Chroma Binary Similarity and Local Alignment Applied to Cover Song Identification

Joan Serrà*, Emilia Gómez, Perfecto Herrera and Xavier Serra

***Abstract*—We present a new technique for audio signal comparison based on tonal subsequence alignment and its application to detect cover versions (i.e., different performances of the same underlying musical piece). Cover song identification is a task whose popularity has increased in the Music Information Retrieval (MIR) community along in the past, as it provides a direct and objective way to evaluate music similarity algorithms. This article first presents a series of experiments carried out with two state-of-the-art methods for cover song identification. We have studied several components of these (such as chroma resolution and similarity, transposition, beat tracking or Dynamic Time Warping constraints), in order to discover which characteristics would be desirable for a competitive cover song identifier. After analyzing many cross-validated results, the importance of these characteristics is discussed, and the best-performing ones are finally applied to the newly proposed method. Multiple evaluations of this one confirm a large increase in identification accuracy when comparing it with alternative state-of-the-art approaches.**

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

I. INTRODUCTION

IN THE present times, any music listener may have thousands of songs stored in a hard disk or in a portable MP3 player. Furthermore, on-line digital music stores own large music collections, ranging from thousands to millions of tracks. Additionally, the ‘unit’ of music transactions has changed from the entire album to the song. Thus, users or stores are faced to search through vast music databases at the song level. In this context, finding a musical piece that fits one’s needs or expectancies may be problematic. Therefore, it becomes necessary to organize them according to some sense of similarity. It is at this point where determining if two musical pieces share the same melodic or tonal progression becomes interesting and useful. To address this issue, from a research perspective, a good starting point seems to be the identification of cover songs (or versions), where the relationship between them can be qualitatively defined, objectively measured, and is context-independent. In addition, from the users perspective, finding all versions of a particular song can be valuable and fun.

It is important to mention that the concept of music similarity, and more concretely, finding cover songs in a database, has a direct implication to musical rights management and licenses. Also, learning about music itself, discovering the musical essence of a song, and other many topics related with music perception and cognition are partially pursued by this research. Furthermore, the techniques presented here can be exploited for general audio signal comparison, where cover/version identification is just an application among other possible ones.

The expressions *cover song* and *version* may have different and somehow fuzzy connotations. A *version* is intended to be what every performer does by playing precomposed music, while the term *cover song* comes from a very different tradition in pop music, where a piece is composed for a single performer or group. Cover songs were, originally, part of a strategy to introduce ‘hits’ that had achieved significant commercial success from other sections of the record-buying public, without remunerating any money to the original artist or label. Nowadays, the term has nearly lost these purely economical connotations. Musicians can play covers as a homage or a tribute to the original performer, composer or band. Sometimes, new versions are made for translating songs to other languages, for adapting them to a particular country/region tastes, for contemporising familiar or very old songs, or for introducing new artists. In addition, cover songs represent the opportunity to perform a radically different interpretation of a musical piece.

Today, and perhaps not being the proper way to name it, a cover song can mean any new version, performance, rendition or recording of a previously recorded track [1]. Therefore, we can find several musical dimensions that might change between two covers of the same song. These can be related to timbre (different instruments, configurations or recording procedures), tempo (global tempo and tempo fluctuations), rhythm (e.g., different drum section, meter, swinging pattern or syncopation), song structure (eliminating introductions, adding solo sections, choruses, codas, etc.), main key (transposition to another tonality), harmonization (adding or deleting chords, substituting them by related ones, adding tensions, ...) and lyrics (e.g., different languages or words).

A robust mid-level characteristic that is largely preserved under the mentioned musical variations is a tonal sequence (or a harmonic progression [2]). Tonality is ubiquitous and most listeners, either musically trained or not, can identify the most stable pitch while listening to tonal music. Furthermore, this process is continuous and remains active throughout the sequential listening experience [3], [4]. From the point of view of the Music Information Retrieval (MIR) field, clear insights about the importance of temporal and tonal features in a music similarity task have been evidenced [5], [6], [7].

Tonal sequences can be understood as series of different

Manuscript received November 30, 2007; revised February 21, 2008; accepted April 5, 2008. This research has been partially funded by the EU-IP project PHAROS IST-2006-045035: http://www.pharos-audiovisual-search.eu

J. Serrà, E. Gómez, P. Herrera and X. Serra are with the Music Technology Group, Universitat Pompeu Fabra, Ocata 1 (3rd floor), 08003 Barcelona, Spain, phone: +34-93-542-2864, fax: +34-93-542-2202, e-mail: {jserra,egomez,pherrera,xserra}@iua.upf.edu

note combinations played sequentially. These notes can be unique for each time slot (a melody) or can be played jointly with others (chord or harmonic progressions). 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 [8], [9], a chord progression [10], [11], or a chroma sequence [12], [13], [14], [15], [16]. Some methods do not take into account (at least explicitly) key transposition between songs [13], [14], but the usual strategy is to normalize these descriptor sequences in respect to the key. This is usually done by means of a key profile extraction algorithm [9], [10], [15], or by considering all possible musical transpositions [12], [8], [11], [16]. Then, for obtaining a similarity measure, descriptor sequences are usually compared by means of Dynamic Time Warping (DTW) [8], [10], [15], an edit-distance variant [7], [11], string matching [12], Locality Sensitive Hashing (LSH) [14], or a simple correlation function or a cosine angle [9], [13], [16]. In addition, a beat tracking method might be used [9], [12], [16], or a song summarization or chorus extraction technique might be considered [9], [15].

Techniques for predominant melody extraction have been extensively researched in the MIR community [17], [18], [19], as well as key/chord identification engines [20], [21]. Also, chroma-based features have become very popular [22], [23], [24], [25], with applications in various domains such as pattern discovery [26], audio thumbnailing and chorus detection [27], [28], or audio alignment [5], [29].

Regarding alignment procedures and sequence similarity measures, DTW [30] is a well known technique used in speech recognition for aligning two sequences which may vary in time or speed and for measuring similarity between them. Also, several edit-distance variants [31] are widely used in very different disciplines such as text retrieval, DNA or protein sequence alignment [32], or MIR itself [33], [34]. If we use audio shingles (i.e., high-dimensional feature vectors concatenations) to represent different portions of a song sequence, LSH solves fast approximate nearest neighbor search in high dimensions [35].

One of the main goals of this article is to present a study of several factors involved in the computation of alignments of musical pieces and similarity of (cover) songs. To do this, the impact of a set of factors in state-of-the-art cover song identification systems is measured. We experiment with different resolution of chroma features, with different local cost functions (or distances) between chroma features, with the effect of using different musical transposition methods, and with the use of a beat tracking algorithm to obtain a tempo-independent chroma sequence representation. In addition, as DTW is a well known and extensively employed technique, we test two underexplored variants of it: DTW with global and local constraints. All these experiments are oriented to elucidate the characteristics that a competitive cover song identification system should have. We then apply this knowledge to a newly proposed method, which uses sequences of feature vectors describing tonality (in our case Harmonic Pitch Class Profiles [25], from now on HPCP), but it presents relevant differences in two important aspects: we use a novel binary similarity function between chroma features, and we develop a new local alignment algorithm for assessing resemblance between sequences.

The rest of the paper is organized as follows. First, in section II, we explain our test framework. We describe the methods used to evaluate several relevant parameters of a cover song identification system (chroma resolution and similarity, key transposition, beat tracking and DTW constraints), and the descriptors employed across all these experiments. We also introduce the database and the evaluation measures that are employed along this study. Then, in section III, we sequentially present all the evaluated parameters and the obtained results. In section IV, we propose a new method for assessing the similarity between cover songs. This is based on the conclusions obtained through our experiments (summarized in section III-F) and on two main aspects: a new chroma similarity measure and a novel dynamic programming local alignment algorithm. Finally, a short conclusions section closes the study.

II. Experimental framework

A. Tonality descriptors

All the implemented methods use the same feature set: sequences of *Harmonic Pitch Class Profiles* (HPCP) [25]. 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. Chroma features are widely used in the literature and proven to work quite well for the task at hand [13], [15], [16]. In general, chroma features should be robust to noise (e.g., ambient noise or percussive sounds), independent of timbre and played instruments (so that the same piece played with different instruments has the same tonal description), and independent of loudness and dynamics. These are some of the qualities that might make them lead to better results for cover song identification when comparing them, for instance, with MFCCs [7], [14].

In addition to using the local maxima of the spectrum within a certain frequency band, HPCPs are tuning independent (so that the reference frequency can be different from the standard A 440 Hz), and consider the presence of harmonic frequencies. The result of HPCP computation is a 12, 24 or 36-bin (depending on the desired resolution) octave-independent histogram representing the relative intensity of each 1, 1/2 or 1/3 of the 12 semitones of the equal tempered scale. A schema of the extraction process and a plot of the resulting HPCP sequence are shown in figures 1 and 2.

We start by cutting the song into short overlapping and windowed frames. For that, we use a Blackman-Harris (62 dB) window of 93 ms length with a 50% frame overlapping. We perform a spectral analysis using the Discrete Fourier Transform (DFT), and the spectrum is whitened by normalizing the amplitude values with respect to the spectral envelop. From the obtained spectrum, we compute a set of local maxima or peaks and we select the ones with frequency values $f_i \epsilon (40, 5000)$ Hz. The selected spectral peaks are summarized in an octave-independent histogram according to a reference frequency (around 440 Hz). This reference frequency is estimated by

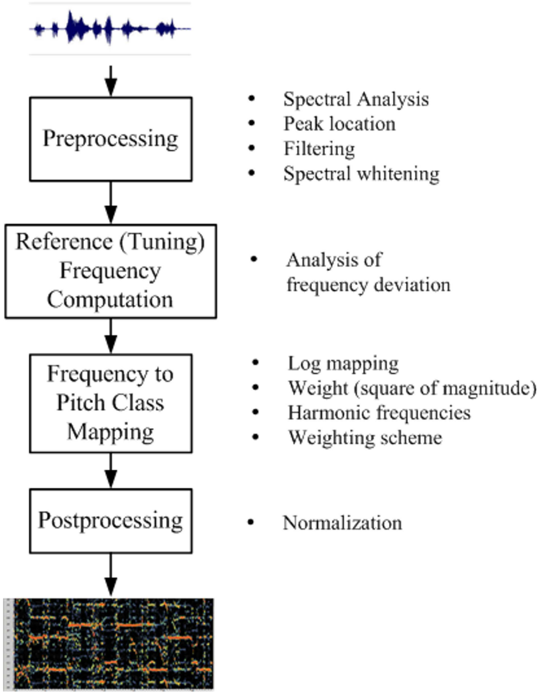

Fig. 1. General HPCP feature extraction block diagram. Audio (top) is converted to a sequence of HPCP vectors (bottom) that evolves with time.

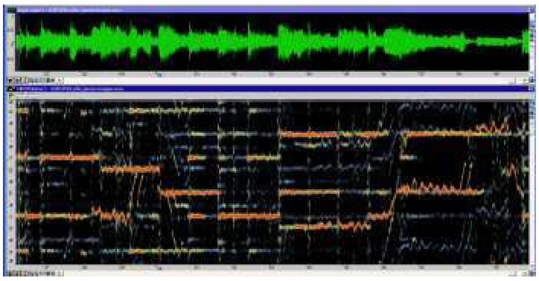

Fig. 2. Example of a high-resolution HPCP sequence (bottom panel) corresponding to an excerpt of the song "Imagine" by John Lennon (top panel). In the HPCP sequence, time (in frames) is represented in the horizontal axis and chroma bins are plotted in the vertical axis.

analyzing the deviations of the spectral peaks with respect to an equal-tempered chromatic scale. A global estimate of this reference frequency is employed for all the analyzed frames.

Instead of contributing to a single HPCP bin, each peak frequency f_i contributes to the HPCP bin(s) that are contained in a certain window around its frequency value. The peak contribution i is weighted using a cos^2 function around the bin frequency. The length of the weighting window l have been empirically set to $\frac{4}{3}$ semitones. This weighting procedure minimizes the estimation errors that we find when there are tuning differences and inharmonicity present in the spectrum, which could induce errors when mapping frequency values into HPCP bins.

In addition, in order to make harmonics contribute to the pitch class of its fundamental frequency, we also introduce an additional weighting procedure: each peak frequency f_i has a contribution to its $nHarmonics = 8$ sub-harmonics. We make this contribution decrease along frequency using an exponential function.

The HPCP extraction procedure employed here is the same that has been used in [15], [36], [37], [25], and the parameters mentioned in this paragraph have been proven to work well for key estimation and chord extraction in the previously cited references.

An exhaustive comparison between 'standard' chroma features and HPCPs is presented in [25] and [38]. In [25], a comparison of different implementations of chroma features (Constant-Q profiles [39], Pitch Class Profiles (PCP) [20], chromagrams [21] and HPCP) with MIDI-based Muse Data [40] is provided. The correlation of HPCP with Muse Data was higher than 0.9 for all the analyzed pieces (48 Fugues of Bach's WTC) and HPCPs outperformed the Constant-Q profiles, chromagrams and PCPs. We also compared the use of different HPCP parameters, arriving to optimal results with the ones used in the present work. In [38], the efficiency of different sets of tonal descriptors for music structural discovery was studied. Herein, the use of three different pitch-class distribution features (i.e., Constant-Q Profile, PCP and HPCP) was explored to perform structural analysis of a piece of music audio. A database of 56 audio files (songs by The Beatles) was used for evaluation. The experimental results showed that HPCP were performing best, yielding an average of 82% of accuracy in identifying structural boundaries in music audio signals.

B. *Studied methods*

We now describe two methods that have served us to test several important parameters of a cover song identification system, as a baseline for further improvements [16], [25]. We have chosen them because they represent in many ways the state-of-the-art. Their main features are the use of global alignment techniques and common feature dissimilarity measures. In subsequent sections, we differentiate these two methods by its alignment procedure (cross-correlation or Dynamic Time Warping), but other procedures are characteristic for each one (such as audio features, dissimilarity measure between feature vectors, etc.).

1) Cross-correlation approach: A quite straightforward approach is presented in [16]. This method finds cover versions by cross-correlating chroma vector sequences (representing the whole song) averaged beat-by-beat. It seems to be a good starting point since it was found to be superior to other methods presented to MIREX 2006 evaluation contest[1]. We worked with a similar version of the forementioned system. We re-implemented the algorithm proposed by the authors[2] in order to consider the same chroma features for all the methods (HPCPs) and to ease the introduction of new functionalities and improvements. We now describe the followed steps.

First of all, HPCP features are computed. Each frame vector is normalized by dividing it by its maximum amplitude, as shown in figure 1. In addition, beat timestamps are computed

[1] See the complete results at http://www.music-ir.org/mirex/2006/index.php/Audio_Cover_Song (Accessed 28 Jan. 2008)

[2] http://labrosa.ee.columbia.edu/projects/coversongs (Accessed 28 Jan. 2008)

with an algorithm adapted from [41], [42] using the *aubio* library[3].

The next step is to average the frame-based HPCP vectors contained in between each two beat timestamps. With this, we obtain a tempo-independent HPCP sequence. In order to account for key changes, the two compared HPCP sequences are usually transposed to the same key by means of a key extraction algorithm or an alternative approach (see section III-C). Another option is the one proposed in [16], where the sequence similarity measure is computed for all possible transpositions and the maximum value is then chosen.

In this approach, sequence similarity is obtained through cross-correlation. That is, we calculate a simple cross-correlation between each two tempo-independent HPCP sequences for each song being compared (with possibly different lengths). The cross-correlation values are further normalized by the length of the shorter segment, so that the measure is bounded between zero and one. Note that a local distance measure between HPCPs must be used. The most usual thing is to use an euclidean-based distance, but other measures can be tried (see section III-B).

In [16], the authors 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. So, to maximize these local maxima, cross-correlation was high-pass filtered. Finally, the final measure representing the dissimilarity between two songs is obtained with the reciprocal of the maximum peak value of this high-pass filtered cross-correlation.

2) Dynamic Time Warping approach: Another approach for detecting cover songs was implemented, reflecting the most used alignment technique in the literature: Dynamic Time Warping (DTW). The followed method has a very high resemblance with the one presented in [25].

We proceed by extracting HPCP features in the same way as the previous approach (section II-B1). Here, we do not use any beat tracking method because DTW is specially designed for dealing with tempo variations (see section III-D). For speeding up calculations, a usual strategy is to average each k consecutive descriptors vectors (frames). We call this value (k) the *averaging factor*. Here, each HPCP feature vector is also normalized by its maximum value. We deal with key invariance just in the same way than the previous approach (section II-B1) and transpose the HPCP sequences representing the two songs' tonal progressions to a common key.

To align these two sequences (which can have different lengths n and m), we use the DTW algorithm [30]. It basically operates by recursively computing an $n \times m$ cumulative distance matrix by using the value of a local cost function. This local cost function is usually set to be any euclidean-based distance, though in [15], [25] the correlation between the two HPCP vectors is used to define the dissimilarity measure (see section III-B). With DTW, we obtain the total alignment cost between two HPCP sequences in matrix element (n, m). We can also obtain an alignment path whose length acts as a normalization factor.

C. Evaluation methodology

To test the effectiveness of the implemented systems under different parameter configurations, we compiled a music collection comprising 2053 commercial songs distributed in different musical genres. Within these songs, there were 451 original pieces (we call them *canonical versions*) and 1462 covers. Songs were obtained from personal music collections. The average number of covers per song was 4.24, ranging from 2 (the original song plus 1 cover) to 20 (the original song plus 19 covers). There were also 140 'confusing songs' from the same genres and artists as the original ones that were not associated to any cover group. A special emphasis was put in the variety of styles and the employed genres for each cover set. A complete list of the music collection can be found in our web page[4].

Due to the high computational cost of the implemented cover song identification algorithms, we have restricted the music collection for preliminary experiments. We simultaneously employed two non-overlapping smaller subsets of the whole song database, intended to be as representative as possible of the entire corpus. We provide some statistics in table I.

TABLE I
SONG COMPILATIONS USED. DB75, DB330 AND DB2053 CORRESPOND TO THE NAMES WE GIVE TO THE DIFFERENT DATABASES. '⋆' DENOTES AVERAGE NUMBER OF COVERS PER GROUP. IN DB75 AND DB330 THERE WERE NO 'CONFUSING SONGS'

	DB75	DB330	DB2053
Total number of songs	75	330	2053
Number of cover sets	15	30	451
Covers per set	5	11	4.24⋆

We queried all the covers and canonical versions and obtained a distance matrix whose dimensions depended on the number of songs. This data was further processed in order to obtain several evaluation measures. Here, we mainly show the results corresponding to standard F-measure and average Recall (R_x) [43]. This last measure was computed as the mean percentage of identified covers within the first x answers. All experiments were evaluated with these measures, and, most of the time, other alternative metrics were highly correlated with the previous ones. A qualitative assessment of valid evaluation measures for this cover song system was presented in [44].

III. EXPERIMENTS

The next subsections describe the tests carried out to evaluate the impact of several system parameters and procedures in both methods explained in section II-B. Our hypothesis was that these had a strong influence in final identification accuracy and shouldn't be blindly assigned. To our knowledge, this is one of the first systematic study of this kind that has been made until now (with, perhaps, the exception of [11], where the author evaluated the influence of key shifting, cost gap

[3] http://aubio.org (Accessed 28 Jan. 2008)

[4] http://mtg.upf.edu/∼jserra/files/coverdatabase.csv.tar.gz

insertions and character swaps in a string alignment method used for cover song identification, in addition to the use of a beat-synchronous set).

In our experiments, we aimed at measuring, on a state-of-the-art cover song identification system, the impact of the following factors [45]: (a) the resolution of the chroma features, (b) the local cost function (or distance) between chroma features, (c) the effect of using different key transposition methods, and (d) the use of a beat tracking algorithm to obtain a tempo-independent chroma sequence representation. In addition, as DTW is a well known and extensively employed technique, we wanted to (e) test two underexplored variants of it: DTW with global and local constraints. A wrap-up discussion on these factors is provided in section III-F. Finally, we want to highlight that through all experiments reported in this section, all combinations of parameters cited in each subsection were studied. We report average performance results for each subsection given that all parameter combinations resulted in similar behaviours. Different behaviours are properly highlighted through the text, if any.

A. Effect of chroma resolution

Usually, chroma features are represented in a 12-bin histogram, each bin corresponding to 1 of the 12 semitones of the equal-tempered scale. But higher resolutions can be used to get a finer pitch class representation. Other commonly used resolutions are 24 and 36 bins [25] (corresponding to 1/2 or 1/3 of a semitone). We tested these three values in our experiments. The resolution parameter was changed in the HPCP extraction method of the approaches explained in section II-B.

The average identification accuracy across experiments with two different chroma similarity measures (section III-B) and two key transposition methods (section III-C) are shown in table II. In all the experiments, and independently of the HPCP distance used and the transposition made, the greater the HPCP resolution, the better the accuracy we got (F-measure more than 12% better).

TABLE II
F-MEASURE AND AVERAGE RECALL WITHIN THE FIRST FOUR RETRIEVED SONGS FOR DIFFERENT HPCP RESOLUTIONS. AVERAGE OF DIFFERENT CROSS-CORRELATION APPROACH VARIANTS EVALUATED WITH DB75

Resolution	F-measure	R_4
12 bins	0.495	0.429
24 bins	0.511	0.435
36 bins	0.558	0.489

B. Effect of chroma similarity measures

In order to test the importance of the used HPCP distance measure, we evaluated two similarity measures: cosine similarity and the correlation between feature vectors. These two measures were chosen because they are commonly used in the literature. Correlation has been used in [15], [25], and is inspired on the cognitive aspects of pitch processing in humans [46]. Furthermore, for key extraction, it was found to work better than the simple euclidean distance between HPCP vectors [25].

Tests were made with the methods exposed in section II-B and the two measures cited above. The results are shown in table III. We observe that the employed HPCP distance plays a very important role. This aspect of the system can yield to more than a 13% accuracy improvement for some tests [45]. In all trials made with different resolutions and ways of transposing songs, correlation between HPCPs was found to be a better similarity measure than cosine distance[5]. The former gives a mean F-measure improvement, among the tested variants, of approximately 6%.

TABLE III
F-MEASURE AND AVERAGE RECALL WITHIN THE FIRST FOUR RETRIEVED SONGS FOR COSINE DISTANCE (d_{COS}) AND CORRELATION DISTANCE (d_{CORR}). AVERAGE OF DIFFERENT CROSS-CORRELATION APPROACH VARIANTS EVALUATED WITH DB75

Distance used	F-measure	R_4
d_{COS}	0.504	0.436
d_{CORR}	0.537	0.461

C. Effect of key transposition

In order to account for songs played in a different key than the original one, we calculated a global HPCP vector and we transposed (circularly-shifted) one HPCP sequence to the other's tonality. This procedure was introduced in both methods described in section II-B. A global HPCP vector was computed by averaging all HPCPs in a sequence, and it was normalized by its maximum value as all HPCPs. With the global HPCPs of two songs ($\overrightarrow{h_A}$ and $\overrightarrow{h_B}$), we computed what we call the *Optimal Transposition Index* (from now on OTI), which represents the number of bins that an HPCP needs to be circularly shifted to have maximal resemblance to the other:

$$OTI(\overrightarrow{h_A}, \overrightarrow{h_B}) = \underset{0 \le id \le N_H - 1}{\operatorname{argmax}} \{\overrightarrow{h_A} \cdot circshift_R(\overrightarrow{h_B}, id)\} \quad (1)$$

where '·' indicates a dot product, N_H is the HPCP size considered, and $circshift_R(\overrightarrow{h}, id)$ is a function that rotates a vector ($\overrightarrow{h}$) id 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. Then, to transpose one song, for each HPCP vector i in the whole sequence we compute:

$$\overrightarrow{h_{A,i}^{Tr}} = circshift_R(\overrightarrow{h_{A,i}}, OTI) \quad (2)$$

where superscript ^{Tr} denotes musical HPCP transposition.

In order to evaluate the goodness of this new procedure for transposing both songs to a common key, an alternative way of computing a transposed HPCP sequence was introduced. This consisted on calculating the main tonality for each piece using a key estimation algorithm [25]. This algorithm is a state-of-the-art approach with an accuracy of 75% for real

[5] http://mtg.upf.edu/∼jserra/chromabinsimappendix.html

audio pieces [36], and scored among the first classified algorithms in the MIREX 2005 contest[6] with an accuracy of 86% with synthesized MIDI files. With this alternative procedure, once the main tonality was estimated, the whole song was transposed according to this estimated key. A possibly better way of dealing with key changes would be to calculate the similarity measures for all possible transpositions and then take the maximum [16]. We have not tested this procedure since for high HPCP resolutions it becomes computationally expensive.

OTI and key transposition methods were compared across several HPCP resolutions (section III-A) and two different HPCP distance measures (section III-B). The averaged identification accuracy is shown in table IV. It can be clearly seen that a key estimation algorithm has a detrimental effect to overall results (F-measure 17% worse). This was also independent of the number of bins and the HPCP distance used[7]. We have evaluated dependence of the number of HPCP bins, and HPCP distance, and we have found that they had similar behavior. Therefore, it seems appropriate to transpose the songs according to the *OTI* of the global HPCP vectors. Apart from testing the appropriateness of our transposition method, we were also pursuing the impact that different transposition methods could have, which we see is quite important in table IV.

TABLE IV
F-MEASURE AND AVERAGE RECALL WITHIN THE FIRST FOUR RETRIEVED SONGS FOR GLOBALHPCP + OTI TRANSPOSITION METHOD AND BY USING A KEY ESTIMATION ALGORITHM. AVERAGE OF DIFFERENT CROSS-CORRELATION APPROACH VARIANTS EVALUATED WITH DB75

Method	F-measure	R_4
GlobalHPCP + OTI	0.569	0.500
Key finding algorithm	0.474	0.400

D. Effect of beat tracking and averaging factors

In the cross-correlation approach (section II-B1), HPCP vectors were averaged beat-by-beat. With the DTW approach of section II-B2, we expected DTW being able to cope with tempo variations. To demonstrate this, we performed some tests with DTW. In these, several *averaging factors* were also tried.

Experiments were done with 5 different DTW algorithms (see section III-E). In these and subsequent experiments HPCP resolution was set to 36, correlation was used to assess the similarity between HPCP vectors and we employed OTI-based transposition. Results shown in table V are the average identification accuracy values obtained across these different implementations. We have to note that taking the arithmetic mean of the respective evaluation measures masks the concrete behaviour of them along different averaging factors (information regarding the effect of different averaging factors upon considered constraints can be found in subsequent section III-E). Nevertheless, for all the tested variants, better accuracies were reached with averaging HPCPs in a frame basis, than using beat-by-beat averaging. A similar result using the Needleman-Wunsch-Sellers algorithm [47] reported in [11] supports our findings.

TABLE V
F-MEASURE AND AVERAGE RECALL WITHIN THE FIRST FOUR RETRIEVED SONGS FOR DIFFERENT *averaging factors* (INCLUDING BEAT AVERAGING). CORRESPONDING TIME FACTOR IS EXPRESSED IN THE SECOND COLUMN. AVERAGE OF DIFFERENT DTW APPROACH VARIANTS EVALUATED WITH DB75

Averaging factor (frame count)	Averaging length (seconds)	F-measure	R_4
Beat	variable	0.469	0.417
5	0.232	0.470	0.419
10	0.464	0.494	0.448
15	0.696	0.511	0.465
20	0.929	0.514	0.463
25	1.161	0.512	0.466
30	1.393	0.510	0.461
40	1.856	0.487	0.434

E. Effect of DTW global and local constraints

We can apply different constraints to a DTW algorithm in order to decrease the number of paths considered during the matching process. These constraints are desirable for two main purposes: to reduce computational costs and to prevent 'pathological' warpings. 'Pathological' warpings are considered the ones that, in an alignment, assign several multiple values of a sequence to just one value of the other sequence. This is easily seen as a straight line in the DTW matrix (an example is shown in the first plot of figure 3).

To test the effect of these constraints we implemented 5 variants of a DTW algorithm: the one mentioned in section II-B2, two globally constrained DTW algorithms, and two locally constrained ones:

- Simple DTW: This implementation corresponds to the standard definition of DTW, where no constraints are applied [30].
- Globally constrained DTW: Two implementations were tried. One corresponds to Sakoe-Chiba constraints [48] and the other one to the Itakura parallelogram [49]. With these global constraints, elements far from the diagonal of the $n \times m$ DTW matrix are not considered (see figure 4). A commonly used value for that in many speech recognition tasks is 20% [30].
- Locally constrained DTW: To further specify the optimal path, some local constraints can be applied in order to guarantee that excessive time scale compression or expansion is avoided. We specified two local constraints that were found to work in a plausible way with speech recognition [50]. From this reference, *Type 1* and *Type 2* constraints were chosen (we denote them *MyersT1* and *MyersT2* respectively). For both, the recursive relation of DTW is changed in such a way that in element (i, j) of a DTW cumulative distance matrix, we only pay attention to warpings $(i-1, j-1)$ (no tempo deviation), $(i-2, j-1)$ ($2x$ tempo deviation) and $(i-1, j-2)$ ($0.5x$ tempo deviation). So, we allow maximal deviations of the double

[6] http://www.music-ir.org/mirex/2005/index.php/Audio_and_Symbolic_Key_Finding (Accessed 29 Jan. 2008)

[7] http://mtg.upf.edu/∼jserra/chromabinsimappendix.html

or half the tempo. This seems reasonable for us since, for instance, if the original song is at 120 B.P.M., a cover may not be at less than 60 B.P.M. or more than 240 B.P.M. The difference between *MyersT1* and *MyersT2* constraints relies in the way we weight this warpings: considering intermediate distances for the former, and double-weighting the distance between elements i and j for the latter [50].

These three implementations were evaluated across different averaging factors (see section III-D) and the means of the F-measure and average recall within the 4 first answered items (R_4) were taken. Results can be seen in table VI. In general, better accuracies are achieved with local constraints, whereas global constraints yielded the worst results.

TABLE VI
F-MEASURE AND AVERAGE RECALL WITHIN THE FIRST FOUR RETRIEVED SONGS FOR DIFFERENT DTW ALGORITHMS IMPLEMENTING GLOBAL AND LOCAL CONSTRAINTS. EVALUATION WAS DONE WITH DB75

Alg. name	Constr. type	F-measure	R_4
Sakoe-Chiba	Global	0.321	0.283
Itakura	Global	0.344	0.304
Simple DTW	No constr.	0.600	0.541
MyersT2	Local	0.608	0.552
MyersT1	Local	0.624	0.570

There is one important fact about local constraints that needs to be remarked and that can be appreciated in table VII. In general (except for the locally constrained methods), as the framelength decreases, it can be seen that identification accuracy does so. This is due to the fact that lower framelengths introduce the creation of 'pathological' warping paths (straight lines in the DTW matrix) that do not correspond to the true alignment (a straight line indicates several points of one sequence aligned just to one point of the other, left picture in figure 3). This makes the path length to increase, and since we normalize the final result by this value to yield sequence length independence, the final distance value decreases. Then, false positives are introduced in the final outcomes of the algorithm. Figure 3 shows the same part for matrices obtained after a simple and a locally constrained DTW approach. Local constraints prevent DTW from these undesired warpings. If there is a single horizontal or vertical step in the warping path, they force them to be the opposite way in next recurrent step. This is why the accuracy of locally constrained methods keeps increasing while lowering the averaging factor.

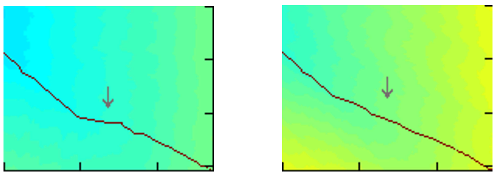

Fig. 3. Parts of the matrix obtained with a simple (left) and locally constrained (MyersT1, right) DTW approach for the same two songs. On the left we can observe some 'pathological' warpings, while on the right, these have disappeared.

Also in table VII, we observe that the identification accuracy for globally constrained methods is significantly lower than for the other ones. This is due to the fact that, by using these global constraints, we restrict the paths to be around the DTW matrix main diagonal. To understand the effect of that, as an example, we consider a song composed by two parts that are the same ($S_1 = AA$) and another song (a cover) with nearly half the tempo ($'$) and composed by only one of these parts ($S_2 = A'$). The plots in figure 4 graphically explain this idea. The first one (left) was generated using a method with no constraints. We observe that the best path (straight diagonal red line) goes from $(1, 1)$ to more or less $(20, 10)$ (horizontal axis lower-half part). This is logical since S_2 (vertical axis) is a half-tempo kind-of repetition of one part of S_1 (horizontal axis). The middle plot corresponds to the same matrix with Sakoe-Chiba constraints. We observe that the 'optimal' path we could trace with the first plot has been broken by the effect of the global constraints. A similar situation occurs with Itakura constraints (right plot).

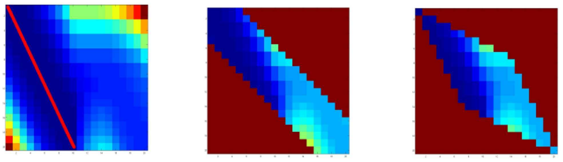

Fig. 4. Examples of an unconstrained DTW matrix (left), and Sakoe-Chiba (center) and Itakura (right) global constraints for S_1 (x-axis) and S_2 (y-axis). As this is an intuitive example, coordinate units in the horizontal and vertical axes are arbitrary.

F. Discussion

In previous subsections we have studied the influence of several aspects in two state-of-the-art methods for cover song identification. All the analyzed features proved to have a direct (and sometimes dramatic) impact in the final identification accuracy. We are now able to summarize some of the key aspects that should be considered when identifying cover songs. These aspects have been considered as a basis to design our approach, which will be presented in the following sections.

1) Audio features: The different musical changes involved in cover songs, as discussed in section I, give us clear insights on which features to use. As chroma features have been evidenced to work quite well for this task [13], [15], [16] and proven to be better than timbre oriented descriptors as MFCC [7], [14], our approaches are based on HPCPs, given their usefulness for other tasks (e.g., key estimation) and their correspondence to pitch class distributions (see [25], [38] for a comparison with alternative approaches).

In section III-A, we have shown that HPCP resolution is important with both cosine and correlation distances. We have tested 12, 24, and 36-bin HPCPs with different variants of the methods presented in section II-B, and the results suggest that accuracy increases as the resolution does so. On the other hand, increasing resolution also increases computational costs, so that higher resolution is not considered. In addition, 36 seems to be a good resolution for key estimation [36] and structural analysis [51].

TABLE VII
F-MEASURE FOR DIFFERENT AVERAGING FACTORS AND CONSTRAINTS. DTW APPROACH EVALUATION WITH DB75

Alg. name	Constr. type	5	10	15	20	25	30	40
Sakoe-Chiba	Global	0.259	0.282	0.327	0.332	0.342	**0.355**	0.331
Itakura	Global	0.256	0.286	0.362	0.353	0.360	**0.395**	0.388
Simple DTW	No constr.	0.537	0.606	0.611	0.632	**0.638**	0.634	0.598
MyersT1	Local	0.647	**0.651**	0.641	0.643	0.624	0.625	0.577
MyersT2	Local	**0.651**	0.646	0.617	0.614	0.599	0.566	0.542

2) Similarity measure between features: In section III-B we have stated the importance of the similarity employed to compare chroma vectors. Furthermore, we have shown that using a similarity measure that is well correlated with cognitive foundations of musical pitch [46] improves substantially the final system accuracy. When using tonality descriptors, some papers do not specify how a local distance between these feature vectors is computed. They are supposed to assess chroma features' similarity as the rest of studies: with an euclidean-based distance. Since tonality features such as chroma vectors are proven not to be in an euclidean space [52], [53], [54], [55], this assumption seems to be wrong. Furthermore, any method (e.g., a classifier) using distances and concepts just valid for an euclidean space will have the same problem. This is an important issue that will be dealt in the proposed method (section IV).

3) Chroma transposition: To account for main key differences, one song is transposed to the tonality of the other one by means of computing a global HPCP for each song (section III-C) and circularly shifting by the OTI (equation 1). This technique has been proven to be more accurate than transposing the song to a reference key by means of a key estimation algorithm. In this case, the use of a less-than-perfect key extraction algorithm degrades the overall identification accuracy. Through the testing of two transposition variants we have pointed out the relevance this fact has in a cover song identification system or in a tonal alignment algorithm.

4) The use of beat tracking: We have seen that the DTW approach summarized in section II-B2 could lead to better results without beat tracking information (tables V and VII). Better results for DTW without beat tracking information were also found when comparing against the cross-correlation approach (which uses beat information). We can see this in table IX and in figure 8 (we also provide an extra comparative figure in a separate web page[8]). This is another fact that makes us disregard the use of 'intermediate' processes such as key estimation algorithms and beat tracking systems (citing the two that have been tested here), or chord and melody extraction engines. We feel that this can be a double-edged sword. Due to the fact that all these methods do not have a fully reliable performance[9], they may decrease the accuracy of a system comprising (at least) one of them. The same argument can be applied to any audio segmentation, chorus extraction, or summarization technique. We can also take a look at state-of-the-art approaches. For instance, common accuracy values for a chord recognition engine range from 75.5% [56] to 93.3% [57] depending on the method and the considered music material. Also, in this last case, once the chords are obtained, the approach to measure distances between them is still an unsolved issue, involving both some cognitive and musicological concepts that are not fully understood yet. So, errors in these 'intermediate' processes might be added (in case we are using more than one of them), and be propagated to the overall system's identification accuracy (the so called *weakest link* problem).

5) Alignment procedure: Several tests have been presented with chroma features DTW alignment. DTW allows us to restrict the alignment (or 'warping') paths to our requirements (section III-E). Consequently, we have tested four 'standard' constraints on these paths (two local and two global constraints). With global constraints we are not considering paths (or alignments) that might be far from the DTW matrix main diagonal. A problem arises when this path can represent a 'correct' alignment (as the example illustrated in figure 4). We have also seen that the accuracy decreases substantially with these constraints. As mentioned in section I, covers can substantially alter the song structure. When this happens, the 'correct' alignment between two covers of the same *canonical song* may be outside of the main DTW matrix diagonal. Therefore, the use of global constraints dramatically decreases the system detection accuracy. These two facts reveal the incorrectness of using a global alignment technique for cover song identification. Regarding local constraints, we have seen that these can help us by reducing 'pathological' warpings that arise when using a small *averaging factor* (table VII). Consequently, this allows us to use much detail in our analysis, and, therefore, to get a better accuracy.

Many systems for cover song identification use a global alignment technique such as DTW or entire song cross-correlation for determining similarity (except the ones that use a summarization, chorus extraction or segmentation technique, which would suffer from the problem of the 'weakest link', cited above). In our opinion, a system considering similarity between song subsequences, and thus, using a local similarity or alignment method, is the only way to cope with strong song structural changes.

IV. PROPOSED METHOD

In this section we present a novel method for cover song identification which tries to avoid all the weak points that conventional methods may have and which have been analyzed in previous section. The proposed method uses high-resolution HPCPs (36-bin) as these have been shown to lead to better accuracy (section III-A). To account for key transpositions, the

[8] http://mtg.upf.edu/∼jserra/chromabinsimappendix.html

[9] To account for accuracies of those systems you can visit, e.g., MIREX 2006 wiki page: http://www.music-ir.org/mirex/2006/index.php/Main_Page (Accessed 29 Jan. 2008)

OTI transposition method explained in section III-C is used instead of a conventional key finding algorithm. We avoid using any kind of 'intermediate' technique as key estimation, chord extraction or beat tracking, as these might degrade the final system identification accuracy (as discussed in section III-F). The method does not employ global constraints, and takes advantage of the improvement given by the local constraints explained in section III-E. Furthermore, it presents relevant differences in two important aspects that boost its accuracy in a dramatic way: it uses a new binary similarity function between chroma features (we have verified the relevance of distance measures in section III-B), and employs a novel local alignment method accounting for structural changes (considering similarity between subsequences, as discussed in section III-F).

A quite resemblant method to the one proposed here is [12]. In there, a chroma-based feature named Polyphonic Binary Feature Vector (PBFV) is adopted, which uses spectral peaks extraction and harmonics elimination. Then, the remaining spectral peaks are averaged across beats and collapsed to a 12-element binary feature vector. This results in a string vector for each analyzed song. Finally, a fast local string search method and a Dynamic Programming (DP) matching are evaluated. The method proposed here also extracts a chroma feature vector using only spectral peaks (HPCP, see section II-A), but we do not do beat averaging, which we find has a detrimental effect in the accuracy of DP algorithms such as Dynamic Time Warping (DTW) (section III-D). Another important difference to the proposed method is the similarity between vectors. In [12], this is computed between binarized vectors, while in the proposed method, what is binarized is the similarity measure, not the vectors themselves (equation 3). Finally, we also think that using an exhaustive alignment method like the one proposed in next section IV-A is also determinant for our final system identification accuracy.

A. System description

Figure 5 shows a general block diagram of the system. It comprises four main sequential modules: pre-processing, similarity matrix creation, dynamic programming local alignment (DPLA) and post-processing.

From each pair of compared songs A and B (inputs), we obtain a distance between them (output). Pre-processing comprises HPCP sequence extraction and a global HPCP averaging for each song. Then, one song is transposed to the key of the other one by means of an *Optimal Transposition Index* (OTI). From these two sequences, a binary similarity matrix is then 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.

1) Pre-processing: For each song, we extract a sequence of 36-bin HPCP feature vectors as made before, using the same parameters specified in section II-A. An averaging factor of 10 was used as it was found to work well in sections III-D and III-E. As we are using local constraints for the proposed method, it is not surprising to find a quite similar identification accuracy curve for different values of the averaging factor when comparing the proposed method with the locally constrained DTW algorithms explained in section III-E. In an electronic appendix to this article[10], the interested reader can find a figure showing the accuracy curves for the proposed method and for DTW with local constraints [45].

A global HPCP vector is computed by averaging all HPCPs in a sequence, and normalizing by its maximum value. With the global HPCPs of two songs ($\overrightarrow{h_A}$ and $\overrightarrow{h_B}$), we compute the OTI index, which represents the number of bins that an HPCP needs to circularly shift to have maximal resemblance to the other (see equation 1 in section III-C).

The last operation of the pre-processing block consists in transposing both musical pieces to a common key. This is simply done by circularly shifting each HPCP in the whole sequence of just one song by $OTI(\overrightarrow{h_A}, \overrightarrow{h_B})$ bins (remember we denote musical transposition by superscript ^{Tr}).

2) Similarity matrix: The next step is computing a similarity matrix S between the obtained pair of HPCP sequences. 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 features. First, as these features might not be in an euclidean space [46], 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). Second, using only two values to represent similarity, the possible paths through the similarity matrix become more evident, providing us with a clear notion of where the two sequences agree and where they mismatch (see figure 6 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. [32]. 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 to calculate a resemblance index for all possible transpositions. Then, if the transposition that leads to maximal similarity corresponds to less than a semitone (accounting for slight tuning differences), the two HPCP vectors are claimed to be the same. This idea can be formulated

[10] http://mtg.upf.edu/∼jserra/chromabinsimappendix.html

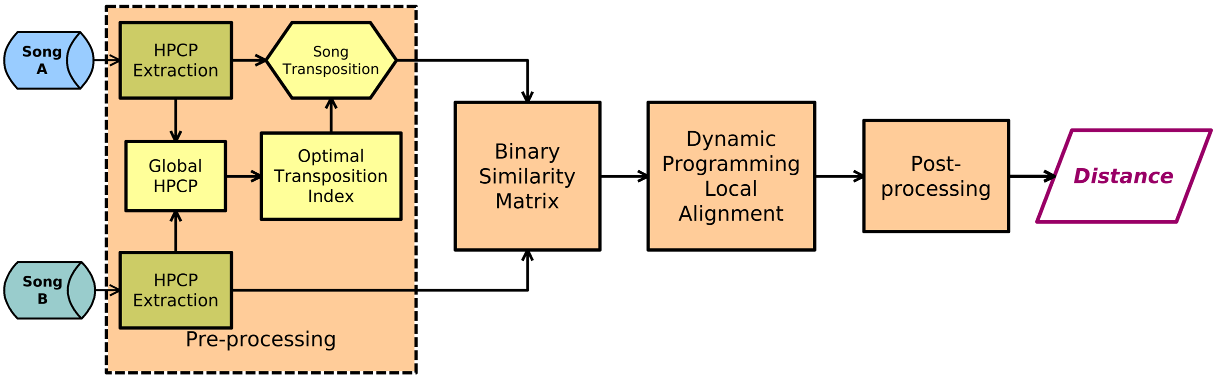

Fig. 5. General block diagram of the system.

in terms of the OTI explained in equation 1. So, as we are using a resolution of a $1/3$ of a semitone (36 bins), the binary similarity measure between the two vectors is then obtained by:

$$s(\overrightarrow{h^{Tr}_{A,i}},\overrightarrow{h_{B,j}}) = \begin{cases} \mu_+ & \text{if } OTI(\overrightarrow{h^{Tr}_{A,i}},\overrightarrow{h_{B,j}}) \in \{0,1,N_H\text{-}1\}, \\ \mu_- & \text{otherwise.} \end{cases} \tag{3}$$

where μ_+ and μ_- are two constants that indicate match or mismatch. These are usually set to a positive and a negative value (e.g., +1 and -1). Empirically, we found that a good choice for μ_+ and μ_- were +1 and -0.9 respectively. Ranges of μ_+ and μ_- between ± 0.7 and ± 1.25 resulted in changes smaller than an 5% of the evaluation measures tested. We show two examples of this type of similarity matrix in figure 6.

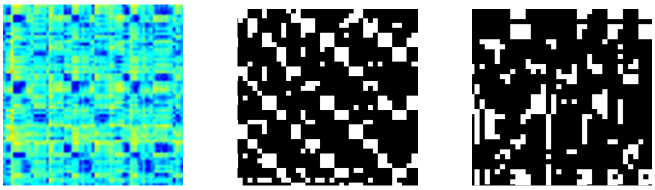

Fig. 6. Euclidean-based similarity matrix for two covers of the same song (left), OTI-based binary similarity matrix for the same covers (center) and OTI-based binary similarity matrix for two songs that do not share a common tonal progression (right). We can see diagonal white lines in the second plot, while this pattern does not exist in the third. Coordinate units in the horizontal and vertical axes correspond to 1 sec frames.

3) Dynamic programming local alignment (DPLA): A binary similarity matrix S is the only input to our DPLA algorithm. In section III-E 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 accuracy. Instead, the use of local constraints [50] can help us preventing ‘pathological warpings’ and just admitting certain ’logical’ tempo changes. Also, in section III-F, 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 [58] is a well-known algorithm for performing local sequence alignment in 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:

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + S_{i-1,j-1} - \delta(S_{i-2,j-2}, S_{i-1,j-1}) \\ H_{i-2,j-1} + S_{i-1,j-1} - \delta(S_{i-3,j-2}, S_{i-1,j-1}) \\ H_{i-1,j-2} + S_{i-1,j-1} - \delta(S_{i-2,j-3}, S_{i-1,j-1}) \\ 0 \end{cases} \tag{4}$$

for $4 \le i \le n{+}1$ and $4 \le j \le m{+}1$. Each $S_{i,j}$ corresponds to the value of the binary similarity matrix S at element (i,j), and $\delta()$ 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 $S_{i-1,j-1} \le 0$. More concretely:

$$\delta(a,b) = \begin{cases} 0 & \text{if } b > 0 \text{ (no gap)} \\ c_1 & \text{if } b \le 0 \text{ and } a > 0 \text{ (gap opening)} \\ c_2 & \text{if } b \le 0 \text{ and } a \le b \text{ (gap extension)} \end{cases} \tag{5}$$

Good values were empirically found to be $c_1 = 0.5$ for a gap opening, and $c_2 = 0.7$ for a gap extension. Small variability of the evaluation measures was shown for c_1, c_2 values between 0.3 and 1. We used the songs in DB90 for empirically estimating these parameters and then evaluated the method with DB2053 (see section IV-B).

Values of H can be interpreted considering 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 $\overrightarrow{h_{B,j-1}}$. The first 3 rows and columns of H can be initialized to have a 0 value.

An example of the resultant matrix H is shown in figure 7. We clearly observe 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 $H_{400,100}$, where sub-indices respectively denote rows and columns).

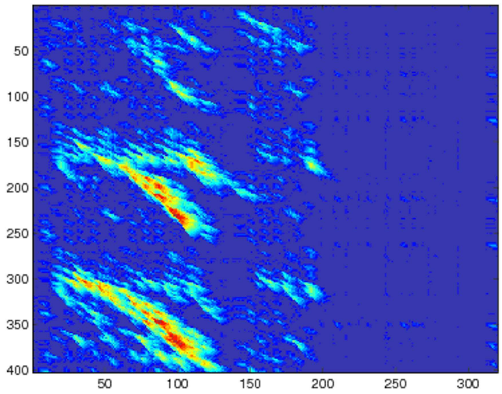

Fig. 7. Example of a local alignment matrix H between two covers. It can be seen that the two songs do not entirely coincide (just in two fragments), and that, mainly, their respective second halves are completely different. Coordinate units in the horizontal and vertical axes correspond to 1 sec averaging across frames.

4) 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:

$$Score(HPCP_A^{Tr}, HPCP_B) = \max\{H_{i,j}\} \quad (6)$$

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

Finally, to obtain a dissimilarity value that is independent of song duration, the score is normalized by the compared song lengths [45] and the inverse is taken:

$$d(song_A, song_B) = \frac{n+m}{Score(HPCP_A^{Tr}, HPCP_B)} \quad (7)$$

where n and m are the respective lengths for songs A and B.

B. Evaluation

We now display the results corresponding to the evaluation of our method. This has been made with the music collection presented in section II-C and within the framework of the MIREX 2008 Audio Cover Song Identification contest as well. As the databases used in this part of the paper may have more than 5 covers per set, the first 10 retrieved items were considered for evaluation.

Firstly, as we have proposed a new distance measure between chroma features, we provide results for a comparison between common distance measures and the proposed OTI-based binary distance in table VIII. To perform this comparison, we have thresholded common distance measures and applied the same DPLA algorithm (with the same parameters) to all of them. Several thresholds were tested for each distance in order to determine the ones leading to best identification accuracy. We observe that OTI-based binary similarity matrix outperforms other binary similarity matrices obtained through thresholding common similarity measures between chroma features. In the case of these last measures, best identification accuracy values for different thresholds tested are shown.

TABLE VIII
IDENTIFICATION ACCURACY FOR DPLA ALGORITHM WITH 5 DIFFERENT BINARY SIMILARITY MATRICES AS INPUT. EVALUATION DONE WITH DB2053

Distance used	F-measure	R_{10}
Dot product	0.132	0.136
Euclidean distance	0.218	0.216
Cosine similarity	0.221	0.219
Correlation	0.239	0.247
OTI-based similarity	0.601	0.576

We next show the general evaluation results corresponding to our personal music collection. Within these, we compare identification accuracy between the proposed method and the best variants of the cross-correlation and DTW methods tested. In table IX we report the F-measure values for the three different databases presented. Recall is shown in figure 8. In there, we plot an average Recall figure for all the implemented systems (best variants). Vertical axis represents Recall and horizontal axis represents different percentages of the retrieved answer. As this was set to a maximum length of 10, the numbers represent 0 answers (giving a Recall of 0), 1 answer, 2 answers and so forth. We can see that with the newly proposed method the accuracy is around 58% of correctly retrieved songs within the first 10 retrieved answers. This value is highly superior to the accuracies achieved for the best versions of the cross-correlation and DTW methods that we could implement (around 20 and 40 percent respectively), and is very far from the the baseline corresponding to just guessing by chance, which is lower than 0.3%.

TABLE IX
F-MEASURE FOR THE PROPOSED METHOD, THE DTW AND THE CROSS-CORRELATION APPROACHES. PARAMETERS FOR THE CROSS-CORRELATION AND THE DTW METHODS WERE ADJUSTED ACCORDING TO THE BEST VALUES AND VARIANTS FOUND IN SECTION III

Method	DB75	DB330	DB2053
Cross-correlation	0.638	0.348	0.169
DTW	0.651	0.485	0.399
Proposed method	0.868	0.688	0.601

If we take a look to MIREX 2007 contest data (where we participated with this algorithm), we observe that our system was the best performing one with a substantial difference to others [59]. A total of 8 different algorithms were presented to the MIREX 2007 Audio Cover Song task. Table X shows the overall summary results obtained[11]. The present algorithm (SG, first column) performed the best in all considered evaluation measures, 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.

[11] See the complete results and details about the evaluation procedure at http://www.music-ir.org/mirex/2007/index.php/Audio_Cover_Song_Identification_Results (Accessed 29 Jan. 2008)

TABLE X

RESULTS FOR MIREX 2007 AUDIO COVER SONG TASK. ACCURACY MEASURES EMPLOYED WERE THE TOTAL NUMBER OF COVERS IDENTIFIED WITHIN THE FIRST 10 ANSWERS (TNCI_{10}), THE MEAN NUMBER OF COVERS IDENTIFIED WITHIN THE 10 FIRST ANSWERS (MNCI_{10}), THE MEAN OF AVERAGE PRECISION (MAP) AND THE AVERAGE RANK OF THE FIRST CORRECTLY IDENTIFIED COVER (RANK_1). CLOCK TIME MEASURES ARE REPORTED ON THE LAST LINE OF THE TABLE (NUMBER OF USED THREADS IN BRACKETS). VALUES FOR THE ALGORITHM PRESENTED HERE ARE SHOWN IN THE FIRST COLUMN (SG)

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)

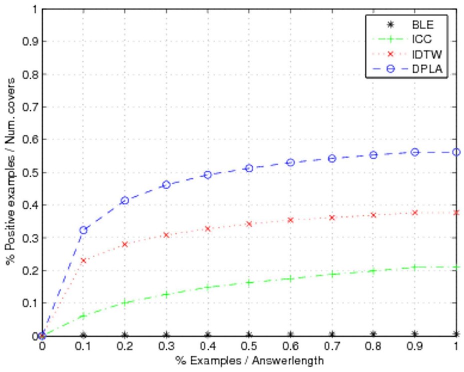

Fig. 8. Average Recall figures comparing the proposed approach (blue circles) with the cross-correlation (green sum signs) and the DTW (red crosses) methods for DB2053. Parameters for the cross-correlation and the DTW methods compared were adjusted according to the best values found in section III. A base-line identification accuracy (BLE) is also plotted (black bottom asterisks).

Furthermore, the next best performing system reached and an $MNCI_{10}$ of 3.658 and a *MAP* of 0.330, which represents a substantial difference to the one proposed in this paper (57.88% superior in terms of *MAP*). In addition, statistical significance tests showed that the results for the system were significantly better than those of the other six systems presented in the contest.

A basic error analysis [45] shows that the best identified covers are “A forest”, originally performed by The Cure and “Let it be”, originally performed by The Beatles. Other correctly classified items are “Yesterday”, “Dont let me down” and “We can work it out”, all originally performed by The Beatles and “How insensitive” (Vinicius de Moraes). This high amount of Beatles’ songs within the better classified items can be due to the fact that there were many Beatles’ cover sets (e.g., 14 out of 30 in DB330), but it can also be justified considering the clear simplicity and definition of their tonal progressions, that, in comparison with other more elaborated pieces (e.g., “Over the rainbow” performed by Judy Garland), leads to better identification. Within this set of better identified covers there are several examples of structural changes and tempo deviations. In the electronic appendix[12], we provide a confusion matrix with labels corresponding to cover sets (rows and columns).

We detected that there were some songs, such as “Eleanor Rigby” and “Get Back”, that caused ‘confusion’ more or less with all the queries made. One explanation for this might be that these two songs are built over a very simple chord progression involving just two chords: the tonic and the mediant (e.g., C and Em for a C major key) for the former, and the tonic and the subdominant (e.g., C and F for a C major key) for the latter. So, as they rely half of the time in the tonic chord, any song being compared to them will share half of the tonal progression. Other poorly classified items are “The battle of Epping forest” (Genesis) or “Stairway to heaven” (Led Zeppelin). Checking their wrongly associated covers, we find that, most of the time, the alignment, the similarity measure and the transposition are performing correctly according to the features extracted. Thus, we have the intuition that the tonal progression might not be enough for some kinds of covers. This does not mean that HPCPs could be sensitive to timbre or other facets of the musical pieces. On the contrary, we are able to detect many covers that have a radical change in the instrumentation, which we think it is due to the capacity of HPCPs to filter timbre out.

An interesting misclassification appears with “No woman no cry”, originally performed by Bob Marley. These covers are associated more than 1/3 of the times with the song “Let it be” (The Beatles). When we analyzed the harmonic progression of both songs, we discovered that they share the same chords in different parts of the theme (C - G - Am - F). Thus, this might be a logical misclassification using chroma features. Another source of frequent confusion is the classical harmonic progression I - IV - I or I - V - IV - I, which many songs share.

V. CONCLUSIONS

In this paper we have devised a new method for audio signal comparison focused on cover song identification that by large outperforms state-of-the-art system. This has been achieved after experimenting with many proposed techniques and variants, and testing their effect in final identification accuracy, which also was one of the main objectives in writting this article.

We have first presented our test framework and the two state-of-the-art methods that we have used in further experiments. The performed analysis has focused on several variants that could be taken for these two methods (and, in general, for any method based on chroma descriptors): (a) the chroma features resolution - section III-A; (b) the local cost function (dissimilarity measure) between chroma features - section

[12] http://mtg.upf.edu/∼jserra/chromabinsimappendix.html

III-B; (c) the effect of using key transposition methods - section III-C; and (d) the use of a beat tracking algorithm to obtain a tempo-independent representation of the chroma sequence - section III-D. In addition, as DTW is a well known and extensively used technique, we tested two underexplored variants of it, apart from the simple one mentioned in section II-B2: DTW with global and with local constraints (section III-E). The results of these cross-validated experiments have been summarized in section III-F.

Finally, we have presented a new cover song identification system that takes advantage of the results found and that has been proven, using different evaluation measures and contexts, to work significantly better than other state-of-the-art methods. 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 work done and the method presented here represent an important milestone.

Acknowledgements

The authors would like to thank their colleagues and staff at the Music Technology Group (UPF) for their support and encouragement, especially Graham Coleman for his review and proofreading. Furthermore, the authors wish to thank the anonymous reviewers for very helpful comments.

References

[1] R. Witmer and A. Marks, *'Cover', Grove Music Online*, L. Macy, Ed. Oxford University Press, 2006, (Accessed 25 Oct. 2007), http://www.grovemusic.com.

[2] S. Strunk, *'Harmony', Grove Music Online*, L. Macy, Ed. Oxford University Press, 2006, (Accessed 26 Nov. 2007), http://www.grovemusic.com.

[3] S. Dalla Bella, I. Peretz, and N. Aronoff, "Time course of melody recognition: A gating paradigm study," *Perception and Psychophysics*, vol. 7, no. 65, pp. 1019–1028, 2003.

[4] M. D. Schulkind, R. J. Posner, and D. C. Rubin, "Musical features that facilitate melody identification: How do you know it's your song when they finally play it?" *Music Perception*, vol. 21, no. 2, pp. 217–249, 2003.

[5] N. Hu, R. B. Dannenberg, and G. Tzanetakis, "Polyphonic audio matching and alignment for music retrieval," *IEEE Workshop on Apps. of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 185–188, 2003.

[6] N. H. Adams, N. A. Bartsch, J. B. Shifrin, and G. H. Wakefield, "Time series alignment for music information retrieval," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 303–310, 2004.

[7] M. Casey and M. Slaney, "The importance of sequences in musical similarity," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5, pp. V–V, May 2006.

[8] 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," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 183–190, 2005.

[9] M. Marolt, "A mid-level melody-based representation for calculating audio similarity," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 280–285, 2006.

[10] Ö. Izmirli, "Tonal similarity from audio using a template based attractor model," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 540–545, 2005.

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

[12] H. Nagano, K. Kashino, and H. Murase, "Fast music retrieval using polyphonic binary feature vectors," *IEEE Int. Conf. on Multimedia and Expo (ICME)*, vol. 1, pp. 101–104, 2002.

[13] M. Müller, F. Kurth, and M. Clausen, "Audio matching via chroma-based statistical features," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 288–295, 2005.

[14] M. Casey and M. Slaney, "Song intersection by approximate nearest neighbor search," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 144–149, October 2006.

[15] 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.

[16] D. P. W. Ellis and G. E. Poliner, "Identifying cover songs with chroma features and dynamic programming beat tracking," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, pp. 1429–1432, April 2007.

[17] A. Klapuri, "Signal processing methods for the automatic transcription of music," Ph.D. dissertation, Tampere University of Technology, Finland, April 2004.

[18] M. Goto, "A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Comunication*, vol. 43, no. 4, pp. 311–329, September 2004.

[19] G. E. Poliner, D. P. W. Ellis, A. Ehmann, E. Gómez, S. Streich, and B. S. Ong, "Melody transcription from music audio: approaches and evaluation," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, pp. 1247–1256, 2007.

[20] A. Sheh and D. P. W. Ellis, "Chord segmentation and recognition using em-trained hidden markov models," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 183–189, 2003.

[21] C. A. Harte and M. B. Sandler, "Automatic chord identification using a quantized chromagram," *Conv. of the Audio Engineering Society (AES)*, pp. 28–31, 2005.

[22] T. Fujishima, "Realtime chord recognition of musical sound: a system using common lisp music," *Int. Computer Music Conference (ICMC)*, pp. 464–467, 1999.

[23] G. Tzanetakis, "Pitch histograms in audio and symbolic music information retrieval," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 31–38, 2002.

[24] S. Paws, "Musical key extraction from audio," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 96–99, 2004.

[25] E. Gómez, "Tonal description of music audio signals," Ph.D. dissertation, MTG, Universitat Pompeu Fabra, Barcelona, Spain, 2006, http://mtg.upf.edu/ egomez/thesis.

[26] R. B. Dannenberg and N. Hu, "Pattern discovery techniques for music audio," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 63–70, 2002.

[27] N. A. Bartsch and G. H. Wakefield, "To catch a chorus: using chroma-based representations for audio thumbnailing," *IEEE Workshop on Apps. of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 15–18, 2001.

[28] M. Goto, "A chorus-section detection method for musical audio signals and its application to a music listening station," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1783–1794, September 2006.

[29] M. Müller, *Information Retrieval for Music and Motion*. Springer, 2007.

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

[31] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, pp. 707–710, 1966.

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

[33] P. Cano, M. Kaltenbrunner, O. Mayor, and E. Batlle, "Statistical significance in song-spotting in audio," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 77–79, 2001.

[34] R. L. Kline and E. P. Glinert, "Approximate matching algorithms for music information retrieval using vocal input," *ACM Multimedia*, pp. 130–139, 2003.

[35] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," *Very Large Databases Journal*, pp. 518–529, 1999.

[36] E. Gómez and P. Herrera, "Estimating the tonality of polyphonic audio files: cognitive versus machine learning modelling strategies," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 92–95, 2004.

[37] ——, "The song remains the same: identifying versions of the same song using tonal descriptors," *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 180–185, 2006.

[38] B. S. Ong, E. Gómez, and S. Streich, "Automatic extraction of musical structure using pitch class distribution features," *Workshop on Learning the Semantics of Audio Signals (LSAS)*, pp. 53–65, 2006.

[39] H. Purwins, “Profiles of pitch classes. circularity of relative pitch and key: experiments, models, computational music analysis, and perspectives.” Ph.D. dissertation, Berlin University of Technology, Germany, 2005.

[40] D. Huron, “Scores from the ohio state university cognitive and systemathic musicology laboratory - bach well-tempered clavier fugues, book ii,” *Online: http://kern.ccarh.org/cgi-bin/ksbrowse?l=/osu/classical/bach/wtc-2*, 1994, (Last access Jan. 2008).

[41] M. E. P. Davies and P. Brossier, “Beat tracking towards automatic musical accompainment,” *Conv. of the Audio Engineering Society (AES)*, May 2005.

[42] P. Brossier, “Automatic annotation of musical audio for interactive applications,” Ph.D. dissertation, Queen Mary, London, 2007.

[43] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. ACM Press Books, 1999.

[44] J. Serrà, “A qualitative assessment of measures for the evaluation of a cover song identification system,” *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 319–322, September 2007.

[45] ——, “Music similarity based on sequences of descriptors: tonal features applied to audio cover song identification,” Master’s thesis, MTG, Universitat Pompeu Fabra, Barcelona, Spain, 2007.

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

[47] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequences of two proteins,” *Journal of Molecular Biology*, vol. 48, pp. 443–453, 1970.

[48] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimisation for spoken word recognition,” *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 26, pp. 43–49, 1978.

[49] F. Itakura, “Minimum prediction residual principle applied to speech recognition,” *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 23, pp. 52–72, 1975.

[50] C. Myers, “A comparative study of several dynamic time warping algorithms for speech recognition,” Master’s thesis, Massachussets Institute of Technology (MIT), USA, 1980.

[51] B. S. Ong, “Structural analysis and segmentation of music signals,” Ph.D. dissertation, MTG, Universitat Pompeu Fabra, Barcelona, Spain, 2007.

[52] R. N. Shepard, “Structural representations of musical pitch,” *The Psychology of Music*, 1982.

[53] D. Lewis, *Generalized musical intervals and transformations*. Newhaven: Yale University Press, 1987.

[54] R. Cohn, “Neo-riemannian operations, parsimonious trichords, and their tonnetz representations,” *Journal of Music Theory*, vol. 1, no. 41, pp. 1–66, 1997.

[55] E. Chew, “Towards a mathematical model of tonality,” Ph.D. dissertation, Massachussets Institute of Technology (MIT), USA, 2000.

[56] J. P. Bello and J. Pickens, “A robust mid-level representation for harmonic content in music signals,” *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 304–311, 2005.

[57] K. Lee and M. Slaney, “Automatic chord recognition using an hmm with supervised learning,” *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 133–137, 2006.

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

[59] J. Serrà and E. Gómez, “A cover song identification system based on sequences of tonal descriptors,” *MIREX extended abstract*, 2007.

Joan Serrà obtained both the degrees of Telecommunications and Electronics (sound and image specialization) at Enginyeria la Salle, Universitat Ramón Llull (URL), Barcelona, Spain, in 2002 and 2004 respectively. Further studies were focused in digital audio signal processing and audio recording and engineering (Audiovisual Technologies department, URL). After that, he worked in the R+D department of Music Intelligence Solutions Inc., developing patented approaches to music and visual media discovery. In 2006-07, he did the MSc in Information, Communication and Audiovisual Media Technologies (TICMA) at Universitat Pompeu Fabra (UPF), Barcelona, Spain. He is currently a researcher and a PhD student at the Music Technology Group (MTG) of the UPF. He has also been a semi-professional musician for more than 10 years. His research interests include (but are not limited to): machine learning, music perception and cognition, time series analysis, signal processing, data visualization, dimensionality reduction and information retrieval.

Emilia Gómez is a post-doctoral researcher at the Music Technology Group (MTG) of the Universitat Pompeu Fabra (UPF). She graduated as a Telecommunication Engineer specialized in Signal Processing at the Universidad de Sevilla. Then, she received a DEA in Acoustics, Signal Processing and Computer Science applied to Music (ATIAM) at the IRCAM, Paris. In July 2006, she completed her PhD in Computer Science and Digital Communication at the UPF, on the topic of Tonal Description of Music Audio Signals. During her doctoral studies, she was a visiting researcher at the Signal and Image Processing (TSI) group of the École National Supérieure de Télécommunications (ENST), Paris and at the Music Acoustics Group (TMH) of the Stockholm Institute of Technology, KTH. She has been involved in several research projects funded by the European Commission and the Spanish Ministry of Science and Technology. She also belongs to the Department of Sonology, Higher Music School of Catalonia (ESMUC), where she teaches music acoustics and sound synthesis and processing. Her main research interests are related to music content processing, focusing on melodic and tonal facets, music information retrieval and computational musicology.

Perfecto Herrera received the degree in Psychology from the University of Barcelona, Spain, in 1987. He was with the University of Barcelona as a Software Developer and an Assistant Professor. His further studies have focused on sound engineering, audio postproduction, and computer music. Now he is finishing his PhD on Music Content Processing in the Universitat Pompeu Fabra (UPF), Barcelona. He has been working in the Music Technology Group, (UPF) since its inception in 1996, first as the person responsible for the sound laboratory/studio, then as a Researcher. He worked in the MPEG-7 standardization initiative from 1999 to 2001. Then, he collaborated in the EU-IST-funded CUIDADO project, contributing to the research and development of tools for indexing and retrieving music and sound collections. This work was somehow continued and expanded as Scientific Coordinator for the Semantic Interaction with Music Audio Contents (SIMAC) project, again funded by the EU-IST. He is currently the Head of the Department of Sonology, Higher Music School of Catalonia (ESMUC), where he teaches music technology and psychoacoustics. His main research interests are music content processing, classification, and music perception and cognition.

Xavier Serra (Barcelona, 1959) is the head of the Music Technology Group of the Universitat Pompeu Fabra in Barcelona. After a multidisciplinary academic education he obtained a PhD in Computer Music from Stanford University in 1989 with a dissertation on the spectral processing of musical sounds that is considered a key reference in the field. His research interests are on the understanding, modeling and generating music through computational approaches. He tries to find a balance between basic and applied research with methodologies from both scientific/technological and humanistic/artistic disciplines. Dr. Serra is very active in promoting initiatives in the field of Sound and Music Computing at the international level, being editor and reviewer of a number of international journals, conferences and programs of the European Commission, and giving lectures on current and future challenges of the field. He is the principal investigator of more than 10 major research projects funded by public and private institutions, the author of 31 patents and has published more than 40 research articles.