LARGE-SCALE COVER SONG IDENTIFICATION USING CHORD PROFILES

Maksim Khadkevich
Fondazione Bruno Kessler-irst,
via Sommarive 18, Povo 38050, Italy
hadkevich@gmail.com

Maurizio Omologo
Fondazione Bruno Kessler-irst,
via Sommarive 18, Povo 38050, Italy
omologo@fbk.eu

ABSTRACT

This paper focuses on cover song identification among datasets potentially containing millions of songs. A compact representation of music contents plays an important role in large-scale analysis and retrieval. The proposed approach is based on high-level summarization of musical songs using chord profiles. Search is performed in two steps. In the first step, the Locality Sensitive Hashing (LHS) method is used to retrieve songs with similar chord profiles. On the resulting list of songs a second processing step is applied to progressively refine the ranking. Experiments conducted on both the Million Song Dataset (MSD) and a subset of the Second Hand Songs (SHS) dataset showed the effectiveness of the proposed solution, which provides state-of-the-art results.

1. INTRODUCTION

Recent advances in digital media have allowed for extensive wide-spread growth of musical collections. We entered an era of content-based multimedia search engines, boosting the demand for advanced audio analysis tools and applications. Cover song identification based on the analysis of audio contents is a challenging problem, caused by the fact that different renditions of a song can differ in tempo, instrumentation, key, or genre. Given this, audio spectral contents of two covers can vary significantly from one another. During the last decade, the problem of cover song identification has been of a great interest to scientists working in the Music Information Retrieval (MIR) research area [1–3]. Identifying cover songs can help detect copyright infringements and correctly handle music license management.

In this paper, we propose an approach to large-scale cover song identification using chord progressions and chord profiles. A chord progression is extracted from audio or from chroma features provided with the Million Song Dataset (MSD) [4]. For the most part, the approaches proposed in the literature are based on the alignment of local features, which is typically performed by Dynamic Time Warping (DTW) [1], or string alignment [5], and require a significant amount of computational resources. To build a system that can operate on a large scale, we propose the use of chord profiles for indexing and fast retrieval. A chord profile of a song is a compact representation that summarizes the rate of occurrence of each chord. To solve the problem of cover song identification, we propose the use of a well-established approach for fast retrieval in multi-dimensional spaces, which is Locality-Sensitive Hashing (LHS). A more accurate comparison is then performed between the top k results, based on the alignment of chord progressions.

1.1 Previous work

Most of the cover song identification systems reported in the literature work on small datasets (up to several thousands of songs) and involve pair-wise comparison, when a query song is matched against all the songs in the database [1, 2]. This makes them impractical for large collections, containing millions of items. A good overview of existing approaches is given in [3].

Recent works on large-scale datasets are based on indexing and fast retrieval. So far, several indexing schemas have been introduced which allow for fast searching over large databases. Bertin-Mahieux and Ellis [6] proposed using "chroma jump codes". They extract beat-synchronized chromagram, discover chroma bins with high energy, and construct "jump codes" between such pairs. For retrieval, "jump codes" extracted from a queried song are searched in the database and results are ranked according to the number of matching pairs. In the approach of Bertin-Mahieux and Ellis [7], 2-D Fast Fourier Transform (FFT) of overlapping chromagram segments corresponding to 75 beats are averaged with subsequent application of Principal Component Analysis (PCA). Distances between vectors of PCA components are used to generate ranked lists.

Several approaches have been proposed based on chord sequence alignment. Lee [8] used chord sequences for cover song identification. His approach is based on the extraction of chords by means of a HMM-based recognizer trained on data generated from MIDI. The songs are ranked according to a DTW-based pairwise similarity on key-transposed sequences. Bello [5] extended this approach, systematically evaluating key shifting, the cost of gap insertions and chord swaps in string alignment. Martin et al. [9] adapted tools from computational biology to align

chord sequences. However, their system could not perform well on a subset of the MSD dataset, probably due to some simplifications in the chord representations; instead of taking into account chord types, they extract only chord roots. These works show that chord progressions can be successfully used as high-level features for the problem of cover song identification.

The previous works listed above operate either on frame-level features such as chroma [1, 2] or chords [5, 8], or by extracting very compact descriptors of the whole song [6, 7]. The former approaches are more accurate but do not scale to large databases, since brute-force sequence matching is computationally expensive. The latter approaches allow for scalability, but usually are limited in terms of performance. In this paper, the problem of cover song identification is approached trying to find a compromise between fast retrieval and exhaustive matching.

1.2 Organization of the paper

The datasets and the evaluation methodology are presented in Section 2. Section 3 briefly describes the proposed system architecture, including front-end processing and indexing schema. Section 4 is devoted to experimental results. Some conclusions and future work are finally presented in Section 5.

2. DATASET AND EVALUATION METHODOLOGY

The lack of large datasets for cover song identification is a problem, which has always been an obstacle to develop systems suitable for commercial use, i.e., able to run queries on a large database comprising millions of songs. The Million Songs Dataset (MSD) is the first attempt to create a large dataset for different MIR tasks. The MSD dataset contains features and meta-data for one million songs. All the features are extracted using the EchoNest [1] API. Unfortunately, the distribution of the MSD dataset does not contain chords. We used chroma segments and beats to estimate chord sequences from MSD features using the template-matching approach [10].

The Second Hand Songs (SHS) [2] dataset consists of meta-data for around 250000 songs. For about 50000 songs a link to YouTube [3] video is available.

An intersection of the SHS and the MSD datasets was used to build the SHS-MSD dataset. 18196 songs from the SHS dataset are chosen to form the SHS-MSD dataset. It was proposed to divide the SHS-MSD dataset into training (12960 tracks) and test (5854 tracks) parts [6]. They will be referred to as SHS-training and SHS-test, respectively.

Due to the fact that data provided with the MSD dataset does not contain waveforms, and extracted chroma features are probably not the best solution to produce chord recognition rate comparable to state-of-the-art systems, we collected our own dataset. It will be referred to as SHS-WAV dataset. We used the SHS website to download 24282 videos, from which audio tracks were extracted. All the tracks from the SHS-WAV dataset are divided in 5650 cover groups. The largest group contains 128 covers, which is "Summertime" by George Gershwin. To our knowledge, this is the largest collection of audio waveforms suitable for the evaluation of large-scale cover song identification systems. The list of the songs for both the datasets and the corresponding extracted chords are publicly available [4].

We follow the same evaluation methodology as proposed in [7]. As with many other information retrieval systems that produce a list of ranked items as a result, we adopt Mean Average Precision (MAP) as the main evaluation metric. We also report Average Rank (AR), but this metric is less informative and can be misleading when used alone. AR is mostly influenced by the most difficult covers, while differences in the top of the rank are neglected [7]. We also present distribution statistics of the ranked songs for the whole MSD dataset with a particular emphasis on the top ten results.

3. PROPOSED SYSTEM ARCHITECTURE

Chords and melody are considered to be essential characteristics of a song. They are the two attributes that describe tonal and harmonic properties of a musical piece and allow us to identify a song among many others, regardless of tempo, instrumentation, or genre. The proposed cover song identification system is based on the use of chord progressions and chord profiles. Chords are considered to be a high-level descriptor. Despite possible local changes in the different renditions of a song (e.g., a major chord is replaced by a minor one) the general characteristics of the chord progressions embedded in it are typically preserved from one cover to another. Therefore, in this paper we explore different ways of exploiting chord progressions and chord profiles to build a robust and large-scale cover song identification system. The block diagram of the proposed system presented in Figure 1. It comprises the following structural components: high-level feature extraction, indexing and retrieval.

3.1 Chord progression extraction

Chord progressions and chord profiles are the two high-level features used in the proposed system. The extraction of beat-synchronous chord progression is the first step. Two different datasets are used to evaluate the proposed approach, one containing audio waveforms and the other containing only features and metadata. Correspondingly, two different chord extraction techniques are adopted as discussed below.

In the case of SHS-WAV dataset, for the extraction of beats and chords from audio waveform we use Vamp [5] plugins *Chordino* and *BarBeatTracker* that show state-of-the-

[1] http://www.echonest.com
[2] http://www.secondhandsongs.com/
[3] http://www.youtube.com/
[4] https://github.com/FBK-SHINE/CoverSongData
[5] http://www.vamp.org

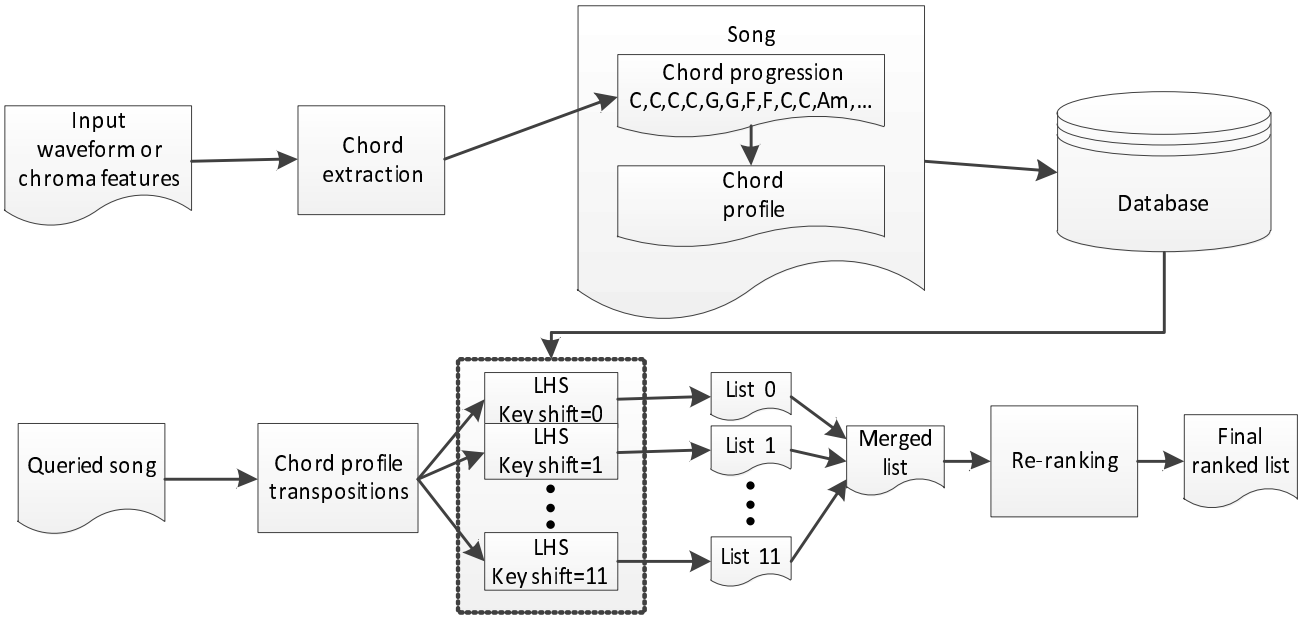


Figure 1: Block diagram of the proposed system

art results. The beat structure is used to obtain a tempo-independent sequence of chords. Once chords and beat structure are extracted, the chords are split into beat segments so that each beat segment contains one chord. If a chord transition occurs inside a beat segment, the chord segment that has the longest intersection with the current beat segment is used to derive the chord label. The chord dictionary comprises two chord types, which are *major* and *minor*.

To derive chord progressions from the MSD dataset, beat-synchronous chroma features are first extracted. We follow the approach of Bertin-Mahieux and Ellis [6], where chroma vectors are averaged across beat segments. In the second step, we apply the template matching technique proposed in [10]. Template matching for chord recognition is based on the idea of introducing a set of templates for each chord type. The template configurations are derived heuristically. We define a binary mask as a 12-dimensional chord template in which the pitch classes that correspond to constituent notes of the given chord are set to one, while the other components are set to zero. A binary template T is defined as

$$T = [Z_C, Z_{C\sharp}, Z_D, Z_{D\sharp}, Z_E, ..., Z_{A\sharp}, Z_B] \qquad (1)$$

where Z_p denotes the mask value that corresponds to the pitch class p. For example, binary masks for C major and D minor chords would take the following form:

```
T(C:maj)=[1,0,0,0,1,0,0,1,0,0,0,0]
T(D:min)=[0,0,1,0,0,1,0,0,0,1,0,0]
```

The template that produces the highest cosine similarity between chroma vectors is used to generate a chord label for the given beat segment. The cosine similarity between vectors a and b is defined as

$$S_c(a,b) = \frac{a \cdot b}{\|a\| \, \|b\|} \qquad (2)$$

where $\|\cdot\|$ denotes Euclidean distance.

The resulting sequence of beat-aligned chords is subsequently used as a compact representation of the harmonic structure of the song. However, the comparison of chord progressions is usually done by sequence alignment, which is a computationally expensive operation and cannot be used on a large scale. Therefore, we propose to further compress the extracted high-level features. This can be done by discarding the temporal information and compacting all the related contents in a chord profile. A chord profile is a 24-dimensional vector, in which each dimension corresponds to the rate of occurrence of a chord. Let n_i be number of beat segments containing chord i, where $i \in 1..24$. Then, the i-th component of the chord profile vector c is calculated as $c_i = \frac{n_i}{N}$, where N is the total number of beat segments.

Chord profiles and chord progressions extracted for each song of a given dataset are stored in a database. In the retrieval stage, high-level features extracted from a queried song are used to derive a ranked list of possible covers from that database, as described in the following section.

3.2 Retrieval

The proposed cover song identification system relies on a two-step retrieval schema. Given a large database of chord profiles and a queried song, we address the problem of finding the nearest neighbors. Finding the nearest neighbors of an element in large databases is a well-known problem addressed in many areas of information retrieval. For low-dimensional data, nearest neighbor search can be performed by partitioning the search space using, for example, k-d trees as mentioned in [11]. Data with high number of dimensions cause the so-called "curse of dimensionality", when the distance between neighboring points tend to be large [12]. Locality Sensitive Hashing is a probabilistic approach to reduce dimensionality by hashing features

so that items that are close to each other fall in the same bucket with high probability. Casey and Slaney [13] used LHS for fast shingle retrieval from a comparatively large database. Casey et al. [14] extended their work, performing analysis of optimal parameters and giving examples of LHS application for different MIR tasks. Yu et al. [15] proposed an adapted two-level LHS scheme to tackle the problem of retrieving multi-variant audio tracks. In their work, a study on trade-off between identification accuracy and efficiency is presented.

For small datasets containing several thousands of songs, a straightforward approach suggests computing the distances between a queried song and all the items in the dataset. When working with larger databases, containing several millions of songs, a significant increase in speed can be achieved by using LHS. Following the approach of Casey and Slaney [13], in the first step we use LHS to retrieve the nearest neighbours. L_1 distance is adopted as distance metric. Given two chord profiles a and b, the distance between them is defined as

$$||a - b||_1 = \sum_{i=1}^{24} |a_i - b_i| \quad (3)$$

Note that L_2 distance was used in our early experiments, with definitely worse results. However, this topic should be matter of further investigation.

Due to the fact that a cover of a given song can be performed in a different key, we should introduce a mechanism to make the distance between chord profiles key-invariant. This can be achieved by performing a circular permutation of a queried song chord profile 12 times, taking into account all the possible key transpositions. For each transposition, we retrieve a list of candidates. In the final part of the first stage, all the lists are merged and all the retrieved songs are ranked according to (3).

In the second step, the top k results are re-ranked by computing edit (or Levenshtein) distances between chord progressions. The edit distance is the number of insertions, deletions and substitutions to transform one sequence into another. Thus, a more accurate matching is performed, which takes temporal information into account. Time complexity of computing edit distance is $O(nm)$, where n and m are chord progression lengths of the songs under comparison. Choosing k depends on the balance between speed and precision. In our experimental setup, we set $k = 2000$, which means that the top 2000 results from the merged rank list obtained in the first stage are re-ranked according to edit distance between chord progressions. In this way, the output is refined taking into account temporal alignment between a queried song and the top k items from the merged ranked list.

4. EXPERIMENTAL RESULTS

4.1 Evaluating MSD features

The errors produced by the chord extraction propagate through successive processing steps and eventually have an impact on cover song identification performance. As a result, it was important to understand if the features provided by the EchoNest API could be used for an accurate chord estimation. As opposed to frame-based [10] or beat-synchronous [1] approaches to extracting chroma features, the EchoNest API has its own segmentation algorithm which is independent of beat positions. The chroma vectors are averaged across segments that can contain several beats. On the other hand, some beats can contain several such segments. Another limitation of these chroma features is the absence of bass information. It has been shown that using a lower frequency content, as extra-feature, leads to a significant improvement in performance [16, 17]. Given this, the chroma vectors delivered with EchoNest API might not be the best features for automatic chord extraction.

In fact, in our preliminary experiments we applied the chroma feature extraction available with the EchoNest API on the MIREX 2011 corpus to evaluate chord recognition performance. The corpus consists of 220 songs of Beatles, Queen, Zweieck and Carol King. The template-based approach described in Section 3.1 was used to generate chord labels.

The experiment led to a chord recognition rate of 55.7%, compared to 77.8% obtained using the frame-level template matching based on time-frequency reassigned chroma features proposed in [18]. In the latter case, a 24-dimensional chroma vector was used, where the first and the second 12 dimensions corresponded to bass and treble contents, respectively. This suggests that using alternative chroma feature sets to represent a song can lead to an improvement in chord recognition rate, and as a consequence in cover song identification performance.

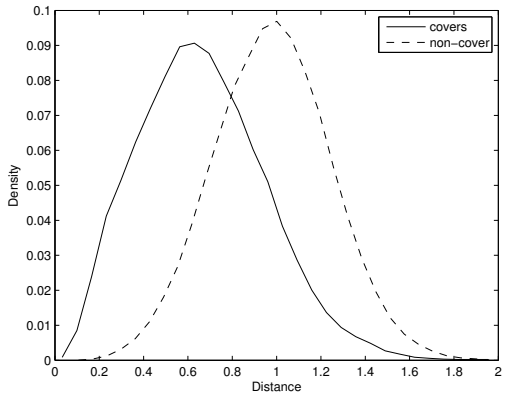
In the following experiments, we compare the performance of the proposed system when applied to the MSD dataset and to the collected SHS-WAV dataset.

4.2 Chroma profiles

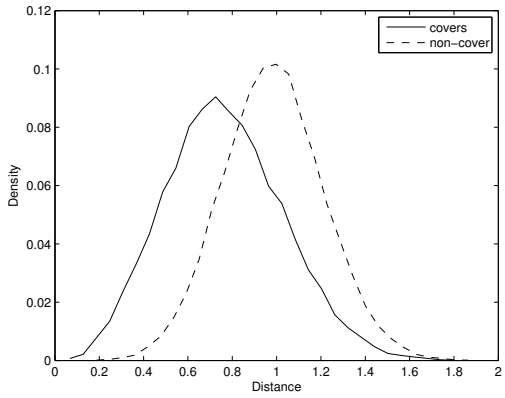
The first set of experiments aimed to collect statistics of chord profile distances between covers and non-covers. The results for the SHS-training and the SHS-WAV datasets are presented in Figure 2.

To generate statistics on covers, we calculated distances between chord profiles corresponding to all cover pairs in the given dataset. As for non-covers, for each song we randomly choose a non-cover song. In order to take into account possible key shifts, the distance between two chord profiles is defined as the minimum distance among those obtained for all the 12 possible circular permutations.

As shown in Figure 2a, for SHS-WAV dataset, Gaussian-like distributions are obtained with mean values and standard deviations of (0.66, 0.28) and (0.97, 0.26), for cover and non-cover pairs, respectively. A similar behavior is obtained with SHS-training dataset, for which mean values and standard deviations are (0.76 0.26) and (0.98, 0.22), respectively.



(a) SHS-WAV dataset



(b) SHS-training dataset

Figure 2: Probability density function of distance between cover and non-cover pairs for SHS-WAV and SHS-training datasets

4.3 Cover song identification results

Table 1 presents the results on the three datasets. We compare our system with "chroma jumps" [6] and 2D-FFT [7]. On all the datasets, the proposed approach showed the best results. The results reported for "SHS-training" were obtained when using the training subset of MSD-SHS dataset (12960 tracks). The results reported for "SHS-WAV" were obtained when using waveforms extracted from YouTube videos (24282 tracks). The "Full MSD" labeled columns refer to the most important experimental setup, which indicates the scalability of the approach and shows the performance of the proposed features on the full MSD dataset. Covers from SHS-test dataset (5854 tracks) were used for querying.

It is interesting to see that the proposed system performed significantly better on SHS-WAV dataset, if compared to SHS-training dataset. This fact confirms that the chroma features provided with MSD dataset are not the best solution for chord extraction. It is likely that the very low performance at this moment obtained on the full MSD dataset can be significantly improved by processing the original waveforms.

Figures 3 and 4 show the distribution of ranks for the experiments on the full MSD dataset. 10% of the covers were ranked in the top 1000, and around 25% of them in the top 10000. For 255 queries, a cover was placed in the first position, while 716 covers appeared in the top-ten ranked list.

Table 1: Experimental results on three datasets

System	Average Rank	MAP
SHS-training		
proposed approach	958.2	0.10753
2DFTM (200 PC) [7]	3,005.1	0.09475
2DFTM (50 PC) [7]	2,939.8	0.07759
SHS-WAV		
proposed approach	1,378.4	0.2062
Full MSD		
proposed approach	114,951	0.03709
2DFTM (200 PC) [7]	180,304	0.02954
2DFTM (50 PC) [7]	173,117	0.01999
jcodes 2 [6]	308,370	0.00213

Table 2: Runtime and memory footprints

Dataset	Size (songs)	sec/query	memory
SHS-WAV	24282	1.46	346M
SHS-training	12960	1.08	198M
full MSD	1 million	7.56	3690M

Average querying runtime for each dataset is presented in Table 2. All the experiments were conducted on a modern laptop with 8GB of RAM installed and CPU Intel i7-2760QM running at 2.4 GHz. Due to the compactness of the chord progressions and chord profiles, it is possible to put all the features extracted from full MSD dataset into a hash table and store it in RAM. Memory footprint was around 3.5 GB. In terms of speed, average runtime per query without any parallelization appeared to be 7.56 seconds when searching in the full MSD dataset.

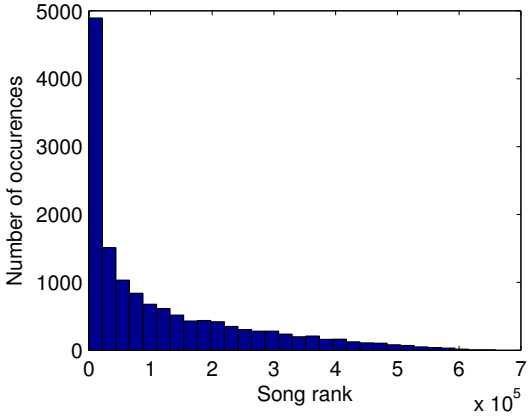


Figure 3: Rank distribution for the results on full MSD dataset

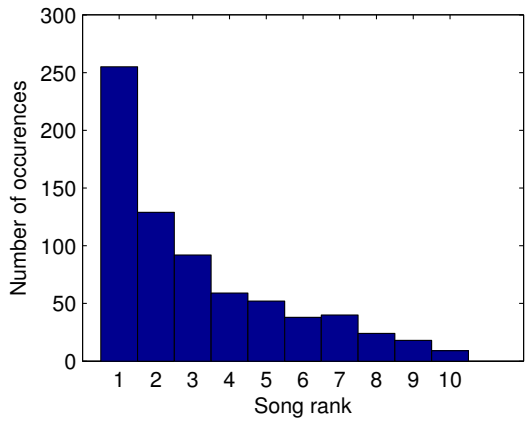


Figure 4: Top ten rank distribution for the results on full MSD dataset

5. CONCLUSION

In this paper, we proposed a new system for scalable cover song identification. The experimental results showed that chord profiles can be used as an extremely compact high-level feature that summarizes harmonic properties of a song. The proposed two-step approach improves the systems on which we compared performance and suggests room for an improvement. More sophisticated distances than Levenshtein could be used for sequence alignment, such as NeedlemanWunsch or Smith-Waterman, which were used in [5] and [2], respectively. More efficient ways of introducing key-invariance can be investigated. Instead of using 12 circular permutation to make a query, an alternative feature vector representation can be utilized. Experimental results obtained on different datasets showed that switching from precomputed feature data provided by the EchoNest API to state-of-the-art high-level feature extractors may further improve the performance.

6. REFERENCES

[1] D. P. W. Ellis and G. E. Poliner, "Identifying 'cover songs' with chroma features and dynamic programming beat tracking," in *Proc. ICASSP*, vol. 4, April 2007, pp. IV–1429–IV–1432.

[2] J. Serrà and E. Gómez, "Audio cover song identification based on tonal sequence alignment," in *Proc. ICASSP*. Las Vegas, USA: IEEE, 2008, pp. 61–64.

[3] J. Serrà, "Identification of versions of the same musical composition by processing audio descriptions," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, 2011.

[4] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proc. ISMIR*, Miami, USA, 2011, pp. 591 – 596.

[5] J. P. Bello, "Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats," in *Proc. ISMIR*, Vienna, Austria, 2007, pp. 239–244.

[6] T. Bertin-Mahieux and D. P. W. Ellis, "Large-scale cover song recognition using hashed chroma landmarks," in *Proc. WASPAA*. New York, USA: IEEE, 2011, pp. 117 – 120.

[7] T. Bertin-Mahieux and D. P. W. Ellis, "Large-scale cover song recognition using the 2d fourier transform magnitude," in *Proc. ISMIR*, Porto, Portugal, 2012, pp. 241 – 246.

[8] K. Lee, "Identifying cover songs from audio using harmonic representation," in *MIREX task on Audio Cover Song Identification*, 2006.

[9] B. Martin, D. G. Brown, P. Hanna, and P. Ferraro, "Blast for audio sequences alignment: A fast scalable cover identification tool," in *Proc. ISMIR*, Porto, Portugal, 2012, pp. 529 – 534.

[10] L. Oudre, Y. Grenier, and C. Févotte, "Template-based chord recognition : Influence of the chord types," in *Proc. ISMIR*, Kobe, Japan, 2009, pp. 153–158.

[11] M. Slaney, Y. Lifshits, and J. He, "Optimal parameters for locality-sensitive hashing," *Proceedings of the IEEE*, vol. 100, no. 9, pp. 2604–2623, 2012.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[13] M. Casey and M. Slaney, "Fast recognition of remixed music audio," in *Proc. ICASSP*, vol. 4, Honolulu, Hawaii, USA, 2007, pp. IV–1425–IV–1428.

[14] M. Casey, C. Rhodes, and M. Slaney, "Analysis of minimum distances in high-dimensional musical spaces," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 5, pp. 1015–1028, July 2008.

[15] Y. Yu, M. Crucianu, V. Oria, and L. Chen, "Local summarization and multi-level lsh for retrieving multi-variant audio tracks," in *Proc. ACM international conference on Multimedia*, Beijing, China, 2009, pp. 341–350.

[16] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proc. ISMIR*, Utrecht, Netherlands, 2010, pp. 135–140.

[17] M. Khadkevich and M. Omologo, "Time-frequency reassigned features for automatic chord recognition," in *Proc. ICASSP*, Prague, Czech Republic, 2011, pp. 181 – 184.

[18] M. Khadkevich and M. Omologo, "Reassigned spectrum-based feature extraction for gmm-based automatic chord recognition," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, pp. 1–12, 2013.