Performing Query-by-Melody on Audio Collections

Matija Marolt

University of Ljubljana, Faculty of Computer and Information Science,
Trzaska 25, 1000 Ljubljana, Slovenia
matija.marolt@fri.uni-lj.si

Abstract. Mid-level representations are increasingly used to bridge the gap between high-level (semantic) and low-level audio representations. A mid-level representation that integrates melodic and rhythmic aspects of a music signal is introduced. The representation is formed by first performing multi-pitch detection on consecutive audio frames and then searching for dominant melodic lines within the detected pitches. Beat-tracking is also performed to yield a beat-synchronous representation, independent of tempo variations. The representation is used within a query-by-melody audio retrieval system. An approximate nearest neighbor search algorithm is employed to compare fragments of the mid-level representation of a query to indexed fragments of mid-level representations of songs in a collection. Symbolic queries may also be used. Results are ranked according to a weighted sum of matched fragments. Locality sensitive hashing is used for fast indexing and retrieval. Retrieval was tested on the cover song identification task, where the goal is to retrieve different interpretations of a song in a collection. Results on a collection of 2424 songs are presented.

Keywords: music similarity, audio retrieval, melodic representation

1 Introduction

Calculating music similarity is one of the key areas in music information retrieval, as it enables search and organization of music collections. Although melody is the most natural descriptor of (Western) music [1], querying audio collections by melody is still an elusive goal. Most current approaches to audio similarity, such as audio fingerprinting [2] or genre classification techniques [3] are based on low-level audio features. Audio fingerprinting techniques typically rely on spectral representations, which are processed to be resistant to various types of noise. Because of such low-level representations, a query results in a match only if the exact same piece of music resides in the queried database. Genre or mood classification techniques mostly rely on MFCC coefficients and other low-level descriptors, leading to timbre-based similarity measures.

Query by melody is possible, if symbolic data are available [4]; for most recorded music this is not the case. Transcription and melody extraction techniques are improving, but are still unreliable - the most successful MIREX’06 melody extractor achieved ~73% accuracy [5]. Shwartz et al. [6] presented a system for querying audio collections by melody, but it requires a symbolic representation of the query and does not account for audio to audio matching.

Mid-level representations are an attempt to reduce the semantic gap between low-level and symbolic representations by extracting some higher-level semantic features from music signals, while still avoiding symbols. Dixon et al. [7] introduced rhythmic templates that represent typical rhythmic patterns of a piece and may be used for calculating rhythmic similarity. Bello and Pickens [8] introduced a mid-level harmonic representation, based on chroma features and showed its usability for segmentation.

Melody is an important descriptor of a piece of music and therefore very desirable for querying a music collection. For this purpose, we propose a mid-level melody-based representation, demonstrate how it can be used for retrieval in audio collections and present results obtained on the task of finding different interpretations of a song in a music collection.

2 Mid-level Melodic Representation

In our proposed mid-level representation, we seek to combine melodic and rhythmic aspects of a piece of music to obtain a representation suitable for calculating music similarity.

2.1 Finding dominant melodic lines

An ideal representation of a music signal that would be useful for the proposed task of query-by-melody (such as finding different interpretations of a musical piece), would contain representation of the melody of the analyzed piece. As current approaches for extraction of melody from music signals are still quite unreliable, we chose to use an approach that approximates melody by extracting dominant melodic lines in a piece of music. The proposed representation therefore contains most of the melody, as well as parts of other melodic lines in the analyzed piece that may belong to accompaniment.

Predominant pitches in the audio signal are extracted by a method modeled on Klapuri's joint multipitch extractor [9]. Predominant pitches are linked in time to form a set of melodic lines, that can be defined as triplets $m(t)=(f(t), a(t), s(t))$, representing time-varying frequency, amplitude and pitch salience of a melodic line. Overall, melodic lines contain most parts of the melody (~80% on the MIREX 2004/05 test set) with some additional fragments of other melodic lines, such as accompaniment when lead is not present. Average polyphony (concurrent number of melodic lines) on this test set is 1.3.

2.2 Tempo and octave invariance

Calculating intra- and inter-piece similarities is difficult when tempo varies and we therefore make our representation tempo-independent by using a beat tracker [10]. We first perform beat detection and then align the representation to the beat grid, thus making it beat-synchronous. Beat boundaries are used to resample the representation to 2 frames per beat (typically equal to an eight note), leading to a tempo-invariant melodic representation. In the process, we also resample the frequency axis to a semitone scale, resulting in 12 values per octave. The coarse scale has been selected to reduce effects of vibrato or similar pitch fluctuations on the representation. We also wrap the frequency axis to the range of one octave, resulting in a pitch-class type of representation, thus reducing octave errors, which are quite common in the melody extraction procedure used.

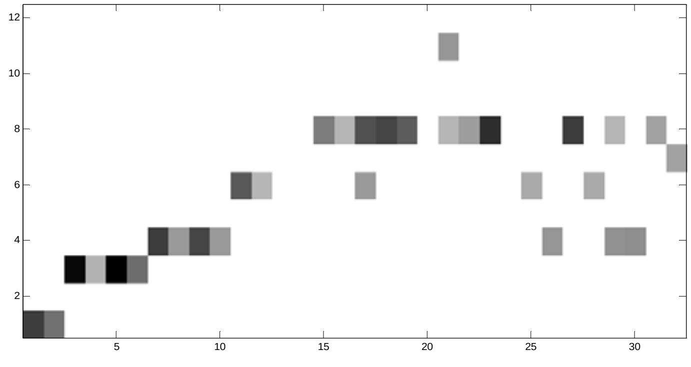

Figure 1. Mid-level melody-based representation of 16 beats of song Love is in the Air. Time (beats) is on horizontal and pitch class on vertical axis

The resulting mid-level representation contains most parts of the main melodic line, with some additional fragments of competing lines. It is octave and tempo invariant. An example is given in Figure 1, which shows an excerpt from "Love is in the Air" (sung by J.P. Young). The main melody is visible, as well as some other lines where more pitches were dominant.

3. Performing retrieval

We perform retrieval in an audio collection by comparing short fragments of melodic representations of songs in the collection. We divide the melodic representation of a song into a set of overlapping sections, which are *N*-beat long; as each section contains only dominant melodic lines, we call them melodic fragments. Fragments (such as the one seen in Figure 1) represent basic units of storage and retrieval in our retrieval system.

3.1 Key invariance

Most listeners perceive pitches within a melody as relative to one-another rather than as absolute values; melodies transposed to different keys are therefore still perceived as the same melodies, because intervals between pitches are preserved. To attain key invariance of melodic fragments, we apply a shift invariant transform to each fragment. Shift invariant transforms are widely used in image processing; after several experiments with different transforms, we decided to use the 2D power spectrum, which provides (circular) shift invariance in both pitch and time axes.

3.2 Locality Sensitive Hashing

To make retrieval efficient on large databases, we apply locality sensitive hashing to key-invariant melodic fragments. A locality sensitive hashing scheme is a distribution on a family of hash functions operating on a collection of objects, such that for two objects x, y, the probability of collision of hash values h(x) and h(y) of the two objects is proportional to a similarity function sim(x,y) defined on the collections of objects [11]. Hash values calculated with such hash functions are compact representations of objects so that similarity of objects can be estimated from their hash values, which leads to efficient algorithms for approximate nearest neighbor search and clustering. To hash melodic fragments, we use random hyperplane based hash functions, which are designed to approximate cosine similarity of hashed vectors. Cosine similarity of two fragments can be estimated by calculating the normalized Hamming similarity of hash values of two vectors.

3.3 Retrieval

Given a query, we perform retrieval by querying a database with each of the melodic fragments belonging to the query. For each fragment, its hash value is calculated. Normalized Hamming similarities of this hash value to values stored in the database are measures of similarity of the queried fragment to fragments in the database. Finding fragments similar to a given query fragment is therefore reduced to finding nearest neighbors of its hash value in the database. To perform this nearest neighbor search efficiently, we use the algorithm proposed by Charikar [11], which is based on random permutations of hash values. The algorithm results in a set of approximate nearest neighbors *NN*(*q*) of a queried fragment *q* in the database. To estimate the similarity of the query to all items in the collection, we use tf-idf (term frequency inverse document frequency) weighting on cosine similarity estimates of the returned nearest neighbors. The cumulative similarity measure of songs *q* and *s* can thus be written as:

$$\mathrm{sim}(q,s) = \sum_{f\in q} \sum_{g\in NN(f)\cap s} \mathrm{sim}(f,g)\, tfidf(g)\,, \qquad (1)$$

where *q* and *s* are the query and a song retrieved from the database, *f* and *g* fragments belonging to q and *NN*(*q*) respectively, sim the approximate cosine similarity and *tfidf* is the weighting function, which represents the importance of each melodic fragment in the database. The importance of a fragment increases as the frequency of the fragment increases in a song (i.e. refrain parts are more important than intros), but the degree of importance diminishes depending on how common the fragment is in all songs in the database (fragments often appearing in different songs are given lower importance). A high tf–idf weight is reached by high fragment frequency (in the given song) and low overall frequency of the fragment in the whole song collection.

A query thus results in a set of approximate nearest neighbors of the queried song, ranked according to cumulative similarity defined by eq. (1). Computational complexity of retrieval is in the order of $O(\log|S|)$, where $|S|$ is the number of songs in the database, which makes the approach suitable for real-world retrieval on large databases.

4. Experiment

The described approach to calculating similarity was tested for retrieval of different performances of a song from a larger song collection. For this task, we first collected a set of different performances of 34 songs, totaling 147 songs. Each song had from two to sixteen cover versions in this set, either by the same or by different performers. We injected the 147 songs into a larger collection of 2424 songs of similar, mostly pop and rock genres. The task was to retrieve the different performances of a given song from the collection. In the experiment, two tests were performed: one with audio and one with symbolic queries. Due to randomness of the hashing process and nearest neighbor search, retrieval was repeated ten times for each set of parameters.

Audio queries

We queried the collection with each of the 147 cover songs and calculated mean average precision of all queries. Best results were achieved with fragment sizes of 32 beats, which corresponds to 8 bars in 4/4 time signature, which in turn corresponds to the typical length of a phrase in verse and refrain sections in popular music. 160 bit long hash values were large enough to give good retrieval results for this fragment size. Overall, we obtained mean average precision of 0.4, which is comparable to other state of the art systems presented at MIREX 2007. Good retrieval results were achieved for songs with similar rhythmic and melodic interpretations, even though instrumentation and structure of songs were quite different. This was expected, as the nature of cosine similarity does not tolerate large deviations in either axis (pitch or time).

Symbolic queries

An advantage of the proposed approach is that it can handle symbolic queries (i.e. MIDI files) as well as audio queries. If we convert the symbolic representation of a song’s melody into pitch class piano roll representation, we can use the described retrieval algorithm to query a database of audio recordings. We collected symbolic representations of eight melodies of cover songs in our database and used them to estimate retrieval accuracy with symbolic queries. On average, performance with symbolic queries is comparable to audio queries. For some cases, where melody is difficult to estimate, performance may actually be better than with audio queries, while where performance suffers, it is mostly due to quantization of symbolic events and lack of dynamics (all notes are equally loud). Overall, songs with faster tempi and more rigid pop/dance interpretations are detected with high accuracy, while looser interpretations that may be successfully retrieved with audio queries because of similar phrasing, may not be detected with symbolic queries due to the their rigidity.

Acknowledgments

This work was supported in part by the Slovenian Government-Founded R&D Project V2-0216 ETHNOMUSE: multimedia digital archive of Slovenian folk music and folk dance culture.

References

[1] E. Selfridge-Field, "Conceptual and Representational Issues in Melodic Comparison," *Melodic Similarity: Concepts, Procedures, and Applications*, MA: MIT Press, 1998.

[2] J. Haitsma, and T. Kalker, “A highly robust audio fingerprinting system with an efficient search strategy,” *Journal of New Music Research*, vol. 32, no. 2, pp. 211-221, Jun, 2003.

[3] E. Pampalk, S. Dixon, and G. Widmer, “Exploring music collections by browsing different views,” *Computer Music Journal*, vol. 28, no. 2, pp. 49-62, Sum, 2004.

[4] R. Typke, F. Wiering, and R. C. Veltkamp, “Transportation distances and human perception of melodic similarity,” *Musicae Scientiae*, pp. 153-181, 2007.

[5] Wiki. "Music Information Retrieval Evaluation eXchange (MIREX) 2004-2007," http://www.music-ir.org/mirexwiki/index.php/Main_Page.

[6] S. Shalev-Shwartz, S. Dubnov, N. Friedman *et al.*, “Robust temporal and spectral modeling for query by melody,” in 25th annual international ACM SIGIR conference on Research and development in information retrieval, Tampere, Finland, 2002, pp. 331 - 338.

[7] S. Dixon, G. Fabien, and W. Gerhard, “Towards Characterisation of Music via Rhythmic Patterns,” in ISMIR 2004, 5th International Conference on Music Information Retrieval, Barcelona, Spain, 2004.

[8] J. P. Bello, and J. Pickens, “A Robust Mid-level Representation for Harmonic Content in Music Signals,” in International Symposium on Music Information Retrieval, London, UK, 2005.

[9] A. Klapuri, “Multiple fundamental frequency estimation by summing harmonic amplitudes,” in ISMIR, Victoria, Canada, 2006.

[10] S. Dixon, “Automatic extraction of tempo and beat from expressive performances,” *Journal of New Music Research*, vol. 30, no. 1, pp. 39-58, Mar, 2001.

[11] M. S. Charikar, “Similarity Estimation Techniques from Rounding Algorithms,” in ACM Symposium on Theory of Computing, 2002.