An Efficient Identification Methodology for Improved Access to Music Heritage Collections

Nicola Montecchio Emanuele Di Buccio
Department of Information Engineering, University of Padua, Italy
Email: {nicola.montecchio,emanuele.dibuccio}@dei.unipd.it

Nicola Orio
Department of Cultural Heritage, University of Padua, Italy
Email: nicola.orio@unipd.it

Abstract—A comprehensive methodology for automatic music identification is presented. The main application of the proposed approach is to provide tools to enrich and validate the descriptors of recordings digitized by a sound archive institution. Experimentation has been carried out on three different datasets, including a collection of digitized vinyl discs, although the methodology is not linked to a particular recording carrier. Automatic identification allows a music digital library to retrieve metadata about music works even if the information was incomplete or missing at the time of the acquisition. Automatic segmentation of digitized material is obtained as a byproduct of identification, allowing the music digital library to grant access to individual tracks, even if discs are digitized using a single file for a complete disc side. Results show that the approach is both efficient and effective.

Index Terms—Digital Libraries; Feature Extraction and Representation; Audio Processing

I. INTRODUCTION

In recent decades many sound archive institutions started to transfer their collections of analog recordings into digital form. To this end, a significant effort has been made at the international level (in particular by the European Community through funding of preservation initiatives) to digitize significant amounts of music and speech recorded in discs, tapes and other carriers. The digitization of the audio content however represents only a single, although crucial, aspect in the transformation process of a physical sound archive into a music digital library. In particular, having lost their connection with the physical support, digital objects require suitable cataloging values that allow users to retrieve and access them.

Let us consider the common case of music archives containing hundreds of vinyl discs in the form of 33rpm LPs. A complete digitization process would require annotating all the individual tracks for each side of every disc; metadata for each track should then be made available, ideally in structured form, about title, author, performers and other relevant information about the music work.

A digitization process requires several competences to be exploited at the same time: technical expertise for taking care of the digital transfer, cataloging expertise to extract relevant information from record covers and possibly accompanying booklets, and musicological expertise to segment the digitized audio and match individual tracks with corresponding cataloging values. The extraction of correct cataloging values from record covers is a complex task because the intended audience usually is not interested, nor has specific competences, in music metadata.

Several digitization campaigns have been carried out paying significantly more attention to the acquisition process than to the creation of consistent metadata and to the development of suitable tools for enabling direct access to individual tracks. This is particularly true for smaller sound archives, which do not have the resources to take care of the complete process, but often applies to larger archives as well. For instance, both the Fonoteca of the University of Alicante (Spain) – from now on simply Fonoteca – and the Discoteca di Stato of Rome (Italy), with which the authors have been in contact for the development of the methodology presented herein, decided not to segment LP sides in individual tracks because of the lack of funding for this particular task. Moreover, it is difficult to verify the correctness of metadata, because this requires listening to the complete collections that contain thousands of hours of recorded material.

An example of problems that arise because of imprecise or missing metadata is given in Figure 1, where the cover and the transcribed metadata of an LP are presented as they are contained in the Fonoteca. It can be seen that, although the composer Gabriel Faure was mentioned on the cover, only Maurice Ravel is reported in the metadata regarding the author of the works. Only a pre-existing familiarity with Ravel’s Trio may help a user realize that side B contains the work of another composer. This example illustrates a common situation in classical music albums, where there is often an emphasis on performers and on very well known music works.

The methodology reported in this paper addresses issues related to music access and retrieval, with particular attention to music digital libraries created from analogue sound archives. The goal is to identify automatically the music content of the digitized material and, at the same time, to segment the audio files into single tracks. Identification is carried out through the automatic matching of the digitized material with a set of digital recordings for which reliable metadata are available. The focus is on
II. RELATED WORK

A typical approach to automatic music identification is based on the extraction of an audio fingerprint from digital recordings. A fingerprint is a compact set of music features that allows for the identification of digital copies even in the presence of noise, distortion, and compression; it can be seen as a content-based signature that summarizes an audio recording [2]. Commercial applications of audio fingerprinting are available, from systems that track music usage in radio broadcasts to Web-based services that, given few seconds of an unknown recording, provide the users with relevant metadata [3]. Audio fingerprinting approaches are designed to identify a particular recording and are usually unable to generalize audio features to different performances of the same music work. However, in the particular case of classical music, which in Europe is tightly related to cultural heritage, the same music works could have hundreds of different recordings, and it is not feasible to collect all of them to create a different fingerprint for each one.

The more general task of identifying different versions of the same work is often based on the extraction of a sequence of audio features that are related to high level characteristics of music works – melody, harmony, rhythm – and their automatic alignment using well-known techniques such as Hidden Markov Models (HMMs) and Dynamic Time Warping (DTW). Chroma vectors, introduced in [4] to capture the harmonic content of a signal as explained in Section IV, are probably the most popular audio feature in the literature on music identification.

For instance, quantization of chroma vectors is proposed in [5] to obtain an index-based matching procedure, aimed at efficient music identification. Results showed that the approach is robust to typical variations due to different interpretation of classical music pieces. Chroma vectors are also exploited for music identification for the pop and rock music genres, although in this case the task is usually addressed as cover identification. In [6] chroma vectors are synthesized from MIDI files while in [7] chroma with different resolutions are tested, together with different similarity measures between them. In both cases, the similarity between pairs of recordings is computed through DTW. A chroma-based cover identification sys-
tem is proposed also in [8], which also provided the Cover80 collection, one of the first publicly available benchmarks for this task.

The identification of classical music has been part of the task on cover identification of the Music Information Retrieval Evaluation Exchange [9] campaigns since 2009. The test collection used for classical music was based on a number of Chopin’s mazurkas, collected within the Mazurka Project\(^1\). Presented approaches in this case are also often based on chroma alignment. In [10], a methodology combining chroma descriptors indexing and HMM-based identification was proposed to identify short excerpts of classical music. Index-based retrieval was exploited to efficiently extract a small subset of potential candidates from the collection to be matched with the query through an audio-to-audio matching approach based on [11].

A different set of features has been proposed by [12], although the goal was the online alignment of different versions of classical music pieces based on DTW. In this case, and similarly to the approach for audio hashing proposed in [13], music is described by the signal energy contained in a number of predefined frequency bands. To this end, it is interesting to note that a major difference between audio fingerprint and music identification is the modeling of time variations, which is considered unnecessary in the former because of the assumed constant speed of actual players and is required in the latter because different performances may have large variants in tempo.

In this paper, we propose an approach to cover identification that aims at efficiency. The initial idea is based on the concept of Locality Sensitive Hashing (LSH) [14], which is a general approach to handle high dimensional spaces by using ad-hoc hashing functions to create collisions between vectors that are close in the high dimensional space. LSH has been applied to efficient search in music collections [15] and in different media [16]. The application of LSH allows us to exploit indexing techniques to speed up the identification process, although in this case the information on the temporal evolution of the music signal is not taken into account.

III. OVERVIEW OF THE METHODOLOGY

In this work the problem of music identification is addressed by way of a comprehensive methodology that includes: an audio content analysis procedure to extract audio content descriptors, the computation of similarity among audio documents based on a *bag of feature* representation, and a further refinement of the identification process by way of an audio alignment algorithm.

The *audio content analysis* is based on a two-level representation. At the first level we consider a music recording as a sequence of shorter elements that can be useful for capturing a possible structure in a music work, due for example to repetitions of a theme or modulations. Since automatic identification of such constituting elements might introduce segmentation errors, we exploit a simplified approach where a recording is divided into overlapping excerpts of fixed length. Content-based descriptors are then extracted from each of the resulting segments, obtaining a second representation level. The basic idea is to represent the audio signal corresponding to a segment by means of musically meaningful units, e.g. notes. Since the extraction of such units is prone to errors that might affect the accuracy of the overall methodology, the adopted approach consists in computing content-based descriptors at equally spaced time intervals.

In line with most of the approaches presented in Section II the adopted descriptors are *chroma features*, 12-dimensional vectors representing the energy associated to each pitch class in a short time frame. Chroma extraction is preceded by *tuning frequency adjustment* in order to be robust to adoption of different reference frequencies, and is followed by a *key finding* procedure, which allows us to deal with different versions of the same music work performed in different keys. The choice of key invariant chroma features as content descriptors is based on the consideration that listeners use mostly harmonic and melodic cues to decide whether two recordings are from the same music work. Differences in tonality, tempo and duration do not substantially affect the similarity judgment and thus our representation aims at being robust to changes in these music dimensions.

The resulting chroma vectors are not directly exploited as music descriptors to perform identification. Each chroma vector is indeed transformed in a single integer value by means of a hashing function which is based on the rank of the elements of chroma vectors. The integer value corresponding to a chroma vector depends on the energy distribution among the pitch classes.

The representation steps map a recording in a sequence of segments, each of them being represented as a sequence of hashes. Both the recording to be identified, hereafter named *query*, and the recordings in the collection, hereafter named *documents*, undergo the same three-step representation process. A similarity function is subsequently applied to each query-document pair, where documents are ranked in decreasing order of the similarity score assigned by the function. The basic intuition behind our similarity function is that similar recordings present a similar energy distribution among pitch classes, and therefore have many hashes in common.

The bag of words simplification can negatively affect the identification accuracy. Therefore, our methodology is based on a two-phase identification process. The first phase relies on the above bag of feature paradigm and aims at obtaining a first results list where candidate documents are ranked at the top \(k\) positions; ranking is then refined in a second phase through an audio alignment technique based on particle filtering that is performed only on the top ranked results; Section VI describes in detail the proposed alignment technique. This hybrid approach allows us to speed up the identification process with respect to the sole application of audio alignment, which

\(^1\)http://mazurka.org.uk/
is computationally more expensive.

The methodology is depicted in Figure 2. Each of the following sections provides an in-depth description of the blocks constituting the complete approach.

IV. AUDIO ANALYSIS

Western music is characterized by the use of a 12 tone scale, in which the ideal note frequencies are logarithmically spaced, each being 12\sqrt{2} times the previous one (assuming an equal temperament); this structure defines a family of scales, of which each member is uniquely identified by a reference frequency assigned to one of its notes. Although setting the A above middle C to 440 Hz is identified by a reference frequency assigned to one of its notes, a family of scales, of which each member is uniquely identified by a reference frequency assigned to one of its notes. Although setting the A above middle C to 440 Hz is identified by a reference frequency assigned to one of its notes.

Even though a limited offset of the tuning frequency from the standard 440 Hz does not lead to significant problems in typical MIR systems – a perfect tuning of each note is unrealistic for the human voice or any acoustic instrument – the issue is critical in those cases where the reference frequency is so far from the standard that a listener cannot decide whether notes should be classified according to a certain pitch class or to the one immediately above/below.

In the literature the problem has not been dealt with extensively, with the exception of [18], [19]. Below we report our algorithm for estimating the tuning frequency, and evaluate its performance by considering the improvement in identification accuracy with respect to a baseline.

The tuning frequency detection algorithm works by first filtering the input signal according to different reference frequencies, which are then interpolated in order to produce a final estimation.

In particular, from a semitone-wide interval, centered on 440 Hz, an even number \( m \) of reference tuning frequencies \( f_1 \ldots f_m \), equally spaced on a logarithmic scale, is chosen (see Figure 4(a)); with respect to each of these frequencies, the input audio content is filtered according to a bank of narrow, semitone-spaced bandpass filters, and the total energies of the resulting filtered signals are summed in order to get, for each candidate tuning frequency \( f_i \), a measure of “fitness” to the input audio content \( e_i \) (Figure 4(b)). The underlying motivation is that in a realistic performance, in which most notes are played in tune, the output energies of the filter banks that are centered nearest to the actual tuning frequency should exhibit the highest values.

The estimation of the reference tuning frequency makes use of circular statistics, and in particular of the von Mises distribution (or circular normal distribution) on the circle. The pdf for the distribution on the angle \( \theta \) is:

\[
p(\theta | \mu, \kappa) = \frac{\exp(\kappa \cos(\theta - \mu))}{2\pi I_0(\kappa)} \quad \theta \in [0, 2\pi) \tag{1}
\]

where \( \mu \in [0, 2\pi) \) and \( \kappa \geq 0 \) are, respectively, analogous to the center and the variance of the normal distribution, and \( I_0(\cdot) \) is the Bessel function of order 0.

The choice of the von Mises distribution is motivated by the assumption that an offset of an integer number of semitones in the initial \( m \) frequencies should not alter the resulting estimation (Figure 4(c)) – thus the need for a periodic distribution. In order to fit into this framework, the frequencies \( f_1 \ldots f_m \) are mapped to the angles \( \theta_1 \ldots \theta_m \) in the \([-\pi, \pi]\) range (Figure 4(d)). Considering an integer quantization \( \hat{e}_i \) for the energy \( e_i \) of a filtered signal as the number of realizations of \( \theta_i \), a maximum likelihood estimation can be performed to get the most probable tuning frequency (the mean \( \mu \) of the distribution, which is also the mode) and a measure of confidence (a higher \( \kappa \) parameter corresponds to a sharper distribution, thus to a higher confidence). In particular, let \( \hat{N} = \sum_{i=1}^{m} \hat{e}_i \) (i.e. \( \hat{N} \) corresponds to the total number of samples), then

\[
\hat{\mu} = \text{Arg} \left( \sum_{i=1}^{m} \hat{e}_i \exp(j\theta_i) \right) \tag{2}
\]

\[
\frac{I_0(\hat{\kappa})}{I_0(\kappa)} = \frac{1}{\hat{N}} \sum_{i=1}^{m} \hat{e}_i \cos(\theta_i - \hat{\mu}) \tag{3}
\]

Taking the limit of the quantization step to 0 and normalizing the energy values with respect to their sum, the number of samples \( \hat{e}_1 \ldots \hat{e}_m \) can be replaced by the real energy values \( e_1 \ldots e_m \), yielding exact computations, although the computation of the \( \kappa \) parameter still requires numerical approximation because of the terms involving Bessel functions. The resulting fitted distribution is shown in Figure 4(e).

Once the reference frequency has been normalized, the next step consists in the computation of audio descriptors from the music signal. Among the different features, chroma vectors are the ones that seem to capture the most relevant information for an identification task.

A chroma vector \( c = (c_1 \ldots c_{12}) \) is a 12-dimensional vector, computed from the spectral representation of a windowed signal, in which each component is associated with a specific pitch class.
to a pitch class (the 12 notes of the Western music scale). A general formula to compute chroma vectors from a windowed signal $s(t)$ is the following

$$c_i = \sum_f B_i(f) \cdot S(f)$$

(4)

where $S(f)$ is a representation of $s(t)$ in the frequency domain and $B_i(f)$ is a bank of bandpass filters, each centered on the semitones belonging to pitch class $i$ and with a bandwidth of a semitone.

The actual computation of $c$ may vary depending on a number of parameters. For instance, $S(f)$ can be the discrete Fourier transform of the signal, but can also be based on the instantaneous frequency of the signal or represent the main peaks to overcome the problems due to leakage. Moreover, the shape of the bandpass filters $B_i(f)$ and their support in the audible range are expected to influence the ability of chroma to discriminate different music signals.

Although there is a general consensus on the effectiveness of chroma vectors for a music identification task, as reported in Section II, the effect of the way chroma are calculated has not been investigated in detail. For this reason, in this work we investigate three distinct approaches for chroma feature extraction: two approaches reported in the literature [8], [20], and a third approach developed on purpose which is based on a peak picking algorithm that considers only local maxima in the discrete Fourier transform of $s(t)$ to compute chroma vectors.

Last, a common problem when dealing with music identification is that different versions of the same music work can be performed in different keys. A brute-force approach to overcome this issue is to perform the same query multiple times, once for each possible transposition. This usually means having to perform the query 12 times and selecting the best match for each document in the collection, a task which is both demanding computationally and less accurate than performing the query just once, because the possibility of identification errors is replicated.

A more refined approach requires the use of the audio content to detect the key of the recording. Key finding algorithms can be used to transpose to the same tonality – e.g. in C major – both the recordings in the collection and the query. For the aims of music identification, it is important that a key-finding algorithm assigns the tonalities consistently, even in case it assigns the same wrong tonality to two different recordings. For instance, if a recording in E minor is mistakenly labeled as in G major, a query of that music work will be correctly identified only if the key finding algorithm makes the same mislabeling from a minor to major. Below we present a simple yet effective key finding algorithm, for which the parameters are learned in a supervised setting.

The first step towards key estimation is to produce a single chroma vector, which represents a harmonic profile...
for the whole recording, by summing all the normalized chroma vectors for which the kurtosis value is higher than a certain threshold. The use of such a threshold allows us to discard chroma vectors that do not have a clear harmonic profile, such as crowd noise.

In order to detect the most probable key, an inner product is carried out between a suitably chosen vector of weights and all the possible transpositions (circular rotations) of this profile. The most probable keys for the recordings are finally extracted by sorting the transposition indices according to the results of the inner products.

The vector of weights is learned from the audio content using a randomized hill climbing approach: starting from random initial values, at each step the algorithm tries to maximize the number of recording pairs for which the relative key difference matches manual annotation by repeatedly moving the weight vector to nearby points in the space; since we are unable to match all of the key differences, we also allow a match for the second and third most probable key difference, at the cost of a higher penalty in the training function. This relaxation corresponds to a real scenario in which the recordings in the collection are transposed to their most probable key, and a query is executed by transposing it at most three times (instead of 12).

V. EFFICIENT IDENTIFICATION

Once audio recordings have been represented by chroma vectors, two further steps are introduced to allow for a compact representation of chroma vectors: quantization and hashing. The quantized version \( q \) of a chroma vector \( c \) is obtained by taking into account the ranks of the chroma pitch classes, sorted by their values. Let \( r_i \) be the position in which the \( i \)-th component \( c_i \) would be ranked after a descending sort (starting from 0); a \( k \)-level rank-representation of \( c \) is constructed by considering a base 12 number computed as:

\[
q = \sum_{i: r_i < k} i \cdot 12^{r_i}
\]

For example, the 3-level quantization of the chroma vector depicted in Figure 5 yields the value \( \tilde{q} = 10 \cdot 12^0 + 3 \cdot 12^1 + 7 \cdot 12^2 = 1054 \). This approach has been already applied to develop a clustering component for a statistical approach to classical music identification, as reported in [10]. As discussed in Section VII, we will explicitly investigate the effect of the number of quantization levels on identification effectiveness.

As described in Section III, each recording is segmented in short excerpts of fixed length. Thus the result at this step is a set of segments, where each segment is represented by a set of hashes. Identification is carried out by computing a similarity score among the query and the documents in the collection, and ranking the documents in decreasing order of this similarity score. The similarity
between segments is computed according to

$$
S(Q, D) = \prod_{q \in Q} \max_{d \in D} \left\{ \sum_{t \in q \cap d} \min \left( \frac{tf(t, d)}{|d|}, \frac{tf(t, q)}{|q|} \right) \right\}
$$

Similarity at segment level

where $|d|$ and $|q|$ denote respectively the number of hashes in a document and in a query segment, and the term frequency $tf(t, d)$ denotes the number of occurrences of hash $t$ in a segment $d$. We can identify two different steps when computing this measure. The first step implies the computation of local similarity between segments $d$ and $q$ as the (normalized) number of terms they have in common

$$
S_L(d, q) = \sum_{t \in q \cap d} \min \left( \frac{tf(t, d)}{|d|}, \frac{tf(t, q)}{|q|} \right)
$$

while the second step aggregates the contributions of all the query segments, by computing the geometric mean of the best local similarity values for each query segment:

$$
S(Q, D) = \sqrt[|Q|]{\prod_{q \in Q} \max_{d \in D} S_L(d, q)}
$$

The above similarity function can be efficiently implemented by interpreting a hash in a segment as a term in a textual document. The remainder of this section will describe how this hash-to-textual term mapping has allowed us to exploit data structures and ranking algorithms commonly adopted in text retrieval for music identification.

The computation of the similarity score for a query-document pair requires information on the hash frequency in each query and document segment. Information on the frequency of occurrence of a hash in a specific query segment can be extracted at query time and efficiently accessed by means of data structure maintained in memory. The current implementation of the architecture exploits a list of maps, where each list entry, namely each map, corresponds to a segment and the $<\text{key},\text{value}>$ pairs retained by each map are $<\text{hash},\text{frequency}>$ pairs.

Information on the frequency of occurrence of an hash in a document can be efficiently accessed by means of an inverted index. An inverted index shares the same intuition of a book index. Let us denote with vocabulary the set of distinct terms appearing in the book. For a given term, the book index provides information on the pages where the term is used; the book index allows the reader to efficiently access the passages of the book where the term is mentioned. In an inverted index data structure an inverted list is associated to each distinct term appearing in at least one of the documents in the collection; the entries of the inverted list are the (identifiers of the) documents where the term occurs. Moreover, additional information necessary by the ranking function can be stored in the inverted list entries, e.g. the frequency of occurrence of the terms in the documents or the positions where the terms occur.

In the methodology adopted in this paper, the vocabulary entries are chroma hashes, and each segment of a recording is interpreted as a textual document. More specifically, our approach can be related to segment-to-passage mapping, where a textual document is sub-divided into passages, and inverted list entries maintain term information at a passage level. Figure 6 provides an example of the indexing process when applied to a recording, represented as a sequence of hashes. Such a sequence undergoes a segmentation process where possibly overlapping subsequences are extracted from the recording sequence; in Figure 6 the number of hash per segment is seven and the overlap involves two hashes. After the segmentation, a recording is represented by a set of hash sequences. In accordance with the bag of features paradigm we hypothesize that information on hash occurrences at a segment level allows us to effectively identify a recording; positional information is therefore ignored, thus each segment is effectively treated as a set of hashes. Using an inverted index we can efficiently store hash occurrence information: our vocabulary is the set of distinct hashes in all the recordings in the collection, more-
over, the key finding algorithm described in Section IV is applied to the query, thus obtaining diverse transpositions that the system treats as distinct queries that can be processed in parallel. Moreover, after the computation of the hash values representing the audio content, the query is split into a number of “segment-queries” q’s. Each segment-query is processed by a Document At A Time (DAAT) strategy. DAAT strategies evaluate the contributions of every query term with respect to a single document before considering the next document. One advantage of the DAAT strategy is that it does not require intermediate document scores to be maintained during the entire ranking process, thus limiting the run-time memory usage.

For each segment-query the system returns a ranked list whose entries are the document segments of all the documents in the collection ranked in decreasing order of segment level similarity score. The maximum among all the document segments is then computed by going through the returned ranked list. The next step consists in the aggregation of the results returned by the diverse segment-queries, q’s, according to Equation 7. Last, the results obtained from the diverse considered transpositions are merged to obtain a final results list. One of the main advantages of this retrieval strategy is that the diverse query (transpositions) and the diverse segment-queries can be processed in parallel. The modules for hashing, quantization, indexing and retrieval have been implemented in a software architecture named FALCON, built on top of the popular open source text search engine Lucene and released under an open source license.

Identification of Multiple Works Within a Single Query

The algorithm detailed above assumes that a query can be matched with a recording of the same music material. This is the typical scenario for cover song identification systems, but in order to handle the case of a query containing multiple works – such as digitizations of tapes or vinyl discs containing several tracks – the methodology must be modified.

To this end, an additional time-resolution level was added to the segmentation hierarchy. Basically, the recording on an LP side is divided into shorter elements, each one considered as an individual query. Figure 7 shows the procedure, which provides us with a number of resulting rank lists that are merged into a structure called rank matrix. We will refer to these short elements as “chunks”, to disambiguate them from the segments in which a single track is divided.

The rationale behind this choice is related to the way the similarity between a query and the documents is computed. As can be seen from Equation 7, it involves a maximization over indexed segments of the local similarities for each query segment. As a consequence, the system is designed to support short queries containing a portion of a corresponding recording while it becomes ineffective with long queries containing segments of different recordings, because the geometric mean will be computed also from local scores with very low similarity values. It should be noticed that, while the maximization and averaging indexes of Equation 7 − d and q respectively – could be in principle reversed (there is no theoretical reason for the asymmetry of the global similarity function), several implementation choices aimed at efficiency depend on this configuration.

The similarity $S_i$, associated to the recording corresponding to the result indexed in the i-th row of the rank matrix is computed with a simple data fusion approach:

$$S_i = \max_{w=1 \ldots L-W} \sum_{k=w}^{w+W-1} \frac{1}{(\max(r_{i,j,C}))^2}$$

where $L$ denotes the number of query chunks (columns of the rank matrix), and $r_{i,j}$ represents the position of recording i in the rank list produced from chunk j.

The underlying idea is to consider each row of the rank matrix, and to analyze small windows of length $W$: an indexed recording spanning multiple chunks is supposed to consistently rank in high positions, and the windowing (along with the saturation constant $C$) acts as a filter for documents whose behavior in the ranking sequence is noisy.

The choice of using rank values instead of similarity scores is motivated by the consideration that averaging similarity scores works only if all segments belong to the same recording, whereas different recordings induce radically different similarity values. As for many data fusion approaches, the choice of rank values reduces the effect of large differences in the similarity scores. Moreover, the choice of ranks enables independence of the particular parametrization adopted for the lower-level local similarity computation.

VI. ALIGNMENT-BASED IDENTIFICATION

The alignment algorithm is based on the idea that the progress of a music performance can be tracked using

---

another recording as a reference, modeling the relative position of one with respect to the other using stochastic motion equations [21]. Since it is not possible to make any assumption about the initial relative position of the recordings (e.g. an excerpt of a music piece might be aligned with the complete work), the algorithm proceeds by assuming several hypotheses and selecting the correct one during the course of the inference procedure.

The aim of the post-processing step is dual: the analysis of an alignment is particularly strong evidence for the relevance classification of a document and, in case of positive relevance, the alignment itself enables fine-resolution access to corresponding points in the recordings.

The system state is modeled as a two-dimensional random variable \( x = (s, t) \), representing the current position in the reference audio and tempo respectively; \( s \) is measured in seconds and \( t \) is the speed ratio of the performances. The incoming signal processing frontend is based on spectral features extracted from the FFT analysis of an overlapping, windowed signal representation, with hop size \( \Delta T \). In order to use sequential Montecarlo methods to estimate the hidden variable \( x_k = (s_k, t_k) \) using observation \( z_k \) at time frame \( k \), the state evolution is assumed to be Markovian.

Let \( p(z_k|x_k) \) denote the likelihood of observing an audio frame \( z_k \) of the take given the current position along the reference performance \( s_k \). A simple spectral similarity measure is considered, defined as the Kullback-Leibler divergence between the power spectra at frame \( k \) of the take and at time \( s_k \) in the reference.

The pdf \( p(x_k|x_{k-1}) \) for the state transition makes use of tempo estimation in the previous frame, assuming that it does not change abruptly:

\[
p(x_k|x_{k-1}) = \mathcal{N}\left(\begin{bmatrix} s_k \\ t_k \end{bmatrix} \mid \mu_k, \Sigma\right) \quad (9)
\]

\[
\mu_k = \begin{bmatrix} \mu_{k-1} + \Delta T t_{k-1} \\ t_{k-1} \end{bmatrix} \quad (10)
\]

\[
\Sigma = \begin{bmatrix} \sigma^2 s_k & 0 \\ 0 & \sigma^2 \Delta T \end{bmatrix} \quad (11)
\]

Sequential Montecarlo inference methods [22] work by recursively approximating the current distribution of the system state using the technique of Sequential Importance Sampling: a random measure \( \{x_k^i, w_k^i\}_{i=1}^{N_s} \) is used to characterize the posterior pdf with a set of \( N_s \) particles over the state domain and associated weights, and is updated at each time step as in Algorithm 1. In particular, \( q(x_k|x_{k-1}, z_k) \) is the particle sampling function. The resampling step is used to address the degeneracy problem, common to Sequential Montecarlo approaches [23].

The decoding of position and tempo is carried out by computing the expected value of the resulting random measure (efficiently computed as \( \mathbb{E}[x_k] = \sum_{i=1}^{N_s} x_k^i w_k^i \)).

Initialization plays a central role in the performance of the algorithm. The initial relative position is unknown, which corresponds in a probabilistic context to an appropriate choice of the prior distribution: \( p(x_0) \) is set to be uniform over the whole duration \( L \) of the reference performance, and the algorithm is expected to “converge” to the correct position after a few iterations of the inference update step. Figure 8 shows a typical evolution of the probability distribution for the position of the input at different moments of the alignment.

**Algorithm 1: SIS Particle Filter - Update step**

for \( i = 1 \ldots N_s \) do

\[
sample \ x_k^i \text{ according to } q(x_k^i|x_{k-1}, z_k)
\]

\[
\tilde{w}_k^i \leftarrow w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1})}{q(x_k^i|x_{k-1}, z_k)}
\]

\[
\tilde{w}_k^i \leftarrow \frac{\tilde{w}_k^i}{\sum_{j=1}^{N_s} \tilde{w}_k^j} \quad \forall i = 1 \ldots N_s
\]

\[
N_{eff} \leftarrow \left(\sum_{i=1}^{N_s} (\tilde{w}_k^i)^2\right)^{-1}
\]

if \( N_{eff} < \text{resampling threshold} \) then

resample \( x_1^i \ldots x_k^i \) according to ddf \( w_k^1 \ldots w_k^{N_s} \)

\[
w_k^i \leftarrow N_s^{-1} \quad \forall i = 1 \ldots N_s
\]

A relevant parameter of Algorithm 1 is the resampling threshold. The sparsity of the distribution in the initial phases of the alignment imposes a careful choice of resampling threshold, otherwise many relevant hypotheses can be lost in the resampling phase.

It is clear at a visual inspection if an alignment corresponds to a positive identification or not: in the first case its plot resembles a straight line – which, supposedly that the tempo of the performances is similar, should be characterized by a roughly unit slope – such as that of Figure 9(a), while in the second case it displays irregular behavior as pictured in Figure 9(b).

Logistic regression is used to classify alignments. The features used by the classifier are the components of a 14-dimensional histogram of the slopes between consecutive alignment points. As Figure 9(c) shows, the concentration of slope values around 1 is particularly revealing of a
correct alignment, whereas incorrect alignments are characterized by irregular distributions. In [21] an alternative algorithm is proposed for detecting the relevant portion of a correct alignment by performing regression on the data points. We chose to use a classifier in this work because of the advantage of learning the model from examples that can be quickly hand-labeled.

Given a rank list produced by FALCON, a simple strategy for improving the accuracy of the results consists in aligning the query against the top results and moving to the top of the list those for which the alignment is classified as correct.

The remainder of this section is organized as follows. Section VII-A will briefly introduce the measure adopted for evaluating identification effectiveness. Section VII-B will discuss the specific research questions investigated by the experiments together with the experimental methodology to address them. Section VII-C will provide details on the adopted test collections. Last, Section VII-D will report on the obtained results.

A. Evaluation Measure

Let us denote with $D$ an ordered list of documents retrieved by a system in response to a query and with $R$ the subset of relevant documents. Precision is the fraction of retrieved documents that are relevant to the search, and recall is the fraction of relevant documents that are successfully retrieved:

\[
\text{Precision} = \frac{|R \cap D|}{|D|} \quad \text{Recall} = \frac{|R \cap D|}{|R|}
\]

(12)

Precision (recall) at $n$, denoted as $P@n$ ($R@n$) measures precision (recall) considering only the first $n$ retrieved documents.

The Average Precision for a query is computed as the average of the precision values (the ratio of relevant documents over retrieved document) at each of the relevant documents in the ranked sequence. Let $R_i(j) = 1$ if the document at rank $j$ is relevant for the $i$-th query, 0 otherwise; then for $n$ queries the Mean Average Precision (MAP) value is computed as the mean of the average precision for each query:

\[
\text{MAP} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sum_{j} R_i(j)} \sum_{j} P_i@j R_i(j)
\]

(13)

B. Experimental Methodology

The main points reported below are concerned with the trade off between effectiveness and efficiency. Experiments are focused on two identification tasks: in the first task a music performance is given as query and the objective is to identify existing versions of the query in the document collection; the second task is concerned with the case of queries containing multiple works.

Effect of the chroma vectors extraction procedure on identification effectiveness

Equation 4 describes the basic rationale underlying the computation of chroma vectors. Such descriptors can however be calculated in different ways; the choice of a particular algorithm can greatly affect identification effectiveness, and should be motivated by experimental evidence. Three algorithms are the object of investigation. The first one, hereafter named MOD, is based on a peak picking algorithm that considers only local maxima in the discrete Fourier transform of the acoustic signal. The other two approaches are the ones proposed in [8] and [20], to which we will refer as LabRosa and MirToolbox, respectively.

VII. EXPERIMENTAL EVALUATION

The main points addressed by the experiments reported in this section are concerned with the trade off between effectiveness and efficiency. Experiments are focused on two identification tasks: in the first task a music performance is given as query and the objective is to identify existing versions of the query in the document collection; the second task is concerned with the case of queries containing multiple works that was discussed in Section V.
Effect of the Tuning Adjustment Algorithm

When dealing with music identification, a significant issue is that a work can be performed with an instrument tuned according to a non-standard reference frequency. The tuning adjustment algorithm proposed in Section IV aims at addressing this issue. The effectiveness of the tuning algorithm on the chroma vector extraction procedure is therefore evaluated comparing the resulting accuracy.

Effect of Quantization and Segmentation

The methodology described in Section V is concerned with the transformation of a chroma-based representation of a music work into a bag-of-hashes representation, where each work is segmented into possible overlapping sets of hashes. Quantization and segmentation involve a number of parameters that can affect identification effectiveness. The first parameter is the quantization level: by increasing the number of quantization levels, the size of the vocabulary of descriptors increases, with dramatic consequences on the accuracy of the system.

The result of the hashing procedure is a hash sequence that undergoes a segmentation process in order to capture possible structure in the music work or to identify possible works in a long performance. Segmentation involves two parameters: the segmentation length and the overlap. The value of these settings may affect both identification effectiveness and efficiency: highly overlapped segments are expected to have a high capability to capture structure in a song, but increasing the number of documents in the index slows down the identification process. It is therefore important to choose an appropriate tradeoff.

In order to address these questions, variations in terms of MAP are measured with respect to modification of the parametrization. Since the selection of the diverse parameter values can affect the effectiveness of the chroma vectors extraction procedures as well as the effectiveness of the tuning adjustment algorithm, different combinations of chroma extraction, tuning, quantization and segmentation are examined.

Effect of Reranking using Audio Alignment

The adoption of an efficient strategy at the expense of accuracy is motivated by the ability of quickly returning a promising list of relevant candidates among the documents in a large collection, with the aim of refining the initial list using a more sophisticated, but also computationally expensive, technique. It is clear then that recall is more important than precision in the first stage. The margin of improvement in accuracy that is obtainable using audio alignment remains to be examined.

In order to address this point, the top $k$ documents in the list returned by the system described in Section V are provided as input to the proposed audio alignment approach and the initial rank list is modified according to the documents that are associated to correct alignments. This hybrid approach depends on the value of the parameter $k$: higher values improve the quality of identification but also increases the processing time proportionally.

Identification of Multiple Works in a Single Query

The last point concerns the capability of the proposed methodology to identify multiple works within a music recording. In particular, it should be verified whether a fixed-length segmentation strategy is able to support identification of recordings characterized by radically different durations. This issue is addressed through the approach described in Section V.

C. Test Collections

An experimental collection was recently introduced and adopted in the Audio Cover Song Identification track in MIREX, in order to compare our approach to other approaches proposed in the literature. The collection is a selection of recordings collected by the Mazurka Project, from which we sampled 11 versions from 49 mazurkas obtaining a total of 539 tracks. In order to replicate the MIREX evaluation procedure, we randomly selected 10 diverse sets of 49 mazurkas and averaged the results.

Last, a collection of 100 LPs from the Alicante Fonoteca was used to evaluate the procedure detailed in Section V, matching their contents with recordings in the MusiCLEF2011 collection. The vinyl discs were digitized using common LP record player equipment.

D. Results

Experiments were performed on a machine featuring a dual-core 3.4 Ghz CPU and a 7200 RPM hard disk. Chroma features were extracted with an analysis length of 200ms and hopsize of 100ms.

Table I displays the effect of Chroma extraction algorithm choice, tuning estimation, segmentation of the audio content and quantization level on the Mean Average Precision and mean query time, using the MusiCLEF2011 collection (results are averaged on 651 queries). Considering the accuracy/efficiency tradeoff, we selected the highlighted configuration (MirToolbox features with tuning estimation, segments of 50s without overlap and 3-level chroma quantization) for the subsequent tests.

Table II shows the results of the selected configuration on the Mazurka datasets, comparing the approach to previous MIREX evaluation campaigns. On this dataset (539 files) our system is able to process about 5 queries per second. Due to the choice of rather long segments
TABLE I.

Effect of Chroma extraction algorithm, tuning estimation, segmentation of the audio content (length L, overlap O) and quantization level Q on the accuracy (in terms of Mean Average Precision) and query time of the proposed methodology.

<table>
<thead>
<tr>
<th>Segmentation</th>
<th>LabRosa Chroma Vectors</th>
<th>MOD Chroma Vectors</th>
<th>Mir/Toolbox Chroma Vectors</th>
<th>Query Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>L [s]</td>
<td>O [s]</td>
<td>Q normal</td>
<td>tuned</td>
<td>normal</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>2</td>
<td>55.6</td>
<td>64.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>53.1</td>
<td>61.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>19.9</td>
<td>20.1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>58.4</td>
<td>68.5</td>
<td>72.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>55.6</td>
<td>64.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>27.4</td>
<td>24.7</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>2</td>
<td>55.3</td>
<td>64.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>54.3</td>
<td>63.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>33.0</td>
<td>34.6</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>58.0</td>
<td>67.5</td>
<td>71.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>56.6</td>
<td>65.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>38.0</td>
<td>42.0</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>2</td>
<td>54.6</td>
<td>64.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>55.4</td>
<td>63.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>39.5</td>
<td>42.2</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>57.4</td>
<td>66.2</td>
<td>69.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>57.6</td>
<td>65.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>44.1</td>
<td>47.6</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>2</td>
<td>53.7</td>
<td>62.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>55.2</td>
<td>64.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>42.3</td>
<td>47.7</td>
</tr>
<tr>
<td>25</td>
<td>2</td>
<td>55.9</td>
<td>64.2</td>
<td>68.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>56.0</td>
<td>66.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>44.8</td>
<td>51.3</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>2</td>
<td>52.3</td>
<td>60.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>53.5</td>
<td>63.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>44.0</td>
<td>50.4</td>
</tr>
<tr>
<td>30</td>
<td>2</td>
<td>54.6</td>
<td>62.8</td>
<td>66.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>55.7</td>
<td>65.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>46.0</td>
<td>52.6</td>
</tr>
</tbody>
</table>

(V0s), a particularly short mazurka is never retrieved; results are shown twice, depending on its inclusion.

Table III and IV report the effect of different choices for the segmentation and length of the analysis window on accuracy and efficiency of the algorithm for identification of multiple works within a query. 100 LPs from the Alicante fonoteca were matched against 6680 recordings.

Last, Figure 10 presents improvement of reranking using the algorithm presented in Section VI. A minute of audio can be processed in 2.8s for $N_s = 10^5$ particles on our reference machine (using a single core); experiments were performed with $N_s = 100$ particles for every minute of the reference recording.

VIII. CONCLUSIONS

In this paper we present a complete methodology for music identification. The main application that motivates our approach is the automatic segmentation and identification of recordings digitized from a sound archive. In particular, we focus on the experimentation on a collection of vinyl discs, provided by the Fonoteca of the University of Alicante, although the methodology is not linked to a particular carrier and can be readily extended to audio tapes, shellack discs, and so on. Automatic identification allows the music digital library to retrieve relevant metadata about the music works even if this information was incomplete or missing at the time of the digital acquisition. Automatic segmentation is obtained as a subproduct of the identification of individual recordings within a single audio file allowing the music digital library to grant access to single tracks, instead of the whole side of a disc (as usually happens in the case of digitized material). We believe that the availability of systems for the automatic identification of music work will promote the dissemination of music cultural heritage, allowing final users to retrieve and to access individual recordings.

We carried out the evaluation on three alternative datasets, all of them of classical music, which in Europe is particularly related to the dissemination of cultural heritage. Results are consistent across collections and, although the methodology aims at efficiency, also the effectiveness of the proposed approach is in line with the results described at MIREX for the Mazurka dataset, and with other results presented in the literature for classical music and other genres. It is interesting to note that the need for a tool that automatically identifies different recordings of the same work became apparent also during the experimental evaluation. The lack and imprecision of metadata of the Fonoteca collection made it difficult to create a ground truth for the experiments, and most of
Measure MIREX 2009 MIREX 2010 FALCON
Mean Average Precision 0.91 0.96 0.56 0.79 0.82 0.88 0.90
Correct matches in top 10 8.83 9.58 5.27 7.55 6.04 7.94 8.76 8.92

TABLE II.
COMPARISON OF THE SYSTEM DESCRIBED IN SECTION V WITH PAST MIREX EVALUATION CAMPAIGNS.

<table>
<thead>
<tr>
<th>Segment Length [s]</th>
<th>Overlap [s]</th>
<th>Window length</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>0</td>
<td>3</td>
<td>68.4</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>67.5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>66.9</td>
</tr>
<tr>
<td>60</td>
<td>3</td>
<td>72.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>72.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>72.5</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>3</td>
<td>67.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>72.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>74.2</td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>0</td>
<td>3</td>
<td>57.8</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>57.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>56.1</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>3</td>
<td>67.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>67.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>67.1</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>3</td>
<td>63.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>68.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>69.2</td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>0</td>
<td>3</td>
<td>51.1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>49.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>49.3</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>3</td>
<td>61.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>62.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>61.2</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>3</td>
<td>62.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>65.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>64.7</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>0</td>
<td>3</td>
<td>43.8</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>43.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>43.3</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>3</td>
<td>53.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>53.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>52.3</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>3</td>
<td>51.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>56.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>56.0</td>
<td></td>
</tr>
</tbody>
</table>

TABLE III.
EFFECT OF SEGMENTATION AND LENGTH OF THE ANALYSIS WINDOW W ON THE LP IDENTIFICATION TASK.

<table>
<thead>
<tr>
<th>Segment Length [s]</th>
<th>Overlap [s]</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>120</td>
<td>60</td>
<td>32</td>
</tr>
<tr>
<td>120</td>
<td>80</td>
<td>46</td>
</tr>
<tr>
<td>180</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>180</td>
<td>90</td>
<td>27</td>
</tr>
<tr>
<td>180</td>
<td>120</td>
<td>40</td>
</tr>
<tr>
<td>240</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>240</td>
<td>120</td>
<td>25</td>
</tr>
<tr>
<td>240</td>
<td>160</td>
<td>38</td>
</tr>
<tr>
<td>300</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>300</td>
<td>150</td>
<td>27</td>
</tr>
<tr>
<td>300</td>
<td>200</td>
<td>40</td>
</tr>
</tbody>
</table>

TABLE IV.
IDENTIFICATION TIME OF A COMPLETE LP DISC.

the recordings have been manually inspected. Moreover, the analysis of the results showed that some of the false positives were actually correct matches, because the audio content corresponded although metadata in the RTI collection differed. A notable example is Mozart’s Requiem, where the last movement “Lux Aeterna” has been added by another composer as a variation of the second movement “Kyrie”, written by Mozart: When “Kyrie” was used as a query, our system ranked at the top position “Lux Aeterna” and viceversa. Other examples are the arrangement of a known theme, such as Mozart’s “Turkish March” reprised by Max Reger, that have been recorded with a different name. We decided to leave these examples when computing effectiveness, because they are representative of real world cases.

The main application of the proposed methodology is as a supervised tool for automatically retrieving relevant metadata during the acquisition process of audio recordings, either in analog and digital formats. The efficiency of the approach allows the user to carry out identification at the same time the analog recordings are digitized. The possibility to provide the user with a list of candidate metadata during the first cataloguing of the material may reduce the number of cataloguing errors and speed up the process in case of incomplete or inconsistent information on the covers. Although the methodology has been tested with a collection of LPs, an important application will be
to catalog analogue tape recordings of rehearsals made in theatres and concert halls, where the information on the tape covers is usually extremely poor and requires a deep knowledge of the music domain even to provide basic information such as title and composer.

The natural extension of the proposed approach is the application to other music genres. In particular, cultural heritage applications could be related to applications in ethnometrics where identification of particular themes can help to highlight relationships among music styles of different geographical areas or historical periods. A major part of the proposed methodology is directly extensible to other genres of Western music, because the ideas behind fine tuning, chroma features, and tonality finding are not linked to a particular genre. What is needed for a future application to other genres is a refinement of the alignment approach in order to take into account large difference in the music structure, which may happen in the case of covers in the pop and rock genres.

ACKNOWLEDGMENT

The authors are particularly grateful to David Rizo and the Fonoteca of the University of Alicante for providing access to the collection of digitized LPs.

REFERENCES


