Hindawi Publishing Corporation
EURASIP Journal on Applied Signal Processing
Volume 2006, Article ID 27390, Pages 1–12
DOI 10.1155/ASP/2006/27390

# Semantic Context Detection Using Audio Event Fusion: Camera-Ready Version

**Wei-Ta Chu,[1] Wen-Huang Cheng,[2] and Ja-Ling Wu[1, 2]**

[1] *Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan*
[2] *Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 106, Taiwan*

Received 31 August 2004; Revised 20 February 2005; Accepted 5 April 2005

Semantic-level content analysis is a crucial issue in achieving efficient content retrieval and management. We propose a hierarchical approach that models audio events over a time series in order to accomplish semantic context detection. Two levels of modeling, audio event and semantic context modeling, are devised to bridge the gap between physical audio features and semantic concepts. In this work, hidden Markov models (HMMs) are used to model four representative audio events, that is, gunshot, explosion, engine, and car braking, in action movies. At the semantic context level, generative (ergodic hidden Markov model) and discriminative (support vector machine (SVM)) approaches are investigated to fuse the characteristics and correlations among audio events, which provide cues for detecting gunplay and car-chasing scenes. The experimental results demonstrate the effectiveness of the proposed approaches and provide a preliminary framework for information mining by using audio characteristics.



## 1. INTRODUCTION

As the rapid advance in media creation, storage, and compression technologies, large amounts of multimedia content have been created and disseminated by various ways. Massive multimedia data challenge users in content browsing and retrieving, thereby motivating the urging needs of information mining technologies. To facilitate effective or efficient multimedia document indexing, many research issues have been investigated. Shot boundary detection algorithms are amply studied [1, 2] to discover the structure of video. With the understanding of video structure, video adaptation applications [3] are then developed to manipulate information more flexibly. Moreover, techniques for genre classification are also investigated to facilitate browsing and retrieval. Audio classification and segmentation techniques [4, 5] are proposed to discriminate different types of audio, such as speech, music, noise, and silence. Additional work focuses on classifying musical sounds [6] and automatically constructing music snippets [7]. For video content, genres of films [8] and TV programs [9] are automatically classified by exploring various features. Features from audio, video, and text [10] could be exploited to perform content analysis, and multimodal approaches are proposed to efficiently cope with the access and retrieval issues of multimedia content.

On the basis of physical features, the paradigms described above are developed to automatically analyze multimedia content. However, they pose many problems in today's applications. The semantic gap between low-level features and high-level concepts degrades the performance of multimedia content management systems. Similarities in low-level features do not certainly match with user's perception. Scenes or shots are associated due to semantics rather than physical features like color layouts and motion trajectories. Therefore, it would be more reasonable to discover information from meaningful events or objects rather than physical features.

To diminish the differences between analysis results and user's expectation, two research directions are emerged. The first is to detect attractive parts of movies or TV programs by exploiting the domain knowledge and production rules. According to media aesthetics [11], which includes the study and analysis of media elements commonly applied, the related studies attempt to uncover the semantic and semiotic information by computational frameworks. Preliminary results have been reported on film tempo analysis [12] and scare scene detection in horror movies [13].

Semantic indexing is another emerging study that identifies objects and events in audiovisual streams and facilitates semantic retrieval or information mining. The major challenge of this work is to bridge the gaps between physical features and semantic concepts. Studies on semantic indexing can be separated into two levels: isolated audio/video event detection and semantics identification. Former studies [14, 15] took advantage of HMM-based approaches to

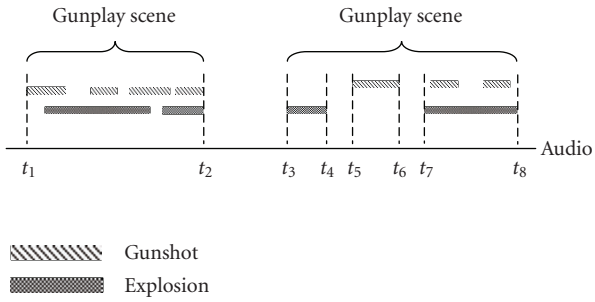


Figure 1: Examples of audio semantic contexts.

tackle event detection. Audio events such as applause, laughter, and cheer are modeled. However, in today's applications, detecting isolated audio/video events is not quite intuitive to users. For example, rather than identifying individual gunshots in an action movie, we are more likely to recognize a scene of gunplay, which may consist of a series of gunshots, explosions, sounds of jeeps, and screams from soldiers. Such a scene conveys a solid semantic meaning and is at a reasonable granularity for semantic retrieval. For modeling visual semantics, some approaches based on Bayesian network [16] and support vector machine [17] have been proposed to fuse the information of visual events and to infer some semantic concepts, such as "outdoor" or "beach" concepts. However, few studies are reported to perform audio-based semantic context detection. In some types of videos, such as action movies, audio information plays a more important role than visual ones. For example, a gunplay scene may occur in a rainforest or a downtown street, at day or night, which has significant variations in vision. On the contrary, aural information remains similar in different gunplay scenes, and some typical audio events (e.g., gunshot and explosion sounds in gunplay scenes) significantly provide the clues for detecting semantic concepts.

Due to rapid shot changes and dazzling visual variations in action movies, our studies focus on analyzing audio tracks and accomplish semantic indexing via aural clues. In this paper, an integrated hierarchical framework is proposed to detect two semantic contexts, that is, "gunplay" and "car chasing," in action movies. To characterize these two semantic contexts by event fusion, "gunshot" and "explosion" sound effects are detected for "gunplay" scenes, and "car-braking" and "engine" sounds are detected for "car-chasing" scenes. For audio event modeling, HMM-based approaches that have been applied in visual event modeling [15] are used. Then, "gunplay" and "car-chasing" scenes are modeled based on the statistical information from audio event detection. For semantic context modeling, generative (hidden Markov model) and discriminative (support vector machine) approaches are investigated. We view semantic context detection as a problem of pattern recognition, and similar feature values (detection results of audio events) would be fused to represent a semantic context. For example, gunplay scenes may have similar gunshot and explosion occurrence patterns and are distinguished from other scenes by pattern recognition techniques. We discuss how the fusion approaches work and show the effectiveness of this event fusion framework. The results of semantic context detection can be applied to multimedia indexing and facilitate efficient media access.

The remainder of this paper is organized as follows. Section 2 describes the definitions of audio event and semantic context and states the concept of hierarchical audio models. The audio features we used for event modeling are briefly introduced in Section 3. In Section 4, HMMs are used to model audio events, and we introduce the idea of event fusion by constructing pseudosemantic features. Sections 5 and 6 address issues on fusion schemes based on HMM and SVM, respectively. Performance evaluation, comparison, and some discussions are shown in Section 7, and the concluding remarks are given in Section 8.

## 2. HIERARCHICAL AUDIO MODELS

The semantic indexing process is performed in a hierarchical manner. At the audio event level, the characteristics of each audio event are modeled by an HMM in terms of the extracted audio features. At the semantic context level, the results of audio event detection are fused by using generative (HMM) or discriminative (SVM) schemes.

### 2.1. Audio event and semantic context

Audio events are defined as short audio clips which represent the sound of an object or an event. On the basis of elaborately selected audio features, fully connected (ergodic) HMMs are used to characterize audio events, with Gaussian mixtures modeling for each state. Four audio events, including gunshot, explosion, engine, and car braking, are considered in this work.

In this study, we aim at indexing multimedia documents by detecting semantic concepts. A semantic concept may be derived from the association of various events. Therefore, we introduce the idea of modeling a semantic concept via the context of relevant events, which is then called semantic context for short. To characterize a semantic context, the information of specific audio events, which are highly relevant to some semantic concepts, are collected and modeled. In action movies, the occurrences of gunshot and explosion events are used to characterize "gunplay" scenes. The occurrences of engine and car-braking events are used to characterize "car-chasing" scenes.

For a semantic context, there may be no specific evolution pattern along the time axis. For example, in a gunplay scene, we cannot expect that explosions always occur after gunshots. Moreover, there may be some silence segments which contain no relevant audio events, but they are viewed as parts of the same gunplay scene in human's sense. Figure 1 illustrates examples of "gunplay" semantic concepts. The audio clip from $t1$ to $t2$ is a typical gunplay scene which contains mixed relevant audio events. In contrast to this case, no relevant event exists from $t4$ to $t5$ and from $t6$ to $t7$. However, the whole audio clip from $t3$ to $t8$ is viewed as the same scene in user's sense, as long as the duration of the "irrelevant

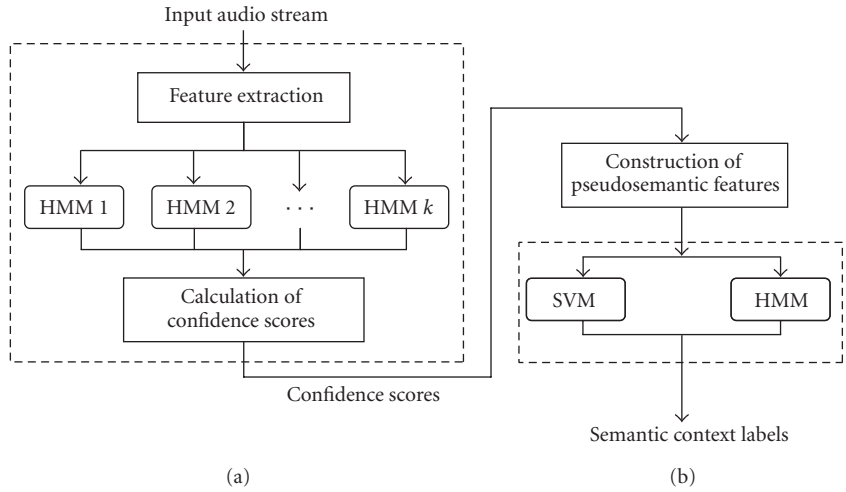


Figure 2: The proposed hierarchical framework contains (a) audio event and (b) semantic context modeling.

clip" does not exceed users' tolerance. Therefore, to model the characteristics of semantic contexts, we develop an approach that takes a series of events along the time axis into account rather than just the information at a time instant.

Note that multiple audio events may occur simultaneously, as shown in the duration from $t1$ to $t2$ in Figure 1. Some studies have been conducted to separate mixed audio signals in speech and music domains, by using independent component analysis [18]. The reported works are mainly performed on synthetically mixed audio signals or sounds recorded at simple acoustic conditions. However, separating mixed audio effects recorded in complicated real-world situations is not widely studied. In this work, when multiple audio events are mixed, we simply select two representative events to describe the characteristics of the corresponding audio clip. Although separating mixed audio effects is possible, elaborate studies on this issue are beyond the scope of this paper.

### 2.2. Hierarchical framework

The proposed framework consists of audio event and semantic context modeling. Some essential audio features from training corpus are first extracted and modeled by HMMs, as shown in Figure 2(a). After constructing each audio event model, the likelihood of a test audio segment with respect to each audio event can be computed through the Forward algorithm [19]. To determine how a segment is close to an audio event, a confidence metric based on the likelihood ratio test [20] is defined. We say that the segments with higher confidence scores from the gunshot model, for example, imply higher probability of the occurrence of gunshot sounds.

In the stage of semantic context modeling/detection, the confidence values from event detection constitute the cues for characterizing high-level semantic contexts. The *pseudosemantic features* that indicate the occurrences of events are constructed to represent the association of audio clips. We call them pseudosemantic features because they represent the interrelationship of several audio events, which are grounds for users to realize what the clip presents. With these features, two approaches based on generative and discriminative strategies are investigated to model semantic contexts, as shown in Figure 2(b). As the usage in pattern recognition and data classification, HMM and SVM shed lights on clustering these pseudosemantic features and facilitate detection processes.

## 3. AUDIO FEATURE EXTRACTION

One important factor for pattern recognition is the selection of suitable features that characterize original data adequately. To analyze audio sequences, several audio features from time-domain amplitude and frequency-domain spectrogram are extracted and utilized. In our experiments, all audio streams are downsampled to the 16 KHz, 16 bits, and monochannel format. Each audio frame is of 25 milliseconds, with 50% overlaps. Two types of features, that is, perceptual features and Mel-frequency cepstral coefficients (MFCC), are extracted from each audio frame. The perceptual features include short-time energy, band energy ratio, zero-crossing rate, frequency centroid, and bandwidth [10]. These features are shown to be beneficial for audio analysis and are widely adopted [4–7, 14].

Short-time energy (STE) is the total spectrum power of an audio signal at a given time and is also referred to loudness or volume in the literature. It provides a convenient representation of the amplitude variations over time. To reduce the clip-level fluctuation of volume mean, we normalize the volume of a frame based on the maximum volume of the corresponding audio clip.

In order to model the characteristics of spectral distribution more accurately, the band energy ratio (BER) is considered in this work. The entire frequency spectrum is divided into four sub-bands with equal frequency intervals, and the

ratio number is calculated from the energy of each band divided by the total energy value.

Zero-crossing rate (ZCR) is defined as the average number of signal sign changes in an audio frame. It gives a rough estimate of frequency content and has been extensively used in many audio processing applications, such as voiced and unvoiced components discrimination, endpoint detection, and audio classification.

After Fourier transformation, frequency centroid (FC) and bandwidth (BW) are calculated to present the first- and second-order statistics of the spectrogram. They, respectively, represent the "center of gravity" and variances of the spectrogram, and their reliability and effectiveness have been demonstrated in previous studies [10].

Mel-frequency cepstral coefficients (MFCCs) are the most widely used features in speech recognition and other audio applications. They effectively represent human perception because the nonlinear scale property of frequencies in the human hearing system is considered. In this work, based on the suggestion in [21], 8-order MFCCs are computed from each frame.

The extracted features from each audio frame are concatenated as a 16-dimensional (1(STE) + 4(BER) + 1(ZCR) + 1(FC) + 1(BW) + 8(MFCC)) feature vector. Details of the audio feature extraction processes can be found in [10]. Note that the temporal variations of the adopted features are also considered. That is, the differences of the features between two adjacent frames are calculated. Therefore, by concatenating the feature vector of the $i$th frame and the differences between the $i$th and the $(i+1)$th frames, a 32-dimensional (32D) vector is finally generated for each audio frame.

(1) Define two thresholds: $t_1$ and $t_2$, with $t_1 > t_2$.

(2) Take the first sample $\mathbf{v}_1$ as the representative of the first cluster: $\mathbf{z}_1 = \mathbf{v}_1$, where $\mathbf{z}_1$ is the center of the first cluster.

(3) Take the next sample $\mathbf{v}$ and compute its distance $d_i(\mathbf{v}, \mathbf{z}_i)$ to all the existing clusters, and choose the minimum of $d_i$ : $\min\{d_i\}$.

  (a) If $\min\{d_i\} \leqq t_2$, assign $\mathbf{v}$ to cluster $i$ and update the center of this cluster: $\mathbf{z}_i$.
  (b) If $\min\{d_i\} > t_1$, a new cluster with center $\mathbf{v}$ is created.
  (c) If $t_2 < \min\{d_i\} \leqq t_1$, no decision will be made as the sample $\mathbf{v}$ is in the intermediate region.

(4) Repeat *Step 3* until all samples have been checked once. Calculate the variances of all clusters.

(5) If the current variance is the same as that of the last iteration, the clustering process has converged, go to *Step 6*. Otherwise, return to *Step 3* for further iteration.

(6) If the number of unassigned samples is larger than a certain percentage $\rho (0 \leqq \rho \leqq 1)$, increase $t_1$ or decrease $t_2$ while remaining $t_2 > 2t_1$ and start with *Step 2* again. Otherwise, assign the unassigned samples to the nearest clusters and end the process.

Algorithm 1: Adaptive sample set construction.

## 4. AUDIO EVENTS MODELING

Detecting specific events in audio streams is crucial, which will benefit the higher-level analysis of multimedia documents and facilitate the modeling of the human attention and perception more accurately. This section addresses some issues of audio event modeling, including the determination of model size, model training process, and the construction of pseudosemantic features for semantics modeling.

### 4.1. Model size estimation

We use HMMs to describe the characteristics of audio events. The 32D feature vectors from a type of audio event are grouped into several sets. Each set denotes one kind of timbre, and is modeled later by one state of an HMM. Determining a proper model size is crucial in applying HMMs. The state number should be large enough to describe the variations of features, while it should also be compact when we consider computational cost of model training process. In this work, an adaptive sample set construction technique [22] is adopted to estimate a reasonable model size of each audio event. The algorithm is described in Algorithm 1.

The thresholds $t_1$, $t_2$, and $\rho$ are heuristically designated such that different clusters (states) have distinct differences. In this work, $\rho$ is set as 0.1 to guarantee more than ninety percent of data are clustered. The initial values of $t_1$ and $t_2$ could be empirically set, as their initial values just affect the number of iterations for convergency, but not the final results that indicate the number of major clusters. The distance measure $d_i(\mathbf{v}, \mathbf{z}_i)$ we used is the Euclidean distance. As Gaussian mixtures are able to handle the slight differences within each state, we tend to keep the number of states less than ten by considering the effectiveness and efficiency of the training process.

A professional sound effects library is used to be the training corpus [23]. Through the above process, the estimated state numbers for car braking and engine are two and four, and both the state numbers for gunshot and explosion are six. These results make sense because, for each audio event, various kinds of sounds are collected in this sound library, and these numbers represent the degree of variations of each audio event. For example, the sounds of rifle and hand/machine gun are all collected as the gunshots. They vary significantly and should be represented by more state numbers than simple sounds, such as the sharp but simple car-braking sounds.

### 4.2. Model training

For modeling gunplay and car-chasing scenes in action movies, the audio events we modeled are gunshot, explosion,

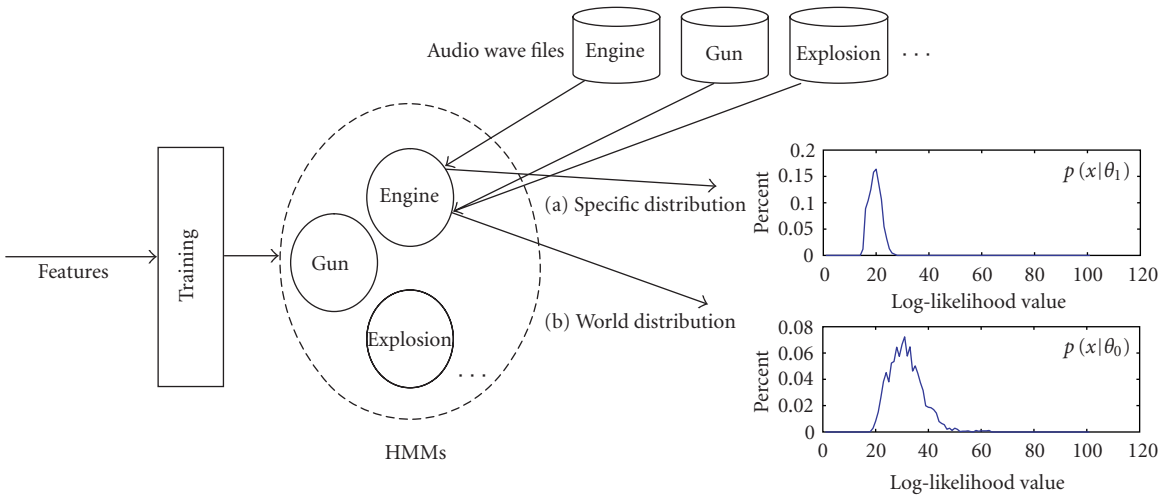


Figure 3: Construction of (a) specific distribution $p(x \mid \theta_1)$ and (b) world distribution $p(x \mid \theta_0)$ for engine events.

engine, and car braking. For each audio event, 100 short audio clips, each with length 3–10 seconds, are selected from the SoundIdeas sound effects library as the training data. In the training stage, the training audio streams are segmented into overlapped frames, and the features described in Section 3 are extracted. Based on these features, a complete specification of HMM, which includes two model parameters (model size and number of mixtures in each state) and three sets of probabilities (initial probability, observation probability, and transition probability), are determined. The model size and initial probability could be decided by the clustering algorithm described in the previous subsection, and the number of mixtures in each state is empirically set as four because it is insensitive to the system performance according to our experiments. The Baum-Welch algorithm is then applied to estimate the transition probabilities between states and the observation probabilities within each state. Finally, four HMMs are constructed for the audio events we concern. Details of the HMM training process can be found in [19].

### 4.3. Specific and world distributions

After audio event modeling, for a given audio clip, the log-likelihood values with respect to each event model are calculated by the Forward algorithm. Because a sound effect often lasts more than one second, the basic units we analyze for event detection are 1-second audio segments (called *analysis window* in this work) with 50% overlapping with adjacency segments. In event detection, the most important issue is how to decide whether an event occurs. According to the definition of HMM's evolution problem, the solution of Forward algorithm scores how well a given model matches a given observation sequence. However, unlike audio classification or speech recognition, we cannot simply classify an audio segment as a specific event even if it has the largest log-likelihood value. It may just present general environmental sound, and does not belong to any predefined audio event. Therefore, to evaluate how likely an audio segment belongs to a specific audio event, a log-likelihood-based decision method motivated from the speaker and world models in speaker verification [24] is proposed.

For each type of audio event, two distributions are constructed from the log-likelihood values. The first distribution represents the distribution of the log-likelihood values obtained from a specific event model $i$ with respect to the corresponding audio sounds. For example, from the "engine" model with the set of engine sounds as inputs, the resulting log-likelihood values are gathered to form the distribution. Figure 3(a) illustrates this construction process, and we call this distribution the *specific distribution,* $p(x \mid \theta_1)$, of the engine model. In contrast, the second distribution represents the distribution of the log-likelihood values obtained from a specific audio event model with respect to other audio sounds. As shown in Figure 3(b), the *world distribution,* $p(x \mid \theta_0)$, of the engine model is constructed from the log-likelihood values gathered from the engine model with the sets of gun, explosion, and car-braking sounds as inputs. Overall, engine model's specific distribution describes how the engine HMM evaluates engine sounds, while its world distribution describes how the engine HMM evaluates other kinds of sounds. These two distributions show how log-likelihood values vary with respect to a specific audio event and help us discriminate a specific audio event from others.

### 4.4. Pseudosemantic features

Based on the distributions, we can evaluate how likely an audio segment (as the unit of analysis window) belongs to a specific audio event and compute a confidence score. The audio segments with low average short-time energy and zero-crossing rate are first marked as silence, and the corresponding confidence scores with respect to all audio events are set as zero. For non-silence segments, the extracted feature vectors are input to the four HMMs. For a given audio segment, assume that the log-likelihood value from an event model is

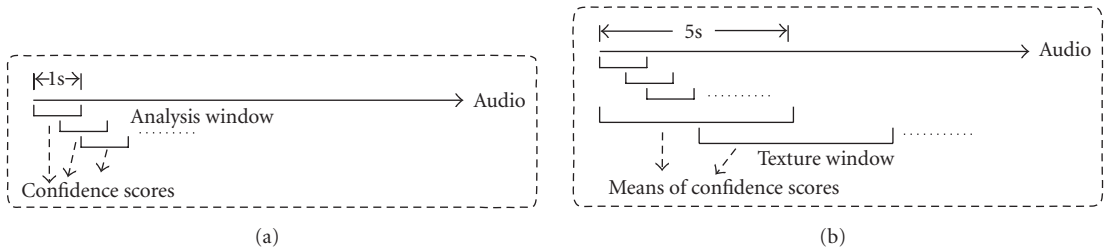


Figure 4: Pseudosemantic features calculation for semantic contexts modeling: (a) analysis windows and (b) texture windows.

$x$, the confidence score with respect to audio event $i$ is defined as

$$c_i = \frac{p_i(x \mid \theta_1)}{p_i(x \mid \theta_0)}, \tag{1}$$

where $p_i(x \mid \theta_1)$ and $p_i(x \mid \theta_0)$, respectively, denote the magnitudes of log-likelihood value $x$ with respect to the specific and world distributions of event $i$. The value $c_i$ represents the confidence score of the audio segment belonging to event $i$. Note that if the testing audio segment is out of the predefined set, both log-likelihood values with respect to the specific and world distributions are very likely to be zeros. We heuristically set the value $c_i$ as zero for rejection in this case.

By the definition in Section 2.1, a semantic context often lasts for at least a period of time, and not all the relevant audio events exist at every time instant. Therefore, the confidence scores of several consecutive audio segments are considered integrally to capture the temporal characteristics in a time series [6]. We define a *texture window* (cf., Figure 4(b)) of 5-second long, with 2.5-second overlaps, to go through the confidence scores of *analysis windows*.

For describing the semantic contexts of audio streams, *pseudosemantic features* that are constructed from the results of event detection are proposed. Based on the idea of event fusion, the pseudosemantic features for each texture window are constructed as follows.

(1) For each texture window, the mean values of confidence scores are calculated:

$$m_i = \text{mean}(c_{i,1}, c_{i,2}, \ldots, c_{i,N}), \quad i = 1, 2, 3, 4, \tag{2}$$

where $c_{i,j}$ denotes the confidence score of the $j$th analysis window with respect to event $i$, and $N$ denotes the total number of analysis windows in a texture window.

By the settings described above, nine analysis windows ($N = 9$), with 50% overlapping, construct a texture window. The corresponding sound effects to events 1 to 4 are "gunshot," "explosion," "engine," and "car braking."

(2) Let $b_i$ be a binary variable describing the occurrence situation of event $i$. The pseudosemantic feature vector $v_t$ for the $t$th texture window is defined as

$$v_t = [b_1, b_2, b_3, b_4], \tag{3}$$

$b_i = 1$ and $b_j = 1$ if the corresponding $m_i$ and $m_j$ are the first and the second maximums of $(m_1, m_2, m_3, m_4)$. Otherwise, $b_k = 0$.

(3) The total pseudosemantic features $V$ is represented as

$$V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_T \end{bmatrix}, \tag{4}$$

where $T$ is the total number of texture windows in the audio clip.

Calculating running mean values of confidence scores is to describe the characteristics over a number of analysis windows. We did also consider running variances in pseudosemantic features construction, but the final detection performance does not change significantly. The process of binarization is to emphasize the differences between confidence values with respect to different events. If a sound effect is more apparent than others, larger confidence score will be obtained. Therefore, we prompt the events with the first and the second largest confidence values and suppress those with smaller confidence values.

We call the features pseudosemantic features because they represent the intermediate characteristics between low-level physical features and high-level semantic concepts. The audio segments with higher confidence scores in the audio events relevant to a concept are more likely to convey this concept. For example, the audio segments with higher confidence scores in gunshot and explosion events somehow drop hints on the occurrence of gunplay scenes. To accomplish fusing information from different events, we investigate generative and discriminative approaches to model the pseudosemantic features. HMM is selected to be the instance of generative approach, and SVM is treated as the instance of discriminative approach.

## 5. HMM FOR SEMANTIC CONTEXT MODELING

For describing a sophisticated semantic context, a general model; for example, Gaussian mixture model, that only covers the event data distributions may not be enough. It is preferable to explicitly model the time duration density by including the concept of state transition. The appearance of relevant events does not remain the same at every time instant. There would be some segments with low confidence scores because the sound effect is unapparent or is influenced by other environmental sounds. On the other hand,

some segments may pose higher confidence because the audio events raise or explosively emerge. A model with more descriptive capability should take the temporal variations into consideration.

HMM is widely applied in speech recognition to model the spectral variations of acoustic features. It captures the time variation and state transition duration from training data. In speech-related applications, the left-right HMMs, which only allow state index increasing (or staying the same) as time goes by, are considered to be suitable. But in the case of semantic context modeling, there is no specific consequence formally representing the time evolution. Therefore, ergodic HMMs, or the so-called fully connected HMMs, are used in this work.

### 5.1. Model training

To perform model training, ten gunplay and car-chasing scenes, each with length 3–5 minutes, are manually selected from several Hollywood action movies as the training corpus. Based on user's sense, the movie clips that completely present gunplay or car-chasing scenes are selected, no matter how many gunshots, engine, or other relevant audio events occur. In model training, audio events are first detected and the pseudosemantic features are constructed based on the results of event detection. The pseudosemantic features from each semantic context are then modeled by an HMM again. For each HMM, the state number is estimated as two and the characteristics of each state are described by one Gaussian mixture. The obtained HMMs elaborately characterize the densities of time-variant features and present the structures of sophisticated semantic contexts.

### 5.2. Semantic context detection

The semantic context detection process is conducted following the same idea as that of the audio event detection. For every 5-second audio segment (a texture window), the log-likelihood calculated by the Forward algorithm represents how the semantic context models match the given pseudosemantic features. The binary indicator $\alpha_{s,t}$ is defined to show the appearance of semantic context $s$ at the $t$th texture window, $s = 1$ and 2, respectively, for gunplay and car-chasing scenes. That is,

$$\text{If } \sigma_s > \varepsilon, \quad \alpha_{s,t} = 1. \qquad \text{Otherwise,} \quad \alpha_{s,t} = 0, \tag{5}$$

where $\sigma_s$ is the log-likelihood value under semantic context model $s$, and $\varepsilon$ is a predefined threshold for filtering out those texture windows with too small values. The threshold can be adjusted by the user to tradeoff the precision and recall of semantic context detection.

## 6. SVM FOR SEMANTIC CONTEXT

Support vector machine (SVM) has been shown to be a powerful discriminative technique [25]. It focuses on structural risk minimization by maximizing the decision margin. The goal of SVM is to produce a model which predicts target value of data instances in the testing set. In our work, we view the detection process as classifying testing feature vectors (pseudosemantic features) into one of the predefined classes (semantic context). Thus we exploit SVM classifiers to distinguish the textures of "gunplay," "car-chasing," and "others" scenes. Although the features obtained from the same semantic context may disperse variably in the feature space (which is caused by the various patterns of the same semantic context), the SVM classifier, which maps features into a higher dimensional space and finds a linear hyperplane with the maximal margin, can effectively distinguish one semantic context from others.

Note that SVMs were originally designed for binary classification. In this work, we should classify a segment into three scenes, thus the SVM classifiers should be extended to handle multiclass classification both in training and testing processes.

### 6.1. Model training

Recently, a few researches are conducted to reduce a multiclass SVM into several binary SVM classifiers [26]. According to the performance analysis of multiclass SVM classifiers [27], we adopt the "one-against-one" strategy to model these three scenes. Three SVM models are constructed, that is, "gunplay versus car chasing," "gunplay versus others," and "car chasing versus others." For training each classifier, feature vectors are collected and their labels are manually determined to construct instance-label pairs $(x_i, y_i)$, where $x_i \in R^n$ and $y_i \in \{1, -1\}$. An SVM finds an optimal solution of data separation by mapping the training data $x_i$ to a higher dimensional space by a kernel function $\phi$ up to a penalty parameter $C$ of the error term. In model training, the kernel function we used is the radial basis function (RBF), which has been suggested in many SVM-based researches. That is, our kernel function is

$$K(x, y) = \exp\left(-\gamma ||x_i - x_j||^2\right), \quad \gamma > 0. \tag{6}$$

It is crucial to find the right parameters $C$ and $\gamma$ in RBF. Therefore, we apply five-fold cross validation with a grid search of varying $(C, \gamma)$ on the training set to find the best parameters achieving the highest classification accuracy.

For training SVM classifiers, the pseudosemantic features obtained from four audio events are labeled manually based on the unit of a texture window. Then all labeled texture windows are collected together to produce the training vectors. Three binary SVM classifiers will be combined later to identify which semantic context a texture window belongs to.

### 6.2. Semantic context detection

In semantic context detection, the decision-directed acyclic graph SVM algorithm (DAGSVM) [26] is applied to combine the results of one-against-one SVMs. The DAGSVM algorithm has been shown to be superior to existing multiclass SVM algorithms in both training and evaluation speeds. Figure 5 illustrates one example of the detection procedure. Initially, the test vectors are viewed as the candidates for all

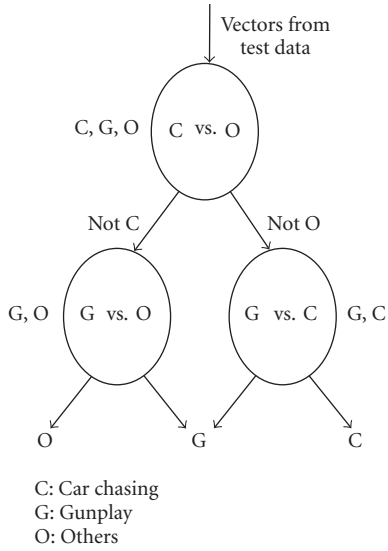


Figure 5: The testing procedure of DAGSVM.

three concepts. In the first step of detection, the test vectors are input to the root SVM classifier, that is, "car-chasing versus others" classifier. After this evaluation, the process branches to left if more vectors are predicted as the "others" category; and the "car-chasing" concept is removed from the candidate list. The "gunplay versus others" classifier is then used to reevaluate the test vectors. After these two steps, the vectors representing the characteristics of texture windows are labeled as "gunplay" or "others."

The DAGSVM separates the individual classes with large margins. It is safe to discard the losing class at each one-against-one decision because, for the hard margin case, all of the examples of the losing class are far away from the decision surface. Hence, the choice of the class order in detection procedure is arbitrary.

## 7. PERFORMANCE EVALUATION

We may first describe the characteristics of sound effects in movies before preparing the evaluation data. According to our observations, although the acoustic conditions may vary differently in different movies, the sound effects indicating a specific semantic concept fall into several fixed types. The reasons for this phenomenon include the following: (1) there have been some conventions to construct a concept in movie making, and (2) the sound effects are often added or embellished after shooting according to commonly used techniques. For example, in a gunplay scene, the sounds of gunshots can be often categories as several canonical types: hand gun, rifle, machine gun, and ricochet. Therefore, very huge amount of training data are not necessarily required.

For each audio event, 100 short audio clips, each with length 3–10 seconds, are selected. For semantic context modeling, because there is no standard corpus for audio semantic contexts, the evaluation data are manually selected from Hollywood movies. Thirty movie clips, each with length 3–5 minutes, are selected and labeled for each semantic context. Half of them are used as the dataset for model training, while half of them are used for model testing. Note that the criteria of selecting training data for audio events and semantic contexts are different. For semantic context modeling, we collected the "gunplay" and "car-chasing" scenes based on the experienced user's subjective judgments, no matter how many relevant audio events exist in the scene. On the contrary, the training data for audio event modeling are short audio segments that are exactly the audio events.

We evaluate the performance for both audio event detection and semantic context detection. Moreover, the effectiveness of this later fusion approach is evaluated by comparing it with the baseline approach that only exploits low-level audio features and works in an early fusion manner.

### 7.1. Evaluation of audio event detection

In audio event detection, audio streams are segmented into audio clips through analysis windows, as illustrated in Figure 4(a), and the log-likelihood values of audio clips in each analysis window with respect to four audio events are evaluated. The audio clip in an analysis window is correctly detected as the event $i$ if its corresponding confidence score is larger than a predefined threshold and is the maximum value with respect to all events. That is,

$$C = \max(c_1, c_2, c_3, c_4), \quad C > \delta, \tag{7}$$

where $c_i (i = 1, \ldots, 4)$, calculated from (1), is the confidence score with respect to event $i$, and $\delta$ is determined by the Bayesian optimal decision rule [20] on the basis of specific and world distributions. We decide that the analysis window with confidence score $x$ belongs to a specific event (category $\theta_1$) if

$$\lambda_{01} P(\theta_1 \mid x) > \lambda_{10} P(\theta_0 \mid x), \tag{8}$$

where $\lambda_{ij}$ is the cost incurred for deciding $\theta_i$ when the true state of nature is $\theta_j$.

By employing Bayes formula, we can replace the posterior probabilities by the prior probabilities and conditional densities. Then we decide $\theta_1$ if

$$\lambda_{01} p(x \mid \theta_1) P(\theta_1) > \lambda_{10} p(x \mid \theta_0) P(\theta_0), \tag{9}$$

and otherwise decide $\theta_0$.

Then we alternatively rewrite (9) and decide $\theta_1$ if

$$C = \frac{p(x \mid \theta_1)}{p(x \mid \theta_0)} > \frac{\lambda_{10}}{\lambda_{01}} \frac{P(\theta_0)}{P(\theta_1)} = \delta. \tag{10}$$

The prior probabilities are estimated based on our training data. The costs $\lambda_{10}$ and $\lambda_{01}$ could be adjusted to vary the value of threshold such that higher precision or recall could be achieved in the detection stage.

#### 7.1.1. Overall performance

The overall detection performance is listed in Table 1. The average recall is over 70%, and the average precision is about

Table 1: Overall performance of audio event detection.

| Audio event | Recall | Precision |
|---|---|---|
| Gun | 0.938 | 0.95 |
| Explosion | 0.786 | 0.917 |
| Brake | 0.327 | 0.571 |
| Engine | 0.890 | 0.951 |
| Average | 0.735 | 0.847 |

85%. Although the detection accuracy is often sequence-dependent and affected by confused audio effects, the reported performances support the applicability and superiority of the event modeling. In addition, different audio events have different evaluation results. Because the car-braking sounds are often very short in time (less than one second, which is the length of one basic analysis unit defined in our work) and are mixed with other environment sounds, the detection accuracy is particularly worse than others. This situation is different from gunshot sounds because there is often a continuity of gunshots (the sounds of a machine gun or successive handgun/rifle shoots) in a gunplay scene.

The detection performance is more encouraging if we neglect the particular case in car-braking detection. For other audio events, the average recall is 87%, and the average precision is 94%. On the other hand, because the car-braking sound is a representative audio cue of car-chasing scenes, we still take the detection results of car-braking sounds into account in car-chasing context modeling.

We also briefly investigate how different thresholds in (7) affect the detection performance. When we penalize misclassifying $\theta_0$ as $\theta_1$ (false alarm) more than the converse (i.e., $\lambda_{10} > \lambda_{01}$), we get larger threshold $\delta$, and hence higher precision but lower recall is expected. Figure 6 shows detection performance with four different thresholds ($\delta_1 > \delta_2 > \delta_3 > \delta_4$) from three different test sequences. Note that the trend of detection performance conforms to the general principle of pattern classification, while detection results are sequence-dependent.

#### 7.1.2. Performance comparison

To compare the performance of video retrieval/indexing between various approaches, some institutes such as TREC Video Retrieval Evaluation [30] developed corpus for video event evaluation. However, few standard datasets are designed for audio event detection. Most works of audio event detection (including our work) use privately collected datasets. Direct comparison between different approaches, which use different datasets and model different events, is not plausible. However, in order to show that the proposed approach achieves promising performances in detecting various audio events, we refer to other works that focused on audio events in sports games [28], TV shows [14], and movies [29].

Because not all referred works report precision and recall values, we only list the detection accuracy (precision) in Table 2 for fair comparison. In [28], four audio events including "acclaim," "whistle," "commentator speech," and "silence" are detected in soccer videos, while the "speech" and "silence" generally are not viewed as special sound effects. More than 90% of detection accuracy is achieved. In [14], the events of "laughter," "applause," and "cheer" are detected in TV shows. For each event, average precision values from three test sequences are listed. The most similar work to ours is [29]. It also introduces a variation of HMM to model audiovisual features of explosion events. More than 86% of explosion events are correctly detected, while we achieve 91.7% of precision. From these results, we can see that the proposed audio event detection module works at least as well as other reported approaches, and is capable of being a robust basis for higher level modeling.

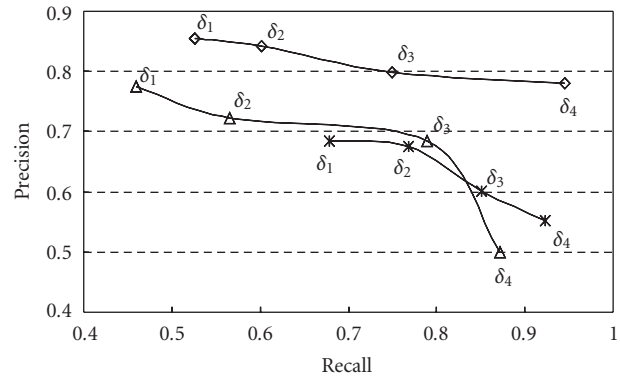


Figure 6: Three examples of detection performance with different thresholds ($\delta_1 > \delta_2 > \delta_3 > \delta_4$).

### 7.2. Evaluation of semantic context detection

In semantic context detection, the models based on HMM and SVM are evaluated, respectively. As the basic analysis unit is one texture window, the metrics of recall and precision are calculated to show the detection performance, as shown in Table 3. We tested movie clips from "We Were Soldiers," "Windtalker," "The Recruit," "Band of Brother," and so forth, for gunplay and movie clips from "Terminator 3," "Ballistic: Ecks vs. Sever," "The Rock," "2 Fast 2 Furious," and so forth, for car chasing. The detection performance is somewhat sequence-dependent because different movies posses different acoustic conditions. However, both the HMM-based and SVM-based approaches averagely achieve over 90% recall and near 70% precision in detecting gunplay and car-chasing scenes. These results show the promising achievement of the proposed fusion schemes.

Due to various acoustic conditions, the detection performances vary in different sequences. The accuracy of semantic context detection would degrade when bad audio event detection is involved. For example, in Table 4, the detection performance from two fusion schemes remains similar in the first two gunplay test sequences. However, the precision of "Imposter" degrades significantly, while the corresponding recall is similar to "44 Minutes." The reason is that many people yelling, strong alarm sounds, and violent background

Table 2: Detection accuracy of different approaches.

| | [28] | | [14] | | [29] | | Our approach | |
|---|---|---|---|---|---|---|---|---|
| Audio events | Acclaim | 98% | Laughter | 82.3% | Explosion | 86.8% | Explosion | 91.7% |
| | Whistle | 97.3% | Applause | 87.4% | | | Gun | 95% |
| | Commentator speech | 92.6% | Cheer | 92.6% | | | Brake | 57.1% |
| | Silence | 91.1% | | | | | Engine | 95.1% |

Table 3: Average performance of semantic context detection by (a) HMM and (b) SVM.

| Semantic context | Recall (a) | Precision (a) | Recall (b) | Precision (b) |
|---|---|---|---|---|
| Gunplay | 0.612 | 0.727 | 0.531 | 0.741 |
| Car chasing | 0.697 | 0.731 | 0.661 | 0.702 |

Table 4: Some detailed results in semantic context detection by (a) the HMM-based approach and (b) the SVM-based approach.

| Semantic context | | Recall (a) | Precision (a) | Recall (b) | Precision (b) |
|---|---|---|---|---|---|
| Gunplay | "We Were Soldiers" | 0.88 | 0.75 | 0.859 | 0.832 |
| | "44 Minutes" | 0.98 | 0.95 | 0.98 | 0.813 |
| | "Imposter" | 0.982 | 0.659 | 0.965 | 0.567 |
| Car chasing | "Ballistic: Ecks vs. Sever" | 0.99 | 0.83 | 0.98 | 0.839 |
| | "2 Fast 2 Furious" | 0.985 | 0.917 | 0.977 | 0.914 |
| | "The Rock" | 0.99 | 0.629 | 0.99 | 0.619 |

music occur in the test audio clip. These sound effects are often misdetected as explosion sounds and degrade the detection performance. Similar situations occur in the case of "The Rock" in car-chasing detection.

We further investigate how system performance varies with respect to different lengths of texture windows. The F1-metric, which jointly considers precision and recall, is used to indicate the system performance:

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{11}$$

Figure 7 shows the relationship between average performance of the HMM-based approach and lengths of texture windows. It is clear that the proposed system works particularly well when the length of texture window is set as five or six seconds. In this work, we simply take 5-second segments as the basic units for semantic context detection.

### 7.3. Comparison with baseline system

To show the superiority of the proposed framework, we compare the detection performance with that of the baseline case. The baseline system models semantic contexts directly by low-level features. For the semantic context training data, the audio features described in Section 3 are first extracted. Then these features are modeled by HMMs rather than constructing pseudosemantic features. In the experiment, the same training and testing data are used for the baseline system and the proposed framework.

Figure 8 illustrates the recall-precision curves of average detection performance. The proposed hierarchical framework shows its superiority over the baseline system. Because the baseline system does not take account of the information at event level, the precision rate degrades significantly as we increase recall. Linking the low-level features and high-level semantics by event fusion, that is, the construction of pseudosemantic features, provides a more robust performance in semantic context detection.

### 7.4. Discussion

Both the HMM-based fusion scheme and the SVM-based fusion scheme show their promising performance achievements. The most important advantage of event fusion approach is that event models can be trained separately, and new impacts from other events can be easily added to the framework. For example, more gunplay-related events such as helicopter-flying or people yelling can be modeled to augment the pseudosemantic features.

Although the effectiveness of this work has been shown, some issues should be discussed more. The main reason of performance degradation is (a) mixed audio signals and (b) confused acoustic characteristics between different sounds. One example of the former case is the simultaneous occurrence of gunshot and explosion, while the bass environmental sound may be misclassified as an engine event because of their acoustic similarity. For the problem (a), one of the solutions may be separating multisource audio signals and analyzing them individually. The studies of independent component analysis [18] would provide a new idea for this work. For the problem (b), more acoustic features should be explored specifically for event modeling and discrimination.

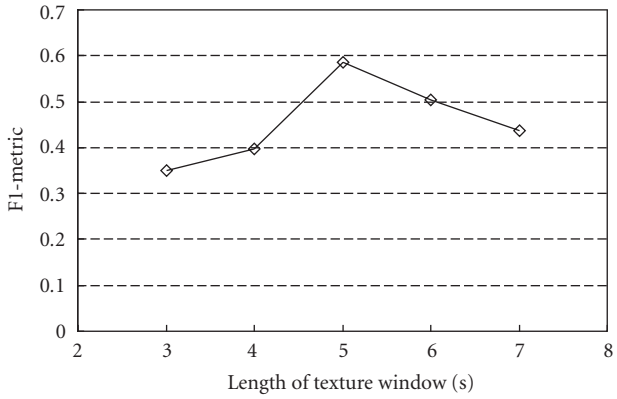


Figure 7: Relationship between lengths of texture windows and system performance.

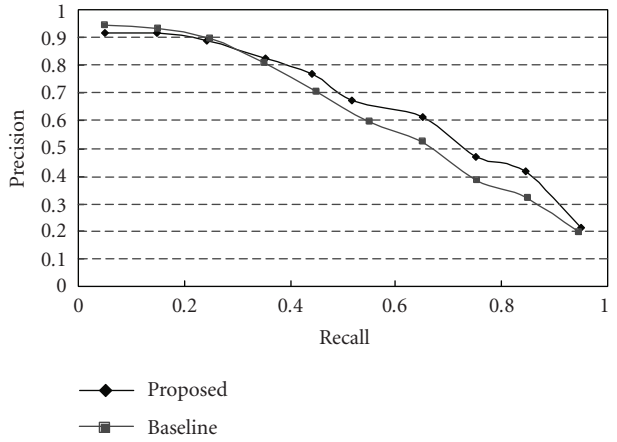


Figure 8: Comparison of the baseline and the proposed HMM-based approaches.

### 7.5. Semantic indexing based on the proposed framework

This work presents a preliminary try to identify the context of a semantic concept to facilitate multimedia retrieval. The results of semantic context detection index videos with distributions of semantic concepts rather than occurrences of isolated events or objects. It provides the idea that concept-based indexing could be achieved by fusing the information of relevant events/objects. Although the proposed framework is only applied to action movies, it is believed to be generalized to other types of videos. Meanwhile, another encouraging idea of this work is the introduction of the late fusion of individual classifiers. Individual classifiers can be trained separately and added adaptively to the final meta-classifier. On the basis of this framework, different semantic contexts could be modeled and detected by taking account of various visual and aural events. For example, replacing audio event models by visual object models, visual semantic context such as multispeaker conversation could be modeled by the same framework. Results from different modalities can also be fused (by careful design of pseudosemantic features) to construct a multimodal metaclassifier. Hence the proposed framework can qualify general semantic indexing tasks.

## 8. CONCLUSION

We present a hierarchical approach that bridges the gaps between low-level features and high-level semantics and facilitates semantic indexing in action movies. The proposed framework hierarchically conducts modeling and detection at two levels: audio event level and semantic context level. After careful selection of audio features, HMMs are applied to model the characteristics of audio events. At the semantic context level, generative (HMM) and discriminative (SVM) approaches are used to fuse pseudosemantic features obtained from the results of event detection. Experimental results demonstrate a remarkable performance of the fusion schemes, and signify that the proposed framework draws a sketch for constructing an efficient semantic indexing system.

The proposed framework can be extended to model different semantic concepts. It may be necessary to consider different combinations of events or include visual information according to the production rules of targeted films. Another possible improvement may include the elaborate feature selection by developing an automatic feature induction mechanism or applying the techniques of blind signal processing to deal with the problem of mixed audio effects.

## ACKNOWLEDGMENT

This work was partially supported by the National Science Council and the Ministry of Education of Taiwan under the Contract no. NSC94-2752-E-002-006-PAE and NSC94-2213-E-002-078.

## REFERENCES

[1] R. W. Lienhart, "Comparison of automatic shot boundary detection algorithms," in *Storage and Retrieval for Image and Video Databases VII*, vol. 3656 of *Proceedings of SPIE*, pp. 290–301, San Jose, Calif, USA, January 1999.

[2] A. Hanjalic, "Shot-boundary detection: unraveled and resolved?" *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 2, pp. 90–105, 2002.

[3] S.-F. Chang and A. Vetro, "Video adaptation: concepts, technologies, and open issues," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 148–158, 2005.

[4] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Transactions Speech Audio Processing*, vol. 10, no. 7, pp. 504–516, 2002.

[5] T. Zhang and C.-C. Jay Kuo, "Hierarchical system for content-based audio classification and retrieval," in *Multimedia Storage and Archiving Systems III*, vol. 3527 of *Proceedings of SPIE*, pp. 398–409, Boston, Mass, USA, November 1998.

[6] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions Speech Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[7] L. Lu and H.-J. Zhang, "Automated extraction of music snippets," in *Proc. 11th ACM International Conference on Multimedia*, pp. 140–147, Berkeley, Calif, USA, November 2003.
[8] S. Fischer, R. Lienhart, and W. Effelsberg, "Automatic recognition of film genres," in *Proc. 3rd ACM International Conference on Multimedia*, pp. 295–304, San Francisco, Calif, USA, November 1995.
[9] Z. Liu, J. Huang, and Y. Wang, "Classification of TV programs based on audio information using hidden Markov model," in *Proc. IEEE 2nd Workshop on Multimedia Signal Processing (MMSP '98)*, pp. 27–31, Redonda Beach, Calif, USA, December 1998.
[10] Y. Wang, Z. Liu, and J.-C. Huang, "Multimedia content analysis-using both audio and visual clues," *IEEE Signal Processing Magazine*, vol. 17, no. 6, pp. 12–36, 2000.
[11] H. Zettl, *Sight Sound Motion: Applied Media Aesthetics*, Wadsworth, Belmont, Calif, USA, 1999.
[12] C. Dorai and S. Venkatesh, *Media Computing: Computational Media Aesthetics*, Kluwer Academic, Boston, Mass, USA, 2002.
[13] S. Moncrieff, S. Venkatesh, and C. Dorai, "Horror film genre typing and scene labeling via audio analysis," in *Proc. IEEE International Conference on Multimedia and Expo (ICME '03)*, vol. 2, pp. 193–196, Baltimore, Md, USA, July 2003.
[14] R. Cai, L. Lu, H.-J. Zhang, and L.-H. Cai, "Highlight sound effects detection in audio stream," in *Proc. IEEE International Conference on Multimedia and Expo (ICME '03)*, vol. 3, pp. 37–40, Baltimore, Md, USA, July 2003.
[15] M. R. Naphade, T. Kristjansson, B. Frey, and T. S. Huang, "Probabilistic multimedia objects (multijects): a novel approach to video indexing and retrieval in multimedia systems," in *Proc. International Conference on Image Processing (ICIP '98)*, vol. 3, pp. 536–540, Chicago, Ill, USA, October 1998.
[16] M. R. Naphade and T. S. Huang, "Extracting semantics from audio-visual content: the final frontier in multimedia retrieval," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 793–810, 2002.
[17] J. R. Smith, M. Naphade, and A. Natsev, "Multimedia semantic indexing using model vectors," in *Proc. IEEE International Conference on Multimedia and Expo (ICME '03)*, vol. 2, pp. 445–448, Baltimore, Md, USA, July 2003.
[18] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, New York, NY, USA, 2001.
[19] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
[20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, New York, NY, USA, 2001.
[21] S. Z. Li, "Content-based audio classification and retrieval using the nearest feature line method," *IEEE Transactions Speech Audio Processing*, vol. 8, no. 5, pp. 619–625, 2000.
[22] S.-T. Bow, *Pattern Recognition and Image Preprocessing*, Marcel Dekker, New York, NY, USA, 2002.
[23] "Sound Ideas: Sound Effects Library," http://www.sound-ideas.com/.
[24] R. D. Zilca, "Text-independent speaker verification using covariance modeling," *IEEE Signal Processing Letters*, vol. 8, no. 4, pp. 97–99, 2001.
[25] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.
[26] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," in *Advances in Neural Information Processing Systems*, vol. 12, pp. 547–553, MIT Press, Cambridge, Mass, USA, 2000.
[27] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
[28] J. Wang, C. Xu, E. Chng, and Q. Tian, "Sports highlight detection from keyword sequences using HMM," in *Proc. IEEE International Conference on Multimedia and Expo (ICME '04)*, vol. 1, pp. 599–602, Taipei, Taiwan, June 2004.
[29] M. R. Naphade, A. Garg, and T. S. Huang, "Audio-visual event detection using duration dependent input output Markov models," in *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL '01)*, pp. 39–43, Kauai, Hawaii, USA, December 2001.
[30] "TREC Video Retrieval Evaluation," http://www-nlpir.nist.gov/projects/trecvid/.

**Wei-Ta Chu** received the B.S. and M.S. degrees in computer science and information engineering from National Chi Nan University in Nantou, Taiwan, in 2000 and 2002. He is currently pursuing his Ph.D. degree in the Department of Computer Science and Information Engineering, National Taiwan University, Taiwan. As he is in the Communication and Multimedia Laboratory, his research interests include digital content analysis, multimedia indexing, digital signal process, and pattern recognition.

**Wen-Huang Cheng** received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2002 and 2004, respectively, where he is currently pursuing the Ph.D. degree in the Graduate Institute of Networking and Multimedia. His research interests include multimedia data management and analysis.

**Ja-Ling Wu** received the B.S. degree in electronic engineering from TamKang University, Tamshoei, Taiwan, in 1979, and the M.S. and Ph.D. degrees in electrical engineering from Tatung Institue of Technology, Taipei, Taiwan, in 1981 and 1986. Since 1987, he has been with the Department of Computer Science and Information Engineering, National Taiwan University, where he is presently a Professor. He has published more than 200 journal and conference papers. His research interests include algorithm design for DSP, data compression, digital watermarking, and multimedia systems. He was the recipient of the Excellent Research Award from NSC, Taiwan, in 1999, 2001, and 2004.