Multimedia Systems (2005) 00: 1–14
DOI 10.1007/s00530-005-0183-6

REGULAR PAPER

Wei-Ta Chu · Wen-Huang Cheng · Jane Yung-Jen Hsu · Ja-Ling Wu

Toward semantic indexing and retrieval using hierarchical audio models

Received: 16 April 2004 / Revised: 20 November 2004 / Published online: 10 May 2005

Abstract Semantic-level content analysis is a crucial issue in achieving efficient content retrieval and management. We propose a hierarchical approach that models the statistical characteristics of audio events over a time series to accomplish semantic context detection. Two stages, audio event and semantic context modeling, are devised to bridge the semantic gap between physical audio features and semantic concepts. In this work, hidden Markov models (HMMs) are used to model four representative audio events, i.e., gunshot, explosion, engine, and car-braking, in action movies. At the semantic-context level, Gaussian mixture models (GMMs) and ergodic HMMs are investigated to fuse the characteristics and correlations between various audio events. They provide cues for detecting gunplay and car-chasing scenes, two semantic contexts we focus on in this work. The promising experimental results demonstrate the effectiveness of the proposed approach and exhibit that the proposed framework provides a foundation in semantic indexing and retrieval. Moreover, the two fusion schemes are compared, and the relations between audio event and semantic context are studied.

Keywords Audio event · Semantic context · Semantic gap · Hidden Markov model · Gaussian mixture model

W.-T. Chu (✉)
Department of Computer Science and Information Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan 106
E-mail: wtchu@cmlab.csie.ntu.edu.tw

W.-H. Cheng
Graduate Institute of Networking and Multimedia, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan 106
E-mail: wisley@cmlab.csie.ntu.edu.tw

J. Y.-J. Hsu · J.-L. Wu
Department of Computer Science and Information Engineering; Graduate Institute of Networking and Multimedia, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan 106
E-mail: {yjhsu, wjl}@csie.ntu.edu.tw

1 Introduction

Large amounts of multimedia content have been created, stored, and disseminated as a result of the rapid advance in media creation, storage, and compression technologies. Massive multimedia data present challenges to users in content browsing and retrieval, thereby diminishing the benefits brought by digital media. Technologies for effective and efficient multimedia document indexing are therefore essential to ease the load of media access.

Many research issues have been investigated to facilitate efficient access and usage of multimedia data. Shot boundary detection algorithms [1, 2] are developed to segment video clips into shots, each of which presents visual continuity. The keyframes of each shot are then selected to summarize a video clip and are applied to video abstraction [3, 4] and content-based retrieval [5]. On the other hand, techniques for genre classification are also widely studied. For audio tracks, classification and segmentation techniques [6, 7] are proposed to discriminate different types of audio, such as speech, music, noise, and silence. Additional work focuses on classifying types of music [8] and automatically constructing music snippets [9]. For video content, genres of films [10] and TV programs [11] are automatically classified. Various features from audio, video, and text [12] are exploited, and multimodal approaches are proposed to cope with the access and retrieval issues of multimedia content.

Although the paradigms described above are efficient for browsing and low-level search, problems do exist in today's applications. The first is the apparent gap between low-level audiovisual features and high-level semantics. Similarities in low-level features do not always match user perceptions. The second problem is that, from the viewpoint of end users, scenes/shots are associated due to semantic meaning rather than color layouts or other computational features. A tool that provides semantic-based query is more practical than tools supporting unlabeled shots or rough genre classification.

Recently, there have been two research directions in analyzing multimedia documents from a user's point of view.

The first is to build a user attention model [13, 14] to find user focus on audiovisual streams. Through combining the impacts of features or events (e.g., color contrast, intensity, face region, and motion) that may draw user attention, an integrated saliency map indicates the estimated attention focus. This work can be widely applied to deliberate video summarization [13] and user-centric scalable video streaming [15] and was demonstrated to be an efficient approach to capture user attention through empirical validation [16].

The second direction in usercentric multimedia analysis is to construct semantic indices for multimedia documents. Studies on semantic indexing can be separated into two phases: isolated audio/video event detection and semantics identification. Former studies [17, 18] took advantage of HMM-based approaches to tackling event detection. For example, in [17], several audio highlight events such as applause, laughter, and cheer are modeled by HMMs. In the test stage, an audio clip is divided into overlapping segments, and the audio features of each segment are extracted to be the input of three event models. Through a log-likelihood decision algorithm [17], highlighted events in audio clips are detected.

The approaches described above primarily detect audio events or video objects in audiovisual streams. However, detecting isolated audio/video events is not quite intuitive to the user. For example, rather than identifying gunshots individually 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. Instead of just modeling isolated events, approaches based on Bayesian network [19, 20] and support vector machine (SVM) [21] have been proposed to fuse information from isolated events and infer the semantic concept of a specific audio/video segment. Naphade and Huang [19, 20] adopt a probabilistic framework with the help of factor graphs [22] to model high-level semantic concepts, such as the scenes with "outdoor" or "beach." This framework models the correlations between different objects/events and provides inference functionalities. It boosts the performance of semantic object detection by taking inter-object correlations into account.

The framework described above can be applied to fuse various multimedia objects and to infer objects that are not easily modeled from low-level features. However, they do not model semantic contexts involving several multimedia objects over a time series. In multimedia retrieval, semantic contexts with a complete and continuous semantic meaning can often serve as the basic units that users want to retrieve. Therefore, a fusion scheme that models various audio events along the temporal axis should be devised to describe the context of a semantic concept.

In fact, similar ideas have been applied in many other research fields, such as speech recognition [23], data mining [24], and human–computer interaction [25]. Classifiers for individual objects or media are first constructed separately. The impacts of different classification results are viewed as intermediate features and are used to construct a higher-level classifier, called *metaclassifier*, for integrated applications. In speech recognition [23], a word-level recognizer is constructed through combining the results of syllable recognition. In data mining [24] and machine learning applications, metalearning provides a unifying and scalable solution when it is applied to large amounts of data. Moreover, the results from different modalities such as face and speech could be considered integrally to facilitate personal memory collection/retrieval [25]. Motivated by these works, we introduce the concept of metaclassifier to multimedia indexing and perform studies on different fusion schemes.

In this paper, an integrated hierarchical framework is proposed to model contexts of two semantic concepts (named "*semantic contexts*" for short in this paper), i.e., gunplay and car chase, in action movies. Due to rapid shot changes and dazzling visual variations in action movies, our investigation focuses on analyzing audio tracks and accomplishes semantic indexing via aural clues. By using the HMM-based approaches presented in [17, 18], low-level events, such as gunshot and explosion sounds, are modeled first. Gunplay and car-chasing scenes are then modeled based on the statistical information collected from various audio event detection results. Two methods are investigated to fuse this information: Gaussian mixture model (GMM) and hidden Markov model (HMM). The fusion work is viewed as a pattern recognition problem, and similar features (detection result of audio events) would be fused (clustered) to represent a semantic context. For example, gunplay scenes share similar gunshot and explosion occurrence patterns and can be distinguished from other scenes by pattern recognition and machine learning techniques. We discuss how the fusion approaches work and show the effectiveness of the hierarchical framework for semantic indexing.

The paper is organized as follows. In Sect. 2, an integrated hierarchical framework consisting of low-level audio event and high-level semantic context modeling is introduced. Essential feature extraction is described in Sect. 3. Section 4 presents audio event modeling, where the training issue and confidence evaluation are discussed. In Sects. 5 and 6, GMM and HMM are introduced to fuse information from audio event detection, respectively. We compare the performance of audio event detection and two fusing approaches in Sect. 7, and conclude this work in Sect. 8.

2 Hierarchical audio models

The semantic indexing process is performed in a hierarchical manner. Two stages of models, i.e., audio event modeling and semantic context modeling, are constructed to hierarchically characterize audio clips. 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 from audio event detection are fused by

using probabilistic models. Two schemes, GMM and HMM, are investigated to take on semantic context modeling.

2.1 Audio event and semantic context

Audio events are defined as short audio clips that represent the sound of an object or an event. They can be characterized by statistical patterns and temporal evolution of audio features. Therefore, an HMM [26] that accounts for both local variations and state transition is adopted to model audio events. Focusing on discovering semantic indices in action movies, four audio events, *gunshot*, *explosion*, *engine*, and *car-braking*, are modeled to capture significant events. Each audio event is modeled as a complete connected (ergodic) HMM, with continuous Gaussian mixtures modeling for each state. Given the feature vectors consisting of the extracted audio features, the HMM parameters are estimated by using the Baum–Welch algorithm [26].

From the viewpoint of [27], the latest multimedia content analysis gradually moves from processing on isolated objects, behaviors, or events to modeling of contexts and variations that cover spatiotemporal relations between successive or related objects/events. A more practical content analysis system is expected to include the concept of multimedia context in which we take into account the context of settings and cinematographic practices. Motivated by this idea, we consider "the context of a semantic concept" for modeling and detecting. It is called "*semantic context*" for short and is an analysis unit that represents more reasonable granularity for multimedia content usage or access. Instead of isolated audiovisual events or rough audio/video genres, a semantic context is a scene representing a solid concept to facilitate users in realizing video narrative. Typical examples include gunplay scenes (impressive scenes) in action movies and negotiation scenes (key scenes) in drama movies. The term "context" denotes that we deal with temporal variations or distribution of successive data. The term "semantic" is introduced because the proposed detection unit lies between low-level events/objects and high-level semantics/knowledge. In this work, we focus our efforts on detecting *gunplay* and *car-chasing* scenes via aural cues. It is believed that the concept and granularity of semantic context are more suitable for next-generation multimedia indexing and access.

Note that a semantic context may or may not contain all relevant audio events at every time instant. There is no specific evolution pattern along the time axis. For example, in a gunplay scene, one should not expect that explosions always occur after gunshots. Moreover, silence shots without relevant audio events can be viewed as part of the gunplay scene from a human perspective. Figure 1 illustrates the idea of semantic contexts. The audio clip from t1 to t2 is a typical gunplay scene that contains mixed relevant audio events. In contrast, the whole audio segment from t3 to t8 is viewed as a single scene even if no relevant audio event exists from t4 to t5 and from t6 to t7. Therefore, to model the characteristics of semantic contexts, we develop an approach that takes

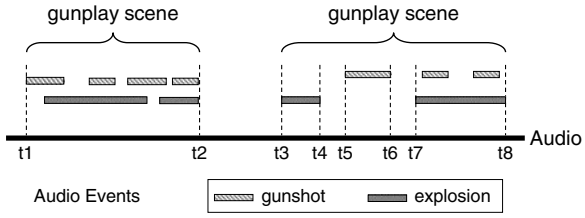

Fig. 1 Examples of audio semantic contexts

a series of events along the time axis into account rather than just the information at a single time instant.

This research aims to index multimedia documents by detecting high-level semantic contexts. To characterize a semantic context, audio events highly relevant to specific semantic concepts are collected and modeled. In our work, the occurrence patterns of *gunshot* and *explosion* events are used to characterize "*gunplay*" scenes, and the patterns of *engine* and *car-braking* events are used to characterize "*car-chasing*" scenes.

2.2 Hierarchical framework

The proposed framework consists of two stages: audio event modeling and semantic context modeling. First, as shown in Fig. 2a, the input audio stream is divided into overlapped segments, where audio features are extracted. Each HMM module takes the extracted features as input, and the Forward algorithm [26] is used to compute the log-likelihood of an audio segment with respect to each audio event. To determine how a segment is close to an audio event, a confidence metric based on the likelihood ratio test is defined. We say that the segments with higher confidence scores from the gunshot audio event model, for example, imply a higher probability of the occurrence of gunshot sounds.

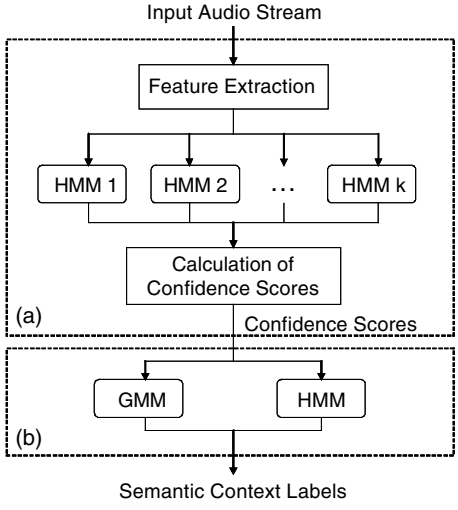

Fig. 2 The proposed hierarchical system framework contains **a** audio event and **b** semantic context modelings

In the stage of semantic context modeling, the confidence values from event detection constitute *pseudosemantic features* for characterizing high-level semantic contexts. Given an audio segment, the confidence values obtained from different event models are concatenated as a feature vector, which represents the characteristics of the audio segment with respect to several audio events. We call them *pseudosemantic features* because they characterize the interrelationship of several audio events, which are elements for users to realize what the segment represents. With these features, two modeling approaches are investigated to perform semantic context modeling, as shown in Fig. 2b. By the tools for pattern recognition and data classification, GMM and HMM shed lights on clustering these pseudosemantic features and facilitate detection processes.

3 Feature extraction

One important factor for pattern recognition is the selection of suitable features that characterize the original data adequately. To analyze audio sequences, several time and frequency domain audio features are extracted and utilized. In our experiments, all audio streams are downsampled to the 16-KHz, 16-bit monochannel format. Each audio frame is comprised of 400 samples (25 ms), with 50% overlaps. Two types of features, i.e., 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 [12]. These features are shown to be beneficial for audio analysis and are widely adopted [6–9,17].

Short-time energy (STE) is the total spectrum power of an audio signal at a given time and is also referred to as loudness or volume in the literature. It provides a convenient representation of signal 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.

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 subbands with equal frequency intervals, and the ratio is calculated from the energy of each band divided by the total energy.

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 "variance" of the spectrogram, and their reliability and effectiveness in characterizing the spectral information have been demonstrated in previous studies [12].

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 human hearing system is considered. In this work, based on the suggestion in [28], an eight-order MFCC is 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 [12]. Note that the temporal variations of the adopted features are also considered. That is, the differences of the features between adjacent frames are calculated and integrated into the original features. Therefore, a 32-dimensional feature vector is finally generated for each audio frame.

4 Audio event modeling

Audio events are defined as short audio segments that represent the sound of an object or an event. Audio event modeling is crucial to multimedia content analysis and is the foundation of advanced audio-based applications. This section addresses some issues of audio event modeling, including the determination of model size, model training process, and calculation of confidence scores from detection results.

4.1 Model size estimation

We use HMMs to characterize different audio events. The 32-dimensional feature vectors from a specific type of audio event are segmented into several sets, with each set denoting one kind of timbre, and are modeled later by one state of an HMM. Determination of model size is crucial in applying HMMs. The state number should be large enough to characterize the variations of features while it should be compact enough for efficient model training. In this work, an adaptive sample set construction technique [29] is adopted to estimate a reasonable model size. The algorithm is described as follows:

1. Take the first sample $\mathbf{x}_1$ as the representative of the first cluster: $\mathbf{z}_1 = \mathbf{x}_1$, where $\mathbf{z}_1$ is the center of the first cluster.
2. Take the next sample $\mathbf{x}$ and compute its distance $d_i = d(\mathbf{x}, \mathbf{z}_i)$ to the means of all existing clusters. Choose the minimum of d_i: min $\{d_i\}$.
   (a) Assign $\mathbf{x}$ to $\mathbf{z}_i$ if

$$\min\{d_i\} \leq \theta\tau, \ 0 \leq \theta \leq 1,$$

   where τ is the membership boundary for a specified cluster.
   (b) A new cluster with center $\mathbf{x}$ is created if

$$\min\{d_i\} > \tau.$$

(c) No decision will be made if

$$\theta\tau < \min\{d_i\} \leq \tau.$$

In this case, sample **x** is in the intermediate region.

(d) Update the mean of each cluster if case (a) or (b) is satisfied.

3. Repeat step 2 until all samples have been checked once. Calculate the variances of all the clusters.
4. If the variance is the same as the previous iteration, the clustering process has converged, go to step 5. Otherwise, go to step 2 for further iteration.
5. If the ratio of samples in the intermediate region is larger than a percentage $\rho(0 < \rho < 1)$, adjust θ and τ and go to step 2 again. Otherwise, the process ends.

The thresholds θ, τ, and ρ are heuristically designated such that different clusters (states) have distinct differences and physical meanings. The distance measure $d(\mathbf{x}, \mathbf{z}_i)$ is 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.

Through this process, the state number for *car-braking* is two, the state number for *engine* is four, and the state number for *gunshot* and *explosion* is six. These results make sense because we elaborately collect various kinds of training sounds for each audio event, and these results represent the variations of different audio events. For example, the sounds of rifle, handgun, and machine gun are all collected as the training data of *gunshot*. They vary significantly and should be represented by more states than simpler sounds, like the sharp but simple *car-braking* sounds.

4.2 Model training

Two semantic contexts, i.e., *gunplay* and *car-chasing*, in action movies are modeled in terms of the *gunshot*, *explosion*, *engine*, and *car-braking* audio events. For each audio event, 100 short audio clips each 3–10 s in length are selected from a professional sound-effects library (http://www.sound-ideas.com) as the training data. Based on the features extracted from the training data, a complete specification of HMM with two model parameters (model size and number of mixtures in each state) and three sets of probability measures, i.e., initial probability, observation probability, and transition probability, would be 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. The well-known Baum–Welch algorithm [26] in speech recognition fields is then applied to estimate the transition probabilities between states and observation probabilities in each state. Finally, four HMMs are constructed for the audio events considered. Details of the HMM training process will be further described in Sect. 6, where HMMs are also used for semantic context modeling.

4.3 Confidence evaluation

After audio event modeling, for each 1-s audio segment (0.5-s overlaps with adjacent segments), the log-likelihood values with respect to each event model are calculated by the Forward algorithm [26]. However, unlike audio classification, we cannot simply classify an audio segment as a specific event according to the highest log-likelihood value. It does not necessarily belong to any predefined audio event. In order to evaluate whether an audio segment belongs to a specific audio event, an approach based on the concept of *likelihood ratio test* [30] is applied.

For each type of audio event, two likelihood functions are constructed. The first function $f_i(x \mid \theta_1)$ represents the distribution of the log-likelihood obtained from a specific audio event model i with respect to the corresponding audio sounds. For example, from the *engine* model with *engine* sounds as inputs, the resulting log-likelihood values are gathered to form the distribution of the engine model with respect to engine sounds. We call it "*within distribution*" of the engine events, and Fig. 3a illustrates this construction process. In contrast, the second function $f_i(x \mid \theta_0)$ represents the distribution of the log-likelihood obtained from a specific audio event model with respect to other audio sounds. Like the previous example, as shown in Fig. 3b, the "*outside distribution*" of the *engine* event is constructed from the log-likelihood values gathered from the *engine* model with *gun*, *explosion*, and *car-braking* sounds as

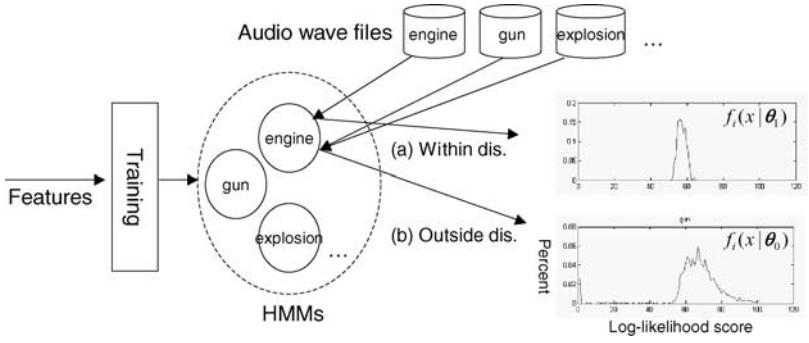

Fig. 3 Calculation of **a** within-distribution $f_i(x \mid \theta_1)$ and **b** outside-distribution $f_i(x \mid \theta_0)$ for audio event i

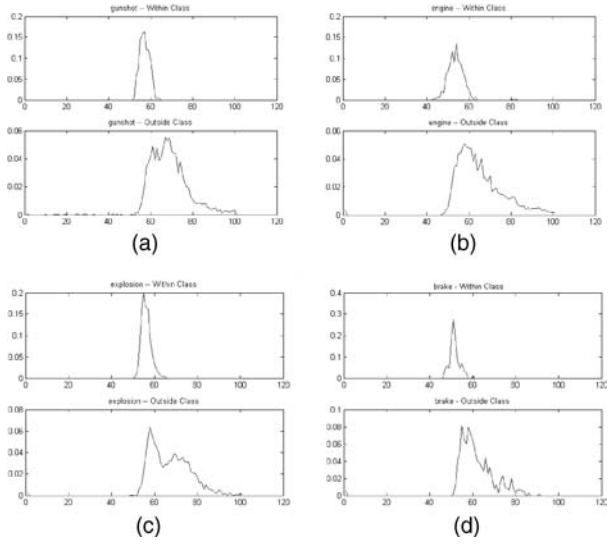

Fig. 4 Examples of within and outside distributions, including **a** gunshot, **b** engine, **c** explosion, and **d** car-braking

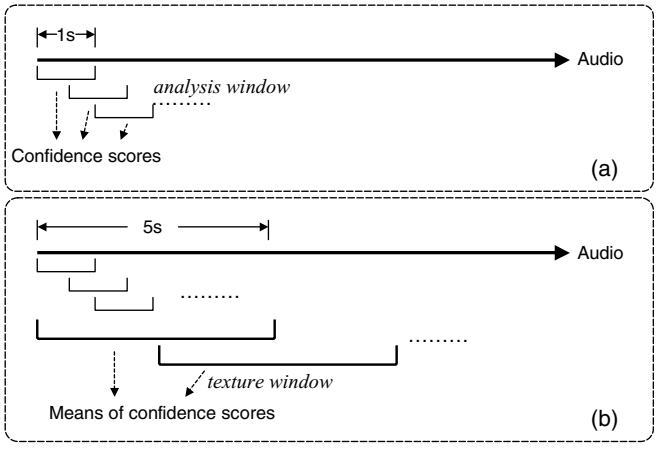

Fig. 5 Pseudosemantic features construction for semantic context modeling

inputs. These two distributions show how log-likelihood values vary within or outside a specific audio event and help us discriminate a specific audio event from others. Figure 4 shows some examples of "within" and "outside" distributions.

In the process of confidence evaluation, the segments with low average volume and zero-crossing rate are first marked as silence and the corresponding confidence values with respect to all audio events are set to zero. From non-silence segments, the extracted feature vectors are input to all four HMMs, and the corresponding log-likelihood values are calculated. For a given audio segment, assuming that the log-likelihood value from an event model is x, its confidence score with respect to audio event i is defined as:

$$c_i(x) = \frac{f_i(x \mid \theta_1)}{f_i(x \mid \theta_0)}. \tag{1}$$

The confidence scores calculated from each audio event are then considered to characterize an audio segment. The audio segments with higher confidence scores in the gunplay-related events, for example, are more likely to convey gunplay concepts. These confidence scores are the input of high-level modeling and provide important clues to bridge audio event and semantic context.

5 Gaussian mixture model for semantic context

5.1 Feature preprocessing

In this section, we aim at detecting high-level semantic contexts based on the results of audio event detection. The confidence scores for some specific audio events that are highly relevant to the semantic concept are collected and modeled. To perform model training, 30 gunplay and car-chasing scenes each 3–5 min in length are manually selected from 10 different Hollywood action movies as a training/testing dataset.

By the definition in Sect. 2.1, a semantic context often lasts for a period of time, and not all relevant audio events exist at every time instant. Therefore, the confidence scores of consecutive audio segments are considered integrally to capture the temporal characteristics in a time series [8]. We define a *texture window* (Fig. 5b) 5 s long, with 2.5-s overlaps, to go through the confidence values of 1-s audio segments (*analysis windows* in Fig. 5a). The means of confidence values in each texture window are then calculated, and the pseudosemantic features for each analysis 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 jth 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 of 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 tth texture window is defined as:

$$\mathbf{v}_t = [b_1, b_2, b_3, b_4], \tag{3}$$

   where $b_i = 1$ 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 overall pseudosemantic feature vector V is defined as:

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_T \end{bmatrix}, \tag{4}$$

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

Finally, the mean confidence values from all texture windows form the feature vector $\mathbf{V}$ for performing semantic modeling. We study pseudosemantic feature modeling by applying two statistical techniques: GMM and HMM.

5.2 Gaussian mixture model training

The use of GMM is motivated by the interpretation that the Gaussian components represent some general semantic-dependent spectral shapes and the capability of Gaussian mixtures to model arbitrary densities [31]. The individual component Gaussians in a GMM-based semantic context model are interpreted as the classes of audio events corresponding to specific semantic contexts. They reflect context-dependent configurations that are useful for modeling the characteristics of a specific semantic context. Furthermore, linear combination of Gaussian basis functions is capable of representing a large class of sample distributions. In this work, we respectively construct one four-mixtures GMM for gunplay and car-chasing semantic contexts.

A Gaussian mixture density is a weighted sum of M component densities. That is,

$$p(\vec{x}_i \mid \lambda) = \sum_{i=1}^{M} w_i b_i(\vec{x}_i), \tag{5}$$

where $\vec{x}_i = (\mathbf{V}_{1,i}, \mathbf{V}_{2,i}, \ldots, \mathbf{V}_{N,i})$ is the ith column vector of $\mathbf{V}$ and w_i is the weight of the ith mixture. Each component density $b_i(\vec{x}_i)$ is a D-variate Gaussian function of the form

$$b_i(\vec{x}_i) = \frac{1}{(2\pi)^{D/2} \mid \Sigma_i \mid^{1/2}} \times \exp\left\{-\frac{1}{2}(\vec{x}_i - \vec{\mu}_i)' \sum_{i}^{-1} (\vec{x}_i - \vec{\mu}_i)\right\} \tag{6}$$

with mean $\vec{\mu}_i$ and covariance matrix Σ_i, and D is 4 in this work. The mixture weights have to satisfy the constraint $\sum_{i=1}^{M} w_i = 1$.

The complete Gaussian mixture density $\lambda = \{w_i, \mu_i, \Sigma_i\}, i = 1, \ldots, M$, is parameterized by the mixture weights, the mean vectors, and the covariance matrix of all component densities. In model training, the pseudosemantic features are computed and the maximum-likelihood (ML) estimation is adopted to determine the model parameters. For a sequence of T training vectors $\mathbf{X} = x_1, \ldots, x_T$, the expectation–maximization (EM) algorithm [32] is applied to guarantee a monotonic increase in likelihood values:

$$\text{Mixture weights:} \quad \bar{w}_i = \frac{1}{T}\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda). \tag{7}$$

$$\text{Mean update:} \quad \vec{\bar{\mu}}_i = \frac{\sum_1^T p(i \mid \vec{x}_t, \lambda)\vec{x}_t}{\sum_1^T p(i \mid \vec{x}_t, \lambda)}. \tag{8}$$

$$\text{Variance update:} \quad \bar{\sigma}_i^2 = \frac{\sum_{t=1}^T p(i \mid \vec{x}_t, \lambda)x_t^2}{\sum_{t=1}^T p(i \mid \vec{x}_t, \lambda)} - \bar{\mu}_i^2. \tag{9}$$

The a posteriori probability for the ith mixture is then given by

$$p(i \mid \vec{x}_t, \lambda) = \frac{w_i b_i(\vec{x}_t)}{\sum_{k=1}^M w_k b_k(\vec{x}_t)}. \tag{10}$$

5.3 Semantic context detection

In semantic context detection, audio event detection is first applied. On the basis of confidence scores, each texture window is evaluated by checking the pseudosemantic features. If all feature elements are located in the "range of detection," we say that the audio segment corresponding to this texture window belongs to the specific semantic context. As shown in Fig. 6, the range of detection is defined as $[\mu_i - \delta_i, \mu_i + \delta_i]$, where μ_i and δ_i denote the mean and the standard deviation of the ith Gaussian mixture, respectively. In the case of *gunplay* scenes, if all the feature elements of *gunshot* and *explosion* events are located in the detection regions, it is said that the segment conveys the semantics of gunplay.

Although the relationships between semantic context and relevant audio events should be decided manually in the training stage, this work is not time consuming or tedious. There have been some movie production rules that elucidate which basic visual and aural elements can be synthesized into a complete semantic context [33]. The

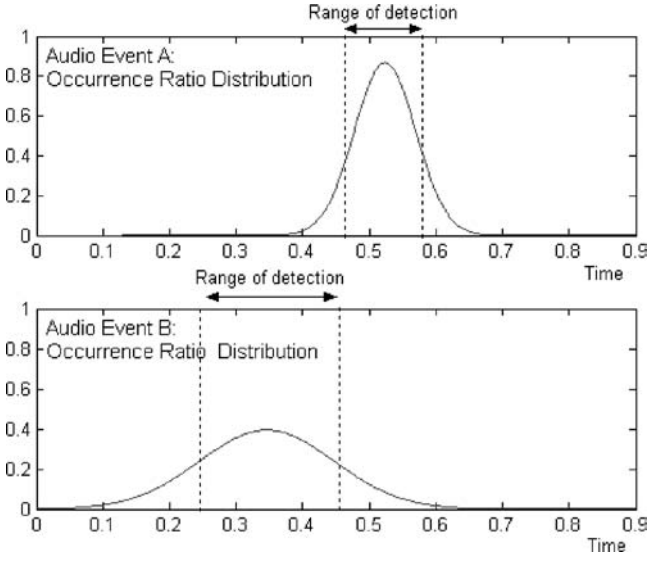

Fig. 6 The GMM for modeling semantic contexts

well-applied production rules, or the so-called *media aesthetics* [34], motivate us to take advantage of repeated use of certain objects/events to be the clues for detecting some specific semantic contexts. In other words, we can easily choose some visual/aural elements to be the primitive components to construct semantic context models. Then the modeling techniques take charge of representing the characteristics and relationships of these relevant events. More audio events could be applied to model semantic contexts more accurately. However, the experimental results show that these simple but representative audio events have achieved promising detection results.

6 Hidden Markov model for semantic context

The GMM-based fusion scheme constructs a general model for each semantic context and tackles different combinations of relevant events. However, for describing a sophisticated semantic context, a general model that only covers the event data distributions may not be enough. It is preferable to explicitly model the time duration density by introducing the concept of state transition. For example, the confidence scores of relevant events do 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 environment 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 consider the variation in time domain.

HMM is widely applied in speech recognition to model the spectral variation of acoustic features in time. It captures the time variation and state transition duration from training data and provides different likelihood values based on different given test data. In speech-related applications, left–right HMMs are considered suitable that only allow that state index increases (or stays the same) as time goes by. But in the case of semantic context modeling, there is no specific consequence that formally represents the time evolution. Therefore, ergodic HMMs, or the so-called fully connected HMMs, are used in our work.

6.1 Hidden Markov model training

A hidden Markov model $\lambda = (\mathrm{A}, \mathrm{B}, \pi)$ consists of the following parameters.

1. N, the number of states in the model. It can be decided by the algorithm described in Sect. 4.1. The individual states are labeled $1, 2, \ldots, N$, and the state at time t is denoted as q_t.
2. M, the number of distinct observation symbols in all states, i.e., the types of audio events we used. The individual symbols are denoted as $\mathbf{V} = \{v_1, v_2, \ldots, v_M\}$.
3. The state transition probability distribution $A = \{a_{ij}\}$, where
$$a_{ij} = P[q_{t+1} = j \mid q_t = i], \quad 1 \le i, \quad j \le N.$$
4. The observation probability distribution $B = \{b_j(k)\}$, where
$$b_j(k) = P[x_t = v_k \mid q_t = j], \quad 1 \le k \le M, \quad 1 \le j \le N.$$
5. The initial state distribution $\pi = \{\pi_i\}$ in which
$$\pi_i = P[q_1 = i], \quad 1 \le i \le N.$$

For each semantic context, the parameters of HMM are estimated from the Baum–Welch algorithm by giving sets of training data. The state number N is set at four, and the number of distinct observation symbols M is also four in our work. After the training process, parameters of two ergodic HMMs (for gunplay and car-chasing scenes, respectively) are estimated. These models elaborately characterize the densities of time-variant features and present the structures of sophisticated semantic contexts.

6.2 Semantic context detection

The detection process is conducted following the same idea as that of the audio event detection. To evaluate the likelihood of the observation sequence, $O = (o_1 o_2 \ldots o_T)$, given the model λ, the Forward algorithm is applied. Consider the forward $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(o_1 o_2 \ldots o_t, q_t = i \mid \lambda).$$

That is, the probability of the partial observation sequence, $o_1 o_2 \ldots o_T$, and state i at time t, given the model λ. The state sequence from time 1 to T is $q = (q_1 q_2 \ldots q_T)$. We can solve for $\alpha_t(i)$ inductively, as follows:

$$\text{Initialization:} \quad \alpha_1(i) = \pi_i b_i(o_1), \quad 1 \le i \le N. \tag{11}$$

$$\text{Induction:} \quad \alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_i(o_{t+1}), \quad 1 \le t \le T-1, \quad 1 \le j \le N. \tag{12}$$

$$\text{Termination:} \quad P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{13}$$

Through the Forward algorithm, the log-likelihood value that represents how likely a semantic context is to occur is calculated by taking the logarithm of $P(O \mid \lambda)$.

7 Performance evaluation

The training data for audio event modeling are taken from a professional sound-effects library, i.e., the SoundIdeas Library (http://www.sound-ideas.com). For each audio event, 100 short audio clips, each 3–10 s long, are selected. For semantic context modeling, given that there is no standard corpus for audio semantic contexts, the evaluation data are manually selected from Hollywood movies. Thirty movie clips, each 3–5 min long, from 10 different films are selected and labeled for each semantic context. In the experiments, fivefold cross validation is applied in semantic context model training. In each time, 24 sequences are randomly selected as the training dataset, while 6 other sequences are left for model testing. The average testing results are averaged and reported in Sect. 7.2.

Note that the criteria for 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 users' subjective judgments, no matter how many relevant audio events exist within the scene. On the other hand, the training data for audio event modeling consist of short audio segments that are exactly the audio events.

To determine the ground truth of semantic context, we define the boundary and label of a scene by watching the movies. Some may argue that we define the ground truth based on audiovisual streams, but we only exploit aural information for semantic context detection. However, in the movies we focused on, the aural information often presents more consistent characteristics in different action scenes and ease event/context modeling. If we include visual information in this framework, variations and uncertainties would increase greatly. For example, gunplay scenes may occur in a rainforest or a downtown street, day or night, where visual characteristics vary significantly. In contrast to this situation, aural information remains similar in different gunplay scenes. Furthermore, the mechanism for producing action movies is often straightforward. It is rare to have a gunplay concept without gunshot sounds or a car-chasing concept without engine sounds. Therefore, aural information is assumed to be more distinguishable than visual information for gunplay and car chase modeling.

Overall, the detection performance is evaluated at two levels: audio event detection and semantic context detection.

7.1 Performance of audio event detection

The effectiveness of audio event modeling affects the results of semantic context detection. In audio event detection, a "correct detection" is declared if a 1-s segment is evaluated as an audio event and its corresponding confidence score is larger than a predefined threshold. The length of an analysis unit, 1-s segment, is chosen for the tradeoff of the framework's efficiency and accuracy [17]. It could be set as other values to adapt to other kinds of audio events or other modeling methodologies.

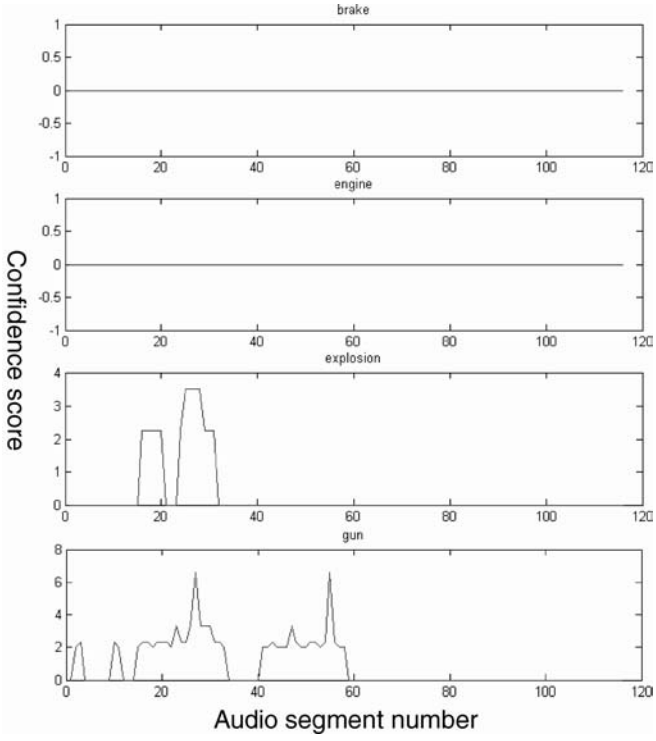

Fig. 7 Example of audio event detection

7.1.1 Performance of the proposed approach

Figure 7 illustrates a sample result of audio event detection. This audio clip is extracted from the movie "We Were Soldiers," with gunplay scene in the first 30 s. As shown in this figure, most gunshot and explosion sound effects are detected, while the confidence scores of the other audio events are zero. The result apparently indicates the clues of the appearance of a gunplay scene.

The overall detection performance is listed in Table 1. The average recall is over 70%, and the average precision is close to 85%. Although the detection accuracy is often sequence dependent and affected by confused audio effects, the reported performances sufficiently support effective semantic context modeling. In addition, different audio events have different evaluation results. Because the car-braking sounds are often very short (less than 1 s, which is the length of one basic audio unit used in our work) and are mixed with other environment sounds, the detection accuracy is particularly worse than the others. This situation is different from gunshot sounds because there is often a continuity

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

Table 2 Detection accuracy of different approaches

	[35]		[17]		[36]		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%

of gunshots (the sounds of a machine gun or successive handgun/rifle shots) in a gunplay scene.

The detection performance is more encouraging if we neglect the particular case in car-braking detection. For the 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 consider the detection results of car-braking sounds in car-chasing context modeling.

As shown in Table 1, the precision rate is generally larger than the recall rate in audio event detection. This indicates the high confidence of detection results, which is especially important in detecting a specific audio event in a chaotic acoustic condition caused by various sound effects. Furthermore, some misdetections will be disregarded by the process of pseudosemantic features, which integrally takes several overlapping audio segments into account. Hence, the reported results of audio event detection provide a promising basis for semantic context modeling.

7.1.2 Performance comparison

To compare the detection performance of various approaches, some institutes such as TREC Video Retrieval Evaluation[1] have developed corpora 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 one of the top performances in detecting various audio events, we refer to other works that focus on audio events in sports [35], TV shows [17], and movies [36].

Because not all referred works report precision and recall values, we only list the detection accuracy in Table 2 (precision) for fair comparison. In [35], 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 [17], the events "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 [36]. They also introduce 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% 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 serving as a robust basis for higher-level modeling.

[1] http://www-nlpir.nist.gov/projects/trecvid/

7.2 Performance of semantic context detection

In semantic context detection, the models based on GMM and HMM are evaluated. The metrics of recall and precision are calculated to show the detection performance, and the false alarm rate is also examined to show the robustness of these methods. The basic unit for calculating these metrics is the texture window, which covers the detection results of audio events in 5 s. In this section, the relationship between the event detection and semantic contexts is also investigated by comparing four test sequences.

To show how likely a texture window matches the targeted semantic concepts, we estimate the "semantic likelihood" to facilitate deciding whether a segment belongs to a specific concept. For the segments with feature values closer to the classical patterns, larger semantic likelihood values would be assigned.

7.2.1 GMM performance

The semantic likelihood value SL_i of texture window i is defined as:

$$SL_i = \text{mean}(s_1, s_2, \ldots, s_j), \tag{14a}$$

$$s_j = \frac{p(x_j \mid \lambda_j) - p(\delta_j \mid \lambda_j)}{p(\mu_j \mid \lambda_j) - p(\delta_j \mid \lambda_j)}, \tag{14b}$$

where x_j is the feature value from the jth audio event and is considered only when $x_j \in [\mu_j - \delta_j,\ \mu_j + \delta_j]$ (the range of detection defined in Sect. 5.3). The term λ_j denotes the jth Gaussian mixture, and μ_j and δ_j respectively denote its mean and standard deviation. The value s_j identifies how close the testing data are to the mean value of the jth mixture and is normalized to the range of [0,1] by Eq. 14b. The overall semantic likelihood value SL_i is the mean of the values from all relevant mixtures. This process gives different weights to different segments and facilitates possible applications in ranking the retrieval results and video summarization. Meanwhile, we consider the segments with semantic likelihood values larger than zero for calculating the precision and precision metrics. That is,

$$S = \{S_i \mid (SL_i > 0) \wedge (S_i \in G)\},$$
$$D = \{S_i \mid SL_i > 0\}, i = 1, 2, \ldots, N,$$

Table 3 Performance of semantic context detection by (a) GMM and (b) HMM in different test sequences

Semantic context		Recall (a)	Precision (a)	False alarm	Recall (b)	Precision (b)	False alarm
Gunplay	Clip 1	0.553	0.776	0.224	0.511	0.762	0.238
	Clip 2	0.220	0.542	0.458	0.668	0.754	0.246
	Clip 3	1.0	1.0	0	0.800	0.736	0.264
	Clip 4	0.800	0.414	0.586	0.778	0.390	0.610
	Clip 5	0.506	0.955	0.045	0.419	0.868	0.132
	Clip 6	0.533	0.828	0.172	0.498	0.850	0.150
	Average	0.602	0.752	0.248	0.612	0.727	0.273
Car-chasing	Clip 7	0.570	0.966	0.034	0.760	0.927	0.073
	Clip 8	0.192	0.700	0.300	0.863	0.624	0.376
	Clip 9	0.400	0.857	0.143	0.533	0.800	0.200
	Clip 10	0.692	0.346	0.654	0.769	0.303	0.697
	Clip 11	0.228	0.667	0.333	0.532	0.764	0.236
	Clip 12	0.351	0.971	0.029	0.723	0.971	0.029
	Average	0.406	0.751	0.249	0.697	0.731	0.269

$$\text{Precision} = \frac{|\ S\ |}{|\ D\ |}, \tag{15a}$$

and

$$\text{Recall} = \frac{|\ S\ |}{|\ G\ |}, \tag{15b}$$

where S denotes the set of correctly detected segments, D denotes the set of detected segments, and G denotes the set of ground truth of a test sequence. The element S_i is the ith testing segment (with length of a texture window) in a test sequence, and N is the number of segments in it.

The recall and precision rates of semantic context detection for selected test sequences are shown in Table 3a. We show six 5-min movie segments (selected from "We Were Soilders," "Windtalker," "The Recruit," and "Band of Brothers") for detecting gunplay and six 5-min movie segments (selected from "Terminator 3," "Ballistic: Ecks vs. Sever," "The Rock," and "2 Fast 2 Furious") for detecting car chases. The average recall for gunplay is 60.2%, and the precision is 75.2%. For car-chasing detection, the average recall and precision are 40.6% and 75.1%, respectively. The performances of detecting different semantic contexts are not identical because different semantic contexts posses different essential characteristics. The detection performance is also affected by the results of audio event detection. Therefore, the poorer performance in car-chasing scenes is due to the slightly weaker detection of engine and car-braking events. Figure 8 illustrates an example showing both the detection results for audio events and semantic contexts and demonstrates the correspondence between these two-level detections.

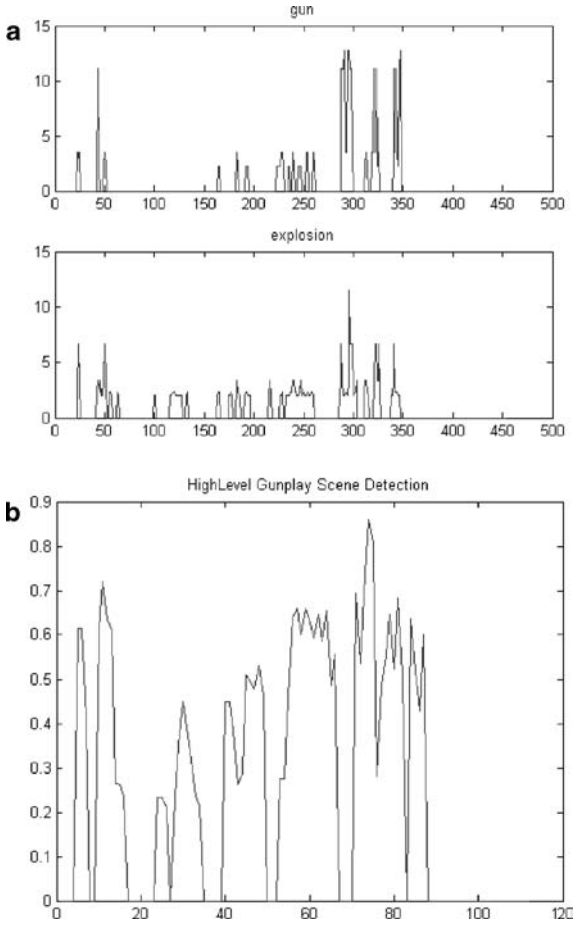

Fig. 8 Detection of audio events and semantic contexts. **a** Detection on relevant events for gunplay scenes. **b** Gunplay scenes

7.2.2 HMM Performance

The same dataset is applied to the HMM-based approach. Like the testing process conducted in GMM, every 5-s texture window is evaluated. The semantic likelihood value is defined as the logarithm of the likelihood $P(O \mid \lambda)$ obtained by the Forward algorithm:

$$SL_i = \log(P(O \mid \lambda)). \tag{16}$$

The texture window with the semantic likelihood value larger than a threshold is declared as a hit. That is,

$$S' = \{S_i \mid (SL_i > \varepsilon) \wedge (S_i \in G)\},$$
$$D' = \{S_i \mid SL_i > \varepsilon\}, i = 1, 2, \ldots, N,$$

Table 4 Performance evaluation using HMMs with different model sizes

Semantic Context		Recall	Precision	False Alarm
Gunplay	Four states, four mixtures	0.612	0.727	0.273
	Two states, two mixtures	0.612	0.727	0.273
Car-chasing	Four states, four mixtures	0.697	0.731	0.269
	Two states, two mixtures	0.530	0.722	0.278

$$\text{Precision} = \frac{|\ S'\ |}{|\ D'\ |}, \tag{17a}$$

and

$$\text{Recall} = \frac{|\ S'\ |}{|\ G\ |}, \tag{17b}$$

where ε is the threshold manually defined for filtering out the texture window with too small likelihood.

Table 3b shows the performance using the HMM-based approach for different test sequences. Based on the HMM-based approach, the average recall of gunplay is 61.2% and the average precision is 72.7%. For car-chasing scenes, the average recall is 69.7% and the average precision is 73.1%.

We also evaluate the detection performance with different HMM model sizes. The models with four states each with four Gaussian mixtures and two states each with two Gaussian mixtures are constructed and tested. From the experimental results, as shown in Table 4, the models with larger state numbers work slightly better than simple ones. This result conforms to the principles of probabilistic models and leads us to choose appropriate parameters for modeling. However, like the gunplay detection, almost the same performances are achieved with different parameters. It reveals that the pattern of gunplay scenes is relatively stationary and is simpler to model.

7.2.3 Discussion

According to the results in Table 3, although the detection results are sequence dependent, both approaches on average provide promising detection results either in precision or recall. In car-chasing detection, the recall rate of the HMM-based approach is generally superior to that of the GMM-based approach. Because the detection performance of relevant audio events varies more than that in gunplay scenes, it is believed that HMMs have greater capability for modeling variations.

In both approaches, some misdetections exist when multiple audio events are mixed. Moreover, some false alarms occur due to similar acoustic characteristics of different audio events. For example, in a violent scene, collisions are often misdetected as explosions, and some music segments played with bass drum or contrabass are often misdetected as explosions or engine sounds. This problem can be fixed by extracting more-representative audio features in audio event modeling or considering the cues from visual information. Moreover, by the emerging techniques of blind source separation [37], sounds from different sources may be separated so that different sound effects could be analyzed separately. This idea is reasonable because many sound effects in movies are imposed in the editing stage rather than in onsite recording.

We further investigate the relation between audio event detection and semantic context detection. As shown in Table 5, four sequences that have significant differences in audio event detection are deliberately selected to show the association between the two-stage detections. In both gunplay and car-chasing detection, the sequence with better audio event modeling/detection apparently has better performance in semantic context detection, especially in the precision and false alarm rates. Although this trend matches our expectation, this result further shows that semantic context modeling, with the help of pseudosemantic feature processing on texture windows, can provide acceptable performance even when the underlying event detection is not excellent.

To elaborately evaluate the performance of the proposed methods, we should compare them with other approaches. Although a few studies [19] have been done to achieve hierarchical semantic indexing, they emphasize visual information and do not report detailed experimental results. In [21], a few results are reported on detecting rocket-launch scenes based on aural (rocket engine and explosion) and visual cues. However, because there is no standard dataset and the performance is highly sequence dependent, we can

Table 5 Based on the HMM-base approach, examples of the relation between audio event detection and semantic context detection

	Audio event detection			Semantic context detection		
Test sequences		Recall	Precision	Recall	Precision	FA
"Ballistic: Ecks vs. Sever"	Engine	1	0.955	0.910	0.835	0.165
	Brake	0.455	0.556			
"The Rock"	Engine	1	0.862	0.863	0.624	0.376
	Brake	0.385	0.50			
"Band of Brother"	Gun	1	0.98	0.668	0.754	0.246
	Explosion	1	0.921			
"Tears of the Sun"	Gun	0.529	0.563	0.665	0.436	0.564
	Explosion	0.462	0.667			

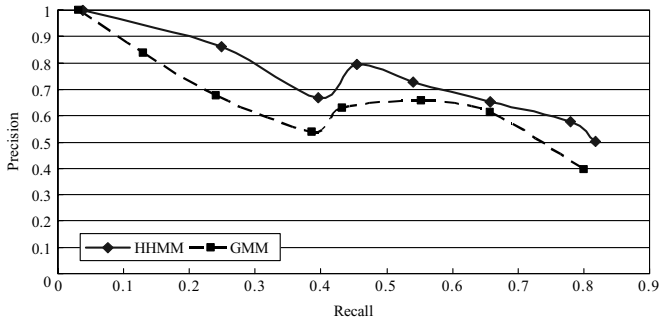

Fig. 9 Precision–recall curves of gunplay detection by using HMM and GMM approaches

hardly perform fair comparisons between different methods. Therefore, to show the effectiveness of the proposed approaches, we illustrate the overall precision–recall curve of gunplay detection in Fig. 9. This figure shows the promising performance and the differences between HMM-based and GMM-based approaches.

7.2.4 Generalization of the proposed framework

Although the proposed framework is only applied in the modeling of semantic concepts in action movies, it is assumed to be generalized to other types of videos, as long as the relationships between targeted semantic concepts and audiovisual characteristics follow two suggested prerequisites:

1. Concepts match between aural and visual information: for a given video clip, the concept drawn from aural and visual information should match consistently so that we can feel free to exploit extracted aural/visual features to characterize targeted semantic concepts.
2. Characteristic consistency between different sequences: aural or visual characteristics should be consistent between different video clips with the same concept. Otherwise, the high-variance information would burden modeling work. That is why the visual information is disregarded in modeling gunplay and car-chasing scenes within action movies, while both visual and aural information may be considered together in modeling important events within sports videos, for example.

Another encouraging idea to come out of this work is introducing late fusion for the results of individual classifiers. The merit of late fusion is that individual classifiers can be trained separately and added adaptively to the final metaclassifier. The proposed hierarchical framework can be generalized to other semantic concepts, as long as the modeled concepts have consistent visual/aural characteristics over different sequences in the same types of videos. For example, replacing the audio event models by visual object models, visual semantic context such as multispeaker conversation could be modeled by the same framework. We can even fuse the preliminary classification results (by careful design of pseudosemantic feature construction) from different modalities to construct a multimodal metaclassifier. Therefore, the proposed framework is attractive not only for its exciting scene detection in action movies, but for its generality, which facilitates building applications for different genres of videos, different types of media, and different granularities of usage.

8 Summary

We presented a hierarchical approach that bridges the gaps between low-level features and high-level semantics to facilitate semantic indexing and retrieval. The proposed framework hierarchically conducts modeling and detection at two levels: audio event and semantic context. The audio events that are highly related to some specific semantics are selected to provide important clues for modeling. After careful selection of audio features, HMMs are applied to model the characteristics of audio events. According to the production rules for action movies, gunshot and explosion sounds are adopted for detecting gunplay scenes, and car-braking and engine sounds are adopted for detecting car-chasing scenes.

At the semantic context level, the proposed fusion schemes that include pseudosemantic feature construction and probabilistic modeling take the results of audio event detection as a basis for characterizing semantic context. The pseudosemantic features present the interrelationship of several audio events and convey the variations of different types of semantic contexts. Based on the pseudosemantic features, two probabilistic models, i.e., GMM and HMM, are then exploited to model semantic contexts. GMM describes the distributions of different audio events, and HMM further presents the time duration density to model sophisticated contexts.

The experimental results demonstrate the effective performance of the fusion schemes and signify that the proposed framework draws a sketch for constructing an efficient semantic retrieval system. We investigate detection accuracy and robustness in detecting two semantic contexts and discuss the relations between audio events and semantics. Generally, the HMM-based approach achieves slight gains over the GMM-based method, as this trend is more apparent in the situation of poor audio event detection.

The proposed framework can be extended to other types of videos as different audio events and semantic contexts are modeled. It may be necessary to consider different combinations of events or include visual information according to the production rules of targeted videos, such as color distribution in sitcoms or particular spatial layout in games. One improvement to this framework may include elaborate feature selection from a candidate pool by developing an automatic feature induction mechanism. Moreover, some machine learning techniques, such as SVM, are also options for fusing pseudosemantic features and may engender new issues in video segmentation, discrimination, and indexing.

Acknowledgements The authors would like to thank the anonymous reviewers for the constructive comments that helped to greatly improve the quality of the manuscript. This work was partially supported by the CIET-NTU(MOE) and National Science Council of R.O.C. under Contracts NSC93-2622-E-002-033, NSC93-2752-E-002-006-PAE, NSC93-2213-E-002-006.

References

1. Yeo, B.L., Liu, B.: Rapid scene change detection on compressed video. IEEE Trans. Circuits Syst. Video Technol. **5**(6), 533–544 (1995)
2. Hanjalic, A.: Shot-boundary detection: unraveled and resolved? IEEE Trans. Circuits Syst. Video Technol. **12**(2), 90–105 (2002)
3. Li, Y., Zhong, T., Tretter, D.: An overview of video abstraction techniques. Technical Report, HPL-2001191, Hewlett-Packard, Palo Alto, CA (2001)
4. Pfeiffer, S., Lienhart, R., Fischer, S., Effelsberg, W.: Abstracting digital movies automatically. J. Vis. Commun. Image Represent. **7**(4), 345–353 (1996)
5. Dimitrova, N., Zhang, H.J., Shahraray, B., Huang, T.S., Zakhor, A.: Applications of video-content analysis and retrieval. IEEE Multimedia **9**(3), 42–55 (2002)
6. Lu, L., Zhang, H.J., Jiang, H.: Content analysis for audio classification and segmentation. IEEE Trans. Speech Audio Process. **10**(7), 504–516 (2002)
7. Zhang, T., Kuo, C.C.J.: Hierarchical system for content-based audio classification and retrieval. Proc. SPIE Multimedia Storage Archiv. Syst. III **3527**, 398–409 (1998)
8. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Trans. Speech Audio Process. **10**(5), 293–302 (2002)
9. Lu, L., Zhang, H.J.: Automatic extraction of music snippets. In: Proceedings of the ACM Multimedia Conference, pp. 140–147 (2003)
10. Moncrieff, S., Venkatesh, S., Dorai, C.: Horror film genre typing and scene labeling via audio analysis. In: Proceedings of the IEEE International Conference on Multimedia and Expo **2**, 193–196 (2003)
11. Liu, Z., Huang, J., Wang, Y.: Classification of TV programs based on audio information using hidden Markov model. In: Proceedings of the IEEE Signal Processing Society Workshop on Multimedia Signal Processing. 27–32 (1998)
12. Wang, Y., Liu, Z., Huang, J.C.: Multimedia content analysis using both audio and visual cues. IEEE Signal Process. Mag. **17**(6), pp. 12–36 (2000)
13. Ma, Y.F., Lu, L., Zhang, H.J., Li, M.: A user attention model for video summarization. In: Proceedings of ACM Multimedia, pp. 533–542 (2002)
14. Itti, L., Koch, C.: Computational modeling of visual attention. Nature Rev. Neurosci. **2**(3), 194–203 (2001)
15. Ho, C.C.: A study of effective techniques for user-centric video streaming. Ph.D. dissertation, National Taiwan University (2003)
16. Ouerhani, N., von Wartburg, R., Hugli, H., Muri, R.: Empirical validation of the saliency-based model of visual attention. Electron. Lett. Comput. Vis. Image Anal. **3**(1), 13–24 (2004)
17. Cai, R., Lu, L., Zhang, H.J., Cai, L.H.: Highlight sound effects detection in audio stream. In: Proceedings of the IEEE International Conference on Multimedia and Expo, **3**, 37–40 (2003)
18. Naphade, M.R., Kristjansson, T., Frey, B., Huang, T.S.: Probabilistic multimedia objects (multijects): a novel approach to video indexing and retrieval in multimedia system. In: Proceedings of the IEEE International Conference on Image Processing, **3**, 536–540 (1998)
19. Naphade, M.R., Huang, T.S.: Extracting semantics from audiovisual content: the final frontier in multimedia retrieval. IEEE Trans. Neural Netw. **13**(4), 793–810 (2002)
20. Naphade, M.R., Huang, T.S.: A probabilistic framework for semantic video indexing, filtering, and retrieval. IEEE Trans. Multimedia **3**(1), 141–151 (2001)
21. Adams, W.H., Iyengar, G., Lin, C.Y., Naphade, M.R., Neti, C., Nock, H.J., Smith, J.R.: Semantic indexing of multimedia content using visual, audio, and text cues. Eurasip J. Appl. Signal Process. **2003**(2), 170–185 (2003)
22. Kschischang, F.R., Frey, B.J.: Factor graphs and the sum-product algorithm. IEEE Trans. Inf. Theory **47**(2), 498–519 (2001)
23. Sethy, A., Narayanan, S.: Split-lexicon based hierarchical recognition of speech using syllable and word level acoustic units. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, **1**, 772–775 (2003)
24. Stolfo, S., Prodromidis, A., Tselepis, S., Lee, W., Fan, D., Chan, P.: JAM: Java agents for meta-learning over distributed databases. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 74–81 (1997)
25. Lin, W.-H., Hauptmann, A.: Meta-classification: Combining multimodal classifiers. In: Zaiane, O.R., Simoff, S., Djeraba, C. (eds.) Mining Multimedia and Complex Data, pp. 217–231. Springer, Berlin Heidelberg New York (2003)
26. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
27. Dimitrova, N.: Context and memory in multimedia content analysis. IEEE Multimedia **11**(3), 7–11 (2004)
28. Li, S.Z.: Content-based classification and retrieval of audio using the nearest feature line method. IEEE Trans. Speech Audio Process. **8**(5), 619–625 (2000)
29. Bow, S.T.: Pattern Recognition and Image Preprocessing. Marcel Dekker, New York (2002)
30. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, New York (2001)
31. Reynolds, D.A., Rose, R.C.: Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Trans. Speech Audio Process. **3**(1), 72–83 (1995)
32. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. **39**, 1–38 (1977)
33. Dorai, C., Venkatesh, S.: Media Computing: Computational Media Aesthetics. Kluwer, Dordrecht (2002)
34. Zettl, H.: Sight Sound Motion: Applied Media Aesthetics, 3rd edn. Wadsworth, Belmont, CA (1999)
35. Wang, J., Xu, C., Chng, E., Tian, Q.: Sports highlight detection from keyword Sequences using HMM. In: Proceedings of the IEEE International Conference on Multimedia and Expo (2004)
36. Naphade, M.R., Garg, A., Huang, T.S.: Audio-visual event detection using duration dependent input output Markov models. In: Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 39–43 (2001)
37. Cardoso, J.F.: Blind signal separation: satistical principles. Proc. IEEE **9**(10), 2009–2025 (1998)