IDENTIFYING 'COVER SONGS' WITH CHROMA FEATURES AND DYNAMIC PROGRAMMING BEAT TRACKING

Daniel P.W. Ellis and Graham E. Poliner

LabROSA, Dept. of Electrical Engineering
Columbia University, New York NY 10027 USA
{dpwe,graham}@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 different musicians 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 several databases indicate good performance, including best performance on an independent international evaluation, where the system achieved a mean reciprocal ranking of 0.49 for true cover versions among top-10 returns.

Index Terms— Music, Database searching, Acoustic signal analysis, Dynamic programming, Correlation

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 specific version of this 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 often to investigate a 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.

[12] considered this problem for popular music with a system that attempted to extract only the sung melody as a tempo-independent note sequence. [11] used semitone-based chroma features (as we do) and temporal blurring to identify alternate performances of classical orchestral pieces. [3] presented a number of approaches to matching sequential content (such as melody and harmony) in music recordings, emphasizing the weaknesses of many music similarity systems that largely collapse signal statistics across time. This paper presents a system drawing on all of these ideas, rendered to a working state, and evaluated on a relatively large and realistic database.

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 the 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. Although it is very unlikely that cover versions will match beat-for-beat along the whole duration, we found this approach more successful than trying to establish a more precise correspondance. Sub-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. Matching sequences in many cases will be only a small portion of the total track, yet will still contribute a peak large enough to correctly indicate the match. To distinguish between genuine matches and incidental high cross-correlations, we apply high-pass filtering (along time time skew dimension) to emphasize rapid variations in the cross-correlation – i.e. particular lags at which alignment is suddenly high despite being low at offsets only one or two beats different. To accommodate transposition between versions (performances in different keys), we cross-correlate between all twelve possible semitone transpositions (rotations) of the chroma vectors. [1]

3. BEAT TRACKING

Since our basic representation consists of one feature vector per beat, we must start by identifying the beat segmentation times in the music audio. Our system is based on the one described in [9], extended to

[1] More details and Matlab code are available at http://labrosa.ee.columbia.edu/projects/coversongs/.

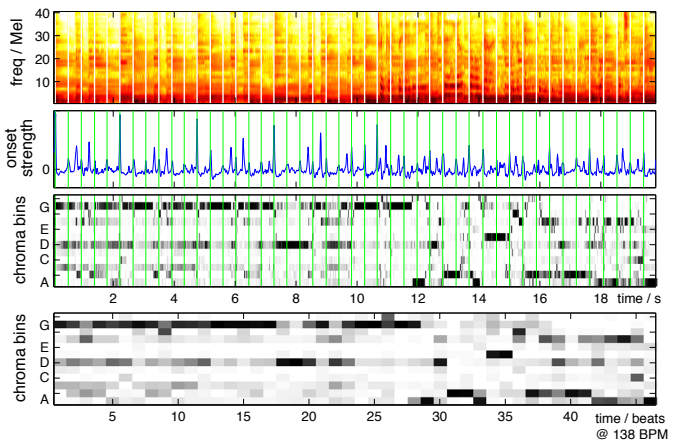

Fig. 1. Excerpt showing the Mel-scale spectrogram (top pane), the smoothed onset strength envelope (second pane), per-frame chroma vectors (third frame), and per-beat chroma vectors (bottom frame) for the first 20 s of the Elliot Smith track. Chosen beats are shown as vertical divisions. Notice the extensive syncopation (strong onsets midway between perceived beats).

use dynamic programming. The description here is brief; for more details please see [6].

The first stage of beat tracking converts the audio into an "onset strength" value at a 250 Hz sampling rate. This is derived by taking the first-order difference along time in a log-magnitude 40-channel Mel-frequency spectrogram, throwing away negative values, then summing across frequency. Slowly-varying offsets (corresponding to variations in gain in the original signal) are removed by a high-pass filter at about 0.5 Hz. Then, an approximate global tempo is estimated by autocorrelating the onset strength, applying a 'preference window' which is a Gaussian on a log-time axis, and choosing the period with the largest windowed autocorrelation as the tempo. As discussed in section 6, we varied the center of the preference window between 0.25 and 0.5 s – i.e. between 240 and 120 beats per minute (BPM) – to obtain beat segments at different points in the metrical hierarchy of the music.

We then use dynamic programming to find the set of beat times that optimize both the onset strength at each beat (to prefer the strongest onsets as beats) and the spacing between beats (to reflect the global tempo parameter set in the previous stage). Dynamic programming is an efficient way to search all possible beat sequences to optimize a total cost that can be broken down into a local score at each beat time (the onset strength), and a transition cost. Conventionally, the transition cost is additive, but we implemented it as a scaling window, again a Gaussian on a log-time axis, applied to the onset strength envelope for $0.5 \ldots 2.0\times$ the tempo period prior to the current time, with the maximum at the target period. For every possible beat time, the best preceding beat time is located (as the maximum of the scaled onset strength within the window), and the cumulative score up to that beat is calculated. Then, the largest score close to the end of the audio is located, and the entire sequence of beats leading to that beat time is recovered through a 'backtrace' table storing the predecessor for every beat time.

Figure 1 shows an example of the beats found in the first 20 s of "Drink Up Baby" performed by Elliott Smith. This track consists of guitar and vocals only, and includes significant syncopation, making it a challenge for beat tracking. The advantage of dynamic programming is that it effectively searches all possible sets of beat instants, and is guaranteed to find the best-scoring sequence up to any point. This allows the best global beat sequence to be found, even if it involves some locally-poor matching: Beats that fall during pauses or uninflected sustained notes are spaced evenly to "bridge" between the clearer beats on either edge.

4. CHROMA FEATURES

If the beat tracking can identify the same main pulse in different renditions of the same piece, then 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 [7, 2]. Chroma features record the intensity associated with each of the 12 semitones (e.g. piano keys) within one octave, but all octaves are folded together. The idea of calculating harmonic features over beat-length segments appears to have been developed several times; we first became aware of it in [10].

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 (instantaneous frequency) 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 [4, 1]. (This technique to remove nontonal components and improve frequency resolution beyond FFT bin level has similar motivation and impact to the sinusoid-modeling-based preprocessing proposed by [8], but we argue it is conceptually and computationally simpler.) We found that using only components up to 1 kHz in our chroma features worked best. An interesting aural rendition of the extracted information can be generated by using the 12 chroma bins to modulate Shepard tones (mixtures of harmonics in octave relationships only).

In an effort to avoid problems when the a piece is played slightly out of tune, the mapping of frequencies to chroma bins is adjusted for each piece by up to ± 0.5 semitones to make the single strongest frequency peak from a long analysis window line up exactly with a chroma bin center. The lower panes of figure 1 show chroma features before and after averaging into beat-length segments.

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 cannot expect these to occur in exactly the same places, in absolute or relative terms, in the two versions, for instance due to minor errors in the beat tracking, or as a result of variations in the structure (number of verses etc.). 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 feature matrices gave better 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

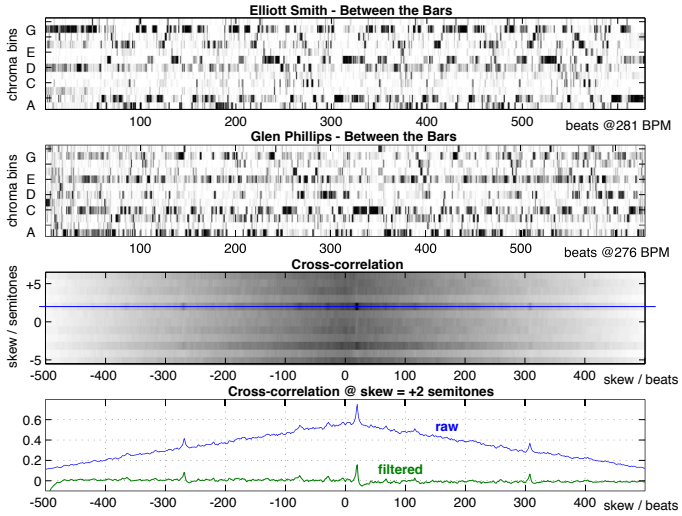

Fig. 2. Illustration of cover song matching. Top two panes are beat-chroma matrices for two versions of "Between the Bars". Third pane is two-dimensional cross correlation for all possible chroma rotations out to ± 500 beats skew. Bottom pane shows slice through the cross-correlation at $+2$ semitones (indicated by line), plus result of high-pass filtering to emphasize only peaks that result from precise beat alignment. Note the subpeaks at around 290 beats relative to the main peak, resulting from structural repetition in the song.

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 cross-correlations of large magnitudes, but that these large values occurred in narrow local maxima in the cross-correlations that fell off rapidly as the relative alignment changed 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' value measured between two pieces is simply the reciprocal 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) as reported in our experiments below.

Figure 2 shows these stages in the matching of the Elliott Smith track to a cover version recorded live by Glen Phillips. The top two panes shows the normalized, beat-synchronous IF-based chroma feature matrices for both tracks (which have tempos about 2% different). The third pane shows the raw cross-correlation for relative timings of $-500 \ldots 500$ beats, and all 12 possible relative chroma skews. The bottom panel shows the slice through this cross-correlation matrix for the most favorable relative tuning (Phillips transposed up 2 semitones) both before and after high-pass filtering; it is clear that filtering removes the triangular baseline correlation but preserves the sharp peak at around $+20$ beats indicating the match between the versions. (The Phillips version includes some audience noise at the start of the track, which causes this delay.) Note that the beat tracking in the live version is far from perfect, but the matching succeeds anyway.

6. EVALUATION

We have developed and evaluated this system on three databases: a small development set of contemporary pop music, a larger collection of pop music covers including live versions, and as part of the independent international 2006 MIREX evaluation.

6.1. Development set

We developed the system on set of 15 pairs of pop-music tracks that were versions of the same song by different artists. They were extracted from the uspop2002 dataset by making a list of all tracks from the total set of 8764 tracks that had the duplicate names (yielding about 600 tracks), 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 first list was compared to every track in the second list; the track that was most similar was reported as the cover version. 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.

6.2. Test set

To get a finer-grained sense of the performance, and to make a test that was independent of our development set, we identified a collection of 94 pairs from among our personal music collections. Many of these consisted of comparing live performances of pieces with the studio recordings by the same band; however, these live versions often showed significant stylistic variations. Table 1 lists the performance on this test set of several variants of our algorithm. We see that using the instantaneous-frequency-based chroma features is critical to the discriminability of the representation, but that using multiple tempo candidates was not successful, indicating that the "dominant" tempo is consistently chosen across different versions, for the most part. Biasing towards a lower tempo (120 bpm) has a slight negative impact on accuracy, but this may be acceptable given the smaller and more efficient representation.

6.3. MIREX Evaluation

MIREX (Music Information Retrieval Evaluation eXchange) is an international effort to develop formal, common evaluation standards

Table 1. Performance on 94-pair test set. Accuracy is the proportion of time that the correct cover version was returned as most similar to a query, and Mean Reciprocal Rank (MRR) reflects the rank of the first correct response in ordered returns. "Standard" uses IF-based chroma features and a single tempo biased towards 240 bpm. "P-Chroma" uses a chroma feature based on peak-picking in the spectrum without further spectral resolution improvement (similar to [8]). "Two-Tempos" employs two representations of each song are used, corresponding to the two most likely tempos for the piece, with the overall similarity based on the best pair. "120 BPM" biases the tempo detector towards a slower rate of 120 BPM, leading to smaller (but more blurred) representations.

System	Accuracy	MRR
Standard	59%	0.63
P-Chroma	29%	0.40
Two-Tempos	35%	0.50
120 BPM	51%	0.53

for music information retrieval. For the first time in 2006 there was an evaluation for cover song identification, which operated by having participants submit working code to the organizers at UIUC, who then independently applied the algorithms to their secret test set of 11 versions for each of 30 songs spanning "a variety of genres (e.g., classical, jazz, gospel, rock, folk-rock, etc.)". For each of the 330 test songs, the top 10 most similar tracks were examined. Evaluation was in terms of the total number of correct cover tracks returned (out of a theoretical maximum of 3300), the mean reciprocal rank (MRR) of the first correct cover version (between 0 and 1, with larger better) as well as some other metrics. Submissions included 4 systems specific to cover song detection, and 4 general music similarity systems (aimed at capturing a listener's judgment of similarity between pieces of music).

Our system (an earlier version of Standard from table 1) identified 761 cover songs and had an MRR of 0.49. The next best performing system identified 365 covers with an MRR of 0.22. Significance testing confirmed that our system was significantly superior to the others, with none of the remainder differing significantly from one another [5].

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 intractable; 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.

8. 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 authors and do not necessarily reflect the views of the NSF. Portions of this paper appeared in the system description abstracts for the MIREX-06 contest.

9. REFERENCES

[1] T. Abe and M. Honda. Sinusoidal model based on instantaneous frequency attractors. *IEEE Tr. Audio, Speech and Lang. Proc.*, 14(4):1292–1300, 2006.

[2] 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.

[3] M. Casey and M. Slaney. The importance of sequences in musical similarity. In *Proc. ICASSP-06*, pages V–5–8, Toulouse, 2006.

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

[5] J. S. Downie, K. West, E. Pampalk, and P. Lamere. Mirex2006 audio cover song evaluation, 2006. `http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song_Identification_Results`.

[6] D. Ellis. Beat tracking with dynamic programming. In *MIREX 2006 Audio Beat Tracking Contest system description*, 2006.

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

[8] E. Gómez. Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing, Special Cluster on Computation in Music*, 18(3):294–304, 2006.

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

[10] 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.

[11] M. Mueller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In *Proc. Int. Conf. on Music Info. Retr. ISMIR-05*, pages 288–295, London, 2005.

[12] W.-H. Tsai, H.-M. Yu, and H.-M. Wang. A query-by-example technique for retrieving cover versions of popular songs with similar melodies. In *Proc. Int. Conf. on Music Info. Retr. ISMIR-05*, pages 183–190, London, 2005.