Identifying ‘Cover Songs’ with Beat-Synchronous Chroma Features

Daniel P.W. Ellis
LabROSA, Dept. of Electrical Engineering
Columbia University, New York NY 10027 USA
dpwe@ee.columbia.edu

Abstract

Large music collections, ranging from thousands to millions of tracks, are unsuited to manual searching, motivating the development of automatic search methods. When two musical groups perform the same underlying song or piece, these are known as ‘cover’ versions. We describe a system that attempts to identify such a relationship between music audio recordings. To overcome variability in tempo, we use beat-tracking to describe each piece with one feature vector per beat. To deal with variation in instrumentation, we use 12-dimensional chroma feature vectors that collect spectral energy supporting each semitone of the octave. To compare two recordings, we simply cross-correlate the entire beat-by-chroma representation for two tracks and look for sharp peaks indicating good local alignment between the pieces. Evaluation on a small set of 15 pairs of pop music cover versions identified within the USPOP2002 collection achieves a performance of around 60% correct.

Keywords: Music Similarity, Cover Songs, Chroma Features, Beat Tracking

1. Introduction

Immediate access to large music collections is now commonplace – be they the thousands of songs on the MP3 player in your pocket, or the millions of songs available at online music stores. But finding music within such collections can be very problematic, leading to the current interest in automatic music similarity estimation. In this paper, we address a slightly different problem: rather than trying to find music whose genre, style, or instrumentation match particular query examples, we are trying to find versions of the *same piece* of music, despite the fact that they may be performed with very different styles, instrumentation, etc. These alternate versions of the same underlying piece of music are known as ‘cover versions’.

Cover versions will typically retain the essence of the melody and the lyrics (for a song) but may vary greatly in other dimensions. Indeed, in pop music, the main purpose of recording a cover version is typically to investigate a more-or-less radically different interpretation of a song (although in different recordings of classical music the variations may be more subtle). Thus, to solve this problem, we must devise representations and matching schemes that are robust to changes in tempo, instrumentation, and general musical style.

2. Overview

Our representation has two main features: We use a beat tracker to generate a beat-synchronous representation with one feature vector per beat. Thus, variations in tempo are largely normalized as long as the same number of beats is used in each phrase. The representation of each beat is a normalized chroma vector, which sums up spectral energy into twelve bins corresponding to the twelve distinct semitones within an octave, but attempting to remove any distinction between different octaves. Chroma features capture both melodic information (since the melody note will typically dominate the feature) and harmonic information (since other notes in chords will result in secondary peaks in a given vector).

To match two tracks represented by such beat-vs-chroma matrices, we simply cross-correlate the entire pieces. Long sequences of beats with similar tonal structure will result in local maxima at the appropriate lags in the cross-correlation, with the size of the peak increasing both with the degree of similarity in the chroma features, and the length of matching sequences. To distinguish between genuine matches and incidental high cross-correlations, we emphasize rapid variations in the cross-correlation (i.e. particular lags at which alignment is high despite being low at neighboring lags) through high-pass filtering. To accommodate transposition between versions (performances in different keys), we cross-correlate between all twelve possible semitone transpositions of the chroma vectors.

3. Beat tracking

Our beat-tracker is based on the description of Jehan [5]. A log-magnitude 40-channel Mel-frequency spectrogram is calculated for 8 kHz downsampled mono versions of the original recording with a 32 ms window and 8 ms hop between frames. The first-order difference along time in each frequency channel is half-wave rectified (to leave only onset information) then summed across frequency. This “onset envelope” is high-pass filtered with a 3 dB point at 0.01

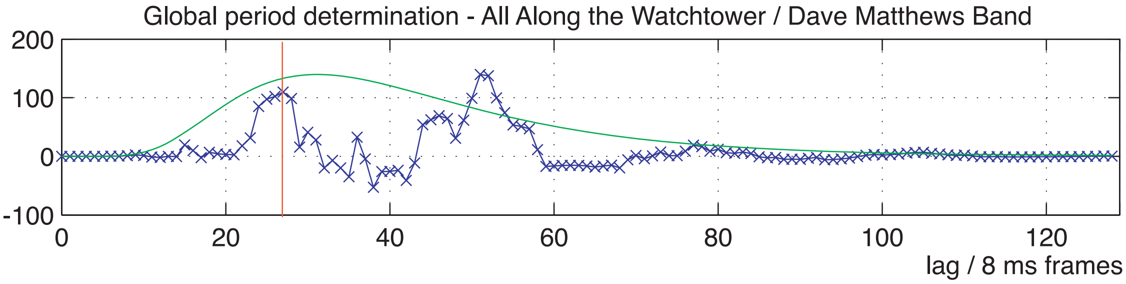

Figure 1. Autocorrelation of the first 90 s of a piece, used to choose the global target tempo. The Gaussian weighting window is shown overlaid, and the chosen period (27 samples = 278 bpm) is shown by a vertical line.

rad/samp to remove d.c. offset, and the first 90 s of the piece are autocorrelated out to a lag of 128 points (1.024 s). This autocorrelation is windowed with a Gaussian in log-period centered on 240 bpm, with a half-width of 1.5 octaves. Then the shortest lag that is a local maximum with a value at least 0.4 times the largest maximum in the windowed autocorrelation is taken as the global target period. This favors the multiple of the basic beat of the piece that is closest to 240 bpm i.e. closer to the tatum (shortest melody note duration) than what would be the notated tempo of the piece. Figure 1 shows an example of the global autocorrelation

The onset envelope is then filtered by a periodicity enhancing smoothing window composed of cos^8 at the global target period, Hann-windowed out to ±3 periods. Beats are then chosen as the local maxima of this enhanced onset function within each beat-length window centered one beat on from the last marked beat. However, if no maxima reaches 0.25 of the magnitude of the last-picked maxima, the default predicted beat position is used instead, and the search continues forward. This allows the tracker to continue through short stretches of weak or absent beat.

4. Chroma features

To the extent that the beat tracking can identify the same main pulse in different renditions of the same piece, representing the audio against a time base defined by the detected beats normalizes away variations in tempo. We choose to record a single feature vector per beat, and use twelve element 'chroma' features to capture both the dominant note (typically melody) as well as the broad harmonic accompaniment [4, 1]. The idea of calculating harmonic features over beat-length segments appears to have been developed several times; we first became aware of it in [6].

Rather than using a coarse mapping of FFT bins to the chroma classes they overlap (which is particularly blurry at low frequencies), we use the phase-derivative within each FFT bin both to identify strong tonal components in the spectrum (indicated by spectrally-adjacent bins with close instantaneous frequencies) and to get a higher-resolution estimate of the underlying frequency [2]. We found that using only components up to 1 kHz in our chroma features worked best. Figure 2 shows an example of the chroma features alongside the beat-tracked mel spectrum of the fragment they describe.

5. Matching

From the processing so far, we have each recording represented by a matrix of 12 chroma dimensions by however many beats are detected in the entire piece. We expect cover versions to have long stretches (verses, choruses, etc.) that match reasonably well, although we don't particularly expect these to occur in exactly the same places, absolutely or relatively, in the two versions. We initially experimented with chopping one piece up into multiple fragments and looking for the best cross-correlation of each fragment in the test piece, but in addition to being very slow it was difficult to choose the best length of fragment size. In the end, the simpler approach of cross-correlating the entirety of the two matrices gave us the best results. Although this is unable to reward the situation when multiple fragments align but at different relative alignments, it does have the nice property of rewarding both a good correlation between the chroma vectors and a long sequence of aligned beats, since the overall peak correlation is a product of both of these. Chroma vectors are intrinsically non-negative; we scaled them to have unit norm at each time slice. The cross-correlation is further normalized by the length of the shorter segment, so the correlation results are bounded to lie between zero and one. We perform the cross-correlation twelve times, once for each possible relative rotation (transposition) of the two feature matrices.

We observed, however, a number of spurious large correlations from relatively long stretches dominated by a single chroma bin; this occurs in many tracks. We found that genuine matches were indicated not only by absolutely large cross-correlations but also by sharp local maxima in cross-correlations that fell off rapidly as the relative alignment changes from its best value. To emphasize these sharp local maxima, we choose the transposition that gives the largest peak correlation then high-pass filter that cross-correlation function with a 3 dB point at 0.1 rad/sample. The 'distance' reported for the evaluation is simply the reciprocal

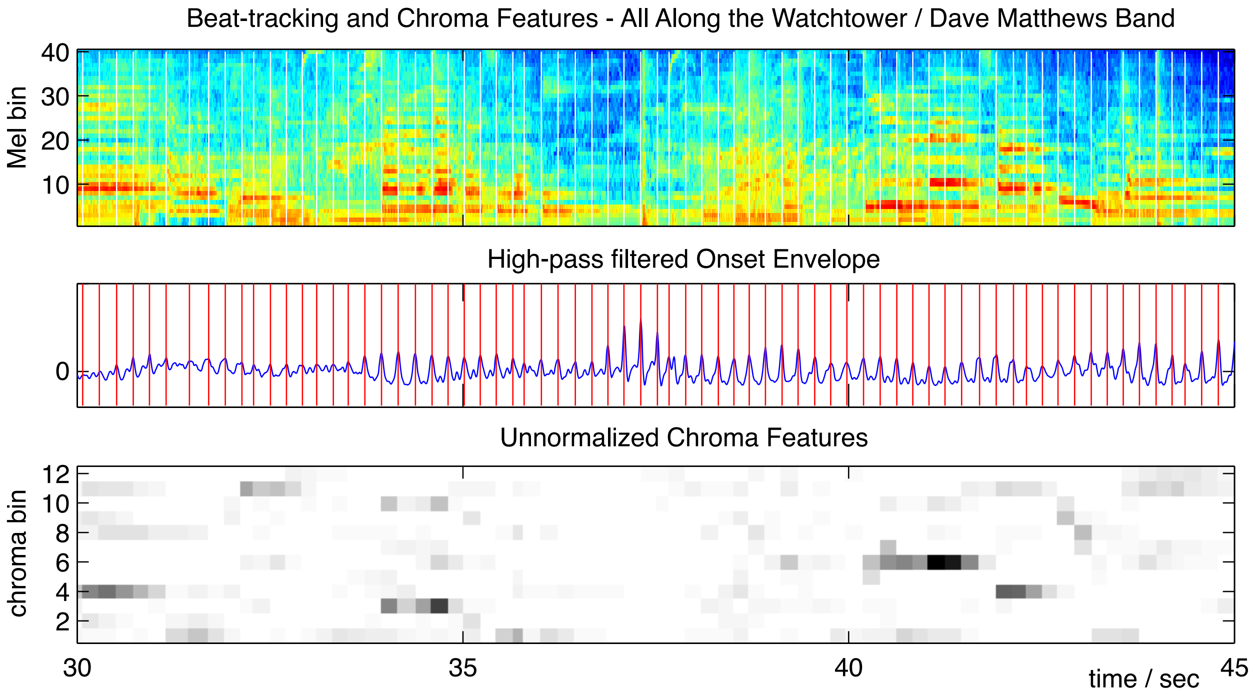

Figure 2. Excerpt showing the Mel-scale spectrogram (top pane), the periodicity-enhanced onset envelope (middle pane, with chosen beat instants indicated), and the unnormalized per-beat chroma feature vectors (bottom pane).

of the peak value of this high-pass filtered cross-correlation; matching tracks typically score below 20, whereas unrelated tracks are usually above 50.

Matching will fail if the feature extraction is based on beats with different relations to the music i.e. if one version tracks twice as many beats per song phrase. To accommodate this, we experimented with including two representations of each track, the original plus one using double the beat length (i.e. around 120 bpm) but this did not offer any advantage in our experiments.

6. Evaluation

We developed the system on set of 15 pairs of pop-music tracks that were alternate versions of the same song. They were extracted from the USPOP2002 dataset [3] by making a list of all tracks from the total set of 8764 tracks that had the same name, then listening to each pair to see if they were in fact the same piece; about 20% were. We stopped after we had found 15 pairs. Interestingly, it was often hard to tell if two tracks were the same until the verse began, at which point the lyrics quickly indicated matching tracks.

We made two lists of tracks, each containing one of the two versions of each track. In the evaluation, each track in the A list was compared to every track in the B list, and called a cover version of the track that it was most similar to; thus, the task was to identify the cover version knowing that one exists, rather than deciding if two songs were similar enough to be considered covers. Our best system (over variations in parameters such as filter breakpoints for the chroma features and matching) correctly identified 10 of 15 tracks; typical performance varied between 6 and 9 correct (where guessing would give one). Four of the pairs were clearly difficult for our representation and were almost never correctly identified. The test set is detailed in table 1.

7. Conclusions

Identifying cover tracks is an interesting new direction for content-based search of music audio databases. However, it is much more computationally expensive than the time-insensitive feature-distribution models typically used in genre and artist classification: our initial experiments took up to 30 s to compare each pair of tracks, making search in large databases completely intractible; we managed to speed this up by a factor of 100, but this still limits the size of database that we can afford to search by such direct means.

Our plan is to use these techniques to identify a dictionary of smaller fragments that can provide the most efficient coverage of large music databases. These can then be used as (possibly redundant) ‘index terms’ to permit the use of more rapid indexing schemes, as well as potentially revealing interesting repeated motifs and shared structure within music collections.

Table 1. Cover version test set from uspop2002, along with typical system performance.

Title	"A" artist	"B" artist	comments
Abracadabra	Steve Miller Band	Sugar Ray	easy
Addicted to Love	Robert Palmer	Tina Turner	hard
All Along the Watchtower	Dave Matthews Band	Jimi Hendrix	hard
America	Paul Simon (live)	Simon and Garfunkel	
Before You Accuse Me	Creedence Clearwater Revival	Eric Clapton	
Blue Collar Man	REO Speedwagon	Styx	easy
Caroline No	Beach Boys	Brian Wilson (live)	easy
Cecilia	Paul Simon (live)	Simon and Garfunkel	very hard
Claudette	Everly Brothers	Roy Orbison	easy
Cocaine	Eric Clapton	Nazareth	
Come Together	Aerosmith	Beatles	easy
Day Tripper	Beatles	Cheap Trick	easy
Faith	George Michael	Limp Bizkit	
God Only Knows	Beach Boys	Brian Wilson (live)	hard
Gold Dust Woman	Fleetwood Mac	Sheryl Crow	easy

Acknowledgments

This work was supported by the Columbia Academic Quality Fund, and by the National Science Foundation (NSF) under Grant No. IIS-0238301. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF.

References

[1] M. A. Bartsch and G. H. Wakefield. To catch a chorus: Using chroma-based representations for audio thumbnailing. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk, New York, October 2001.

[2] F. J. Charpentier. Pitch detection using the short-term phase spectrum. In *Proc. ICASSP-86*, pages 113–116, Tokyo, 1986.

[3] D. Ellis, A. Berenzweig, and B. Whitman. The "uspop2002" pop music data set, 2003. `http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html`.

[4] T. Fujishima. Realtime chord recognition of musical sound: A system using common lisp music. In *Proc. ICMC*, pages 464–467, Beijing, 1999.

[5] T. Jehan. *Creating Music by Listening*. PhD thesis, MIT Media Lab, Cambridge, MA, 2005.

[6] N. C. Maddage, C. Xu, M. S. Kankanhalli, and X. Shao. Content-based music structure analysis with applications to music semantics understanding. In *Proc. ACM MultiMedia*, pages 112–119, New York NY, 2004.