Unsupervised Content Discovery in Composite Audio

Rui Cai
Department of Computer Science and Technology, Tsinghua Univ.
Beijing, 100084, China
cairui01@mails.tsinghua.edu.cn

Lie Lu
Microsoft Research Asia
No. 49 Zhichun Road
Beijing, 100080, China
llu@microsoft.com

Alan Hanjalic
Department of Mediamatics
Delft University of Technology
2628 CD Delft, The Netherlands
A.Hanjalic@ewi.tudelft.nl

ABSTRACT
Automatically extracting semantic content from audio streams can be helpful in many multimedia applications. Motivated by the known limitations of traditional supervised approaches to content extraction, which are hard to generalize and require suitable training data, we propose in this paper an unsupervised approach to discover and categorize semantic content in a composite audio stream. In our approach, we first employ spectral clustering to discover natural semantic sound clusters in the analyzed data stream (e.g. *speech*, *music*, *noise*, *applause*, *speech mixed with music*, etc.). These clusters are referred to as *audio elements*. Based on the obtained set of audio elements, the *key audio elements*, which are most prominent in characterizing the content of input audio data, are selected and used to detect potential boundaries of semantic audio segments denoted as *auditory scenes*. Finally, the auditory scenes are categorized in terms of the audio elements appearing therein. Categorization is inferred from the relations between audio elements and auditory scenes by using the information-theoretic co-clustering scheme. Evaluations of the proposed approach performed on 4 hours of diverse audio data indicate that promising results can be achieved, both regarding audio element discovery and auditory scene categorization.

Categories and Subject Descriptors
H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing – *signal analysis*, *synthesis and processing*; *Systems*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *Indexing methods*; I.5.3 [**Pattern Recognition**]: Clustering – *Algorithms*; *Similarity measures*.

General Terms
Algorithms, Design, Experimentation, Management, Theory.

Keywords
Content-based audio analysis, unsupervised approach, key audio element, auditory scene, spectral clustering, information-theoretic co-clustering.

MM'05, November 6–11, 2005, Singapore.

1. INTRODUCTION
An automation of semantic content extraction from digital audio streams can be beneficial to many multimedia applications, such as context-aware computing [10][21] and video content parsing, including highlight extraction [2][24][25] and video abstraction and summarization [12][16]. To detect and categorize semantic content in audio signals, considerable research effort has been invested in developing the theories and methods for content-based audio analysis to bridge the "semantic gap" separating the low-level audio features and the high-level audio content semantics.

A typical approach to content-based audio analysis can be represented by the general flowchart shown in Fig. 1 [14]. There, the input audio stream is first segmented into different *audio elements* such as speech, music, various audio effects and any combination of these. Then, the *key audio elements* are selected, being the audio elements that are most characteristic for the semantics of the analyzed audio data stream [5][25]. In the next step, the *auditory scenes* [23], which are the temporal segments with coherent semantic content, are detected and classified based on the (key) audio elements they contain. For example, in [2][25], the elements such as *applause*, *cheer*, *ball-hit*, and *whistling*, are used to detect the highlights in sports videos; and in film indexing [5][6][17], *humor* and *violence* scenes are categorized by detecting the key audio elements like *laughter*, *gun-shot*, and *explosion*.

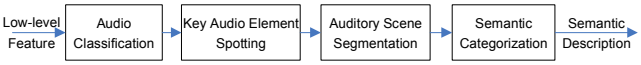


Fig. 1. A unified approach to content-based audio analysis.

Previous attempts of realizing the scheme in Fig. 1, either as a whole or in parts, usually adopted supervised data analysis and classification methods. For instance, hidden Markov models (HMMs) [2][6] and support vector machines (SVMs) [25] are often used to model and identify audio elements in audio signals. As for auditory scene categorization, heuristic rules such as "if double whistling, then Foul or Offside" are widely employed to infer the events in soccer games [25], while in [6] and [17] Gaussian mixture models (GMMs) and SVMs are used to statistically learn the relationships between the key audio elements and the higher-level semantics of auditory scenes.

Although the supervised approaches have proved to be effective in many applications, they show some critical limitations. First, the effectiveness of the supervised approaches relies heavily on the quality of the training data. If the training data is insufficient or badly distributed, the system performance drops significantly. Second, in most real-life applications, it's difficult to list all audio ele-

† This work was performed at Microsoft Research Asia.

ments and semantic categories that are possible to be found in data. For example, in the applications like pervasive computing [10] and surveillance [22], both the audio elements and the semantic scenes are unknown in advance. Thus it is impossible to collect training data and learn proper statistical models in these cases.

In view of the described disadvantages of the supervised methods, a number of recent works introduced unsupervised approaches into multimedia content analysis. For example, an approach based on time series clustering is presented in [22] to discover "unusual" events in audio streams. In [10], unsupervised analysis of personal audio archive is performed to create an "automatic diary". For the purpose of video summarization and abstraction, unsupervised approaches have also shown promising results. For instance, affective video content characterization and highlights extraction can be performed using the theory and methods proposed in [12]. Also in many other existing approaches (e.g. [16][19][24]), the techniques like clustering and grouping are utilized for semantic analysis, instead of supervised classification and identification. However, these existing methods are not meant to provide generic content analysis solutions, as they are either designed for specific applications [12][16][19][24], or only address some isolated parts of the scheme in Fig. 1 [10][22].

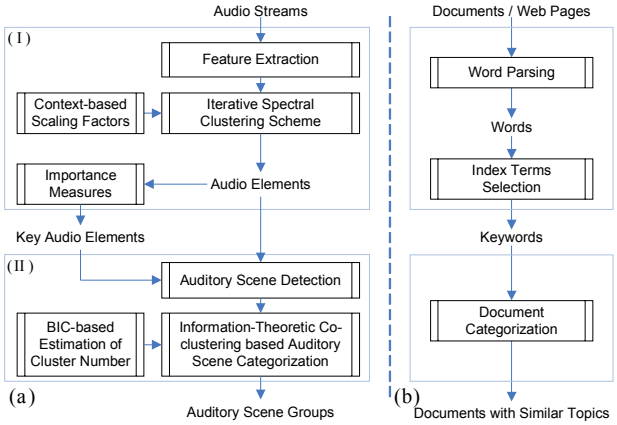


Fig. 2. (a) The flowchart of the proposed approach to unsupervised audio content analysis, which consists of two major parts: (I) audio element discovery and key element spotting; and (II) auditory scene categorization. (b) A comparable process of the topic-based document categorization.

Working towards a more generic and robust realization of the system in Fig.1, we propose in this paper a novel unsupervised approach to audio content analysis, which is capable of dealing with arbitrary composite audio data streams. The detailed flowchart of the proposed approach is given in Fig. 2 (a). It consists of two major steps: I) audio elements discovery and key audio element spotting, and II) auditory scenes categorization. Both steps are unsupervised and domain- and application-independent. It also facilitates audio content discovery in different semantic levels, such as mid-level audio elements and high-level auditory scenes. Our proposed approach can also be seen as an analogy to the topic-based text document categorization [1], as shown in Fig.2 (b). Here, audio elements are similar to words, while key audio elements correspond to keywords.

In the proposed scheme, the input is an arbitrary composite audio stream. After feature extraction, an iterative spectral clustering method is proposed to decompose the audio stream into audio elements. Spectral clustering [18] has proved to be successful in many complicated clustering problems, and is also very suitable in our case as well. To improve the clustering performance in view of the inhomogeneous distribution densities of various sounds in the feature space, we adjust the standard spectral clustering scheme [18] by using the context-dependent scaling factors. Using this clustering method, the segments with similar low-level features in the audio stream are grouped into natural semantic clusters that we adopt as audio elements. Then, a number of importance measures are defined and employed to filter the obtained set of audio elements and select the key audio elements.

In the auditory scene categorization step, the potential auditory scenes are first detected by investigating the co-occurrences among various key audio elements in the input audio stream. Then, these auditory scenes are grouped into semantic categories by using the information-theoretic co-clustering algorithm [7], which exploits the relationships among various audio elements and auditory scenes. Moreover, we propose a strategy based on the Bayesian Information Criterion (BIC) for selecting the optimal cluster numbers for co-clustering.

The rest of this paper is organized as follows. Section 2 presents the algorithms for audio element detection, including feature extraction, audio stream decomposition, and key audio elements selection. In Section 3, the procedure for auditory scene detection and categorization is described. Experiments and discussions can be found in Section 4, and Section 5 concludes the paper.

2. AUDIO ELEMENT DISCOVERY

2.1 Feature Extraction

We first divide the audio data stream into frames of 25ms with 50% overlap. Then, we compute a number of audio features to characterize each audio frame.

Many audio features have been proposed in previous works on content-based audio analysis [2][6][15], and have been proved to be effective in characterizing various audio elements. Inspired by these works, we extract both the temporal and spectral features for each audio frame. The set of temporal features consists of short-time energy (STE) and zero-crossing rate (ZCR), while the spectral features include sub-band energy ratios (BER), brightness, bandwidth, and 8-order Mel-frequency cepstral coefficients (MFCCs). Moreover, to provide a more complete description of audio elements and to be able to discern a greater diversity of audio elements, two new spectral features proposed in our pervious works [3][5], including the *Sub-band Spectral Flux* and the *Harmonicity Prominence*, are also extracted for each audio frame. In our experiments, the spectral domain is equally divided into 8 sub-bands in Mel-scale and then the sub-band features are extracted. All the above features are collected into a 29-dimensional feature vector per audio frame.

In order to reduce the computational complexity of the proposed approach, we choose to group audio frames into longer temporal audio segments of the length t, and to use these longer segments as the basis for the subsequent audio processing steps. For this purpose, a sliding window of t seconds with Δt seconds overlap is used to segment the frame sequence. In order to balance the detection resolution and the computational complexity, we choose t as 1.0 second and Δt as 0.5 seconds. At each window position, the mean and standard deviation of the frame-based features are computed and used to represent the corresponding audio segment.

2.2 Audio Stream Decomposition

The decomposition of audio streams is carried out by grouping audio segments into the clusters corresponding to audio elements. Audio elements to be found in complex composite audio streams, such as sound tracks of movies, usually have complicated and irregular distributions in the feature space. However, traditional clustering algorithms such as K-means are based on the assumption that the cluster distributions in the feature space are Gaussians [9], and such assumption is usually not satisfied in complex cases. As a promising alternative, spectral clustering [18] recently emerged and showed its effectiveness in a variety of complex applications, such as image segmentation [26][27] and the multimedia signal clustering [10][19][22]. We therefore choose to employ spectral clustering to decompose audio streams into audio elements. To further improve the robustness of the clustering process, we adopt the self-tuning strategy [27] to set context-based scaling factors for different data densities, and build an iterative scheme to perform a hierarchical clustering of input data.

2.2.1 Spectral Clustering Algorithm

For a given audio stream, the set $U = \{u_1, \ldots, u_n\}$ of feature vectors u_i is obtained through the feature extraction described in Section 2.1. Each element u_i of the set U represents the feature vector of one audio segment. After specifying the search range $[k_{min}, k_{max}]$ for the most likely number of audio elements existing in the stream, the standard spectral clustering algorithm is carried out, which consists of the following steps [18]:

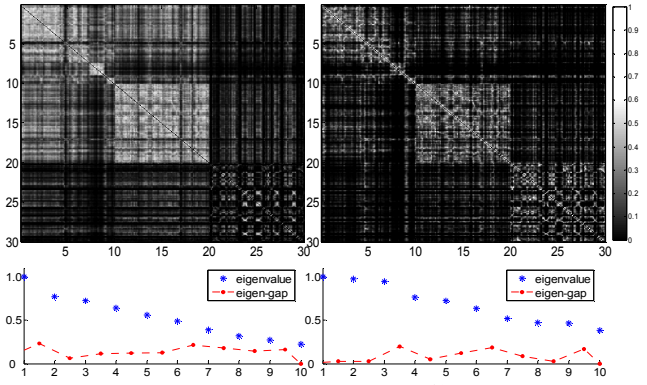
Algorithm I: Spectral_Clustering (U, k_{min}, k_{max})

1. Form an affinity matrix A defined by $A_{ij} = \exp(-d(u_i, u_j)^2/2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$. Here, $d(u_i, u_j) = \|u_i - u_j\|$ is the Euclidean distance between the feature vectors u_i and u_j, and σ is the scaling factor. The selection of σ will be discussed later in this section.
2. Define D to be a diagonal matrix whose (i, i) element is the sum of A's i^{th} row, and construct the normalized affinity matrix $L = D^{-1/2}AD^{-1/2}$.
3. Suppose $(x_1, \ldots, x_{k_{max}+1})$ are the k_{max}+1 largest eigenvectors of L, and $(\lambda_1, \ldots, \lambda_{k_{max}+1})$ are the corresponding eigenvalues. The optimal cluster number k is estimated based on the eigen-gaps between adjacent eigenvalues [18], as:
$$k = \arg\max_{i \in [k_{\min}, k_{\max}]} (1 - \lambda_{i+1} / \lambda_i) \quad (1)$$
Then, form the matrix $X = [x_1 x_2 \ldots x_k] \in \mathbf{R}^{n \times k}$ by stacking the first k eigenvectors in columns.
4. Form the matrix Y by renormalizing each of X's rows to have unit length, that is:
$$Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2} \quad (2)$$
5. Treat each row of Y as a point in $\mathbf{R}^k$, cluster them into k clusters via the cosine-distance based K-means. The initial centers in the K-means are selected to be as orthogonal to each other as possible [26].
6. Assign the original data point u_i to cluster c_j if and only if the row i of the matrix Y is assigned to c_j.

The clustering is followed by smoothing of possible discontinuities between audio segments assigned to the same cluster. For example, if the consecutive audio segments are assigned to clusters A and B as "A-A-B-A-A", they will be smoothed to "A-A-A-A-A". Each obtained series of audio segments that belong to the same cluster is considered as one instance (occurrence) of the corresponding audio element in the input audio data stream.

2.2.2 Context-based Scaling Factors

In the spectral clustering algorithm, the scaling factor σ affects how rapidly the similarity measure A_{ij} decreases when the Euclidean distance $d(u_i, u_j)$ increases. In this way, it actually controls the value of A_{ij} at which two audio segments are considered similar. In the standard spectral clustering algorithm, σ is set uniformly for all data points (for example, the average Euclidean distance in the data), based on the assumption that each cluster in the input data has a similar distribution density in the feature space. However, such assumption is usually not satisfied in composite audio data, which often contain clusters with different cluster densities. Fig. 3 (a) illustrates an example affinity matrix of a 30-second audio stream composed of *music* (0-10s), *music with dense applause* (10-20s), and *speech* (20-30s), with a uniform scaling factor. From Fig. 3 (a), it is noticed that the density of *speech* is sparser than those of other elements, and *music* and *music with dense applause* are close to each other and hard to be distinguished. Thus the standard spectral clustering can not properly estimate the number of clusters based on the eigenvalues and eigen-gaps shown at the bottom of Fig. 3 (a).



(a) (b)

Fig. 3. The similarity matrices, with top 10 eigenvalues and the eigen-gaps, of a 30-second audio stream, which consists of *music* (0-10s), *music with dense applause* (10-20s), and *speech* (20-30s): (a) using a uniform scaling factor, (b) using the context-based scaling factors.

To obtain a more reliable similarity measure and improve the clustering robustness, the self-tuning strategy [27] is employed to select context-based scaling factors in our approach. That is, for each data point u_i, the scaling factor is set based on its context data density, as:
$$\sigma_i = \sum_{j | u_j \in close(u_i)} d(u_i, u_j) / n_b \quad (3)$$
where $close(u_i)$ denotes the set containing n_b nearest neighbors of u_i, and n_b is experimentally set to 5 in our approach. Then the affinity matrix is re-defined as:
$$A_{ij} = \exp(-d(u_i, u_j)^2 / (2\sigma_i \sigma_j)) \quad (4)$$
Fig. 3 (b) shows the corresponding affinity matrix computed with the context-based scaling factors. It is noticed that the three blocks on the diagonal are more distinct than those in Fig. 3 (a). In Fig. 3 (b), *speech* segment shows more concentrated in the similarity matrix, while *music* and *music with dense applause* are more apart. According to equation (1), it is also noted the prominent eigen-gap between the 3^{rd} and 4^{th} eigenvalues can appropriately predict the correct number of clusters.

2.2.3 An Iterative Clustering Scheme

In order to prevent audio segments of different types to be merged into the same cluster, we also propose an iterative clustering scheme to verify whether a cluster can be divided any further. That is, at each iteration, each cluster obtained from previous iteration is further clustered using the spectral clustering scheme. A cluster is considered to be inseparable if the spectral clustering returns only one cluster. Although some clusters may be inseparable in the similarity matrix of the previous iteration (a global scale), they may become separable in the new similarity matrix during the next iteration (a local scale). The iterative scheme is described by the following pseudo code:

```
Iterative_Clusteirng(U, k_min, k_max) {
    [k, {c_1, …, c_k}] = Spectral_Clusteing(U, k_min, k_max);
    if (k = = 1) return;
    for (j = 1; j ≤ k; j++)
        Iterative_Clusteirng(c_j, 1, k_max);
}
```

2.3 Key Audio Element Spotting

After we discovered the audio elements in the input audio stream, we wish to spot those audio elements that are most characteristic for the semantic content conveyed by the stream. For example, while an award ceremony typically contains the elements such as *speech*, *music*, *applause* and their different combinations, the audio elements of *applause* can be considered as a good indicator of the highlights of the ceremony. To spot the key audio elements, we draw an analogy to keyword extraction in text document analysis, that is, some similar criteria are proposed to compute the "importance" of each audio element.

As a first importance indicator, we consider the occurrence frequency of an audio element, which is a direct analogy to the term frequency in text [1]. However, the difference to text analysis is that keywords in text usually have higher occurrence frequency than other words, while key audio elements may not. This can be drawn from the following analysis. For example, the major part of the sound track of a typical *action* movie segment consists of "usual" audio elements, such as *speech*, *music*, *speech mixed with music* and some "standard" background noise (*car-engine*, *opening/closing a door*, etc.), while the remaining smaller part includes audio elements that are typical for *action*, like *gun-shots* or *explosions*. As the usual audio elements can be found in any other (e.g. *romantic*) movie segment as well, it is clear that only this small set of temporal segments containing specific audio elements is the most important to characterize the content of a particular movie segment.

We apply the similar reasoning as above to extend our "importance" measure by other relevant indicators. The total durations and the average lengths of each occurrence of an audio element are, typically, very different for various sounds in a stream. Background sounds are usually majorities in streams while key audio elements are minorities. For instance, in a situation comedy, both the total duration and the average length of the *speech* are considerably longer than that of the *laughter*. Further, different sounds usually have different variations of their occurrence lengths. Key elements usually have relatively consistent length in each occurrence, as opposed to the strongly varying lengths of the segments of background sounds. For example, the sound of *applause* in a tennis game usually has similar length in each of its occurrences. The same holds for the large majority of gunshots and explosions in action movies.

Based on the above observations, we propose four heuristic importance indicators for spotting key audio elements. One of these indicators is the occurrence frequency related, and the other three are designed to capture the observations made regarding element duration. For a given audio element c_i in the stream S, these indicators are defined as follows:

- ***Element Frequency*** is used to take into account the occurrence frequency of c_i in S:

$$efrq(c_i, S) = \exp(-(n_i - \alpha \cdot n_{avg})^2 / (2n_{std}^2)) \quad (5.1)$$

  Here, n_i is the occurrence number of the audio element c_i, and n_{avg} and n_{std} are the corresponding mean and standard deviation of the occurrence numbers of all the audio elements. The factor α adjusts the expectation of how often the key elements can likely occur. By this indicator, the audio elements that appear far more or far less frequently than the expectation $\alpha \cdot n_{avg}$ are punished.
- ***Element Duration*** takes into account the total duration of c_i in the stream:

$$edur(c_i, S) = \exp(-(d_i - \beta \cdot d_{avg})^2 / (2d_{std}^2)) \quad (5.2)$$

  Here, d_i is the total duration of c_i, and d_{avg} and d_{std} are the corresponding mean and standard deviation. The factor β adjusts the expectation of key audio element duration, and has a similar effect as α.
- ***Average Element Length*** takes into account the average segment length of c_i over all its occurrences, as:

$$elen(c_i, S) = \exp(-(l_i - \gamma \cdot l_{avg})^2 / (2l_{std}^2)) \quad (5.3)$$

  Here, l_i is the average segment length of c_i, and l_{avg} and l_{std} are the corresponding mean and standard deviation of all the elements. The factor γ is similar to α and β and adjusts the expectation of the average segment length of key audio elements.
- ***Element Length Variation*** evaluates the constancy of the segment lengths of the audio element c_i in S:

$$evar(c_i, S) = \exp(-v_i / (\delta \cdot l_i)) \quad (5.4)$$

  Here, v_i is the standard deviation of the segment lengths of c_i, and δ adjusts the tolerance of v_i related to l_i.

The heuristic importance indicators defined above can be tuned adaptively for different applications, based on the available domain knowledge. For example, to detect unusual sounds in surveillance videos, the factor α, β, and γ could be set relatively small, since such sounds are not expected to occur frequently and are of a relatively short duration. On the other hand, to detect the more repetitive key sounds like *laughter* in situation comedies, we may decrease δ correspondingly, as the segments of such sounds typically have a more-or-less constant length. Assuming the above four indicators are independent with each others, in our system the following "importance" score is proposed to measure the importance of each audio element:

$$score(c_i, S) = efrq(c_i, S) \cdot edur(c_i, S) \cdot elen(c_i, S) \cdot evar(c_i, S) \quad (6)$$

The key elements are finally selected as the first K audio elements with the highest importance scores, while the remaining audio elements are further referred to as background elements. The number of key audio elements, K, is chosen as:

$$K = \arg\max_k \{\sum_{i=1}^{k} d_i' \le \eta \cdot L_S\} \quad (7)$$

where d'_i denotes the duration of the i^{th} audio element on the list of audio elements ranked in the descending order based on the score

(6), L_S is the total duration of S, and η is a tuning parameter that can be set depending on the target applications. For instance, η can be set to a relatively small value when detecting unusual sounds in a surveillance video, as the total duration of unusual sounds compared to L_S is expected to be small. In our experiments, η is set to 0.25, as we assume that the key audio elements will not cover more than 25% of the whole input audio signal.

3. AUDITORY SCENE CATEGORIZATION

Having localized the audio segments containing the key- or background audio elements, we now use this information to detect and classify higher-level auditory scenes based on the type of audio elements they contain. For both of these tasks, we exploit the co-occurrence phenomena among audio elements, and in particular, of the key audio elements. In general, some (key) audio elements will rarely occur together in the same semantic context. This is particularly useful in detecting possible brakes (boundaries) in the semantic content coherence between consecutive auditory scenes. On the other hand, the auditory scenes with similar semantics usually contain similar sets of typical key audio elements. For example, many *action* scenes will contain *gunshots* and *explosions*, while a scene in a situation comedy is typically characterized by a combination of *applause*, *laughter*, *speech* and *light music*. In this sense, the relation between the co-occurrence of audio elements and the semantic similarity of auditory scenes can be exploited for scene categorization.

3.1 Key Element-based Scene Detection

Fig. 4 illustrates an example audio element sequence, where the shaded segments are the key element segments and the white segments are the background segments. To locate the boundaries between auditory scenes in such a continuous audio stream, an intuitive idea is to measure the semantic affinity between two adjacent key-element segments in the audio stream. If this affinity is low, then there is a potential auditory scene boundary between these segments. However, it is usually difficult to measure the semantic affinities among key elements from the low-level features. In our approach, the affinity measure is based on the following assumptions: i) there is a high affinity between two segments if the corresponding key audio elements usually occur together, and ii) the larger the time interval between two adjacent key element segments, the lower their affinity.

Based on the above assumptions, we define the affinity between the l^{th} and $(l+1)^{th}$ key element segments as:

$$a_{l,l+1} = \exp(-E_{k(l)k(l+1)} / \mu_E) \cdot \exp(-t_{l,l+1} / T_m) \quad (8)$$

where $k(l)$ is the label of the l^{th} key element segment in the stream. E_{ij} is the average occurrence interval between the i^{th} key element and the j^{th} key element, μ_E is the mean of all the E_{ij}, $t_{l,l+1}$ is the time interval between the two key element segments, and T_m is a scaling factor, which is set to 16 seconds in our experiments, following the discussions on human memory limit [23]. In equation (8), we use exponential function to simulate the affinity distribution; and its two parts actually reflect the two assumptions we made above.

Using (8), an affinity curve can be obtained, as illustrated in Fig. 4. Thresholding the obtained curve will result in coarse auditory scenes, indicated by the intervals S^*_1, S^*_2 and S^*_3 in Fig. 4. We set this threshold experimentally as $\mu_a+\sigma_a$, where μ_a and σ_a are the mean and standard deviation of the affinity curve, respectively.

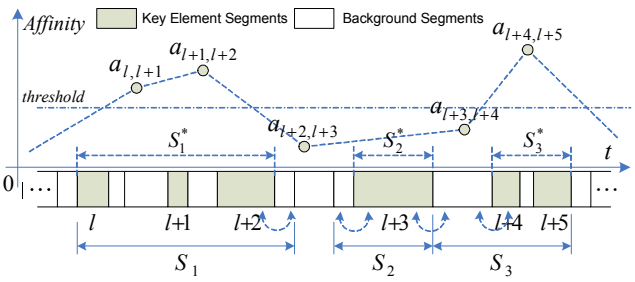


Fig. 4. An illustration of the key audio element-based auditory scene detection, where the shaded and white segments represent the key- and background audio elements, respectively. The scenes are first located by measuring the affinities between adjacent key element segments, as shown by the intervals S^*_1, S^*_2 and S^*_3. Then, the scene boundaries are further revised by merging the surrounding background segments, as shown by the intervals S_1, S_2, and S_3.

The boundaries of the coarsely located auditory scenes are defined by the key elements. To determine these boundaries more precisely, we need to find out whether the surrounding segments of background elements could be merged into the corresponding auditory scene, so we can adjust the boundaries accordingly. In our approach, a background segment is merged into an auditory scene if its affinity with the key element on the scene boundary is large enough (larger than a threshold). Moreover, if a background segment has a sufficient affinity to both surrounding scenes, it is merged with the scene whose key element on the corresponding scene boundary has a larger affinity value with the background element.

In the implementation of the above mechanism, the affinity between the key element and the background element is based on the criterion similar to equation (8) that evaluates the average occurrence interval between them. Again, a pre-defined threshold, set experimentally, is used to evaluate the affinity, in the same way as the threshold in Fig. 4. After the boundary refinement, the coarse auditory scenes are updated, as the intervals S_1, S_2, and S_3 show in Fig. 4.

As stated above, we base the detection of auditory scene boundaries on the comparison of two subsequent key audio elements, which may be a little strict. A more intuitive approach would be to allow more flexibility in the ordering of key audio elements, as long as their mutual distance remains acceptable. In this way we would come close to the classical video scene segmentation approaches, such as those based on fast-forward linking [11] or content recall [13]. In our approach, we choose to stick to the strict criterion (8) in order to prevent that two semantically different auditory scenes are seen as one. Clearly, the proposed method will most likely result in an over-segmentation of the input audio stream. This, however, is not a problem as the semantically similar scenes will be grouped together in the following step.

It is also noted that, with this scheme, some background elements may not belong to any auditory scene, or, in other words, each of these background elements could also be considered as an individual auditory scene. For these scenes, the categorization can be simply based on (or the same as) their element label. Therefore, in the following scene categorization and evaluations, we don't consider the auditory scenes which contain only a background element, and just consider those auditory scenes which contain both key- and background audio elements.

3.2 Co-clustering based Scene Categorization

In the proposed approach, audio elements are used as mid-level representations of the auditory scene content. Thus, for the auditory scene categorization, the semantic similarity between auditory scenes can be measured based on the audio elements they contain. Although previous works usually use key audio elements to infer the semantics of auditory scenes [6][25], in our approach, all the audio elements are used in the auditory scene categorization, since those background elements can be considered as the context of the key elements, and thus can also provide extra useful information in the semantic grouping.

In scene categorization, it is useful to consider the "grouping" tendency among audio elements [4]: some audio elements usually occur together in the scenes of similar semantics, such as the co-occurrence of *gun-shots* and *explosions* in *war* scenes, and of *cheer* and *laughter* in *humor* scenes. Clearly, in a reliable similarity measure of auditory scenes, the "distance" between two audio elements in a same "group" should be smaller than those among different element "groups" [4]. Therefore, to obtain reliable results of auditory scenes categorization, audio elements also need to be grouped according to their co-occurrences in various auditory scenes. Essentially, the processes of clustering auditory scenes and revealing likely co-occurrences of audio elements can be considered dependent on each other [4]. That is, the information on semantic scene clusters can help reveal the audio elements co-occurrence and vice versa.

An analogy to the above can again be found in the domain of text document analysis, where a solution in the form of a *co-clustering* algorithm was recently proposed for unsupervised topic-based document clustering, which exploits the relation between document clusters and the co-occurrences of keywords. Unlike traditional one-way clustering such as k-means, co-clustering is a two-way clustering algorithm. With this algorithm, the documents and keywords are clustered simultaneously based on the concept of mutual information.

In our approach, the information-theoretic co-clustering [7] is adopted to co-cluster the auditory scenes and audio elements. Moreover, we also extend the algorithm with the Bayesian Information Criterion (BIC) to automatically select the cluster numbers for both auditory scenes and audio elements.

3.2.1 Information-Theoretic Co-clustering Algorithm

The information-theoretic co-clustering can effectively exploit the relationships among various audio elements and auditory scenes, based on the *mutual information* theory [7]. Following the work introduced in [7], we suppose there are m auditory scenes and n audio elements, and all the auditory scenes could be considered as being generated by a discrete random variable S, whose value s is taken from the set $\{s_1, \ldots, s_m\}$; and similarly, all the audio elements are generated by another discrete random variable E, whose value e is taken in the set $\{e_1, \ldots, e_n\}$. Let $p(S, E)$ be a matrix, with each element $p(s, e)$ representing the co-occurrence probability of an audio element e and the auditory scene s. Then, the *mutual information* $I(S; E)$ is calculated as:

$$I(S;E) = \sum_s \sum_e p(s,e) \log_2 (p(s,e) / p(s)p(e)) \quad (9)$$

It was proved in [7] that an optimal co-clustering should minimize the *loss of mutual information* after the clustering, i.e. the optimal clusters should minimize the difference:

$$I(S;E) - I(S^*;E^*) = KL(p(S,E), q(S,E)) \quad (10)$$

where $S^*=\{s^*_1, \ldots, s^*_k\}$ and $E^*=\{e^*_1, \ldots, e^*_l\}$ are k and l disjoint clusters formed from the elements of S and E. $KL(\cdot)$ is the *Kullback-Leibler* (*K-L*) divergence, and $q(S,E)$ is also a distribution in the form of an $m \times n$ matrix, given as:

$$q(s,e) = p(s^*,e^*)p(s \mid s^*)p(e \mid e^*), \text{where } s \in s^*, e \in e^* \quad (11)$$

After some transformations, equation (10) can be further expressed in a symmetrical manner [7]:

$$KL(p,q) = \sum_{s^*} \sum_{s \in s^*} p(s) KL(p(E \mid s), q(E \mid s^*)) \quad (12.1)$$

$$KL(p,q) = \sum_{e^*} \sum_{e \in e^*} p(e) KL(p(S \mid e), q(S \mid e^*)) \quad (12.2)$$

Expressions (12.1) and (12.2) show that the loss of mutual information can be minimized by minimizing the *K-L* divergence between $p(E|s)$ and $q(E|s^*)$, or between $p(S|e)$ and $q(S|e^*)$. This leads to the following iterative four-step co-clustering algorithm, which was proved to monotonically reduce the *loss of mutual information* and converge to a local minimum [7]:

Algorithm II: Co_Clustering (p, k, l)

1) Initialization: Group all auditory scenes into k clusters, and the audio elements into l clusters. These initial clusters are formed such that their centroids are "maximally" far apart from each other [8]. Then calculate the initial value of the q matrix.
2) Updating row clusters: First, for each row s, find its new cluster index i as:
$$i = \arg\min_k KL(p(E \mid s), q(E \mid s^*_k)) \quad (13)$$
Thus the *K-L* divergence of $p(E|s)$ and $q(E|s^*)$ decreases in this step. With the new cluster indices of rows, update the q matrix according to equation (11).
3) Updating column clusters: Based on the updated q matrix in step 2, find a new cluster index j for each column e as:
$$j = \arg\min_l KL(p(S \mid e), q(S \mid e^*_l)) \quad (14)$$
Thus the *K-L* divergence of $p(S|e)$ and $q(S|e^*)$ decreases in this step. With the new cluster indices of columns, update the q matrix again.
4) Re-calculate the *loss of mutual information* using equation (10). If the change in the *loss of mutual information* is smaller than a pre-defined threshold, stop the iteration process and return the clustering results; otherwise go to step 2 to start a new iteration. To increase the quality of the local minimum, the local search strategy is applied [8].

Since what we know about the input audio stream is the presence and duration of discovered audio elements per each detected audio scene, the occurrence probability of the audio element e_j and the auditory scene s_i is approximated in our implementation simply by the duration percentage $occr_{ij}$ of e_j in s_i. If an audio element doesn't occur in the scene, its duration percentage is set to zero. Finally, to satisfy the requirement that the integral (sum) of the co-occurrence distribution is equal to one, the co-occurrence matrix $p(S, E)$ is normalized across the whole matrix, that is:

$$p(s_i, e_j) = occr_{ij} / \sum_{i=1}^{m} \sum_{j=1}^{n} occr_{ij} \quad (15)$$

3.2.2 BIC-based Cluster Number Estimation

In the above co-clustering algorithm, the row cluster number k and the column cluster number l are assumed to be known. However, in an unsupervised approach, it's not possible to specify the cluster numbers beforehand.

In our proposed approach, the Bayesian Information Criterion (BIC) is utilized to select the optimal cluster numbers for co-clustering. In general, the BIC searches for a tradeoff between the data likelihood and the model complexity, and has been successfully employed to select the optimal cluster number for *K*-means clustering [20]. In our co-clustering scheme, assuming that the model preserving more mutual information would better fit the data, the data likelihood is described by the logarithm of the ratio between the *mutual information* after clustering $I(S^*; E^*)$ and the original *mutual information* $I(S; E)$. As co-clustering is a two-way clustering, the model complexity here should consist of two parts: the size of the row clusters ($n \times k$: k cluster centers of dimensionality n) and the size of the column clusters ($m \times l$: l cluster centers of dimensionality m). According to the definition of BIC [4], these two parts are further modulated by the logarithm of the numbers in row and column, i.e. log$m$ and log$n$, respectively. Thus, the BIC in our algorithm can be formulated as:

$$BIC(k,l) = \lambda \log(I(S^*;E^*)/I(S;E)) - (nk \log m + ml \log n)/2 \quad (16)$$

In our implementation, λ is set experimentally as $m \times n$, which is the size of the co-occurrence matrix. The algorithm searches over all the (k, l) pairs in a pre-defined range, and the model with the highest BIC score is chosen as the optimal set of cluster numbers.

4. EVALUATION AND DISCUSSION

In this section, we present the evaluation results obtained for the proposed approach on the basis of several composite audio streams, and addressing both the audio element discovery and auditory scene categorization.

4.1 Database Information

The proposed framework was evaluated on sound tracks extracted from various types of video, including sports, situation comedy, award ceremony, and movies, and in the total length of about 4 hours. These sound tracks contain an abundance of different audio elements, and are of different complexity, in order to provide a more reliable base for evaluating the proposed approach under different conditions. For example, in the test dataset, the sound track of the tennis game is relatively simple, as compared to a far more complex sound track from the *war* movie "Band of Brothers - Carentan".

Table 1. Information of the experimental audio data

No.	Video	category	duration
A_1	*Tennis Game*	sports	0:59:41
A_2	*Friends*	situation comedy	0:25:08
A_3	*59th Annual Golden Globe Awards*	award ceremony	1:39:47
A_4	*Band of Brothers - Carentan*	war movie	1:05:19

Detailed information on the sound tracks we used is listed in Table 1. All the audio streams are in 16 KHz, 16-bit and mono channel format, and are divided into frames of 25ms with 50% overlap for feature extraction. To balance the detection resolution and the computational complexity, the length of the sliding window introduced in Section 2.1 is chosen as one second, with 0.5 seconds overlap.

4.2 Audio Element Discovering and Key Element Spotting

In this section, the performance of the proposed unsupervised approach to audio element discovery and key element spotting is evaluated. Due to the page limitation, we do not list all the detailed performance figures for all test audio streams. Instead, we first provide an exhaustive performance presentation on the example of one test audio stream, and then give a summary of performance figures obtained on all other audio streams. In each evaluation step, a different test stream is chosen as an example.

4.2.1 Audio Element Discovery

In the spectral clustering for audio element discovery, the boundaries of the search ranges for selecting the numbers of clusters are set experimentally as k_{min}=2 and k_{max}=20 for all the sound tracks. Moreover, to illustrate the effectiveness of the proposed spectral clustering scheme with context-based scaling factors, we compare this scheme with the standard spectral clustering.

Table 2. Comparison of the results of the standard spectral clustering and the spectral clustering with context-based scaling factors on the sound track of "Friends" (A_2) (unit: second)

	No.	N	S	A	L	L&M	M	precision
Spectral clustering with context-based scaling factors	1	42	2		0.5			0.944
	2	7	1132.5	1	8			0.986
	3			5				1.000
	4	1	2		215			0.986
	5	3			8	31.5		0.741
	6	0.5					46.5	0.980
	7	0.5					2.5	
	recall	0.778	0.996	0.833	0.929	1.000	1.000	**0.978**
Standard spectral clustering	1	50.5	43.5					0.537
	2	1.5	527.5		4	2		0.977
	3		290	6	2	1	7	
	4		267		1.5			
	5	2	8.5		224	28.5	42	0.734
	recall	0.935	0.954	0.000	0.968	0.000	0.000	**0.901**

Abbr. noise (**N**), speech (**S**), applause (**A**), laughter (**L**), and music (**M**)

Table 2 shows the comparison results of the two spectral clustering algorithms on the sound track of "Friends". In this experiment, we obtained 7 audio elements using the spectral clustering with context-based scaling factors, and only 5 audio elements using the standard spectral clustering. To enable a quantitative evaluation of the clustering performances, we established the ground truth by combining the results obtained by three unbiased persons who analyzed the content of the sound track and the obtained audio elements. This process resulted in 6 sound classes that we labeled as *noise* (**N**), *speech* (**S**), *applause* (**A**), *laughter* (**L**), *music* (**M**), and *laughter with music* (**L&M**). In Table 2, each row represents one discovered audio element and contains the durations (in seconds) of its occurrences in view of the ground truth sound classes. We manually grouped those audio element occurrences associated to the same ground truth class (indicated by shaded fields in Table 2), and then calculated the precision, recall and accuracy (the duration percentage of the correctly assigned audio segments in the stream) based on the grouping results. As shown in Table 2, the accuracies of the two algorithms are in average 97.8% and 90.1%, respectively, for the sound track of "Friends" (A_2).

Table 2 shows that each class in the ground truth can be covered by the audio elements discovered with the spectral clustering using context-based scaling factors. In the standard spectral clustering, the sounds of *applause* (**A**), *music* (**M**) and *laughter with music* (**L&M**) were missed and falsely included into other clusters, while *speech* (**S**) is divided over three discovered audio elements. As demonstrated in Section 2.2.2, this phenomenon may be caused by the unharmonious distributions of various sound classes in the feature space. For instance, the feature distribution of *speech* (**S**) is relatively sparse and is with large divergence, while those of *music* (**M**)

and *laughter with music* (**L&M**) are more "tight". The influence of unharmonious sound distributions can be reduced by setting different scaling factors for different data densities, as done in our approach.

Table 3. Performance comparison between the spectral clustering with and without context-based scaling factors on all the sound tracks

No.	#gc	Standard spectral clustering		Spectral clustering with context-based scaling factors	
		#nc / #miss	accuracy	#nc / #miss	accuracy
A_1	6	7 / 3	0.747	7 / 0	0.951
A_2	6	5 / 3	0.901	7 / 0	0.978
A_3	7	8 / 2	0.814	11 / 0	0.928
A_4	6	5 / 3	0.621	16 / 0	0.930

The performance of the audio element discovery on all test sound tracks is summarized in Table 3, which lists the number of ground truth sounds (***#gc***), the number of discovered audio elements (***#nc***), the number of missed ground truth audio elements (***#miss***), and the overall accuracy. Table 3 shows that by using the standard spectral clustering algorithm, around 44% of sound classes in the ground truth are not properly discovered, and the average accuracy is only around 77.1%. The table also shows that the spectral clustering with context-based scaling factors performs better on all the test sound tracks, and achieves an average accuracy of around 94.7%. In particular, no sound classes in the ground truth are missed in the obtained set of audio elements. Hence, the use of context-based scaling factors in the spectral clustering of complex audio streams can notably improve the clustering performance.

4.2.2 Key Audio Element Spotting

Based on the discovered audio elements, the heuristic rules proposed in Section 2.4 are employed to automatically spot the key audio elements. In the experiments, the parameters α, β, γ, and δ in equation (5.1)-(5.4) are simply set as 1 without hard tuning, assuming that the key element in the streams has medium occurrence frequency and duration, such as the *applause* in the tennis game (A_1) and the *laughter* in the situation comedy (A_2).

Table 4 contains the results of the key elements spotting in the sound track of "Tennis" (A_1). For 7 discovered audio elements in the stream, the table lists their total duration (***dur***), the number of segments included in the corresponding clusters (***nseg***), and the mean and standard deviation of the segment length (***avgl*** and ***stdl***). Based on these properties, the importance score of each audio element is computed and an "educated guess" is made for the most likely number of key elements using equation (7). In the tennis soundtrack, we finally obtain two key elements, *applause with speech* and *pure applause*, as indicated by the shaded fields in Table 4.

It is noted that, in Table 4, the cluster 7 which contains *ball-hit* sound was not labeled as a key audio element, although one could expect that *ball-hit* is also a key element in the case of a tennis sequence. Actually, the obtained importance score for this cluster also shows that it is an excellent candidate for being the key audio element. However, due to the finite resolution of the stream decomposition, defined by the sliding-window overlap of 0.5s, in our approach, the segments of *ball-hit* usually constrain long-time *silence* or *noise*. Therefore, in audio element clustering, the segments of *ball-hit* are mixed with an amount of *silence* or *noise* segments in the obtained cluster 7. Thus, cluster 7 in fact describes background sounds in tennis match and is not taken as a ground truth key element in this paper. It is also not selected as a key element in the experiments, since the overall duration of this cluster – when added to the overall durations of the first two key audio elements – is too large for the threshold set in equation (7).

Table 4. Key element spotting on the track of "Tennis" (A_1)

No.	Description	dur	nseg	avgl	stdl	score
1	clean speech	1658.0	250	6.632	7.984	0.020
2	applause with speech	341.0	108	3.157	1.721	0.928
3	pure music	22.0	1	22.00	0.000	0.008
4	pure applause	319.5	106	3.014	1.961	0.908
5	silence	837.5	173	4.841	3.599	0.633
6	noise	96.5	32	3.016	2.502	0.399
7	silence with ball-hit	307.5	145	2.121	1.278	0.820

Table 5. Discovered key elements in all the sound tracks

No.	k_0/dur_0	k/dur	rec. / prec.	Key elements (in descending order)
A_1	2 11:16	2 11:01	0.926 0.947	applause with speech, pure applause
A_2	3 04:29	4 06:10	0.998 0.726	laughter, noise*, laughter with music, applause
A_3	4 27:01	5 24:49	0.869 0.946	applause, applause with light-music1, applause with light-music2, applause with dense-music, applause with speech
A_4	2 14:38	1 11:54	0.743 0.913	gun-shot mixed with explosion, pure gun-shot**

Note: (*) Falsely spotted key elements; (**) Missed key elements

The spotted key audio elements from all test sound tracks are shown in Table 5. For each sound track, the number of ground truth key elements (***k_0***) and the total duration of corresponding audio segments (***dur_0***) are listed, as well as the spotted key element number (***k***) and their total duration (***dur***). The ground truth is established here again by combining the results obtained by three unbiased persons who analyzed the content of the test sound tracks in the search for the most characteristic sounds and sound combinations. Based on these values, the recall and precision are computed. The table shows that the performance on relatively simple audio streams (A_1 and A_3) is satisfying. All the key elements in the ground truth are well spotted and no false alarms are introduced. The average recall and precision are above 90%. On the other hand, for complex audio streams such as the situation comedy TV (A_2) and the war movie (A_4), some false alarms are introduced and some key elements are missed. For example, the *noise* in A_2 is falsely detected as key element, since it has similar occurrence frequency and duration as the expected key elements. Also in A_4, some real key elements such as the *pure gunshot* are missed, since the characteristics of key elements in complex audio streams vary too large and are inconsistent. These problems indicate that the heuristic rules proposed in this paper can not yet give a complete description to all the characteristics of key elements in complex audio streams, and need to be improved in the future works. However, the overall performance of key element spotting using the proposed rules on our testing audio streams is still acceptable, and more than 90% (10 out of 11) of the key elements in the ground truth can be properly spotted.

4.3 Auditory Scene Categorization

To better evaluate the performance of the auditory scene categorization with co-clustering algorithm, we first do some minor corrections in the key element spotting based on the available ground truth information. That is, we delete the *noise* from the key element list of A_2, and add the *pure gunshot* to the list of A_4. Then, the auditory scenes are automatically located with the strategy proposed in Section 3.1. Finally, we again employed three persons to manually

group these scenes into a number of semantic categories. Based on this manual grouping, we established the ground truth for further evaluation.

To illustrate the effectiveness of the proposed co-clustering scheme for scene categorization, we compare it with a traditional one-way clustering algorithm. Here, the X-means algorithm [20], in which BIC is also used to estimate the number of clusters, is adopted for the comparison. We search for the proper cluster number K in the range of $1 \le K \le 10$ in the X-means clustering; while in the co-clustering, we search for the optimal number k of auditory scene categories and the optimal number l of audio element groups in the ranges of $1 \le k \le 10$ and $1 \le l \le n$ (n is the number of audio elements in the corresponding sound track), respectively.

Table 6. Detailed results comparison between the X-means and the Co-clustering for auditory scene categorization on the sound track of the "59th Annual Golden Globe Awards" (A_3)

	No.	S_1	S_2	S_3	prec.		No.	S_1	S_2	S_3	prec.
Co-clustering	1	21			1.000	**X-means**	1	8	1	0	0.947
	2	26					2	4	0	1	
	3	11					3	13	0	0	
	4		33		1.000		4	17	0	0	
	5	3		21	0.875		5	12	1	0	
	recall	0.951	1.000	1.000	**0.974**		6	3	15	0	0.824
							7	2	13	1	
							8	1	1	7	0.792
							9	1	2	12	
							recall	0.885	0.848	0.900	**0.878**

Note: (S_1) scenes of hosts or winners coming to or leaving the stage; **(S_2)** scenes of audience applauding the winners; **(S_3)** scenes of hosts introducing the winner candidates.

Table 6 shows the detailed comparison results of the two clustering algorithms on the example sound track of the "59th Annual Golden Globe Awards" (A_3). In this stream, there are totally 115 obtained scenes, which are manually classified into 3 semantic categories: 1) the scenes of hosts or winners coming to or leaving the stage (**S_1**), which are mainly composed of *applause* and *music*; 2) the scenes of audience applauding the winners (**S_2**), and 3) the scenes of hosts introducing the winner candidates (**S_3**), which are mainly composed of *applause* and *speech*.

In the experiments, we obtained 5 auditory scene categories using the information-theoretic co-clustering and 9 scene categories by the X-means. In Table 6, each row represents one obtained cluster and the distribution of the auditory scenes contained therein across the ground truth categories. Similar to Table 2, we also manually group those clusters associated to the same ground truth category (as indicated by the shaded fields in Table 6), and then calculate the corresponding precision and recall per grouped cluster. The results reported in Table 6 show that the co-clustering algorithm can achieve better performance in auditory scene categorization. First, the number of auditory categories obtained by co-clustering is closer to the number of ground truth categories, than that achieved with the X-means. In other words, the co-clustering can provide a more exact approximation of the actual semantic content classes existing in audio streams. Second, co-clustering performs better than the X-means clustering, both in terms of precision and recall. In average, around 97.4% of the scenes are correctly clustered with the co-clustering algorithm, while the accuracy of the X-means is 87.8%.

The performance comparison between the X-means and the co-clustering on all the sound tracks is summarized in Table 7. Similar to the categorization results on A_3, co-clustering achieves higher accuracies on all test sound tracks, and also has a closer approximation of the ground truth categories in all cases.

Table 7. Performance comparison between the X-means and the Co-clustering on all the sound tracks

No.	Labeled semantic group num.	X-means		Co-clustering	
		Group num.	Accuracy	Group num.	Accuracy
A_1	3	7	0.900	4	0.930
A_2	3	5	1.000	4	1.000
A_3	3	9	0.878	5	0.974
A_4	2	6	0.839	4	0.871

Furthermore, with the co-clustering algorithm we also obtain several audio element groups for each test audio stream, as shown in Table 8. These clustering results realistically reveal the grouping (co-occurrence) tendency among the audio elements, as explained in Section 3.2. For example, in the "59th Annual Golden Globe Awards" ceremony (A_3), we observed that the sounds of *applause with light-music* and *applause with dense-music* usually occurs together in the scenes of "the hosts or winners coming to or leaving the stage", and they are correctly grouped together with the co-clustering algorithm.

Table 8. The audio element groups obtained using the co-clustering algorithm on each sound track

No.	#G	audio element groups
A_1	4	{clear speech}; {applause with speech, silence}; {pure music}; {pure applause, silence, silence with ball-hit}
A_2	3	{noise}; {laughter, laughter with music}; {music1, music2, speech, applause}
A_3	5	{speech1, speech2, speech3, applause with speech}; {applause}; {music with speech, music}; {noise} {applause with dense-music, applause with light-music1, applause with light-music2};
A_4	4	{speech with sparse gun-shot, gun-shot mixed with explosion, pure gun-shot }; {heavy noise, noise, speech with noise}; {speech1,speech2, speech3, speech4, silence1, silence2, silence3, applause, music with speech}; {music}

5. CONCLUSIONS

In this paper, an unsupervised approach is proposed to discover semantic auditory content in composite audio streams. In this approach, a spectral clustering-based scheme is presented to segment and cluster the input stream into audio elements. By sorting the obtained elements according to their importance scores, key audio elements are discovered and then used to locate the potential auditory scenes in audio streams. Finally, an information-theoretic co-clustering based categorization approach is utilized to group the auditory scenes with similar semantics, by exploiting the relations among different audio elements and auditory scenes. It is noted although in these steps there are a lot of tuning parameters, most of them can be set simply and adaptively, without hard tuning. It also indicates the generality of the proposed approach. Experimental evaluations have shown that the proposed unsupervised approach can achieve very encouraging results on various audio streams, both with respect to audio elements discovery and key audio element spotting, and to auditory scenes categorization.

While the results reported in this paper are promising, the proposed solution for audio content discovery still leaves considerable room for further investigation and improvement. For instance, the heuristic rules used for key element spotting and auditory scene location

may not always be capable of handling highly complex content to be found in some audio streams. We aim at making these rules more robust and reliable in our future works, for example, by considering the relationships among various importance indicators in key element spotting.

6. REFERENCES

[1] Baeza-Yates, R., and Ribeiro-Neto, B. *Modern Information Retrieval*. Addison-Wesley, Boston, MA, 1999.

[2] Cai, R., Lu, L., Zhang, H.-J., and Cai, L.-H. Highlight sound effects detection in audio stream. In *Proc. of the 4th IEEE International Conference on Multimedia and Expo*, 2003, vol. 3, 37-40.

[3] Cai, R., Lu, L., Zhang, H.-J., and Cai, L.-H. Improve audio representation by using feature structure patterns. In *Proc. of the 29th IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004, vol. 4, 345-348.

[4] Cai, R., Lu, L., and Cai, L.-H. Unsupervised auditory scene categorization via key audio effects and information-theoretic co-clustering. In *Proc. of the 30th IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005, vol. 2, 1073-1076.

[5] Cai, R., Lu, L., Hanjalic, A., Zhang, H.-J., and Cai, L.-H. A flexible framework for key audio effects detection and auditory context inference. to appear in *IEEE Trans. Speech Audio Processing*, May, 2006.

[6] Cheng, W.-H., Chu, W.-T., and Wu, J.-L. Semantic context detection based on hierarchical audio models. In *Proc. of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2003, 109-115.

[7] Dhillon, I. S., Mallela, S., and Modha, D. S. Information-theoretic co-clustering. In *Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003, 89-98.

[8] Dhillon, I. S., and Guan, Y. Information theoretic clustering of sparse co-occurrence data. In *Proc. of the 3rd IEEE International Conference on Data Mining*, 2003, 517-520.

[9] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification, Second Edition*. John Wiley & Sons, NJ, 2000.

[10] Ellis, D., and Lee, K. Minimal-impact audio-based personal archives. In *Proc. of ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, 2004, 39-47.

[11] Hanjalic, A., Lagendijk, R. L., and Biemond, J. Automated high-level movie segmentation for advanced video-retrieval systems. *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 580-588, Jun. 1999.

[12] Hanjalic, A., and Xu, L.-Q. Affective video content representation and modeling. *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 143-154, Feb. 2005.

[13] Kender, J. R., and Yeo, B.-L. Video scene segmentation via continuous video coherence. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998, 367-373.

[14] Lu, L., Cai, R., and Hanjalic, A. Towards a unified framework for content-based audio analysis. In *Proc. of the 30th IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005, vol. 2, 1069-1072.

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

[16] Ma, Y.-F., Lu, L., Zhang, H.-J., and Li, M.-J. A user attention model for video summarization. In *Proc. of ACM International Conference on Multimedia*, 2002, 533-542.

[17] Moncrieff, S., Dorai, C., and Venkatesh, S. Detecting indexical signs in film audio for scene interpretation. In *Proc. of the 2nd IEEE International Conference on Multimedia and Expo*, 2001, 989-992.

[18] Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: analysis and an algorithm. *Advances in Neural Information Processing Systems 14* (*Proc. of NIPS 2001*), 849-856.

[19] Ngo, C.-W., Ma, Y.-F., and Zhang, H.-J. Video summarization and scene detection by graph modeling. *IEEE Trans. Circuits and Systems for Video Technology*, vol. 15, no. 2, pp. 296-305, Feb. 2005.

[20] Pelleg, D., and Moore, A. W. X-means: extending K-means with efficient estimation of the number of clusters. In *Proc. of the 17th International Conference on Machine Learning*, 2000, 727-734.

[21] Peltonen, V., Tuomi, J., Klapuri, A. P., Huopaniemi, J., and Sorsa, T. Computational auditory scene recognition. In *Proc. of the 27th IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002, vol. 2, 1941-1944.

[22] Radhakrishnan, R., Divakaran, A., and Xiong, Z. A time series clustering based framework for multimedia mining and summarization using audio features. In *Proc. of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2004, 157-164.

[23] Sundaram, H., and Chang, S.-F. Determining Computable scenes in films and their structures using audio visual memory models. In *Proc. of the 8th ACM International Conference on Multimedia*, 2000, 95-104.

[24] Xie, L., Chang, S.-F., Divakaran, A., and Sun H. Unsupervised mining of statistical temporal structures in video. *Video Mining*, Kluwer Academic Publishers, 2003, 279-307.

[25] Xu, M., Maddage, N., Xu, C.-S., Kankanhalli, M., and Tian, Q. Creating audio keywords for event detection in soccer video. In *Proc. of the 4th IEEE International Conference on Multimedia and Expo*, 2003, vol. 2, 281-284.

[26] Yu, S. X., and Shi, J. Multiclass spectral clustering. In *Proc. of the 9th IEEE International Conference on Computer Vision*, 2003, vol. 1, 313-319.

[27] Zelnik-Manor, L., and Perona, P. Self-tuning spectral clustering. *Advances in Neural Information Processing Systems 17* (*Proc. of NIPS 2004*), 1601-1608.