

# Signal-based Music Searching and Browsing

Mark B. SANDLER, *Senior Member, IEEE* and Mark LEVY

**Abstract**—This paper describes an approach to the problem of finding songs in some sense similar to a query song. This is a problem of increasing importance, because consumers owning large digital music collections wish to navigate these and to add new songs by searching on-line. The technological approach is described, leading to the description of a simple demonstrator.

## I. INTRODUCTION

Today, there are several web-sites that music fans may go to to find music they like. Pandora [1] offers a personalized streaming service, based on a user's preferred band or track and its own human-generated, musicological analysis. MusicIP [2] offers songs you should like based on a machine analysis of songs in its database: you cannot however then listen to the whole song. Several other startups already operate in this space, and we can expect it to get more crowded as consumers, with ever larger collections, need and demand new ways to navigate their own collection, and to find new material to buy or audition.

Current wisdom is that while Pandora's service offers interesting music (not everyone agrees!), the human mark-up does not scale to collections of millions of songs. This has spawned several approaches to analysis and retrieval of music that are entirely based on the audio content.

Previous approaches to this *content-based (similarity) search* for music [3,4,5] use the same basic procedure. Features capturing the overall timbre of the music (typically Mel-Frequency Cepstral Coefficients (MFCCs) are extracted for each audio frame and similarity between tracks is measured by comparing feature sets.

Our approach differs from this in two ways: first we do not compare features for the whole of each track, but only for a salient segment (which we call its thumb-nail) identified in a pre-processing step, and secondly we do not use fixed sized analysis frames of the audio but instead tie our analysis to beats in the audio, again identified by a pre-processing step.

This paper first discusses, in section II, the approach we take to segmentation and consequently the way in which thumb-nails are extracted. Then we describe how we perform a search on thumb-nails. In section III, we discuss a prototype system for performing the analysis and search, called SoundBite. Finally we conclude with some observations of how SoundBite might be used in various consumer scenarios.

The authors wish to acknowledge the UK Engineering & Physical Sciences Research Council (EPSRC) for supporting this work under grant GR GR/S84750/01 (Hierarchical Segmentation and Semantic Markup of Musical Signals)

## I. SEGMENTATION, THUMB-NAILS AND SEARCH

Most music has a sectional structure, perhaps the most familiar being the verse/chorus form of modern pop songs. Automatic approaches to structural segmentation normally focus on identifying and labeling repeated stretches of audio within a given track.

### A. Music Segmentation

We identify and label sections according to their timbral features. Where possible these are obtained from analysis windows based on the beat of the music (typically 300 – 400 milliseconds), estimated using a beat-tracking algorithm [6].

Previous work [7,8] has used clustering to label short frames as belonging to a given number of underlying timbre types corresponding loosely to different combinations of instruments or voices. Our approach extends this by observing that although timbre changes from beat to beat, the distribution of timbre-types remains fairly consistent over a structural segment and can be used to characterize segment types.

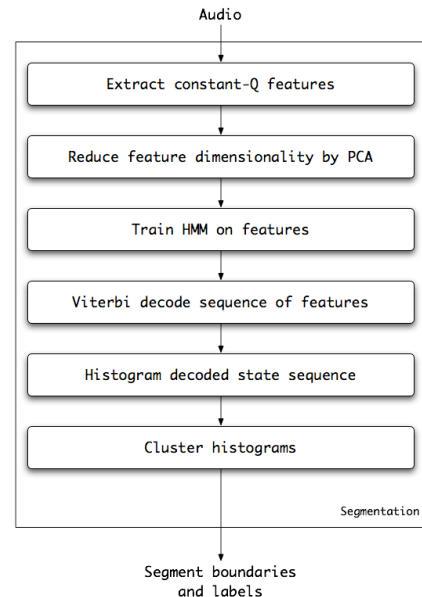


Fig. 1. Segmenting the audio track

The steps involved in our method are shown in Fig. 1. Constant-Q spectral features are extracted for each beat of the music. A Hidden Markov Model [9] is trained on these features, its hidden states corresponding to timbre types spanning the entire space of timbre within the track. The feature sequence is then Viterbi decoded to recover the most likely sequence of underlying timbre-types to have generated the observed features. Each beat is then labeled with its corresponding timbre-type.

Finally we compute a high-level structural segmentation from the sequence of beat labels. We first calculate normalized histograms of timbre-types over a sliding window of 7 beats. The characteristic mixture of timbre-types for each segment are estimated by clustering the histograms[ms1] - see [10].

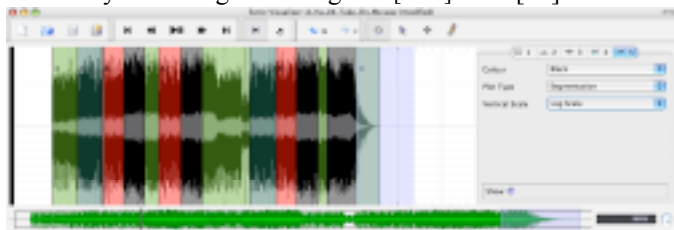


Fig. 2. An automatic segmentation of a song presented in the time domain. Similar segments are similarly coloured. This is presented using Sonic Visualiser [13] and a segmentation plug-in.

### B. Music Thumb-nailing

Because we use a model-based approach, our method yields a full segmentation of the audio – see Fig. 2, rather than just identifying certain sections as similar to others. The musical overview provided makes it straightforward to generate representative musical thumbnails of each track. Our method is devised both to provide a human listener with a quick impression of the track, and for machine use in searching (see §II.C).

First, segments are counted to find the most frequent segment type (ignoring short segments). If there is a tie for first place, we select the segment type with the highest energy. We then pick the second segment of this type, say the chorus, believing it to be more representative than its first appearance.

### C. Searching with Segment Models

Finding a match in a search operation relies on the computation of a distance measure between the features of the query and those of each song in the target database. In recent research, it has been shown that calculating the popular Kullback-Leibler divergence costs about 3ms per track [3] [4]. This clearly will lead to unacceptable delays if searching over a song collection of even modest size– this would be 30 seconds for 10,000 tracks.

By modeling only the chosen thumbnail with a single Gaussian model built from a 20-order MFCC reduces this cost approximately 1000-fold. Also, the search-index metadata size per track is only 1280 bits, implying consequently, a minimal storage overhead.

To provide a quantitative metric on performance we used a genre classification task over a set of 700 songs of mixed genre. The performance in this task is very close to that of the state-of-the-art approach in [5] in spite of the greatly reduced computation and storage.

## II. SOUNDBITE: A BROWSER AND SEARCH ENGINE

SoundBite is an implementation of our segmentation, thumb-nailing and similarity search methods. It is illustrated in Fig. 3. SoundBite manages a database of tracks, allowing the entry of simple metadata (e.g. ID3 tags) as each new track is added, while automatically extracting and saving segmentation information and a thumbnail segment. Tracks

can be browsed and searches made based on similarity to a chosen track. As well as serving as a demonstration of the use of thumbnails in presenting music search results, this offers a way to experience and experiment with searches based directly on the results of the segmentation process.

SoundBite is a demonstration of principle. With a test set of 1500 thumbnails, a mixture of excellent and interesting responses are provided. It is clear that additional features would beneficially enhance the quality of the retrieval.

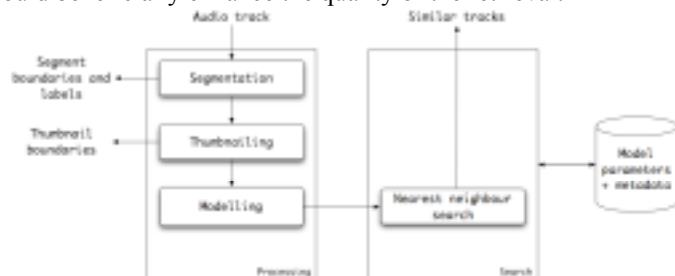


Fig. 3. SoundBite Framework

## III. CONCLUSIONS

We have described a practical and scalable method for searching for songs. The search itself is speeded up by selecting a sub-set of features using musical prior knowledge (those of the thumbnail) and by a matching algorithm that is 1000 times faster than a state-of-the-art existing method.

Users can browse their own collections based on these thumb-nails, and built play-lists simply. They can also browse within a song using the segmentation. For example, a user might skip to the chorus and click *repeat*. Service providers can offer intelligent navigation (and playlisting) of massive song databases, for example as a subscription service.

## REFERENCES

- [1] <http://www.pandora.com>
- [2] <http://www.musicip.com>
- [3] E. Pampalk, A. Flexer, and G. Widmer, “Improvements of audio-based music similarity and genre classification,” in *Proc. ISMIR*, 2005.
- [4] P. Roy, J.-J. Aucouturier, F. Pachet, and A. Beurive, “Exploiting the trade-off between precision and cpu-time to speed up nearest neighbour search,” in *Proc. ISMIR*, 2005.
- [5] A. Flexer, “Statistical evaluation of music information retrieval experiments,” Tech. Rep., Österreichisches Forschungsinstitut für Artificial Intelligence, 2005.
- [6] M. E. P. Davies and M. D. Plumbley, “Beat tracking with a two state model,” in *Proc. ICASSP*, 2005.
- [7] J.-J. Aucouturier, F. Pachet, and M. Sandler, “The way it sounds : Timbre models for analysis and retrieval of polyphonic music signals,” *IEEE Transactions of Multimedia*, vol. 7, no. 6, pp. 1028–1035, 2005.
- [8] B. Logan and S. Chu, “Music summarization using key phrases,” in *International Conference on Acoustics, Speech and Signal Processing*, 2000.
- [9] L. R. Rabiner, “A tutorial on hidden markov models and selection applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [10] M. Levy, M. Sandler, and M. Casey, “Extraction of high-level musical structure from audio data and its application to thumbnail generation,” in *Proc. ICASSP*, 2006.
- [11] B. Logan and A. Salomon, “A music similarity function based on signal analysis,” in *Proc. ICME*, 2001.
- [12] J.-J. Aucouturier and F. Pachet, “Music similarity measures: What’s the use?,” in *Proc. ISMIR*, 2002.
- [13] Sonic Visualiser Open Source software from <http://sv1.sourceforge.net/>