AUDIO COVER SONG IDENTIFICATION BASED ON TONAL SEQUENCE ALIGNMENT

Joan Serrà and Emilia Gómez

Music Technology Group
Universitat Pompeu Fabra, Barcelona, Spain
{jserra,egomez}@iua.upf.edu

ABSTRACT

Nowadays, the term cover song (or simply cover) can mean any new version, performance, rendition, or recording of a previously recorded track. Cover song identification is a task that has received increased popularity in the Music Information Retrieval (MIR) community in recent years, as it provides a direct and objective way for evaluating music similarity. In this paper, we propose a new method for determining the similarity between tonal sequences and, therefore, for identifying cover songs. This is based on a novel chroma similarity measure, and on a newly developed dynamic programming local alignment technique. Results confirm that the performance of the proposed system is significantly superior to other state-of-the-art approaches (more than 57% better).

Index Terms— Music, Information retrieval, Acoustic signal analysis, Multidimensional sequences, Dynamic programming

1. INTRODUCTION

As music collections are growing, it becomes necessary to keep them organized. Because manually annotating metadata is costly, an 'intelligent' system for organizing songs based in its content would be desirable. One critical block of such a system should deal with the concept of music similarity. But this can be very subjective, ill-defined and context-dependent. So, from a research perspective, a good starting point seems to be the identification of cover songs (or versions), where the similarity between them can be better defined, objectively measured, and context-independent. In addition, from the users perspective, finding all versions of a particular song can be useful and fun.

Tonal sequences are useful descriptors for cover song identification. In popular music, the main purpose of recording a version might be to investigate a radically different interpretation of the original song. Then, important changes at different musical facets (timbre, tempo, rhythm, structure, key, lyrics, language, etc.) are involved. Thus, it seems that the features that are mostly preserved are the main melody, and the overall tonal sequence.

Systems for cover song identification usually exploit these aspects and attempt to be robust against changes in other musical facets. In general, they either try to extract the predominant melody [1, 2], a chord progression [3, 4], or a chroma sequence [5, 6, 7]. Then, for obtaining a similarity measure, these sequences of descriptors are usually compared by means of Dynamic Time Warping (DTW) [1, 3, 5], an edit-distance variant [4, 6], or a simple correlation function [2, 7].

The method exposed here uses sequences of feature vectors describing tonality (in our case *Harmonic Pitch Class Profiles* [8], from now on HPCP), but presents relevant differences in two important aspects: we use a new binary similarity function between chroma features, and we develop a new local alignment algorithm for assessing resemblance between sequences.

2. COVER SONG IDENTIFICATION METHOD

2.1. Overview

In figure 1 we show a general block diagram of the system. It comprises four main sequential modules: pre-processing, similarity matrix creation, dynamic programming local alignment and post-processing.

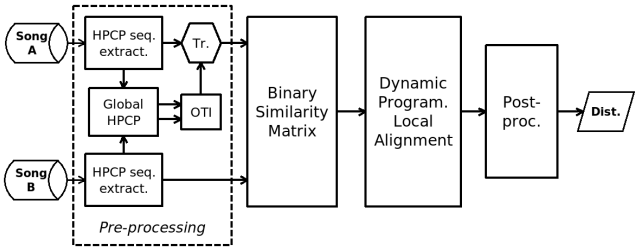

Fig. 1. General block diagram of the system.

From each pair of songs A and B being compared (inputs), we can obtain a distance between them (output). Pre-processing comprises extracting HPCP sequences and a global HPCP for each song. Then, one song is transposed to the key of the other (module named "Tr") by means of an *Optimal Transposition Index* (OTI, section 2.2). From these two sequences, a binary similarity matrix is computed. This last is the only input needed for a *Dynamic Programming Local Alignment* (DPLA) algorithm, which calculates a score matrix that gives highest ratings to best aligned subsequences. Finally, in the post-processing step, we obtain a normalized distance between the two processed songs. We now explain these steps in detail. Further information and justifications of our choices can be found in [9].

2.2. Pre-processing

For each song (A and B), we extract a sequence of HPCP feature vectors. The HPCP is an enhanced pitch class distribution (or chroma) feature computed in a frame-by-frame basis only using the local maxima of the spectrum within a certain frequency band. HPCPs

consider the presence of harmonic frequencies, and they are normalized to eliminate the influence of dynamics and instrument timbre (represented by a spectral envelope). The result (for our case) is a 36-bin octave-independent histogram representing the relative intensity of each $1/3$ of the 12 semitone equal tempered scale (HPCP vectors, equation 1). We refer to [8] for a detailed explanation on the feature extraction process. We represent an HPCP sequence by:

$$\begin{aligned} HPCP_A &= [\overrightarrow{h_{A,1}}, \overrightarrow{h_{A,2}}, \ldots, \overrightarrow{h_{A,i}}, \ldots, \overrightarrow{h_{A,n}}] \\ HPCP_B &= [\overrightarrow{h_{B,1}}, \overrightarrow{h_{B,2}}, \ldots, \overrightarrow{h_{B,j}}, \ldots, \overrightarrow{h_{B,m}}] \end{aligned} \quad (1)$$

In addition, a global HPCP vector is computed by averaging all HPCP vectors in a sequence, and this, as all HPCPs, is normalized by its maximum value. With the global HPCPs of two songs ($\overrightarrow{h_A}$ and $\overrightarrow{h_B}$), we compute what we call an *Optimal Transposition Index* (OTI), which represents the number of bins that an HPCP needs to circularly shift to have maximal resemblance to the other:

$$OTI(\overrightarrow{h_A}, \overrightarrow{h_B}) = \underset{0 \leq n \leq N_B - 1}{\operatorname{argmax}} \{\overrightarrow{h_A} \cdot circshift_R(\overrightarrow{h_B}, n)\} \quad (2)$$

where '$\cdot$' indicates a dot product, N_B is the number of bins of the feature vector considered, and $circshift_R(\overrightarrow{h}, n)$ is a function that rotates a vector ($\overrightarrow{h}$) n positions to the right. A circular shift of one position is a permutation of the entries in a vector where the last component becomes the first one and all the other components are shifted to the right. Equation 2 can be computed in $O(N_B \cdot \log(N_B))$ time (and thus, not becoming prohibitively time consuming for high resolution HPCPs) by means of the Fast Fourier Transform (FFT) properties related to the circular convolution [9].

The last operation of the pre-processing block consists in transposing both musical pieces to a common key. This is done by circularly shifting $OTI(\overrightarrow{h_A}, \overrightarrow{h_B})$ positions each HPCP in the whole sequence of one song (module labeled "Tr" in figure 1). So, the i-th HPCP for song A becomes:

$$\overrightarrow{h^{Tr}_{A,i}} = circshift_R(\overrightarrow{h_{A,i}}, OTI(\overrightarrow{h_A}, \overrightarrow{h_B})) \quad (3)$$

where superscript 'Tr' denotes musical transposition.

2.3. Similarity matrix

The next step is to compute a similarity matrix S between the pair of HPCP sequences obtained in previous section 2.2. Notice that the sequences can have different lengths n and m, and that, therefore, S will be an $n \times m$ matrix. Element (i, j) of the similarity matrix S, has the functionality of a local sameness measure between HPCP vectors $\overrightarrow{h^{Tr}_{A,i}}$ and $\overrightarrow{h_{B,j}}$ ($S_{i,j} = s(\overrightarrow{h^{Tr}_{A,i}}, \overrightarrow{h_{B,j}})$). In our case, this is binary (i.e., only two values are allowed).

We outline some reasons for using a binary similarity measure between chroma or HPCP features. First, as these features might not be in an euclidean space [10], we would prefer to avoid the computation of an euclidean-based (dis)similarity measure (in general, we think that tonal similarity, and therefore chroma feature distance, is a still far to be understood topic, with many of perceptual and cognitive open issues that require lots of research). Second, using only two values to represent similarity, the possible paths through the similarity matrix become more evident, leading us with a clear notion of where the two sequences agree and where they don't (see figure 2 for an example). In addition, binary similarity allows us to operate like many string alignment techniques do: just considering if two elements of the string are the same. With this, we have an expanded range of alignment techniques borrowed from string comparison, DNA or protein sequence alignment, symbolic time series similarity, etc. [11]. Finally, we believe that considering the binary similarity of an HPCP vector might be an easier (or at least more affordable) task to assess, than obtaining a reliable graded scale of resemblance between two HPCPs correlated with (sometimes subjective) perceptual similarity.

An intuitive idea to consider when deciding if two HPCP vectors refer to the same tonal root, is to keep circularly shifting one of them and calculate a similarity index for all possible transpositions. Then, if the transposition that leads to maximal similarity corresponds to less than a semitone, the two HPCP vectors are claimed to be the same. This idea can be formulated in terms of the OTI function explained in equation 2. As we are using a resolution of a $1/3$ of a semitone, the binary similarity measure between the two vectors is then obtained by:

$$s(\overrightarrow{h^{Tr}_{A,i}}, \overrightarrow{h_{B,j}}) = \begin{cases} +1 & \text{if } OTI(\overrightarrow{h^{Tr}_{A,i}}, \overrightarrow{h_{B,j}}) \in \{0,1,N_B\text{-}1\}, \\ -1 & \text{otherwise.} \end{cases} \quad (4)$$

Although the dot product is used in the calculation of $OTI(\overrightarrow{h^{Tr}_{A,i}}, \overrightarrow{h_{B,j}})$, the nonlinearity of the proposed $s(\overrightarrow{h^{Tr}_{A,i}}, \overrightarrow{h_{B,j}})$ leads us to a non-euclidean similarity measure that empirically works dramatically better than thresholding an euclidean-based distance[1].

(i) (ii) (iii)

Fig. 2. Examples of comparing two covers of the same song (i) and two songs that do not share a common tonal progression (ii) with the new binary similarity measure. We can see diagonal white lines in the former, while this pattern does not exist in the latter. The same similarity matrix of the two songs in (i) with an euclidean similarity measure is shown in (iii).

2.4. Dynamic programming local alignment (DPLA)

A binary similarity matrix S is the only input to our DPLA algorithm. Dynamic programming algorithms (such as DTW) have been proven to be a powerful tool for dealing with tempo variations [12]. In [9] we have seen that using global constraints and, thus, forcing warping paths to be around the alignment matrix main diagonal had a detrimental effect in final system performance. Instead, the use of local constraints [13] can help us preventing 'pathological warpings' and just admitting certain 'logical' tempo changes. Also in [9] it has been discussed the suitability of performing a local alignment to overcome strong song structure changes (i.e., to check all possible subsequences). The Smith-Waterman algorithm [14] is a well-known procedure for performing local sequence alignment in

[1] See table at `http://www.iua.upf.es/~jserra/ICASSP08appendix.html`

Molecular Biology. It was originally designed for determining similar regions between two nucleotide or protein sequences. Instead of looking at the total sequence, the Smith-Waterman algorithm compares segments of all possible lengths and optimizes the similarity measure.

So, in the same manner as the Smith-Waterman algorithm does, we create an $(n+1)\times(m+1)$ alignment matrix H through a recursive formula, that, in addition, incorporates some local constraints:

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + S_{i-1,j-1} - \delta_1(i,j) \\ H_{i-2,j-1} + S_{i-1,j-1} - \delta_2(i,j) \\ H_{i-1,j-2} + S_{i-1,j-1} - \delta_3(i,j) \\ 0 \end{cases} \quad (5)$$

for $3 \leq i \leq n+1$ and $3 \leq j \leq m+1$. $S_{i-1,j-1}$ corresponds to the value of the binary similarity matrix S at row $i-1$ and column $j-1$, and $\delta_k(i,j)$ denotes a penalty for a gap opening or extension. This latter value is set to 0 if $S_{i-1,j-1} > 0$ (no gap between $S_{i-1,j-1}$ and either $S_{i-2,j-2}$, $S_{i-3,j-2}$ or $S_{i-2,j-3}$), or to a positive value if $S_{i-1,j-1} \leq 0$. Empirically, a good value was found to be 0.5 for a gap opening (e.g., negative $S_{i-1,j-1}$ but positive $S_{i-3,j-2}$ in equation 5, option 2), and 0.6 for a gap extension (e.g., negative values in both $S_{i-1,j-1}$ and $S_{i-3,j-2}$ in the same option as the previous example). Notice that options 1, 2 and 3 in equation 5 represent a 'hybrid' between the ones employed in [13] and in [14]. Values of H have the interpretation that $H_{i,j}$ is the maximum similarity of two segments ending in $\overrightarrow{h^{Tr}_{A,i-1}}$ and $\overrightarrow{h_{B,j-1}}$ respectively. The zero is included to prevent negative similarity, indicating no similarity up to $\overrightarrow{h^{Tr}_{A,i-1}}$ and $\overrightarrow{h_{B,j-1}}$. Alignment matrix H is initialized as follows:

$$\begin{array}{ll} H_{i,1} = 0; & H_{1,j} = 0; \\ H_{k,2} = S_{k-1,1}; & H_{2,l} = S_{1,l-1}; \end{array} \quad (6)$$

for $1 \leq i \leq n+1$, $1 \leq j \leq m+1$, $2 \leq k \leq n+1$ and $2 \leq l \leq m+1$.

An example of the resultant matrix H is shown in figure 3. We can see clearly two local alignment traces, which correspond to two highly resemblant sections between two versions of the same song (from $H_{150,25}$ to $H_{250,100}$ and from $H_{280,25}$ to $H_{400,100}$).

2.5. Post-processing

In the last step of the method, only the best local alignment in H is considered. This means that the score determining the local subsequence similarity between two HPCP sequences, and, therefore, what we consider to be the similarity between two songs, corresponds to the value of H's highest peak:

$$Score(HPCP_A^{Tr}, HPCP_B) = \max\{H_{i,j}\} \quad (7)$$

for any i, j such that $1 \leq i \leq n+1$ and $1 \leq j \leq m+1$.

Finally, to obtain a distance value that is independent of the compared song lengths, the inverse of equation's 7 result is normalized by the maximum path length possible:

$$d(Song_A, Song_B) = \frac{n+m-1}{Score(HPCP_A^{Tr}, HPCP_B)} \quad (8)$$

where n and m are the respective lengths for songs A and B. A proper justification of this choices is done in [9].

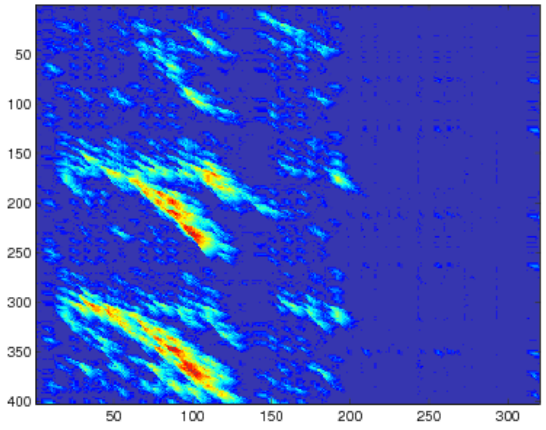

Fig. 3. Example of a local alignment matrix H between two covers. It can be seen that the two songs do not coincide entirely (just in two fragments), and that, mainly, their respective second halves are completely different.

3. EVALUATION

3.1. Personal music collection

To test the effectiveness of the system we compiled a music collection comprising 2053 songs distributed in different genres. Within these songs, there were 451 original pieces, and 1462 covers. The average number of covers per song was 4.24, ranging from 2 (the 'original' song + 1 cover) to 20 (the 'original' song + 19 covers). In order to add difficulty, there were also 140 songs from the same genres and artists as the originals that were not associated to any group of covers.

We queried all the covers and *canonical versions* and obtained a 1913x2053 distance matrix. This data was further processed in order to obtain several evaluation measures. The database and the methodology used are properly explained in [9]. With this music collection, we obtained an average Recall of 0.562 within the 10 first retrieved items. Precision at 1 was 0.722 and we reached an F-measure of 0.601.

3.2. MIREX Evaluation

The Music Information Retrieval Evaluation eXchange (MIREX) is an international effort to develop formal, common evaluation frameworks for MIR. It is coordinated and managed by the International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL) at the University of Illinois at Urbana-Champaign (UIUC). For the first time in 2006, there was an evaluation for cover song identification that was repeated in 2007 with an increasing number of participants.

The MIREX test data was composed of 30 *cover sets*, each one having 11 different versions. Therefore, the total cover song collection contained 30x11 = 330 songs. These were embedded in a database summing up a total of 1000 tracks. The test collection included a wide diversity of genres (e.g., classical, jazz, gospel, rock, folk-rock, etc.), and the variations spanned a variety of styles and orchestrations.

Measure	Range	**SG**	EC	JB	JEC	KL1	KL2	KP	IM
$TNCI_{10}$	[0-3300]	**1653**	1207	869	762	425	291	190	34
$MNCI_{10}$	[0-10]	**5.009**	3.658	2.633	2.309	1.288	0.882	0.576	0.103
MAP	[0-1]	**0.521**	0.330	0.267	0.238	0.13	0.086	0.061	0.017
$Rank_1$	[0-1000]	**9.367**	13.994	29.527	22.209	57.542	51.094	46.539	97.470
Runtime	[HH:MM]	**01:37(1)**	04:28(5)	04:32(8)	00:47(8)	10:45(8)	02:37(1)	03:51(1)	02:04(1)

Table 1. Results for MIREX 2007 Audio Cover Song task. Clock time measures are reported on the last line of the table (number of used threads in brackets). Performance values for the algorithm presented here are shown in the first column (SG).

Each of the 330 cover songs were used as queries and the systems were required to return a 330x1000 distance matrix. Systems were evaluated on the number of the songs from the same class/set as the query that were retrieved. Four measures were used to evaluate the performance of the algorithms: the total number of covers identified in top 10 answers ($TNCI_{10}$), the mean number of covers identified in top 10 (average performance, $MNCI_{10}$), the arithmetic Mean of Average Precision (*MAP*), and the rank of the first correctly identified cover ($Rank_1$). Notice that $MNCI_{10}/10$ leads an average Recall measure like the one used in section 3.1. Clock time measures (not CPU time) and the number of threads used were also provided.

A total of 8 different algorithms were presented to the MIREX 2007 Audio Cover Song task. Table 1 shows the overall summary results obtained[2]. Our algorithm (SG, first column) performed the best in all evaluation measures considered, reaching an average accuracy of 5.009 of correctly identified covers within the 10 first retrieved elements ($MNCI_{10}$) and a Mean Average Precision (*MAP*) of 0.521. Furthermore, the next best performing system reached and $MNCI_{10}$ of 3.658 and a *MAP* of 0.330, which represents a substantial difference to ours (57.88% superior in terms of *MAP*). In addition, statistical significance tests showed that the results for our system were significantly better than the 6 other systems presented in the contest.

4. CONCLUSIONS

We have presented a method for determining the similarity between songs by comparing tonal subsequences. The system's main novelty relies on two facts: a new binary similarity measure for chroma features and a custom-made dynamic programming local alignment algorithm for determining subsequence similarity.

The performance of the system was assessed by retrieving cover songs in a big music collection, reaching an average Recall value of 0.562 with a 2053 song database. Furthermore, the method explained here was evaluated in the MIREX 2007 Audio Cover Song contest, obtaining the highest values for all the evaluation measures considered, and being substantially superior to all the other algorithms that participated in it.

Although cover song identification is still a relatively new research topic, and systems dealing with this task can be further improved, we think that the method presented in this paper represents an important milestone into it.

5. ACKNOWLEDGEMENTS

The authors wish to thank their colleagues and staff at the MTG (UPF), specially Perfecto Herrera for his helpful ideas, reviews and constant support, and Graham Coleman. They also want to mention all the IMIRSEL team for the organization and running of MIREX evaluation.

This research has been partially funded by the EU-IP project PHAROS[3].

6. REFERENCES

[1] 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," *Proc. of the Int. Symposium on Music Information Retrieval (ISMIR)*, pp. 183–190, 2005.

[2] M. Marolt, "A mid-level melody-based representation for calculating audio similarity," *Proc. of the Int. Symposium on Music Information Retrieval (ISMIR)*, 2006.

[3] Ö. Izmirli, "Tonal similarity from audio using a template based attractor model," *Proc. of the Int. Symposium on Music Information Retrieval (ISMIR)*, 2005.

[4] J. P. Bello, "Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats," *Proc. of the Int. Symposium on Music Information Retrieval (ISMIR)*, pp. 239–244, September 2007.

[5] E. Gómez, B. S. Ong, and P. Herrera, "Automatic tonal analysis from music summaries for version identification," *Conv. of the Audio Engineering Society (AES)*, October 2006.

[6] M. Casey and M. Slaney, "The importance of sequences in musical similarity," *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2006.

[7] D. P. W. Ellis and G. E. Polliner, "Identifying cover songs with chroma features and dynamic programming beat tracking," *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, April 2007.

[8] E. Gómez, *Tonal description of music audio signals*, Ph.D. thesis, MTG, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[9] J. Serrà, "Music similarity based on sequences of descriptors: tonal features applied to audio cover song identification," M.S. thesis, MTG, Universitat Pompeu Fabra, Barcelona, Spain, 2007.

[10] C. L. Krumhansl, *Cognitive foundations of musical pitch*, Oxford University Press, New York, 1990.

[11] D. Gusfield, *Algorithms on strings, trees and sequences: computer sciences and computational biology*, Cambridge University Press, 1997.

[12] L. R. Rabiner and B. H. Juang, *Fundamentals of speech recognition*, Prentice, Englewood Cliffs, NJ, 1993.

[2] See the complete results at `http://www.music-ir.org/mirex/2007/index.php/Audio_Cover_Song_Identification_Results`

[3] `http://www.pharos-audiovisual-search.eu/`

[13] C. Myers, "A comparative study of several dynamic time warping algorithms for speech recognition," M.S. thesis, Massachussets Institute of Technology (MIT), USA, 1980.

[14] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, pp. 195–197, 1981.