# Content-Based Movie Analysis and Indexing Based on AudioVisual Cues

Ying Li, *Member, IEEE*, Shrikanth Narayanan, *Senior Member, IEEE*, and C.-C. Jay Kuo, *Fellow, IEEE*

***Abstract*—A content-based movie parsing and indexing approach is presented in this paper, which analyzes both audio and visual sources and accounts for their interrelations to extract high-level semantic cues. Specifically, the goal of this work is to extract meaningful movie events and assign them semantic labels for the content indexing purpose. Three types of key events, namely, 2-speaker dialogs, multiple-speaker dialogs, and hybrid events, are considered in this work. Moreover, speakers present in the detected movie dialogs are further identified based on the audio source parsing. The obtained audio and visual cues are then integrated to index the movie content. Our experiments have shown that an effective integration of the audio and visual sources can lead to a higher level of video content understanding, abstraction and indexing.**

***Index Terms*—Audiovisual integration, content-based video indexing, movie event detection, silence detection, speaker identification, video content analysis, video segmentation.**

## I. Introduction

With the fast growth of multimedia information, content-based video analysis, indexing and retrieval have attracted increasing attention in recent years. Many applications have emerged in areas such as video-on-demand, distributed multimedia systems, digital video libraries, distance education, entertainment, surveillance and geographical information systems [1]. The need for content-based video indexing and retrieval has also been foreseen by ISO/MPEG that found the basis for the definition of a new international standard: "Multimedia Content Description Interface," in short, MPEG-7 [1].

Content-based video analysis aims at obtaining a structured organization of the original video content and understanding its embedded semantics like humans do. Content-based video indexing is the task of tagging semantic video units obtained from content analysis to enable convenient and efficient content retrieval. However, although content understanding is an easy task for humans, it is very difficult for a computer to emulate because of the limitations of machine perception under unconstrained environments and the unstructured nature of video data. Robust techniques are still lacking today despite a large amount of effort in this area [2]–[6].

So far, the predominant approach to this problem is to first extract some low- to mid-level audiovisual features, then partially derive or understand the video semantics by analyzing and integrating these features. Fig. 1 shows a hierarchical video indexing structure, where many popularly used features such as color, texture, shape, motion, shots [7], keyframes [2], object trajectories, human faces [6], as well as classified audio classes [8], constitute the low- to mid-level indexing features. Obviously, a semantic gap still exists between the real video content and the video contexts derived from these features.

This work proposes to extract two types of video indexing features, namely, *video events* and *speaker identity*, at the semantic level based on the integration of audio and visual knowledge. The movie content is the major focus of this work. Three types of events have been considered, which are *2-speaker dialogs*, *multiple-speaker dialogs*, and *hybrid events*. The extracted event information can be utilized to facilitate movie content browsing, abstraction and indexing, since these events have retained the most informative parts of the movie. In the second stage, we proceed to identify target speakers from movie dialogs so as to index the movie content with recognized cast names.

The rest of this paper is organized as follows. We first give an overview of our approach and compare it with existing techniques in Section II. Low-level audiovisual content analysis is briefly reviewed in Section III. Section IV elaborates on the work of movie event extraction and characterization. In Section V, we give details on the speaker identification work. Experimental results obtained from three test movies are reported and discussed in Section VI, and finally concluding remarks are drawn in Section VII.

## II. Approach Overview

Automatic video content understanding and indexing is a difficult problem, and is only tractable in specific application domains such as sports and news. For generic video such as movies, the content can be abstracted into a series of events, where an *event* is defined as a video paragraph which contains a meaningful theme and usually progresses under a consistent environment. On the other hand, speaker identity is another important type of information that can effectively index the movie content. Therefore, we propose to index the movie content based on extracted key events and identified key speakers as shown in Fig. 2.

The key research issues lie in: 1) linking low-level audiovisual features to semantic events and 2) identifying speakers in an adverse environment with various background sounds. These issues are discussed in the following subsections.

Manuscript received December 31, 2002; revised March 21, 2003 and May 31, 2003. This research was supported in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, under Cooperative Agreement Number EEC-9529152 and in part by the Hewlett-Packard Company. This paper was recommended by Associate Editor F. Pereira.

Y. Li was with the Integrated Media Systems Center and Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2564 USA and is now with the IBM T. J. Watson Research Center, Hawthorne, NY 10532 USA (e-mail: yingli@us.ibm.com).

S. Narayanan and C.-C. J. Kuo are with the Integrated Media Systems Center and Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2564 USA (e-mail: shri@sipi.usc.edu; cckuo@sipi.usc.edu).

Digital Object Identifier 10.1109/TCSVT.2004.831968

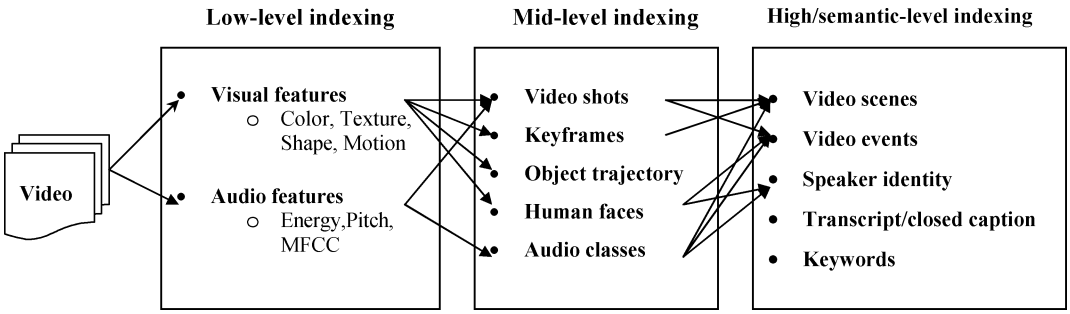



Fig. 1. Generic three-level video indexing structure where arrows between nodes indicate a causal relationship.

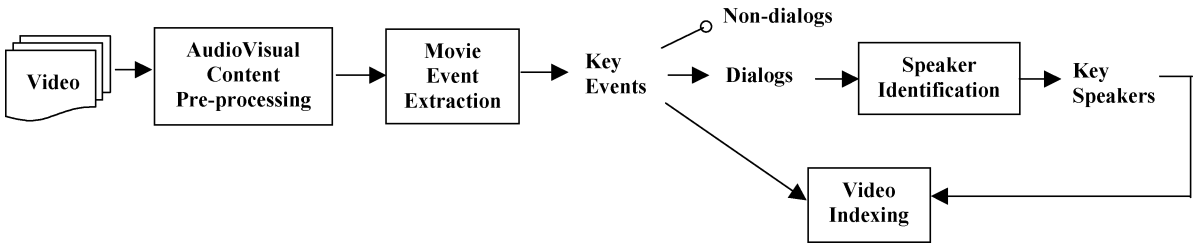



Fig. 2. Overview of the proposed system framework.

### A. Event Detection and Extraction

Since the concept of event is rather subjective, we will first review related previous work that addresses similar concepts or has similar research goals.

Shot detection, where a *shot* is defined as a set of contiguously recorded image frames, is usually the first step toward video content understanding. So far, many research work has been published on shot detection in both compressed and uncompressed video domains [7], [9]–[12]. However, while the shot forms the building block of the video content, this low-level structure does not correspond to the underlying video semantics in a direct and convenient way.

Recent work starts to understand the video semantics based on extracted video *scenes*, where a scene is defined as a collection of semantically related shots that depict and convey a high-level concept or story. For instance, Rui and Yeung [13], [14] proposed to extract video scenes by grouping visually similar and temporally adjacent shots. In [15]–[17], the temporal and spatial structures of TV news were predefined to help understand the video content. To take the advantage of multiple media sources, Huang *et al.* [18] proposed to detect scenes by analyzing the changes in all audio, visual and motion contents. Similar ideas were also explored in the Informedia project [19] where audio, image and text keywords were combined to determine scenes. In [20], continuing background sounds, similar color settings and orientations were used as scene indicators. In [21], video shots were combined with speaker change detection to locate scenes in TV news. Sundaram and Chang [5] reported their work on extracting two types of computable scenes (N-type and M-type) in feature films by combining audio and visual cues. The N-type scene was then further classified into pure dialog, progressive scene and hybrid scene.

However, while a scene does provide a higher-level video context than a shot, not every scene contains a meaningful thematic topic, especially for movies where progressive scenes, which are frequently inserted to establish the story situation, are actually unimportant for content understanding. Therefore, we need a video unit to operate at a higher semantic level so as to better reveal, represent and abstract the content. Such a unit is called *event*, and its extraction is one of the major concerns of this work.

There have been some efforts reported on the detection of event, although it has taken on different meanings. For example, Mahmood and Srinivasan [22] presented a query-driven approach to detect discussion topics using image and text contents of query foils (slides) found in a lecture. In [23], highlights of a baseball game were extracted by detecting the announcer and audiences' speech, the game-specific sounds and various other background noise. Also targeting at sports video, Chang *et al.* [24] applied both image and speech analysis to locate the touch-down points in football games. Similar work in the sports domain can also be found in [25] and [26], where heuristics of tennis and basketball game structures were employed to guide the highlight extraction process.

In contrast, this work focuses on extracting events from movies, which has not been well explored in this content domain. The reason we choose the movie application is that it has a clear story structure and can be well exploited by our approach. Moreover, a movie has many special characteristics, such as the complex film editing techniques required to produce a successful movie [27]. Therefore, it is not only interesting but also challenging to work with movies, since all these special features need to be taken into account for a better content understanding.

Because a movie plot is usually developed through either dialogs or actions, we identify the following three types of events in this research: *the 2-speaker dialogs*, *the multiple-speaker dialogs*, and *the hybrid events* which accommodate for events with less speech and more visual action. The detection of dialogs has been explored by some previous work. For instance, Yeung and Yeo [4] characterized a temporal event into either dialog, action or others. In particular, they detected a dialog by searching a shot sequence with a repetitive nature of two dominant shots such

as "A B A B A B" no mater whether there is a true conversation going on or not. A similar periodic analysis transform was also employed in [5] for dialog detection. However, since the arrangement of shot sequences in a dialog basically varies with the film genre and also heavily depends on the directorial style, strict periodic analysis appears to be too restrictive for a general scenario. In addition, the problem becomes more complex when multiple speakers are present. Finally, the speech information, which is an important indicator for dialogs, was not considered in both [4] and [5]. Therefore, false alarms may occur when a nonconversational scene presents a repetitive shot structure.

In this paper, we will try to address these problems and accomplish the three-type event extraction task by analyzing the movie content structure and exploiting film's special editing features. Specifically, we first group visually similar and temporally close shots into shot sinks using the proposed "window-based sweep algorithm". Then all sinks are clustered and characterized into three categories (periodic, partly-periodic, and nonperiodic) using unsupervised k-means algorithm. Finally, events are extracted from the sink structure and a post-processing step is carried out which employs both speech and face cues to reduce the false alarms. In a summary, compared to previous work [4] and [5], this work has developed an approach which considers true dialog scenarios (i.e. conversational scenes with human dialogs), more complex dialog scenarios (i.e. dialog scenes with irregular shot structures) as well as the existence of multiple speakers within a dialog.

It is worthwhile to point out that our proposed "window-based sweep algorithm" share similar ideas with the "time-adaptive shot grouping" approach proposed in [13] and the "time-constrained shot clustering" approach proposed in [14]. That is, they are all applied to group visually similar and temporally adjacent shots into either sinks, groups or clusters. However, our work is different from [13] and [14] in three ways. First, the major goal of both [13] and [14] was to detect video scenes based on either shot groups or shot clusters. Thus, they did not attempt to analyze the scene content and classify the scene type. In contrast, in our work, once an event is extracted, we go one step further by categorizing it into either 2-speaker dialog, multiple-speaker dialog or hybrid event based on the event content analysis. Thus, our work attempts to reveal more video semantics than [13] and [14] do. Second, Yeung *et al.* [14] constructed a scene transition graph (STG) to extract the story unit, which is however not needed in our work since video paragraphs that present a sequential content will naturally form event delimiters as discussed in Section IV-C. Third, no audio or face information has been employed in [13] or [14], while our work utilizes both information to improve the event detection accuracy. To summarize, although we have used some visual processing techniques that are similar to those presented in [13] and [14] in generating shot sinks as intermediate entities, yet since our ultimate goal is to extract and characterize video events instead of detecting scenes, new algorithms have been developed, and more media cues such as audio and face information have been integrated to fulfill this task, which has made our work different from them.

### *B. Speaker Identification*

Automatic speaker identification has been an active research topic for many years with bulk of the progress facilitated by work on standard speech databases such as YOHO, HUB4, and SWITCHBOARD [28]. For instance, in [29], a speaker detection algorithm based on a likelihood ratio calculation was developed to estimate the target speaker segments from the HUB4 broadcast news database. In [30], the problem of labeling speaker turns by automatically segmenting and clustering a continuous audio stream was addressed. An efficiency of 70% was obtained on the 1996 Hub4 development data. In [31], the performance of the support vector machine (SVM) on speaker verification and identification tasks was assessed on the YOHO database. [32] reported its work on speaker change detection where the speaker model was created from successive utterance as a codebook using vector quantization.

There has been some recent work on identifying speakers for video content analysis based on audiovisual cues. For instance, Tsekeridou and Pitas [6] proposed to identify speakers by integrating cues from both speaker recognition and facial analysis schemes. This system is, however, impracticable for generic video types since it assumes there is only one human face in each video frame. Similar work was also reported by [33], where a person is identified based on the integration of both speaker and face cues. Encouraging results were reported on a TV sitcom, yet the system performance needs further verification on more complex video sources.

This work identifies target speakers in movie dialogs based on the integration of both audio and visual cues. A maximum-likelihood-based approach is employed for identification purpose. Moreover, Gaussian mixture models (GMMs) are chosen to build speaker models. Finally, noting that there are various kinds of ambient sounds in feature films, we have developed an adaptive silence detector to isolate individual speech segments from the background, which also makes this work different from the previous.

## III. Audio and Visual Content Pre-Analysis

The first step toward visual content analysis is shot detection. In this work, we employ a color histogram-based approach to fulfill this task [34], which has achieved an average of 92.5% precision and 99% recall rates. In the second step, we proceed to extract one or more keyframes from each shot to represent its underlying content. To achieve fast processing speed, currently we assign the first and last frames of each shot as its keyframes without sacrificing the system performance.

The audio content analysis mainly deals with audio content classification, where each shot is classified into one of the following four classes: *silence*, *speech*, *music*, and *environmental sounds*. Five audio features are extracted for the classification purpose which include the short-time energy function, the short-time average zero-crossing rate, the short-time fundamental frequency, the energy band ratio and the silence ratio. An average of 88% classification accuracy has been achieved in the current work. More detailed description of this part can be found in [35].

Facial analysis is also performed to detect human faces in the frontal view or faces rotated by plus or minus 10 degrees from the vertical direction. Currently, we use the face detection library provided by the HP Labs [36], which reports a detection accuracy of 85%. However, due to the complex motions of movie casts, we may get both high false negatives and false

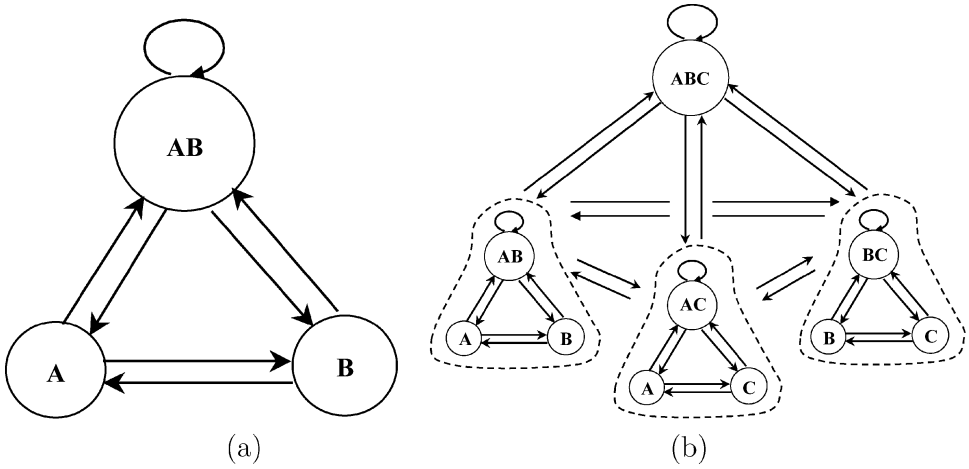



Fig. 3. Typical movie dialog models for: (a) a 2-speaker dialog (speakers A and B), and (b) a multiple-speaker dialog (speakers A, B, and C), where node A means the shot contains speaker A.

positives. Since false negatives do not severely affect the system performance, we mainly focus on reducing the false positives. In particular, a simple face tracking system is applied in the current work, where only faces appearing in several consecutive frames are retained.

## IV. Movie Event Extraction

Movie, known as a recording art, is practical, environmental, pictorial, dramatic, narrative and musical [37]. Because a film operates in a limited period of time, all movie shots are efficiently organized by a film-maker in such a way that audiences will follow his or her own way of story-telling. Specifically, this goal is achieved by presenting audiences a sequence of cascaded events that gradually develop the movie plot. In this work, we consider the underlying event as the basic movie story unit.

There are basically two ways to develop a thematic topic in an event: through actions where recorded movements tell the story or through dialogs where words carry out the theme [27]. Based on the film genre and film-makers' directorial flavor, either or both of these two styles could be frequently used. However, no matter which filming style is used, they share one common feature, i.e. certain shots will present a repetitive visual structure. For instance, during a chase sequence, we frequently see shots of the pursued and the pursuer despite a constantly changing background. This repetitive pattern is even more distinct in a dialog scene, which is the result of the so-called *montage* effect as described in [38], "*One of the binding and immutable conditions of cinema is that actions on the screen have to be developed sequentially, regardless of the fact of being conceived as simultaneous or retrospective .... In order to present two or more processes as simultaneous or parallel, you have to show them one after the other, they have to be in sequential montage*." This means that, in order to convey conversations, innuendos or reactions, film-makers have to repeat important shots to express the content and motion continuity. This feature will be employed to detect the three-target events, i.e. the 2-speaker dialog, the multiple-speaker dialog and the hybrid event, where a dialog refers to an actual conversation between two or more people in this work.

Fig. 3 gives two dialog models that are constructed based on the analysis of the movie dialog editing styles [27]. Specifically, Fig. 3(a) models a 2-speaker dialog, and (b) models a multiple-speaker dialog (here we use three speakers as an example). Each node in the figure represents a shot that contains the indicated speaker(s), and arrows are used to denote the switches between two shots. From these two models, we see that there are certain shot repeating patterns in both cases, although the former one presents more periodic patterns than the latter one since fewer speakers are involved. Based on this observation, we propose to extract movie events in the following four steps:

1) shot sink computation, where a sink contains temporally close and visually similar shots;
2) sink clustering and characterization, where each sink is recognized to be either periodic, partly-period, or nonperiodic;
3) event extraction and classification;
4) post-processing based on integrated speech and face cues.

Each of these steps is detailed in the following subsections.

### A. Computing Shot Sinks Using Visual Information

Since an event is generally characterized by a repetitive visual structure, our first step is to extract all video paragraphs that possess this feature. A new concept called *shot sink* is defined for this purpose. Particularly, a shot sink contains a pool of shots which are temporally close and visually similar. Shot sinks are generated using the proposed window-based sweep algorithm as described below.

*1) Window-Based Sweep Algorithm:* Given shot $i$, this algorithm finds all shots that are visually similar to $i$, and push them into its sink. However, since an event practically occurs within a certain temporal locality, we naturally restrict the search range to a window of length $\mathrm{win}L$ as shown in Fig. 4(a). To compare the visual similarity of two shots, in principle we should compare every pair of video frames, with each being taken from one shot. However, due to the inherent complexity in such an operation, keyframes are usually used in the place of regular frames. This is acceptable since keyframes could be seen as the shot representatives in most cases. One thing worth mentioning is, although currently we use the shot's first and last two frames

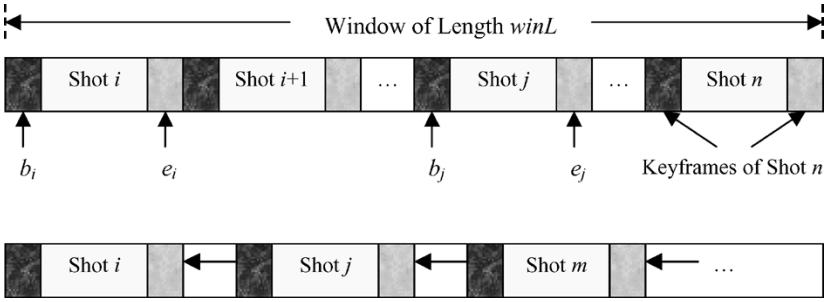



Fig. 4. (a) Shots contained in a window of length $\mathrm{win}L$. (b) Computed sink of shot $i$.

as its keyframes, yet more complex keyframe extraction algorithms could be applied and integrated into the system.

Denote shots $i$ and $j$'s keyframes by $b_i, e_i$, and $b_j, e_j$ $(i < j)$ as shown in Fig. 4(a), we compute the similarity between shots $i$ and $j$ as

$$\mathrm{Dist}_{i,j} = \frac{1}{4}\left(w_1 \times \mathrm{dist}(b_i, b_j) + w_2 \times \mathrm{dist}(b_i, e_j) + w_3 \times \mathrm{dist}(e_i, b_j) + w_4 \times \mathrm{dist}(e_i, e_j)\right) \quad (1)$$

where $\mathrm{dist}(b_i, b_j)$ could be either the Euclidean distance or the histogram intersection between $b_i$ and $b_j$'s color histograms. $w_1$, $w_2$, $w_3$, and $w_4$ are four weighting coefficients computed as

$$w_1 = 1 - \frac{L_i}{\mathrm{win}L} \qquad w_2 = 1 - \frac{L_i + L_j}{\mathrm{win}L}$$
$$w_3 = 1 \qquad w_4 = 1 - \frac{L_j}{\mathrm{win}L} \quad (2)$$

where $L_i$ and $L_j$ are shot lengths in the unit of frames. The derivation of these four coefficients is explained as follows. First, due to the "montage" effect, we know that when shots $i$ and $j$ are within the same thematic topic, they share certain visual similarity although shot $j$ further advances shot $i$'s content. Therefore, in order to test the content similarity between shots $i$ and $j$, we shall first check the similarity between $e_i$ and $b_j$ since they form the closest frame pair, and should have the smallest distance if shot $j$ does continue shot $i$'s content. Thus we set $w_3$ to be 1. On the contrary, the similarity between $b_i$ and $b_j$ becomes smaller as shot $i$ gets longer, therefore, the distance between them does not help us as much to determine if shots $i$ and $j$ are similar. Hence, we set $w_1$ to be $1 - (L_i/\mathrm{win}L)$ where $\mathrm{win}L$ is introduced for the normalization purpose. We can derive the formulas for $w_2$ and $w_4$ in similar ways.

Now, if $\mathrm{Dist}_{i,j}$ is less than a predefined threshold $\mathrm{shot}T$, we consider shots $i$ and $j$ to be similar, and put shot $j$ into shot $i$'s sink. As shown in Fig. 4(b), all shots similar to shot $i$ are nicely linked together in their temporal order. One thing worth mentioning is that if shot $i$'s sink is not empty, we have to compute distances from the current shot, say, shot $m$, to all other resident shots in the sink (shots $i$ and $j$ in this case), and shot $m$ is only qualified to be in the sink when the average of all distances is less than $\mathrm{shot}T$.

Basically we will run this algorithm for every shot. However if one shot has already been included in a sink, we will skip this shot and continue with the next.

Two parameters are used in this algorithm, i.e. the window length $\mathrm{win}L$ and the threshold $\mathrm{shot}T$. Below are some discussions on how to determine them.

1) *Determining Window Length* $\mathrm{win}L$: We have tried two ways to choose parameter $\mathrm{win}L$, namely, a fixed value and an adaptive value that varies with every incoming movie. In the former case, we empirically set $\mathrm{win}L$ to be a predefined value that covers the duration of an ordinary movie scene. In the latter case, $\mathrm{win}L$ is set to be proportional to the average shot length. Hence, the faster the movie tempo, the shorter the window length. Based on our experiments, we find that a fixed value usually produces better results, which is perhaps due to the reason that as a semantic unit, scene is somehow independent of the underlying shot structure. $\mathrm{win}L$ is empirically set to be 2000 (*frames*) in the current work based on experimental results.

2) *Determining Threshold* $\mathrm{shot}T$: Parameter $\mathrm{shot}T$ is used to threshold the similarity measurement between two shots. Since our distance metric employs color information, and since different movies tend to have different primary hue, an empirically set threshold may not always work. Fig. 5(a) shows a shot distance histogram for one test movie where each distance is computed from one shot to another within the temporal window. As we can see, a Gaussian density function $N(\mu;\sigma)$ can be used to approximate this distance distribution. Inspired by this finding, we propose to determine the threshold as follows. First, we normalize each computed distance $\mathrm{Dist}_{i,j}$ with $\mu$ and $\sigma$, i.e. $\mathrm{Dist}_{i,j} = [(\mathrm{Dist}_{i,j} - \mu)/\sigma]$; then, we compare it with another threshold $\mathrm{shot}T'$ which is derived from the Gaussian density function. Parameter $\mathrm{shot}T'$ can be easily adjusted to fit all movies since it applies to normalized distances. Empirically, we find that $\mathrm{shot}T' = -1.35$ produces a good result for all test data, where about 9% of the shots in the timing window are qualified for the sink since $P(X < x | x = \mathrm{shot}T' = -1.35) = 0.089$ as shown in Fig. 5(b).

### *B. Clustering Shot Sinks Using K-Means Algorithm*

In this stage, we will cluster and characterize each sink into one of the following three predefined classes: *periodic*, *partly-periodic* and *nonperiodic*. The evaluated shot repetition degree is used to make this decision. For instance, if shot $i$'s sink contains shots $i, i+2, i+4$ and $i+6$, we will classify it into the first class since a very strict shot repetition pattern is observed. This situation could appear frequently in a 2-speaker dialog event. For the sink in the second class, although it usually presents partial periodicity, the periodic pattern may not be strictly adhered.

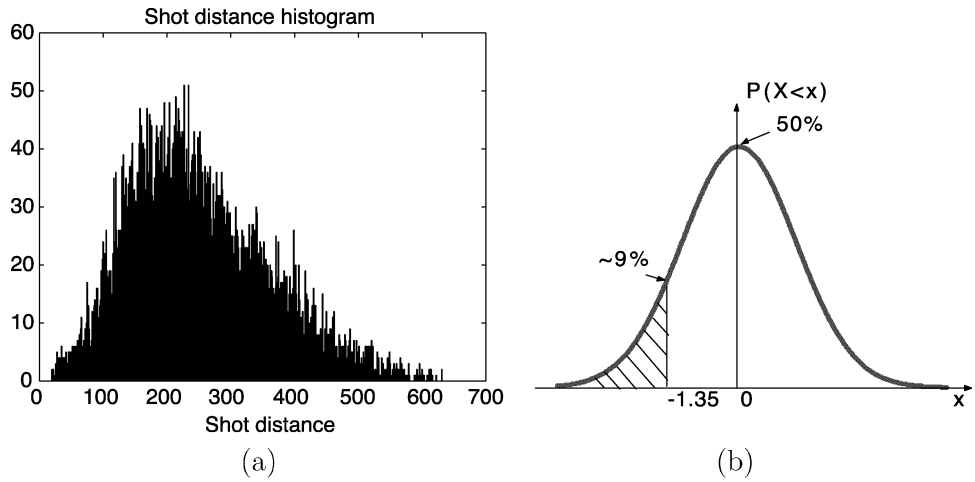



Fig. 5. (a) Shot distance histogram for a test movie. (b) Normalized distribution of shot differences.

This scenario often occurs in a multiple-speaker dialog event. For the last class, it is called nonperiodic since no specific conclusion can be made regarding its shot distribution pattern. If shot $i$'s sink only contains itself, this sink will be discarded and excluded from further consideration.

The following three processing steps are employed to quantitatively determine the sink periodicity.

1) For each sink, calculate the relative temporal distance between each pair of neighboring shots. For example, if shot $i$'s sink contains shots $i$, $i+2$, $i+4$, $i+7$ and $i+10$, then the distance sequence would be 2, 2, 3, 3.

2) Compute mean $\mu$ and standard deviation $\sigma$ for each sink's *distance* sequence and set them as its features. Thus, given the sink in the above example, it will have mean 2.5 and standard deviation 0.5. Intuitively, a sink belonging to a periodic class will have a smaller standard deviation than the one belonging to the nonperiodic class.

3) Group all sinks into the three desired classes using K-means algorithm in terms of their features. With the K-means algorithm which deals with unsupervised clustering, we can circumvent the trouble of determining thresholds. Furthermore, the K-means algorithm is a least-squares partitioning method that naturally divides a collection of objects into $K$ groups. Hence, it is more tolerant to "noisy" data as compared to other approaches.

Given a 2-speaker dialog, although typically we will have a series of alternating close-up shots of the two speakers, we can also have speakers in medium or long shots as well as shots with both speakers. Moreover, different camera angles will definitely produce different shots even for the same speaker. Therefore, to detect the dialog scene, if we use an approach that strictly requires every two shots be similar while adjacent shots be distinct like those reported in [4], [5], it will probably fail in certain scenarios. On the other hand, when the K-means algorithm is applied, we can somehow tolerate these "off-track" points.

Fig. 6(a) and (b) gives clustering results for two movies where both features ($\mu$ and $\sigma$) are used. As shown in these plots, all shot sinks have been well categorized into three groups, with the leftmost group belonging to the periodic class and the rightmost belonging to the nonperiodic class. Figs. 6(c) and (d) correspond to the results when only $\sigma$ is used for clustering. As we can see, all periodic sinks are clustering closely to the $x$-axis.

### C. Extracting and Classifying Events

Now, we are ready to organize classified shot sinks into events. The basic idea is to group all temporally overlapped sinks into one event. This is due to the fact that no shots that are semantically interrelated with each other will belong to different events, since different events have different thematic topics. Moreover, shots that do not belong to these sinks but are physically covered by their temporal ranges will also be included in the same event. For example, if shot $i$'s sink contains shots $i$, $i+2$, $i+4$ and $i+7$, and shot $i+1$'s sink contains shots $i+1$, $i+3$, and $i+8$, then they will be grouped into one event ranging from shot $i$ to shot $i+8$.

The event boundary is determined as follows. According to the event definition, every event contains an independent thematic topic, which means that, between two consecutive events there may exist some video passages that do not belong to any event. Usually, these are the so-called progressive scenes that consist of some sequential, nonrepetitive shots. Apparently, these scenes form natural gaps between unrelated shot sinks and can act as event delimiters. Although it is still possible that two events are closely developed one after another, it is less common since directors will usually need time to establish situations for the next event.

After extracting all events, we proceed to classify them into three classes, i.e. the 2-speaker dialog, the multiple-speaker dialog and the hybrid events, based on the following three heuristically derived rules:

1) If an event contains at least two periodic, at most one partly-periodic, and no nonperiodic shot sinks, it is declared as a 2-speaker dialog. This rule is quite intuitive since during a typical movie conversation, the camera will track the speakers back and forth, producing a series of alternating close-up shots of the two people.

2) If the event contains several partly-periodic sinks, or if the periodic and nonperiodic shot sinks coexist, we label it as

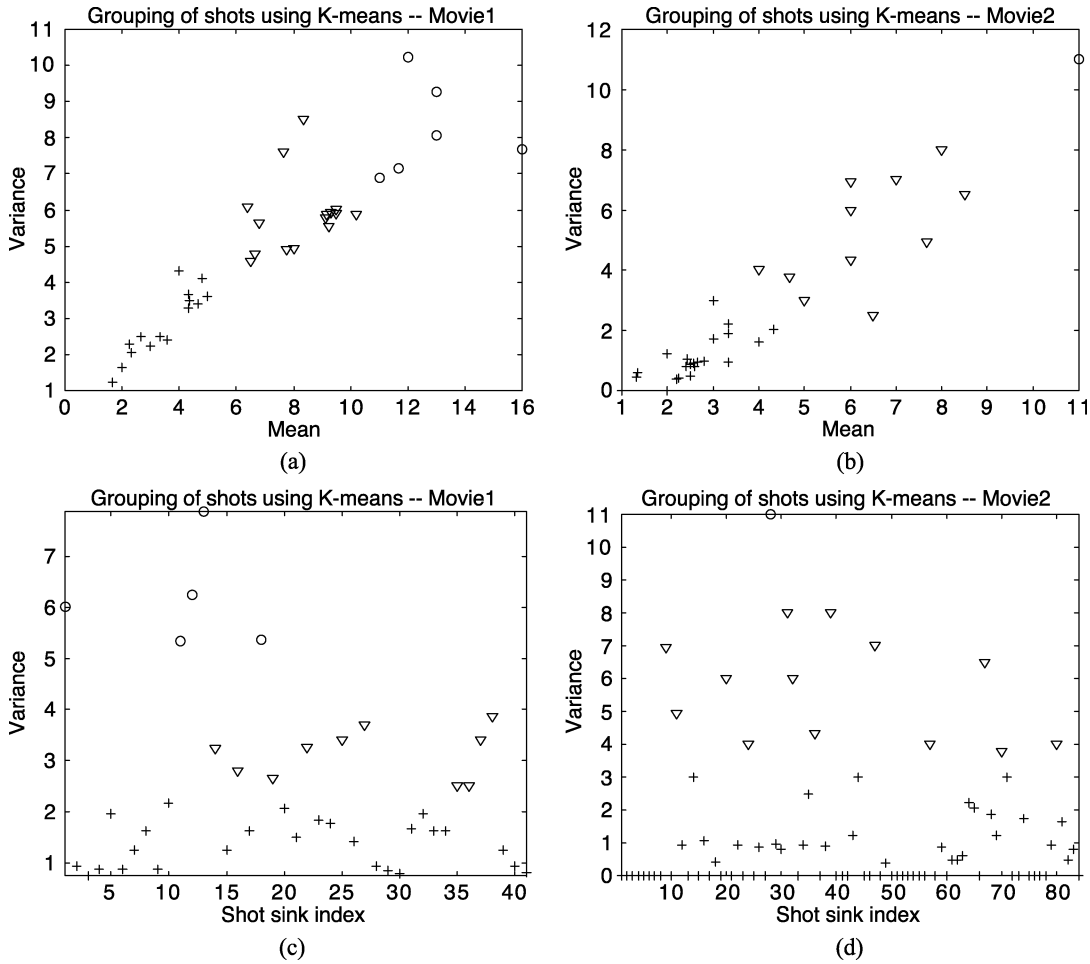



Fig. 6. Clustering shot sinks with the K-means algorithm. (a) Movie1 with both features used. (b) Movie2 with both features used. (c) Movie1 with only $\sigma$ used. (d) Movie2 with only $\sigma$ used. Crosses, triangles, and circles stand for sinks in the periodic, partly periodic and nonperiodic classes, respectively.

Fig. 7. Falsely detected 2-speaker dialog.


Fig. 8. Keyframes extracted from 2 neighboring shots in a falsely detected 2-speaker dialog, where the face detection result is superimposed with detected faces boxed by rectangles and eyes indicated by crosses.

a multiple-speaker dialog. The reason for tolerating nonperiodic sinks is that, during a multiple-speaker conversation, we have no way to predict who will be the next speaker since everyone has an equal opportunity to talk.

3) All remaining events are labeled as the hybrid type.

Finally, a post-processing step is carried out which aims to prescreen the obtained event results and correct some easily detected errors. Specifically, the following two features are computed and checked for each event: 1) *the event length*, we require that an event's length should be above a certain threshold which is set to be 15 s in our current work and 2) *the temporal variance*, which is computed as the average variance of the color histogram of all the shots within the event [3]. To some extent, this value indicates the amount of motions involved in an event. Because an ordinary dialog usually contains less motion, we require that its temporal variance is lower than a certain threshold, which is empirically set to be 80.

### *D. Integrating Speech and Face Information*

Due to the limitation of the pure color information, the following two major types of false alarms have been observed in our coarse-level event results.

*Type I: Misdetect a Conversation-Like Montage Presentation as a Spoken Dialog:* Fig. 7 gives one such example. This event describes a hunting scene where the camera shuttles back and forth between the hunter and the prey so as to generate a tense atmosphere. However, due to its periodic shot pattern, this event has been declared as a 2-speaker dialog, which is obviously wrong. In fact, such an event type is not unusual in feature films

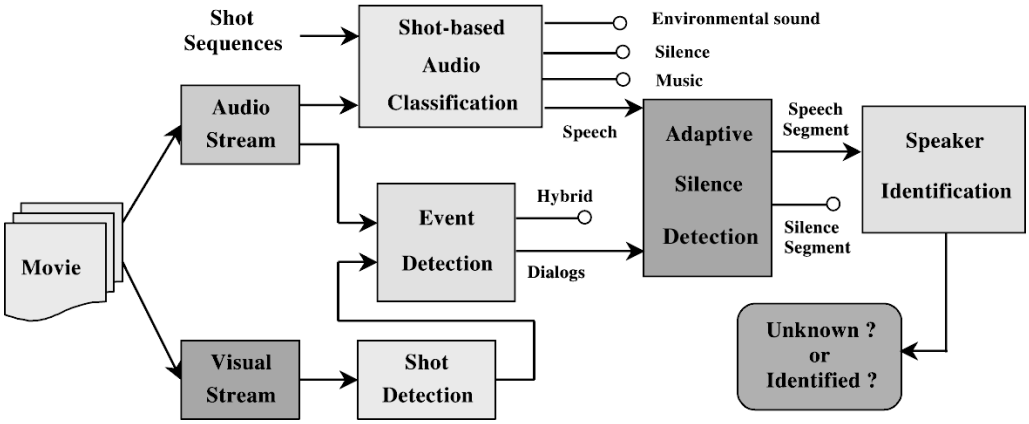



Fig. 9. Framework of the proposed speaker identification system.

and is called "thematic dialog" in [5]. Other similar scenarios include kissing or hugging scenes, where no actual conversation goes on between the two people in the scene.

*Type II: Misclassify a Multiple-Speaker Dialog as a 2-Speaker Dialog:* Fig. 8 shows two keyframes extracted from a falsely detected 2-speaker dialog, which actually belongs to a multiple-speaker dialog. This false alarm is caused by a detected repetitive shot pattern resulted from frequent camera switches between the two couples instead of among the individual speakers. Errors will also occur in the scenario where one person dominates the dialog while the rest of speakers talk less.

To reduce the false alarm of Type I, we integrate the embedded audio information into the detection scheme. Specifically, to be qualified as a spoken dialog, an event should contain a high speech ratio. Detailed processing steps are given as follows. First, we classify every shot in the candidate dialog into one of the four audio classes described in Section III. Then, we calculate the ratio of its contained speech shots. If the ratio is above a certain threshold, we confirm the event to be a dialog; otherwise, we label it as a hybrid event. Currently, we set this threshold to be 0.4, i.e. we require that at least 40% of the event shots contain speech.

To reduce the false alarm of Type II, we include the facial cue into the detection scheme. Specifically, for each shot in a 2-speaker dialog, we first perform a face detection on its underlying frames and output the average number of detected faces. Thus if there are $n$ shots in the dialog, we will get $n$ output values. Then, we check if more than half of these values are larger than one. If yes, we re-label this event as a multiple-speaker dialog. This is because that, a 2-speaker dialog should not have more than one face in most of its component shots if it presents a periodic shot repeat pattern. Some face detection examples are shown in Fig. 8, where detected faces are boxed by rectangles and eyes are indicated by crosses.

## V. Speaker Identification for Movie Dialogs

In this stage, target speakers engaged in movie dialogs will be identified by exploiting both audio and visual sources. Because there are generally tens of casts in a movie, we are dealing with an "open-set" identification problem. Here, we restrict the problem by only identifying a subset of casts.

Fig. 9 depicts the proposed system framework which includes the following four major modules: shot-based audio classification, event detection, adaptive silence detection, and speaker identification. As shown, for every speech shot in a detected movie dialog, the silence detection module will first extract speech segments from the background, then pass them onto the last module for the identification purpose.

The identification module functions as follows. Given an input speech signal, we first decompose it into a set of overlapped audio frames. Then features are extracted from each frame to form a feature vector $\vec{x}$. Next, we calculate the likelihood values $P_i(\vec{x}|M_i)$ between $\vec{x}$ and all pre-trained speaker models $M_i$ and subsequently normalize it against a background model. Finally, the total likelihoods over all speech frames with respect to each speaker model are summed up, and the speaker whose model produces the maximum value is claimed to be the target speaker.

### A. Feature Selection, Extraction, and Speaker Modeling

Although there are no exclusively speaker-distinguishing speech features, the speech spectrum has shown to be effective for speaker identification applications [39]. This is because that the spectrum reflects a person's vocal tract structure, which is the predominant physiological factor that distinguishes one person's voice from others. In this work, the cepstral coefficients derived from the Mel-frequency filterbank, i.e. the Mel-frequency cepstral coefficient (MFCC), is chosen to represent the short-time speech spectra due to its robustness to noisy signals.

The feature extraction proceeds as follows. Given a speech frame, we first remove its DC-mean and pre-emphasize it with an FIR filter $(H(z) = 1 - 0.97z^{-1})$. The purpose of this process is to spectrally flatten the speech signal and increase the relative energy of its high-frequency spectrum [40]. MFCC coefficients are then extracted from the frame's magnitude spectrum mapped with a simulated mel-scale filterbank [41]. Moreover, considering the various background sounds in movies, a cepstral mean normalization is further carried out on these features. However, while some previous work reported an improved system performance with adding time derivatives to the basic spectral features [39], our experiments have shown worse results in this case. This is probably due to the fact that

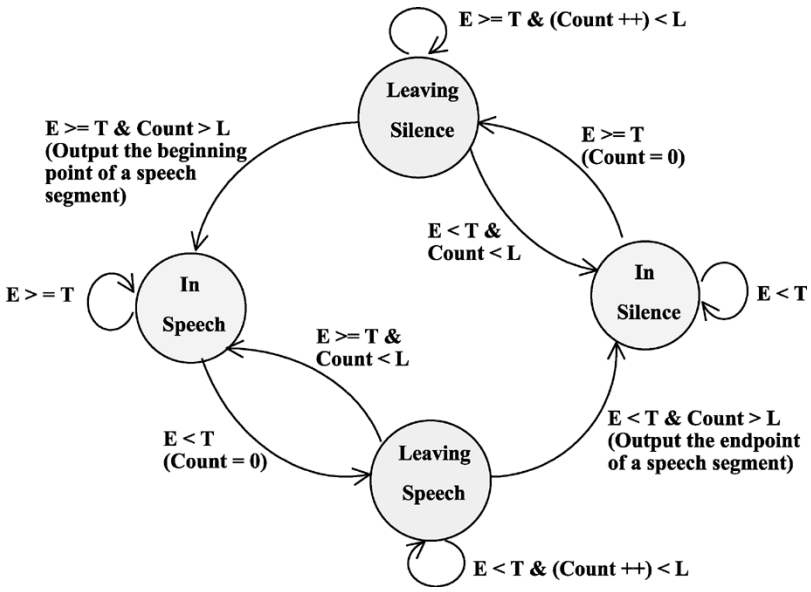



Fig. 10. State transition diagram for speech-silence segmentation where $T$ stands for the derived adaptive threshold, $E$ denotes the frame energy, *count* is a frame counter, and $L$ indicates the minimum speech/silence segment length.

casts' emotions, talking rates and voice volumes are frequently changing in movies.

GMMs are chosen to model speakers since the individual Gaussian component can reflect general speaker-dependent vocal tract configurations that are useful for speaker identity modeling [39]. GMM has been successfully applied in both speech and speaker recognition.

A Gaussian mixture density is a weighted sum of $m$ component densities given by

$$P(\vec{x}|M) = \sum_{i=1}^{m} p_i b_i(\vec{x}) \tag{3}$$

where $\vec{x}$ is a D-dimensional feature vector, $p_i$ and $b_i(\vec{x})$ are the $i$th component's weight and density, respectively. The mixture weights satisfy the constraint $\sum_{i=1}^{m} p_i = 1$.

### *B. Likelihood Calculation and Normalization*

Let $M_i$ be the GMM model corresponding to the $i$th enrolled speaker and let $X$ be the observation sequence consisting of $\tau$ frames $\vec{x}_t$, $t = 1, \ldots, \tau$. Assuming that all observation frames are independent, the average log likelihood between $X$ and $M_i$ can be computed as

$$P(X|M_i) = \frac{1}{\tau} \sum_{t=1}^{\tau} \log p(\vec{x}_t|M_i) \tag{4}$$

where $p(\vec{x}_t|M_i)$ is given in (3). The speaker whose model gives the maximum likelihood is claimed as the target speaker.

Likelihood normalization, while proved to be very necessary for speaker verification, is usually not needed in typical speaker identification systems, since decisions made based on the likelihood from a single utterance require no interutterance likelihood comparison [42]. In this work, however, we do need a normalization step since we are dealing with an "open-set" identification problem.

To accommodate for nontarget speakers in the movie, we have built a background model $M_b$, which is trained with 40-second speech data collected from various unregistered speakers. The normalized version of $p(\vec{x}_t|M_i)$ is given as

$$p_{\text{norm}}(\vec{x}_t|M_i) = \frac{p(\vec{x}_t|M_i)}{p(\vec{x}_t|M_b)}. \tag{5}$$

### *C. Adaptive Silence Detection Scheme*

As there are various background noises in feature films, the very first step toward a successful identification task is to isolate individual speech segments from the background. This is the so-called "speech–silence discrimination" problem. A classical approach to this problem relies on a global energy thresholding scheme [35], [43]. This simple scheme, however, does not work well with dynamic or complex audio content. More recent work in this area focuses on the end-of-utterance detection which mainly targets at real-time automatic speech recognition (ASR) under an adverse environment [44], [45]. These methods are also not applicable to our work since their ultimate goal is to collect every utterance, rather than excluding every silent period as in our case.

In this work, we propose to detect silence by adapting to the underlying dynamic audio content. Particularly, given the audio signal of one speech shot, we first sort all audio frames into an array based on their energies pre-computed in the decibel scale. Then, for all frames whose energy values are greater than a preset threshold $\text{engy}T$, we quantize them into $N$ bins where $\text{bin}_1$ has the lowest, and $\text{bin}_N$ has the highest average energy. Because we already know that both silence and speech signals are present, Thus, $bin_1$ and $\text{bin}_N$ must possess the lower and upper limits of the silence and speech energies, respectively. Therefore, we calculate the threshold $T$ that separates speech and silence as: $T = \text{ENGY}_{sl} + \alpha \cdot (\text{ENGY}_{sp} - \text{ENGY}_{sl})$, where $\text{ENGY}_{sl}$ and $\text{ENGY}_{sp}$ are the average energies in the first and last three bins, respectively. $\alpha$ is a weighting coefficient

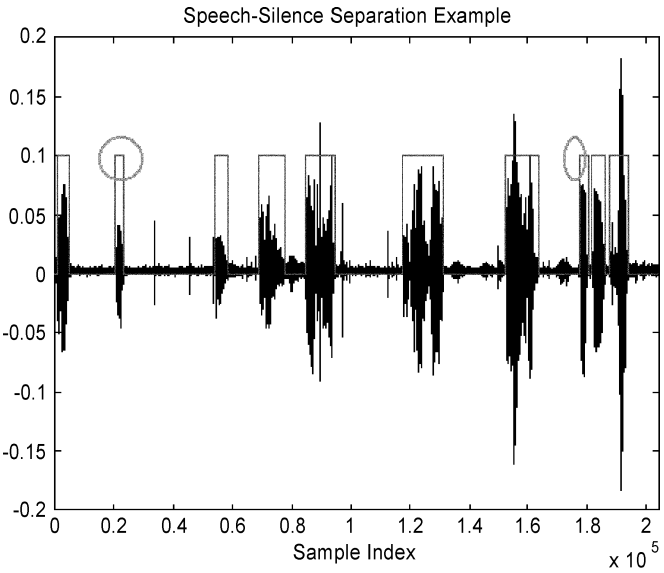



Fig. 11. Adaptive silence detection result on a clip taken from a speech shot, where detected speech segments are bounded by the passbands of superimposed pulse curves.

which equals 0.4 in the current work. Also, we set engy$T$ to be 30.0, and $N$ to be 10. As we can see, $T$ can always be adaptive to the background changes.

Next, we employ a 4-state transition diagram [44] to separate speech segments from the background as shown in Fig. 10. The input of this state machine is a sequence of frame energies, and the output is the beginning and ending frame indices of detected speech segments. The transition conditions between two states are labeled on each edge, and the corresponding actions are described in parentheses. In particular, *Count* is a frame counter, $E$ denotes the frame energy, and $L$ indicates the minimum length of a silence or speech segment which is set to be 300 ms in the current work. As we can see, this state machine basically groups blocks of continuous silence/speech frames as silence/speech segments while removing impulsive noises at the same time.

This algorithm works best when it is performed within a shot range since the background can be assumed to be quasistationary in this case. For the rest of this section, the two terms *silence* and *background noise* will be used interchangeably since they mean the same thing in this work.

Fig. 11 gives a speech-silence segmentation example on audio signals taken from a speech shot, where the $x$-axis shows the index of the audio samples obtained with 11-kHz sampling rate. A pulse curve is used to illustrate the results where detected speech segments are bounded by the passbands. Two of the detected speech fragments (indicated by the circles), are considered to be too short for the subsequent speaker identification. Overall, all speech fragments are precisely isolated from the background. Moreover, each of these fragments is guaranteed to be from one speaker, although a long sentence will probably be separated into several segments due to the intermittent short pauses.

As a comparison, we also attempted to detect the silence based on a global silence model, which is statistically trained using 40-s audio data collected from various kinds of background sounds. The detection results were not as satisfactory as the one shown in Fig. 11, which means that a global silence model can not catch the local background variations well.

TABLE I
EVENT DETECTION RESULTS FOR MOVIE1—TRAGIC ROMANCE

| Combining Speech & Face cues | | | | | |
|---|---|---|---|---|---|
| Event | Hits | Misses | F. alarm | Precision | Recall |
| Multiple-speaker | 4 | 0 | 0 | 100% | 100% |
| 2-speaker | 6 | 1 | 0 | 100% | 86% |
| Without Speech/Face cues | | | | | |
| Event | Hits | Misses | F. alarm | Precision | Recall |
| Multiple-speaker | 4 | 0 | 1 | 80% | 100% |
| 2-speaker | 6 | 1 | 3 | 67% | 86% |

TABLE II
EVENT DETECTION RESULTS FOR MOVIE2—COMEDIC DRAMA

| Combining Speech & Face cues | | | | | |
|---|---|---|---|---|---|
| Event | Hits | Misses | F. alarm | Precision | Recall |
| Multiple-speaker | 7 | 0 | 0 | 100% | 100% |
| 2-speaker | 14 | 0 | 0 | 100% | 100% |
| Without Speech/Face cues | | | | | |
| Event | Hits | Misses | F. alarm | Precision | Recall |
| Multiple-speaker | 5 | 2 | 0 | 100% | 72% |
| 2-speaker | 14 | 0 | 2 | 88% | 100% |

TABLE III
EVENT DETECTION RESULTS FOR MOVIE3—ACTION

| Combining Speech & Face cues | | | | | |
|---|---|---|---|---|---|
| Event | Hits | Misses | F. alarm | Precision | Recall |
| Multiple-speaker | 5 | 1 | 0 | 100% | 83% |
| 2-speaker | 13 | 0 | 0 | 100% | 100% |
| Without Speech/Face cues | | | | | |
| Event | Hits | Misses | F. alarm | Precision | Recall |
| Multiple-speaker | 5 | 1 | 0 | 100% | 83% |
| 2-speaker | 13 | 0 | 3 | 81% | 100% |

## VI. EXPERIMENTAL RESULTS

### A. Event Detection Results

For all the experiments reported in this section, video streams are compressed in MPEG-1 format with a frame rate of 29.97 frames/s. To validate the effectiveness of the proposed approach, representatives of various movie genres were tested. Specifically, the test set includes Movie1 ("*The Legend of the Fall*", a tragic romance), Movie2 ("*When Harry Met Sally*", a comedic drama) and Movie3 ("*Braveheart*", an action movie). Each movie clip is approximately one hour long.

TABLE IV
SPEAKER IDENTIFICATION RESULT OBTAINED FROM USING THE ADAPTIVE SILENCE DETECTOR

| | A | B | C | D | Unknown | FR | IA |
|---|---|---|---|---|---|---|---|
| A' | 37 | 1 | 2 | 2 | 3 | 18% | 82% |
| B' | 0 | 49 | 0 | 1 | 2 | 6% | 94% |
| C' | 3 | 2 | 81 | 3 | 11 | 19% | 81% |
| D' | 0 | 2 | 1 | 22 | 3 | 21% | 79% |
| Unknown | 6 | 15 | 9 | 6 | 166 | | |
| FA | 19% | 28% | 12% | 35% | | | |

Due to the inherent subjectivity of the event definition, we do not attempt to discuss the appropriateness of extracted events since people's opinions may differ. Instead, we will only examine the correctness of the event classification results, for which it is easier to reach a concensus. Experimental results are shown in Tables I–III for all three movies which contain 80 events in total. Each table is split into two parts, where Part 1 gives the results obtained by combining the speech and face cues while Part 2 gives the ones without the post-processing. Moreover, since the hybrid class contains all events excluding the dialogs, it is omitted from these tables. Precision and recall rates are computed to evaluate the system performance, where

$$\text{Precision} = \frac{\text{hits}}{\text{hits} + \text{false alarms}} \quad \text{Recall} = \frac{\text{hits}}{\text{hits} + \text{misses}}.$$

As shown in these tables, encouraging event extraction results have been achieved. When audio and facial cues are integrated, both precision and recall rates are higher than 83% in all three movies, which is a big improvement in system performance. Regarding the misses observed in these tables, the missed 2-speaker dialog in Movie1 was misclassified as a hybrid event, where one of the speakers was walking all time, which resulted in a frequent background change and therefore an irregular periodicity. In Movie3, a multiple-speaker dialog was misdetected due to the reason that people were talking in a too random fashion in that scene, thus an irregular shot repeat pattern has resulted.

### *B. Speaker-Identification Results*

To evaluate the robustness and effectiveness of the proposed speaker identification scheme, sophisticated studies were carried out on all three movies. Experimental results on the extended version of Movie1 (around two hours) are reported here.

Four key movie casts were chosen as target speakers, and for each speaker, we collected approximately 40-second data to train his/her GMM model. Currently, the training data were randomly collected across the entire movie sequence instead of coming from a particular training set. Each model was composed of 16 mixture components, and trained using the standard expectation maximization (EM) algorithm. Fourteen-dimensional MFCC coefficients were extracted from each frame. A background model was also built as discussed in Section V-B.

In total, 33 movie dialogs were detected, and the speaker identification result is tabulated in Table IV in the form of a confusion matrix. For simplicity, the four casts are indexed as A, B, C, D, and their respective movie characters are denoted by A′, B′, C′ and D′. "Unknown" is used for all other nontarget speakers. The value of each grid, say grid (A′, B), indicates the number of speech segments where character A′ is talking yet actor B is identified. Obviously, the larger the number in the diagonal, the better the performance. Three parameters, namely, *false acceptance* (FA), *false rejection* (FR) and *identification accuracy* (IA), are calculated to evaluate the system performance. They are defined as follows:

$$\text{FR} = \frac{\text{sum of off}-\text{diagonal numbers in the row}}{\text{sum of all numbers in the row}} \tag{6}$$

$$\text{FA} = \frac{\text{sum of off}-\text{diagonal numbers in the column}}{\text{sum of all numbers in the column}} \tag{7}$$

$$\text{IA} = 1 - \text{FR}. \tag{8}$$

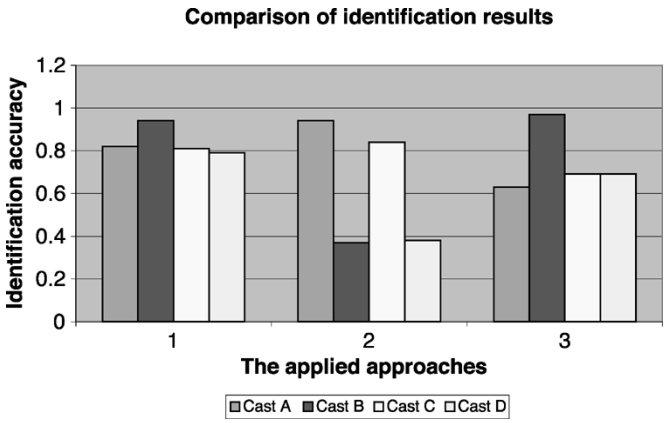



Fig. 12. Comparison of identification results obtained from: 1) the proposed adaptive silence detector; 2) the global silence model; and 3) the use of pure audio cue.

We observe from this table that there are certain cases where characters are mis-recognized as "unknown". This is mainly caused by the failure of identifying some very short speech segments. This table presents an average of 84% IA, which appears to be acceptable considering all kinds of changing factors in the movie. Also, the average FA and FR are as low as 23.5% and 16%, respectively.

Fig. 12 gives the comparison of identification results obtained using the proposed adaptive silence detector, the global silence model, and the pure audio cue, respectively. As we can see, the identification accuracy becomes very unstable and inconsistent when the global silence model is applied. A lot of false alarms occurs due to the incorrect isolation and imprecise boundary detection of the speech segments. Only an average of 63.25% IA is achieved, and the average FA has now risen to 31%.

By using the pure audio cue, we mean that no shot boundary information is used in the adaptive silence detection process. Thus in this case, the silence detector will be applied to the entire dialog instead of to every speech shot as in the first case. As shown in the figure, acceptable results have been achieved, yet some performance degradation is also observed. This is due to the fact that the proposed silence detector will no longer work well when the audio background becomes nonstationary.

To conclude, better identification results could be achieved when we isolate speech segments from the background by using

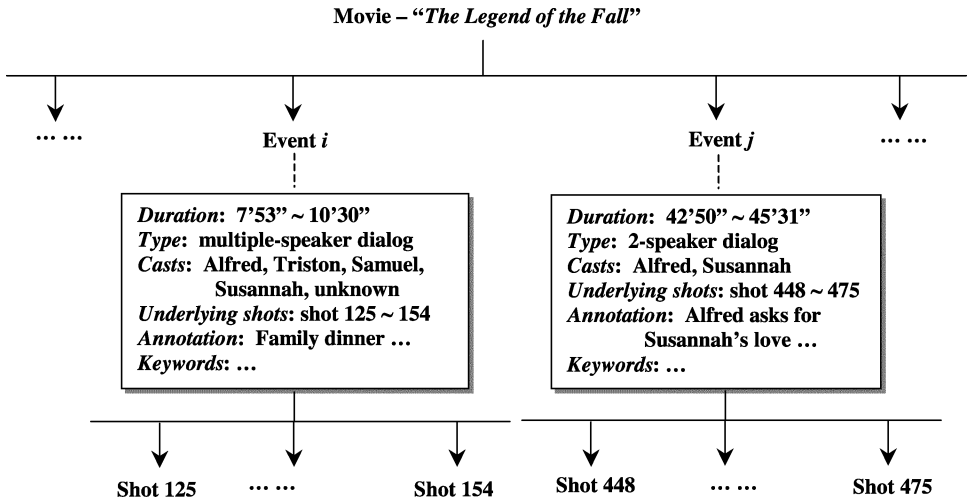



Fig. 13. Example of indexed movie content.

the proposed adaptive silence detector and by exploiting both audio and visual cues.

### C. An Example of Indexed Movie Content

After obtaining the event and speaker identity information, we can organize the movie content into a tree-like structure and index it in an efficient way. Fig. 13 gives such an example. As shown, the movie "*The Legend of the Fall*" is first decomposed into a series of events, where each event has certain properties such as the duration, the type, the present casts, the underlying shots, as well as some annotation or keywords if available. Then, for each event, it could be further decomposed into a series of shots with each shot being annotated in a similar way. As shown, with the aid of this structure, user's content access such as browsing and retrieval could be greatly facilitated.

## VII. Conclusion

An ideal content-based video analysis and indexing system should offer flexible and efficient tools for video browsing and retrieval. The design of such a system demands effective ways to extract semantic information from various media sources such as audio, visual and textual, such that multi-level video abstraction can be efficiently performed. This work presented a content-based movie analysis and indexing scheme which aims at extracting semantically meaningful movie events and identifying target speakers in movie dialogs. Although feature films have been our major focus, the methodology presented here could be easily extended to other types of generic videos. More robust event extraction results could be achieved by integrating other image/video processing techniques such as human/object tracking and face recognition. An adaptive audiovisual-based speaker identification scheme, which is still under development, will make this system more practical and robust.

## Acknowledgment

The authors would like to acknowledge the HP Labs, Palo Alto, CA, for providing the face-detection library.

## References

[1] B. Manjunath, P. Salembier, T. Sikora, and P. Salembier, *Introduction to MPEG 7: Multimedia Content Description Language*. New York: Wiley, June 2002.

[2] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu, "Video parsing, retrieval and browsing: An integrated and content-based solution," *ACM Multimedia*, pp. 15–24, Nov. 1995.

[3] D. Zhong, H. J. Zhang, and S. F. Chang, "Clustering methods for video browsing and annotation," in *Proc. SPIE*, vol. 2670, 1996, pp. 239–246.

[4] M. M. Yeung and B. L. Yeo, "Video content characterization and compaction for digital library applications," in *Proc. SPIE*, vol. 3022, Feb. 1997, pp. 45–58.

[5] H. Sundaram and S. F. Chang, "Determining computable scenes in films and their structures using audio-visual memory models," in *ACM Multimedia*, Marina Del Rey, CA, Nov. 2000, pp. 95–104.

[6] S. Tsekeridou and I. Pitas, "Content-based video parsing and indexing based on audio-visual interaction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 522–535, Apr. 2001.

[7] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Syst.*, vol. 1, no. 1, pp. 10–28, June 1993.

[8] T. Zhang and C.-C. J. Kuo, "Audio-guided audiovisual data segmentation, indexing and retrieval," in *Proc. SPIE*, vol. 3656, 1999, pp. 316–327.

[9] J. Nam and A. H. Tewfik, "Combined audio and visual streams analysis for video sequence segmentation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, 1997, pp. 2665–2668.

[10] R. Lienhart, "Comparison of automatic shot boundary detection algorithms," in *Proc. SPIE*, vol. 3656, Jan. 1999, pp. 290–301.

[11] J. Meng, Y. Juan, and S. F. Chang, "Scene change detection in an MPEG compressed video sequence," in *Proc. SPIE*, vol. 2419, Apr. 1995, pp. 14–25.

[12] B. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 533–544, Dec. 1995.

[13] Y. Rui, T. S. Huang, and S. Mehrotra, "Constructing table-of-content for video," *ACM Multimedia Systems*, vol. 7, no. 5, pp. 359–368, 1998.

[14] M. Yeung, B. Yeo, and B. Liu, "Extracting story units from long programs for video browsing and navigation," in *Proc. IEEE Multimedia Computing and Systems*, 1996, pp. 296–305.

[15] H. J. Zhang, S. Y. Tan, S. W. Smoliar, and G. Y. Hong, "Automatic parsing and indexing of news video," *Multimedia Syst.*, vol. 2, no. 6, pp. 256–266, 1995.

[16] I. Mani, D. House, D. Maybury, and M. Green, "Towards content-based browsing of broadcast news video," in *Intell. Multimedia Inform. Retrieval*, M. Maybury, Ed. Menlo Park, CA: AAAI Press/MIT Press, 1997.

[17] A. Merlino, D. Morey, and M. Maybury, "Broadcast news navigation using story segmentation," in *ACM Multimedia*, Seattle, WA, Nov. 1997, pp. 381–391.

[18] J. Huang, Z. Liu, and Y. Wang, "Integration of audio and visual information for content-based video segmentation," in *Proc. IEEE Int. Conf. Image Process.*, Chicago, IL, October 1998, pp. 526–529.

[19] A. G. Hauptmann and M. A. Smith, "Text, speech, and vision for video segmentation: the informedia project," in *Proc. AAAI Fall Symp. Computer Models for Integrating Language and Vision*, 1995, pp. 10–15.
[20] R. Lienhart, S. Pfeiffer, and W. Effelsberg, "Scene determination based on video and audio features," *Multimedia Tools and Applications*, vol. 15, no. 1, pp. 59–81, 2001.
[21] J. Nam, A. Cetin, and A. H. Tewfik, "Speaker identification and video analysis for hierarchical video shot classification," in *IEEE Int. Conf. Image Process.*, vol. 2, Oct. 1997, pp. 550–553.
[22] T. S. Mahmood and S. Srinivasan, "Detecting topical events in digital video," in *ACM Multimedia*, Marina Del Rey, CA, Nov. 2000, pp. 85–94.
[23] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," in *ACM Multimedia*, Marina Del Rey, CA, Oct. 2000, pp. 105–115.
[24] Y. L. Chang, W. Zeng, I. Kamel, and R. Alonso, "Integrated image and speech analysis for content-based video indexing," in *Proc. Multimedia*, Sept. 1996, pp. 306–313.
[25] G. Sudhir, J. C. M. Lee, and A. K. Jain, "Automatic classification of tennis video for high-level content-based retrieval," *IEEE Workshop Content-based Access of Image and Video Database*, pp. 81–90, Jan. 1998.
[26] D. D. Saur, Y. P. Tan, S. R. Kulkarni, and P. J. Ramadge, "Automated analysis and annotation of basketball video," in *Proc. SPIE*, vol. 3022, Jan. 1997, pp. 176–187.
[27] K. Reisz and G. Millar, *The Technique of Film Editing*. New York: Hastings, 1968.
[28] University of Pennsylvania. Linguistic Data Consortium. [Online]. Available: http://www.ldc.upenn.edu.
[29] I. M. Chagnolleau, A. E. Rosenberg, and S. Parthasarathy, "Detection of target speakers in audio databases," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Phoenix, AZ, 1999, pp. 821–824.
[30] S. E. Johnson, "Who spoke when?—Automatic segmentation and clustering for determining speaker turns," in *Eurospeech*, vol. 5, 1999, pp. 2221–2224.
[31] V. Wan and W. M. Campbell, "Support vector machines for speaker verification and identification," in *Proc. IEEE Signal Processing Society Workshop Neural Networks*, vol. 2, 2000, pp. 775–784.
[32] K. Mori and S. Nakagawa, "Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May 2001, pp. 413–416.
[33] D. Li, G. Wei, I. K. Sethi, and N. Dimitrova, "Person identification in TV programs," *J. Electron. Imaging*, vol. 10, no. 4, pp. 930–938, 2001.
[34] Y. Li and C.-C. Kuo, "Real-Time Segmentation and Annotation of MPEG Video Based on Multimodal Content Analysis I & II," Univ. Southern California, Los Angeles, Tech. Rep., 2000.
[35] T. Zhang and C.-C. Kuo, "Audio content analysis for on-line audiovisual data segmentation," *IEEE Trans. Speech Audio Processing*, vol. 9, pp. 441–457, Nov. 2001.
[36] Computational Video Group, HP Labs, "The HP face detection and recognition library," in *User's Guide and Reference Manual, Version 2.2*, Dec. 1998, .
[37] J. Monaco, *How to Read a Film: The Art, Technology, Language, History and Theory of Film and Media*. New York: Oxford Univ. Press, 1982.
[38] A. Tarkovsky, *Sculpting in Time—Reflections on the Cinema*. Austin, TX: University of Texas Press, 1986.
[39] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 72–83, Jan. 1995.
[40] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall., 1993.
[41] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. (2000, July) *HTK Book, Version 3.0* [Online] http://htk.eng.cam.ac.uk/docs/docs.html
[42] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Commun.*, vol. 17, no. 1–2, pp. 91–108, 1995.
[43] L. Rabiner and R. Schafer, *Digital Processing of Speech Signal*. Englewood Cliffs, NJ: Prentice-Hall , 1978.
[44] Q. Li, J. Zheng, Q. Zhou, and C. Lee, "A robust, real-time endpoint detector with energy normalization for ASR in adverse environments," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, May 2001, pp. 233–236.
[45] R. Hariharan, J. Hakkinen, and K. Laurila, "Robust end-of-utterance detection for real-time speech recognition applications," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, 2001, pp. 249–252.

**Ying Li** (M'00) received the B.S. and M.S. degrees in computer science and engineering from Wuhan University, Wuhan, China, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1993, 1996, and 2003, respectively.

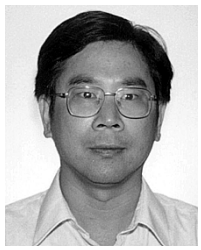
Since March 2003, she has been with IBM T. J. Watson Research Center as a Research Staff Member. Her research interests include digital image processing, content-based image analysis and retrieval, multimodal-based video content analysis, indexing and representation, computer vision and pattern recognition.

Dr. Li is a member of SPIE.

**Shrikanth Narayanan** (M'95–SM'02) received the M.S., Engineer, and the Ph.D. degrees in electrical engineering, from the University of California, Los Angeles, in 1990, 1992, and 1995, respectively.

From 1995 to 2000, he was with AT&T Labs—Research, Florham Park, NJ (formerly AT&T Bell Labs, Murray Hill, NJ) first as a Senior Member and later as a Principal Member of its Technical Staff. He is currently an Associate Professor at the Signal and Image Processing Institute of the University of Southern California's (USC), Los Angeles, electrical engineering department. He is also a Research Director for the Integrated Media Systems Center, an National Science Foundation Engineering Research Center and a Member of the faculty in Linguistics at USC. His research interests include signal processing and systems modeling with an emphasis on speech and language processing applications. His recent work has been in the areas of conversational human–computer interfaces, multimodal systems, automatic speech recognition algorithms, speech synthesis, and speech production modeling. He is an author or coauthor of more than 60 publications and holds three U.S. Patents.

Dr. Narayanan is an Associate Editor of the IEEE TRANSACTIONS OF SPEECH AND AUDIO PROCESSING (2000–present) and serves on the Speech Communication technical committee of the Acoustical Society of America. He is a member of Tau Beta Pi and Eta Kappa Nu.



**C.-C. Jay Kuo** (M'87–SM'92–F'99) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1980 and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, MA, in 1985 and 1987, respectively, all in electrical engineering.

He was the Computational and Applied Mathematics (CAM) Research Assistant Professor in the Department of Mathematics at the University of California, Los Angeles, from October 1987 to December 1988. Since January 1989, he has been with the Department of Electrical Engineering—Systems and the Signal and Image Processing Institute at the University of Southern California, Los Angeles, where he currently has a joint appointment as Professor of electrical engineering and mathematics. His research interests are in the areas of digital signal and image processing, audio and video coding, wavelet theory and applications, multimedia technologies, and Internet and wireless communications. He has authored more than 380 technical publications in international conferences and journals.

Dr. Kuo is a member of SIAM, ACM, and a fellow of SPIE. He is Editor-in-Chief for the *Journal of Visual Communication and Image Representation* and was the Editor-in-Chief for IEEE TRANSACTION ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY during 1995–1997. He received the National Science Foundation Young Investigator Award (NYI) and Presidential Faculty Fellow (PFF) Award in 1992 and 1993, respectively.