COVER SONG IDENTIFICATION WITH 2D FOURIER TRANSFORM SEQUENCES

Prem Seetharaman

Northwestern University
Evanston, IL, USA
prem@u.northwestern.edu

Zafar Rafii

Gracenote, Inc.
Emeryville, CA, USA
zrafii@gracenote.com

ABSTRACT

We approach cover song identification using a novel time-series representation of audio based on the 2DFT. The audio is represented as a sequence of magnitude 2D Fourier Transforms (2DFT). This representation is robust to key changes, timbral changes, and small local tempo deviations. We look at cross-similarity between these time-series, and extract a distance measure that is invariant to music structure changes. Our approach is state-of-the-art on a recent cover song dataset, and expands on previous work using the 2DFT for music representation and work on live song recognition.

***Index Terms*—** Cover song identification, audio fingerprinting, Constant Q transform, 2D Fourier transform, adaptive thresholding

1. INTRODUCTION

Cover song identification is the act of identifying when two musical recordings are derived from the same music composition (e.g., Jimi Hendrix's *All Along The Watchtower* is a cover of the original by Bob Dylan). The cover of a song can be drastically different from the original recording. It can change key, tempo, instrumentation, musical structure or order, etc. Identifying cover songs automatically involves finding a representation of the audio that is robust to these transformations. In this work, we present an audio representation using sequences of 2D Fourier Transforms (2DFT) for the purpose of cover song identification.

[1] is the first work that tackles cover song identification, using an approach based on beat-synchronous chromagram representations of audio. The chromagrams of the covers and the originals are cross-correlated in pitch and time. When the cover and the original match, there will be a peak in the cross-correlation matrix. The beat-tracking makes their method resilient to tempo variation, and the cross-correlation in chroma and time makes it resilient to key changes and time skews. As the chromagram retains only pitch class information, it is somewhat resilient to instrumentation changes. Our work does not use the chromagram, but instead uses fingerprinting techniques to be robust to instrumentation changes. It does not do cross-correlation, and instead uses the properties of the 2DFT to be robust to time and pitch skews.

[2] further explores chromagram-based work for cover song identification, using techniques based on tonal subsequence alignment. [3] proposes a tempo-invariant approach by sampling the chromagram at different time rates, skews, and pitch shifts. In [4], songs are converted into a series of local mel-frequency cepstrum coefficients self-similarity patches (timbral shape sequences). These series of patches are aligned using local dynamic time warping for cover song identification. [5] uses a pretraining phase on the reference set, where characteristic phrases are extracted from each reference (music shapelets). These music shapelets are then used to identify cover songs. [6] proposes a fast and effective method for subsequence matching of time-series for cover song identification. For a more in-depth review of cover song identification, the reader is referred to [7]. In [8], [9] and [10], ensemble techniques are explored for cover song identification. This work proposes a new method that could be used in these ensemble approaches.

Some works address large-scale cover song recognition. In large-scale cover song recognition, a distance matrix must be computed between every pair of songs (on the order of millions of comparisons) efficiently. This requires features with low complexity and fast computation, which will act as a first stage to cull the possible reference set. On the reduced reference set, a more complex feature (such as the one in our work) can be applied. In [11], the problem is reduced to searching for chord sequences, represented as text. In [12], a hashing approach is proposed for the large-scale search.

The most closely related work to our own is [13], which is the first to propose the 2DFT for the purpose of cover song identification at a large scale. In their work, the 2DFT is computed on overlapping segments of the beat synchronous chromagram. These 2DFT chromagrams are then summarized into a single image patch. Cover song identification is done by comparing these image patches. [14] proposes using the 2DFT for identifying repeating sections in music, and uncovering musical structure in audio. Similarly, [15] looks at the cover song identification problem through the lens of music structure similarity.

Audio fingerprinting is the act of identifying exact renditions of references in complex audio scenes [16, 17, 18]. In between the cover song identification problem and the audio fingerprinting problem is live song recognition, which is explored in [19] and [20]. In live song recognition, the task is to recognize a live song (e.g., at a concert) performed by the original artist. In this task, there may be key variation, slight tempo variation, and musical structure changes (e.g., for an artist that is known to improvise). The signal may also be degraded (e.g., crowd noise, bad microphone), as in the regular audio fingerprinting task. In this work, we extract features from the fingerprint proposed in [19]. We find that deriving representations from this fingerprint is important for our algorithm. It may be a useful direction for other cover song identification algorithms as well.

2. PROPOSED METHOD

2.1. The 2D Fourier Transform on Musical Signals

The 2DFT, like the 1DFT in music analysis, is a popular technique in digital image processing, and is used for image denoising and compression, among other things [21]. The 2DFT breaks down images into sums of sinusoidal grids at different periods and orientations, represented by points in the 2DFT. On a spectrogram with a log-frequency scale, points along the y-axis of the transform represent periodicities along the frequency domain of the spectrogram, and points along the x-axis represent periodicities along the time domain of the spectrogram. The information about the exact position of the sinusoidal grids in the original image is kept entirely in the phase. A useful representation of musical audio is the Constant Q Transform (CQT) [22, 23]. The CQT is a transform with a logarithmic frequency resolution, with spacings between frequencies mirroring the human auditory system, and the Western musical scale. A linear shift in frequency in the CQT corresponds to a pitch shift in the music. By taking the magnitude of the 2DFT on the CQT, we obtain a key-invariant representation of the audio.

2.2. Fingerprint with CQT and Adaptive Thresholding

The three steps of our system can be seen in Figure 1. In the first step, the entire time-domain audio signal is converted into a CQT, with frequencies corresponding to the musical scale between C3 (130.81 Hz) and C7 (2093 Hz), with a frequency resolution of 2 frequency bins per semitone and a time resolution of 10 frames per second. The CQT is an important step because a cover song in a different key will correspond to a linear shift in the CQT. The CQT of *Can't Help Falling in Love* (Elvis Presley) is shown in the top image in Figure 1.

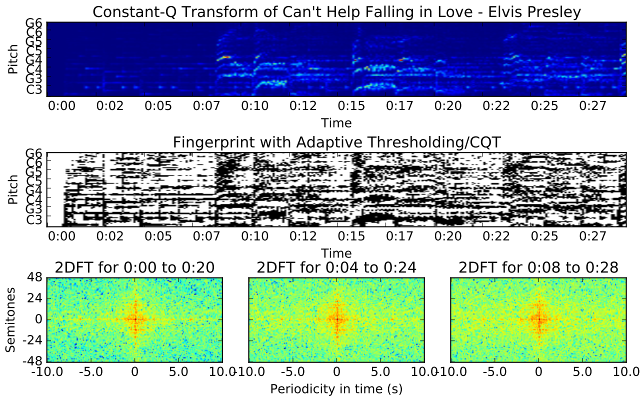

Fig. 1. Overview of our system. First, we take the CQT of the audio. Next, we fingerprint the CQT using an adaptive thresholding method [19]. Finally, we take 2DFTs over sliding windows on the fingerprint, producing a 2DFT sequence, shown here with power compression for visualization purposes.

In the second step, we use the adaptive thresholding technique presented in [19] to binarize the CQT. The technique slides a patch of a specified size along the CQT. Inside the patch, values are set to 1 if they are above the median of the patch (e.g., the local median), and 0 otherwise. This has the effect of scrubbing timbral information as well as balancing the sound levels of different sources in the mixture. Thus, if a source (e.g., a distorted guitar) dominates the mixture in the original recording, but is not present in the cover (e.g., an acoustic cover with no distorted guitar), the fingerprint will be robust to these drastic changes in timbre and energy. In [13], they found that pre-processing the chromagram to increase high-energy bins relative to low-energy bins improved performance. They argue that it accentuates the main patterns in the signal. We find similar results, and argue that this fingerprint would also be useful for other problems. The fingerprint is shown in the middle image in Figure 1.

Finally, we take overlapping 20 second windows of the fingerprint, and compute the 2DFT of each. For each 2DFT, we take the magnitude, discarding the phase, and add a small amount of Gaussian blurring ($\sigma = .375$). We then hop forward by 4 seconds, and take another 2DFT. We continue this until we've reached the end of the fingerprint. Each 20 second window has a dimensionality of 96 by 200, and each 2DFT has the same dimensionality. The sequence of 2DFTs, shown in Figure 1, is used to represent the audio and the representations are compared for cover song identification. It is key-invariant due to the properties of the magnitude 2DFT. The values along the vertical axis of the 2DFT capture characteristic periodic patterns in frequency, and values along the horizontal axis capture characteristic periodic patterns in time. These are useful for cover song identification, as these patterns will often be retained in a cover song.

This representation is key-invariant, retains spectral and temporal structural information, and scrubs loudness and timbral information. All of these may change considerably between a cover song and the reference recording. Small tempo deviations between the cover and the reference can be taken

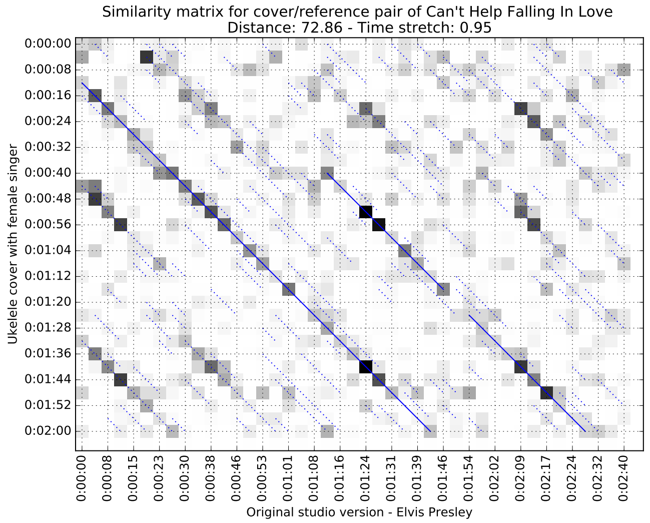

Fig. 2. Similarity matrix constructed using Euclidean distance between 2DFT sequences on a ukulele cover of *Can't Help Falling In Love* and the original studio version. The subsequence matches are indicated by the solid blue lines.

care of by the Gaussian blurring we applied to each 2DFT, which blurs the exact location of temporal periodicities in the audio. However, large tempo deviations need to be dealt with more directly. To do this, we simply change the sampling rate on the reference recordings, and repeat our fingerprinting procedure on the recordings with different sampling rates. This causes a pitch shift in the CQT, due to the similarity theorem for Fourier transforms. However, the key-invariance of the magnitude 2DFT means this pitch shift does not matter. For each reference recording, we fingerprint between 0.5 and 2 times the original sample rate at 0.05 intervals. These correspond to half speed and double speed, respectively. This results in 30 2DFT sequences for each reference. For a query recording, we fingerprint only at the original sample rate.

2.3. Search

In cover song identification, we are given a query song and compare it against a database of reference songs using a distance measure. The reference songs are then ranked in ascending distance. A good distance measure will have the correct reference song highly ranked for a given query. We compute this distance measure from a similarity matrix between the query and a candidate reference song. The query and the reference are represented as a sequence of 2DFTs. We compute the Euclidean distance between every pair of 2DFTs and store them in a similarity matrix (SM). We save the energy of the unnormalized SM E. We then normalize the SM by its maximum value.

We then post-process the SM by convolving it with a checkerboard kernel, as in [24] and [17]:

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Positive elements in the convolved SM correspond to diagonal matches and negative elements correspond to non-diagonal matches. We set all negative elements to 0, resulting in a SM with just the diagonals, as shown in Figure 2. The checkerboard kernel leaves sequential matches in, and takes spurious matches between single 2DFT patches between the cover and reference out. A diagonal of length 2 in the processed similarity matrix corresponds to a 24 second match between the query and the reference.

To compute the distance between a query and reference, we extract diagonals from the SM, and record their sum and length. To do this, we iterate through each diagonal in the SM and segment it into non-zero sequences. These sequences are matches beginning and ending at offsets in the cover and the reference. For each sequence, we record its sum w and its length l. The diagonal is scored by the product of these two: wl. We then sort all of the diagonals in the SM by decreasing score, and take the sum of the top three diagonals. These top three diagonals are the dominant subsequence matches between the query and reference. Finally, we divide the energy of the unnormalized SM, E, by the sum of the top 3 diagonals to obtain a distance measure:

$$d(q,r) = \frac{E}{\sum_{i=1}^{3} w_i l_i}, w_i, l_i \in diags(SM_{q,r})$$

where i is the index of the sorted diagonals list extracted from $SM_{q,r}$, the similarity matrix between the query (q) and the reference (r), via the function $diags$, as seen in Figure 2. The number of diagonals to sum is a free parameter. 3 was determined experimentally on a development dataset.

This approach only cares about sequential matches happening somewhere in the SM, and not where they are. This is similar to the subsequence matching for cover song identification, as in [5] and [6]. As a result, the distance measure is invariant to music structure changes (e.g., a skipped bridge or verse, or added intro and outro) which a straightforward dynamic time warping (DTW) approach would be sensitive to. In addition, this approach is much faster than DTW. However, if the tempo of the cover is significantly different than the tempo of the original, no strong diagonals would be present and this technique for subsequence matching will fail. We account for this by computing the SM across all the resampled versions of the reference audio. The final distance between a query q and reference r is then:

$$distance(q,r) = \min_{x \in \{.5,.55,\ldots,2\}} d(q, r_x)$$

where r_x indicates the reference resampled by a factor x.

The SM for a cover and its correct reference is shown in Figure 2. In this cover, the verse starts at 16 seconds, and in

Algorithm	MAP	P@10	MR1
DTW [5]	0.425	0.114	11.69
Silva et al. [5]	0.478	0.126	8.49
Serra et al. [2]	0.525	0.132	9.43
Silva et al. [6]	0.591	0.140	**7.91**
Proposed (on CQT)	0.521	0.122	9.75
Proposed (on fingerprint [19])	**0.648**	**0.145**	8.27

Table 1. Mean average precision (MAP), precision at 10 (P@10), and mean rank of first correctly identified cover (MR1) for the *YouTube Covers* dataset for existing approaches and our approach. Since there are two possible correct references for each query, P@10 has a maximum value of 0.2.

the reference, the verse starts at 8 seconds. In the SM, a clear diagonal can be seen indicating sequential matches. In the cover, the singer also adds a longer intro section than in the original song. In addition, the instrumentation has changed considerably. The original song has drums, guitar, piano, backing vocals, and a male lead singer. The cover song has a ukulele playing the chords, and a female singer on the melody, which is now an octave higher than the original, and is slightly slower. There are also structural changes. The best alignment was found at a resampling factor of 0.95. The SM in Figure 2 shows that the proposed representation and proposed distance measure is robust to these changes.

3. EVALUATION

We test our approach on the *Youtube Covers* dataset, which was also used in [5] and [6]. The dataset consists of 50 compositions, with 7 recordings of each. Of these recordings, 1 is the original studio version, 1 is a live version performed by the original artist, and 5 are covers drawn from YouTube. The reference set consists of the original studio versions and the live versions for each composition. The query set consists of the 5 covers for each composition. In all, the size of the reference set is 100, and the size of the query set is 250. In the experiment, we take each query and compare to every reference, getting a distance for each query/reference pair. We then rank the references for each query, and compute the mean average precision, the precision at 10, and the mean rank of the first correctly identified cover, as in [5]. These evaluation measures are the same as the ones used in the MIREX cover song identification task[1].

We developed our algorithm using the *covers80* dataset [1]. This dataset consists of 80 compositions, with 2 recordings of each, one of which is a cover, and the other the original. Once we discovered good parameters for our approach on this dataset (44/80 with adaptive thresholding, and 36/80 without), we tested on the *Youtube Covers* dataset with no algorithm or parameter changes (2DFT window size of 20 seconds, hop size of 4 seconds, $\sigma = .375$). We compare our approach with recent cover song identification approaches: music shapelets [5], similarity matrix profiles [6], chroma binary similarity with local alignment [2], and dynamic time warping of Chroma Energy distribution Normalized Statistics feature vectors [25] [5]. We also compare two variants of our approach: one with adaptive thresholding and one without.

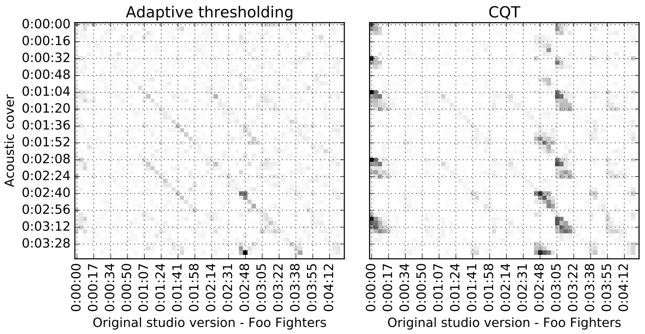

Fig. 3. Similarity matrices for a cover/reference pair with and without the adaptive thresholding step. The diagonals are more visible when the CQT is adaptively thresholded. The presence of a loud heavily distorted guitar in the original causes a poor match with the acoustic cover.

Our results can be seen in Table 1. In mean average precision and precision at 10, the proposed approach surpasses current state-of-the-art methods on this dataset. The proposed approach finds 164 covers out of 250 correctly at top one. The impact of the adaptive thresholding step is significant, causing a jump in mean average precision of 0.127, and going from the worst performing approach to the best performing approach in terms of P@10. The adaptive thresholding step emphasizes structure over timbre and energy leading to a more robust similarity measure, as seen in Figure 3. Our approach does well on covers where a critical mass of periodic and harmonic structural information is retained (e.g. chord progression/rhythm/melody), but fails when the cover retains a small portion of the original (e.g. just the melody in multiple jazz renditions of My Favorite Things).

4. CONCLUSION

We have presented an approach for cover song identification that uses a time-series representation of audio based on the magnitude 2DFT. The audio is represented as a sequence of magnitude 2D Fourier transforms. The representation is robust to key changes, timbral changes, and small local tempo deviations. We look at similarity between these time-series representations, and extract a distance measure that is invariant to structural changes. We note that the adaptive thresholding is an important pre-processing step. Our approach is state-of-the-art on a cover song dataset, and expands on previous work using the 2DFT for music representation.

[1] http://www.music-ir.org/mirex/

5. REFERENCES

[1] D. P. Ellis and G. E. Poliner, "Identifying cover songs with chroma features and dynamic programming beat tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.

[2] J. Serra, E. Gómez, P. Herrera, and X. Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, 2008.

[3] J. H. Jensen, M. G. Christensen, D. P. Ellis, and S. H. Jensen, "A tempo-insensitive distance measure for cover song identification based on chroma features," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

[4] C. Tralie and P. Bendich, "Cover song identification with timbral shape sequences," in *International Society for Music Information Retrieval Conference*, 2015.

[5] D. F. Silva, V. M. A. d. Souza, G. E. d. A. P. A. Batista, *et al.*, "Music shapelets for fast cover song recognition," in *International Society for Music Information Retrieval Conference*, 2015.

[6] D. F. Silva, C.-C. M. Yeh, G. E. Batista, and E. Keogh, "SIMPle: Assessing music similarity using subsequences joins," in *International Society for Music Information Retrieval Conference*, 2016.

[7] J. Serra, E. Gómez, and P. Herrera, "Audio cover song identification and similarity: background, approaches, evaluation, and beyond," in *Advances in Music Information Retrieval*, Springer, 2010.

[8] S. Ravuri and D. P. Ellis, "Cover song detection: from high scores to general classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010.

[9] J. Osmalsky, J.-J. Embrechts, P. Foster, and S. Dixon, "Combining features for cover song identification," in *International Society for Music Information Retrieval Conference*, 2015.

[10] J. Osmalsky, M. Van Droogenbroeck, and J.-J. Embrechts, "Enhancing cover song identification with hierarchical rank aggregation," in *International for Music Information Retrieval Conference*, 2016.

[11] M. Khadkevich and M. Omologo, "Large-scale cover song identification using chord profiles," in *International Society for Music Information Retrieval Conference*, 2013.

[12] T. Bertin-Mahieux and D. P. Ellis, "Large-scale cover song recognition using hashed chroma landmarks," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2011.

[13] T. Bertin-Mahieux and D. P. Ellis, "Large-scale cover song recognition using the 2D Fourier transform magnitude," in *International Society for Music Information Retrieval Conference*, 2012.

[14] O. Nieto and J. P. Bello, "Music segment similarity using 2D-Fourier magnitude coefficients," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.

[15] J. P. Bello, "Measuring structural similarity in music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, 2011.

[16] A. L.-C. Wang, "An industrial strength audio search algorithm," in *International Society for Music Information Retrieval Conference*, 2003.

[17] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *International Society for Music Information Retrieval Conference*, vol. 2002, 2002.

[18] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of audio fingerprinting," *Journal of VLSI Signal Processing Systems*, vol. 41, 2015.

[19] Z. Rafii, B. Coover, and J. Han, "An audio fingerprinting system for live version identification using image processing techniques," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.

[20] T. Tsai, T. Prätzlich, and M. Muller, "Known-artist live song id: a hashprint approach," in *International Society for Music Information Retrieval Conference*, 2016.

[21] R. C. Gonzalez and R. E. Woods, "Digital image processing," *Prentice Hall*, 2008.

[22] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *Journal of the Acoustical Society of America*, vol. 89, 1991.

[23] J. C. Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America*, vol. 92, 1992.

[24] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *IEEE International Conference on Multimedia and Expo*, 2000.

[25] M. Müller, F. Kurth, and M. Clausen, "Audio matching via chroma-based statistical features," in *International for Music Information Retrieval Conference*, 2005.