \in K_j} \left(D(f^x, f^y)\right), \tag{5.1}$$

where D is defined in Eq. 2.2.1 and K represents the key frame sets of respective shots. Backward shot coherence for shot i is then computed by taking the maximum shot coherence between shot i and the previous N shots:

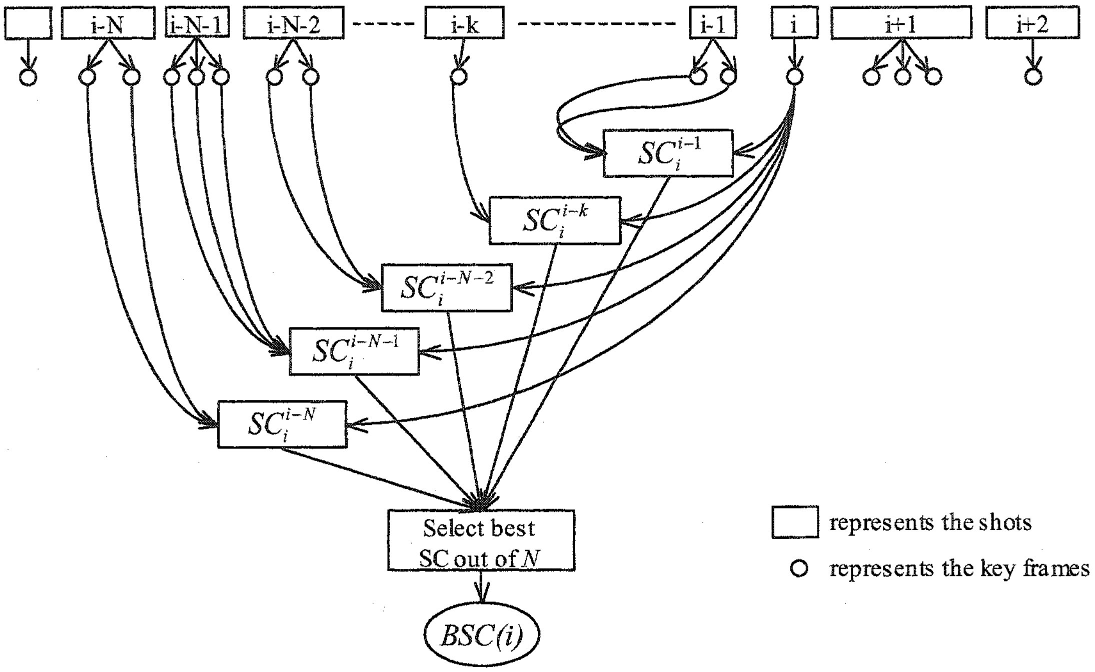


Figure 5.4: Computation of BSC. Rectangles in the figure represent shots and circles indicate key frames.

$$BSC_i = \max_{1 \leq k \leq N} \left(SC_i^{i-k} \right), \tag{5.2}$$

where BSC_i is the *backward shot coherence* of shot i.

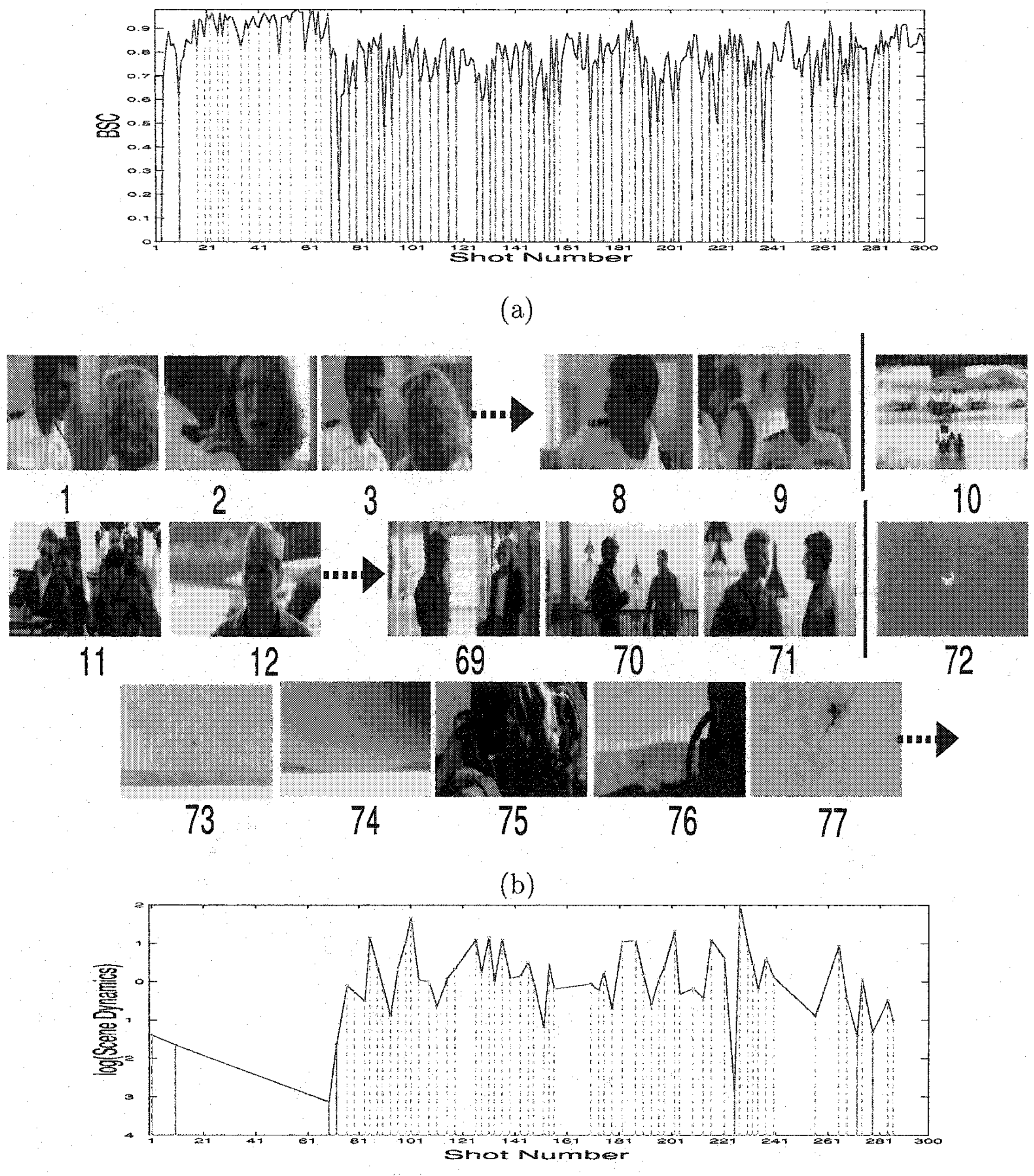
Figure 5.4 shows a graphical representation of this step. A scene is defined as a collection of contiguous shots in time which are taken at the same location and show similar visual content (according to the first definition, Section 5.3). In the beginning of a new scene, the initial shots do not resemble the shots of the previous scene due to the dissimilarities in the physical settings. BSCs of these shots are very small and show a poor similarity to the shots of the previous scene. As the scene progresses, the shots are repeated and therefore BSCs of shots attain higher values. This continues until the start of a new scene. The beginning of a new scene can be detected by locating valleys in the smoothed plot of BSC.

We call these valleys *potential scene boundaries* (PSBs) as they are the candidates for the starting point of a new scene. In some cases, a PSB may be detected within a scene as an outlier. A shot in the middle of a scene which is unique in that scene (like a flashback) can cause a false valley in the BSC. To suppress the outliers and prevent oversegmentation in this phase, we compare the color attributes of key frames of neighboring potential scenes. If a pair of key frames of two adjacent potential scenes are found to be similar, then the scene boundary between the two is removed and the scenes are merged into one scene. Let k and $k+1$ be the two potential scenes. The PSB between the scenes will be removed if:

$$D\left(f^i, f^j\right) \geq T_{color}, \tag{5.3}$$

where $f^i \in Scene_k$ and $f^j \in Scene_{k+1}$ and T_{color} is a fixed threshold. Figure 5.5(a) shows the plot for BSC of first 300 shots of the movie "Top Gun" for pass one. These shots span over five scenes as segmented by the human observer (See Tab. 5.3, scenes 1-5). Solid (blue) vertical lines indicate the PSBs. Dotted (green) vertical lines are the weak PSBs that were removed after adjacent scene merging. Figure 5.5(b) shows the first key frame of some shots. Note that the transition from shot 9 to shot 10 results in a sharp valleys as the latter shot was never shown before and hence a PSB is detected. Similarly, the transition from shot 71 to 72 is also identified as a scene boundary. Several outliers were successfully removed. Note that the first three segments (solid lines at shot numbers 10, 68 and 72 in Figure 5.5(a)) correspond to the first three scenes in the ground truth. The fourth scene in the movie is about pilot training and is broken into a large number of potential scene boundaries (solid blue lines from shots 76 to 273). This occurs due to the high dynamics and nonrepetitive structure of the scene.

The computation of BSC is controlled by the selection of the window size N. It can be considered to be a memory parameter which mimics a human's ability to recall a shot seen in the past (also suggested by Yeo. et. al. [71]). However, the choice of N greatly affects the initial segmentation. If this value is too large, it may span several scenes and a wrong



(a)

(b)

(c)

Figure 5.5: Detection of scenes in the movie "Top Gun". (a) The plot of BSC is shown for 300 shots. (b) The first key frame of each shot is shown with its shot index. (c) Plot of *Scene Dynamics* of potential scenes detected in pass one.

estimate of BSC may be obtained. On the other hand, if N is very small, the shot may not be compared with a sufficient number of shots within the scene, causing oversegmentation of video. $N = 10$ was used in our experiments that provided satisfactory results.

5.3.2 PASS TWO: SCENE DYNAMICS ANALYSIS

Most non-action scenes, such as dialogue scenes, with repetitive structure are well segmented during the first pass. However, scenes with weak structure are often broken in several scenes. In particular, action scenes are divided into several scenes due to non-repetitiveness of shots. The poor match among the shots causes an oversegmentation of the video. For a semantically meaningful segmentation, these potential scenes need to be merged together. The oversegmentation in pass one implies that the use of only color information is not enough for an appropriate segmentation of videos. Therefore, we have incorporated shot length and shot motion content as useful features to analyze scene properties. A characteristic of such scenes is their high motion activity and small shot length. Therefore, a weight *Scene Dynamics* (SD) is computed for each potential scene as follows:

$$SD_i = \frac{\sum_{j \in Scene_i} SMC_j}{\sum_{j \in Scene_i} L_j}, \tag{5.4}$$

where SD_i is the scene dynamics of scene i, SMC_j is the shot motion content of the j^{th} shot in the scene and L_j is the length of the corresponding shot. Large values of SMC and smaller values of L_j in dynamic scenes cause SD to be large. On the other hand, relatively calm scenes which span numerous frames with small SMC return very small values. The scene dynamics of every pair of adjacent potential scenes is analyzed. The PSB between two consecutive scenes k and $k + 1$ is removed if the SD of both scenes exceed a fixed threshold. See Figure 5.5(c) which shows the final scene boundaries for first 300 shots of the movie "Top Gun". Note that PSBs (dotted green lines) are removed where the scene dynamics are relatively high for consecutive scenes.

5.4 EXPERIMENTS

We have experimented with video sequences from five Hollywood movies including "Terminator II", "Golden Eye", "Gone in 60 Seconds", "Top Gun", and "A Beautiful Mind". Each movie sample was 35-60 minutes long and taken from the middle of the movie. We have also experimented with one episode of a sitcom, "Seinfeld" (21 minutes of running time), and one complete hour show of "Larry King Live". The videos were digitized at 29.97 fps except "Larry King Live" which was digitized at 10 fps. For each video, a human observer identified the ground truth scene boundaries. Chapter information from the DVDs was also incorporated to evaluate the results.

Table 5.1 summarizes the data set, the ground truth, and results obtained by our proposed method. This table also lists the number of false positive and false negative scenes. To evaluate the performance, we have also listed the recall and precision figures for each video. A detected scene is considered correct if it is detected in a neighborhood of ± 10 seconds of the ground truth. Also, scenes with less than 3 shots are removed as being insignificant. It is clear from the table that our results are pretty encouraging.

The detailed scene detection results for videos can be found in Tables 5.2-5.4. Here we discuss the results of the movie "Golden Eye". This video consisted of 107,724 frames (about 60 minutes of running time). The total number of shots found in the video was 1,519 (see Table 5.2). The first column in the table shows the chapters from the DVD. The second column lists the titles for each scene segment identified by a human observer. It should be noted that DVD chapters are a superset of scenes identified by the human observer. Columns 3 and 4 provide the number of shots and the number of frames in the segmented scenes. The last column indicates the number of scenes detected by the algorithm. Note that the algorithm detected more scene boundaries than the human observer as seen in Table 5.1. We believe that a slight oversegmentation is acceptable over undersegmentation, since split scenes can be combined by further analysis. While browsing a video, it is preferable to have two segments of one scene rather than one segment consisting of two scenes. There are a few missed scenes which are indicated with 'x' in the table. These are the scenes which were wrongly merged with the previous scene boundary.

To demonstrate that the algorithm presented here works equally well on other video genres, we also conducted an experiment on one sitcom show "Seinfeld" that belongs to the genre of video with mostly dialogues and very little action content in the shots. Table 5.4 lists the scene detection for this show, which demonstrates that it performs adequately for a very different genre of video.

5.4.1 SCENE DETECTION IN INTERVIEW SHOWS

We have also tested our algorithm on one hour long "Larry King Live" program, digitized at 10 frames per second. The video consisted of 8 segments in which the guest was interviewed by the host, Larry King. There were 7 segments of commercials between the interview segments (see Table 5.5). Like movies, the segment of the program showing the interview can be considered as one scene, whereas the commercials together can be considered as another scene. The algorithm proposed here worked very well in detecting the scene boundaries between the interview and commercial segments. Shots belonging to interview segments are clustered together due to strong repetitive structure. Commercials, on the other hand, happened to have similar characteristics to action shots (higher motion content and smaller shot length). Due to the non-repetitive structure of commercials, several clusters were found during pass one. However, in pass two, all commercials are combined together by computing their scene dynamics, thus separating them from the program segments. This video also had several small clips of news footage, which were shown either before or after every interview segment together with the commercials. These small clips are not similar to interview segments as they resemble the commercials in terms of non-repeating structure and short shot lengths. As a result, these clips were separated from the main interview segments and merged with the commercials. Similarly, in the 6^{th} interview segment, when news footage was shown in the middle of the interview, that particular segment was divided into two scene units. The rest of the segments were detected successfully as shown in Table 5.5.

Table 5.1: Summary of data set and experimental results for five Hollywood movies and one sitcom.

Video	Duration	Frames	Shots	Gnd. Truth	Scenes Det.	False Neg.	False Pos.	Rec.	Prec.
Terminator II	55 min	98,505	1,632	36	38	5	7	86.1%	81.6%
Golden Eye	60 min	107,724	1,519	25	35	3	13	88.0%	62.9%
Gone in 60 Sec.	58 min	105,226	1,869	39	43	6	10	84.6%	76.7%
Top Gun	50 min	89,999	1,105	26	30	3	7	88.5%	76.7%
Beautiful Mind	36 min	65,121	446	17	21	2	6	88.2%	71.4%
Sienfeld	21 min	31,508	318	22	27	3	8	86.4%	70.0%

5.5 SECOND METHOD: A GRAPH THEORETICAL APPROACH

In this framework, we find scene boundaries by transforming this problem into a graph partitioning problem. In our approach, we construct a weighted undirected graph called a *shot similarity graph*, SSG, and detect scenes by recursively partitioning the SSG. The undirected graph consists of nodes such that each node represents a shot and edges connect

Table 5.2: Scene boundary detection on 60 minutes of the movie "Golden Eye". 'x' shows the missed scene boundaries.

DVD Chapter	Human Observation	Shots	Frames	Scenes Det.
Boris and Natalia	Boris and Natalia	18	1,763	1
Ourumov's Timed Test	Enemy Arrives	35	3,858	2
N/A	Killing in the station	38	1,024	1
Aiming Goldeneye	Activating Goldeneye	83	4,191	2
N/A	Villain Attempts to Kill Natalia	x	x	x
After Hours Briefing	Bond and Secretory	16	4,010	1
Queen Of Numbers	Bond Visits Q	7	1,042	1
Survivor / Spy Talk	Destruction	272	7,478	3
N/A	Bond and Q Discuss G. Eye	31	4,098	2
N/A	Natalia Escapes From Rubble	21	2,933	1
M Informs Her Spy	Bond and M	15	2,101	2
Tt. Petersburg,Russia	Russian Officials	36	4,119	2
Q's Latest Gadgets	Bond Gets Gadgets	25	4,761	1
Jack Wade, C.I.A.	Bond Arrives In Russia	15	3,192	2
N/A	Bond's Car Broke	14	2,199	2
Computerized Warning	Natalia Chats with Boris	27	4,196	2
Betrayal / Getting Even	Bond Arrives In Church	x	x	x
N/A	Natalia Is Trapped	13	1,473	1
N/A	Bond in the Casino	34	5,158	1
N/A	Bond In The Hotel	20	2,772	1
007 Gets a Grip	Attacked At The Pool	55	6,057	1
Back From The Dead	Bonds Meets with 006	51	6,707	1
Trapped In The Tiger	Bond and Natalia in the Tiger	111	10,947	1
Tough Interrogation	Interrogation	13	1,280	2
Archives Ambush	Fight	541	19,569	1
All "Tanked" up	x	x	x	x
Derailing Evil Plans	Train Crash	28	2,796	1

Table 5.3: Scene boundary detection on 50 minutes of the movie "Top Gun". 'x' shows the missed scene boundaries.

DVD Chapter	Human Observation	Shots	Frames	Scenes Det.
Crash and Burn	First encounter with Charlie	9	1,198	1
Charlie	Charlie's presentation	59	5,452	1
N/A	After the presentation	3	1,911	1
Turn and Burn	Flying training	202	6,645	2
No Flexibility	The locker room	20	2,384	1
N/A	In the boss's office	25	4,869	1
Flying Against a Ghost	Maverick and Goose	23	2,920	2
Tempted	Charlie offers a dinner to Maverick	23	3,270	1
Playing with the Boys	Playing volleyball	53	2,695	3
No Apologies	Dinner at Charlie's home	65	9,808	1
N/A	In the elevator	30	3,325	1
N/A	Goose's family arrives	8	1,356	1
Textbook Maneuvers	Training session	60	6,101	2
N/A	In the bedroom	4	2,013	1
N/A	Charlie wakes up alone	3	670	1
The Need for Speed	Flight competition	169	8,283	2
Your Attitude	The locker room	10	1,843	1
N/A	Maverick in the bed room	33	5,350	1
Great Balls of Fire	Dinner	x	x	x
Every Point Counts	Goose F-14 crashes	267	11,670	2
Let Him Go	Goose is dead	x	x	x
N/A	I will be here	11	2,071	1
N/A	Goose's room	8	1,401	1
N/A	Maverick and Goose's wife	17	4,399	1
N/A	In the court	x	x	x
Get Him Up Flying	Maverick in the F-14	3	365	1

Table 5.4: Scene boundary detection on sitcom "Seinfeld". 'x' shows the missed scenes boundaries.

Scene Title	Shots	Frames	Scenes Det.
1. Restaurant	34	3,786	1
2. In the apartment	22	1,373	2
3. George and Tina	10	1,125	1
4. In the theater lobby	18	2,461	2
5. George and Tina	12	902	1
6. Theater	6	718	1
7. George in the car	x	x	x
8. Theater	18	1,399	2
9. Apartment	x	x	x
10. Restaurant	20	1,972	1
11. Apartment	21	2,033	1
12. George and Tina in the park	20	1,605	1
13. Restaurant	5	1,042	1
14. Theater	8	1,016	1
15. George's car on fire	23	2,098	3
16. In the theater lobby	52	4,616	3
17. George car being toed	22	1,598	1
18. Jerry in the car	5	532	2
19. George and Tina	4	906	1
20. Jerry in the car	x	x	x
21. Kramer in the dressing room	6	1,088	1
22. George in the costume	12	1,241	1

the nodes. A weight is associated with every edge and is proportional to the shot similarity. The shot similarity is computed as a function of color and motion features of shots. The scene boundaries are detected by partitioning the SSG into subgraphs that maximize the intra-subgraph similarities and minimize the inter-subgraph similarities. We use the *normalized cut* method of graph partitioning to cluster shots. It should be noted that our

Table 5.5: Scene boundary detection 60 minutes of the "Larry King Live" show.

Human Observation	Shots	Frames	Scenes Det.
Interview Segment 1	38	4,565	1
Commercials + News Reel	86	1,434	1
Interview Segment 2	34	4,377	1
News Reel + Commercials	72	1,593	1
Interview Segment 3	27	3,259	1
News Reel + Commercials + News Reel	65	1,098	1
Interview Segment 4	16	1,863	1
News Reel + Commercials + News Reel	19	779	1
Interview Segment 5	41	4198	1
Commercials 7	30	936	1
Interview Segment 6 + News Reel + Interview Segment	47	3,269	2
News Reel + Commercials	64	1,542	1
Interview Segment 7	20	2,132	1
Commercials	32	318	1
Interview Segment 8	59	3,215	1

algorithm considers the *global* similarities of shots to detect boundaries rather than *local* similarities which is the basis of several related approaches (See Section 5.2). The proposed algorithm is robust and is not affected by any *local* mismatch between two shots that belong to two different scenes. Furthermore, no specific thresholds are required to tune the algorithm for a particular genre of video. We have conducted extensive experiments which validate the approach. These results are presented in Section 5.6.

5.5.1 COLOR AND MOTION SIMILARITIES BETWEEN THE SHOTS

As explained in Section 5.3, shots that belong to one scene often have similar visual (color) and/or action (motion) contents. Generally, dialogue shots span many frames and are filmed with a fixed physical setting. Due to the repetitive transitions between the fixed camera views, the shots in this scene category have higher visual correlation. On the other hand, shots in fight and chase scenes change rapidly and last for only a few frames (Arijon [4]). In a similar fashion, the motion content of shots also depends on the nature of the scene. The dialogue shots are relatively calm (neither actors nor the camera exhibit large motion). Although camera pans, tilts and zooms are common in dialogue shots, they are generally smooth. In fight and chase shots, the camera motion is jerky and haphazard with larger movements of actors. For a given scene, these two attributes are kept consistent over time to maintain the pace of the movie. Thus, we compute the similarities between shots as a function of their visual and motion content features. That is, the similarity between shots i and j will be:

$$ShotSim(i,j) = \alpha \cdot VisSim(i,j) + \beta \cdot MotSim(i,j), \tag{5.5}$$

where α and β are the weights given to each shot feature such that $\alpha + \beta = 1$. We used $\alpha = \beta = 0.5$ during the experiments which provided satisfactory results. The $VisSim$ between shots i and j is now defined as follows:

$$VisSim(i,j) = SC_i^j, \tag{5.6}$$

where SC is defined in Eq. 5.1. In words, the $VisSim$ for any arbitrary pair of shots is the maximum color similarity of all possible pairs of their key frames.

It is likely that the consecutive shots of a particular scene will have similar motion contents (consider an action scene, for example). We compute the motion similarity, $MotSim$, between two shots as follows:

$$MotSim(i, j) = \frac{2 \times min(SMC_i, SMC_j)}{SMC_i + SMC_j}, \tag{5.7}$$

where SMC is the shot motion content as defined in Section 2.2.4. Thus, if two shots have similar motion content, their $MotSim$ will have a higher value. Note that both $VisSim$ and $MotSim$ are in the range 0-1.

5.5.2 THE SHOT SIMILARITY GRAPH, SSG

Given N shots, we construct a weighted undirected graph called a *shot similarity graph* $G = (V, E)$, such that each shot i is represented by a node v_i, where i is the shot index. Let $e(i, j) \in E$ be the edge between the nodes i and j with an associated weight $W(i, j)$ which reflects the likelihood that two shots belong to one scene. It is less likely that two shots farther apart in time will belong to one scene. That is, the higher the temporal distance between them, the lower the probability that the shots belong to one scene. Therefore, the weight $W(i, j)$ is proportional to the $ShotSim(i, j)$ and temporal proximity of the shots. This is formulated as:

$$W(i, j) = w(i, j) \times ShotSim(i, j), \tag{5.8}$$

where $w(i, j)$ is a decreasing function of the temporal distance between the shots. We chose an exponential weighting function for its relative simplicity, as well as neutrality. It can be considered as a memory parameter in that the ability to recall a shot decreases with

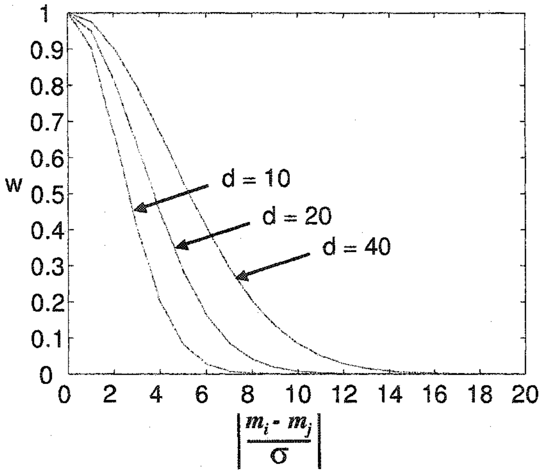


Figure 5.6: Computation of $w(i, j)$, which is an exponentially decreasing function of temporal distance between two arbitrary shots, i and j. Note that the weight decreases at a slower rate for larger values of d.

the time. Thus the weight $w(i, j)$ decays with the temporal distance between the middle frames of two shots under consideration, that is:

$$w(i, j) = e^{-\frac{1}{d} \cdot |\frac{m_i - m_j}{\sigma}|^2}, \tag{5.9}$$

where m_i and m_j are the indices of the middle frames of each shot and σ is the standard deviation of the shot durations in the entire video. The rate of decay is controlled by a factor d which determines how long a shot should be *remembered.* We set $d = 20$ in our experiments, which was kept constant throughout the experiments. Figure 5.6 shows the plot of w against the temporal distance between shots. With this value, a shot is *forgotten* if the the temporal distance is more than 10 times the standard deviation of all shots present in the video.

Figure 5.7 shows the shot similarity graph constructed for 36 minutes of the movie "A Beautiful Mind". There are 219 shots in the video. The similarities between the nodes are represented by pixel intensities such that lower intensity means higher similarity. Please

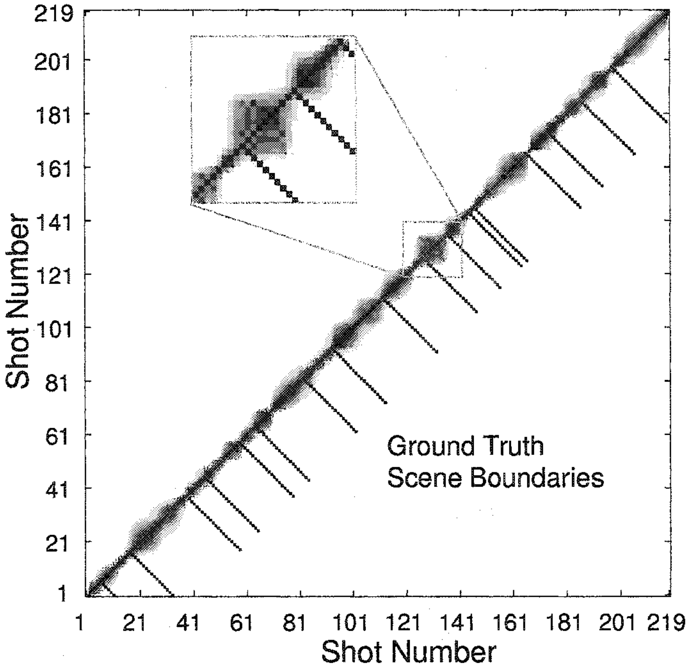


Figure 5.7: A *shot similarity graph* for 36 minutes of the movie “A Beautiful Mind”. Higher similarity is represented by darker pixels. Diagonal pixels have the lowest intensity as they represent the self similarities of shots. The ground truth scene boundaries are indicated with lines on the lower-right side of the diagonal. Note that shots that belong to a particular scene form distinct clusters as seen in the zoomed-in section of the SSG.

note that the diagonal pixels represent the self similarity of shots and therefore, have the maximum similarity and the lowest pixel intensity. A human observer identified the scene boundaries present in the video which are indicated by lines on the lower-right side of the diagonal. Also note that the shots that belong to a particular scene form distinct clusters as seen in the zoomed-in section of the SSG. Figure 5.8 shows a similar graph for one episode of the sitcom “Seinfeld”.

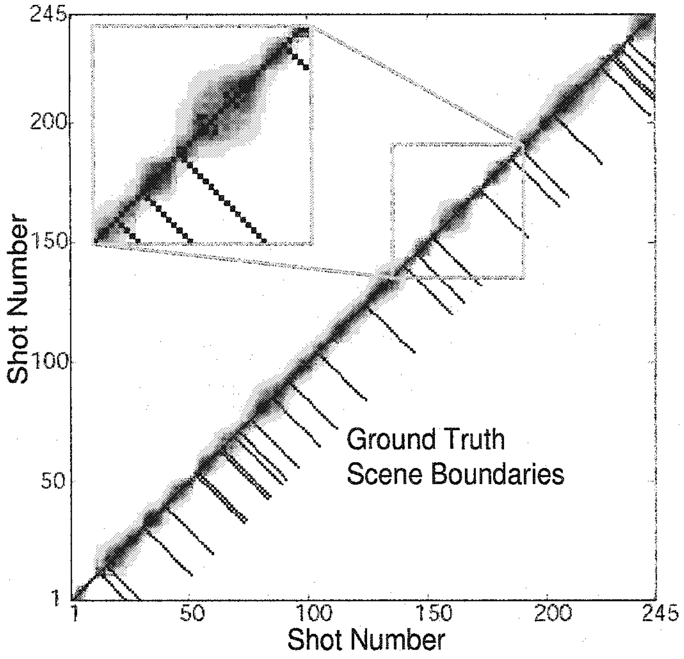


Figure 5.8: A *shot similarity graph* for 18 minutes of the sitcom "Seinfeld". Higher similarity is represented by darker pixels. Diagonal pixels have the lowest intensity as they represent the self similarities of shots. The ground truth scene boundaries are indicated with lines on the lower-right side of the diagonal. Note that shots that belong to a particular scene form distinct clusters as seen in the zoomed-in section of the SSG.

5.5.3 SCENE DETECTION USING GRAPH CUTS

Graph partitioning techniques are known for effective perceptual grouping. Several algorithms have been proposed for segmenting images based on pixel proximity and color intensity similarity, for example, [58], [69] and [56]. Generally, the partitioning solution is achieved by recursive bipartitioning, that is, at each step the graph is divided into two parts based on a partitioning measure. We employ the graph partitioning technique proposed by Shi and Malik [58] called *normalized cuts.* Starting with an initial SSG, $G = (V, E)$, we seek a partitioning into two disjoint subgraphs, $G' = (V', E')$ and $G'' = (V'', E'')$ such that $V' \cup V'' = V$ and $V' \cap V'' = \emptyset$. Such a partition is achieved by removing the edges connecting subgraphs G' and G''. There exist an exponential number of such partitions. However, for videos, we seek a partition such that all shots that belong to a particular subgraph are time continuous. That is, the following condition holds:

$$(i < j \text{ or } i > j) \text{ and } i \neq j \textit{ for all } \; v_i \in V', v_j \in V''.$$

Thus, the complexity of partitioning an SSG into two subgraphs is of order N which is the number of shots present in the segment of the video. In graph theory literature, the summation of weights associated with the edges being removed is called a *cut* and it reflects the degree of dissimilarity between the two parts, that is:

$$cut(V', V'') = \sum_{i \in V', j \in V''} W(i, j). \tag{5.10}$$

The normalized cut value for such a partitioning is expressed as:

$$Ncut(V', V'') = \frac{cut(V', V'')}{assoc(V', V)} + \frac{cut(V', V'')}{assoc(V'', V)}, \tag{5.11}$$

where $assoc(X, V)$ is the summation of weights associated with the edges connecting all nodes in X to all nodes in V, that is:

$$assoc(X, V) = \sum_{c \in X, d \in V} W(c, d). \tag{5.12}$$

We apply a recursive algorithm of graph partitioning such that the intra-subgraph similarities are maximized and the inter-subgraph similarities are minimized; that is, the shots in each subgraph will have higher visual (color) and activity (motion) similarities.

This approach results in the clustering of shots that are more likely to be in one scene. It should be noted that unlike several other graph-based video segmentation approaches, there are no specific thresholds that control the segmentation. Hence, our method does not suffer in accuracy due to any mismatch between the shots of different scenes and therefore it is more robust to noise. Figures 5.9 and 5.10 show the scene detection for the movie "A

Beautiful Mind" and the sitcom "Seinfled" respectively. The detected scene boundaries are identified with lines on the upper-left side of the diagonal (a detected scene is considered correct if it is detected in a neighborhood of ± 30 seconds of the ground truth). A Please refer to Section 5.6, which discusses the experiments on video segments taken from both *action* and *non-action* Hollywood movies.

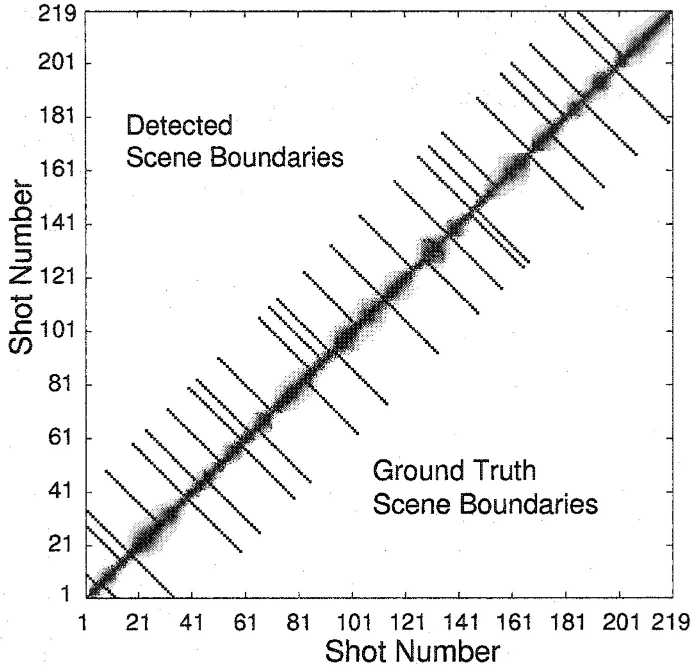


Figure 5.9: Scene detection for 36 minutes of the movie "A Beautiful Mind". Detected scene boundaries are indicated with lines in the upper-left side of the diagonal. Out of 18 scene boundaries, 15 scene boundaries are identified correctly.

5.6 EXPERIMENTAL RESULTS

To evaluate the performance of proposed algorithm, we performed experiments on the same video data set as in Section 5.4. The videos in the data set represent a variety of film genres,

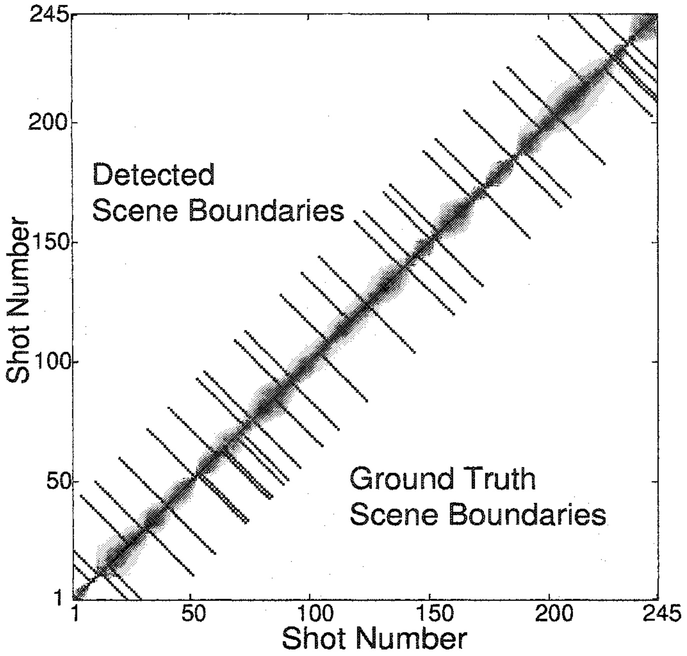
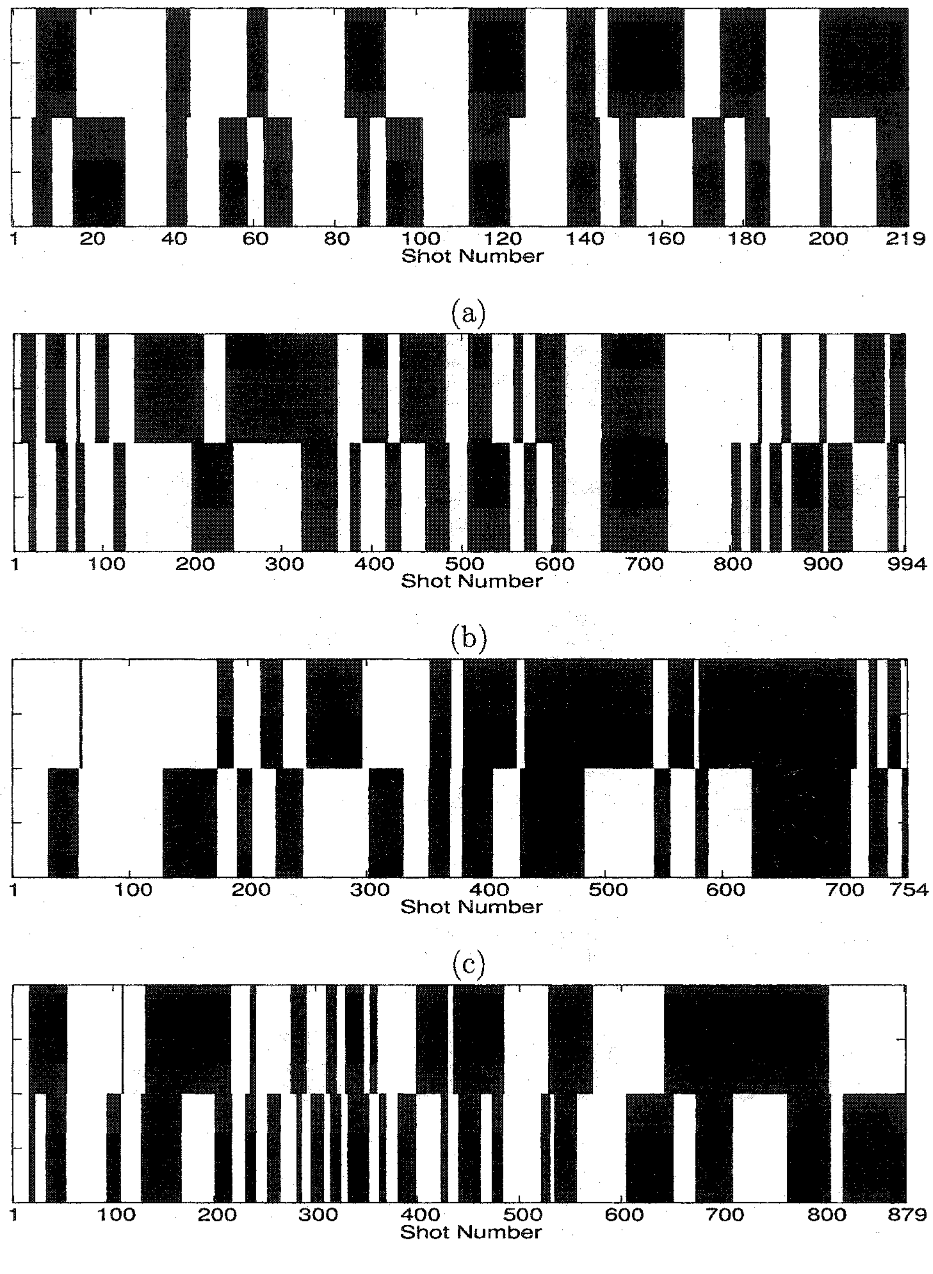


Figure 5.10: Scene detection for 18 minutes of the sitcom "Sienfeld". Detected scene boundaries are indicated with lines in the upper-left side of the diagonal. Out of 28 scene boundaries, 23 scene boundaries are identified correctly.

Video	Duration	Frames	Shots	Gnd. Truth	Scenes Det.	Correct Det.	False Neg.	False Pos.	Rec.	Prec.
Beautiful Mind	36 min	65,122	219	18	28	15	3	13	0.833	0.536
Terminator II	55 min	98,506	994	36	39	27	9	12	0.750	0.692
Top Gun	50 min	90,000	754	23	26	18	5	8	0.783	0.692
Gone in 60 Sec.	58 min	105,226	1,332	39	57	29	10	28	0.744	0.509
Golden Eye	60 min	107,786	879	25	44	22	3	22	0.880	0.500
Sienfeld	17 min	31,885	245	28	27	23	5	4	0.821	0.852

Table 5.6: Summary of data set and experimental results for five Hollywood movies and one sitcom.

such as action and drama movies as well as a TV sitcom which has a very different shooting



(a)

(b)

(c)

(d)

Figure 5.11: Ground truth scenes vs. detected scenes. (a) "A Beautiful Mind", (b) "Terminator II", (c) "Top Gun" and (d) "Golden Eye". The upper row represents the scenes identified by a human observer. Consecutive scenes are shown with alternating black and white patterns. The bottom row shows the scenes detected by our algorithm for each video.

style from feature films. The experiments show that the algorithm is robust regardless of the film genre.

Table 5.6 summarizes the data set, the ground truth, and results obtained by our proposed method. This table also lists the number of false positive and false negative scenes. To evaluate the performance, we also provide the recall and precision figures for each video. The boundaries detected by the proposed method are compared against the ground truth using the *best match* method. A 30 second sliding window is swept over the detected boundaries as the tolerance factor. Thus, a detected scene is considered correct if it is detected in a neighborhood of ± 30 seconds of the ground truth. It is clear from the table that our results are pretty encouraging.

The detailed scene detection results for videos can be found in Tables 5.7, 5.8 and 5.9. Here we discuss the results for the movie "Terminator II". This video consisted of 98,506 frames (about 55 minutes of running time). The total number of shots found in the video was 994, see Table 5.8. The first column in the table shows the chapters from the DVD. The last column indicates whether or not a scene is correctly detected by our algorithm. There are few missed scenes which are indicated with 'x' in the table.

Figure 5.11 shows the scene detection results against the ground truth. The ground truth scenes are indicated by alternating black/white patterns w.r.t. the shot numbers in the upper row. The bottom row shows the detected scenes. We have observed that the algorithm works better for slow paced scenes, such as dialogue scenes, than for fast paced scenes. This is due to the fact that slow paced scenes are often well structured. Action scenes, on the other hand, are poorly structured and appear as multiple clusters in the graph. We believe that the use of audio information as a similarity measure can be incorporated to improve the segmentation task.

Table 5.7: Scene Boundary Detection for the movie "A Beautiful Mind". Correctly identified scenes are marked with ✓. '*' represents the scenes identified by the human observer and did not appear in the movie's DVD chapter selection menu.

No.	**DVD Chap./Human Observer(*)**	**Scene Detected?**
1	Mathematicians	✓
2	Reflections *	✓
3	Princeton Dorm *	✓
4	Drinking *	✓
5	A Challenge	✓
6	The Need to Focus	✓
7	The Bar *	×
8	Princeton *	✓
9	Dorm Room *	✓
10	Governing Dynamics	✓
11	Research *	×
12	With the Principle *	✓
13	Celebrations *	✓
14	The Pentagon	✓
15	Wheelers Defense Labs *	×
16	Teacher and Students	✓
17	Code Breaker	✓
18	Laboratory *	✓

Table 5.8: Scene Boundary Detection for the movie "Terminator II". Correctly identified scenes are marked with ✓. 30 out of 36 scenes are listed.

No.	DVD Chapter	Scene Detected?
1	Meet John Conner	✓
2	Sarah Connor	×
3	T-1000 Visits The Voights	✓
4	Easy Money	×
5	Sanity Review	×
6	Cyberdyne Systems	✓
7	Model Citizen	✓
8	Target Acquired	×
9	The Galleria	✓
10	Zeroed In The Corridor	✓
11	Into The Streets	✓
12	Canal Chase	✓
13	Time Out	✓
14	Never This Nice	✓
15	Photos	✓
16	Mission Parameter	✓
17	Pescadero State Hospital	✓
18	Lewis The Guard	✓
19	Sarah Breaks Out	×
20	215 Bones	✓
21	I Swear	✓
22	Syringe Point	✓
23	Come With Me If You Want To Live	✓
24	Escape From Pescadero	✓
25	Security Car	✓
26	Nice Bike	✓
27	Night Repairs	×
28	Head South	✓
29	No Problemo	×
30	Detailed Files	✓

Table 5.9: Scene Boundary Detection for the movie "Top Gun". Correctly identified scenes are marked with ✓.

No.	DVD Chapter	Human Observer	Scene Detected?
1	Charlie	Charlie's presentation	✓
2		After the presentation	✓
3	Turn and Burn	Flying training	×
4	No Flexibility	The locker room	✓
5		In the boss's office	✓
6	Flying Against a Ghost	Maverick and Goose	×
7	Tempted	Charlie offers a dinner to Maverick	×
8	Playing with the Boys	Playing volleyball	✓
9	No Apologies	Dinner at Charlie's home	✓
10		In the elevator	✓
11		Goose's family arrives	✓
12	Textbook Maneuvers	Training session	✓
13		In the bedroom	✓
14	The Need for Speed	Flight competition	×
15	Your Attitude	The locker room	✓
16		Maverick in the bed room	✓
17	Great Balls of Fire	Dinner	✓
18	Every Point Counts	Goose F-14 crashes	✓
19	Let Him Go	Goose is dead	✓
20		I will be here	✓
21		Goose's room	×
22		Maverick and Goose's wife	✓
23		In the court	✓

5.7 SCENE REPRESENTATION

A scene representation using one or multiple images is crucial for building an interactive tool for video browsing. Before watching a video by scenes, the user can look at a single image and get an idea of the scene. Therefore, this representation needs to be consistent with the content of the scene. In DVDs, which are available with the chapter selection option, each chapter is represented by one key frame. The creators, who have complete access to the script of the movies, manually pick a frame that adequately reflects the scenario. Since this is a subjective process, the choices of frames may vary from one individual to another. However, the main objective of the key frame is to give a hint of the height of drama, suspense and/or action of the scene. In this section we address the issue of automatic selection of key frames of scenes.

In our approach, we first compute a shot goodness measure as a function of shot visual similarity, shot length and shot activity. We have noticed by analyzing the key frames in DVDs that images with multiple faces are preferred over one face or with no faces for scene representation. For example, a scene where two actors are talking, a frame showing both is chosen over frames with a single person. One reason for doing so is to introduce as many characters of the scene as possible. Sometimes an image of a building or a landscape is preferred as a key frame to give an idea of the whereabouts of the scene. However, this is not frequent. Thus, the criteria for a good representative shot can be summarized as:

- the shot is shown several times (higher visual similarity with other shots),
- the shot spans a longer period of time (longer shot length),
- the shot has minimal action content (smaller shot motion content), and
- the shot has multiple people.

After computing the shot goodness, a fixed number of shots with the highest shot goodness value are selected as candidates for scene key frame (we used 3 shots in our experiments). These shots are then tested for the presence of faces. The shot with the maximum votes is selected as the representative shot for the corresponding scene.

5.7.1 MEASURING SHOT GOODNESS

The shot goodness is computed by analyzing three properties of every shot which includes shot coherence, shot length and shot activity. For each shot in the scene, its coherence with every other shot is computed. For a scene with N shots, a correlation matrix of dimension $N \times N$ is constructed where element (i, j) is the coherence of shot i with shot j (Eq. 5.1). It should be noted that the diagonal elements are unity i.e. $(i, i) = SC_i^i = 1$, and need not be computed. This matrix is also symmetric i.e. $SC_i^j = SC_j^i$ and hence the computation complexity reduces to $(N^2 - N)/2$. When two shots are similar, their coherence is high. Therefore, the sum of all column elements represents the correlation of a shot with the rest of the shots in the scene. If the shot is shown several times, this value is large. On the other hand, shots seen fewer times will have smaller values of correlation sum. Let $C(i)$ be the correlation sum of shot i, then:

$$C(i) = \sum_{j \in Scene} SC_i^j. \tag{5.13}$$

We associate a weight with each shot as follows:

$$W(i) = \frac{C^2(i) \times L_i}{log(SMC_i + \delta)}, \tag{5.14}$$

where W is the shot goodness, L_i is the shot length in the number of frames, and δ is used to prevent division by zero. The squared value of $C(i)$ is used to give more emphasis to shot coherence, whereas the log term for shot motion content is incorporated to reduce its effect on shot goodness. In our experiments the mean of SMC of all shots in a scene was used for δ, i.e.

$$\delta = \frac{1}{K} \sum_{j=1}^{K} SMC_j, \tag{5.15}$$

where K is the total number of shots in the scene. In the second step, three shots with the highest $W(i)$ are selected as candidate shots and face detection is performed on the first key frame of each shot using the method explained in the section below.

5.7.2 DETECTION OF FACES

Several face detection algorithms have been proposed in the literature (for example Rowley et al. [54], Sung et al. [62] and Roth et al. [53]). In shots obtained from video tracks of movies, we encounter faces with different levels of scale together with different orientations which makes face detection a difficult task. Therefore, we apply a simple but robust method of skin detection approach on the frames to detect faces. We detect skin by using a method proposed by Kjeldsen et al. [32] that requires training on color space (in our implementation, the RGB color space is used and images from the movie data set are used to train the color predicate). The first key frame of candidate shots are tested for skin pixels. Each isolated segment of skin is considered to be a face and the frame with the most faces is taken as the scene key frame. In the case of a tie or when no face is detected in any candidate key frame, the key frame of the shot with the highest goodness value, W, is selected.

Figure 5.12 shows a stepwise overview of the key frame detection for a scene from the movie “Golden Eye”. The scene title is “Boris and Natalia” in Table 5.2, chapter 1. This scene consists of 18 shots in which Boris and Natalia are having a conversation about breaking into the FBI security system. The key frame for each shot is shown in Figure

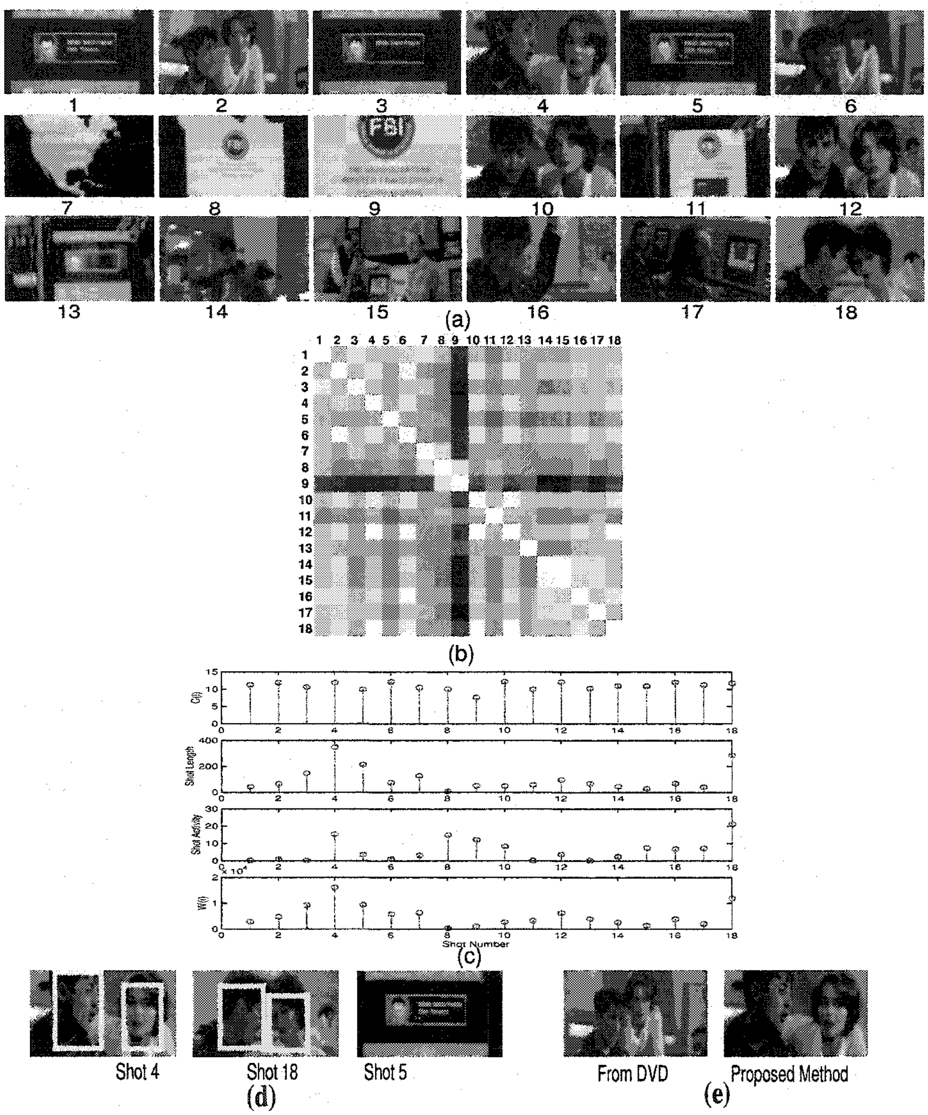


Figure 5.12: Best key frame detection for the scene content representation in a scene from the movie "Golden Eye". The scene consists of 18 shots as shown in (a). (b) the correlation of shots with each other. (c) the plot of correlation sum, $C(i)$, *shot length*, *shot activity* and *shot goodness* measure of each shot. (d) the three selected key frames with the highest shot goodness values. Rectangles show the detected faces in the images. (e) Compare the key frame selected by our algorithm (right) with the image obtained from the DVD (left).

5.12(a). Figure 5.12(b) shows the shot coherence matrix, in which brighter color represents higher coherence. In Figure 5.12(c), the values of correlation (C), shot length, and shot activity for corresponding shots are plotted. The small variations in $C(i)$ advocate the use of the square term in Eq. 5.14, whereas the higher variations in shot activity support the use of the log function. Figure 5.12(d) shows the three key frames with highest values of shot goodness (shots 4, 18 and 5 respectively). Note that shot 4, which is repeated the most, gets the highest weight. Using the skin detection method, two faces are detected in shots 4 and 18 and none in shot 5. Since shot goodness of shot 4 is higher than shot 18, it is chosen as the representative key frame for this scene. Figure 5.12(e) (left) is the image chosen by the creators of the DVD for this chapter of the movie. Compare the similarity of this image with Figure 5.12(e) (right) selected by our algorithm. Similarly, Figure 5.13 shows the key frame selection steps for scene title "I Swear" of the movie "Terminator II" (Table 5.8, chapter 21). The key frame of shot 1 is selected because it has higher shot length and smaller shot activity as well as two faces detected in the key frame (compared to shot 18 in which only one face is detected). Figures 5.14 and 5.15 show the results of key frame detection for scene representation for the movies "Terminator II", and "Golden Eye". A comparison has been made between the frames present in the DVD and those detected by our algorithm. For a detailed discussion please refer to Section 5.7.3.

5.7.3 SCENE REPRESENTATION RESULTS

Figure 5.14 shows the results for scene representation using a single key frame for the movie "Terminator II". For every DVD chapter, images from the DVD are shown in the left columns. Images detected by our algorithm are shown in the right columns. When the scene is split into more than one (for example Chapter 2 in Figure 5.14), a key frame for each detected scene is shown. Scene boundaries which are missed by the algorithm are shown as "False Negative". In the majority of scenes, key frames found by the algorithm are very similar to those of the DVDs. For example, Chapter 2, 5, 12, and 28 are some of them in which the two key frames are nearly the same. Our observation is that the key frames belonging to scenes with longer shots such as dialogue are detected with higher accuracy than action scenes. In DVDs, a frame is selected that reflects the height of the action in the latter category of scenes. As a matter of fact, this violates the assumption made in our algorithm (see Section 5.7). The selection of one key frame from a scene largely depends on the discretion and choice of the person. Therefore, different people can choose different frames for representing a scene. For example, in the DVD Chapter 1, "Meet John Conner", of the movie "Terminator II", a close shot of John and his friend is chosen. On the other hand, our algorithm selected a frame with three actors; which may be preferred over the original one, as it provides more details about the scene. Similarly, in Chapter 15 the frame presented in the DVD is a closeup shot of actors, whereas the frame detected by our algorithm is a *long shot*, however, both frames show the same content. Figure 5.15 shows scene representation results for the movie "Golden Eye". N/A denotes that the corresponding chapters were not present in the DVD and were segmented by human observer as ground truth.

5.8 SUMMARY

In this chapter, we presented two novel approaches to detect scene boundaries in videos. Our methods utilize several computable features of video which include color similarity, shot activity and shot length to perform a higher level segmentation. We exploited the fact that shots that belong to one particular scene often have similar visual (color) and action (motion) attributes. The first method is a two-pass algorithm in which potential scene boundaries are found using only visual cues in the first pass. In the next pass, scenes are combined by examining their shot length and shot motion attributes. This is a supervised method which requires two thresholds to be specified for segmentation. The second method is however a single pass algorithm which is superior in that it considers all the shots and captures the global similarities of shots rather than the local similarities. It is unsupervised and does not require any explicit threshold. It is robust to noise and produces semantically meaningful scenes. The first method, which can be *tuned* for different movies provides better precision and recall figures. However, if the segmentation is required without any human intervention, the second method is preferred since there is no *tuning* necessary for different video genres. We also proposed a method to represent the scene content using only one key frame. We presented extensive experimental evaluation on several Hollywood movies, a TV sitcom and a talk show, which validate the proposed approaches. Thus, we have provided a complete system to organize huge amount of videos without any human intervention that can be utilized to offer online browsing facilities to the viewers in the electronic form. The research work discussed in this chapter has been published in referred conferences and workshops [47] and [46].

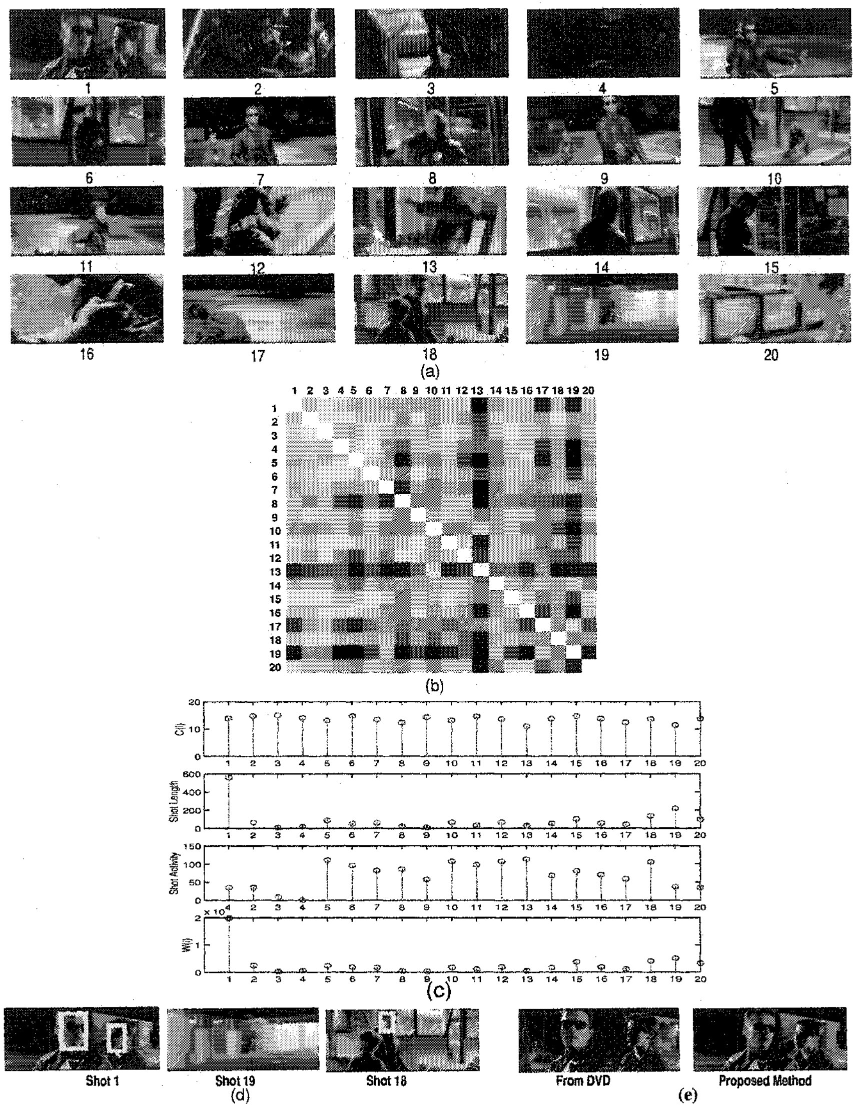


Figure 5.13: Best key frame detection for the scene content representation of a scene from the movie "Terminator II". The scene consists of 20 shots as shown in (a). (b) shows the correlation of shots with each other. (c) shows the plot of correlation sum, $C(i)$, *shot length*, *shot activity* and *shot goodness* measure of each shot. (d) shows three selected key frames with the highest goodness values. Rectangles show the detected faces in the images. (e) Compare the key frame selected by our algorithm (right) with the image obtained from the DVD (left).

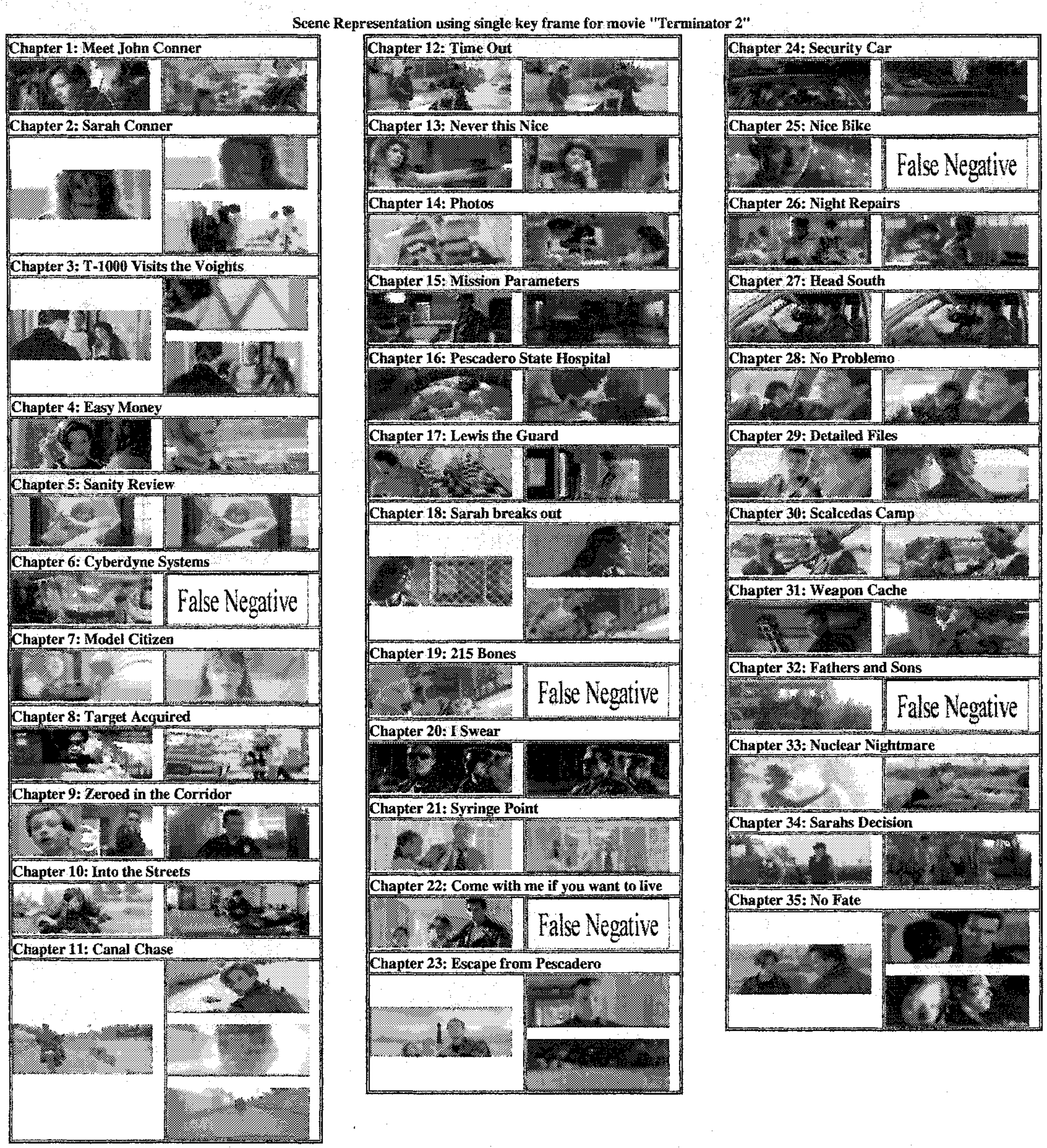

Scene Representation using single key frame for movie "Terminator 2"
Chapter 1: Meet John Conner
Chapter 2: Sarah Conner
Chapter 3: T-1000 Visits the Voights
Chapter 4: Easy Money
Chapter 5: Sanity Review
Chapter 6: Cyberdyne Systems
False Negative
Chapter 7: Model Citizen
Chapter 8: Target Acquired
Chapter 9: Zeroed in the Corridor
Chapter 10: Into the Streets
Chapter 11: Canal Chase
Chapter 12: Time Out
Chapter 13: Never this Nice
Chapter 14: Photos
Chapter 15: Mission Parameters
Chapter 16: Pescadero State Hospital
Chapter 17: Lewis the Guard
Chapter 18: Sarah breaks out
Chapter 19: 215 Bones
False Negative
Chapter 20: I Swear
Chapter 21: Syringe Point
Chapter 22: Come with me if you want to live
False Negative
Chapter 23: Escape from Pescadero
Chapter 24: Security Car
Chapter 25: Nice Bike
False Negative
Chapter 26: Night Repairs
Chapter 27: Head South
Chapter 28: No Problemo
Chapter 29: Detailed Files
Chapter 30: Scalcedas Camp
Chapter 31: Weapon Cache
Chapter 32: Fathers and Sons
False Negative
Chapter 33: Nuclear Nightmare
Chapter 34: Sarahs Decision
Chapter 35: No Fate

Figure 5.14: Scene representation using a key frame for the movie "Terminator II". In each column, images on the left side are the ones obtained from the DVD. Images on the right are the key frames selected by our algorithm. Multiple images have been shown for the chapters for which the scene is broken into multiple scenes by our algorithm. "False Negative" denotes that the corresponding scene was not detected by our scene detection algorithm.

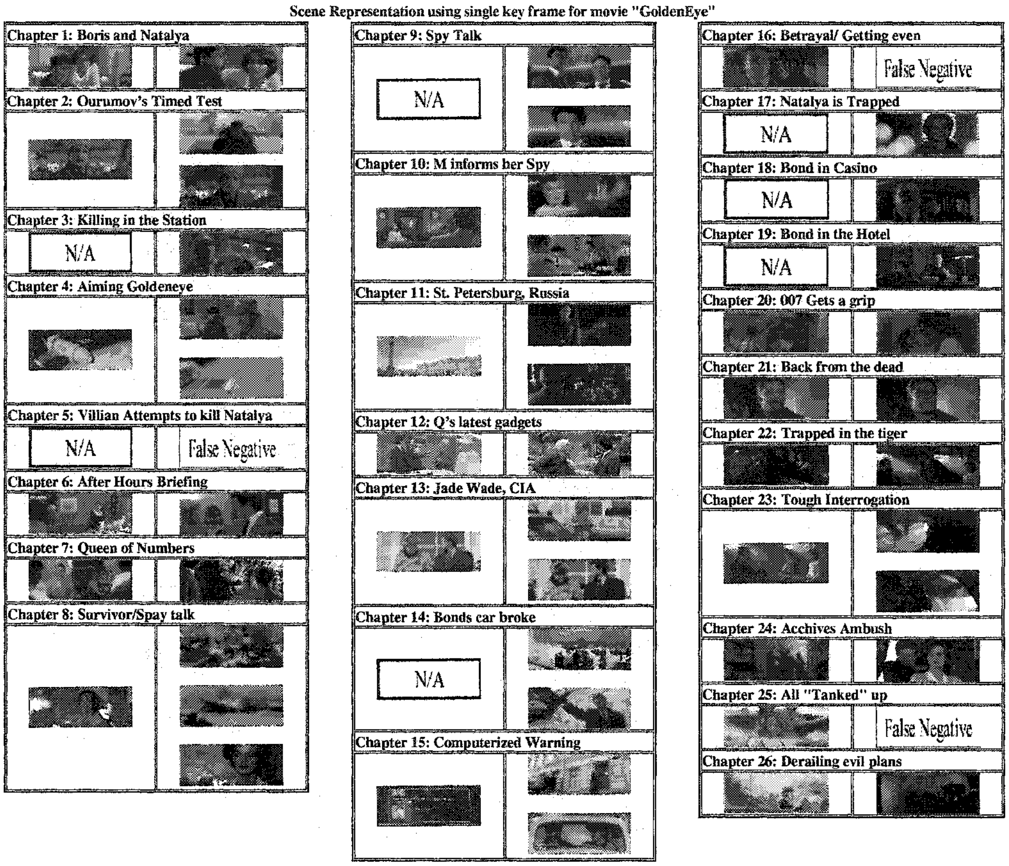

Scene Representation using single key frame for movie "GoldenEye"
Chapter 1: Boris and Natalya
Chapter 2: Ourumov's Timed Test
Chapter 3: Killing in the Station
N/A
Chapter 4: Aiming Goldeneye
Chapter 5: Villian Attempts to kill Natalya
N/A
False Negative
Chapter 6: After Hours Briefing
Chapter 7: Queen of Numbers
Chapter 8: Survivor/Spay talk
Chapter 9: Spy Talk
N/A
Chapter 10: M informs her Spy
Chapter 11: St. Petersburg, Russia
Chapter 12: Q's latest gadgets
Chapter 13: Jade Wade, CIA
Chapter 14: Bonds car broke
N/A
Chapter 15: Computerized Warning
Chapter 16: Betrayal/ Getting even
False Negative
Chapter 17: Natalya is Trapped
N/A
Chapter 18: Bond in Casino
N/A
Chapter 19: Bond in the Hotel
N/A
Chapter 20: 007 Gets a grip
Chapter 21: Back from the dead
Chapter 22: Trapped in the tiger
Chapter 23: Tough Interrogation
Chapter 24: Acchives Ambush
Chapter 25: All "Tanked" up
False Negative
Chapter 26: Derailing evil plans

Figure 5.15: Scene representation using a key frame for the movie "Golden Eye". In each column, images on the left side are the ones obtained from the DVD. Images on the right are the key frames selected by our algorithm. Multiple images have been shown for the chapters for which the scene is broken into multiple scenes by our algorithm. N/A denotes that the corresponding chapters were not present in the DVD and were segmented by a human observer as ground truth. "False Negative" denotes that the corresponding scene was not detected by our scene detection algorithm.

CHAPTER 6

CONCLUSION AND FUTURE WORK

We presented a systematic way of categorizing and organizing commercially produced videos. We demonstrated that low level features can be mapped to high level semantics using the domain knowledge. To bridge the gap between the lower and higher level information, we proposed a set of appropriate computable features and combined them with the production rules often followed by program directors. The important aspects of our work are the temporal segmentation and categorization of videos and the initial framework for semantic labelling. We showed that appropriate temporal segmentation is possible for commercially created videos and conducted experiments on Hollywood movies, sitcoms, talk and game shows. For talk and game show videos, we categorized shots into host and guests shots with the knowledge of production rules. Thus, no specific training was required for any particular show or host. We also built a framework to categorize movies into genres by exploiting the audio-visual computable features of the previews. It was demonstrated that when the previews are projected in the feature space, they form distinct cluster depending on their association with movie genres thus fortifying our claim that a mapping does indeed exist between low-level features and high-level concepts.

Based on this research, we propose to classify the segmented videos into semantic categories. We have shown that the computable features can be obtained for each shot in the scene. Thus, the average value of each feature for the whole scene can then be calculated and each scene can be represented by its characteristic feature vector. Since movie makers follow certain rules (called grammar) while making films, we can express higher level concepts in terms of our feature vector. For example, an action scene will generally have low shot length and high motion content. This part of the work requires expert knowledge, and we resort to the literature on professional film making to develop these rules. Using these rules, if the gap between high level movie concepts and low level computable features can be bridged, very meaningful and efficient indexing of movies will be possible. Such semantic labelling of scenes would allow far more flexibility to search movie databases and allow a user to get meaningful results from complex queries. A real world application of our proposed work is to find the semantic relevance of a given video with other videos in the database. For example, a computer can execute a scene level analysis of previously viewed movies and generate an automatic recommendation for a user. Furthermore, the movies could be rated on scene level rather than having one description for the entire video.

Current issues towards scene level categorization include the extraction of appropriate low level features. So far, we have focused on features which are global in nature. For example, key-lighting or shot length. However, an in depth study on semantic interpretation would require feature extraction at finer levels, such as image segmentation [31], foreground object detection [28], object classification and action recognition [44]. The computer vision community has produced much research work in tracking objects in single camera views [29] to multiple cameras [26, 30] for various purposes, for example surveillance and behavior analysis. We intend to utilize these methods so that features can be computed precisely. We also realize the fact that a scene could also be represented by a combination of two or more semantic labels. Therefore, we will develop a weighted scheme of features based on experience in film analysis, to grant relative importance to each feature in identifying a particular genre. For example, color key is an important feature in labelling something as a explosion/gun fire, but not so important in classification of conversational scenes. Development and testing of suitable features as well as weighing scheme will be an important part of our future work.

LIST OF REFERENCES

[1] B. Adams, C. Dorai, and S. Venkatesh. Towards automatic extraction of expressive elements from motion pictures: tempo. In *IEEE International Conference on Multimedia and Expo*, pages 641–644, 2000.

[2] Aya Aner and John R. Kender. Video summaries through mosaic-based shot and scene clustering. In *European Conference on Computer Vision*, pages 388–402.

[3] Apple. http:// www.apple.com/ trailers/.

[4] Daniel Arijon. *Grammar of the Film Language.* Hasting House Publishers, NY, 1976.

[5] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *IEEE International Conference on Computer Vision*, volume 2, pages 408–415, July 2001.

[6] Internet Movie Data Base. http:// www.imdb.com/.

[7] A. B. Benitez, H. Rising, C. Jrgensen, R. Leonardi, A. Bugatti, K. Hasida, R. Mehrotra, A. Murat Tekalp, A. Ekin, and T. Walker. Semantics of Multimedia in MPEG-7. In *IEEE International Conference on Image Processing*, 2002.

[8] J. S. Boreczky and L. D. Wilcox. A hidden Markov model framework for video segmentation using audio and image features. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1997.

[9] S. F. Chang, W. Chen, H.J. Horace, H. Sundaram, and D. Zhong. A fully automated content based video search engine supporting spatio-temporal queries. *IEEE Transaction on Circuits and Systems for Video Technology*, pages 602–615, 1998.

[10] C. Colombo, A. Del Bimbo, and P Pala. Retrieval of commercials by video semantics. In *IEEE Computer Vision and Pattern Recognition*, pages 572–577, 1998.

[11] Dorin Comaniciu and Peter Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[12] Daniel DeMenthon, Longin Jan Latecki, Azriel Rosenfeld, and Marc Vuilleumier Stckelberg. Relevance ranking of video data using hidden Markov model distances and polygon simplification. In *Advances in Visual Information Systems*, pages 49–61, 2000.

[13] Y. Deng and B. S. Manjunath. Content-based search of video using color, texture and motion. In *IEEE Intl. Conf. on Image Processing*, pages 534–537, 1997.

[14] N. Dimitrova, L. Agnihotri, and G. Wei. Video classification based on HMM using text and faces. In *European Conference on Signal Processing*, 2000.

[15] S. Fischer, R. Lienhart, and W. Effelsberg. Automatic recognition of film genres. In *Third ACM International Multimedia Conference and Exhibition*, pages 367–368, 1995.

[16] N. C. Haering, R.J. Qian, and M.I. Sezan. A semantic event detection approach and its application to detecting hunts in wildlife video. *IEEE Transaction on Circuits and Systems for Video Technology*, 1999.

[17] Niels Haering. A framework for the design of event detections, (Ph.D. thesis), 1999. School of Computer Science, University of Central Florida.

[18] A. Hampapur, A. Gupta, B. Horowitz, C. F. Shu, C. Fuller, J.R. Bach, M. Gorkani, and R.C. Jain. Virage video engine. In *SPIE, Storage and Retrieval for Image and Video Databases*, pages 188–198, 1997.

[19] Alan Hanjalic, Reginald L. Lagendijk, and Jan Biemond. Automated high-level movie segmentation for advanced video-retrieval systems. *IEEE Transaction on Circuits and Systems for Video Technology*, 9(4):580–588, June 1999.

[20] A. G. Haupmann and M. J. Witbrock. Story segmentation and detection of commercials in broadcast news video. In *Proceedings of the Advances in Digital Libraries Conference*, 1998.

[21] Chung-Lin Huang and Chih-Yu Chang. Video summarization using hidden Markov model. In *International Conference on Information Technology: Coding and Computing*, pages 473–477, 2001.

[22] Informedia. *Informedia Project, Digital video library.* http:// www. informedia. cs.cmu.edu.

[23] A. K. Jain, R. P. W. Duin, and Jianchang Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, Jan 2000.

[24] Omar Javed, Sohaib Khan, Zeeshan Rasheed, and Mubarak Shah. A framework for segmentation of interview videos. In *IASTED International Conference on Internet and Multimedia Systems and Applications*, 2000.

[25] Omar Javed, Sohaib Khan, Zeeshan Rasheed, and Mubarak Shah. Visual content based segmentation of talk and game shows. *International Journal of Computers and Applications*, June 2002.

[26] Omar Javed, Zeeshan Rasheed, Khurram Shafique, and Mubarak Shah. Tracking across multiple cameras with disjoint views. In *IEEE International Conference on Computer Vision*, 2003.

[27] Omar Javed, Zeeshan Rasheed, and Mubarak Shah. A framework for segmentation of talk and game shows. In *IEEE International Conference on Computer Vision*, pages 532–537.

[28] Omar Javed, Khurram Shafique, and Mubarak Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *IEEE Workshop on Motion and Video Computing*, 2002.

[29] Omar Javed and Mubarak Shah. Tracking and object classification for automated surveillance. In *European Conference on Computer Vision*, 2002.

[30] Sohaib Khan, Omar Javed, Zeeshan Rasheed, and Mubarak Shah. Human tracking in multiple cameras. In *IEEE International Conference on Computer Vision*, 2001.

[31] Sohaib Khan and Mubarak Shah. Object based segmentation of video using color, motion and spatial information. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2001.

[32] R. Kjeldsen and J. Kender. Finding skin in color images. In *International Conference on Face and Gesture Recognition*, 1996.

[33] V. Kobla, D. S. Doermann, and C. Faloutsos. Videotrails: Representing and visualizing structure in video sequences. In *Proceedings of ACM Multimedia Conference*, pages 335–346, 1997.

[34] Jia Li and James Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10), 2003.

[35] Yuehu Liu, H. Emoto, T. Fujii, and S. Ozawa. A method for content-based similarity retrieval of images using two dimensional DP matching algorithm. In *11th International Conference on Image Analysis and Processing*, pages 236–241, 2001.

[36] Cheng Lu, Mark S. Drew, and James Au. Classification of summarized videos using hidden Markov models on compressed chromaticity signatures. In *ACM International Conference on Multimedia*, 2001.

[37] Peter Lyman and Hal R. Varian. *School of Information Management and Systems at the University of California at Berkeley*. 2000. http:// www.sims. berkeley.edu/ research/ projects/ how-much-info/.

[38] J. Nam, M. Alghoniemy, and A. H. Tewfik. Audio-visual content based violent scene characterization. In *IEEE International Conference on Image Processing*, pages 353–357, 1998.

[39] Milind R. Naphade and Thomas S. Huang. A probabilistic framework for semantic video indexing, filtering, and retrieval. *IEEE Transactions on Multimedia*, pages 141–151, 2001.

[40] W. Ngo, T. C. Pong, and H. J. Zhang. Motion-based video representation for scene change detection. *International Journal of Computer Vision*, 2001.

[41] N. V. Patel and I. K. Sethi. *The Handbook of Multimedia Information Management.* Prentice-Hall/PTR, 1997.

[42] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[43] R. Qian, N. Haering, and I. Sezan. A computational approach to semantic event detection. In *IEEE Computer Vision and Pattern Recognition*, pages 200–206, 1999.

[44] Cen Rao, Alper Yilmaz, and Mubarak Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2), 2002.

[45] Zeeshan Rasheed and Mubarak Shah. Movie genre classification by exploiting audio-visual features of previews. In *International Conference on Pattern Recognition*, 2002.

[46] Zeeshan Rasheed and Mubarak Shah. A graph theoretic approach for scene detection in produced videos. In *Multimedia Information Retrieval Workshop*, 2003.

[47] Zeeshan Rasheed and Mubarak Shah. Scene detection in Hollywood movies and TV shows. In *IEEE Computer Vision and Pattern Recognition*, 2003.

[48] Zeeshan Rasheed and Mubarak Shah. *Video Mining Techniques: Video Categorization using Semantics and Semiotics.* KLUWER Academic Publishers, 2003.

[49] Zeeshan Rasheed, Yaser Sheikh, and Mubarak Shah. On the use of computable video features for film classification. *IEEE Transaction on Circuits and Systems for Video Technology.*

[50] Zeeshan Rasheed, Yaser Sheikh, and Mubarak Shah. Classification using low-level computable features. In *3rd International Workshop on Multimedia Data and Document Engineering*, 2003.

[51] A. F. Reynertson. *The Work of the Film Director.* Hasting House Publishers, NY, 1970.

[52] Wolf Rilla. *A-Z of movie making, A Studio Book.* The Viking Press, NY, 1970.

[53] D. Roth, M. Yang, and N. Ahuja. A snow based face detector. *Neural Information Processing*, 2000.

[54] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, Jan 1998.

[55] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Constructing table-of-content for videos. *ACM Multimedia Systems Journal, Special Issue Multimedia Systems on Video Libraries*, September 1999.

[56] Sudeep Sarkar and Padmanabhan Soundararajan. Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):504–525, May 2000.

[57] H. Schweitzer. Template matching approach to content based image indexing by low dimensional euclidean embedding. In *Eight IEEE International Conference on Computer Vision*, pages 566–571, 2001.

[58] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), August 2000.

[59] John R. Smith. Videozoom spatio-temporal video browser. *IEEE Transactions on Multimedia*, 1(2):157–171, June 1999.

[60] M. A. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding techniques, 1997.

[61] T. Sobchack and V. Sobchack. *An introduction to Film.* 1987.

[62] K.K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, Jan 1998.

[63] B. T. Truong, S. Venkatesh, and C. Dorai. Automatic genre identification for content-based video categorization. In *IEEE International Conference on Pattern Recognition*, 2000.

[64] A. Vailaya, M.A.T. Figueiredo, A. K. Jain, and Hong-Jiang Zhang. Image classification for content-based indexing. *IEEE Transactions on Image Processing*, 10(1):117–130, 2001.

[65] N. Vasconcelos and A. Lippman. Statistical models of video structure for content analysis and characterization. *IEEE Transactions on Image Processing*, 9(1):3–19, Jan 2000.

[66] Howhard D. Wactlar. The challanges of continuous capture; contemporaneous analysis, and customzed summarization of video content.

[67] Webster. *Webster Dictionary.* http:// www.m-w.com.

[68] W. Wolf. Hidden Markov model parsing of video programs. In *International Conference on Acoustics, Speech and Signal Processing*, pages 2609–2611, 1997.

[69] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.

[70] B. L. Yeo and B. Liu. Rapid scene change detection on compressed video. 5:533–544, IEEE Transaction on Circuits and Systems for Video Technology.

[71] M. M. Yeung, B.-L. Yeo, and B. Liu. Segmentation of video by clustering and graph analysis. *Computer Vision and Image Understanding*, 71(1), 1998.

[72] Herbert Zettl. *Sight Sound Motion: Applied Media Aesthetics.* Wadsworth Publishing Company, second edition, 1990.

[73] Junyu Zhou and Wallapak Tavanapong. Shotweave: A shot clustering technique for story browsing for large video databases. In *International Workshop on Multimedia Data Document Engineering*, March 2002.