EXTRACTION OF CHARACTERISTIC MUSIC TEXTURES *(EIGEN-TEXTURES)* VIA GRAPH SPECTRA AND EIGENCLUSTERS

Saurabh Sood

The Ohio State University
Department of Electrical Engineering
Columbus, OH 43210

Ashok Krishnamurthy

The Ohio State University
Department of Electrical Engineering
Columbus, OH 43210

ABSTRACT

There exists a great diversity in the area of automatic audio segmentation since audio can be segmented based on various desirable aspects. However, the instances of texture change are not equally important for all applications than the texture itself. Typically, audio can contain a variety of textures and some of them are often repeating. Thus, only the texture change instant is not sufficient for complete characterization of given audio since it lacks the ability to judge similar textures discontiguous in time. The accurate identification of characteristic textures is crucial for many applications like classification, indexing, browsing and summarization. In this paper, graph spectra and graph eigenclusters are proposed as a scalable technique for extracting predominant textures or *eigen-textures* in a given musical audio and has yielded encouraging results. This approach not only makes segmentation more tractable and scalable but also helps in modeling given audio in terms of graphical structure, which is more perceptually revealing.

1. INTRODUCTION

A challenge to automatic interpretation of audio content is posed by varied audio contents that might exist in a given audio signal. In order to classify and analyze the audio content, it would be beneficial to extract predominant textures present in the audio as a front-end processing stage, and then apply classification and analysis on these textures rather than on raw audio. Qualitatively classification accuracy of pre-segmented audio is expected to be more than that for raw audio, which has to be automatically segmented and then classified. Thus, segmentation at front end is expected to have obvious advantages. This would be analogous to edge detection in image processing and object recognition in computer vision.

Different textures within a single audio are expected to have a varied appealing level to different people. Thus, in a query-by-example (QBE) system, a query could take form of any of the present textures. Thus this kind of segmentation in first phase will enable greater flexibility for audio classification and retrieval systems like [1]. Moreover, the instances of texture change are not as important for characterization if each eigen-texture of appropriate length are obtained for all existing textures. These *eigen-textures* could form candidates to complement automatic audio indexing and browsing.

Tzanetakis [2, 3] proposed mulitfeature audio segmentation for front end processing of audio. Successive distance calculation of feature vectors of audio frames and peak picking heuristic was used for segmentation. For example: consider the hypothetical sequences of such distance between successive frames: {2, 3, 2, 2, 3, 15, 2, 2, 3, 3, 2, 20, 2, 3, 3, 2}. Existence of three possible segments can be inferred, but to ascertain if last segment is similar to first distance measure alone will fail, and some further processing would be required. In addition, the nature of peak picking heuristic impose limitations on detecting the number of segments automatically.

Lu [4, 5] proposed interesting schemes for synthesizing similar audio textures and audio texture restoration. Foote [6, 7] has presented schemes to extract segment boundaries and music summarization. These schemes embed pair wise similarity of audio frames in a 2-D similarity matrix. Our approach is similar, however involves establishing of similarity among group of frames and proposes adjacency matrix (or relation matrix) instead of similarity matrix. We focus on detecting similarity among music textures in a given audio and extracting the characteristic textures present.

2. SEGMENTATION USING EIGENCLUSTERS

Sarkar [8] has generalized the use of eigenvectors of connectivity relation matrix for change detection from aerial images. We extend its use to audio by modeling audio signal as a undirected graph which is then used for audio segmentation using eigenclusters. The relationship among audio features could be effectively captured in form of a *relation graph* [8], whose nodes represent audio features over defined temporal range and whose links denote compatibility between the features. The relevant theory of graph spectra [9] in context to $eigen - textures$ is presented next.

2.1. Eigenvalues of Graph

Given a weighted relation graph with adjacency matrix **A**, the maximally cohesive node cluster would correspond to eigenvector corresponding to highest eigenvalue of **A**. Also, eigenvector corresponding to second largest eigenvalue will give a maximally cohesive node cluster which is $orthogonal$ to one with highest eigenvalue. This can be generalized at any level (Rayleigh-Ritz theorem). Thus, if nodes in graph consist of features of audio signal (defined over uniform temporal range) and are linked by weighing according to degree of similarity, eigenvectors of **A** constituting spectrum of graph, provide natural segmentation of audio.

2.2. Eigenclusters

Because of lack of physical interpretation for negative values, only $positive$ eigenvectors have been utilized in this study. An eigen vector **x** is said to be positive if all the components of **x** or **-x** are positive and it corresponds to positive eigenvalue. Since

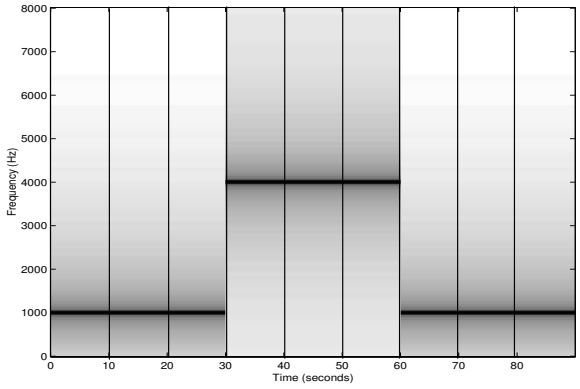

Fig. 1. Spectogram of synthetic two tone audio.

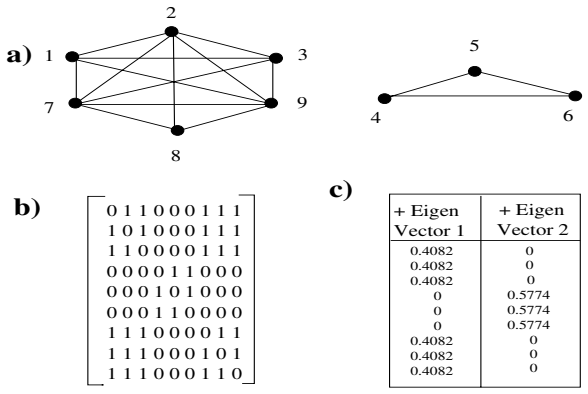

Fig. 2. (a) Graph corresponding to Fig. 1 (b) Adjacency Matrix (c) Positive Eigen Vectors (small index implies larger eigenvalue).

every node have an implicit temporal range defined, the collection of nodes corresponding to the positive components of positive eigenvector constitutes an $eigencluster$, which is identified as an $eigen-texture$.

Consider a synthetic audio file with two tones, 4000 Hz is sandwiched between 1000 Hz, as shown in Fig. 1. Here the node i represents $10(i-1)$ to $10i$ seconds as shown by vertical black lines. Fig. 2a depicts the graph, which captures similarity among nodes $1, 2, 3, 7, 8, 9$ and $4, 5, 6$. It can be seen that two positive eigenvectors (Fig. 2) naturally segment the audio into constituent *eigen-textures*. The kth entry of an eigenvector captures the contribution of the kth node in that cluster. Moreover, orthogonality of eigenvectors results in disjoint audio segments. It is worthwhile to note that eigenvector 1, summarizes this synthetic audio since it represents maximum similarity to whole. Interestingly, Foote [7] also arrives at similar result but using a similarity matrix.

The audio feature set is used to model audio textures and depending on its effectiveness, the graph will have varying degree of $incompatible$ links. Thus in practice definition of positive eigenvector had to be relaxed to account 98% of energy as dominant components. Relaxation could to lead to some overlap in the audio segments.

3. SEGMENTATION SCHEME

The scheme can be divided into two stages: pre-segmentation stage and eigen-texture extraction stage. First, raw audio is broken into variable length pre-segments, that comprise the nodes of the graph. This is similar to finding edges in an image. In next stage, these pre-segments (nodes) are edge-linked whose weights are proportional to a similarity measure. The eigenclusters of adjacency matrix for this graph yield desired *eigen-textures*. This general scheme could find many applications, however novelty lies in finding appropriate pre-segmentation scheme and a similarity measure for that particular application.

Basic audio features are calculated using 16 msec overlapping frames with 4 msec overlap on 16KHz audio. Hamming window is used for windowing. Following audio features were extracted for each frame. Here $x[n]$ denotes the time domain audio frame of length M and $X[f]$ is corresponding FFT magnitude at frequency bin f of N such bins.

1. *Power:* $P = \sqrt{\frac{\sum_1^M x[n]^2}{M}}$, the RMS power.
2. *Centroid:* $C = \frac{\sum_1^N fX[f]}{\sum_1^N X[f]}$, a measure of spectral brightness.
3. *Rolloff:* R such that, $\sum_1^R X[f] = 0.85 \sum_1^N X[f]$, a measure of spectral shape.
4. *ZeroCrossings:* ZCR, the number of zero crossings of $x[n]$.
5. *Mel Filter Bank Output:* 40 mel-scale filters are used to filter $X[f]$ resulting in a vector $\mathbf{fb}$.

3.1. Pre-Segmentation Stage

Pre-segmentation is achieved by high pass filtering ZCR, $\log(\mathbf{fb})$ centroid and $\mathbf{fb}$ centroid features, using a sliding window $w[n]$.

$$w[n] = \begin{cases} -1 & n \geq -(L-1) \ \ \& \ \ n \leq 0 \\ 1 & n \geq 1 \ \ \& \ \ n \leq L \end{cases} \quad (1)$$

Window length (L) corresponding to 0.35 sec. is used. Peaks in resulting signal correspond to a significant long-term change in feature. Small variations are clipped to zero by applying critical limit of 1.5 times mean of absolute valued signal. Resulting peaks from three features are normalized with their respective maxima and merged to yield overall change variation. This is done to capture the changes present in both ZCR and the centroid features. This overall change variation is then post filtered to retain only the peaks in about 0.7 sec interval thus yielding pre-segments demarcated by the peaks. A pre-segment is further sub-divided if it's length is greater than 2 sec. Fig. 3 depicts the pre-segments marked by vertical black lines for Lagaan's *Waltz for a Romance* and it can be observed that important spectral changes are accounted. This scheme performs well empirically, as it pre-segments the audio without missing any $correct$ texture change instances. Also, it improves the computational efficiency by reducing the no. of similarity calculations at pre-segment level as opposed to frame level.

3.2. Extraction of eigen-textures

Similarity measures between pre-segments (p_i and p_j) are based on the audio features. For each pre-segment, upper and lower limit of power, centroid and rolloff are calculated. Since these similarity measures remain consistent over these features, let variable X denote the feature. X could be P power, C centroid or R rolloff. Similarity measures are defined as follows (Fig. 4 depicts expressions used):

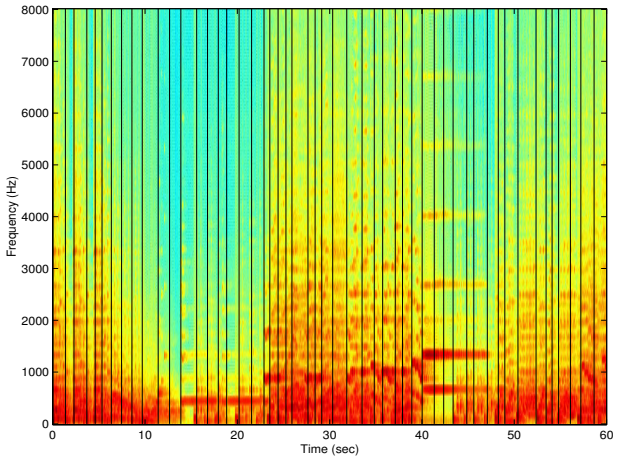

Fig. 3. (a) Pre-Segments for an audio file (Waltz)

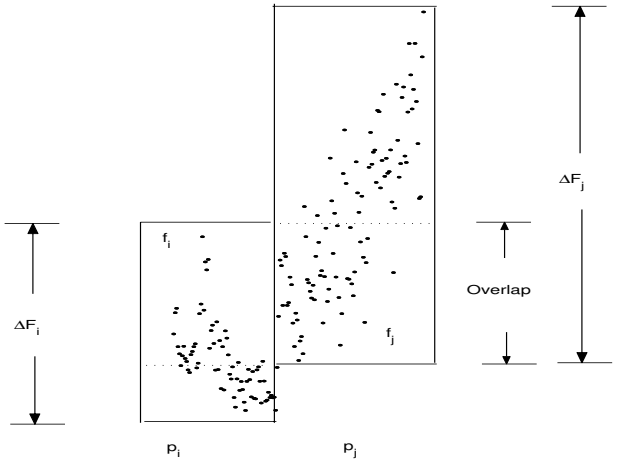

Fig. 4. Illustrates expressions used for similarity measures

1. *Overlap Similarity:* This measures quantifies the degree of overlap over the F feature range.

$$\mathbf{Sim_{OLap\_F}}(p_i, p_j) = min\left(1, \frac{Overlap}{min(\Delta F_i, \Delta F_j)}\right) \quad (2)$$

2. *Overlap Strength:* This establishes the actual fraction of content present in this overlap.

$$\mathbf{Sim_{OStr\_F}}(p_i, p_j) = min(f_i, f_j) \quad (3)$$

The f_i and f_j denote fraction of feature points contained within the overlapped region, in p_i and p_j respectively.

3. *Range Similarity:* This measure relates the range over which the feature changes.

$$\mathbf{Sim_{Range\_F}}(p_i, p_j) = 1 - min\left(1, \frac{|\Delta F_i - \Delta F_j|}{min(\Delta F_i, \Delta F_j)}\right) \quad (4)$$

Using the above notations in Eqs. 2, 3 and 4, following similarity measures were employed for the relation linking.

1. *Power Similarity Measures:*
$\mathbf{Sim_{OLap\_P}}(p_i, p_j)$ measures the degree of overlap over the power range and differentiation between a louder pre-segment from a quiter one is provided by $\mathbf{Sim_{Range\_P}}(p_i, p_j)$. Thus they together serve as an overall power similarity.

2. *Spectral Similarity Measures:*
$\mathbf{Sim_{OLap\_C}}(p_i, p_j)$ and $\mathbf{Sim_{OStr\_C}}(p_i, p_j)$ are used for Centroid based similarity measures. Similarly for rolloff, $\mathbf{Sim_{OLap\_R}}(p_i, p_j)$ and $\mathbf{Sim_{OStr\_R}}(p_i, p_j)$ are used.

An iterative algorithm with following steps was developed to perform similarity linking (on a scale from 0 to 1, with higher value implying more similarity) of pre-segments by setting link weight $w(p_i, p_j)$ between two pre-segments.

1. For time consecutive pre-segments, a simple auto-correlation based heuristic is used to find any periodicity in their power variation. If periodicity is detected, link weight $w(p_i, p_j)$ is set to the magnitude of first peak in normalized auto-correlation plot. This was found to be useful in periodic music textures like beats.

2. $\mathbf{Sim_{OLap\_P}}(p_i, p_j)$, $\mathbf{Sim_{Range\_P}}(p_i, p_j)$, $\mathbf{Sim_{OLap\_C}}(p_i, p_j)$, $\mathbf{Sim_{OStr\_C}}(p_i, p_j)$, $\mathbf{Sim_{OLap\_R}}(p_i, p_j)$ and $\mathbf{Sim_{OStr\_R}}(p_i, p_j)$ are calculated. If their minimum (denoted by $SimMin$) is more than 0.1 then next step is carried out, else $w(p_i, p_j)$ is set to zero.

3. *Rolloff sub-level based correlation:* This forms the heart of the algorithm. The rolloff overlap range of p_i and p_j are divided into fixed length sub-levels (around 100 Hz). Over each of the levels the cross correlation between mean of constituent **fb**'s is calculated. Also if any sub-level corresponds to less than 950 Hz then cross correlation between only 20 lower frequency bins of **fb**'s is calculated. Similarly, above 4.5 kHz only 20 higher frequency bins of **fb**'s are considered. A count is kept for fraction of p_i and p_j (denoted by cf_i and cf_j respectively) for which this cross correlation value is more than the threshold Th. This threshold is linearly decreased in steps as the sub-level amplitude increases. The cf_i and cf_j capture the spectral content that is similar across p_i and p_j. Link weight $w(p_i, p_j)$ is calculated as $min(cf_i, cf_j, SimMin)$. This method was found to be quite efficient in measuring in spectral similarity.

4. *Smoothing:* If two pre-segments (p_i and p_j) are spectrally similar such that the time difference is less than 2 sec. then fraction of similarity measure is incremented to the intermediate pre-segments. This step could be ignored or adjusted depending on the application.

4. EXPERIMENTS

Audio files (each of 1 min. duration for consistency) representing variety of texture patterns were experimented with, yielding quite encouraging results. Eigenvectors and corresponding eigen-textures for Lagaan's *Waltz for a Romance* (1 min. clip from beginning), an orchestral piece are shown in Figs. 5 and 6. Eigen-texture 2 mainly composed of violin and string instruments, captures main melody of the song. Eigen-texture 1, 3 and 4 represents different sounding transition phases.

The Magical Mystery Tour by The Beatles (1 min. clip from beginning), yielded the predominant eigen-texture as occurrences of chorus (”Roll up”). Eigen-texture covered time durations (in

sec.) $9.5-11.7$, $14.8-17.5$, $19.6-22.8$, $26.9-29.2$, $40.6-43.7$, $46.7-48.8$, $51.6-53.9$ and $58.1-60$. All these durations have the same high-pitched chorus content.

Robert Miles' *Children (dream version)* (1 min. clip, 100 sec from beginning) results in four primary eigen-textures corresponding to piano and techno starting $0-7.9$ & $13.5-18.6$, buiding beats with slight shrill at $25.8-30.1$, building up the tempo with beats $30.1-39.8$ and finally loud beats in the $39.8-60$ segment.

Fig. 7 shows characteristic eigen-textures of a (1 min. clip, 15 sec. from beginning) classical, the Spring (allegro) from Vivaldi's *The Four Seasons*. Eigen-texture 3 captures similarity in opening and closing parts and each eigen-texture has its own flavor distinguishing it from another. Detailed results in addition to audio files is available on the Web: **http://www.eleceng.ohio-state.edu/ soods/eig-tex/results.htm**. Average execution time of our MATLAB implementation of algorithm on standard Pentium 4 system is about 36 sec. for 60 sec. of audio.

5. CONCLUSIONS

In this paper, we presented use of graph spectra and various similarity measures for extracting characteristic eigen-textures for music. Experimental results on popular and classical music have shown quite encouraging results with perceivably accurate detection of similar textures. While this approach might not yield an exhaustive set of textures exactly according to psychoacoustic principles, but it does give satisfactory results by identifying primary textures. Also current approach cannot characterize textures in voice only audio. The future research will be directed towards making it even more effective and diverse. Applications for audio summarization and browsing using this approach for is also being explored.

6. REFERENCES

[1] S. Sood, A. Roongta, S. Chaudhury, and A. Kumar, "Content based retrieval in a repository of indian classical music," *Proc. Intl. Conf. on Communications, Computers and Devices*, vol. II, pp. 641–4, 2000.

[2] G. Tzanetakis and P. Cook, "Multifeature audio segmentation for audio browsing and annotation," *IEEE WASPAA*, pp. 103–6, Oct. 1999.

[3] G. Tzanetakis and P. Cook, "Marsyas: a framework for audio analysis," *Organised Sound*, vol. 4, no. 3, 2000.

[4] L. Lu, S. Li, L. Wenyin, H. J. Zhang, and Y. Mao, "Audio textures," *Proc. of ICASSP*, vol. II, pp. 1761–4, 2002.

[5] L. Lu, Y. Mao, L. Wenyin, and H. J. Zhang, "Audio restoration by constrained audio texture synthesis," *Proc. of ICASSP*, vol. V, pp. 636–9, 2003.

[6] J. Foote, "Automatic audio segmentation using a measure of novelty," *Proc. of ICME*, vol. I, pp. 452–5, 2000.

[7] M. Cooper and J. Foote, "Automatic music summarization via similarity analysis," *Proc. of ISMIR*, pp. 81–5, 2002.

[8] S. Sarkar and K. L. Boyer, "Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors," *Computer Vision and Image Understanding*, vol. 71, no. 1, pp. 110–36, July 1998.

[9] H.S.D.M. Cvetkovic and M. Doob, *Spectra of Graphs*, Academic Press, 1979.

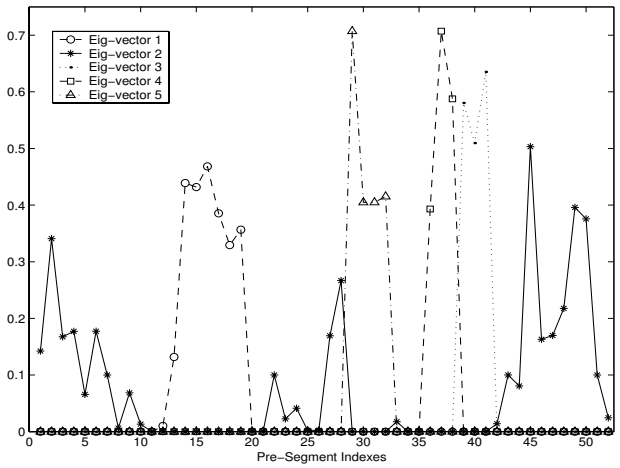

Fig. 5. Five positive eigenvectors obtained for Lagaan's *Waltz*

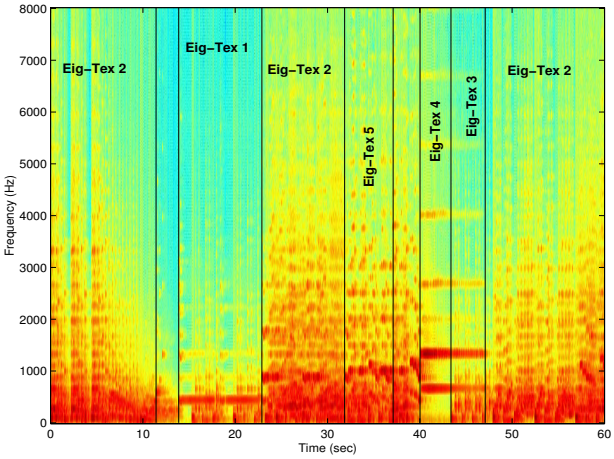

Fig. 6. Five corresponding eigen-textures (Fig. 5) for *Waltz*

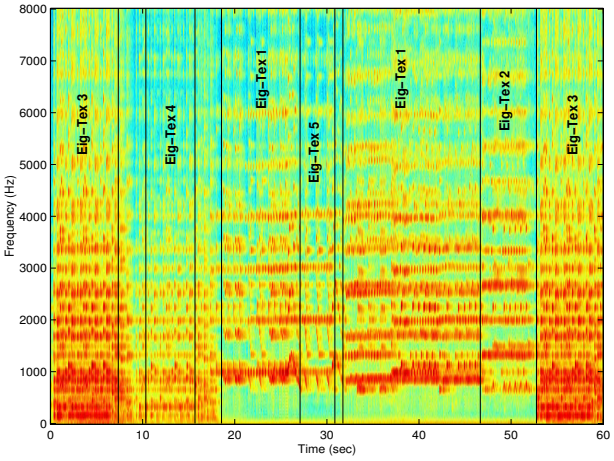

Fig. 7. Eigen-textures for Vivaldi's *Spring*