

# Towards Optimal Audio "Keywords" Detection for Audio Content Analysis and Discovery

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

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

## ABSTRACT

Natural semantic sound clusters in an audio document, also referred to as *audio elements*, can be seen as an analogy to words in a text document. Based on the obtained set of audio elements, the *key audio elements*, or audio "keywords", can be detected, which are most prominent in characterizing the content of audio data. As such, they can be of great use for automatic audio content analysis and discovery. Motivated by the limitations of the existing methods for key audio element detection, we propose in this paper a novel unsupervised approach to audio elements weighting using multiple audio documents, analog to word weighting in text document analysis. In our approach, *dominant feature vectors* (DFV) are first extracted from each audio element, and used to measure the audio elements similarity, based on which the occurrence probability of one audio element in different audio documents can be estimated. Then, four factors, including *expected term frequency*, *expected inverse document frequency*, *expected term duration*, and *expected inverse document duration*, are calculated and combined to give the importance weight of each audio element. Evaluation of the obtained audio "keywords" and their usability for auditory scene segmentation and audio document clustering, performed on 5 hours of diverse audio data, shows highly promising results.

## 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, key audio element extraction, audio content mining and knowledge discovery, audio content parsing, audio content classification.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'06, October 23-27, 2006, Santa Barbara, California, USA.  
Copyright 2006 ACM 1-59593-447-2/06/0010...\$5.00.

## 1. INTRODUCTION

Considerable research effort has been invested in developing the theories and methods for content-based audio analysis, which attempts to detect the high-level semantic content in audio signals. The proposed theories and methods can generally be divided into two major groups, namely (a) those relying on a direct analysis of low-level features [8][16], and (b) those, which use *audio elements* as a mid-level to bridge the gap between low-level features and high-level semantics [3][18]. We refer to audio elements as natural clusters of audio data, which can be pure speech, music, sound effects, noise and any combination of these. An audio element can be seen as an equivalent for a word in a text document. Consequently, *key audio elements* (also referred to as audio events [12], or audio keywords [18]) can be considered as well. Just like keywords in text, they can be used to represent the essence of the semantic content conveyed by the audio data stream, and to parse, cluster and classify audio documents using the general approach to topic detection and classification known from the field of text analysis [1].

Recent studies [3][10] have shown that the (key) audio-element based approaches to semantic parsing and classification of audio outperform the "plain" feature-based approaches. This can, in particular, be observed on the increased precision in the obtained results. For instance, in the case of auditory scene segmentation, audio-element based analysis inherently searches for high-level content breaks only, and neglects irrelevant variations in audio data due to which the feature-based approaches usually result in an over-segmentation.

Previous attempts to detect key audio elements usually adopted supervised data analysis and classification methods. For example, hidden Markov models (HMMs) are employed in [3] to detect 10 key audio elements including *applause*, *cheer*, and *laughter*, and in [6] to detect the key audio elements such as *car-racing*, *siren*, *gun-shot*, and *explosion*. Similarly, support vector machines (SVMs) were employed by Xu et al. [18] to detect key audio effects in soccer games, such as *whistling* and *ball-hit*, and also in [12] to detect *sirens* and *gun shots* for the purpose of movie indexing. Supervised approaches have proved to be effective in many applications. However, the expected key audio elements need to be predefined, and the effectiveness of supervised approaches relies heavily on the quality and quantity of the training data. This makes such approaches difficult to generalize.

In view of the described disadvantages of supervised methods, some recent works introduced unsupervised approaches to key audio element detection. For example, an approach based on time series clustering was introduced in [15] to discover "unusual"

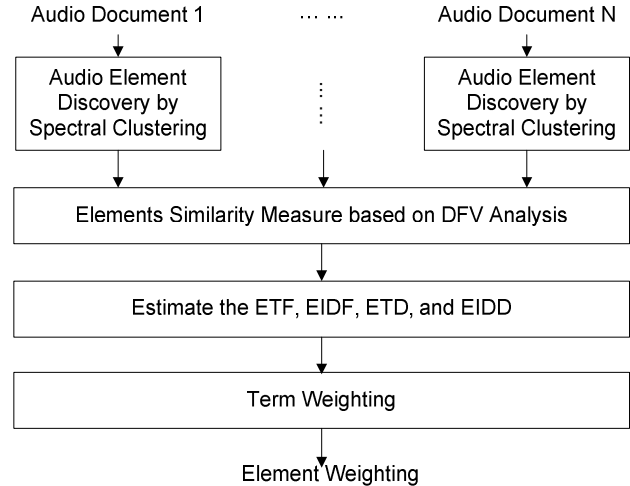
events in audio streams among the outliers of the obtained clusters. A more generic alternative to this method was introduced in [2]. There, spectral clustering is first employed to decompose an audio stream into audio elements, and then, a number of heuristic importance indicators including *element frequency*, *element duration*, *average element length*, and *element length variation*, are defined and employed to filter the obtained set of audio elements and select the key audio elements. The four heuristic importance indicators can be tuned adaptively for different applications, for example, either to detect unusual sounds in surveillance videos, or to detect repetitive characteristic key sounds (e.g. laughter) in situation comedies.

Inspired by the effectiveness of *term frequency (TF)* and *inverse document frequency (IDF)* used for keyword detection in text document analysis, we see the possibility to apply these measures (or their equivalents) to audio documents to improve key audio elements detection in terms of robustness, reliability and level of automation. Actually, a similar idea to *TFIDF* was employed in video summarization [17], where the importance of each video segment is measured based on its rarity and duration. To employ *TF* and *IDF* in audio keyword detection, the number of occurrences of a particular audio “term” in a single document needs to be computed to estimate its *TF* value, while for computing its *IDF* value simultaneous analysis of multiple audio streams needs to be performed. Here, the main difference to text analysis is that an audio term in different parts of a single audio document, and especially in different audio documents, does not necessarily reappear unchanged, but is represented by audio elements that show inevitable variations in their signal properties and duration.

In this paper, we present a new approach to key audio element discovery that, first, computes the similarity between audio elements found in one and multiple audio documents, and then uses the similarity values to compute the probability of the occurrence of one audio term in one and across multiple documents. The obtained probability is used to compute the equivalents for the standard *TF* and *IDF* measures, namely, the *expected term frequency (ETF)* and the *expected inverse document frequency (EIDF)*. In addition, the *expected term duration (ETD)* and *expected inverse document duration (EIDD)* are computed as well, which take into account the discriminative power of the duration of a particular audio element in characterizing the semantics of audio document. For instance, silence intervals in a tennis game are both frequent and long (and semantically important), compared to frequent but short (and not so important) silence intervals between speech segments in the news sound track.

In the proposed approach, as illustrated by the flowchart in Fig. 1, each audio document is first decomposed into audio elements based on iterative spectral clustering, using the methods presented in [2]. Then, the *dominant feature vectors (DFV)* are extracted from each audio element. The DFV are used to measure the similarity between different audio elements, and to estimate the occurrence probability of a given audio term in each audio document. Evaluating the DFV-based audio elements similarity can be considered an equivalent for identifying the matches between words in text that are semantically the same but, for instance, have different endings. This step is necessary as no good analogy to stemming can be found in the audio domain. Finally,

the *ETF*, *EIDF*, *ETD* and *EIDD* measures are computed and combined to give the final importance weight of an audio term. The importance weight is then assigned to all audio elements corresponding to this term, and employed for audio content analysis and discovery tasks.



**Fig. 1 The flowchart of the proposed approach to unsupervised key audio elements discovery using multiple audio documents**

The rest of this paper is organized as follows. Section 2 presents the proposed approach in detail. The section explains the steps of audio element detection, dominant feature vectors extraction, audio element similarity computation, and the calculation of four components of the audio term weight. Section 3 addresses the potential applications of the proposed approach, and explains how the obtained key audio elements can be employed for two major tasks of audio content analysis and discovery, namely auditory scene segmentation and audio document clustering. Experiments evaluating the quality of the obtained key audio elements and their usability for the two abovementioned applications are presented and discussed in Section 4. Section 5 concludes the paper.

## 2. KEY AUDIO ELEMENT DISCOVERY

### 2.1 Audio Stream Decomposition

#### 2.1.1 Features and Audio Segments

Each audio data stream is first divided into frames of 25ms with 50% overlap. Then, a number of audio features are computed to characterize each audio frame. Inspired by previous works on content-based audio analysis [4][6][11], 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 other spectral features proposed in the previous 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.

In order to reduce the computational complexity, we choose to group audio frames into longer temporal audio segments of 1.0 second with 0.5 seconds overlap, by means of a sliding window, and to use these longer segments as the basis for the subsequent audio processing steps. At each window position, the mean and standard deviation of the frame-based features are computed and used to represent the corresponding audio segment.

### 2.1.2 Spectral Clustering

The decomposition of each audio stream is carried out by grouping audio segments into the clusters corresponding to audio elements. Audio elements to be found in complex audio streams (e.g. 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 [7], and such assumption is usually not satisfied in complex cases. As a promising alternative, spectral clustering [13] showed its effectiveness in a variety of complex applications, such as image segmentation [19][20] and the multimedia signal clustering [14][15]. Spectral clustering can be seen as an optimization problem of grouping similar data based on eigenvectors of a (possibly normalized) affinity matrix. 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 [20] to set context-based scaling factors for different data densities. By doing this, we remove the need for the assumption that each cluster in the input data has a similar distribution density in the feature space, which is inherent in the standard spectral clustering algorithm, but usually not satisfied in complex audio data. Moreover, an iterative scheme is used to perform a hierarchical clustering of the input data, in order to further avoid that different audio elements are merged together. More details on our implementation of audio stream decomposition based on spectral clustering can be found in our previous work [2].

## 2.2 Evaluating Similarity of Audio Elements

To take into account possible high-level variations of one and the same audio term (analog to differences in text words due to different endings), and judge which audio elements correspond to the same audio term, we introduce a procedure for measuring the similarity  $S(c_i, c_j)$ , between audio elements  $c_i$  and  $c_j$ , which will be further used to get a reliable indication of audio term reoccurrence.

To measure the similarity between  $c_i$  and  $c_j$ , a possible general approach would be to represent each of them using a Gaussian mixture model (GMM). However, as no assumptions about covariance matrices of GMMs can be made for a general case, computing the distance between GMMs is not likely to be easy. Besides, compared to the similarity computation between audio segments in the spectral clustering step, searching for similarity between audio elements needs to be done with respect to high-level signal descriptors, which will eliminate the influence of irrelevant (low-level) signal variations. We therefore choose for

an alternative approach that employs *Dominant Feature Vectors (DFVs)*.

### 2.2.1 Dominant Feature Vectors

Each audio element usually contains a number of audio segments and thus a number of feature vectors, which usually have complex distribution and multiple salient characteristics. To represent the salient characteristics of an audio element we employ DFVs, which are the principle components in the feature space. The DFVs are computed via the singular value decomposition (SVD) on the feature space of an audio element. Assuming that an audio element contains  $M$  audio segments and each segment is characterized by a feature vector of the length  $N$  (usually  $M \gg N$ ), an audio element can be represented by an  $N \times M$  matrix  $X$ , where each column is a feature vector of the corresponding segment. Thus, using the SVD, the decomposition of  $X$  can be written as

$$X = USV^T \quad (1)$$

where  $U = \{u_1, \dots, u_N\}$  is an  $N \times N$  orthogonal matrix, containing the spectral principle components,  $S = \text{diag}\{\lambda_1, \dots, \lambda_N, 0, \dots, 0\}$  is an  $N \times M$  diagonal matrix of singular values, for which  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  holds, and  $V$  is an  $M \times M$  matrix, presenting the temporal principle components. Those spectral principal components associated with large singular values represent the primary distribution of the audio element in the feature space, and can therefore be adopted as DFVs.

The required number of DFVs describing an audio element is related to the amount of feature variation. In our approach, the number  $m$  of DFVs is chosen as:

$$m = \arg \min_k \left\{ \sum_{i=1}^k \lambda_i / \sum_{i=1}^n \lambda_i > \eta \right\} \quad (2)$$

where the threshold  $\eta$  is set as 0.9 in our experiments.

It should be noted that our approach to DFV extraction is different from traditional PCA applications. While PCA is traditionally used to remove the noisy feature dimensions, our method removes the noisy feature vectors but preserves the dimension of each feature vector. Moreover, dominate feature vectors are extracted to form a *signal subspace*, which represents the most salient characteristics of an audio element. In contrast to this, PCA usually maps feature vectors into the principle feature subspace.

### 2.2.2 Definition of Audio Element Similarity

We now assume to have two audio elements  $c_1$  and  $c_2$ , which contain  $m_1$  and  $m_2$  DFVs respectively. We denote their  $i$ th and  $j$ th DFV as  $q_{c_1, i}$  and  $q_{c_2, j}$ , and the corresponding singular values as  $\lambda_{c_1, i}$  and  $\lambda_{c_2, j}$ , respectively. To measure the similarity between  $c_1$  and  $c_2$ , we first consider the similarity between each pair of their DFVs,  $q_{c_1, i}$  and  $q_{c_2, j}$ , which is usually defined as their inner-product, that is  $s_{i, j} = \|q_{c_1, i}^T q_{c_2, j}\|$ .

Since different DFVs have different importance, which is determined by their corresponding singular values, they should contribute differently to the audio element similarity measure. In order to take this into account, we define the similarity between two audio elements as the weighted sum of the similarity between every pair of their DFVs, that is

$$S = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} w_{i,j} S_{i,j} \quad (3)$$

where the weight  $w_{i,j}$  is determined by the corresponding singular values, as

$$w_{i,j} = \lambda_{c_{1,i}} \lambda_{c_{2,j}} / \sqrt{\sum_{i=1}^{m_1} \lambda_{c_{1,i}}^2 \sum_{j=1}^{m_2} \lambda_{c_{2,j}}^2} \quad (4)$$

The weight is selected as such for the following two reasons:

1. it needs to be proportional to the contributions of the corresponding DFVs, which corresponds to the singular values,  $\lambda_{c_{1,i}}$  and  $\lambda_{c_{2,j}}$ ;
2. the weighted sum (3) should be equal to one, when two audio clips are the same, i.e.,  $q_{c_{1,i}} = q_{c_{2,j}}$  and  $\lambda_{c_{1,i}} = \lambda_{c_{2,j}}$ .

Based on the above, the similarity between two audio elements is now defined as:

$$S_{dfv}(c_1, c_2) = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \lambda_{c_{1,i}} \lambda_{c_{2,j}} \| q_{c_{1,i}}^T \cdot q_{c_{2,j}} \| / \sqrt{\sum_{i=1}^{m_1} \lambda_{c_{1,i}}^2 \sum_{j=1}^{m_2} \lambda_{c_{2,j}}^2} \quad (5)$$

This similarity is symmetric as  $S_{dfv}(c_1, c_2) = S_{dfv}(c_2, c_1)$ , and its value is in the range of  $[0, 1]$ . When the subspaces of  $c_1$  and  $c_2$  are aligned, their similarity is 1; and when the two subspaces are orthogonal with each other, the value is 0.

## 2.3 Audio Element Weighting Scheme

To spot the key audio elements, we draw an analogy to keyword extraction in text document analysis. Assuming that an audio term usually appears frequently in one audio document but seldom in other documents, the criteria like TF and IDF could be used to indicate the importance of that term for characterizing the content of the audio document. To check the reoccurrences of a given audio term, we have to search for audio elements that are sufficiently similar to each other in terms of (5), and that can therefore be said to correspond to one and the same audio term. Due to the missing exact match between audio elements, we can only speak about the probability for reoccurrence of the term, where this probability depends on the value of the similarity measure (5). Based on this probability, the equivalents of the standard TF and IDF measures, namely the *expected term frequency (ETF)* and *expected inverse document frequency (EIDF)*, can be computed, and used to obtain the importance weight of the term.

Another important difference to the text case stems from the fact that audio elements representing one and the same audio term can still show significant variations in their duration. As the duration of the audio elements, just like the number of their occurrences, defines the amount of presence of the corresponding term in an audio document, the overall length of the term in a document is also a parameter that should be taken into account when computing the weight of the term. Further, it can realistically be assumed that the overall duration of a key term is larger in its "own" document than in other documents. To take the duration aspect into consideration, we extend the weight computation scheme to include two additional indicators of term importance, namely the *expected term duration (ETD)* and the *expected inverse document duration (EIDD)*.

### 2.3.1 ETF and ETD

*ETF* and *ETD* define the expected occurrence frequency and occurrence duration of an audio element in one audio stream. Thus, to calculate *ETF* of audio element  $c_i$  in audio stream  $S_k$ , we first need to compute the probability  $P(c_i = c_j)$  for all audio elements  $c_j$  from  $S_k$ . Then, the *ETF* can be obtained as the normalized weighted sum of the occurrence frequencies of all the audio elements  $c_j$  in  $S_k$ , where the abovementioned probabilities serve as the weights:

$$ETF(c_i, S_k) = \frac{\sum_j n_j P(c_i = c_j | c_j \in S_k)}{\sum_j n_j} = \frac{\sum_{c_j \in S_k} n_j S_{dfv}(c_i, c_j)}{\sum_{c_j \in S_k} n_j} \quad (6)$$

where  $ETF(c_i, S_k)$  is the expected term frequency of audio element  $c_i$  in the audio document  $S_k$ . Further,  $P(c_i = c_j | c_j \in S_k)$  is the probability that  $c_i$  represents the same audio term as the audio element  $c_j$ , and is computed using the similarity (5). Finally,  $n_j$  is the number of occurrences of  $c_j$  in the document  $S_k$ . The value of  $n_j$  is simply obtained as the size of the audio segment cluster corresponding to the audio element  $c_j$ . Similarly,  $ETD(c_i, S_k)$  can be defined as,

$$ETD(c_i, S_k) = \frac{\sum_j d_j P(c_i = c_j | c_j \in S_k)}{\sum_j d_j} = \frac{\sum_{c_j \in S_k} d_j S_{dfv}(c_i, c_j)}{\sum_{c_j \in S_k} d_j} \quad (7)$$

where  $d_j$  is the total occurrence duration of  $c_j$  in the document  $S_k$ .

### 2.3.2 EIDF and EIDD

Similar to *IDF* in text document analysis, *EIDF* of an audio element  $c_i$  can be computed as the log of the number of all documents divided by the expected number of documents containing the audio element  $c_i$ . That is,

$$EIDF(c_i) = \log \frac{|S|}{\sum_k P(c_i \in S_k)} \quad (8)$$

where  $|S|$  is the number of documents and  $P(c_i \in S_k)$  is the probability that  $c_i$  appears in the document  $S_k$ . This probability can be calculated as

$$\begin{aligned} P(c_i \in S_k) &= P(c_i = c_{j_1} \cup c_i = c_{j_2} \cup \dots \cup c_i = c_{j_N} | c_{j_1}, \dots, c_{j_N} \in S_k) \\ &= 1 - \prod_j (1 - P(c_i = c_j | c_j \in S_k)) \\ &= 1 - \prod_j (1 - S_{dfv}(c_i, c_j)) \quad (c_j \in S_k) \end{aligned} \quad (9)$$

It is easy to verify that  $P(c_i \in S) = 1$  if the audio element  $c_i$  is obtained from the document  $S$ .

Similarly, the expected inverse document duration (*EIDD*) of audio element  $c_i$  can be calculated as the log of the duration of all documents divided by the expected duration of audio element  $c_i$  appears in all documents. As the expected duration of audio element  $c_i$  in document  $S_k$  is obtained by  $ETD(c_i, S_k)$ , the *EIDD* can be approximated as,

$$EIDD(c_i) = \log \frac{\sum_k D_{S_k}}{\sum_k ETD(c_i, S_k)} \quad (10)$$

where  $D_{S_k}$  is the total duration of audio document  $S_k$ .

### 2.3.3 Final Weighting

To integrate the above four importance indicators into the definitive importance weight of an audio term, we simply combine the indicators into the product

$$W(c_i, S_k) = ETF(c_i, S_k) \cdot EIDF(c_i) \cdot ETD(c_i, S_k) \cdot EIDD(c_i) \quad (11)$$

Audio elements having assigned high values of the weight (11) can be considered key audio elements, and used to characterize audio document in further audio content analysis and discovery processes.

## 3. POTENTIAL APPLICATIONS

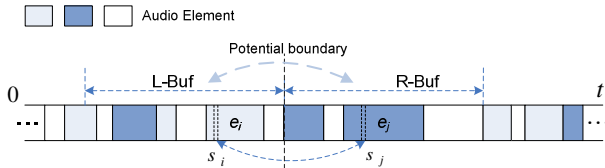
By drawing the analogies to text content analysis, effective methods for audio content analysis and discovery can be defined that employ audio elements and their weights. In this section, we will outline the possibilities for developing such methods for the tasks of auditory scene segmentation, audio document clustering, and audio retrieval using the query-by-example approach, which were seldom addressed in previous literatures. Since these applications and methods are only used to illustrate how to utilize the obtained audio elements and the corresponding weights, their descriptions are kept concise. We will, however, briefly evaluate the effectiveness of these methods for two of the abovementioned applications, that is, auditory scene segmentation and audio document clustering. This evaluation is reported in Section 4.

### 3.1 Auditory Scene Segmentation

One of the basic audio content analysis tasks is auditory scene segmentation, which divides an audio stream into semantically coherent temporal segments, and provides the basis for scene clustering or classification. An auditory scene can be seen as a series of semantically related audio segments. Fig. 2 shows an example sequence of audio segments obtained from an audio stream. Here, audio segments are the same as those mentioned in Section 2.1 - 1 second in length, with 0.5 seconds overlap - and are used as basic units for audio element discovery by spectral clustering. Each time stamp separating two audio segments can be considered a potential auditory scene boundary. Thus, the confidence  $C(t)$  of having a scene boundary at the observed time stamp  $t$  can be obtained by computing the semantic affinity between the audio segments surrounding the observed time stamp:

$$C(t) = \frac{1}{N_l N_r} \sum_{i=1}^{N_l} \sum_{j=1}^{N_r} A(s_i, s_j) \quad (12)$$

where  $N_l$  and  $N_r$  are the numbers of segments left and right from the potential boundary and captured by the intervals  $L\text{-Buf}$  and  $R\text{-Buf}$ . Further,  $s_i$  and  $s_j$  are audio segments, and  $A(s_i, s_j)$  represents the semantic affinity between them.



**Fig. 2 Illustration of an approach to audio element based auditory scene segmentation**

The definition of the semantic affinity between the segments  $s_i$  and  $s_j$  can be based on the following assumptions [9]:

1. there is a high affinity between two segments if the corresponding audio elements usually occur together;
2. the larger the time interval between two audio segments, the lower their affinity; and
3. the larger the weights of the corresponding audio elements, the more significant is their role in computing the inter-segment affinity.

In view of the above assumptions, the semantic affinity can be computed using the following function:

$$A(s_i, s_j) = Co(e_i, e_j) e^{-T(s_i, s_j)/T_m} P_{e_i} P_{e_j} \quad (13)$$

Here, the notation  $e_i$  and  $e_j$  is used to indicate the audio element identities of the segments  $s_i$  and  $s_j$ , that is, to describe their content (e.g. speech, music, noise). Further,  $P_{e_i}$  and  $P_{e_j}$  are the importance weights of audio elements  $e_i$  and  $e_j$ .  $T(s_i, s_j)$  is the time interval between the audio segment  $s_i$  and  $s_j$ , and  $T_m$  is a scaling factor. Introduction of the exponential term in (13), as well as the selection of the value for  $T_m$  (set to 16 seconds in our experiments) is inspired by the previous work on human memory limit [16].  $Co(e_i, e_j)$  measures the co-occurrence of audio elements  $e_i$  and  $e_j$ , and can be computed as

$$Co(e_i, e_j) = e^{-\frac{D_{ij} + D_{ji}}{2\mu_D}} \quad (14)$$

where  $D_{ij}$  is the average time interval between audio elements  $e_i$  and  $e_j$ , and  $\mu_D$  is the average of all  $D_{ij}$  and  $D_{ji}$ . Exponential formula is chosen here to keep the influence of co-occurrence on the overall semantic affinity comparable with the influence of the time interval between the segments. Auditory scenes can now simply be obtained by applying a threshold to the values of the confidence curve (12).

### 3.2 Audio Document Clustering and Retrieval

A fundamental step in obtaining meaningful clusters of audio documents is document representation. While for clustering short audio clips (e.g. for clustering audio segments into speech, music and noise), such representation can be obtained in terms of low-level features, this is not likely to work in the case of longer audio documents due to the richness of signal mixtures and strong variations in signal properties over time. Clearly, a more sophisticated representation scheme needs to be found for clustering long audio documents, which reveals their high-level similarity and neglects irrelevant signal variations. Interestingly, no previous work addressing this problem can be found in recent literatures.

Again, analog to text document clustering, where each document is represented by a vector of words and their weights, an audio document  $S_k$  can be represented by a vector containing all the audio elements and their weights. Using such vectors as inputs in the clustering process, grouping of audio documents will be guided by the similarity between their most significant audio elements (that is, audio “keywords”).

The obtained representative vectors can also be used in audio document retrieval to match the query audio document with the audio documents in a database. Various distances proposed in text analysis can be employed to measure the similarity between the query and the documents in the database. Moreover, a sequence of audio elements can be further used to build the inverse index for the document database.

## 4. EVALUATION AND DISCUSSION

In this section, we present the results obtained by evaluating the proposed method for audio element weighting, and the effectiveness of auditory scene segmentation and clustering approaches based on weighted audio elements.

### 4.1 Database Information

For our experiments we used sound tracks extracted from various types of video, including sports, situation comedy, award ceremony, and war/action movies, and in the total length of about 5 hours. These sound tracks contain an abundance of different audio elements, and are of different complexity in terms of the number and sequencing of audio elements contained therein. For example, the sound track of the tennis match has a relatively simple structure, compared to far more complex sound tracks from the war/action movies "Band of Brothers - Carentan" and "Sword Fish".

**Table 1. Information of the experimental audio data**

No.	Video	category	duration
A <sub>1</sub>	<i>Friends</i>	situation comedy	0:25:08
A <sub>2</sub>	<i>59<sup>th</sup> Annual Golden Globe Awards</i>	award ceremony	1:39:47
A <sub>3</sub>	<i>Tennis Game</i>	sports	0:59:41
A <sub>4</sub>	<i>Band of Brothers - Carentan</i>	war movie	1:05:19
A <sub>5</sub>	<i>Sword Fish</i>	Action movie	1:00:00

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 Weighting

Based on the spectral clustering described in Section 2.1, each audio stream is decomposed into various audio elements. A semantic meaning is further associated to each audio element by combining the results obtained by three unbiased persons who analyzed the content of the sound track and the obtained audio elements. The performance of audio elements discovery has been presented in [2]. This section presents and discusses the results of audio elements weighting, as collected in Table 2 to 6.

The last column in Table 2 to 6 lists the importance weights of the identified audio elements in the sound tracks A<sub>1</sub> - A<sub>5</sub>. Based on these weights, we can do an "educated guess" about which audio elements should be considered the most important or the most representative per audio document. For example, in the "Friends" (A<sub>1</sub>) soundtrack, the top three key audio elements found by our weighting approach are *laughter*, *applause with cheer*, and *laughter with music*. In the "Golden Global Awards" (A<sub>2</sub>) soundtrack, the top three key audio elements found are *applause*, *music with applause 1* and *music with applause 2*, as indicated in the highlighted table rows. As also confirmed by our test panel consisting of three subjects, these high-weighted audio elements indeed correspond to the most representative sounds in these sound tracks.

In "Tennis" (A<sub>3</sub>), the *applause* and *ball-hit*, which are the two game-specific sounds in a tennis match, are correctly assigned high weights. It is noted that two *silence* elements found in this soundtrack are assigned the highest weights (the silence segments between every two ball-hits are clustered together). This is justifiable since *silence* periods are very representative for the game and also are not that pronounced in other sound tracks in the test set.

The war and action movie (A<sub>4</sub> and A<sub>5</sub>) are more complex and thus many more audio elements are discovered. Our results also show that some movie-specific sounds are ranked high, such as the *gunshots*, some *fighting sounds*, some specific *background sounds*, and some background *music* that enhances a tense atmosphere.

The varying diversity of content found in test sequences is also the reason why the obtained weights usually have different scales in different documents. For example, the maximum weight in sound track A<sub>1</sub> is up to 0.96, while in A<sub>4</sub> and A<sub>5</sub>, the maximum weights are only 0.137 and 0.172, respectively.

The weights in Table 2 to 6 are computed based on the *ETF*, *EIDF*, *ETD*, and *EIDD* values that are obtained per audio element and listed in column 5 to 8 of each table. For completeness, we also listed the total number of occurrences (*occu*) and total duration (*dur*) of each audio element. By observing the values in columns 3 to 8, situations can be analyzed that led to a particular weight. For example, the 6<sup>th</sup> audio element "applause with cheer" in Table 2, although it occurs only once and lasts only 5 seconds in this track, it occurs statistically even less in other audio tracks, which makes its *EIDF* (2.15), *EIDD* (2.135) and the final weight high. On the other hand, the 5<sup>th</sup> audio element "music with speech" and the 11<sup>th</sup> audio element "speech" in Table 3, although they appear many times and have long durations (161 times / 900 seconds, and 487 times / 2894 seconds, respectively), they seem to appear often in other soundtracks as well. Thus, their *EIDF*, *EIDD* and the final weight are low. These results show that the TF and IDF concepts from text analysis are indeed applicable to general audio signals.

**Table 2. Audio element weighting on the track of "Friends" (A<sub>1</sub>)**

No	Description	occu.	dur.	ETF	EIDF	ETD	EIDD	weight
1	speech + noise	27	44.5	0.59	0.588	0.691	1.046	0.251
2	laughter	102	218.0	0.699	1.411	0.61	1.597	0.96
3	theme music	1	47.0	0.236	1.466	0.501	1.582	0.274
4	laughter + music	9	42.5	0.515	1.234	0.525	1.421	0.474
5	Speech	124	1148.5	0.785	0.674	0.897	0.967	0.459
6	applause + cheer	1	5.0	0.496	2.15	0.392	2.135	0.892
7	TV music	1	3.0	0.036	1.711	0.038	2.587	0.006

**Table 3. Audio element weighting on the track of "Golden Global Awards" (A<sub>2</sub>)**

No	Description	occu.	dur.	EDF	IDF	EDD	IDD	weight
1	clear speech 1	132	202.5	0.646	0.249	0.705	0.81	0.092
2	clear speech 2	26	27.5	0.607	0.304	0.669	0.91	0.112
3	music + applause 1	110	346.0	0.713	0.395	0.691	0.767	0.149
4	music + applause 2	72	168.5	0.681	0.45	0.654	0.832	0.167
5	music + speech	161	900.5	0.752	0.158	0.795	0.543	0.051
6	Music	22	50.5	0.512	0.349	0.544	0.784	0.076
7	applause	143	485.5	0.506	1.043	0.458	1.374	0.332
8	speech + applause	109	229.5	0.705	0.358	0.708	0.802	0.143
9	background noise	211	423.5	0.747	0.216	0.739	0.622	0.074
10	(dense) music + applause	68	260.0	0.622	0.363	0.623	0.814	0.114
11	speech	487	2893.5	0.776	0.161	0.829	0.581	0.06

**Table 4. Audio element weighting on the track of "Tennis" (A<sub>3</sub>)**

No	Description	occu.	dur.	EDF	IDF	EDD	IDD	weight
1	clear speech	250	1658.0	0.576	0.177	0.66	0.639	0.043
2	speech	108	341.0	0.555	0.304	0.571	0.779	0.075
3	music	1	22.0	0.431	0.651	0.409	1.135	0.13
4	applause	106	319.5	0.42	1.194	0.358	1.464	0.262
5	silence	173	837.5	0.533	0.934	0.491	1.262	0.308
6	(noisy)silence	32	96.5	0.465	1.117	0.404	1.452	0.304
7	ball-hit	145	307.5	0.641	0.54	0.598	0.92	0.19

**Table 5. Audio element weighting on the track of "Band of Brother" (A<sub>4</sub>)**

No	Description	occu.	dur.	EDF	IDF	EDD	IDD	weight
1	speech	187	557	0.684	0.132	0.621	0.568	0.032
2	speech (gun background)	25	62.0	0.624	0.188	0.588	0.591	0.041
3	speech	1	13.0	0.157	1.446	0.151	2.108	0.072
4	speech	72	192.0	0.662	0.119	0.601	0.529	0.025
5	heavy noise	11	24.5	0.501	0.396	0.447	0.921	0.082
6	silence (some noise)	44	127.0	0.438	0.636	0.37	1.173	0.121
7	noise	143	506.5	0.648	0.218	0.579	0.609	0.05
8	speech	122	341.5	0.667	0.125	0.611	0.541	0.027
9	gun + speech	128	714.0	0.4	0.704	0.402	1.069	0.121
10	gun + speech	85	213.5	0.279	1.225	0.291	1.381	0.137
11	background sounds	51	177.5	0.452	0.607	0.384	1.208	0.127
12	applause	3	16.5	0.263	0.912	0.239	1.493	0.085
13	music	48	734.5	0.42	0.517	0.482	1.022	0.107
14	music + speech	4	45.0	0.423	0.475	0.447	1.014	0.091
15	noise + speech	86	153.0	0.706	0.158	0.658	0.544	0.04
16	silence (with sparkle on high frequency)	3	22.5	0.23	0.926	0.225	1.615	0.077

**Table 6. Audio element weighting on the track of "Sword Fish" (A<sub>5</sub>)**

No	Description	occu.	dur.	EDF	IDF	EDD	IDD	weight
1	speech + backgrounds	235	449.5	0.811	0.176	0.795	0.514	0.058
2	fighting sounds	57	228.0	0.31	1.017	0.334	1.557	0.164
3	similar to above	155	453.0	0.645	0.242	0.645	0.72	0.073
4	speech + backgrounds	86	242.0	0.593	0.413	0.573	0.982	0.138
5	speech	241	502.5	0.759	0.107	0.74	0.507	0.031
6	mixed backgrounds	21	44.0	0.528	0.562	0.524	1.107	0.172
7	speech	363	1005.5	0.793	0.115	0.771	0.531	0.037
8	speech + backgrounds	73	114.5	0.782	0.101	0.763	0.534	0.032
9	speech + backgrounds	121	209.0	0.692	0.197	0.67	0.585	0.053
10	backgrounds	404	1266.5	0.737	0.222	0.725	0.587	0.07
11	speech in repressive env.	67	129.5	0.413	0.566	0.401	1.042	0.098
12	music	13	69	0.506	0.468	0.497	0.962	0.113
13	fighting sounds	76	155.5	0.345	0.877	0.34	1.347	0.138
14	backgrounds	102	217.5	0.604	0.448	0.588	0.929	0.148
15	speech + backgrounds	247	445.0	0.818	0.16	0.799	0.559	0.058
16	music	45	311.5	0.601	0.388	0.609	0.83	0.118
17	music	23	115.0	0.394	0.653	0.402	1.248	0.129

### 4.3 Applications

In this section, we briefly evaluate the performance of some applications where the obtained audio elements and their weights are employed. We mainly concentrate on the methods for auditory scene segmentation and audio document clustering as discussed in Section 3.

#### 4.3.1 Auditory Scene Segmentation

In order to have an objective evaluation of auditory scene segmentation, we first need to create the ground-truth for this task. In the experiments, we observed some clues in different sound tracks (with corresponding visual information) and used them as the basis for auditory scene annotation. For example, award ceremony (A<sub>2</sub>) basically consists of a series of different events, like when the host announces the nominees and the winner, or when the winner approaches the stage while the audience is applauding; in Tennis (A<sub>3</sub>), each process from serve to score can be considered as a scene; in Friends (A<sub>1</sub>), some black frames are good boundary indicators of different scenes; and in the action and war movies (A<sub>4</sub> and A<sub>5</sub>), the changes of visual scenes and the change of days are also considered as good scene boundary indicators.

In view of this, and with help of three unbiased persons, we selected the true auditory scene boundaries at the break points between different events. Moreover, we also annotated a number of ‘probable boundaries’ at places where the presence of a true boundary is unclear. For example, in award ceremony, the turn between the played ‘nominated movies’ when a host announces the nominees, can be seen as a probable boundary. In total, we obtained 295 true boundaries and 186 probable boundaries from 5 sound tracks.

In the experiments, 411 boundaries are detected. A detected boundary is associated with an annotated boundary if they are

sufficiently close to each other. Two approach variants are also implemented for comparison, including low-level feature based approach and audio element based approach without using the weight. Both approaches follow the main idea described in Section 3.1 but (13) should be revised accordingly. In the former approach, since we did not detect the audio elements, the similarity between two audio elements in (13) which is measured by co-occurrence should be replaced with the low-level feature similarity between two audio frames. Moreover, both approaches need not consider the weights of audio elements.

Table 7 shows the overall comparison results among different approach variants. The table lists the recall of ‘ground-truth’ boundaries (R1), the recall of ‘probable’ boundaries (R2), the precision (P) and the corresponding F1 measure. F1 is usually a harmonic average of recall and precision, and is used to evaluate the overall performance of a system. In our evaluation, since we have two kinds of recalls, we slightly change the definition of F1, as,

$$F1 = \frac{2(R1 + R2)P}{R1 + R2 + P} \quad (15)$$

**Table 7. The results of the auditory scene segmentation, comparing with different approach variants**

Approach	R1	R2	P	F1
Low-level features	0.71	0.67	0.62	0.86
Element [no weight]	0.72	0.50	0.74	0.92
Element [tf.idf]	0.71	0.60	0.76	0.96

The results show that using audio elements instead of low-level features improves the performance of auditory scene segmentation: the F1 measure shows an increase of 6%. As expected, the low-level feature based approach resulted in a substantial number of false alarms, which led to the lowest precision value. Finally, if

our proposed weights are assigned to audio elements, the F1 measure shows further improved of 4%. This result can be seen as a further (indirect) indication of the effectiveness of the proposed audio element weighting approach.

Detailed results obtained for each sound track are listed in Table 8, in which the recall and precision are not computed as percentages, but indicated by the ratios X/Y of the relevant counts. Thus, it shows more details on there are how many ground-truths, and how many are correctly or falsely detected. For example, in the first line of the R1 column, the ratio 9/17 says that 9 out of 17 boundaries are correctly detected, while 24 / 32 in P column says that 32 boundaries are detected, of which 24 are correct. The results in Table 8 generally support the conclusions we already drew based on the results from Table 7.

**Table 8. Exhaustive results list of the auditory scene segmentation on various sound tracks**

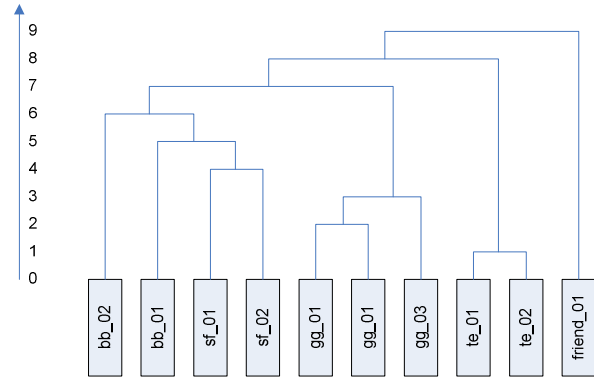
No	Approach	R1	R2	P
A <sub>1</sub>	Low-level features	9 / 17	15 / 25	24 / 32
	Element [no weight]	11 / 17	8 / 25	19 / 22
	Element [tf.idf]	11 / 17	14 / 25	25 / 30
A <sub>2</sub>	Low-level features	76 / 96	39 / 60	115 / 175
	Element [no weight]	72 / 96	29 / 60	101 / 126
	Element [tf.idf]	65 / 96	41 / 60	106 / 131
A <sub>3</sub>	Low-level features	62 / 94	11 / 16	73 / 106
	Element [no weight]	70 / 94	6 / 16	76 / 93
	Element [tf.idf]	69 / 94	9 / 16	78 / 95
A <sub>5</sub>	Low-level features	28 / 37	25 / 38	53 / 106
	Element [no weight]	29 / 37	23 / 38	52 / 87
	Element [tf.idf]	29 / 37	15 / 38	44 / 70
A <sub>5</sub>	Low-level features	33 / 51	35 / 47	68 / 114
	Element [no weight]	30 / 51	26 / 47	56 / 83
	Element [tf.idf]	35 / 51	32 / 47	67 / 93

### 4.3.2 Audio Document Clustering

In this section, we present a preliminary evaluation of audio document clustering. To have a sufficient number of audio documents to cluster, we first split each test audio stream from Table 1 into the parts of about 30-minutes in length. Thus, we artificially create more audio documents (in our case, we obtain 10), and each document has one or more corresponding similar documents. Therefore, the audio stream and the audio category that the audio documents originally come from can be taken as ground truth for the clustering. That is to say, good clustering results should first cluster the audio documents from the same source and the same category.

Based on the document representation proposed in Section 3.2, a hierarchical agglomerative clustering algorithm is employed on the 10 audio documents. In the clustering algorithm, each audio document initially represents one cluster and then two most similar clusters are merged together at each iteration. The detailed clustering process is shown in the Fig. 3, where the numbers at left show the number of iterations. Also the abbreviations of the

document names were used in the leaves. For example, “bb\_01” is the first 30-min part from “Band of Brother”, “sf\_02” is the second 30-min part from “Sword Fish”, and so on.



**Fig.3 Hierarchical agglomerative clustering of 10 audio documents**

From the figure, it can be seen that most documents were clustered correctly. For example, sf\_01 and sf\_02, gg\_01, gg\_02, and gg\_03, te\_01 and te\_02 are clustered within the first four iterations, since they belong to the same sound track. One exception is bb\_01. It is firstly clustered with sf\_01 and sf\_02, and then with bb\_02. This becomes understandable if one realizes that these segments are from action/war movies, and the corresponding audio elements are very similar in these audio documents. These results indicate that the audio document representation based on audio elements and their weights proposed in this paper, is likely to lead to good audio document clustering results.

## 5. CONCLUSIONS

In this paper, a novel unsupervised approach is proposed to weight various audio elements discovered in an audio document, based on their re-occurrence in multiple audio documents. The obtained weights have the purpose of revealing important audio elements as content descriptors that can be used for effective audio content analysis and discovery.

In the proposed approach, *Dominant Feature Vectors* are first extracted from audio elements and used to measure their similarity. The similarity value is employed to estimate the re-occurrence of audio terms in the test corpus. Then, four importance indicators are extracted per audio term, including *expected term frequency*, *expected inverse document frequency*, *expected term duration*, and *expected inverse document duration*, and combined to give the importance weight of the term. These importance indicators can be seen as equivalents for the TF and IDF measures in text analysis.

Experiments evaluating the quality of the obtained key audio elements and their usability for two major steps in audio content analysis and discovery, namely auditory scene segmentation and audio document clustering, show promising results. However, the proposed solution for audio element weighting still leaves some room for further investigation and improvement. This especially holds for the computation of the audio element similarity and importance indicators, which could be made more robust with

respect to the diversity of audio elements belonging to the same audio term. More robustness will lead to a better estimate of the re-occurrence of the term and thus to more reliable weights. In our future work we also intend to evaluate the applicability of this technology in more applications, such as audio retrieval.

## 6. REFERENCES

- [1] Baeza-Yates, R., and Ribeiro-Neto, B. *Modern Information Retrieval*. Addison-Wesley, Boston, MA, 1999.
- [2] Cai, R., Lu, L., Hanjalic, A. Unsupervised Content Discovery in Composite Audio. *Proc. ACM Multimedia 05*, pp. 628-637, 2005
- [3] 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.
- [4] Cai, R., Lu, L., Zhang, H.-J., and Cai, L.-H. Highlight sound effects detection in audio stream. In *Proc. of the 4<sup>th</sup> IEEE International Conference on Multimedia and Expo*, 2003, vol. 3, 37-40.
- [5] Cai, R., Lu, L., Zhang, H.-J., and Cai, L.-H. Improve audio representation by using feature structure patterns. In *Proc. of the 29<sup>th</sup> IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004, vol. 4, 345-348.
- [6] Cheng, W.-H., Chu, W.-T., and Wu, J.-L. Semantic context detection based on hierarchical audio models. In *Proc. of the 5<sup>th</sup> ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2003, 109-115.
- [7] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification, Second Edition*. John Wiley & Sons, NJ, 2000.
- [8] Liu, Z., Wang Y. and Chen, T. Audio Feature Extraction and Analysis for Scene Segmentation and Classification. *Journal of VLSI Signal Processing Systems*, June 1998
- [9] Lu, L., Cai, R., and Hanjalic, A. Audio Elements based Auditory Scene Segmentation. *Proc. ICASSP06*, 2006.
- [10] Lu, L., Cai, R., and Hanjalic, A. Towards a unified framework for content-based audio analysis. In *Proc. of the 30<sup>th</sup> IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005, vol. 2, 1069-1072.
- [11] 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.
- [12] Moncrieff, S., Dorai, C., and Venkatesh, S. Detecting indexical signs in film audio for scene interpretation. In *Proc. of the 2<sup>nd</sup> IEEE International Conference on Multimedia and Expo*, 2001, 989-992.
- [13] 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.
- [14] 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.
- [15] 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 6<sup>th</sup> ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2004, 157-164.
- [16] Sundaram, H., and Chang, S.-F. Determining Computable scenes in films and their structures using audio visual memory models. In *Proc. of the 8<sup>th</sup> ACM International Conference on Multimedia*, 2000, 95-104.
- [17] Uchihashi S., Foote, J., Girgensohn A., and Boreczky J. Video Manga: Generating Semantically Meaningful Video Summaries. *Proc. ACM Multimedia 99*, pp. 383-392, 1999
- [18] 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 4<sup>th</sup> IEEE International Conference on Multimedia and Expo*, 2003, vol. 2, 281-284.
- [19] Yu, S. X., and Shi, J. Multiclass spectral clustering. In *Proc. of the 9<sup>th</sup> IEEE International Conference on Computer Vision*, 2003, vol. 1, 313-319.
- [20] Zelnik-Manor, L., and Perona, P. Self-tuning spectral clustering. *Proc. Advances in Neural Information Processing Systems 17 (NIPS 2004)*, 1601-1608.