Automatic Movie Index Generation Based on Multimodal Information

Ying Li, Shrikanth Narayanan, Wei Ming and C.-C. Jay Kuo
Integrated Media Systems Center and Department of Electrical Engineering-Systems
University of Southern California, Los Angeles, CA 90089-2564
E-mail:{yingli, shri, ming, cckuo}@sipi.usc.edu

ABSTRACT

A fundamental task in video analysis is to organize and index multimedia data in a meaningful manner so as to facilitate user access for tasks such as browsing and retrieval. This paper addresses the problem of automatic index generation of movie databases based on audiovisual information. In particular, given a movie we first extract key movie events including two-speaker dialog scenes, multiple-speaker dialog scenes and hybrid scenes by using the proposed window-based sweep algorithm and the K-means clustering algorithm. Following event detection, the identity of each individual speaker in a dialog scene is recognized based on a statistical maximum likelihood approach. The identification relies on the likelihood ratio calculation between the incoming speech data and Gaussian mixture models of the speakers and the background. It is evident that the event and the speaker identity information will serve as a crucial part of the movie index table. Preliminary experimental results show that, by integrating multiple media information, we can obtain robust and meaningful event detection and speaker identification results.

1 INTRODUCTION

With the fast growing amount of multimedia information, content-based image/video indexing and retrieval have attracted increasing attention in recent days. However, most existing solutions are based on low-level features such as color, texture, shape, spatial relations, keyframes, temporal variance, camera and object motions. Although the extraction of above features is quite straightforward and relatively computationally simple, the corresponding query results are not always satisfactory due to the gap between low-level features and high-level semantics.

Recently, there has been some work on extraction of semantics from multimedia data. Mahmood and Srinivasan [1] presented a query-driven approach to detect topical events by using image and text content of query foils found in a lecture. While multiple media sources were integrated in their framework, identification results were mainly evaluated in the domain of classroom lectures and talks due to the special features used. In Rui and Yeung's work [2] [3], video scenes were constructed from a low-level shot sequence to serve as a video Table of Content (ToC). While scenes do capture more semantic meaning of the underlying video, not all constructed scenes contain meaningful themes due to their temporally sequential nature. Sundaram and Chang [4] reported their work on determining computable scenes in films by combining audio and visual information as well as detecting dialog scenes by using a periodic analysis transform. However, since the arrangement of shot sequences in a dialog basically varies with the film genre and depends heavily on the directorial styles, periodic analysis appears too restrictive for general scenarios. In addition, the problem becomes more complex when there are multiple speakers in a scene.

Compared to the large amount of work in the visual domain for video indexing, considerably less work has been reported in the audio domain. In [5], Tong and Kuo presented an approach for automatic segmentation, indexing and retrieval of audiovisual data based on audio content analysis. However, so far only results experimented on audio databases were reported. In [6], a system consisting shot detection, audio classification and speaker identification was proposed, but the simple audio classification scheme and the need of a large collection of training clips block the further improvement of the system performance. In [7], a content-based video parsing and indexing method was presented based on audio-visual interaction. While promising results have been reported on two TV-news programs, robustness of their algorithms has to be further evaluated on broader content domains.

In this work, we propose an automatic movie index generation scheme that produces two types of important indexing information: event and speaker identity. Specifically, given a movie, we first try to extract movie events by using a window-based sweep algorithm, where an event is typically a scene that contains a meaningful theme and is basically ongoing under a certain fixed environment. Particularly, all similar shots will be first pooled into shot sinks, and then each shot sink will be classified into one of three predefined classes, from which coarse-level events will be detected. A final post-processing procedure will be carried out by utilizing audio information. In the second stage, we proceed to identify speakers present in the extracted dialog scene since this information will be very useful in video browsing and retrieval. Gaussian Mixture Models (GMM) for each of the speakers of interest are trained beforehand, and the frame-based likelihood between the incoming speech data and stored GMM models are computed to help final decision-making for speaker identification.

Figure 1 shows the proposed three-level indexing structure with corresponding features and index contents. Arrows between nodes indicate a causal relationship. Since work involved in the first two modules was already reported in [5] and [8], this paper will mainly focus on generating semantic indexing information performed in the last module.

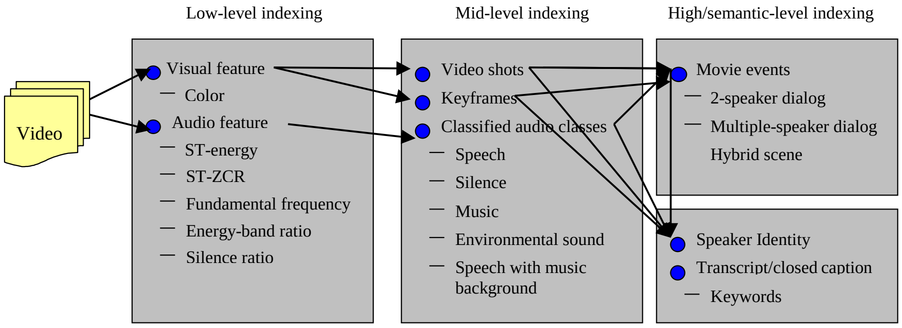


Figure 1: The proposed three-level indexing structure.

The rest of paper is organized as follows. In Section 2, we describe movie event detection algorithms including the generation of shot sinks, sink classification and audio scene detection. Section 3 elaborates on the speaker identification work where the GMM model is briefly introduced and an adaptive silence detection algorithm will be proposed. Preliminary experimental results are reported and discussed in Section 4, and finally concluding remarks and future directions are given in Section 5.

2 MOVIE EVENT EXTRACTION

Movie, known as a recording art, is practical, environmental, pictorial, dramatic, narrative and musical [9]. Since a film operates in limited time, all movie shots are efficiently organized by the film-maker in such a way that the audience will follow his/her own way of story-telling. Specifically, this goal is achieved by presenting the audience a sequence of cascaded events which gradually develop the movie plot. In this work, we consider the underlying event as the basic story unit of the movie.

Basically, an event is the scene which contains a meaningful theme and is usually going on under a certain consistent environment. However, to some extent, an event is more than a scene since it does not have the restriction of consistent chromaticity, lighting conditions and ambient sound as imposed on scene definition [4]. In other words, as long as scenes share the same theme, they belong to the same event.

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. Based on the film genre and film makers' directorial flavor, either (or both) of these two styles could be used frequently. However, no matter which filming style is used, they share one common feature. That is, 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. This is a very interesting observation, and the reason is perhaps due to the fact that since the drama of an event could only be developed within certain spatial and temporal localities, directors have to repeat some essential shots to convey parallelism and continuity of activities due to the sequential nature of film making. In the rest of this section, we will elaborate on the proposed event detection algorithm which is partially developed based on above observation.

2.1 Computing Shot Sinks with Visual Information

Since an event is generally characterized by a repetitive visual structure, our first step is to extract all scenes which possess this feature. In particular, we introduce a new concept called the *shot sink* in this work. A shot sink contains a pool of shots which are visually similar to each other but largely different from shots in other sinks.

2.1.1 Window-based Sweep Algorithm

The window-based sweep algorithm is used to compute the shot sink for each shot in a given shot sequence. Since any event occurs within certain temporal locality, we naturally restrict our search range for visually similar shots within a window of length N as shown in Figure 2(a), where the current window contains $n-i+1$ shots. Given shot i, we choose its keyframes to be its first and last frames and denote them b_i and e_i as shown in the same figure, then the similarity between shot i and j is defined as

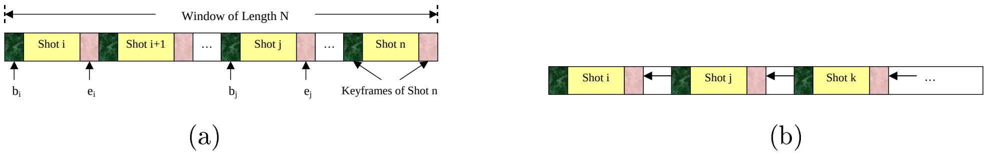
$$Dist_{i,j} = \frac{1}{4}(W_1 * dist(b_i, b_j) + W_2 * dist(b_i, e_j) + W_3 * dist(e_i, b_j) + W_4 * dist(e_i, e_j)),$$

where $dist(b_i, b_j)$ is the standard Euclidean distance between the two keyframes b_i and b_j in terms of their color histograms and W_1, W_2, W_3 and W_4 are four weighting coefficients computed as

$$W_1 = 1 - \frac{L_i}{N}, \quad W_2 = 1 - \frac{L_i + L_j}{N}, \quad W_3 = 1, \quad W_4 = 1 - \frac{L_j}{N},$$

where L_i and L_j are lengths of shots i and j in terms of frames. Here, we do not consider the absolute time separation between shots i and j (i.e. the temporal distance between e_i and b_j) in computing $Dist_{i,j}$. The reason is that since we want to find all similar shots (thus, the name "sweep"), we shall not weaken shots' similarity due to their physical separation as long as they are within the same timing window. However, we do consider the relative distance between each keyframe by introducing the shot length parameters L_i and L_j into the weights. It is actually intuitive that, if shots i and j are similar shots, frame e_i should be more similar to b_j than b_i to b_j due to motion continuity. Hence, we call $Dist_{i,j}$ a *time-adapted distance.*

If $Dist_{i,j}$ is less than a predefined threshold T, we consider them to be similar, and throw shot j into shot i's sink. As shown in Figure 2(b), all shot i's similar shots are neatly linked together based on their temporal order. We repeat this window-based sweep algorithm for every shot to compute their respective shot sink. However, if one shot has already been in other shot's sink, we will skip the current shot and continue with the next one.



(a)

(b)

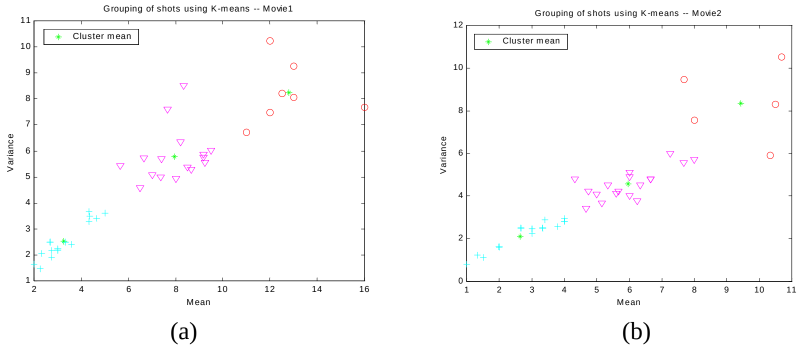
Figure 2: (a) Shots contained in a window of length N and (b) the computed sink of shot i.

2.2 Classifying Shot Sinks with K-means

The next step is to classify shot sinks into 3 predefined classes: periodic, partly-periodic, and random. For shots in the first class, we can observe a near perfect periodic pattern. For instance, shot i is repeated every 2 shots which shall be frequently encountered during a 2-speaker dialog scene. In the second class, a certain rough periodicity will be detected, but it is not always strictly observed. Examples include those shots involved in a dialog scene with multiple speakers who take turns to talk. For the last class, we call it random since no specific conclusion can be made based on shots' distribution. Note also that if shot i's sink contains only one item (i.e. the shot itself), we exclude this shot for further consideration.

To detect the periodicity for each shot sink, we first calculate the relative temporal distance between shot i and its peers. Then, the mean μ and standard deviation σ of these distances are computed and considered to be the sink's features. Intuitively, a shot belonging to a periodic class will generally have smaller statistics than the one belonging to the random class.

After obtaining the two features for every shot sink, we choose to cluster them by using the standard K-means algorithm. There are two reasons for using the K-means in this task. First, we can circumvent the trouble of determining a set of thresholds for the classification purpose. Second, the K-means algorithm is a least-squares partitioning method that naturally divides a collection of objects into K groups. Hence, the K-means algorithm is more tolerant to "noisy" data as compared to others using rigid rules. For example, given a 2-speaker dialog scene, although typically we have a series of alternating close-up shots of two players, we can also have characters in medium or long shots as well as shots with two speakers present. Moreover, different camera angles will definitely produce different shots even for the same speaker. Therefore, if we use an approach which strictly demands that every 2 shots should be similar while adjacent shots are different in a dialog scene [4], it will probably fail in some scenarios. However, if the K-means approach is applied, we may still get correct classification results. Figure 3 shows classification results for two movie clips. As one can see, all shot sinks have been well classified, where the leftmost group belongs to the periodic class and the rightmost one belongs to the random class.



(a)

(b)

Figure 3: Classifying shot sinks with the K-means algorithm: (a) Movie 1 with 43 shot sinks, and (b) Movie 2 with 51 shot sinks.

2.3 Extracting and Classifying Events

Now, we are ready to organize all classified shot sinks into events. The first problem to be solved is the determination of the event boundary. The solution actually resides in the definition of an event. Since each event has a certain thematic topic, events are not temporally sequential, which means that there usually exist some progressive scenes between two consecutive events without any repetitive structure. Therefore, in most cases, a natural gap between unrelated shot sinks will be observed, which can serve as the event delimiter. Of course, it is still possible that two events are tightly developed one after another. However, it is very rare since the director needs time to establish the situation for the next event. The second problem is to assign appropriate types to each assembled event. Three event classes are considered in this work: the 2-speaker dialog scene, the multiple-speaker dialog scene and the hybrid scene. A set of simple rules are introduced here to accomplish this task.

1. If the event contains at least two periodic shot sinks, at most one partly-periodic, and no random shot sinks, it will be declared as a 2-speaker dialog scene. This is actually quite straightforward since during a typical movie conversation scene, the camera will track the speakers back and forth, producing a series of alternating close-up shots of the two players.
2. If the event contains several partly-periodic shot sinks, or if the periodic and random shot sinks coexist, we label it with a multiple-speaker dialog scene. Since we may have many speakers, and every speaker has an equal probability of talking, we may have random shot sinks in this case.
3. All remaining events are labelled as the hybrid scene.

2.4 Integrating Audio Information

Since coarse-level events are detected based on pure visual information, false alarms will occur in some cases. For example, in one of our test movies, there is an event describing a hunter and his prey. The camera shuttles back and forth between them to generate a tense atmosphere. Naturally, this event is declared as a 2-speaker dialog scene since strong periodicity is detected. This type of events is actually not unusual in our daily movies and is called the "thematic dialog" by Sundaram [4]. Other similar cases could be found in events where two people are kissing, hugging, etc. To avoid this type of false alarms, we can integrate audio information into our detection scheme. The rationale here is quite intuitive, that is, if an event is declared as a dialog scene, it should have a higher ratio of speech content.

For every shot in a dialog event, we will classify it into one of 4 audio classes: speech, music, silence and environmental sound based on the technique given in [5]. If it is a speech with music background shot, we still declare it a speech shot since speech is of more importance to us. An event can only be declared as a dialog event when its speech ratio is higher than a certain threshold. Otherwise, it is only a thematic scene.

3 SPEAKER IDENTIFICATION FOR MOVIE DIALOG SCENE

The next step in the proposed index generation scheme is to identify all speakers present in extracted dialog scenes. Speaker identification has been an active research topic for many years. However, most of them were carried out based on standard speech databases such as YOHO, HUB4, and SWITCHBOARD [10], [11], [12]. Although acceptable results have been reported, the identification error tends to be serious when we do not know "who speaks when". The frame-based speaker segmentation and clustering approach can easily introduce errors if only acoustic similarity is considered. We believe that, given a multimedia data stream such as movie, the knowledge from all available media sources such as video and audio should be effectively integrated to achieve robust identification results. In this work, we use shot boundary detection and shot-based audio classification results to aid the speaker identification process.

Figure 4(a) shows a detailed blockdiagram of the speaker identification module using a statistical maximum likelihood approach. The input of this scheme is a speech signal that is viewed as a sequence of overlapping short-term segments called frames. The front-end analysis module then extracts audio features from each frame and forms the feature vector $\vec{X} = \{x_1, x_2, ..., x_m\}$. The likelihood values $P_i(\vec{X}|M_i)$ between this vector $\vec{X}$ and each of the pre-trained speaker models M_i will then be calculated and normalized by using a background model. Finally, the total likelihood over all speech frames with respect to each model will be summed, and the speaker whose model produces the maximum likelihood will be claimed as the identified speaker.

3.1 Feature Selection and Extraction

Although there are no exclusively speaker distinguishing speech features, the speech spectrum has been shown to be very effective for speaker identification applications [13]. This is because the spectrum reflects a person's

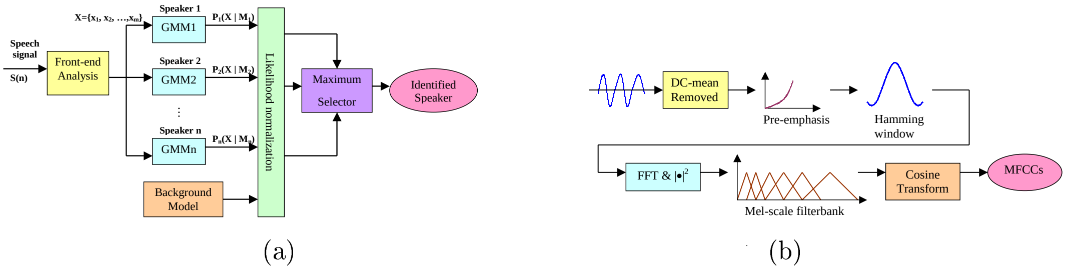


Figure 4: (a) The blockdiagram of the speaker identification module, and (b) the blockdiagram of the front-end analysis module.

vocal tract structure, the predominant physiological factor that distinguishes one person's voice from others. LPC spectral representations, such as LPC cepstral and reflection coefficients, have been used extensively for speaker recognition; however, these model-based representations can be severely affected by ambient noise as pointed out in [14]. In this work, we use cepstral coefficients derived from the mel-frequency filterbank to represent the short-time speech spectra.

The incoming speech frame will be initially DC mean-removed and pre-emphasized by an FIR filter with transfer function $H(Z) = 1 - 0.97Z^{-1}$. The main purpose of this pre-processing is to flatten the speech signal spectrally, thus increasing the relative energy of its high-frequency spectrum [15]. The frame, which proceeds with an overlap of the previous one, is then windowed by a Hamming window so as to minimize the signal discontinuity at its beginning and end. To extract the Mel-Frequency Cepstral Coefficients (MFCCs), each frame's magnitude spectrum will be first processed by a simulated mel-scale filterbank. Then, the log-energy filter outputs will be consine transformed to produce the MFCC coefficients [16]. Figure 4(b) gives the blockdiagram of steps used in the front-end feature extraction.

In addition, we also perform a cepstral mean normalization process on all cepstral coefficients with an expectation that it can reduce effects caused by background sounds, which constantly exist in a movie. Some previous work has also reported that the system performance could be greatly enhanced by adding time derivatives to the basic spectral features [13]. Our experiments however indicated that the inclusion of delta coefficients only brought worse results. This was probably due to the reason that cast's emotions, speech rate, speech level and recording conditions are constantly changing.

3.2 Gaussian Mixture Model (GMM)

The Hidden Markov Model (HMM) as a probabilistic model has been widely used in many speech applications such as speech and speaker recognition. The advantage of HMM is that it models not only the underlying speech sounds, but also temporal sequencing among these sounds. However, although the temporal structure modeling is advantageous for text-dependent tasks, the sequence of sounds found in training data does not necessarily reflect sound sequences found in test data and contains little speaker-dependent information for text-independent tasks as reported by Tishby [17]. In this work, we will employ the Gaussian Mixture Model (GMM) to model speakers with two reasons: first, the individual Gaussian component in a speaker-dependent GMM are interpreted to represent some broad acoustic classes which reflect some general speaker-dependent vocal tract configurations that are useful for modeling speaker identity; second, a Gaussian mixture density is shown to provide a smooth approximation to the underlying long-term sample distribution of observations obtained from utterances by a given speaker [13].

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

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

where $\vec{x}$ is a D-dimensional feature vector, p_i is the i^{th} component weight and $b_i(\vec{x})$, the i^{th} component density, is computed as

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}}\exp\{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)'\Sigma_i^{-1}(\vec{x}-\vec{\mu}_i)\},$$

with mean vector $\vec{\mu}_i$ and covariance matrix Σ_i. The mixture weights satisfy the constraint $\sum_{i=1}^{m} p_i = 1$.

For the rest of this paper, each speaker will be represented by a GMM model by using the notation

$$M = \{p_j, \vec{\mu}_j, \Sigma_j\} \qquad j = 1, \ldots, m.$$

3.3 Likelihood Calculation and Score Normalization

Let M_i be the GMM model corresponding to the i^{th} enrolled speaker and represented by $M_i = \{p_{ij}, \vec{\mu}_{ij}, \Sigma_{ij}\}$ where $j = 1, \ldots, m$, and let X be the observation sequence consisting of T frames $\vec{x}_t$ $t = 1, \ldots, T$. Under the assumption that all observation frames are independent, the average log likelihood between X and M_i can be computed as

$$P(X|M_i) = \frac{1}{T}\sum_{t=1}^{T}\text{log}p(\vec{x}_t|M_i),$$

where $p(\vec{x}_t|M_i)$ is given in (1). Finally, the unknown speaker will be identified from a set of N speakers by choosing the one having the maximum likelihood. That is,

$$\hat{S} = \text{arg max}_{1\leq i\leq N}P(X|M_i).$$

Now since we are dealing with movies and a movie generally contains multiple cast members, it is impractical and actually may be unnecessary to identify every speaker encountered. Very often people are only interested in leading actors or actresses. Therefore in this case, we will have an open-set identification task, where not all speakers are known to our database. Building a background model thus becomes a necessity if we do not want to misclassify the unknown speaker to one of known speakers simply because this speaker gives the minimum distances among all others. In this work, the background model M_b is trained by using speech segments from some other supporting actors/actresses as well as background sounds. The normalized likelihood could be computed as

$$p_{norm}(\vec{x}_t|M_i) = \frac{p(\vec{x}_t|M_i)}{p(\vec{x}_t|M_b)},$$

and consequently,

$$\text{log}p_{norm}(\vec{x}_t|M_i) = \text{log}p(\vec{x}_t|M_i) - \text{log}p(\vec{x}_t|M_b).$$

Now, given an incoming observance sequence X, we can obtain a new normalized score as

$$Sc_i(X|M_i) = \frac{1}{T}\sum_{t=1}^{T}\text{log}p_{norm}(\vec{x}_t|M_i).$$

In this case, the speaker to be chosen will simply depend on which speaker model produces the highest score in $Sc_i(X|M_i)$.

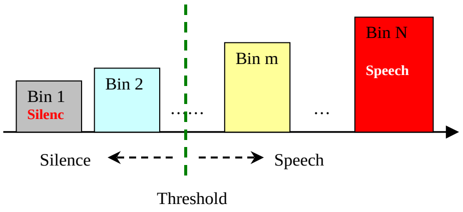
3.4 Adaptive Silence Detection Scheme

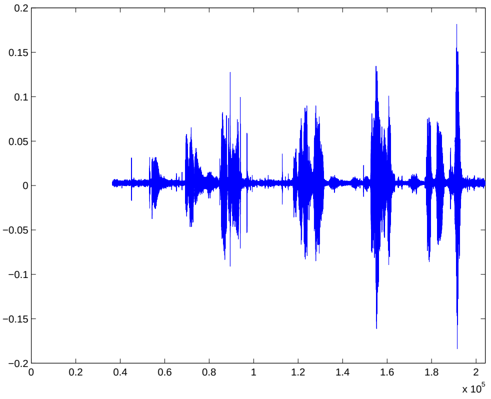
Given audio data corresponding to a speech shot within a dialog scene, the proposed speaker identification system will either identify it to be a known speaker or claim it to be an unknown. This is a reasonable approach since the camera usually focuses on the talking person during the dialog scene. But, is this always true? How about when there are multiple speakers talking within one shot? It turns out that we do have to deal with cases where multiple people take turns to speak within one single shot.

This is the so-called “who speaks when” problem, which has actually attracted many researchers’ interests recently. The approach adopted in most published work in the literature has relied on applying speaker segmentation and clustering techniques, and it tends to introduce errors since all processing is carried out at a low syntactic level [11], [12]. In this work, we propose an *adaptive silence detection algorithm* that performs at a higher semantic level, thus producing better results. In particular, this approach attempts to partition incoming audio signals into separate speech and silence subsegments, where only speech subsegments will be subsequently fed into the identification module. The assumption used here is that a speech segment bounded by two silence segments should be attributed by one single speaker. This is a reasonable assumption, and has been well supported by all data we have considered.

Since input signals from a speech shot are known to contain both speech and silence signals, it will somehow ease our task in the silence detection. Although background audio levels and speech levels are constantly changing with the time, they typically maintain quasi-stationarity within one single shot. We exploited this observation to develop an *adaptive silence detection algorithm* as detailed below.

Given an input audio signal containing N frames, we first sort the frames into an array based on their energies precomputed in the dB scale. Then, for all frames whose energy values are greater than a preset threshold $Discard\_thresh$, we quantize them into $NumBin$ bins where bin_1 has the lowest average energy and bin_{NumBin} has the highest average energy. Since we already know that both silence and speech signals are present, obviously bin_1 gives the lower boundary of the silence energy, and bin_{NumBin} has the upper limit of the speech energy. Thus, the threshold separating the speech from silence should be a value between these two extremes as shown in Figure 5(a). In this work, the threshold value is adjusted based on the sum of the first 3 bins and the sum of the last 3 bins as well as a predefined $Speech\_thresh$ that gives the minimum dB difference between the two signals.





where the speech subsegments correspond to the peaks. Two detected speech fragments, as indicated in the figure, are considered to be too short for further processing. As we can see, all speech fragments have been successfully detected with some impulse noises or loud background sounds removed. These detected fragments are guaranteed to be spoken by one single speaker, although several sentences continuously spoken by a speaker will probably be separated into several segments due to intermittent silence.

In the next step, all qualified speech subsegments will be used as the input into the speaker identification module to make the identity decision as described earlier.

4 EXPERIMENTAL RESULTS

For all experiments reported in this section, video streams are in MPEG-1 compressed format with a frame rate equal to 29.97 frames/sec. To validate the effectiveness of the proposed approach, representatives of various movie genres were tested. Specifically, the test set included Movie1 (romance), Movie2 (adventure), and Movie3 (action). Each movie clip is approximately 1 hour long, and the total length is about 313,200 frames.

For the movie event detection part, due to the inherent subjectivity of the event definition, we will not attempt to discuss the appropriateness of extracted events since people's opinions may differ. Instead, we will only examine the correctness of classified event classes, on which it is easier to reach a common conclusion. Experimental results are shown in Table 1 for all three movies. There are two parts in each sub-table: the results obtained by combining audio cues and the one without audio. The precision and recall rates are computed to evaluate the performance where Precision = hits/(hits + false alarms) and Recall = hits/(hits + misses). In addition, because the hybrid event class contains the rest of events excluding the multi-speaker and the 2-speaker events, its extraction results are omitted from the table. Followings are some observations of the obtained results.

- The event detection results are quite encouraging. All precision and recall ratios are higher than 83% when the audio information is integrated. Also, it is evident that by integrating the audio information into the proposed detection scheme, we have observed distinct improvements in the precision ratio for all cases.
- The missed 2-speaker scene in Movie1 was misclassified as a hybrid scene. In this scene, the two speakers are quite far apart at the beginning of the talk, then one of them walks towards the other, which causes the change of the background and results in the irregularity of periodicity. In Movie2, one of the multiple-speaker scenes is misclassified as a 2-speaker scene due to the fact that one of the speakers dominates the dialog. A similar case is also found in Movie3.

To evaluate the robustness and effectiveness of the proposed speaker identification scheme, study was carried out by using the longer version of Movie1 "Legend of the Fall", which is around 2 hours. Totally 4 speaker models were built corresponding to 4 key actors/actresses by using approximately 40 seconds training speech data per speaker with silence removed. Their GMM models were trained using the standard Expectation Maximization (EM) algorithm [18] and each model consists of 16 mixture components with a diagonal covariance matrix. For each frame, 14-dimensional MFCC coefficients were extracted as audio features. A background model was also built as discussed in last section.

Totally 12 2-speaker dialog scenes were extracted, and the speaker identification results for these scenes are reported in the form of a confusion matrix as shown in Table 2. Each row and each column of the matrix correspond to a key movie character along with his/her name. The "unknown" tag stands for any other actors or characters, since they are deemed to be of no interest for speaker identification.

The left part of Table 2 gives the identification result when no adaptive silence detection is applied to the incoming audio signals. Instead, a simple silence detection scheme based on a fixed threshold is applied, where once the frame energy is less than the threshold it is declared as a silence frame and will be discarded. Moreover, all speech frames from the same shot are considered to be contributed by one single speaker in this case. During the collection of statistics, if there are two speakers within the shot, say, speaker A and B, and suppose A is the

Table 1: Event detection results for Movie1, Movie2 and Movie3

Movie1 – Tragic Romance										
	Combining Audio Information					Without Audio Information				
Event Class	Hit	Miss	False	Precision	Recall	Hit	Miss	False	Precision	Recall
Multi-speaker dialog	4	0	0	100%	100%	5	0	1	100%	83%
2-speaker dialog	6	1	0	86%	100%	9	1	3	90%	75%
Movie2 – Adventure										
	Combining Audio Information					Without Audio Information				
Event Class	Hit	Miss	False	Prec.	Recall	Hit	Miss	False	Prec.	Recall
Multi-speaker dialog	5	1	0	100%	83%	5	1	0	100%	83%
2-speaker dialog	14	0	1	93%	100%	16	0	3	84%	100%
Movie3 – Action										
	Combining Audio Information					Without Audio Information				
Event Class	Hit	Miss	False	Prec.	Recall	Hit	Miss	False	Prec.	Recall
Multi-speaker dialog	7	1	0	100%	88%	9	1	2	81%	90%
2-speaker	15	0	0	100%	100%	17	0	2	89%	100%

Table 2: Identification results with and without adaptive silence detection.

	Without adaptive silence detector					With adaptive silence detector				
Class Labels	Brad Pitt	Julia Ormond	Henry Thomas	Aidan Quinn	Unknown	Brad Pitt	Julia Ormond	Henry Thomas	Aidan Quinn	Unknown
Triston	11	0	3	2	11	18	1	1	0	2
Susan	0	20	0	0	3	0	27	1	0	2
Samuel	1	1	4	0	2	0	1	9	0	2
Alfred	1	1	1	17	6	1	1	1	38	3
Unknown	0	2	2	1	32	0	3	2	1	40

claimed speaker, then we say a) A is correctly identified as A, and b) B is misclassified as A. By observing the table, we see that a couple of movie characters tend to be misdetected as the "unknown" speaker. This false detection usually occurs when the shot contains multiple speakers, where the incoming speech data are actually contributed by several speakers instead of one single speaker. In this case, since no single speaker can be properly claimed as the target speaker, identifying it as "unknown" is perhaps the best solution we could offer. This sub-table presents an average 60.7% identification accuracy, which is not satisfactory.

The right part of Table 2 gives the experimental results with the adaptive silence detection applied. As shown, the numbers on the diagonal have significantly increased since all speakers who talk in the shot will be correctly identified. Even here we still have some cases where characters are misrecognized as "unknown", but these are usually due to the difficulty of identifying too-short speech segments. This sub-table presents an average of 83.3% identification accuracy, which is encouraging. In fact, if we can have more data for the third character (Samuel), we could further improve the classification results.

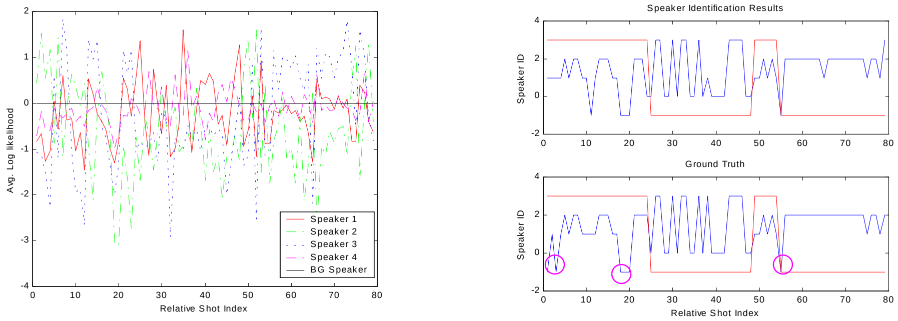
Our another experiment was designed to show that if only the pure audio information is used, we will not be able to get results as good as the right one shown in Table 2. In this experiment, when given a dialog scene, we apply the adaptive silence detection to the entire scene and each detected speech segment will be subsequently input into the identification module. Table 3 shows the results.

Table 3: Identification result with pure audio cues.

	Brad	Julia	Henry	Aidan	Unknown
Triston	17	3	1	1	5
Susan	0	38	0	0	1
Samuel	0	1	9	0	3
Alfred	1	4	6	35	5
Unknown	0	13	4	0	42

An overall accuracy of 74.5% is obtained in the table. The main reason for this performance degradation is that the proposed adaptive silence detection approach is not always correct since it is carried out at the scene range instead of the shot range. Because a movie scene usually covers tens of shots, we can no longer assume a constant background noise level. By carefully observing obtained experimental results, we find that things tend to get worse when there is loud background music/noise in certain shots, which will be generally classified into speech group by our silence detector due to their high energy.

Figure 6 shows an identification example, where four extracted 2-speaker dialog scenes are manually concatenated together to demonstrate the result. The curves in Figure 6(a) gives the computed average log likelihood of the incoming speech data with respect to each model, and the one with the largest likelihood will be identified as the target speaker. Since we applied a score normalization process, the background model produces a straight line at zero. Figure 6(b) shows the identification result as well as the ground truth, where -1 is the speaker ID of the unknown and 0-3, for the four speakers of interest. One thing worth pointing out here is that if there are multiple speakers recognized within one shot, for simplicity we will split the shot into multiple shots with one speaker per shot. The red pulse curves are simply used to delimiter the 4 dialog scenes. As shown, the blue ID curves present nice fluctuations between two speakers illustrating the turn-taking behavior. These experimental results coincide with the ground truth very well. Furthermore, although there are unknown speakers present in the beginning and end of the first and last dialog scenes as indicated in the figure, the 2-speaker rule is strictly observed for all other scenes. Therefore, this finding can also be used to aid in our dialog detection algorithm.



(a)

(b)

Figure 6: A speaker identification example.

5 CONCLUSION AND FUTURE WORK

A novel framework of automatic movie index generation based on the multimodal information was presented in this paper. Two major modules of this work, i.e. movie event extraction and speaker identification for dialog scenes, were described. Preliminary experiments have yielded encouraging results. In our future work, we plan to perform more extensive experiments on broader content domains and explore the feasibility of building and updating speaker models on the fly as more speech data comes in, under an unsupervised condition. Finally, we will look for ways to further improve our system so that multiple modalities including visual, audio and text can be better integrated.

6 REFERENCES

[1] T. S. Mahmood and S. Srinivasan, "Detecting topical events in digital video," *Proc. of ACM Multimedia 2000*, pp. 85–94, Marina Del Rey, November 2000.

[2] Y. Rui, T. S. Huang, and S. Mehrotra, "Constructing table-of-content for video," *ACM Journal of Multimedia Systems*, 1998.

[3] M. Yeung, B.-L. Yeo, and B. Liu, "Extracting story units from long programs for video browsing and navigation," *IEEE Proceedings of Multimedia*, pp. 296–305, 1996.

[4] H. Sundaram and S. F. Chang, "Determining computable scenes in films and their structures using audio-visual memory models," *Proc. of ACM Multimedia 2000*, Marina Del Rey, November 2000.

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

[6] N. V. Patel and I. K. Sethi, "Video classification using speaker identification," *Proc. of SPIE*, vol. 3022, pp. 218–225, 1997.

[7] S. Tsekeridou and I. Pitas, "Content-based video parsing and indexing based on audio-visual interaction," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 4, pp. 522–535, 2001.

[8] Y. Li and C.-C. J. Kuo, "Real-time segmentation and annotation of MPEG video based on multimodal content analysis I & II," *Technical Report, University of Southern California*, 2000.

[9] J. Monaco, *How to read a film: the art, technology, language, history and theory of film and media.* New York: Oxford University Press, 1982.

[10] A. E. Rosenberg, I. M. Chagnolleau, S. Parthasarathy, and Q. Huang, "Speaker detection in broadcast speech databases," *ICSLP'98, Sydney, Australia*, pp. 1339–1342, 1998.

[11] G. Yu and H. Gish, "Identification of speakers engaged in dialog," *ICASSP 93*, pp. 383–386, 1993.

[12] I. M. Chagnolleau, A. E. Rosenberg, and S. Parthasarathy, "Detection of target speakers in audio databases," *ICASSP'99, Phoenix, Arizona*, 1999.

[13] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian Mixture speaker models," *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[14] J. Tierney, "A study of LPC analysis of speech in additive noise," *IEEE Trans. on Acoustic, Speech, and Signal Processing*, vol. ASSP-28, pp. 389–397, 1980.

[15] L. Rabiner and B. H. Juang, *Fundamentals of speech recognition.* Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1993.

[16] S. Yound, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "HTK Book, Version 3.0," *Downloaded from http://htk.eng.cam.ac.uk/index.shtml*, July 2000.

[17] N. Z. Tishby, "On the application of mixture AR hidden Markov models to text independent speaker recognition," *IEEE Trans. on Signal Processing*, vol. 39, pp. 563–570, 1991.

[18] F. Jelinek, *Statistical methods for speech recognition.* Cambridge, Massachusetts, London, England: The MIT Press, January, 1999.