Audio Engineering Society

Convention Paper

Presented at the 120th Convention
2006 May 20–23 Paris, France

Application of segmentation and thumbnailing to music browsing and searching

Mark Levy[1], Mark Sandler[1]

[1]*Centre for Digital Music, Queen Mary, University of London, Mile End Road, London E1 4NS, UK*

Correspondence should be addressed to Mark Levy (mark.levy@elec.qmul.ac.uk)

ABSTRACT

We present a method for segmenting musical audio into structural sections, and some rules for choosing a representative 'thumbnail' segment. We demonstrate how audio thumbnails are an effective and natural way of returning results in music search applications. We investigate the use of segment-based models for music similarity searching and recommendation. We report experimental results of the performance and efficiency of these approaches in the context of SoundBite, a demonstration music thumbnailing and search engine.

1. INTRODUCTION

Beyond the purely sensory and emotional pleasure we get from music, one of our natural first responses to a piece of music is to become aware of its high-level structure, i.e. where the various sections begin and end, and how they relate to one another. This is 'natural' because the huge majority of music is composed, or put together in the recording studio, exactly by repeating and interweaving contrasting sections, each consisting of a small number of phrases of similar length. We can observe this pattern in almost all popular and world music, and in a great deal of conventional classical music, most obviously in the chorus-verse songs which occupy the huge majority of most people's music-listening time.

With the proliferation of large digital music collections, from the millions of tracks in commercial databases to the thousands you can now store in your pocket, increasingly effective ways of navigating between tracks have been developed. Our work on machine segmentation, i.e. the automatic extraction of high-level musical structure, aims to extend these methods to allow navigation *within* tracks. Besides enabling enhancements to media players and audio editors, our particular approach to segmentation also makes it possible automatically to choose a 'thumbnail' or characteristic segment of a particular track, for example for use in browsing rapidly through a list of possible tracks of interest returned by a search engine. By guiding us to the most significant parts of a

music track, it also allows the development of fast and efficient methods for searching very large collections based purely on the audio content of the tracks, sidestepping the computational complexity of existing content-based search methods.

This paper summarises our approach to automatic segmentation and thumbnailing of musical audio, and outlines the design of SoundBite, a prototype music search engine which demonstrates some of the potential applications of our work.

2. MUSIC SEGMENTATION

The huge majority of music has a sectional structure, probably most familiar in the verse-chorus form of conventional pop music. Although musical syntax (cadences, changes of key, etc.) can give us helpful clues as to the location of segment boundaries, the structure of a great deal of music is evident even to the untrained listener simply due to the repetition of content in segments of the same type. Automatic approaches to structural segmentation have therefore focussed on identifying and labelling repeated stretches of audio within a given track.

We use a two-level process to identify and label sections according to their timbral features. When extracting features, where possible we use analysis windows based on the beat of the music (typically 300-400ms), as estimated by a beat-tracking algorithm [1]. This is helpful (although not strictly necessary), because section boundaries usually coincide with beat starts. For the sake of clarity, in the rest of this paper when describing our methods we refer to beats rather than analysis frames.

Previous work [2, 3] has used clustering methods to label short frames as belonging to a given number of underlying 'timbre-types', corresponding loosely to different combinations of instruments or voices, but no high-level structure emerges. Although short sequences of neighbouring frames may be assigned to the same timbre-type, the overall timbre changes frequently during the course of any section of significant length. Our approach extends this with the following insight. Although timbre changes from beat to beat, the distribution of timbre-types remains fairly consistent over the course of structural sections, and can be used to characterise segment-types.

The steps involved in our segmentation method are sketched in Figure 1. We extract feature vectors for each beat of the music based on constant-Q spectra, and train a

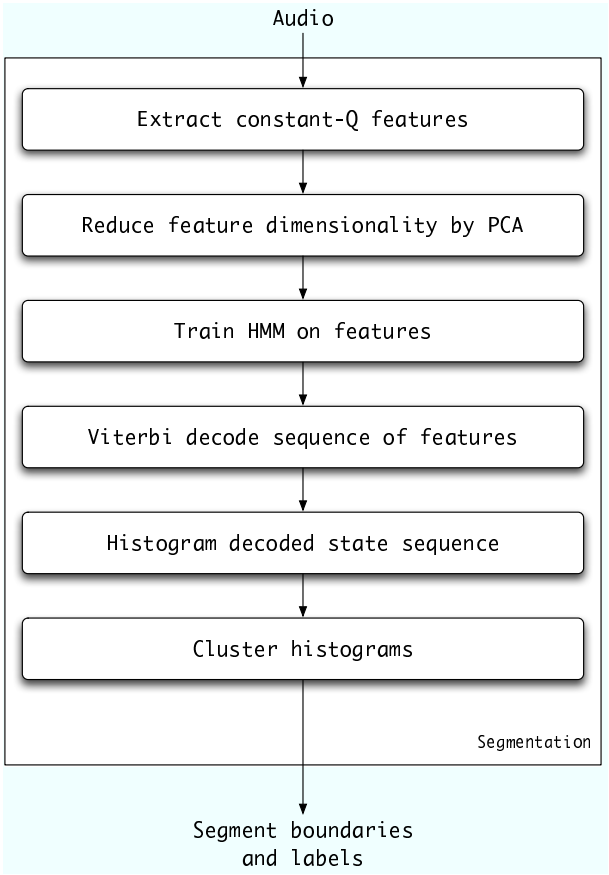

Fig. 1: Segmenting the audio track.

Hidden Markov Model [4] on the features, where the hidden states correspond to timbre-types spanning the overall space of timbre within the track. We then Viterbi decode the sequence of features with the trained model to recover the most likely sequence of underlying timbre-types to have generated the observed features, and label each beat in the music with its corresponding timbre-type. Finally we compute a high-level structural segmentation from this sequence of beat labels.

We use two alternative methods for this high-level segmentation, both of which aim to capture the expectations about segment lengths neglected in previous work. In the method illustrated in Figure 1, we first calculate histograms of timbre-types over a sliding window of 7 beats in length, and then normalise the resulting histograms. We then estimate the characteristic mixture of timbre-types for each segment by clustering the histograms into

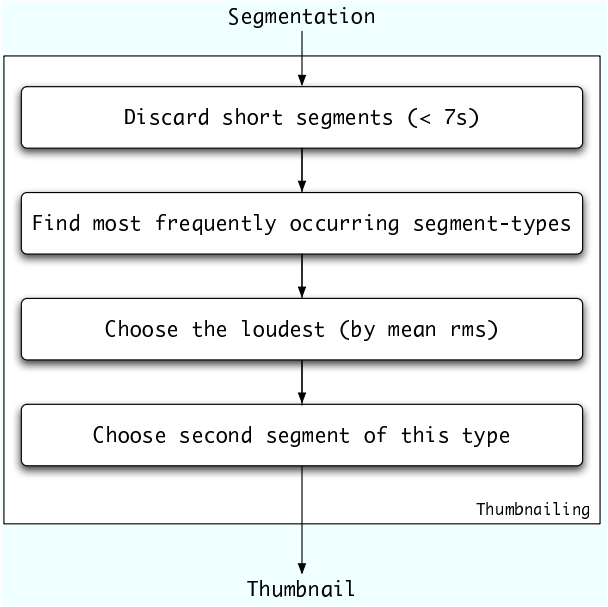

Fig. 2: Choosing a thumbnail segment.

M clusters. The reference histograms for each cluster give distributions of timbre-types $\{h_m\}, m = 1, ..., M$ and the cluster assignments give the corresponding segmentation $S = \{s(1), ..., s(T)\}$, where $s(t)$ gives the segment-type assignment for frame t. We use an adapted form of k-means clustering based on a suitable distance measure for histograms, with a neighbourhood term in the cost function reflecting the number of non-matching labels amongst nearby beats. The effect of this is to favour segments that are at least as long as the neighbourhood size. Our algorithm is also designed to be insensitive to the number of clusters M being sought: redundant clusters are left unoccupied and so we can set M to some abitrary large number. More details are given in [5].

3. MUSIC THUMBNAILING

Because we use a model-based approach to segmentation (in contrast to previous work such as [6, 7]), our method yields a full segmentation of the supplied audio track, rather than just identifying certain sections as similar to others. The musical overview provided by this complete segmentation makes it straightforward to generate representative musical 'thumbnails' of tracks. Our approach to choosing the thumbnail clearly depends on its intended purpose. The method illustrated in Figure 2 is intended to choose a single segment both to give the human listener a quick impression of the track as a whole, and for machine use in searching for similar tracks.

We first count segments to find the most frequently occurring segment-type(s), excluding any very short segments. If there is a tie for first place, we select the segment-type with the highest mean energy: this favours stronger sections, such as choruses, over equally frequent weaker ones, such as verses. We then pick the second segment of the chosen type, as an occurrence of a musical section towards the middle of a track is often more representative of the piece as a whole than one at the very beginning or end. In tests over a small hand-annotated collection of 34 pop songs, the chosen thumbnail contained music from the chorus in 73% of cases, and either from the verse or a characteristic instrumental riff in the others [5].

4. SEARCHING WITH SEGMENT MODELS

4.1. Existing similarity search methods

A number of attempts have recently been made to implement content-based similarity searching for music, i.e. to develop a method which, given a query track, will find other similar-sounding tracks amongst a large collection, where similarity is measured directly according the audio content of the tracks [8, 9, 10]. All these attempts use the same basic procedure. Features capturing the overall timbre of the music, typically Mel-Frequency Cepstrum Coefficients (MFCCs), are extracted for each frame and modelled with a mixture model. Similarity between tracks is then measured by comparing the two models.

While the success of such methods is ultimately subjective, a reasonable evaluation of similarity search that has been widely used is as follows. Given a query containing punk rock, say, we would expect the best match from a large collection usually to be another punk track. Consequently we can evaluate a similarity search method by treating it as if it were a nearest-neighbour genre classifier, and measuring its accuracy on a classification task over a set of tracks whose genre is already known. Because collections are usually built up from entire albums, in practice the nearest neighbour to a given track is another track from the same album. As a useful similarity search engine clearly needs to do more than simply recommend tracks from the same album as the query, this evaluation is more useful when done with an 'artist filter', i.e. excluding other tracks by the artist responsible for the query from consideration when estimating the

query's genre [10]. Although this is clearly a very approximate way of measuring the usefulness of a similarity search method, we will return to it later, as it does enable direct comparison between systems, even if the results have to be treated with some caution.

The major drawback of these existing search methods is that the appropriate distance measure for mixture models, the Kullback-Leibler divergence, can usually only be approximated, and the approximations used are very computationally expensive. This means that such systems, while interesting, are unlikely to scale to online searching of databases of realistic size. The fastest performance reported to date is 3ms for each pair of tracks to be compared [10], while results using approximate methods to speed up similar searches in [11] are at best twice as fast as this. With this level of performance a full search of a collection of say 100,000 tracks would take several minutes, while searching large commercial collections would take hours.

4.2. Using segment models

Existing mixture models for timbre are described as treating a track as a 'bag of frames', because the order of the frames in time is not taken into account when clustering their features into the components that make up the mixture. In contrast, we build segment models by fitting a single Gaussian to features extracted from segments of each significant type, so that mixture components in the model correspond directly to groups of segments in the track. This offers both a more transparent basis for matching tracks, as you can easily hear which sections from each track are being compared, and a direct way to remove noise from the similarity calculation simply by not representing short or rarely-occurring segments in the mixture. More importantly, in the extreme case segment models offer a vastly quicker and less memory-intensive alternative for similarity searching than existing methods, with little or no loss of performance.

The smallest segment model we can build is simply to fit a single Gaussian with a diagonal covariance to the chosen thumbnail segment. Using features consisting of the first 20 MFCCs, our model is then completely parameterised by its mean and covariance, i.e. a vector of 40 floating-point values. The distance between models can then be measured exactly (in contrast to the approximations that have to be used with mixtures) with a symmetrised Kullback-Leibler divergence, which takes only a few multiplications and additions to compute. In an unoptimsed Java implementation on a standard consumer laptop, this reduces comparison time between tracks to around 0.002ms. This makes full online searches of large collections eminently practical, in particular because the model for each track need occupy only 1280 bits, which makes it comparable with the very cheapest representations currently used for audio fingerprinting.

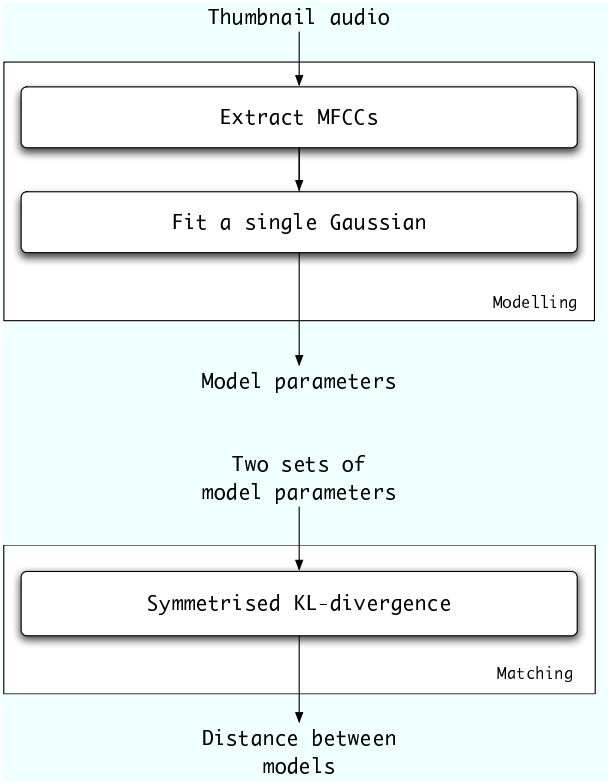

Fig. 3: Extracting and comparing segment models.

Although objective comparison of similarity search systems is well known to be difficult, not least because copyright issues in general mean that test databases cannot be shared, we have been able to test genre classification performance, as outlined above, on a collection over some 700 tracks from the Magnatune[1] collection that has been made available to all research teams working in this field. Typical state of the art performance using mixture models is given as $61.6 \pm 4.9\%$ for a 95% confidence interval over a 10-fold evaluation [12]. The 95% confidence interval on classification accuracy of our single segment model on the same task was $57.5 \pm 0.2\%$.

[1] http://www.magnatune.com.

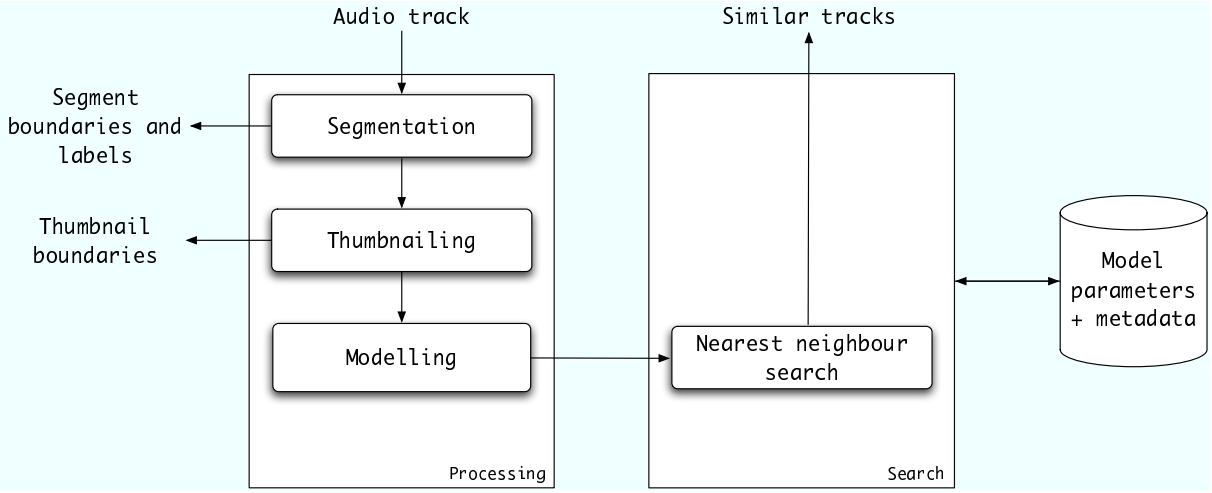

Fig. 4: SoundBite framework.

5. SOUNDBITE: A THUMBNAIL BROWSER AND SEARCH ENGINE

SoundBite is a practical implementation of our segmentation, thumbnailing and similarity search methods. The design of the system is illustrated in Figure 4. SoundBite manages a database of tracks, allowing the entry of simple metadata (album title, trackname, artist, genre, etc.) as each new track is added, while automatically extracting and saving segmentation information and a thumbnail segment. Tracks in the database can be browsed and searches can be made based on similarity to a chosen track: in all cases the results are presented as a search engine-style list, with thumbnail audio immediately available for playback for each track in the list. Besides serving as a demonstration of the use of thumbnails in presenting music search results, SoundBite offers an environment within which to experience and experiment with searches based directly on the results of the segmentation process.

6. CONCLUSIONS

We have presented automatic methods for segmenting music tracks and choosing a representative audio thumbnail. The thumbnails chosen in this way are perceptually effective, often being chorus sections for popular music. Besides its use in user interfaces to music collections and individual tracks, segmentation opens the way for a content-based similarity search method which scales easily to realistic numbers of tracks. As implemented in our SoundBite demo system, segmental similarity search is at least 1000 times faster than existing methods, with at worst only a small loss in accuracy (as measured by genre classification), and requires a footprint of only 160MB to search a collection of a million tracks.

7. REFERENCES

[1] M. E. P. Davies and M. D. Plumbley, "Beat tracking with a two state model," in *Proc. ICASSP*, 2005.

[2] Jean-Julien Aucouturier, François Pachet, and Mark 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.

[3] B. Logan and S. Chu, "Music summarization using key phrases," in *International Conference on Acoustics, Speech and Signal Processing*, 2000.

[4] Lawrence 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.

[5] Mark Levy, Mark Sandler, and Michael Casey, "Extraction of high-level musical structure from audio data and its application to thumbnail generation," in *Proc. ICASSP*, 2006.

[6] Jonathan Foote, "Visualizing music and audio using self-similarity," in *ACM Multimedia (1)*, 1999, pp. 77–80.

[7] Masataka Goto, "A chorus-section detecting method for musical audio signals," in *Proc. ICASSP*, 2003, vol. V, pp. 437–440.

[8] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc. ICME*, 2001.

[9] J.-J. Aucouturier and F. Pachet, "Music similarity measures: What's the use?," in *Proc. ISMIR*, 2002.

[10] E. Pampalk, A. Flexer, and G.Widmer, "Improvements of audio-based music similarity and genre classification," in *Proc. ISMIR*, 2005.

[11] P. Roy, J.-J. Aucouturier, F. Pachet, and A. Beurivé, "Exploiting the trade-off between precision and cpu-time to speed up nearest neighbour search," in *Proc. ISMIR*, 2005.

[12] A. Flexer, "Statistical evaluation of music information retrieval experiments," Tech. Rep., Österreichisches Forschungsinstitut für Artificial Intelligence, 2005.