A Flexible Framework for Key Audio Effects Detection and Auditory Context Inference

Rui Cai, Lie Lu, *Member, IEEE*, Alan Hanjalic, *Member, IEEE*, Hong-Jiang Zhang, *Fellow, IEEE*, and Lian-Hong Cai, *Member, IEEE*

***Abstract*—Key audio effects are those special effects that play critical roles in human's perception of an auditory context in audiovisual materials. Based on key audio effects, high-level semantic inference can be carried out to facilitate various content-based analysis applications, such as highlight extraction and video summarization. In this paper, a flexible framework is proposed for key audio effect detection in a continuous audio stream, as well as for the semantic inference of an auditory context. In the proposed framework, key audio effects and the background sounds are comprehensively modeled with hidden Markov models, and a *Grammar Network* is proposed to connect various models to fully explore the transitions among them. Moreover, a set of new spectral features are employed to improve the representation of each audio effect and the discrimination among various effects. The framework is convenient to add or remove target audio effects in various applications. Based on the obtained key effect sequence, a Bayesian network-based approach is proposed to further discover the high-level semantics of an auditory context by integrating prior knowledge and statistical learning. Evaluations on 12 h of audio data indicate that the proposed framework can achieve satisfying results, both on key audio effect detection and auditory context inference.**

***Index Terms*—Audio content analysis, auditory context, Bayesian network, flexible framework, grammar network, key audio effect, multi-background model.**

I. Introduction

THERE are various audio effects in daily life and multimedia materials, such as *car-horn*, *bell-ringing*, and *laughter*. These special effects play important roles in humans' understanding of the high-level semantics of the auditory context. For instance, *car-horn* and *noisy-speech* are often associated with a scene of *street*; and the mixture of *siren*, *car-racing*, and *car-crash* may indicate a *pursuit* in progress. Therefore, detection and recognition of these key audio effects in a continuous stream are important and helpful in many applications, such as context-aware computing [1], [2] and video content parsing, including highlight extraction [3]–[7] and video summarization [8].

Manuscript received September 13, 2004; revised April 4, 2005. This work was performed at Microsoft Research Asia. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Michael Davies.

R. Cai and L.-H. Cai are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: cairui01@mails.tsinghua.edu.cn; clh-dcs@tsinghua.edu.cn).

L. Lu and H.-J. Zhang are with the Microsoft Research Asia, Beijing 100080, China (e-mail: llu@microsoft.com; hjzhang@microsoft.com).

A. Hanjalic is with the Department of Mediamatics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: a.hanjalic@ewi.tudelft.nl).

Digital Object Identifier 10.1109/TSA.2005.857575

Most relevant research focuses on general audio segmentation and classification [9]–[12]. In these works, the audio stream is coarsely divided and classified into a few basic classes, such as *speech*, *music*, *environmental sounds*, and *silence*. However, these basic audio categories can only provide limited semantic information and cannot meet the requirements of current applications.

A number of recent works have paid attention to exploring more concrete key audio effects with high-level semantics in various applications, such as highlight detection and context identification. For example, in sports video analysis [4], [6], [7], [13], highlight events are detected based on special audio effects like *applause*, *cheer*, *ball-hit*, and *whistling*; and in film indexing [5], [14], sounds like *car-racing*, *siren*, *gun-shot*, and *explosion* are used to identify violent scenes in action movies.

In most of the above works, a sliding window of a given length is usually utilized to presegment an audio stream, and then each window is used as the basic unit to model and recognize audio effects. For example, in [4], support vector machines (SVMs) are built to detect key effects, such as *whistling* and *ball-hit*, based on audio frames of 20 ms; and in [7], the input stream is first segmented into units of 0.5 s with 0.125 s overlapping, then each unit is classified into *applause*, *cheer*, *music*, *speech*, and *speech with music*. However, in most cases, such a sliding window could not cover one complete audio effect, since the durations of different audio effects usually vary largely and the start time of an audio effect in a continuous stream is unknown *a priori*. Thus, a sliding window may chop an audio effect into several parts, or contain several kinds of audio effects. Based on such a window, it is difficult to obtain comprehensive models to satisfy the performance requirements of audio effect detection, especially when the effects have distinct characteristics in their temporal evolution processes.

To solve the problems introduced by sliding windows, a new framework with a hierarchical structure is proposed in this paper. In this framework, we do not segment the audio stream in advance but take it as a whole. The scheme essentially is a hierarchical probabilistic model, where an HMM is first built for each key audio effect based on complete sound samples, and then a high-level probabilistic model is used to connect these individual models. Thus, for a given input stream, the optimal key effect sequence is searched through the candidate paths with the *Viterbi* algorithm, and the location and duration of each key effect in the stream is determined simultaneously by tracing back the optimal path.

Besides the problems associated with sliding windows, there are still some other issues which should be taken into account when detecting key audio effects in a continuous stream.

1) In audio streams, the target key effects are usually sparsely distributed, and there are many nontarget sounds which should be rejected in detection. Some previous works do not consider this case, while others use thresholds to discard the sounds with low confidence [3], [14]. However, the threshold setting becomes troublesome for a large number of key effects.
2) There are some relationships among key audio effects. For example, some key effects, such as *applause* and *laughter*, are likely to happen together, while others are not. However, previous works usually identify each window independently but seldom consider the transition relationships between different key effects.
3) With more key audio effects explored, the features should provide more sufficient representations of the key effects, as well as adequate discrimination among various key effects.

The proposed framework also provides solutions to the above issues. First, comprehensive background models are established to cover all nontarget sounds, as opposed to the target key effects. Thus, the nontarget sounds would be detected as background sounds and excluded from the target key effect sequence. A similar definition of background sounds is also used in previous works. However these works usually focus on special domains, such as sports video in [7], and do not provide a comprehensive formulation of background modeling, as our approach does. To solve the second issue, in our framework, a *Grammar Network* is proposed to organize all the sound models, in which the transition probabilities among various key effects and background sounds are taken into account in finding the optimal key effect sequence. Finally, a set of new audio spectral features are proposed in this framework to improve the description of each key effect and to explore more key effects.

Furthermore, based on the key effects obtained, the framework is further designed to discover the high-level semantics of related auditory contexts. Key audio effects have been proven to be efficient in bridging the gap between low-level audio features and high-level semantics. Most of previous related works utilize heuristic rule-based approaches [4], [13] or statistical classification [5], [14]. For example, in [4], heuristic rules such as "if double whistling, then Foul or Offside" were used to infer the events in soccer games; while in [5] and [14], SVMs and Gaussian mixture models (GMMs) were employed to statistically learn the relationships between key effects and higher semantics. In general, heuristic rules can represent prior knowledge well for semantic inference but it is usually laborious to set up a proper rule set in the complicated applications. Meanwhile, statistical classification can automatically learn the complex relationships between key effects and higher semantics; however, its performance relies highly on the training set. To combine the advantages of these approaches and to compensate for their limitations, our framework utilizes a Bayesian network to detect the semantics at different levels for various applications.

The rest of this paper is organized as follows. An overview of the proposed framework is described in Section II. In Section III, audio features used in the framework are discussed. Section IV presents the algorithms on key audio effect detection, including modeling of the key effects and the background sounds, and the construction of the *Grammar Network*. In Section V, a Bayesian network-based approach is proposed for high-level semantic discovery, and is compared with two other methods. Experiments and discussions are presented in Section VI, and conclusions are given in Section VII.

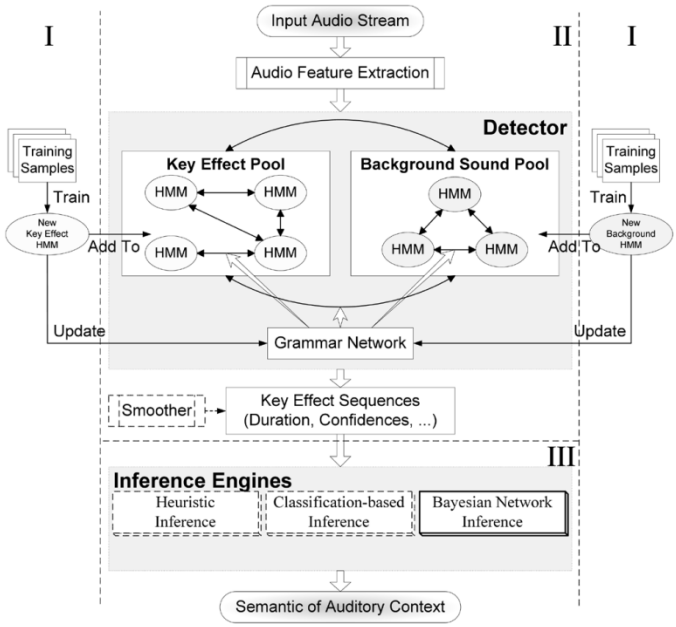


Fig. 1. Framework flowchart for key audio effect detection and auditory context inference. It is mainly composed of three parts: (I) sound modeling, (II) key audio effect detection, and (III) semantic extraction.

II. Framework Overview

The system flowchart of the proposed framework is illustrated in Fig. 1. It is mainly composed of three steps: sound modeling, key audio effect detection, and semantic extraction.

In the framework, an HMM is used to model each target key audio effect and these models ensemble a *Key Effect Pool*. Correspondingly, a *Background Sound Pool* is used to enclose all the nontarget sounds. Since the background in key effect detection is quite complex, the background sounds are divided into a few basic categories, each of which is also modeled with an HMM. The *Grammar Network* is then constructed to organize all the HMMs and to represent the transition probabilities among various sounds. It actually constructs a higher-level probabilistic model for key audio effect detection.

To detect key effects, the acoustic features of each frame, including temporal features and spectral features, are passed through the above hierarchical structure. The optimal sequence of key effects and background sounds are then found with the *Viterbi* algorithm. Post-processing approaches, such as the *Smoother*, could optionally be further applied to improve the detection performance. For example, isolated key effects with very short durations could be deleted.

At last, based on the detected key effect sequence, high-level semantics are detected with inference engines. Different inference engines can be easily integrated into the framework. In our implementation, a Bayesian network-based method is proposed and compared with conventional heuristic and classification-based approaches. Moreover, based on the obtained se-

mantic context, we could further revise the key effect sequence according to prior knowledge.

It should be noted that the proposed framework is flexible in practice. With this framework, it is convenient to add or remove key effects to satisfy new requirements, and the only part that needs to be updated is the transition probability setting in the *Grammar Network*. In contrast, in most of the previous works, such as the decision tree-based system in [5] and the hierarchical SVM-based system in [10], the whole system must be retrained when a new audio effect is added in.

III. Feature Extraction

Feature extraction is one of the most fundamental and important issues in key audio effect detection. In this section, we present the features used in the proposed framework. Besides a number of widely used features, two new spectral features are also proposed to provide a more complete description of key effects and to discern more key effects. In this framework, all these features are concatenated as a feature vector for each audio frame.

A. Basic Audio Features

Many audio features have been proposed in previous works on content-based audio analysis [9]–[12], [15] and have been proved effective in characterizing various key audio effects. Grounded on these works, in our framework, both temporal features and spectral features are extracted for each audio frame. The temporal features consist of short-time energy (STE) and zero-crossing rate (ZCR), and the spectral features consist of band energy ratios (BERs), brightness, bandwidth, and Mel-frequency cepstral coefficients (MFCCs).

In the temporal domain, STE provides a good representation of the amplitude or the loudness of the audio effect sounds and ZCR gives a rough estimation of the frequency content in an audio signal. In the spectral domain, the characteristics of spectral energy distribution, as BER describes, are widely used to discriminate between different audio effects [12]. In our experiments, the spectral domain is equally divided into eight subbands in Mel-scale and the energy in each subband is then normalized by the whole spectrum energy. Brightness and bandwidth are related to the first- and second-order statistics of the spectrogram, respectively. They roughly measure the timbre quality of a sound. MFCCs are subband energy features in Mel-scale, which give a more accurate simulation of the human auditory system. As suggested in [9], eight-order MFCCs are used in the experiments. A more detailed implementation of these features can be found in our previous works [10].

B. New Spectral Features

In the spectral domain, there are two other important characteristics associated with human identification of sounds [16]: 1whether there is a prominent partial at a certain spectral subband; and 2) whether the sound is harmonic. For example, one distinct difference between *cheer* and *laughter* is that *laughter* usually has prominent harmonic partials but *cheer* does not. However, the above basic features are incapable of describing these characteristics. Brightness and bandwidth can only measure the global energy center and the deviation of the whole spectrum. Although BER and MFCC calculate the average energy in subbands, it is still hard to specify whether there exist salient components in some subbands.

Based on our previous works on audio representation [17], [18], two new spectral features, *subband spectral flux* and *Harmonicity Prominence*, are utilized as supplements of those basic features. *Subband spectral flux* is used to measure whether there are salient frequency components in the subband, and *Harmonicity Prominence* estimate the harmonic degree of a sound.

In order to remove the impact induced by the energy variation in different time slices, the spectrum is converted to the decibel scale and is constrained to unit L_2-norm, as suggested in [19]

$$\hat{x} = \frac{10 \log_{10} x}{||10 \log_{10} x||} \tag{1}$$

where x is the original spectral coefficient vector generated by fast Fourier transform, and $\hat{x}$ is the new spectral vector with unit L_2-norm in decibel scale.

1) Subband Spectral Flux: Subband spectral flux estimates the existence of prominent partials by accumulating the variation between adjacent frequencies in each subband. For subbands containing salient components, the flux value should be large; otherwise, it is small. In practice, the spectrum is divided into eight subbands equally in Mel-scale, with 50% overlap between each other. The flux S_f is defined as

$$S_f(i) = \frac{1}{H_i - L_i} \sum_{j=L_i}^{H_i - 1} |\hat{x}(j+1) - \hat{x}(j)| \tag{2}$$

where L_i and H_i are the low and high boundaries of the ith subband, respectively; $S_f(i)$ indicates the corresponding existence probability of salient frequency components.

2) Harmonicity Prominence: Considering the property of an ideally harmonic sound (assuming there is only one dominant fundamental frequency), that is, its *full* spectrum energy is *highly concentrated* and *precisely located* at those predicted harmonic positions which are multiples of the fundamental frequency f_0, the harmonicity measurement can be designed according to the following three factors: 1) the energy ratio between the detected harmonics and the whole spectrum; 2) the deviation between the detected harmonics and predicted positions; and 3) the concentration degree of the harmonic energy. Based on the above factors, the *Harmonicity Prominence* consists of three components and is defined as

$$H_p = \frac{\sum_{n=1}^{N} E^{(n)} \left(\frac{1-|B_r^{(n)} - f_n|}{0.5 f_0}\right) \left(\frac{1 - B_w^{(n)}}{B}\right)}{E} \tag{3}$$

where f_n is the nth predicted harmonic position and is defined by

$$f_n = n f_0 \sqrt{1 + \beta(n^2 - 1)} \tag{4}$$

where β is the inharmonicity modification factor and is set as 0.0005 following the discussions in [20]. In (3), $E^{(n)}$ is the energy of the detected nth harmonic contour in the range of $[f_n - f_0/2, f_n + f_0/2]$ and the denominator E is the total spectrum energy. The ratio between $E^{(n)}$ and E describes the first factor listed above. $B_r^{(n)}$ and $B_w^{(n)}$ are the *brightness* and

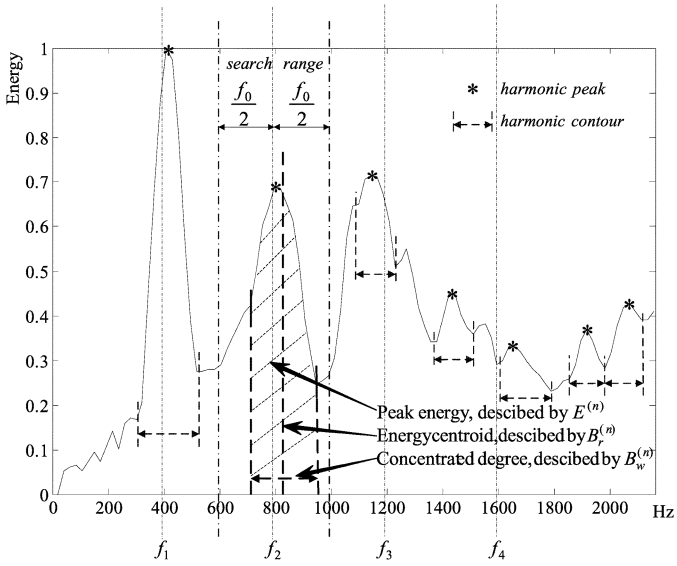


Fig. 2. Definition of the *Harmonicity Prominence*. The horizontal axis represents the frequency, and the vertical axis denotes the energy. The harmonic contour is the segment between the adjacent valleys of a harmonic peak. Based on the harmonic contour, three factors, that is, the peak energy, the energy centroid (brightness), and the concentrated degree (bandwidth), are computed to estimate the *Harmonicity Prominence*, as illustrated at the second harmonic position in this example.

bandwidth [12] of the nth harmonic contour, respectively. The brightness $B_r^{(n)}$ is used here, instead of the detected harmonic peak, in order to estimate a more accurate frequency center. The bandwidth $B_w^{(n)}$ describes the concentration degree of the nth harmonic. It is normalized by a constant B, which is defined as the bandwidth of an instance where the energy is uniformly distributed in the search range. Thus, the component $(1 - |B_r^{(n)} - f_n|/0.5f_0)$ in the numerator of (3) measures the second factor, while the component $(1 - B_w^{(n)}/B)$ approximates the third factor. A clear illustration of the definition of the *Harmonicity Prominence* is shown in Fig. 2.

In our implementation, f_0 is estimated with the autocorrelation based approach; and only the first N (which is set as 4) harmonic partials are considered in the computation, since only these harmonic partials are sufficiently prominent in most cases. Furthermore, following our previous work [18], in a case where the fundamental frequency cannot be precisely predicted, f_0 is varied in a predefined range and the corresponding *Harmonicity Prominences* are calculated, in which the maximum is chosen as the value of H_p for the frame. For a sound without pitch, H_p is set to zero.

IV. Key Audio Effect Modeling and Detection

In this section, we discuss some critical issues in key audio effect detection, including the modeling of the key effects and the background sounds, as well as the construction of the *Grammar Network*.

A. Key Audio Effect Modeling

Most audio effects usually have distinct characteristics along their evolution processes. HMMs provide a natural and flexible way for modeling time-varying process [21], and have been proven to be effective for audio effect modeling in many previous works [3], [13], [14], [22]. In our proposed framework, HMMs are also selected for key audio effect modeling. The main issue that should be addressed with HMM is the parameter selection, including: 1) the optimal model size (the number of states); 2) the number of Gaussian mixtures of each state; and 3) the topology of the model.

In model size selection, we should balance the number of hidden states in the HMMs and the computational complexity in the training and recognition processes. In general, sufficient states are required to describe those significant and representative acoustical characteristics in signals; however, when the number of states increases, the computational complexity grows dramatically and more training samples are required. Unlike speech modeling, in which basic units such as phonemes could be referenced to specify the number of states, generic audio effects lack such basic units, and make the choice of the state numbers difficult. In our approach, a clustering-based method similar to [11] is utilized to estimate the reasonable state number (model size) for each key effect. In the clustering, an improved unsupervised k-means algorithm is employed and after convergence the final cluster number is taken as the model size. More details on the implementation can be found in [11] and [22].

The number of Gaussian mixtures per state is usually determined by experiments [3], [11]. Based on audio effect modeling experiences in previous research works, as well as the size of training data, we adopt 32 Gaussian mixtures for each state in the HMMs. This number is larger than those used in previous works in order that the models have sufficient discriminative capabilities to identify many more key effects in our framework.

As for the topology, the most popular HMM topology is left-to-right or fully connected. The left-to-right structure only permits transitions between adjacent states; while the fully connected structure allows transitions between any two states in the model. In the proposed framework, they are used to model key effects with different properties according to the following rules.

- For key effects with obvious characteristics in their progress phases, such as *car-crash* and *explosion*, the left-to-right structure is adopted.
- For key effects without distinct evolution phases, such as *applause* and *cheer*, the fully connected structure is applied.

B. Background Modeling

To facilitate the detection of the sparsely distributed key audio effects in audio streams and to reject those nontarget sounds, a background model is proposed to describe all the nontarget sounds. Background modeling is extremely critical to the detection performance. Many sounds in the background would be misclassified as key audio effects without sufficient description of the background.

A straightforward approach to background modeling is to build a "huge" HMM, and train it with as many samples as possible. However, background sounds are very complex and their feature vectors are widely scattered in the feature space, so that both the number of states and the Gaussian mixtures per state of such a HMM must be particular large to give a universal representation of all background sounds. Meanwhile, the

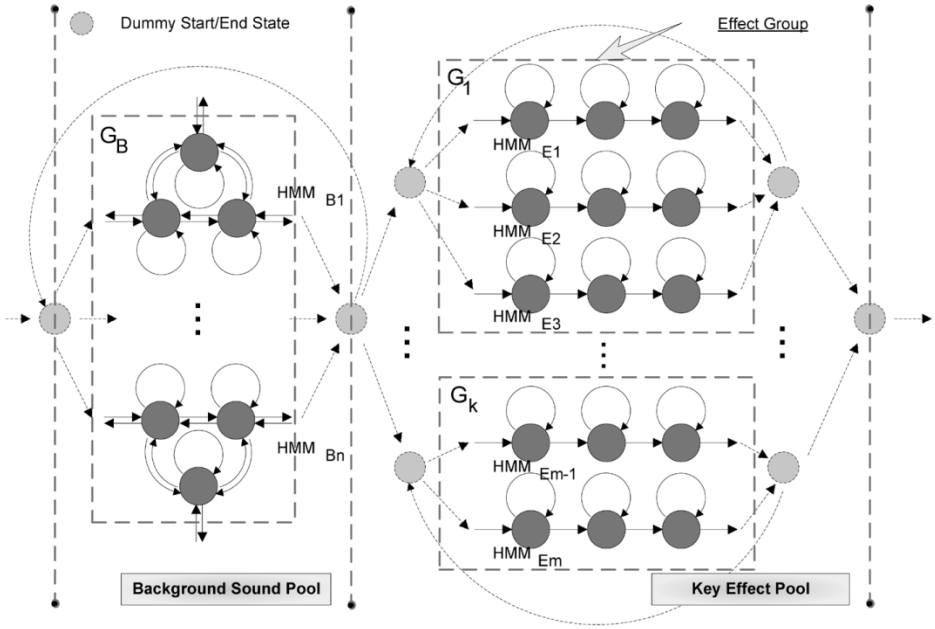


Fig. 3. Illustration of the *Grammar Network* with *Effect Groups*, where G_k is the kth *Effect Group* and G_B is the *Background Sound Pool*. For convenience, all the key audio effect models are presented as three-state left-to-right HMMs, and all the background models are denoted as three-state fully connected HMMs. The dummy start and end states are used to link models, and make the structure more clear.

training time of the model would become prohibitively long, and it would be hard to reach the convergence point.

In order to solve the above issue, we model the background with a set of subsets of general audio classes. It is noted that background sounds in most practical applications can be further classified into a few primary categories, such as *speech*, *music*, and other *noise*. Thus, if we can train background models from all these respective subsets, the training data will be relatively concentrated, and the training time will be reduced. Another advantage of building several subset models is that it could further provide useful information in high-level semantic inference of the auditory context. For example, *music* is usually used in the background of movies, and *speech* is the most dominant component in talk shows. Actually, according to the requirements of applications, the background can be divided and modeled with more categories.

In our current system, three background models are built with fully connected HMMs for *speech*, *music*, and other *noise* respectively. Here, *noise* includes all the background sounds except for *speech* and *music*. To provide comprehensive descriptions of the background, more states and more Gaussian mixtures in each state should be used in modeling. Considering the balance between the representation capability and the computational complexity, in our approach the number of states is chosen as 10 and the number of mixtures per state is experimentally chosen as 128. The influences of the parameter selection are fully investigated in the experiments.

C. Grammar Network Construction

Similar to the language model in speech processing, a *Grammar Network* is proposed to organize all the above HMMs for continuous recognition. Two models are connected in the *Grammar Network*, if the corresponding sounds have the possibility to occur subsequently, both within and between the *Key Effect Pool* and the *Background Sound Pool*. For each connection, the corresponding transition probability is set and taken into account in finding the optimal effect sequence from the input stream.

A simple approach to construct a *Grammar Network* is using a fully connected network and then learning the transition probabilities statistically. However, it is usually difficult to collect sufficient training data in practice. Furthermore, it is noted that some key audio effects usually occur together, while others seldom happen subsequently. For instance, *gun-shot* often happens with *explosion*, but rarely takes place with *laughter*. This indicates that it is not necessary to fully connect all the sound models. Therefore, an alternative way is introduced in our framework, in which an *Effect Group* is proposed to denote a set of key effects which usually occur together, as the G_1–G_k shown in Fig. 3. We assume: 1) only key effects in the same *Effect Group* can happen subsequently, and there should be background sounds between any two key effects in different groups; and 2) one key effect can belong to several *Effect Groups*. Based on these assumptions, the *Grammar Network* is constructed, as shown in Fig. 3.

To avoid the training problem and enhance the detection flexibility, in this work, the transition probabilities between key effects are assumed equal and set with some heuristic rules, based on the above principles. Suppose the set Φ_{E_i} is an ensemble of all the sounds which a given audio effect E_i can transit to (or occur subsequently) in the audio stream, there is

$$\Phi_{E_i} = \left(\bigcup_{k|E_i \in G_k} G_k \right) \bigcup G_B \tag{5}$$

where G_k is the kth *Effect Group* that E_i belongs to (it implies that an audio effect can belong to multiple *Effect Groups*) and G_B denotes the ensemble of the *Background Sound Pool*. Thus, the transition probability from E_i to a given sound s can be

intuitively set as (6), assuming the equal transition probabilities from one effect to another

$$P(E_i, s) = \begin{cases} \frac{l}{|\Phi_{E_i}|}, & s \in \Phi_{E_i} \\ 0, & s \notin \Phi_{E_i}. \end{cases} \quad (6)$$

Similarly, for a background sound B_n, as it can occur with (or be followed by) all the sounds, its transition probability to a given sound s is

$$P(B_n, s) = \frac{1}{|(\bigcup_k G_k) \bigcup G_B|} = \frac{1}{M+N} \quad (7)$$

where M is the total number of key effect models in the *Key Effect Pool* and N is the number of background models in the *Background Sound Pool*.

The above transition probability setting can achieve similar detection performance as statistical learning, as shown in the experiments. Moreover, this scheme keeps the advantage of framework flexibility in various applications. That is, when new target effects are added in or removed from either the *Key Effect Pool* or the *Background Sound Pool*, only the *Effect Groups* need to be redefined, without any extra system retraining.

D. Key Audio Effect Detection

To this end, we have built a hierarchical probabilistic structure by connecting key effects and background models with a *Grammar Network*. The *Viterbi* algorithm is then utilized to choose the optimal state sequence from the continuous audio stream, as

$$S_{opt} = \arg\max_S Pr(S|\mathcal{M}, \mathcal{O}) \quad (8)$$

where S is the candidate state sequence, $\mathcal{M}$ represents the hierarchical structure, and $\mathcal{O}$ is the observation vector sequence which is extracted from each frame using the features presented in Section III.

Thus, for each audio frame, the corresponding state and log-probability are obtained. A complete audio effect or background sound can be determined by merging adjacent frames belonging to the same sound model. The corresponding confidence of a sound is measured by averaging all the frame log-probabilities of this sound. Besides confidence, the start position and duration of each sound is also recorded for further semantic inference in our framework.

V. Auditory Context Inference

Based on the obtained key effect sequence, we further extend the framework to detect high-level semantics in an audio stream. Key audio effects are efficient ways to bridge the gap between low-level features and higher semantics. In many previous works, semantics detection is based on presegmented audio clips [2], [5], [13]. In the case of our continuous streams, the *auditory contexts*, each of which contains several neighboring key effects, are first used to locate those potential segments with consistent semantic meaning; and then the semantic inference is performed on these auditory contexts. Here, the semantic concept of an auditory context means an auditory scene or an event happens in this segment. Thus, the key audio effects provide important cues for these events, and facilitate the corresponding semantic inference.

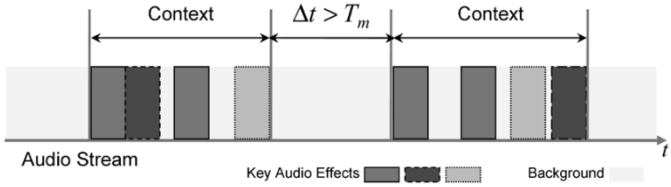


Fig. 4. Examples of the auditory context in an audio stream.

Fig. 4 gives an illustration of the auditory contexts. Two adjacent key effects are assumed to be in the same context if their time interval is short enough, since humans usually only keep a short memory of a past occurrence. As illustrated in Fig. 4, a new context is started if the time interval between two key effects is larger than a predefined threshold T_m. The threshold T_m can be determined based on the upper limit of human memory to perceive a consecutive scene and is set to 16 s in this framework, following the previous work in [23].

To infer high-level semantics from key effects, most of previous works utilize rule-based approaches [4], [13] or statistical classification [5], [14]. In this section, we briefly describe these two methods, and propose a Bayesian network-based solution to solve the disadvantages of those previous methods.

A. Heuristic Inference

A heuristic approach is the most natural and common method to discover semantics based on the obtained key effects. According to human experience and experts' knowledge, a set of rules is found to map key effects to high-level semantics. For instance, the appearance of *laughter* may indicate a *humor* context. Although heuristic inference is straightforward and easily applied in practice, it is laborious to find a proper rule set if the situation is complex. Some rules may be in conflict with others, and some cases may not be well-covered. People are used to designing rules from a positive view but ignoring those negative instances, thus many false alarms are introduced although high recall can be achieved. For example, people like to create rules like "a scene with *laughter* is a *humor* scene," but seldom design rules such as "a scene with *explosion* is NOT a *humor* scene." Actually, in practice, it is impossible to enumerate all those exceptions in designing the rules. Moreover, there are often lots of thresholds defined in heuristic rules. The setting of thresholds becomes another difficult problem.

B. Classification-Based Inference

Classification-based methods provide solutions from the view of statistical learning. Using these means, relationships between the key effects in an auditory context and the corresponding semantics are automatically learned based on training data, either using generative models such as GMM or discriminative models such as SVM [24]. However, like most machine learning approaches, the inference performance relies highly on the completeness and the size of the training samples. Without sufficient data, a positive instance not included in the training set will usually be misclassified. Thus these approaches are usually prone to high precision but low recall. Furthermore, it is inconvenient to

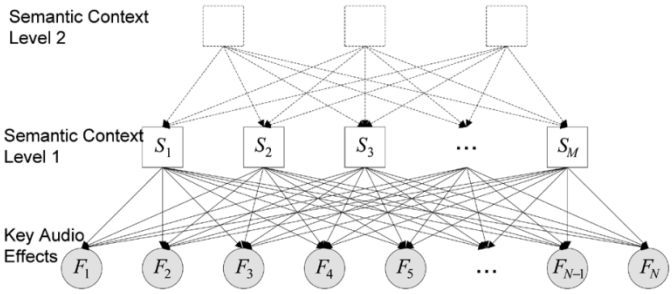


Fig. 5. Example Bayesian network for auditory context inference. Arcs are drawn from cause to effect. Here, we adopt the convention of representing discrete variables as squares while continuous variables as circles, and representing observed variables as shaded while hidden variables as clear.

combine prior knowledge into the classification process in these algorithms.

C. Bayesian Network Inference

To integrate the advantages of heuristic and statistical learning methods, a Bayesian network-based approach is implemented in this framework for high-level semantic discovery. A Bayesian network [25] is a directed acyclic graphical model that encodes probabilistic relationships among nodes which denote random variables related to semantic concepts. A Bayesian network can handle situations where some data entries are missing, as well as avoid the overfitting of training data [25]. Thus, it weakens the influence from unbalanced training samples. Furthermore, a Bayesian network can also integrate prior knowledge by specifying its graphic structure.

Fig. 5 illustrates the graphic topology of an example Bayesian network with three layers. Nodes in the bottom layer are the observed key effects; nodes in the second layer denote high-level semantic categories such as scenes; and those in the top layer denote much higher semantic meanings. In Fig. 5, the nodes in adjacent layers can be fully connected, or partially connected based on the prior knowledge of the application domain. For instance, if it is known *a priori* that some key effects have no relationships with one semantic category, the arcs from that category node to those key effect nodes could be removed. In comparison, a Bayesian network with a manually specified graphic structure utilizes human knowledge in representing the conditional dependencies among nodes, thus it can describe some cases which are not covered in the training samples.

Besides the network topology, we also need to choose the value type and the *conditional probability distribution* (CPD) of each node in the graph. In our approach, the nodes in the upper layers are assumed to be discrete binaries which represent the presence or absence of a corresponding context category; and the nodes in the bottom layer are continuous-valued with Gaussian distribution, as

$$p(F_i|\mathbf{pa}_i) \sim N(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), \qquad (1 \le i \le N) \tag{9}$$

where F_i is the two-dimensional (2-D) observation vector of the ith key effect, and is composed of its normalized duration (d_i) and confidence (c_i) in the context scope, as

$$F_i = (d_i, c_i) \tag{10}$$

TABLE I
INFORMATION OF THE EXPERIMENTAL AUDIO DATA

Movie/TV Title	Category	Duration
Saving Private Ryan	war	2:41:41
Enemy at the Gates	war	2:11:08
Swordfish	action	1:39:17
The Rock	action	2:16:35
3^{rd} Rock from the Sun	situation comedy	0:30:05
Hollywood Squares	TV shows	0:25:43
59^{th} Annual Golden Globe Awards	TV shows	2:14:50

and $\mathbf{pa}_i$ denotes a possible assignment of values to the parent nodes of F_i; $\boldsymbol{\mu_i}$ and $\boldsymbol{\Sigma_i}$ are the mean and covariance of the corresponding Gaussian distribution respectively.

In the training phase, all these CPDs are uniformly initialized and then updated by maximum likelihood estimation using the expectation–maximization (EM) algorithm. In the inference process, the junction tree algorithm [26] is used to calculate the occurrence probability of each semantic category. Thus, given information on the key effects in the context, we could infer the semantics in each layer based on the posterior probabilities. In current experiments, an auditory context in the second level could be classified into the cth semantic category with the maximum marginal posterior probability

$$c = \arg\max_j Pr(S_j|\mathbf{F}), \qquad (1 \le j \le M)$$

where

$$\mathbf{F} = \{F_1, F_2, \ldots, F_N\}. \tag{11}$$

With this scheme, human knowledge and machine learning are effectively combined in the semantic inference. That is, the graphic topology of the network can be designed according to the prior knowledge of the application domains, and the optimized model parameters are estimated by statistical learning.

VI. IMPLEMENTATION AND EVALUATION

In this section, we present the detailed implementations and evaluations of the proposed framework, both on key audio effect detection and on semantic inference of the auditory context.

A. Database Information

The evaluations of the proposed framework have been performed on audio tracks of about 12 h in length extracted from various types of video, including movies and entertainment television shows. Theses videos have relatively abundant audio effects, and the contained audio effects are usually distinct enough to be perceived. Detailed information on these audio tracks is listed in Table I. All audio streams are in 16 kHz, 16-bit, and mono channel format, and are divided into frames of 30 ms with 50% overlapping for feature extraction.

B. Key Audio Effect Detection

To evaluate the performance of the proposed framework with a large number of targets, ten key audio effects are taken into account in the experiments. They are: *applause*, *car-racing*, *cheer*, *car-crash*, *explosion*, *gun-shot*, *helicopter*, *laughter*, *plane*, and

TABLE II
NUMBER OF HMM STATES OF KEY EFFECTS IN EXPERIMENTS

Key Audio Effects	States	Key Audio Effects	States
applause	8	*gun-shot*	10
car-racing	9	*helicopter*	7
car-crash	9	*laughter*	8
cheer	5	*plane*	10
explosion	9	*siren*	11

TABLE III
EFFECT GROUPS FOR THE *GRAMMAR NETWORK* CONSTRUCTION

No.	Key Audio Effects
1	*cheer*, *laughter*, *applause*
2	*car-racing*, *car-crash*, *siren*, *explosion*, *gun-shot*, *helicopter*, *plane*

siren. These audio effects are selected based on two criteria. First, the effects should be distinctly perceived and frequently occur in audio tracks from same video category. For example, *gun-shot* happens in almost all the action movies. Second, the effects should be typical enough to characterize the related semantic events we address in Section VI-B1, that is, *excitement*, *humor*, *pursuit*, *fight*, and *air-attack*.

The training samples are collected from the web and extracted from audio tracks of several other television programs and movies. In total, there are around 100 samples for each key audio effect, and 5 h for background sounds which are then divided into three basic categories: *music*, *speech*, and *noise*. As ground truth, the positions and durations of all the target key audio effects are manually labeled for the audio tracks in the database. For an audio segment in which several key effects are mixed, the dominant one is selected as the label.

In key audio effect modeling, the unsupervised k-means clustering referred in Section IV-A is first performed on the training sets to estimate the HMM states of each key audio effect model. The corresponding results are listed in Table II. It should be noted that more states are used here than were used in our previous work [3], in order to cover the large variety of training samples in our current implementation. Then, a traditional EM algorithm is used to train an HMM for each key effect. Although discriminative training of HMMs can improve the performance of classification and recognition, both for speech and generic audio, it is not suitable in our framework. This is because all the HMMs have to be retrained when new key effects are added into the system with the scheme of discriminative training. This will lead to the loss of the framework flexibility in practice, which is one important advantage of our proposed framework.

To establish the *Grammar Network*, two *Effect Groups* are defined based on the above ten key audio effects. One is for those effects related to the entertainment scenes, and the other is for sounds in violent scenes. The two groups seldom happen together in general television shows and movies. The *Effect Groups* and the related key audio effects are shown in Table III.

In key effect detection, all audio tracks in the database are tested, and the start time, duration and confidence of the detected key effects are recorded for further semantic inference. Then, each frame in a stream is labeled (recognized) according to the corresponding key effects or background sounds, based on which *recall* and *precision* are computed. We use a frame-by-frame approach to evaluate the performance of the framework, since frame is the basic unit in audio effect detection from a continuous stream. Thus, for a given key effect, its precision and recall are defined as

$$\text{precision} = \frac{n_{tp}}{n_p} \quad \text{recall} = \frac{n_{tp}}{n_t} \tag{12}$$

where n_{tp} denotes the number of correctly detected frames, n_p represents the number of all the frames recognized as the target effect, and n_t is the total frame number of the target key effect in the ground truth.

TABLE IV
FOUR SYSTEMS FOR THE PERFORMANCE EVALUATION

System	Features	Backgrounds	*Grammar Network*
A	Basic[a]	Mono-background	Manual[d]
B	Basic and New[b]	Mono-background	Manual
C	Basic and New	Multi-background[c]	Manual
D	Basic and New	Multi-background	Statistical[e]

[a] Basic audio features are presented in Section III-A.
[b] New spectral features are described in Section III-B.
[c] Including three background models: *music*, *speech*, and *noise*.
[d] Transition probabilities are specified by (5)-(7) in Section IV-C.
[e] Transition probabilities are set by statistical learning on the training set.

To fully investigate the performances of the proposed framework on key audio effect detection, a series of experiments are implemented. In the following experiments, we first evaluate the effectiveness of the proposed spectral features and the multi-background model strategy, as well as the reasonableness of the transition probability setting in constructing the *Grammar Network*. These evaluations are described in the following Sections VI-B1–B3, respectively. Moreover, to investigate the influences of the number of Gaussian mixtures in each state of the background model, various mixture numbers, from 8 to 256, are tested in all the above evaluations, and the optimal mixture number is selected in Section VI-B4. Then, the detailed system performance is presented in Section VI-B5. Finally, in Section VI-B6, the comparison between the proposed framework and the traditional sliding window-based approach is reported. The scalability of the proposed framework is also evaluated in Section VI-B7.

To carry out the above evaluations, several preliminary systems have been built and compared based on the proposed framework, as illustrated in Table IV. System $\boldsymbol{A}$ is a baseline which uses basic audio features and a mono-background model, while system $\boldsymbol{B}$ is used to evaluate the new spectral features by comparing them with $\boldsymbol{A}$. System $\boldsymbol{C}$ adopts the multi-background strategy based on $\boldsymbol{B}$; and $\boldsymbol{D}$ is similar to $\boldsymbol{C}$ except that the transition probabilities of the *Grammar Network* are statistically learned.

1) Evaluation of New Spectral Features: In order to evaluate the effectiveness of the proposed new spectral features, the performance of system $\boldsymbol{A}$ and $\boldsymbol{B}$ have been compared, as shown in Fig. 6. It is noted that the recalls of system $\boldsymbol{B}$ are generally increased while the precision is almost kept the same as system $\boldsymbol{A}$, whatever the number of Gaussian mixtures is in the background models. On average, the recall is improved significantly, by around 14.5%, and the precision drops less than 1%. It indicates that with the proposed new spectral features, more target

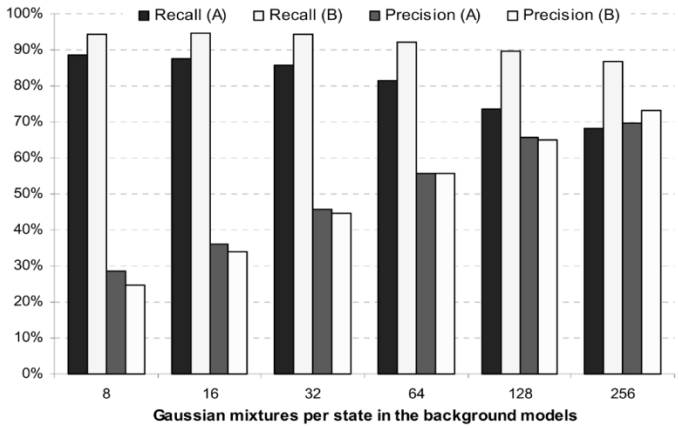


Fig. 6. Comparison of the performances of system $\boldsymbol{A}$ and $\boldsymbol{B}$, to evaluate the effectiveness of the proposed new spectral features.

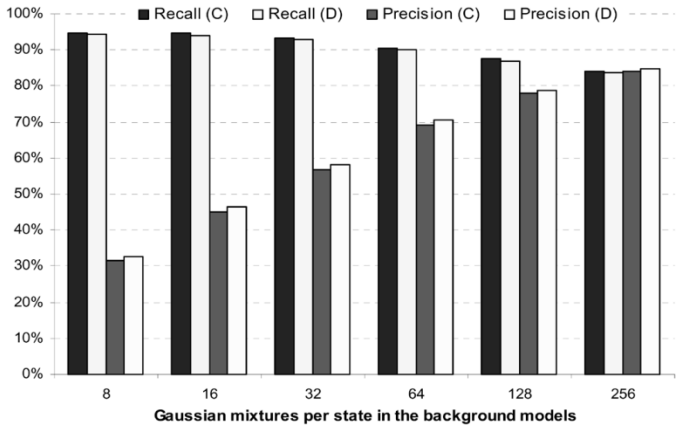


Fig. 8. Comparison of the performances of system $\boldsymbol{C}$ and $\boldsymbol{D}$, to evaluate the reasonableness of the transition probability setting in the *Grammar Network*.

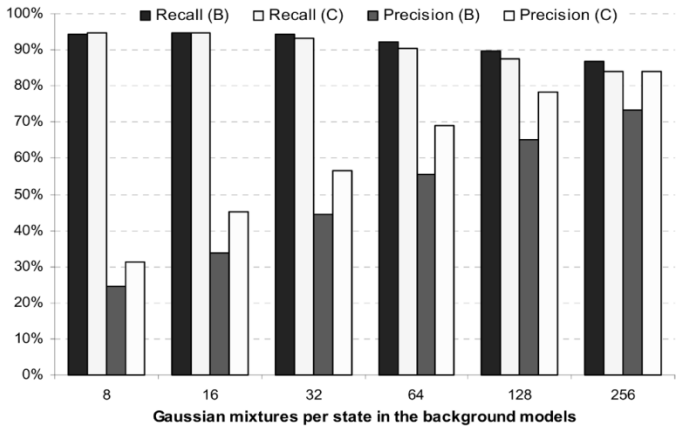


Fig. 7. Comparison of the performances of system $\boldsymbol{B}$ and $\boldsymbol{C}$, to evaluate the effectiveness of the multi-background model strategy.

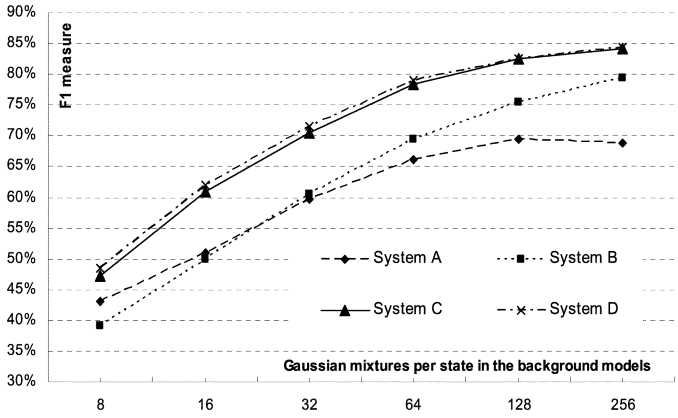


Fig. 9. $F1$ measure of the performances for the four evaluation systems with different numbers of Gaussian mixtures per state in the background models.

effects are accurately discriminated from background sounds and few additional false alarms are introduced.

2) Evaluation of the Multi-Background Model Strategy: Although new spectral features increase the recalls significantly, the precisions in system $\boldsymbol{B}$ are still not as high as required. The lower precisions indicate many background sounds are misclassified as key effects due to the noncomprehensive background model. To give a more comprehensive description of the backgrounds, a multi-background model strategy is proposed in Section IV-B. Fig. 7 illustrates the effectiveness of the multi-background model strategy by comparing system $\boldsymbol{C}$ and $\boldsymbol{B}$. In comparison, system $\boldsymbol{C}$ can markedly improve the precision, by about 24.3% on average. This indicates that the multiple background models can significantly help in discriminating the background sounds from the key effects. Meanwhile, in system $\boldsymbol{C}$, the recalls of the key effects drop slightly by around 1.3% averagely. This indicates that the improved background models introduce few key effects which are falsely recognized as the background sounds, and most of the target key effects are still correctly detected as in system $\boldsymbol{B}$.

3) Transition Probability Setting: As mentioned in Section IV-C, the transition probabilities are ideally statistically learned from training data. However, due to the difficulty in training data collection and in order to keep the flexibility of the framework, in our approach, the transition probabilities are set by *Effect Groups* with (5)–(7). This experiment is designed to show the reasonableness of such a setting, by comparing systems $\boldsymbol{D}$ and $\boldsymbol{C}$. The comparison results are illustrated in Fig. 8, in which system $\boldsymbol{C}$ can achieve very similar results with $\boldsymbol{D}$, either on precision or on recall. This demonstrates that the proposed heuristic settings of the transition probabilities are feasible in practice, and they can effectively approximate the real transition probabilities among all the key effects and the background sounds in a continuous audio stream. This conclusion is critical to guarantee the flexibility property of the whole framework.

4) Selection of Gaussian Mixtures in Background Models: Figs. 6–8 also shows the performance variance with various Gaussian mixture numbers in each state of the background models. In general, the recall drops while the precision increases when the number of Gaussian mixtures increases. In order to balance the recall and precision and to achieve the best overall performance, we should choose a proper mixture number in the background modeling. Here, we adopt the $F1$-measure to characterize the overall performance of all the systems, which is a harmonic mean of recall and precision, as

$$F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{13}$$

The $F1$-measures of all four systems with different Gaussian mixtures in each state of the background models are shown in

TABLE V
CONFUSION MATRIX OF THE SYSTEM WITH ADDITIONAL SPECTRAL FEATURES AND MULTI-BACKGROUND MODELS

Actual Classes	Detection Results												
	applause	*cheer*	*laughter*	*car-racing*	*car-crash*	*explosion*	*gun-shot*	*helicopter*	*plane*	*siren*	*speech*	*music*	*noise*
applause	**0.977**	0.023											
cheer	0.004	**0.972**	0.021								0.003		
laughter	0.001		**0.991**								0.008		
car-racing				**0.865**	0.020	0.011	0.011	0.001		0.003	0.007	0.033	0.050
car-crash				0.009	**0.982**	0.002	0.001	0.001		0.001		0.002	0.002
explosion				0.006	0.009	**0.800**	0.056		0.006		0.003	0.031	0.089
gun-shot				0.008	0.004	0.020	**0.805**				0.021	0.052	0.090
helicopter				0.001	0.001	0.002		**0.987**		0.002	0.001	0.005	0.001
plane					0.007	0.014			**0.928**		0.003	0.032	0.016
siren								0.001		**0.991**	0.004	0.003	0.001
speech	0.001	0.008	0.006				0.002				**0.877**	0.026	0.080
music		0.002	0.004	0.005	0.004	0.003	0.008	0.002			0.020	**0.891**	0.061
noise		0.001	0.004	0.005	0.002	0.006	0.018	0.002	0.001	0.001	0.064	0.043	**0.853**

Fig. 9. It is noted that the increase of $F1$-measures become non-prominent after the mixture number is larger than 128. The $F1$ of system $\boldsymbol{A}$ even decreases after that. Considering the balance between computational burden and relatively high performance, we choose 128 as the mixture number in the background models in the following experiments.

5) Detailed Performance of the Final System: Based on the above evaluations, system $\boldsymbol{C}$ has been selected as the final system, integrating new spectral features, *Effect Groups*, and multi-background models with 128 Gaussian mixtures in each state. Overall, the system achieves recall of about 88% and precision of near 80%. The detailed confusion matrix of the key effect detection is listed in Table V. It shows that high accuracy (the numbers in bold) is achieved for each target effect in the experiments. The average accuracy of the ten key effects is higher than 92%. The accuracy of *gun-shot* and *explosion* are somewhat low, since they are easily misclassified into each other and are usually covered by background *music* and *noise*. From the table, it is noted that the key effects in different *Effect Groups* are seldom misclassified with each other. This again indicates the efficiency of the *Grammar Network* proposed in this work.

Furthermore, to show how accurately the target effects can be located in the audio stream, the boundary shift is measured between the detected effect and the true effect. Fig. 10 illustrates the histogram of the boundary shift duration of all the detected key effects. From Fig. 10, it can be seen that nearly 62% of the detected effect boundaries are less than 0.3 s away from the true boundaries, and 79.6% are less than 1 s. Only around 14% are outside of 2 s. The results indicate that our approach has an acceptable temporal resolution, comparing with the average length of key effects which is around 3.7 s in our experiments. Actually, for audio tracks from movies and televisions, it is usually hard for people to precisely distinguish the sound boundaries with a high resolution, since in most cases, there are no clear boundaries between key effects and backgrounds, which may be caused by mixed sounds and audio transitions such as fade-in and fade-out.

6) Comparison With Sliding Window-Based Approach: The proposed framework is also compared with a traditional sliding window-based approach to key effect detection, similar to the one used in our previous work [3]. In the compared approach, the length of the sliding window is 1 s with 50% overlapping.

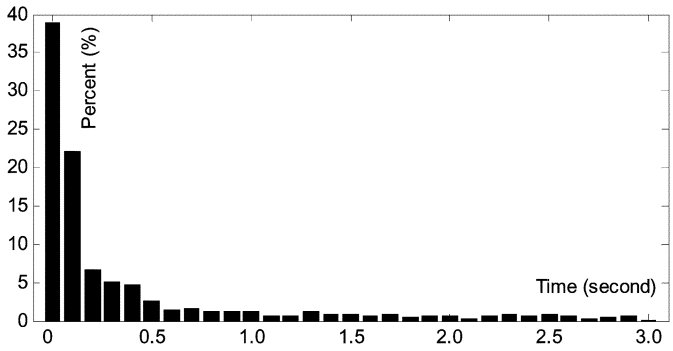


Fig. 10. Histogram of the shift duration from the true effect boundary.

The low-level features and HMM parameters used in the compared approach are the same as those used in system $\boldsymbol{C}$.

Table VI lists the detailed performance comparison between the sliding window-based approach and our approach (system $\boldsymbol{C}$). From Table VI, it is noted that system $\boldsymbol{C}$ achieves much better performance on all the key effects, either on precision or on recall. System $\boldsymbol{C}$ improves the average precision and recall significantly, by around 19.24% and 8.68% respectively, which indicates that it can distinguish different key effects and background sounds better. On the contrary, the sliding window-based approach usually works worse due to two reasons. First, a sliding window usually cannot cover a complete audio effect and may contain several kinds of sounds. Thus it is likely to introduce inaccurate recognitions. Second, the co-occurrence relationships among various key effects are ignored in the sliding window-based approach. Thus some background segments are usually misclassified into key effects, which leads to the low precision of these key effects. In comparison, system $\boldsymbol{C}$ improves the performance by modeling such relationships with the transition probabilities described in Section IV-C.

7) Evaluation of the System Flexibility: Finally, we have preliminarily investigated the scalability of the proposed framework; that is, the performance variance is measured regarding the increase of the number of target effects. The scalability here requires that, with increasing numbers of target effects in system: 1) the detection performances should be stable; and 2) the *Grammar Network* should correctly approximate the real transition probabilities among sounds.

TABLE VI
COMPARISON OF PERFORMANCE BETWEEN THE PROPOSED FRAMEWORK AND THE SLIDING WINDOW-BASED APPROACH ON KEY EFFECT DETECTION

Category	Sliding Window-based		System *C*	
	Recall	Precision	Recall	Precision
applause	0.732	0.471	0.977	0.914
cheer	0.921	0.638	0.972	0.658
laughter	0.903	0.661	0.991	0.711
car-racing	0.819	0.739	0.865	0.856
car-crash	0.892	0.548	0.982	0.694
explosion	0.743	0.630	0.800	0.760
gun-shot	0.725	0.610	0.805	0.768
helicopter	0.929	0.706	0.987	0.850
plane	0.877	0.765	0.928	0.890
siren	0.938	0.817	0.991	0.931
Average	**0.806**	**0.655**	**0.876**	**0.781**
F1-Measure	**0.723**		**0.826**	

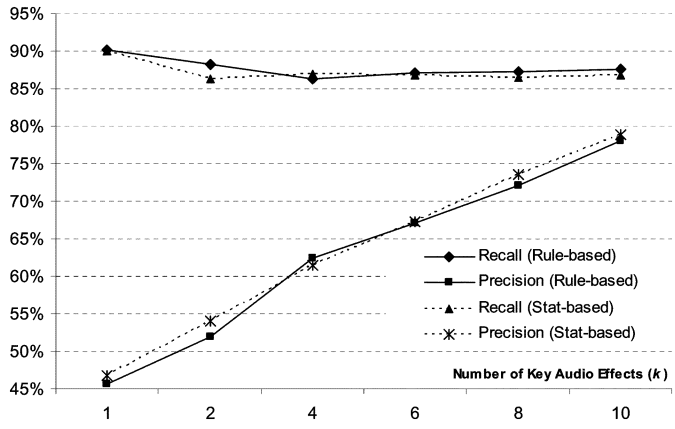


Fig. 11. Evaluation of the scalability of the proposed framework.

In the experiments, $k(= 1, 2, 4, 6, 8, 10)$ key effects are randomly selected from all of the ten effects and then detected by the proposed system. The above process is repeated 20 times and the average performance is taken as the performance of a given k. Moreover, for each k, the transition probability setting schemes are compared between the heuristic rules as (5)–(7) and the statistical setting based on the training data to measure their difference under a different number of target effects. The corresponding average precision and recall are shown in Fig. 11.

First, the performance of the system with rule-based transition probabilities is evaluated. From Fig. 11, it is noted that the recall drops slightly as the number of target effects increases, while the precision is somewhat low when k is small and increases when more key effects are taken into account. This phenomenon can be explained by the following information. When there is not sufficient knowledge (that is, few key effects are modeled) in the system, it is easy for the system to recognize some confusing audio effects as the target effects. For example, we find in experiments that some *laughter* segments are easily misclassified as *cheer* in detection when there is only a *cheer* model in the system. Thus, the number of false alarm is large when k is too small so that the precision is a little bit low. However, when more audio effect models are available, that is, there is sufficient knowledge to discriminate confusing effects, the false alarms decrease and precision increases. In the above example, when models of both *laughter* and *cheer* are available, few *laughter* segments are recognized as *cheer*. Correspondingly, the recall will inevitably decrease since more misclassifications are introduced when more target effects are considered. However, our proposed framework can keep recall stability above 85% with various key effect numbers in detection. The almost stable recall and increasing precision indicates the excellent scalability of our system.

Furthermore, by comparing the different transition probability setting schemes, it can be seen that they achieved very similar performance under different numbers of target effects. This again indicates the proposed heuristic settings are feasible on our current target scale, which can satisfy the requirements of most current applications. However, it is noted that when the amount of target effects increases significantly (for example, 100), the proposed rules might be too simple to describe the complex relationships among various sounds. We will address this issue in our future work.

TABLE VII
RULES FOR THE HEURISTIC INFERENCE IN THE EXPERIMENT

No.	Conditions	Category
1	(*cheer*) or (*cheer* and *applause*)	*excitement*
2	(*laughter*) or (*laughter* and *applause*)	*humor*
3	(*car-crash* or *car-racing*) and (*siren* or *helicopter* or *gun-shot* or *explosion*)	*pursuit*
4	(*gun-shot*) and (*explosion*)	*fight*
5	(*plane*) and (*explosion*)	*air-attack*
6	.....	*others*

C. Auditory Context Inference

Based on the detected key effects in the above section, five relevant semantic categories of the auditory context are further detected: *excitement*, *humor*, *pursuit*, *fight*, and *air-attack*. The *excitement* category mainly consists of *cheer* and *applause*, while *laughter* is the dominant component in the *humor* context. *Pursuit* usually happens in action movies and is associated with *car-crash* and *car-racing*, and sometimes with *siren*, *helicopter*, *gun-shot*, and *explosions* are sometimes included. *Fight* and *air-attack* often occur in war movies. Scenes of *fight* mostly contain *gun-shot* and *explosion*; and *air-attack* includes *plane* and *explosion*.

The semantic category of each auditory context is manually labeled for all of the audio tracks in the database. In detection, a context segment is considered to be correctly recognized if it has the same semantics and more than 50% temporal overlapping with the ground truth. The following experiments consist of two parts. First, based on the key effect detection results, the semantic inference of the auditory context is carried out and compared using heuristic-based, classification-based, and Bayesian network-based approaches respectively, as shown in Sections VI-C1–C3. Second, to show the advantages of the key effect-based approach in semantic inference, we also implemented a low-level feature-based system for comparison, The performance comparison is reported in Section VI-C4.

1) Heuristic Inference: The heuristic rules used in the experiments are listed in Table VII. These conditions are executed

TABLE VIII
PERFORMANCE OF SEMANTIC INFERENCE BY USING DIFFERENT APPROACHES

Category	Heuristic Rule-based		SVM-based		Bayesian network-based			
					Fully connected structure		Manually specified structure	
	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
excitement	0.9394	0.8857	0.7742	0.8889	0.8710	0.9310	0.9355	0.9355
humor	0.9744	0.8636	0.7105	0.9643	0.8421	0.9412	0.9474	0.9231
pursuit	0.9556	0.6880	0.8652	0.8021	0.8876	0.7745	0.9101	0.8265
fight	0.6780	0.5970	0.6854	0.7625	0.7303	0.7647	0.8090	0.7273
air-attack	0.9583	1.0000	0.8400	1.0000	0.8800	0.9167	1.0000	0.8929
Average	**0.8487**	**0.7147**	**0.7721**	**0.8333**	**0.8272**	**0.8212**	**0.8934**	**0.8237**
F1-Measure	**0.7760**		**0.8015**		**0.8242**		**0.8571**	

consecutively and each auditory context is then classified into a related category.

2) Classification-Based Inference: During implementation, half of the context segments in the database are used to train one-against-all SVMs with probabilistic outputs [27] for the semantic categories. The 2-D feature vector [as in (10)] of each key effect or background segment is concatenated to form the feature vector of the corresponding auditory context. For those effects and background sounds which do not appear, their values are set to zero. In recognition, the context is validated against all the SVMs, and then classified into the semantic category with the maximum probabilistic output.

3) Bayesian Network Inference: A two-layer Bayesian network is used in the experiment. There are 13 nodes in the bottom layer to represent the ten key audio effects and three background sound categories in their auditory context; and there are five nodes in the top layer, denoting the five predefined semantic categories. For comparison, the graphic structure is defined using two strategies: 1) fully connected; that is, for each node in the bottom layer, all nodes in the top layer are its parents; 2) manually specify the causal relationships based on the above definitions of the five context categories; that is, the children of a category node only include those key effect nodes which have relationships with this semantic category. For example, the node of *excitement* only connects with the nodes of *cheer* and *applause* in the bottom layer. The model parameters are uniformly initialized. Half of the labeled context segments in the database are selected as training samples to update the CPDs of each node in the model. In our implementation, the model training and semantic inference are implemented with the *Bayes Net Toolbox* [28].

Detailed comparison results are shown in Table VIII, where the precision and recall of each semantic category is listed, and $F1$-measure is again used to evaluate the overall performance of each system.

From Table VIII, it can be seen that the heuristic rule-based method usually obtains high recall but low precision, since the rules are usually set from a positive view, so that the negative samples are often misclassified. In contrast, the SVM-based approach obtains high precision but low recall, since it usually could not detect the instances not included in the training set. Comparatively, the Bayesian network can handle situations where some data entries are missing, so that it has both high recall and high precision and shows a better overall performance than the above two approaches. As shown in Table VIII, the Bayesian network with a fully connected graphic structure offers similar precision to SVM while improving recall by 7.2%, and obtains recall similar to the rule-based approach while improving the precision by about 10.7%.

TABLE IX
PERFORMANCE COMPARISON OF SEMANTIC INFERENCE BETWEEN FROM KEY EFFECTS AND FROM LOW-LEVEL FEATURES

Category	Key-audio effects-based (with manually specified Bayesian network structure)		Low-level feature-based	
	Recall	Precision	Recall	Precision
excitement	0.9355	0.9355	0.7813	0.3571
humor	0.9474	0.9231	0.9474	0.3025
pursuit	0.9101	0.8265	0.7045	0.5391
fight	0.8090	0.7273	0.7356	0.6400
air-attack	1.0000	0.8929	0.5185	0.7368
Average	**0.8934**	**0.8237**	**0.7390**	**0.4752**
F1-Measure	**0.8571**		**0.5784**	

Furthermore, after prior knowledge is utilized to manually specify the structure of the Bayesian network, the performance is further improved. Its average recall reaches about 90%, which is much better than others, while the precision stays similar. This indicates that prior knowledge can be integrated into a Bayesian Network to increase recall and overall performance by decreasing false alarms.

4) Semantic Inference From Low-Level Features: To show the advantages of the key audio effect-based semantic inference, we compared it with a system which infers semantics from low-level features directly. The implementation of the low-level feature-based approach is similar to the system presented in [2], in which the mappings to high-level semantics are modeled using GMMs with 128 mixtures. Half of the context segments in the database are used for training and another half are used for testing. Table IX lists the performance comparison between these two approaches.

From Table IX, it is noted that the overall performance of the key audio effect-based approach (with a manually specified Bayesian network structure) is much better than that of the low-level feature-based approach, especially on the average precision, which is improved by nearly 73%. This indicates that a great deal of background (nontarget) segments is misclassified into the target context categories with the low-level feature-based approach. This is because the context segments usually share a large similarity in the background sounds (*speech* and *music*). Thus, when learned from the low-level features directly, the GMMs of these semantic categories lack the capability to discriminate from the background segments.

Actually, compared with Table VIII, it can be seen that the performance of a low-level feature-based approach is much worse than any of the key effect-based approaches shown in Table VIII. The comparisons again indicate that key audio effects are efficient in bridging the gap between low-level acoustic features and high-level semantics.

VII. Conclusion

In this paper, a flexible framework has been proposed for key audio effect detection in a continuous stream and for semantic inference of related auditory context. In this framework, two new spectral features have been introduced to improve the representation of key effects, and multi-background models are used to achieve more comprehensive characterization of the environment background. All the models are connected by the *Grammar Network* which represents the transition probabilities among various sounds. With this framework, an optimal key effect sequence is obtained directly from the continuous audio stream without the need of sliding window-based presegmentation. Based on the obtained key effect sequence, a Bayesian network-based inference has been proposed to combine the advantages of both prior knowledge and statistical learning in mapping key effects to the high-level semantics of the auditory context. Experimental evaluations have shown that the framework can achieve very satisfying results, both on key audio effect detection and semantic inference of the auditory context.

There is still room to improve the proposed framework. For example, a duration model for each key effect can be built and integrated into the detection process; and N-best result paths can be recorded for the input audio stream to provide more information for further semantic inference. Moreover, the scale of target key effects in current system is still limited. Although such a target scale can meet the requirements of most current applications, some potential problems may arise when the scale increases significantly. For example, the generalization ability of the heuristic setting of the transition probabilities, and the discriminative ability of the HMMs, may both be challenged when more key effects are considered in detection. This is also a direction of our future works.

References

[1] A. Eronen, J. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context awareness—Acoustic modeling and perceptual evaluation," in *Proc. IEEE ICASSP*, vol. 5, Hong Kong, Apr. 2003, pp. 529–532.

[2] V. Peltonen, J. Tuomi, A. P. Klapuri, J. Huopaniemi, and T. Sorsa, "Computational auditory scene recognition," in *Proc. IEEE ICASSP*, vol. 2, Orlando, FL, May 2002, pp. 1941–1944.

[3] R. Cai, L. Lu, H.-J. Zhang, and L.-H. Cai, "Highlight sound effects detection in audio stream," in *Proc. IEEE ICME*, vol. 3, Baltimore, MD, Jul. 2003, pp. 37–40.

[4] M. Xu, N. Maddage, C.-S. Xu, M. Kankanhalli, and Q. Tian, "Creating audio keywords for event detection in soccer video," in *Proc. IEEE ICME*, vol. 2, Baltimore, MD, Jul. 2003, pp. 281–284.

[5] S. Moncrieff, C. Dorai, and S. Venkatesh, "Detecting indexical signs in film audio for scene interpretation," in *Proc. IEEE ICME*, Tokyo, Japan, Aug. 2001, pp. 1192–1195.

[6] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," in *Proc. 8th ACM Multimedia Conf.*, Los Angeles, CA, Oct. 2000, pp. 105–115.

[7] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang, "Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework," in *Proc. IEEE ICASSP*, vol. 5, Hong Kong, Apr. 2003, pp. 632–635.

[8] Y.-F. Ma, L. Lu, H.-J. Zhang, and M.-J. Li, "A user attention model for video summarization," in *Proc. 10th ACM Multimedia Conf.*, Juan-les-Pins, France, Dec. 2002, pp. 533–542.

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

[10] L. Lu, H.-J. Zhang, and S. Li, "Content-based audio classification and segmentation by using support vector machines," *ACM Multimedia Syst. J.*, vol. 8, no. 6, pp. 482–492, Mar. 2003.

[11] T. Zhang and C.-C. J. Kuo, "Hierarchical system for content-based audio classification and retrieval," *Proc. SPIE*, vol. 3527, pp. 398–409, Nov. 1998.

[12] Z. Liu, Y. Wang, and T. Chen, "Audio feature extraction and analysis for scene segmentation and classification," *J. VLSI Signal Process.*, vol. 20, no. 1–2, pp. 61–79, Oct. 1998.

[13] M. Baillie and J. M. Jose, "Audio-based event detection for sports video," in *Proc. 2nd Int. Conf. Image Video Retrieval*, vol. 2728, Lecture Notes in Computer Science, Urbana, IL, Jul. 2003, pp. 300–309.

[14] W.-H. Cheng, W.-T. Chu, and J.-L. Wu, "Semantic context detection based on hierarchical audio models," in *Proc. 5th ACM SIGMM Int. Workshop on Multimedia Information Retrieval*, Berkeley, CA, Nov. 2003, pp. 109–115.

[15] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search, and retrieval of audio," *IEEE Multimedia*, vol. 3, no. 3, pp. 27–36, Fall 1996.

[16] B. Gygi, "Factors in the identification of environmental sounds," Ph.D. dissertation, Dept. Psychology., Indiana Univ., Bloomington, 2001.

[17] R. Cai, L. Lu, H.-J. Zhang, and L.-H. Cai, "Using structure patterns of temporal and spectral feature in audio similarity measure," in *Proc. 11th ACM Multimedia Conf.*, Berkeley, CA, Nov. 2003, pp. 219–222.

[18] ——, "Improve audio representation by using feature structure patterns," in *Proc. IEEE ICASSP*, vol. 4, Montreal, QC, Canada, May 2004, pp. 345–348.

[19] M. A. Casey, "MPEG-7 sound-recognition tools," *IEEE Trans. Comput. Sci. Vehic. Technol.*, vol. 11, no. 6, pp. 737–747, Jun. 2001.

[20] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. New York: Springer-Verlag, 1998.

[21] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[22] M. J. Reyes-Gomez and D. P. W. Ellis, "Selection, parameter estimation, and discriminative training of hidden Markov models for general audio modeling," in *Proc. IEEE ICME*, vol. 1, Baltimore, MD, Jul. 2003, pp. 73–76.

[23] H. Sundaram and S.-F. Chang, "Audio scene segmentation using multiple features, models and time scales," in *Proc. IEEE ICASSP*, vol. 4, Istanbul, Turkey, Jun. 2000, pp. 2441–2444.

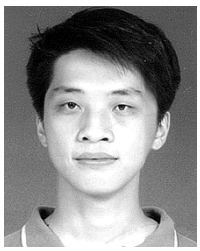
[24] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[25] D. Heckerman, "A tutorial on learning with Bayesian networks," Microsoft Research, Redmond, WA, Tech. Rep. MSR-TR-95-06, 1995.

[26] C. Huang and A. Darwiche, "Inference in belief networks: A procedural guide," *Int. J. Approx. Reas.*, vol. 15, no. 3, pp. 225–263, 1996.

[27] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classifiers*, pp. 61–74, 2000.

[28] K. Murphy, "The Bayes net toolbox for Matlab," *Comput. Sci. Statist.*, vol. 33, pp. 331–350, 2001.



Rui Cai received the B.S. degree from Tsinghua University, Beijing, China, in 2001, where he is currently pursuing the Ph.D. degree in computer science.

He was a Visiting Student with Microsoft Research Asia, Beijing, from 2002 to 2004. His current research interests include content-based multimedia analysis, pattern recognition, statistical learning, and signal processing.

Lie Lu (M'05) received the B.S. and M.S. degrees in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1997 and 2000, respectively.

Since 2000, he has been with Microsoft Research Asia, Beijing, China, where he is currently an Associate Researcher with the Speech Group. His current research interests include pattern recognition, content-based audio analysis and indexing, and content-based music analysis. He has authored more than 40 publications in these areas and has ten patents or pending applications.

Mr. Lu was a member of Technical Program Committee of the IEEE International Conference on Multimedia and Expo, 2004.

Alan Hanjalic (M'00) received the Dipl.-Ing. degree from the Friedrich-Alexander University, Erlangen, Germany, and the Ph.D. degree from the Delft University of Technology, Delft, The Netherlands, in 1995 and 1999, respectively, both in electrical engineering.

He was a Visiting Scientist at Hewlett-Packard Laboratories, Palo Alto, CA, in 1998, a Research Fellow at British Telecom Labs, Ipswich, U.K., in 2001, and a Visiting Scientist at Philips Research Laboratories, Briarcliff Manor, NY, in 2003. He is a tenured Assistant Professor and Head of the Multimedia Content Analysis research cluster at the Department of Mediamatics, Delft University of Technology. His research interests and expertise are in the broad areas of multimedia signal processing, media informatics, and multimedia information retrieval, with focus on multimedia content analysis for interactive content browsing and retrieval and on personalized and on-demand multimedia content delivery. In his areas of expertise, he has authored and coauthored more than 50 publications, including the books *Image and Video Databases: Restoration, Watermarking and Retrieval* (New York: Elsevier, 2000) and *Content-Based Analysis of Digital Video* (Norwell, MA: Kluwer, 2004). He was a Guest Editor of the *International Journal of Image and Graphics* Special Issue on Content-based Image and Video Retrieval, July 2001.

Dr. Hanjalic has served as a Program Committee Member, Session Chair, and Panel Member in many international conferences and workshops, such as IEEE ICME, IEEE ICIP, IEEE ICASSP, ICCV, ACM Multimedia, ACM SIGIR, ACM SIGMM MIR, IS&T/SPIE Storage and Retrieval for Media Databases, IS&T/SPIE Internet Multimedia Management Systems, CIVR, IEEE CSIIT, and IASTED EuroIMSA. He was the initiator and main organizer of the Symposium on Multimedia Retrieval, Eindhoven, The Netherlands, in January 2002. He is a Dutch representative in the Management Committee and a Workgroup Leader of the EU COST 292 action "Semantic Multimodal Analysis of Digital Media." He also serves regularly as an advisor/reviewer of the Belgian Science Foundation (IWT) for project proposals in the area of Information Technology and Systems. Since January 2002, he has been the secretary of the IEEE Benelux Section. He is also a member of the Organizing Committee of the IEEE International Conference on Multimedia and Expo (ICME) 2005, and will be a Co-Chair of the new IS&T/SPIE Conference on Multimedia Content Analysis, Management and Retrieval 2006, San Jose, CA, January 2006.

Hong-Jiang Zhang (S'90–M'91–SM'97–F'04) received the B.S. degree from Zhengzhou University, Zhengzhou, China, and the Ph.D. degree from the Technical University of Denmark, Lyngby, in 1982 and 1991, respectively, both in electrical engineering.

From 1992 to 1995, he was with the Institute of Systems Science, National University of Singapore, where he led several projects in video and image content analysis and retrieval and computer vision. He also worked at the MIT Media Lab, Cambridge, MA, in 1994 as a Visiting Researcher. From 1995 to 1999, he was a Research Manager at Hewlett-Packard Laboratories, where he was responsible for research and technology transfers in the areas of multimedia management, intelligent image processing, and Internet media. In 1999, he joined Microsoft Research Asia, Beijing, China, where he is currently a Senior Researcher and Assistant Managing Director in charge of media computing and information processing research. He has authored three books, over 200 refereed papers and book chapters, seven special issues of international journals on image and video processing, content-based media retrieval, and computer vision, as well as numerous patents or pending applications.

Dr. Zhang is a member of the ACM. He currently serves on the editorial boards of five IEEE/ACM journals and a dozen committees of international conferences.

Lian-Hong Cai (M'98) graduated from the Department of Automation, Tsinghua University, Beijing, China, in 1970.

She is now a Professor with the Department of Computer Science and Technology, Tsinghua University. She directs the Human–Computer Speech Interaction Group. From 1999 to 2004, she led the Institute of Human–Computer Interaction and Multimedia, Department of Computer Science and Technology, Tsinghua University. Her research interests include human–computer interaction, speech synthesis, TTS, speech corpus, and multimedia signal processing. She has published tens of publications and holds several China patents. She has been undertaking several projects of the National High Technology Development 863 Program and the National Grand Fundamental Research 973 Program of China.

Ms. Cai is a member of the Multimedia Committee of Chinese Graphics and Image Society, and a member of the Chinese Acoustics Society. She has been awarded Scientific Progress Prizes and the Invention Prizes from the Ministry of Mechanism and Electronics, and the Ministry of Education, China.