Audiotory Movie Summarization by Detecting Scene Changes and Sound Events

Tong Lu, Yangbing Weng, Gongyou Wang
National Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China 210023
Department of Computer Science and Technology, Nanjing University
Email: lutong@nju.edu.cn

***Abstract*—A novel movie audio summarization framework is presented, which consists of three processing levels, namely, low-level audio feature extraction, mid-level audio event detection, and high-level auditory movie summarization. We first detect auditory changes in the eigen-audiospace to segment movie audio streams, then exploit a scoring algorithm to refine the segments. Audio events from movies are identified in a hierarchical manner from background detection, foreground event separation to key audio event identification, thus generating the final audio summaries from each movie. The experiments on different auditory categories from movies and TVs demonstrate the effectiveness of the propose approach.**

Keywords—Auditory movie summarization; framework; audio segmentation

I. Introduction

Recently, the rapid increase in speed and capacity of smart devices equipped with acoustic sensors has allowed the inclusion of sound as a useful type of data in content-aware applications. As a relatively inexpensive media comparing with vision-based tools which require solving the difficulties from obstruction, weak lighting to multi-view observation, the ever-growing volume of audio data has revealed a strong commercial demand for developing efficient content-oriented algorithms, which aim at obtaining useful audio information, grouping audio data into meaningful categories, or providing audio browsing and retrieval [1, 8]. Defined as the extraction of meaningful information from audio streams, audio content summarization can be beneficial to the existing audio-related real-life applications.

Three leading approaches have been investigated to detect and extract the semantic contents from audio streams. *Harmonic structural analysis approach* typically deals with speech [2, 5] and music [3], which are structured sounds due to their formant characteristics. For example, in [4], a clue of self-similarity is presented for structural analysis to detect the repetitive patterns that are suitable for content-based audio summarizing. However, these methods cannot be directly applied to analyze unstructured sound signals which have a diverse variety to build models. *Feature extraction approach* considers the task of recognizing multisource sounds by analyzing and selecting a variety of audio features. For example, Chu *et al.* [6] perform an empirical feature analysis for audio environment characterization and propose to use the matching pursuit (MP) algorithm to obtain effective time-frequency features. Kyperountas *et al.* [7] create an enhanced set of eigen-audioframes that is related to an audio signal subspace to discover audio background changes. Unfortunately, how to bridge the semantic gap that separates low-level auditory features and high-level auditory contents is still difficult even by combining more heterogeneous features. *Audio representation approach* uses statistic descriptor as the classifier to recognize multisource sounds. In [9], each audio stream is modeled with a histogram which is estimated from annotated training data. The cosine distance between such a histogram and each test individual sound is then calculated to perform audio content recognition. Eronen *et al.* [10] investigate the feasibility of an audio recognition system by extracting simplistic low-dimensional feature vectors and using low-order hidden Markov models. However, the meaningful contents inside a multisource sound stream are not explored systematically.

In this paper, a novel movie sound summarization method is proposed. A movie sound stream here is in general simultaneously composed of structural foreground and unstructured background sounds from a variety of sources. Such a stream can be a clip cut from a movie or a sports broadcast, or directly the record of a person's daily life environment. It is typically noise-like with a broad flat spectrum and sometimes includes strong temporal domain signatures. Our method first locates auditory changes in the eigen-audiospace to obtain auditory segmentations, and then refines the segments through clustering and scoring. Considering the complexity and the unpredicted compositions inside a multisource sound stream, audio events are further identified through a hierarchical framework consisting of background detection, foreground event recognition and key event identification.

Our main contribution is to explore high-level content analysis on complex multisource movie sounds through a hierarchical structure. Considering recognition of noisy multisource sounds is essentially an open and challenging problem due to the fact that no assumptions can be made in advance about the harmonic structure in the signal, our method provides an effective approach to analyze a diverse variety of sound signal compositions from movies. The experiments on the samples of different movie and broadcast categories show the effectiveness of our approach.

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.140

IEEE computer society

II. The Proposed Framework

The framework of our approach is shown in Figure 1, which contains three levels of processing: low-level audio feature extraction, mid-level audio event separation, and high-level auditory movie summarization. Audio event is the core component of our movie summarization framework and is characterized by three properties: start/end times, category and saliency.

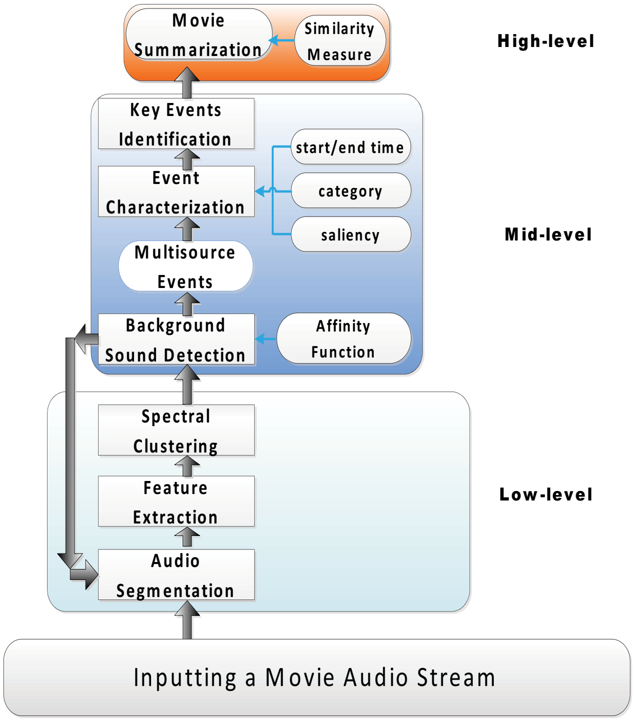

Figure 1: The proposed auditory movie summarization framework.

A. Audio segmentation

Considering the diverse variety of multisource sound signal compositions, we achieve audio segmentation by first projecting it into the eigen-audiospace and then searching for the local minima of the normalized distance between every audioframe and a zero-mean base-line one.

Specifically, let $X = \{x_i\}_{i=1,\ldots,M}$ be an $N \times M$ matrix with each column x_i being a zero-mean and non-overlapping audioframe consisting of N samples that are obtained by subtracting the mean audioframe from the input auditory movie stream I. Considering audio contents to be analyzed typically correspond to larger eigenvalues, we apply PCA to X and choose a set of eigenvectors with the smallest eigenvalues to form an $N \times K$ eigen-audiospace projection matrix E_b. Then X is projected onto the eigenspace with rank K by:

$$T = E_b^T X \tag{1}$$

On the other hand, let v_0 be the mean background that is calculated by empirically averaging all the known multisource background sound samples in the input movie or directly spotted by the user, for any given audioframe v_0, we subtract the mean background from it and thus obtain $\bar{v}_0$. Next, we project $\bar{v}_0$ to the eigen-audiospace by

$$v = E_b^T \bar{v}_0 \tag{2}$$

To detect segmentation points, the normalized distance between v and any projected audioframe t_i can be measured by

$$D(i) = \frac{\left\| t_i - v \right\| - min}{max - min}, i = 1,\ldots, M \tag{3}$$

where *min* and *max* are the minimum and the maximum of $\| t_i - v \|$, respectively.

B. Auditory summarization

In this section, we explore high-level content analysis of complex movie sounds by hierarchically modeling event structures.

1. Clustering audio segments

After obtaining audio segments, we employ the spectral clustering algorithm to identify the characteristic distributions in the feature space by an unsupervised manner. Considering MFCC (Mel-Frequency Cepstral Coefficients) and LDB (Local Discriminatory Base) features have been proved efficient for environmental sound classification, we employ the following steps on the MFCC and LDB features extracted from the audio segments:

1) Suppose an audio segment e_i and its feature matrix $M_i = \{k_{i1},\ldots, k_{in}\}$, where k_{ij} represents its j^{th} feature vector, we apply spectral clustering algorithm on the features of M_i to assign each $k_{ij}\,(j = 1,\ldots, n)$ to one of the K clusters $c_k\,(k = 1,\ldots, K)$;

2) Denoting the number of the elements in cluster c_k by N_k, we assign the audio segment e_i to cluster c_k if N_k is the largest one in all $N_j\,(k = j,\ldots, K)$ and thus obtain the cluster $C = \{c_1,\ldots, c_k\}$ as a result.

2. Background identification

Next, we identify background sound clusters in the movie stream. We have the following two observations to help achieve this task: 1) there is in general a high affinity for a background cluster if the segment duration is longer than the others, and 2) the larger the segment length varies in a cluster, the larger is the affinity.

Obeying these two observations, we define an affinity function to identify the clusters in C to characterize background sounds in the movie stream:

$$aff(c_i, I) = exp\,(d_i - d_{avg})^2 /((2d_{std}^2) \cdot exp\,(v_i / l_i)) \quad (4)$$

where d_i is the total length of c_i, d_{avg} and d_{std} are respectively the mean and the standard deviation of all the audio clusters, while v_i and l_i denote the standard deviation and the mean length of the clusters in c_i, respectively.

3. *Foreground audio events separation*

After spotting background c_b from C, supposing it contains altogether p sound segments in the form of $c_b = \{m_{b1},..., m_{bp}\}$, we compute a new average background audioframe $v_{new} = \sum_{j=1}^{p} m_{bj} / p$. We thereby replace the initially spotted audioframe v_0 with v_{new} and recalculate the normalized distance of (3) to obtain updated audio segments again. Then spectral clustering algorithm is similarly carried out again to calculate updated clusters, which are considered as meaningful foreground audio events $E = \{e_1,..., e_m\}$ in the input movie audio stream.

4. *Key audio events identification*

Given the extracted foreground events, we wish to detect the most salient sounds in them for auditory movie summarization. For this purpose, we define the saliency measure by a scoring function, which integrates the occurrence frequency, the total durations and the average lengths of foreground audio events as follows:

$$score\,(e_i, I) = frq\,(e_i, I) \cdot dur\,(e_i, I) \cdot len\,(e_i, I) \quad (5)$$

where

$$frq\,(e_i, I) = exp\,(-(n_i - n_{avg})^2 /(2n_{std}^2)) \quad (6)$$

$$dur\,(e_i, I) = exp\,(-(d_i - d_{avg})^2 /(2d_{std}^2)) \quad (7)$$

$$len\,(e_i, I) = exp\,(-(l_i - l_{avg})^2 /(2l_{std}^2)) \quad (8)$$

n_i, d_i and l_i are the occurrence number, the total duration and the average segment length of an audio event e_i, $*_{avg}$ and $*_{std}$ denote the mean value and the standard deviations, respectively. We thus call those events that have relatively large scores as key audio events.

5. *Post-processing*

We then use a post-processing step to further extend the detected key audio events. This is necessary because a key audio event is probably short, which makes the summarization results consisting of too many short audio events sometimes meaningless for human perception. Our audio extension strategy is based on the following two assumptions: 1) there is a high similarity if the correlation coefficient between a key audio event and one of its adjacent events is large enough, and 2) the longer time interval between a key audio event and one of its adjacent events, the lower their correlation is. Thereby, we define the following function between two audio events i^{th} and j^{th} to characterize their correlation measure:

$$S_{ij} = \frac{\delta}{d_{ij}} \cdot exp\,(-(d_i - d_j)^2 /(d_i + d_j)) \cdot exp\,(corr_{ij}) \quad (9)$$

where d_{ij} and $corr_{ij}$ represent the distance and the correlation coefficient between the i^{th} and the j^{th} events, respectively, d_i and d_j are their durations, and δ denotes the harmonic factor.

We employ the following scheme to extend each key audio event. Initially, the key audio event is considered in its own movie shot, whose boundary is defined by the event itself. Then, to calculate its audio boundary extensively, we determine whether the audio segments that surround the key audio event could be merged using formula (9). This process is iteratively repeated if any other sound event is correlated to the key audio, else the extension of the key audio event is stopped if no such sound event exists. For example, as shown in Figure 2, there exist four key audio events, namely, r_1, r_2, r_3 and r_4 colored red. The rest consists of non-key events (in blue color) and background sounds (in gray color). Non-key audio events can be merged into a key audio event if formula (9) is satisfied (e.g., $(i+2)^{th}$ can be merged to $(i+1)^{th}$ in r_3 as shown in Fig. 2).

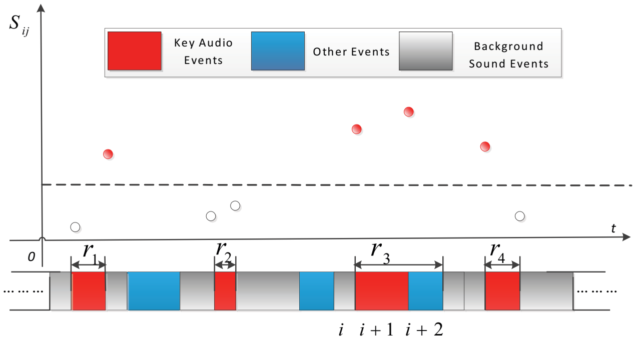

Figure 2: Illustration of audio summarization from multisource sounds through context modeling.

III. Experimental Results

We evaluated the proposed audio movie summarization framework on the dataset consisting of different movie/TV categories as listed in Table 1, in which altogether 65,112 seconds of audio data is collected. All the audio clips are sampled in 44.1KHz and mono channel. We extract 21-dim MFCC and 20-dim LDB features from the audioframes, each containing 1024 sampling points.

Table 1. The experimental dataset consisting of movies and TVs.

Category	Audio Source	Duration(s)
sports	2011 World Table Tennis Championship; ATP 2012 Wimbledon SF1; NBA 2012 Finals; 2012 Olympic Table Tennis	17,717
award ceremony	The 83rd Annual Academy Awards; 39th Annual American Music Awards; CMT Music Awards 2012; Video Music Awards 2012	21,224
comedy movie	The Big Bang Theory; 3rd Rock from the Sun; Friends	5,151
action movie	The Fast And the Furious; State of the Union; Swordfish; The Rock	21,020

Table 2 shows the auditory summarization results from the test sound tracks given in Table 1, in which the extended key events are ranked by their score functions. It can be found that the top 2 or 3 key audio events most probably reveal the content of the movie/TV as auditory semantic labels; namely, by combining several key audio labels one can infer the content of a long duration audio stream easily.

In the second experiment, we manually label the detected key audio events and use them to train an HMM model. Then the labels of detected unknown events can be learned in an automatic way. The results for the events in the four categories are shown in Fig. 3. This may be particularly useful in some real life applications such as movie/TV data indexing/retrieval, annotating media data through audio labels, and media content understanding. Figure 4 further compares the performance of HMM with the sliding window method [7] in automatically labeling audio events, from which we can see that the HMM model achieves an averagely higher accuracy than the sliding window method.

Table 2. Audio semantic summarization results obtained on the datasets in Table 1.

No.	Audio Summaries
1	***cheer***(1), ***applause***(2), ***ball-hit***(3), *speech*, *silence*, *noise*
2	***applause***(1), ***cheer***(2), ***song***(3), *speech*, *music*, *silence*
3	***laugh***(1), ***applause***(2), *noise*, *music*, *silence*, *speech*
4	***gunshot***(1), ***explosion***(2), *speech*, *music*, *noise*

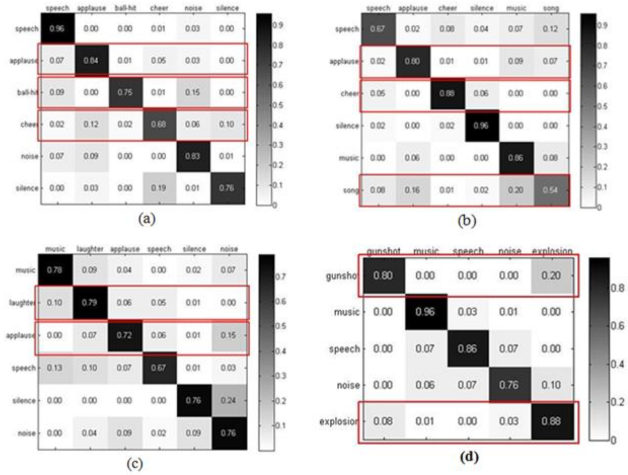

Figure 3: Confusion matrixes for the 4 audio categories with the surrounding red rectangles corresponding to key events.

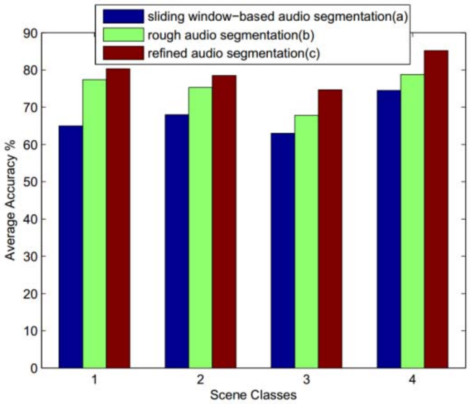

Figure 4: Overall average accuracy of audio event labeling using the sliding window-based method (a), HMM without updating audio segments using spotted backgrounds (b), and HMM with updating audio segments using spotted backgrounds .

IV. Conclusion

A novel movie audio summarization algorithm is presented, which consists of three levels of processing. We detect auditory changes in the eigen-audiospace to segment the movie stream, and exploit a scoring algorithm to refine the segments. Next, audio events are identified in a hierarchical manner consisting of background detection, foreground event recognition and key event identification. The experimental results on different movie/TV categories demonstrate the effectiveness of the propose approach. The proposed framework is useful for movie/TV data indexing or retrieval, annotating media using auditory labels, content analysis for video, etc. Our future work includes accurate recognition of more types of audio events by using acoustic contexts, the establishing of a benchmark dataset for performance evaluation, and the incorporation of other modalities to assist movie/TV computing.

Acknowledgements

The work described in this paper was supported by the Natural Science Foundation of China under Grant No. 61272218, the 973 Program of China under Grant No. 2010CB327903, and the Program for New Century Excellent Talents under NCET-11-0232.

References

[1] L. Yang, F. Su. Auditory context classification using random forests. In *ICASSP 2012*, pages 25-30, March, 2012.

[2] Y.T. Wang, B. Li, X.Q. Jing, F. Liu, and L.H. Wang. Speaker recognition based on dynamic MFCC parameters. In *IASP 2009*, pages 406-409, March 1991.

[3] Y. Shiu, H. Jeong, and C.-C. J. Kuo. Similarity matrix processing for music structure analysis. In *ACM MM 2006*, pages 69-76, October 2006.

[4] M. Sert, B. Baykal, and A. Yazici. Generating expressive summaries for speech and musical audio using self-similarity clues. In *ICME 2006*, pages 941-944, July 2006.

[5] S.H. Yella, V. Varma, and K. Prahallad. Significance of anchor speaker segments for constructing extractive audio summaries of broadcast news. In *SLT 2010*, pages 12-18, December 2010.

[6] S. Chu, S. Narayanan, and C.-C. J. Kou. Environmental sound recognition with time-frequency audio features. *IEEE Trans. On Audio, Speech, and Language Processing*, 17(6):1142-1158, August 2009.

[7] M. Kyperountas, C. Kotropoulos, and I. Pitas. Enhanced eigen-audioframes for audiovisual scene change detection. *IEEE Trans. on Multimedia*, 9(4):785-797, 2007.

[8] F. Su, L. Yang, T. Lu, G.Y. Wang. Environmental sound classification for scene recognition using local discriminant bases and HMM. In *ACM MM 2011*, pages 1389-1392, 2011.

[9] T. Heittola, A. Mesaros, T. Virtanen, A. Eronen. Sound event detection in multisource environments using source sepration. In *CHiME 2011*, pages 36-40, September, 2011.

[10] A.J. Eronen, V.T. Peltonen, J.T. Tuomi, A.P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi. Audio-based context recognition. *IEEE Trans. On Audio, Speech, and Language Processing*, 14(1):321-329, Jan 2006.