A Mid-level Melody-based Representation for Calculating Audio Similarity

Matija Marolt
University of Ljubljana
Trzaska 25
1000 Ljubljana, Slovenia
matija.marolt@fri.uni-lj.si

Abstract

We propose a mid-level melody-based representation that incorporates melodic, rhythmic and structural aspects of a music signal and is useful for calculating audio similarity measures. Most current approaches to music similarity use either low-level signal features, such as MFCCs that mostly capture timbral characteristics of music and contain little semantic information, or require symbolic representations, which are difficult to obtain from audio signals. The proposed mid-level representation is our attempt to bridge the gap between audio and symbolic domains by providing an integrated melodic, rhythmic and structural representation of music signals. The representation is based on a set of melodic fragments extracted from prominent melodic lines, it is beat-synchronous, which makes it independent of tempo variations and contains information on repetitions of short melodic phrases within the analyzed piece. We show how it can be calculated automatically from polyphonic audio signals and demonstrate its use for discovering melodic similarities between songs. We present results obtained by using the representation for finding different interpretations of songs in a music collection.

Keywords: music similarity, searching audio, melody-based representation, mid-level representation

1. Introduction

Calculating music similarity is one of the key areas in music information retrieval, as it enables searching and organization of music collections. Although melody is a very important 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 and are unique for each piece of music; 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'05 melody extractor achieved ~70% 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 calculating inter-song similarities 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, rhythmic and structural aspects of a piece of music.

2.1 Melody

The melodic aspects of the representation stem from our approach to melody extraction. The approach is briefly summarised as follows (for full description, see [9]). First, spectral modelling synthesis (SMS) is used to extract partials from audio signal, which are then subjected to a psychoacoustic masking model. Predominant pitches are extracted from partials with an EM approach, which estimates the most likely pitches to have generated the observed series of partials. Using SMS partial tracking information, the found pitches are linked in time, resulting

in a series of pitch tracks, which are then filtered by removing short and muted tracks. We call this final set of pitch tracks *melodic fragments*, because they represent different parts of melodic lines (lead and accompaniment) in the analyzed piece. Each fragment has its start and end time, time-varying loudness and time-varying pitch. Due to the EM approach taken, in regions of audio where only one pitch is dominant, only a single melodic fragment will be found, while in regions where several pitches are competing for attention, several fragments will appear simultaneously. The resulting representation contains most parts of the main melodic line (~90% on our test set consisting of parts of MIREX'04 and '05 melody extraction datasets) with additional fragments of other melodic lines, especially when lead is not present.

2.2 Rhythm

Events in a piece of music are not perceived in direct relation to time, but in relation to their place within a metric hierarchy, whose basic elements are beats. These in turn relate to time according to tempo and its variation within a piece. Calculating intra- and inter-piece similarities is difficult when tempo varies; dynamic programming approaches can be used to alleviate this problem [10]. Instead, we prefer to make our representation tempo-independent by using a beat tracker [11] to perform beat detection and then aligning the representation to the beat grid, thus making it beat-synchronous. Beat boundaries are used to resample the representation with an averaging filter to 6 frames per beat, leading to a tempo-invariant melodic representation. The resampling rate was chosen experimentally; smaller values had a negative effect on performance, while larger values did not lead to improvements. In the process, we also resample the frequency axis to a half-tone scale, resulting in 24 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 sidestepping octave errors, which are quite common in the melody extraction procedure used.

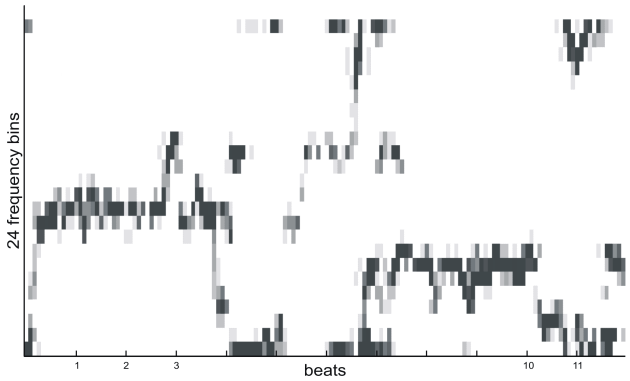

Figure 1. Mid-level melody-based representation of 12 beats of song Love is in the Air

The resulting mid-level representation contains most parts of the main melodic line, together with some fragments of competing lines and 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). Melodic fragments belonging to lead vocals are visible in the beginning and middle sections, as well as several other fragments that mostly occur between vocal parts.

2.3 Structure

We infer the structure of a piece by calculating the self-similarity matrix $\mathbf{S}^{(l)}$ of the beat-synchronous melodic representation. Each element i,j of matrix $\mathbf{S}^{(l)}$ is defined as:

$$s_{ij}^{(l)} = d(b_{i..i+l}, b_{j..j+l}) \qquad i = 1..n, j = 1..n. \quad (1)$$

b_i represents a vertical slice of the beat-synchronous representation at beat position i (24 frequency bins by 6 frames per beat). $b_{i..i+l}$ represents a sequence of slices b starting at beat i and ending at beat $i+l$; values for beat positions beyond the total number of beats n ($i>n$) are set to 0. d is a similarity measure; after some experiments, we decided to use the cosine similarity measure [14], which performed roughly the same as symmetric KL distance or correlation. Since silence matches silence well, we also add a small amount of random noise to slices b_i to avoid high similarity scores for regions of silence. $\mathbf{S}^{(l)}$ is thus a square nxn matrix (n is the total number of beats) and contains similarities between all pairs of excerpts of length l beats taken from the representation.

Parameter l controls the length of excerpts to be compared. l is estimated by calculating the self-similarity matrix $\mathbf{S}^{(l)}$ with a sequence length of l=12 beats, accounting for 3/4 and 4/4 bars. Elements of the matrix above the main diagonal are then averaged across diagonals:

$$d_k = \frac{1}{n-k}\sum_{i=1}^{n-k} s_{i,i+k} \qquad k = 1..n-1 \quad (2)$$

and autocorrelation of resulting vector d calculated. Peaks of the autocorrelation function correspond to typical gaps between repetitions of melodic patterns in the piece. The highest autocorrelation peak above 9 is taken as length l, so that compared sequences are at least 10 beats long. We chose this threshold experimentally, as we found that lengths smaller than 10 beats may contain too little information to be compared reliably. Typical values of parameter l are 16 beats for pieces in 4/4 and 12 or 24 beats for 3/4 pieces.

We also found that multiplying (element-wise) matrices calculated at different lengths l results in a significant noise reduction. We therefore calculate the final self-similarity matrix $\mathbf{S}$ by combining matrices calculated with three different values of l (eq. 3). In addition, we apply filter $\mathbf{F}^{(s)}$ to the resulting product. The filter is a square sxs matrix with diagonal elements equal to 1 and other

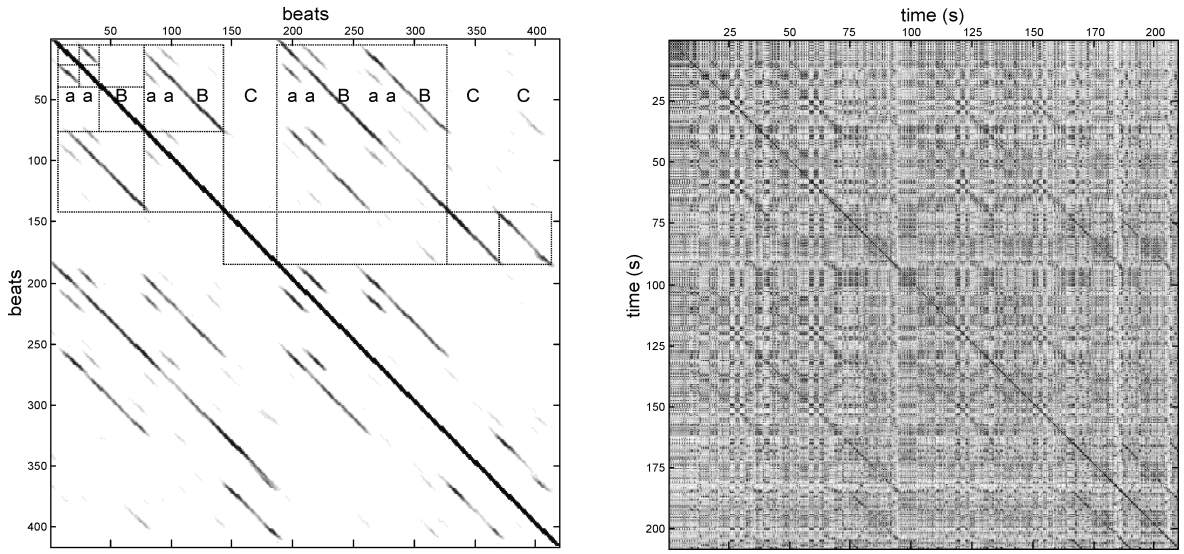

Figure 2. Self-similarity matrix calculated from the beat-synchronous representation of Love is in the Air with structure indicated (left). For comparison: self-similarity matrix calculated from chroma features

elements equal to $-1/(s\text{-}1)$, thus emphasizing diagonals and suppressing off-diagonal repetitions. This reduces the effect of long notes that produce square blocks in the matrix. The self-similarity matrix is calculated as:

$$\mathbf{S} = \left(\mathbf{S}^{(l)} \bullet \mathbf{S}^{(0.75l)} \bullet \mathbf{S}^{(0.5l)}\right) * \mathbf{F}^{(0.25l)} \quad (3)$$

where • denotes element-wise matrix multiplication and * the convolution operator. The left side of Figure 2 shows the matrix calculated for song "Love is in the Air". Due to beat-synchronous melodic representation and long comparison sequence (l=16), the figure clearly reflects the structure of the piece. Diagonals indicate repeated sections and reveal the structure of the piece to be aaBaaBCaaBaaBCC. The structure can be derived from the top part of the matrix, as indicated by overlaid dotted lines. If we compare the resulting matrix to self-similarity matrices calculated from MFCC or chroma features (i.e. [13]), the amount of noise is greatly reduced and structure revealed in places that may be ignored by these approaches due to different timbres or harmonies. We contribute this difference to a more musically meaningful melody-based representation and longer sequences used for self-similarity calculation. Although explicit segmentation is not our goal in this research, obtained self-similarity matrices show promise that the mid-level representation could also be useful for segmentation. We use self-similarity information for extraction of melodic patterns as described in the next section.

3. Calculating Similarity

Our goal is to obtain a representation that would emphasize melodic aspects of a given piece and would be useful for calculating melodic similarity of different pieces, giving high scores for pieces with similar melody even if they have different tempi, different instrumentation or different arrangements. To assess whether the proposed mid-level representation is suitable for such tasks, we tested it within a simple music retrieval system, based on comparisons of melodic patterns extracted from each song.

3.1 Finding Melodic Patterns

We define melodic patterns as parts of melody that are repeated several times in a song. They may be parts of a chorus or verse or entire chorus/verse segments. We use these patterns as a summarized description of a song and use them for calculating similarity. The idea of using patterns to characterize music is not new; Paulus and Klapuri [12] used rhythmic patterns to measure similarity and Dixon et al. [7] to characterize ballroom dances.

The algorithm for extracting melodic patterns follows an approach similar to segmentation and chorus extraction methods, such as Goto's RefraiD algorithm [13]. It is based on the self-similarity matrix, calculated as described previously. Patterns are extracted from the matrix by a simple greedy approach, which can be outlined as:

1. find the most repeated melodic pattern. We first sum the self-similarity matrix across one dimension, resulting in a vector with high values at beat positions that are often repeated. We then smooth this vector with a Gaussian filter. Position of the highest peak in the resulting smoothed vector is taken to lie within the most repeated pattern. The first and last beat of the pattern are then searched for by searching the vector from the chosen peak forward and backward until the first local minima before and after the peak are met. Beat positions of the minima are taken to represent the start and end position of the most repeated pattern;
2. find all salient repetitions of this pattern in the matrix.

3. remove the pattern and all found repetitions from the self-similarity matrix and add them to the list of found patterns, if more than one repetition was found.
4. repeat steps 1-3 until most of the initial matrix is removed (we use a fixed threshold of 90%).

The result is a set of melodic patterns (typically 2-4 on our database consisting of mostly pop/rock songs). Each pattern is characterized by its start and end beat locations and a list of repetitions. We also calculate the median mid-level melodic representation of each pattern and all of its repetitions, which keeps only the most salient features (melody) and removes some background accompaniment that may vary within the piece.

3.2 Compensating for Difference in Keys

If we calculate melodic patterns of all songs in a song collection, we can use the obtained patterns to calculate each song's similarity to all other songs in the collection. This can be done by comparing the median mid-level representation of each pattern of a song to representations of each melodic pattern of the compared song. To perform the comparison, we first need to account for possible differences in keys of both songs.

A key profile of each song is first calculated. We start by summing the entire mid-level representation of a song across time, resulting in the song's pitch profile. We calculate the dot product of the obtained pitch profile with Bayesian key profiles [15] for all 24 major and minor keys, resulting in the song's key profile. When we compare two songs, key profiles of both songs are correlated in all shifted positions and the best match is taken to represent the difference in key between the two songs. This difference is then compensated for by a circular shift of melodic patterns of one of the songs. If we assume that two compared songs have similar melody, such procedure leads to correct key compensation most of the time.

3.3 Calculating Similarity

After key compensation, comparison of a pair of melodic patterns is straightforward. Since patterns will not usually be time-aligned and of equal size, similarity is calculated by shifting the shorter pattern beat-by-beat over the length of the longer pattern and calculating similarity of each shifted position. We again use the cosine similarity measure, the same as used for the self-similarity matrix calculation. The highest similarity of all shifted positions is taken as the pattern similarity score. After similarities of all pairs of patterns of two songs have been calculated, the mean similarity of n best matching patterns is taken as similarity of the two songs. Best value for n was experimentally determined to be 2.

4. Experiment and Discussion

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 8 songs, totaling 36 songs. Each song had at least four different versions in this set, either by the same or by different performers. Beat tracking of songs was manually checked for errors and corrected. The list of songs can be found at http://lgm.fri.uni-lj.si/~matic/similarity. We injected the 36 songs into a larger collection of 1820 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, we compared two different representations of music: the proposed melody-based representation and a representation based on chroma feature vectors [13]. With both, we used the same locations of melodic patterns and the same key compensation technique, only the actual representations of melodic patterns varied. Evaluation was performed by considering each of the known 36 pieces as a query and calculating its similarity to all other pieces in the collection. Table 1 lists 11 point precision averages and percentages of hits in top 5 returned songs for all 36 queries and separately for each of the songs used in queries. 11 point precision averages are often used for MIR system evaluation [4] and are calculated as averages of precision at recall levels 0%, 10%,...,100%. Results for both compared representations are given in columns 3 and 4. Column 5 lists retrieval results obtained with a similarity measure calculated as a sum of similarity measures of both approaches, resulting in a new combined melody+chroma similarity measure. Number of song versions and performers is given in column 2 (n.v./n.p. - see also the list of songs at http://lgm.fri.uni lj.si/~matic/similarity). Note that the queried song was always returned first in the list of hits and was therefore excluded when calculating 11 point precision averages and top 5 scores.

Although we used a very simple method for calculating similarity, we achieved solid retrieval results with a number of songs (1,2,3,7 in Table 1), even though performers and arrangements differed (i.e. Mamas and Papas, Dean Martin, Beautiful South and Ella Fitzgerald for song 3). This shows the robustness of the presented representation to changes in instrumentation.

If songs differ a lot in melody or rhythm, the presented approach fails completely (songs 4 and 8). This has less to do with the mid-level representation than with the simple approach taken to extracting melodic patterns and comparing songs; a dynamic programming approach that would allow for rhythmic variations might lead to better results. Other examples of false negatives are found in songs with several concurrent melodic lines (such as the

Table 1. 11 point precision averages and percentages of hits in top 5 returned songs for melody-based representation, chroma representation and a combined approach

		melody		**chroma**		**combined**	
	n.v ./ n.p.	*11pt. prec*	*% top 5*	*11pt. prec*	*% top 5*	*11pt. prec*	*% top 5*
Mean precision and % of hits in top 5 for all queries		0.20	25	0.15	19	0.22	27
1. A hard day's night	4 / 2	0.35	33	0.39	5	0.40	42
2. All by myself	4 / 4	0.33	42	0.12	17	0.33	42
3. Dream a little dream	5 / 4	0.44	55	0.28	35	0.51	5
4. Georgia on my mind	4 / 4	0	0	0	0	0	0
5. Goodnight Irene	4 / 4	0.13	25	0	0	0.03	8
6. Knockin' on heaven's door	5 / 5	0.02	0	0.20	35	0.20	35
7. Love is in the air	5 / 4	0.33	4	0.18	15	0.32	35
8. Summertime	5 / 5	0.01	0	0	0	0	0

baroque interpretation of song 1), where the mid-level representation either includes too many melodic lines, or picks out the wrong ones. The segmentation procedure is another source of errors (i.e. song 2 by Celine Dion or interpretations of song 6). When the found melodic patterns are not at least approximately time-aligned between interpretations, songs are not found to be similar, even though their melodies and rhythms (and consequently the beat-synchronous melodic representation) are similar. A somewhat more elaborate segment extraction procedure could be the solution for this type of errors.

We have also found the existence of the so-called hubs [16], which are defined as songs that occur frequently as a false positive according to a given similarity measure. Most of these songs turned out to be a problem for either our melody extraction or beat detection algorithms. This led to poor mid-level representations and unrepresentative segments (few segments, short segments or segments full of melodic lines) that appear to match well with segments of most songs. Otherwise, false positives were mostly due to either errors in extraction of melodic lines (too many lines, incorrect lines) or beat tracking errors (incorrect resampling and segment extraction).

Results obtained by using chroma feature vectors indicate that these work well when performances are by the same performer or have very similar arrangements, such as with song 1, where three of four performances are by The Beatles and also for song 6, where arrangements are very similar. In such cases, results are comparable or better than by using the melody-based representation. With different performers and arrangements, however, chroma features lag behind due to their greater dependence on arrangements in comparison with the proposed representation. We have also made an experiment by summing similarity scores of both representations into a combined similarity score, which additionally improved retrieval accuracy (Table 1, column 5).

We should also comment on the possible bias introduced by errors of the beat tracking algorithm that was used to automatically extract beat positions in songs from the database (for the queried songs, beat positions were manually checked for errors and corrected). Although it is hard to estimate how errors in beat tracking affect our results, inspection of false positives shows that these are often caused by incorrect beat tracking, so we could speculate that beat tracking errors affect false positives as well as true negatives and do not introduce a large positive or negative bias into the presented results.

5. Conclusion and Further Work

We described a novel mid-level representation that integrates melodic, rhythmic and structural aspects of a music signal. We described its use for calculating melodic similarity in audio collections. Results show that the proposed representation provides a good basis for tasks such as retrieval of different interpretations of a song from a song collection even if interpretations are by different artists and have different arrangements. We acknowledge that our tests were made on a small database, but the retrieval method used was also very primitive and we argue that better techniques, such as dynamic programming or HMM modeling should lead to good results also on large databases. We also plan to augment the melodic representation with harmonic information [8]; a combined approach should lead to additional improvements, as is indicated by better results achieved with the combined melody+chroma similarity measure. The proposed representation also seems to hold promise for segmentation and summarization, which is another set of tasks we plan to pursue further.

6. Acknowledgments

This work was supported by an International Short Visit grant from the Royal Society that enabled the author to spend six weeks at the Centre for Digital Music, Queen Mary, University of London. I would like to thank Dr. Juan Bello and Dr. Mark Plumbley for invitation, their ideas and support during the visit.

References

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

[2] J. Haitsma, T. Kalker. "A Highly Robust Audio Fingerprinting System," in *ISMIR 2002*, *Proc*., 2002.

[3] E. Pampalk, S. Dixon, and G. Widmer. "Exploring Music Collections by Browsing Different Views," *CMJ*, Vol. 28, No. 2, pp 49-62, 2004.

[4] A.L. Uitdenbogerd. "Music Information Retrieval Technology," *Ph.D. Thesis*, RMIT, 2002.

[5] "MIREX 2005 - 1st Annual Music Information Retrieval Evaluation eXchange,", [Web site] 2005, Available: http://www.music-ir.org/mirex2005/index.php/Main_Page

[6] S. Shwartz, S. Dubnov, N. Friedman, Y. Singer. "Robust Temporal and Spectral Modeling for Query By Melody," *Proc. ACM SIGIR*02*, Tampere, Finland, 2002.

[7] S. Dixon, F. Gouyon, G. Widmer, "Towards Characterization of Music via Rhythmic Patterns," in *ISMIR 2004 Proc.,* 2004.

[8] J.P. Bello, J. Pickens. "A Robust Mid-level Representation for Harmonic Content in Music Signals," in *ISMIR 2005, Proc*. London, UK. September 2005.

[9] M. Marolt. "Audio Melody Extraction Based on Timbral Similarity of Melodic Fragments," in *Proceedings Eurocon 2005*, Belgrade, 2005.

[10] R.J. McNab, L.A. Smith, I.H. Witten, C.L. Henderson, S.J. Cunningham. "Towards the digital music library: Tune retrieval from acoustic input," Proceedings of Digital Libraries '96. ACM, 1996.

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

[12] J. Paulus, A. Klapuri. "Measuring the similarity of rhythmic patterns," in *ISMIR 2002 Proc.,* 2002.

[13] M. Goto. "A chorus-section detecting method for musical audio signals," *ICASSP 2003 Proc.,* 2003.

[14] J. Foote. "Visualizing music and audio using self-similarity," *Proc. ACM international conference on Multimedia*, 1999.

[15] D. Temperley. "A Bayesian Key-Finding Algorithm," in *Music and Artificial Intelligence*, Springer, 2002.

[16] J.J. Aucouturier. "Ten Experiments on the Modelling of Polyphonic Timbre," Ph.D. Thesis, L'Universite Paris 6, 2006.