MUSIC SIMILARITY BASED ON SEQUENCES OF DESCRIPTORS: TONAL FEATURES APPLIED TO AUDIO COVER SONG IDENTIFICATION

Joan Serrà Julià

Master Thesis submitted in partial fulfillment of the requirements for the degree:
Màster en Tecnologies de la Informació, la Comunicació i els Mitjans Audiovisuals

Supervisor: Xavier Serra

Department of Information and Communication Technologies
Universitat Pompeu Fabra
Barcelona, Spain
September 2007

Abstract

As music collections are growing up every day, it becomes necessary to keep them organized. Therefore, one critical block of an 'intelligent' system for doing that should deal with the concept of music similarity. From a computational point of view, this concept has been assessed from many angles, being one of the most important the purely audio content-based similarity systems. Traditionally, they have relied in methods that ignored the temporal evolution of the music audio characteristics.

It is the goal of this thesis to provide some insights into the benefits of considering temporal sequences representing the musical content of an audio signal. We do that while focusing to an application that have been increasing his popularity along these last few years, as it provides a direct and objective way for evaluating music similarity: cover song identification. A cover song (or simply cover) is a new version (performance, rendition or recording) of a previously recorded song.

After making an extensive literature summary on related techiques, methods and systems for music similarity (with a special emphasis on cover detecting systems), and presenting our evaluation methodology, we perform several experiments on cover song identification based on state-of-the-art methods (cross-correlation and Dynamic Time Warping). We also study the blocks of these methods that can report more benefits to the final performance of the system and we make some interesting improvements on them (such as considering different PCP distances, beat tracking algorithms, key transposition methods), apart from assessing the intrinsic algorithms' parameters.

Furthermore, we propose a new method for determining the similarity between tonal sequences and, therefore, to cover songs. This one is based on a novel HPCP similarity measure, and on a newly developed Dynamic Programming local alignment technique. Results confirm that the performance of the proposed system is significantly superior to the other implemented ones.

Along the thesis we keep discussing important details found during the experiments done and future directions to take for accomplishing the chosen task.

Acknowledgments

Ideas, collaborations, help, advice, comments, and many of the interactions that arise from working jointly with a community of people are very difficult (if not impossible) to quantize. But as this is the obejctive of this page, I should, at least, enumerate all the people that have been involved in some sense with the research I've carried out during this thesis.

First of all, I would like to thank Xavier Serra, my supervisor, for giving me the opportunity to join the Music Technology Group of the Pompeu Fabra University, and for supporting my research along this year. It is a pleasure to be part of such a brilliant and estimulating group of people.

There are two persons who I consider as the main contributors of this thesis and who have given to me plenty of support, ideas and advice: Emilia Gómez and Perfecto Herrera. Without them, this work would not have been as it is.

Thirdly, I wish to thank my colleagues from office 316+++: César Alonso, Òscar Celma, Enric Guaus, Cyril Laurier, Montserrat Puiggrós, and Mohamed Sordo. The regular thursday meetings and also the 'irregular' conversations at any hour in the office or outside it, have been useful (for sure) for the research exposed here.

I also want to mention all the other members of the MTG that I have been in touch for some reason, specially Jordi Bonada, Paul Brossier, Pedro Cano, Maarten De Boer, José Pedro García, Pablo García, Amaury Hazan, Gunnar Holmberg and Hendrik Purwins.

Finally, let me add special greetings to the technical and administrative staff of the IUA-MTG for their help and support during this year. With this, I leave for finished this (inevitably) short list.

This research has been partially funded with a scholarship of the Pompeu Fabra University (UPF), and by the EU-IP project PHAROS[1] (Platform for searcH of Audiovisual Resources across Online Spaces).

Joan Serrà Julià
Barcelona, September 2007

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

Contents

List of tables and figures

Tables

Figures

Chapter 1

Introduction

This thesis addresses the similarity of tonal sequences within songs. Determining similar parts of music (in the sense of tone or pitch information) can have many utilities, being a clear one the application this work addresses: cover song identification.

In the next sections it will be shown why the concept of musical resemblance is important, which is the role played by tonal sequences, and how do we come to the idea of using cover songs for evaluating music similarity (how cover songs are related to it). Also, a little insight into the term connotations and its historical background is made. In addition, the context where this research is carried out is highlighted.

1.1 Motivation

Nowadays, anyone who listens to music may have thousands of songs stored in a hard disc, or even in an MP3 portable player. Furthermore, on-line and digital music stores own large music collections, ranging from thousands to millions of tracks.

In general, we are experiencing a radical increment of the volume of music collections. Technological improvements in networks, storage media, compression algorithms, portability of devices and internet services have favoured that. Additionally, the 'unit' of music 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.

When dealing with these extensive music collections, finding a song that fits one's needs or expectancies might be problematic. It is here when it becomes interesting to organize them automatically according to some criteria of resemblance. Thus, a similarity measure between songs seems to be the main block of an 'intelligent' interface to make these collections manageable.

Approximating this issue, can lead to an extensive set of new tools to interact with music, enabling users to find new pieces similar to a given one, providing recommendations of new pieces, automatically organizing and visualizing music collections, creating playlists, personalizing radio streams, etc. There is a social need for these tools. According to [Vignoli 05], end-users identify the concept of finding similar songs, albums or artists as one of the most appreciated features for future music players. Furthermore, commercial success of large music catalogs nowadays is based on finding the music that people wants to listen to [Casey 06b].

Finally, we have to note 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.

1.2 Music Information Retrieval

Music Information Retrieval (MIR) is the interdisciplinary science of retrieving information from music[1]. Despite the emphasis on retrieval in its name, MIR encompasses a number of different approaches aimed at music management, easy access, and enjoyment [Orio 06].

Most of the research, technologies and systems on MIR, are based on audio content. The main idea underlying content-based approaches is that a document can be described by a set of features that are directly computed from its content. There are many disciplines involved in this issue, such as signal processing, musicology, computational modelling, statistics, machine learning, information retrieval, and many more [Fingerhut 04].

Much research over the last years have been devoted to MIR. The annual ISMIR conference[2] is the first established forum for those involved in works on accessing digital musical materials. There has been an increasing number of contributions and attendance to this conference (the number of published articles has evolved from 35 in 2000 to 129 in 2007). At the same time, important contributions to this field have appeared on related events (i.e., the European Conference on Information Retrieval[3], Audio Engineering Society Conventions and Conferences[4] or the Digital Audio Effects Conferences[5], to name a few) and on relevant journals (i.e., Journal of New Music Research[6], Computer Music Journal[7], EURASIP Journal on Applied Signal Processing[8], IEEE Transactions on Speech and Audio Processing[9] or INFORMS Journal on Computing[10]). This increasing amount of scientific literature related to MIR, and, more concretely, to music content processing, reflects the tremendous growth of music-related data available and the consequent need to provide solutions to search this content.

The successful development of robust, large-scale MIR systems also has important social and commercial implications. According to Wordspot[11], an Internet consulting company that tracks queries submitted to Internet search engines, the search for music has displaced the search for sex-related materials as the most popular retrieval request. Yet at this moment, not one of the so-called *MP3 search engines* is doing anything more than indexing the textual metadata supplied by the creators of the files. It is not exaggerated to claim that a successful, commercially based, MIR system has the potential to generate vast revenues [Downie 03].

Beyond commercial implications, the emergence of robust MIR systems creates significant added value to the huge collections of underused music currently warehoused in the world's libraries by making the entire corpus of music readily accessible. This accessibility will be highly beneficial to musicians, scholars, students, and members of the general public alike.

1.3 Music similarity

Due to the forementioned tremendous growth of available audio data (either locally or remotely), and the consequent need to search this content and retrieve music efficiently and effectively, music similarity is a critical aspect of any system dealing with the access to digital music material.

[1] http://en.wikipedia.org/wiki/Music_information_retrieval
[2] http://www.ismir.net
[3] http://irsg.bes.org
[4] http://www.aes.org
[5] http://www.dafx.de
[6] http://www.tandf.co.uk/journals/titles/09298215.asp
[7] http://mitpress2.mit.edu/e-journals/Computer-Music-Journal
[8] http://www.hindawi.com/journals/asp
[9] http://www.ieee.org/organizations/society/sp
[10] http://joc.pubs.informs.org
[11] Data from 2001. http://www.wordspot.com/

There are many facets of musical information that help us to qualitatively assess music resemblance. Elements of sound as used in music are pitch (including melody and harmony), rhythm (including tempo and meter), and sonic qualities of timbre, articulation, dynamics, and texture[12]. Also, we should not forget the editorial, textual and bibliographic facets [Owen 00, Downie 03].

The most traditional and reliable technique for determining music similarity is through human annotations of music attributes. In order for this judgements to be of quality, a high degree of expertise is required, and this has a relatively high cost. Also, this process is clearly unfeasible for large quantities of music, as the music corpus is rapidly increasing. Furthermore, the spectrum of annotated music might be limited to music which is expected to sell. The Music Genome Project[13] (and the Pandora[14] system) would be a paradigmatic example of that. In this effort, a group of musicians and technicians try to "capture the essence of music at the fundamental level" by using over 400 attributes to describe songs.

Large communities can deal with much more music files than a few paid experts could. Collaborative filtering techniques are an alternative to manual classification. These techniques produce personal recommendations by computing the similarity between one persons' preferences (i.e., through playlists) and those of other people, and taking into account recurrences of these references. Some examples of this would be the last.fm radio[15] or the Amazon recommendation system[16]. However, these methods cannot quickly analyze new music, and their main data contain, basically, very commercial and widely known artists. Also, it may be difficult to obtain reliable information from users [Logan 01], and this information may be not objective and/or based on real musical properties.

Thus, a technique for automatically determining song similarity mostly based on the audio content seems to be the key factor in getting into it.

But similarity is an ambiguous term, and may depend on different musical, cultural and personal aspects. Many studies try to define and evaluate the concept of similarity, but there are many factors involved in this problem, and some of them (maybe the most relevant ones) are difficult to measure [Aucouturier 02a, Aucouturier 02b, Berenzweig 03, Pampalk 03].

In general, we could argue that the concept of music similarity is ill-defined and can be very subjective and context-dependent. So, a good starting point seems to be the identification of versions (or cover songs), where the similarity between them can be better defined, objectively measured, and context-independent.

1.4 Cover songs, versions, remakes and so forth

In this section we review some more conceptual aspects about cover songs. In the first subsection, the term is explained and put a little bit into an historical background. Next, we deal with some terminology associated with cover songs and follow with a more 'musicological' view of the characteristics that may change in a cover version. Finally, a general knowledge about the most covered songs is provided, and some resources are presented.

[12] http://en.music-web.org/encyclopedia/Music_Theory
[13] http://en.wikipedia.org/wiki/Music_Genome_Project
[14] http://blog.pandora.com/faq, http://www.pandora.com
[15] http://www.last.fm
[16] http://www.amazon.com

1.4.1 Cover versions

The term *cover version* originally implied an alternative version of a tune recorded by an artist subsequent to an *original version*. It all started in the early 1920's as a commercial strategy to introduce 'hits' that had had significant commercial success from other areas without remunerating any money to the original artist or label. Record companies made musicians 'cover' hit songs by recording a version for their own label in hopes of cashing in on the tune's success.

Popular musicians (and specially modern listeners) began using the word *cover* to refer to any remake of a song. However, some people distinguishes between a *cover version* and a *remake*, being a recording made soon after the original to cash in on its success for the former, or being a recording made much later, usually for artistic reasons or as a homage, for the latter.

Nowadays, musicians play what they call *cover versions* of songs as a tribute to the original performer or group. Established artists often pay homage to artists or songs that inspired them before they started their careers by recording cover versions, or performing unrecorded cover versions in their live performances for variety. In recent years, unrelated contemporary artists have contributed individual cover versions to tribute albums for well established artists who are considered to be influential and inspiring. One example of that would be the album "Who's next" (1971) by The Who, which was entirely covered by several artists in 2002 and still got into the high positions of many selling lists.

Certain songs are largely known for having a large number of cover versions and are called 'standards'. In musical forms like blues or, particularly, jazz, it is not uncommon for musicians to have albums or CDs made up primarily of standards. One just have to take a look at the *Real Book*[17] to find many standards that have been covered several times (for instance, Dave Brubeck's "Take five", George Gershwin's "Summertime", or Mark & Simon's classic "All of me").

Cover versions of many popular songs have been recorded, sometimes with a radically different style, and sometimes virtually indistinguishable from the original. Even, there exist groups and orchestras that have become famous by just recording cover versions of different artists. A paradigmatic example of that would be the String Quartet Tribute[18], a series of string quartet covers released by *Vitamin Records* that put a classical spin on many genres of music, including rock, pop, punk, techno, hardcore, country, metal and rap.

One extended use of cover songs is (and was from the very beginning) the translation of a song into another language. It is a common practice that some artists record a slightly different version with lyrics adapted to a particular country or language. For instance, many Latin-speaking artists such as Ricky Martin, Jennifer López or Shakira (to name a few) often record their songs both in English and Spanish, including also slight differences in the arrangements.

Also, *cover versions* are often used to contemporize familiar songs. That is, adapting old 'hits' to today's musical tastes and trends. An example of that is the hit "A little less conversation", originally recorded by Elvis Presley in 1968, which was covered by The Bosshorns in 2005 using modern arrangements and instrumentation to contemporize it.

Another important use for cover songs is done by record labels or producers to present new artists, who are introduced to the record buying public with performances of well known 'safe' songs. This is particularly evident with programs like *Pop Idol* and its international counterparts.

And, of course, there are also singers and musicians that perform covers of a favourite artist's hit tunes for the simple pleasure of playing a familiar song.

Nevertheless, recently, the term *cover song* has come to mean any recording of a tune previously

[17] http://en.wikipedia.org/wiki/Real_Book
[18] http://en.wikipedia.org/wiki/The_String_Quartet_Tribute

recorded by another artist. As the Wikipedia says[19], "in popular music, a cover version (or simply cover), is a new rendition (performance or recording) of a previously recorded song". That is the main definition that best fits our criteria, and thus, the one being considered when referring to *cover songs* in subsequent chapters of this thesis.

1.4.2 Most covered songs and resources

Paul McCartney's "Yesterday" is often claimed to be the most covered song in popular music history. An on-line cover song database lists a little over a hundred covers for the song[20], but places "Eleanor Rigby" as being more covered than it[21]. The Beatles' "Come Together" has also been covered numerous times. George Gershwin's "Summertime" (from Porgy and Bess) is considered a standard, so has been performed in enough versions that an accurate number might be difficult to ascertain. Irving Berlin's "White Christmas" (from the film "Holiday Inn") is well known for having been covered numerous times. According to one estimate "Cry Me a River", written by Arthur Hamilton, had 115 cover versions.

Other songs which have been released many times as cover versions are enumerated in the Wikipedia[22], or, for instance, in the Second Hand Songs database[23] (see also next subsection). We also have lots of examples in the *Music Collection* (section 3.2) used to test our experiments (see also *Appendix A*).

We can see an increasing interest in cover songs just by looking at the emergence of websites, databases and podcasts in the internet (some of them being really extensive) such as the Covers Project[24], coverinfo.de[25], the Second Hand Songs database[26] or Coverville[27] (to cite only a few).

To take an idea about the numbers, we can take as a reference the Covers Project database, which already contains 55907 songs (13563 originals, 40830 cover songs and 1514 songs with samples) and 20722 artists (performers and songwriters), according to it's website[28].

1.4.3 Types of covers

There is a small amount of literature in the field of Music Information Retrieval related specifically to the identification of versions of the same piece using polyphonic audio. This is due partly to the complexity of establishing in a general way what makes a certain piece be a version instead of a different composition. But some early efforts have been done, and is the object of this subsection to review them and to give a little insight on how might be the term considered in the future.

An attempt of identifying different situations where a song was versioned in the context of mainstream popular music was done in [Gómez 06a]. We list them below:

- Re-mastered track obtained after digital re-mastering of an original version.
- Karaoke version or a version of a song translated to a different language.
- Recorded live track: A song or audio sequence recorded from live performances.

[19] http://en.wikipedia.org/wiki/Cover_version
[20] http://www.secondhandsongs.com/song/1409
[21] http://www.secondhandsongs.com/song/2768
[22] http://en.wikipedia.org/wiki/Cover_song#Most_covered_songs
[23] http://www.secondhandsongs.com/stats/song_cover.html#stat
[24] http://www.coversproject.com
[25] http://www.coverinfo.de
[26] http://www.seconhandsongs.com
[27] http://www.coverville.com
[28] Data from http://www.coversproject.com accessed on June 2007.

- Acoustic track: In some situations, the piece is played using different instrumentation than the original song.
- Cover version: A given artist performs a song from a different one.
- Remix: This term stands for a recording produced by combining sections of existing tracks with a new structure and new material. It can also mean to be a radical transformation of the original song.

In [Yang 01], an algorithm based on spectral features to retrieve similar music pieces from an audio database was proposed. This method considered that two pieces were similar if they were fully or partially based on the same score, even if they were performed by different people or at different tempo (a detailed explanation for this and other methods can be found in section 2.4). The author evaluated this approach using a database of classical and modern music, with classical music being the focus of his study. He defined five different types of similar music pairs, with increasing levels of difficulty:

- Type I: Identical digital copy.
- Type II: Same analog source, different digital copies, possibly with noise.
- Type III: Same instrumental performance, different vocal components.
- Type IV: Same score, different performances (possibly at different tempo).
- Type V: Same underlying melody, different otherwise, with possible transposition.

In this thesis, and for the experiments done in next chapters, we have not done a clear classification and distinction between types of covers, but we have considered cover songs in different performing, recording or situational levels. We think that, considering the huge amount of tags and labels (some of them being just buzzwords for commercial purposes) related to covers, and today's understanding of the term *cover version*, the good distinction to make between them should be based on musical characteristics (section 1.4.4) instead of using more commercial, subjective, or situational tags. But, just in order to provide an overview, some categories where a version may fall can be:

1. Remaster: Creating a new master for an album or song. It generally implies some sort of sound enhancement (compression, equalization, different endings (specially fade-outs), etc.) of sound to a previous, existing product, but it is frequently designed to encourage people to buy a new version of something they already own. As an example we could think on a digitally remastered version of a vinyl LP.
2. Instrumental: Sometimes, a version of a song without the lyrics is released. This might include a karaoke version to sing with, or a rare instrumental take of a song in a special collectionist's CD-box edition.
3. Live performance: A recorded live track is a song or audio sequence recorded from live performances. Here, there is a change on the recording conditions, as there is typically the noise of the audience.
4. Acoustic: In some situations the piece is recorded with a different set of acoustical instruments in a more intimate situation. Due to this fact, there are changes in timbre. Also, the same changes noticed in any live performance might be noticed here.

5. Demo: A demo is a way for musicians to approximate their ideas on tape or disc, and provide an example of those ideas to record labels, producers or other artists. Musicians often use demos as quick sketches to share with bandmates or arrangers. In other cases a songwriter might make a demo to send to artists in hopes of having the song professionally recorded, or a music publisher may need a simple recording for publishing or copyright purposes. A demo is a challenging scenario for cover song identification because songs can change entirely.

6. Duet: A successful piece can be often re-recorded or performed by extending the number of lead performer's outside the original members of the band. These songs are usually adapted to the new performer's needs.

7. Medley: Mostly in live recordings, a band covers a set of songs without stopping between them and linking several themes.

8. Remix: This word may be very ambiguous. From a 'traditionalist' perspective, a remix implies an alternate master of a song, adding or subtracting elements, or simply changing the equalization, dynamics, pitch, tempo, playing time, or almost any other aspect of the various musical components. But some remixes involve substantial changes to the arrangement of a recorded work. One paradigmatic case would be electronic remixes of pop songs, where very little of the original recording is kept. Finally, a remix may also refer to a non-linear re-interpretation of a given work or media other than audio. Such as a hybridizing process combining fragments of various works.

A table showing the music characteristics that may change within this categorization is shown in next section. We also refer the reader to the *Music Collection* presentation (section 3.2), and to *Appendix A* for information and detailed statistics related to the music corpus used in the experiments explained in this thesis.

1.4.4 Involved musical facets

Taking the nowadays concept of *cover song*, one might consider the musical dimensions in which such a piece may vary from the original (also named *canonical song*).

In classical music, different performances of the same piece may have subtle variations and differences, including changes in dynamics, tempo, instrumentation, etc. On the other side, in popular music, the main purpose of recording a version can be to investigate a radically different interpretation of the original one. So, important changes and different musical facets might be involved. It is in this scenario where cover song identification becomes a very challenging task.

Some of the main characteristics that might change in a cover song (taking the most wide definition of the term) are listed below:

1. Timbre: The fact that the new performers can be using different instruments, configurations or recording procedures can confer different timbres to the cover version.

2. Tempo: Even in a live performance of the original artist performing a *canonical version* of a previously released song, there might be a slight (or not so small) change in tempo. It is not so common to play with a metronome in a concert, and this might go in detrimental of expressiveness and contextual feedback. Even in classical music, small tempo fluctuations are done between different renditions of the same piece.

3. Rhythm: The base rhythm changes sometimes depending on the performer's intention or feeling. Not only changes the drum section, but more subtle changes can happen: the meter, a swinging pattern, syncopation, etc.

4. Structure: It is a quite common thing to change the structure of the song. This change can include just eliminating a short intro to the piece or shortening an instrument solo section. But it can imply a radical change in the section or the chorus inter or intra-ordering.

5. Key: The piece can be transposed to a different key (or tonality). This is usually done to adapt the pitch range to a different singer or instrument.

6. Harmonization: Although maintaining the key, the chord progression might change (adding or deleting chords, substituting them by relatives, adding tensions, etc.). This is very common in intros and bridges. Also, in instrument solo parts, the lead instrument voice is practically always different from the original one.

7. Lyrics and language: We have seen that one purpose of performing a cover song is for translating it to other languages. This is commonly done between high-selling artists and big speaking communities.

Also, as in the case of some remixes, the whole characteristics of the song might change, except, perhaps, a lick or a phrase that is on the background of the listening experience, and that it is the only thing that reminds the original song. In this case, it is sometimes very difficult for a human to recognize the original song, even if he/she is familiarized with it.

To have a more clear idea about the musical aspects that may change in some cover categories, in table 1.1 we summarize them within the categories highlighted in previous subsection.

	Timbre	Tempo	Rhythm	Structure	Key	Harm.	Lyrics
Remaster	√						
Instrumental							√
Live performance	√	√					√
Acoustic	√	√	√	√		√	
Demo song	√	√	√	√	√	√	√
Duet	√			√	√	√	√
Medley	√	√	√	√	√		
Remix	√	√	√	√	√	√	√

TABLE 1.1
Involved musical facets of some of the predefined cover song categories (subsection 1.4.3).

Finally, we have to remember that, if we want to consider a cover song in the most general meaning of the term, we have to consider (at least) all the changes mentioned previously in this section.

1.5 Cover song identification in the MIR community

In this section we return to the concept of music similarity within some areas of Music Information Retrieval (MIR).

A multi-faceted concept like music similarity can be viewed from multiple angles: audio artist identification, audio artist similarity, audio genre classification, audio music mood classification, audio fingerprinting, etc., just to name a few from the point of view of the MIR community.

Within this huge spectrum of previous work covering a wide range of music-similarity tasks, it is suitable to emphasize two of them: audio fingerprinting and genre recognition. The work in this thesis falls in the middle of them.

Audio fingerprinting tries to identify a concrete song with a particular performance of an artist in a radio broadcast audio stream or in a database. These techniques [Herre 01, Cano 02a, Cano 02b, Venkatachalam 04] typically rely on spectral representations, which are processed to be resistant to various types of noise and are unique for each piece of music. A query results in a match only if (almost) exactly the same piece of music resides in the queried database. Fingerprinting finds the most salient portions of the musical signal and uses detailed models of the signal to look for exact matches.

Genre classification methods output is much less concrete. They try, in essence, to group songs according to a commercially established label, the genre, where certain characteristics might be more or less the same (see section 2.3.2). Genre classification techniques [Tzanetakis 02b, Aucouturier 03, Pampalk 05] have traditionally relied on Mel Frequency Cepstral Coefficients (MFCC) and other low-level timbre descriptors. This techniques use general models such as probability densities of acoustic features approximated by Gaussian Mixture Models (GMM). These so-called bag-of-feature models [Aucouturier 03] ignore the temporal ordering inherent in the signal and, therefore, we might expect that they are not able to identify specific content within a musical work such as a section of a song.

As a global view, reliable acoustic fingerprints are specific enough to identify a particular performance from among all the recordings of audio that a person or group have made, while genre classification techniques do not care at all about specificity. Cover song identification would not be as specific as audio fingerprinting, nor as general as genre classification (figure 1.1).

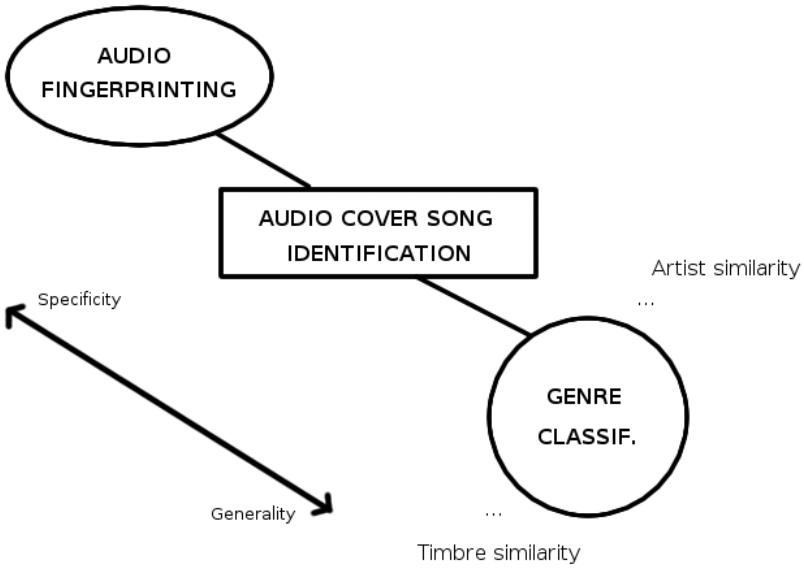

FIGURE 1.1 *Specificity of cover song identification. The most specific target answers retrieved are on the top left corner, while the more generic are on the bottom right part of the figure. Cover song identification fails in between, without properly not being considered as an audio fingerprinting nor a genre classification task.*

Other related tasks are artist similarity [Ellis 02] or timbre similarity [Aucouturier 06]. In section 2.3 we refer to them into much detail and give a more extensive review on audio fingerprinting and genre classification techniques.

When dealing with huge music collections, cover version identification is a relevant problem, because it is usual to find different versions of the same song, and, at the same time, it provides a direct way for evaluating musical similarity.

Besides the MIR community, audio cover song identification is an attractive topic of research from the perspective of intellectual property rights and plagiarism. The problem has conceptual links with the problem of analogy in human cognition, which is also and intriguing and far from being understood topic.

1.6 Sequences of tonal descriptors

We have seen in sections 1.4.3 and 1.4.4 that the most challenging scenario for cover song identification might be in popular music, where several musical facets of the song may change radically. Thus, for the problem of version identification, representations and matching schemes that are robust to changes in tempo, instrumentation, and general musical style must be considered.

Cover songs typically retain the essence of the melody. But they may vary greatly in other dimensions such as instrumentation (or timbre), tempo, rhythm, chord voicings and so forth. The lyrics and the language also might change. A robust mid-level representation that is largely preserved under the musical variations mentioned are *tonal sequences* (or *harmonic progressions*). Tonality is ubiquitous and most listeners musically trained or untrained can identify the most stable pitch while listening to tonal music. Furthermore, this process is continuous and remains active throughout the sequential listening experience.

Tonal sequences can be understood as series of different notes played sequentially. These notes can be unique for each time slot (a melody) or can be played jointly with others (chord or harmonic progressions). The most general case is this last one: chord progressions are central to most modern European-influenced music and the principle study of harmony. Generally, successive chords in a chord progression share some notes, which provides harmonic and linear (voice leading) continuity to a passage. Also, chord progressions are usually associated with a scale and the notes of each chord are usually taken from that scale (or its modally-mixed universe)[29].

We can find more reasons for looking at the temporal events evolution by looking at a very related field such as melody identification from the point of view of music perception. The fundamental units for melody identification appear to be contour-based, holistic properties that stretch across several notes up to and including phrases [Schulkind 03]. Musical knowledge does not appear to be compositional; that is, melodies are not built up from their constituent components (individual notes). Recognizing a well-known melody is not an all-or-one process. Instead, recognition processes develop progressively while the melody unfolds over time. Also, researchers have shown that changes in melodic contour (i.e., the sequence of ups and downs in a melody, regardless of interval size) and in tonality information are both detrimental to recognition [Dalla Bella 03].

From the point of view of the MIR community, clear insights about the importance of temporal features in a music similarity task have been evidenced, for instance, in [Cano 02b, Dannenberg 03, Adams 04] or in [Casey 06a] (see also section 2.4).

The use of tonal descriptors to find similar pieces is also well documented in the MIR literature. For instance, [Sheh 03] report the usefulness of Pitch Class Profile (PCP) features for sequence recognition, and, in [Gómez 06b] we can find an analysis of how tonal descriptors are useful to locate versions of the same song (we comment a large corpus of references using these descriptors in section 2.5).

[29] http://en.wikipedia.org/wiki/Harmonic_progression

1.7 Goals and organization of the thesis

Our main goals can be summarized as:

- Highlight the scientific background of cover song identification and make a literature summary.
- Discuss state-of-the-art systems.
- Establish an evaluation methodology.
- Compile a big and suitable cover song music collection.
- Experiment with state-of-the-art methods and identify relevant aspects of them.
- Improve these state-of-the-art methods.
- Propose and test a new method for determining tonal sequence similarity and apply it to cover song identification.
- Extract conclusions and discuss each relevant point.

We first want to highlight the scientific background that is involved with tonal sequence similarity and with cover song identification. This is done in chapter 2, where we explain the main descriptors and alignment techniques used. We also give an overview on the existing literature and methods regarding closely related tasks such as audio fingerprinting and genre classification (section 2.3), but it is in the existing cover song identification systems where more emphasis is put (section 2.5). Finally, a section is devoted to comment and discuss the most relevant aspects regarding these state-of-the-art systems (section 2.6). This section acts as a starting point of the research carried along this thesis.

For assessing the performance of a cover song identification system, an evaluation methodology has to be established. In chapter 3 we do that while discussing the suitability of certain evaluation measures. We also present the process that will be followed for accomplishing that. In addition, we perform a base-line experiment in order to assess further relevance of obtained results with other approaches.

We have to say that one of our first goals was to compile a big and suitable music collection to be used along the experiments performed. This is presented in section 3.2 and is listed in *Appendix A*.

In chapter 4, several experiments with state-of-the-art methods are performed. The objectives are twofold: we want to check which performance we can achieve with our music collection (sections 4.1 and 4.3), and we want to test which improvements can be beneficial for these systems (sections 4.2 and 4.4). A final discussion section on the most important aspects of the performed experiments and obtained results is done in section 4.5.

Then, a new method for determining tonal sequence similarity applied to covers is fully explained (chapter 5). This mainly puts forward new ways of addressing the problems stated in previous sections, the most important ones being: the proposal of a new distance to assess the resemblance between HPCP feature vectors, and the use of a local alignment algorithm (all in section 5.1.2). Furthermore, we see that we reach very significant improvements in performance compared to other implemented and state-of-the-art methods (section 5.2.1).

Finally, in chapter 6, we highlight some conclusions and open issues that remain of interest for future work.

Chapter 2

Scientific Background

A review of the main features, techniques and works relevant to the task of content-based music similarity focused on tonal sequences is done in this chapter. A special emphasis is put on audio cover song identification, with a detailed review of most relevant literature until now (section 2.5).

2.1 Descriptors

In this first section, we present the features used in the majority of the state-of-the-art approaches, ranging from low-level descriptors to more musically-meaningful ones.

To compute similarities it is necessary to extract features (also known as descriptors) from the audio signal. The basic idea is to transform the raw audio data so that the interesting information is more easily accessible. This often means drastically reducing the overall amount of data. However, this needs to be achieved without losing too much of the information which is critical to human perception.

Note that this section is not intended to be a full compendium of music similarity descriptors. We focus on the features, concepts and algorithms related to the present work are exposed.

2.1.1 Low-level descriptors

If we study the content related to a piece of music, we identify different levels of abstraction, as well as many description facets. Any of these levels of abstraction or description facets might be useful for some users (for instance, to a naive listener or to a musicologist). Regarding the abstraction levels, we usually distinguish between low, mid and high-level descriptors [Lesaffre 05].

The term low-level is usually employed to denote features that are closely related to the raw audio signal, which are computed in a direct or derived way. Most of these descriptors do not have much sense for the majority of the users, but they are easily used by computational systems.

Energy

Acoustic energy is a straightforward descriptor. This is used in some literature to obtain an 'energy profile' of the song for comparison between them. The energy profile is a representation of the average acoustic energy versus time. This is determined by computing the Root Mean Square (RMS) signal power across a time window and dividing by its length. So:

$$RMS = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} [f(t)]^2} \tag{2.1}$$

Where T_1 and T_2 are the starting and ending points of the time window respectively. With a sampled signal the above formula becomes:

$$RMS = \sqrt{\frac{f_s}{n_2 - n_1} \sum_{n=n_1}^{n_2} [x(n)]^2} \tag{2.2}$$

Where f_s denotes the sampling rate, $x(n)$ corresponds to the sampled signal, and n's represent sample numbers ($n_2 - n_1$ equates the energy windowlength in samples).

Timbre

Numerous research efforts have already been devoted to timbre similarity in music, in which timbre refers to the spectral information that correlates with instrumentation and articulation in musical performance. Although we have argued that timbre features might not be good descriptors to deal with cover song similarity (section 1.4.4), we here give a brief overview, as some of them are mentioned (and used) in the literature.

Traditionally, the role of timbre descriptors have been to try to characterize the spectrum through a small corpus of features such as the *Spectral Centroid* (the barycenter point of the spectral distribution), *Spectral Flux* (frame-to-frame spectral difference), *Zero Crossing Rate* (ZCR, the number of time-domain zero crossings), *Spectral Roll-off* (the frequency below which some percentage of the spectrum (i.e. 85%) resides), and several low-order statistics based on these [Lambrou 98, Tzanetakis 01, Aucouturier 03].

More recently, the most used features for timbre description have been the Mel Frequency Cepstral Coefficients (MFCCs) [Rabiner 93]. These are a standard pre-processing technique in speech processing. They were originally developed for automatic speech recognition [Oppenheim 69], but have proven to be useful for MIR [Foote 97, Logan 00a]. A sketch of the calculation of the MFCC would be [Logan 01]:

1. Divide the signal into (windowed) frames.
2. For each frame, obtain the amplitude spectrum (with a Discrete Fourier Transform).
3. Take the logarithm of the amplitude.
4. Map the amplitudes obtained above onto the Mel scale, using triangular overlapping windows.
5. Take the Discrete Cosine Transform (DCT).

The MFCCs are the amplitudes of the resulting spectrum. They represent, in a very uncorrelated way, the information in the spectrum. There are several variants for calculating them, some of them and particular details on the parameters are given in [Zheng 01].

The Mel-scale is approximately linear for low frequencies (below 500 Hz), and logarithmic for higher frequencies. The reference point to the linear frequency scale is a 1000 Hz tone, which is defined as 1000 Mel. A tone with a pitch perceived twice as high is defined to have 2000 Mel, a tone perceived half as high is defined to have 500 Mel, and so forth. The Mel-scale was defined by [Stevens 37]:

$$m = 1127.01048 \cdot log\left(1 + \frac{f}{700}\right) \tag{2.3}$$

Where f is the frequency in Hz and m is the result in Mel. This scale has perceptual origins, since it is used to better model the way humans perceive pitch height relations.

The power spectrum is transformed to the Mel-scale using a filter bank consisting of triangular filters. Each triangular filter defines the response of one frequency band and is normalized such that the sum of weights for each triangle is the same. The triangles overlap each other such that the center frequency of one triangle is the starting point for the next triangle, and the end point of the previous one.

The Discrete Cosine Transform (DCT) is applied to compress the Mel power spectrum. A side effect of the compression is that the spectrum is smoothed along the frequency axis which can be interpreted as a simple approximation of the spectral masking in the human auditory system.

A simplification of the MFCCs commonly used to represent timbre in speech recognition and in some music tasks are the Log-Frequency Cepstral Coefficients (LFCC) [Logan 00b], which are not mapped into the Mel scale.

For the details on the computation of the MFCCs we refer to [Rabiner 93, Logan 01, Pampalk 06].

Tonality

In section 1.6 we have argued about the convenience for our scenario to use sequences of tonal descriptors. This subsection deals with literature on the extraction of these.

In the MIR community, tonal descriptors are often referred to as *chroma features* or *Pitch Class Distributions* (PCD) or *Pitch Class Profiles* (PCP). This terms are sometimes used to refer to the same concept, but they usually rely on very different implementations that try to obtain a similar result: a vector of features describing the different tones (or pitches) in an audio signal, excerpt, or frame.

With the western tonal system (12 semitones), we can have a 12-component vector with values expressing the amount of energy found in the analyzed audio for each semitone. Sometimes, the magnitude is seen as a statistical distribution of pitches (what would formally be a 12-bin *chroma histogram*). Also, a finer analysis is usually done, getting 24 or 36-bin chroma feature vectors (with an interval resolution of 1/2 and 1/3 of a semitone respectively). We can see an example of a chroma feature vector in figure 2.1.

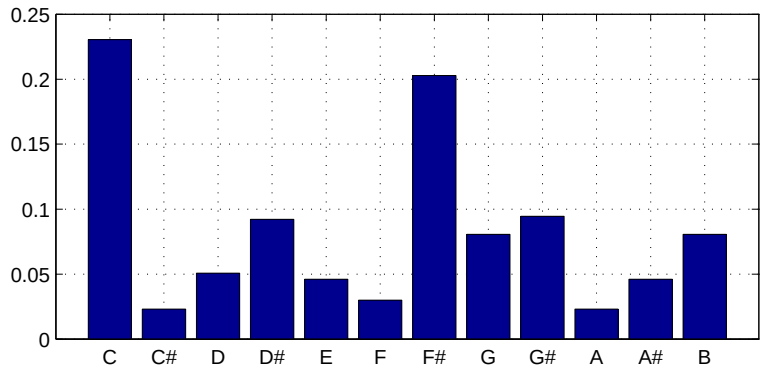

FIGURE 2.1 *12-bin chroma feature vector example. The x-axis values correspond to pitches C, C♯, D, D♯, E, and so forth.*

According to [Gómez 06a], reliable pitch class distribution descriptors should fulfil the following requirements:

1. Represent the pitch class distribution of both monophonic and polyphonic signals.
2. Consider the presence of harmonic frequencies.

3. Be robust to noise that sounds at the same time: ambient noise (e.g. live recordings), percussive sounds, etc.

4. Be independent of timbre and played instrument, so that the same piece played with different instruments has the same tonal description.

5. Be independent of loudness and dynamics.

6. Be independence of tuning, so that the reference frequency can be different from the standard A 440 Hz.

We should note that all these properties are highly indicated for a cover song identification task (see section 1.4.4).

All the approaches for computing an instantaneous evolution of pitch class distributions follow the same schema shown in figure 2.2. The nomenclature for this type of features is quite varied. A first approach for key induction from audio was proposed by [Leman 91, Leman 95], where a set of *pitch patterns* were extracted. The use of *pitch histograms* appears later in [Tzanetakis 02a]. In [Fujishima 99] a system for chord recognition based on the *pitch-class profile* (PCP) is proposed. PCP, as formulated by Fujishima, is a twelve dimensional vector representing the intensities of the twelve semitone pitch classes. The approach presented in [Gómez 06a] uses an extension of the PCP, called *Harmonic Pitch Class Profile* (HPCP). These last are the main tonality descriptors that will be used in this thesis. Constant Q profiles have also been used to characterize the tonal content of audio, as in [Purwins 00]. The chromagram is another relevant feature ([Paws 04]).

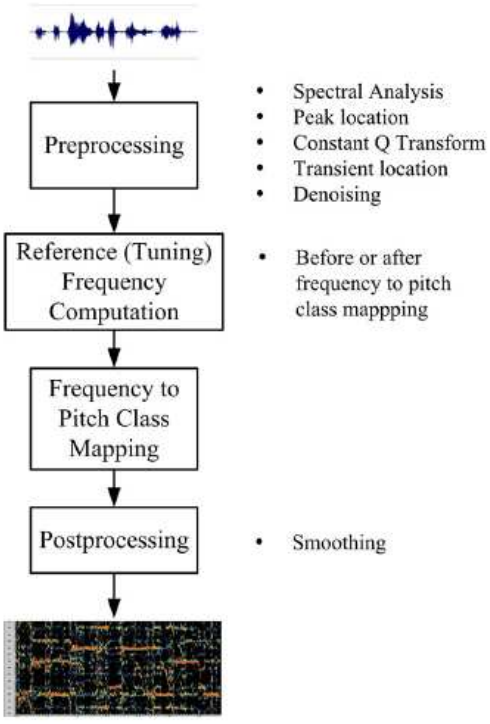

FIGURE 2.2 *General block diagram for methods for pitch class distribution computation from audio. Figure extracted from [Gómez 06a] with permission of the author.*

In subsequent paragraphs we analyze the different approaches for each of the steps of the general schema of figure 2.2 for tonality induction from audio. Following [Gómez 06a], we now elaborate some comments on it.

The main task of the pre-processing step is to prepare the signal for pitch class distribution description and enhance the features that are relevant for this kind of descriptors. Pre-processing should help in having robustness to noise (ambient noise, percussive sounds, speech sounds, etc.). The majority of the approaches found in the literature are based on performing a frequency analysis of the audio signal. In [Fujishima 99], the input sound is transformed to a Discrete Fourier Transform (DFT) spectrum (defining analysis frames of 2048 samples under 5.5 KHz, with a duration of 400 ms). DFT is also used in [Gómez 06a, Gómez 06b], and also in [Paws 04], using similar window duration. It is also common to restrict the analysis to a given frequency region in order to eliminate non audible frequencies and to select the most relevant ones for pitch distribution descriptors. For instance, [Fujishima 99] selects a frequency region between 63.5 and 2032 Hz, [Gómez 06a] uses only a frequency region of the signal between 100 Hz and 500 Hz, or [Paws 04, Gómez 06b] uses frequencies between 25 and 5000 Hz.

Another way of performing frequency analysis is by using the constant-Q transform, which was introduced by [Brown 91]. Later, [Brown 92] proposed an algorithm for its efficient computation. This transform has been employed by [Purwins 00, Purwins 05] for low-level tonal description. The basic formulation for the constant-Q transform is:

$$X[k] = \sum_{n=0}^{N[k]-1} w[n,k] \cdot x[n] \cdot e^{-jw_k n} \tag{2.4}$$

With it, a discrete frequency spectrum ($X[k]$) of the signal with a constant resolution frequency axis is obtained by changing the window size ($w[n,k]$) for every bin (k) depending on the resolution desired. It can also can be seen as an ideal constant bandwidth filter.

In addition to a frequency analysis of the input signal, other pre-processing steps mentioned by [Fujishima 99] include non-linear scaling and silence and attack detection to avoid noisy features; they are not further explained nor evaluated in his work. Transient detection is also used in HPCP computation [Gómez 06a] as a preprocessing step, and therefore, in the basic descriptors of this thesis.

Another interesting pre-processing in order to reduce noise is the inclusion of a peak selection routine only considering the local maxima of the spectrum [Gómez 06a, Gómez 06b]. Finally, [Paws 04] also mentions an enhancement procedure of the spectral components to cancel spurious peaks, although this procedure is not explained either in his paper.

Regarding reference frequency computation, the A 440 Hz is considered as the standard reference frequency for pitch class definition. According to this, the majority of approaches for pitch class distribution computation use a fixed frequency as a reference [Fujishima 99, Purwins 00, Paws 04], which is usually set according to the A 440 Hz. Nevertheless, this is a strong assumption that cannot be accepted: some pieces might not be tuned to 440 Hz, as it is the case for most of the Early Music (Medieval, Renaissance, ...). It also often happens in popular music. So, we should estimate the tuning frequency in order to assure robustness to tuning. This procedure can be performed either above or after the computation of pitch class distribution, and the final feature vector must be adjusted to this reference frequency [Gómez 06a].

Once the reference frequency is known and the signal is converted into a spectrogram by means of the DFT or a constant-Q analysis, there is a procedure for determining the pitch class values from frequency values. The approaches from [Leman 95] and [Tzanetakis 02a] are oriented to multipitch estimation, and a periodicity analysis is applied. Then, the predominant frequencies are detected and

matched to pitch classes using log mapping with respect to the reference frequency. Instead of using only predominant pitches, [Fujishima 99] considers all the frequencies of the DFT, where the weight of each frequency to its corresponding pitch class is given by the square of the spectral amplitude. The approach in [Gómez 06a] introduces a weighting scheme using a cosine function. It also considers the presence of harmonics.

As it is usual in fundamental frequency estimation methods, some post-processing techniques are used after computing the pitch class distribution. One of the mentioned requirements for pitch class distribution features is the robustness against variations on dynamics. This is usually obtained by normalization. [Gómez 06a] proposes to normalize the PCP vector for each frame by its maximum value.

Finally, we now provide some details for the particular feature extraction algorithm that we have used in the experiments for this thesis. In figure 2.3, we show a general block diagram for HPCP computation.

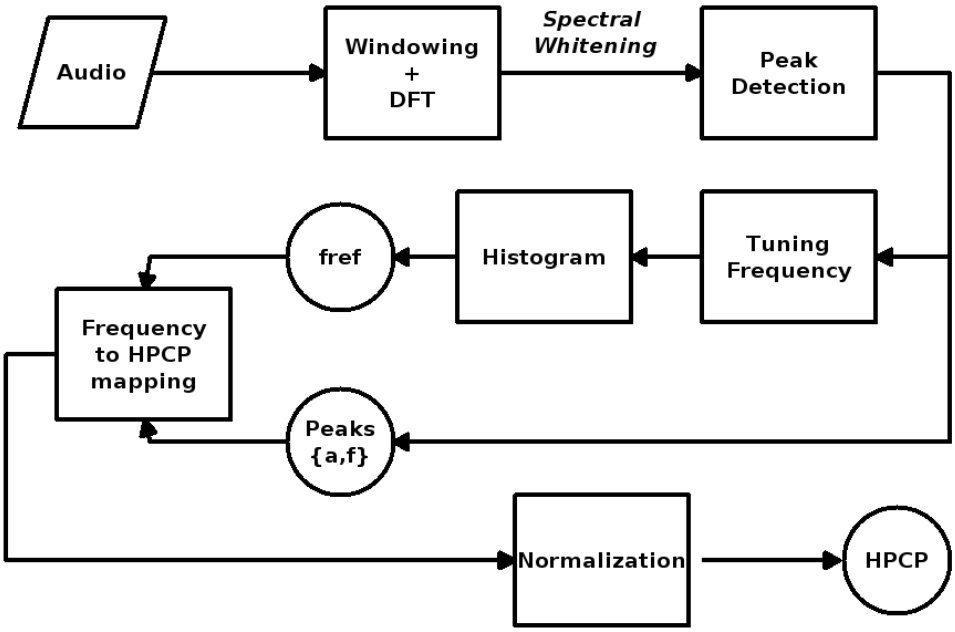

FIGURE 2.3 *Basic block diagram for HPCP computation.*

In the HPCP computation, the signal is cut into short overlapping frames around 93 ms. with a 50% overlapping. A Blackmann-Harris window of length 93 ms is used for windowing. After this procedure, a Discrete Fourier Transform (DFT) is performed.

A *Spectral Whitening* procedure is performed after the DFT computation. With this procedure, the spectrum is normalized according to its spectral envelope, in order to convert it to a flat spectrum. Using this timbre normalization, notes on high octaves contribute equally to the final HPCP vector than those on low pitch range, and the results are not influenced by different equalization procedures.

In the *Peak Detection* module, the local maxima of the spectra are extracted. These represent the harmonic part of the spectrum. The method will work from now on only with the spectral peaks (notice that we keep track of the magnitude a and position f for each peak i (see figure 2.3)). The frequency range for peak detection ranges from 40 Hz to 5 KHz, and the number of peaks tracked is 8.

The *Frequency Tuning* is done by computing the deviation of frequency values with respect to the A 440 Hz mapped to a semitone scale, and computing a histogram of values. Then the tuning frequency is defined to be the maximum value of the histogram (f_{ref}).

The module named *Frequency to HPCP mapping* includes a logarithmic mapping proportional to the HPCP size. In the case of 12 semitones, this mapping would be:

$$n = 12 \cdot log_2\Big(\frac{f_i}{f_{ref}}\Big) \tag{2.5}$$

Where n is the corresponding bin of the HPCP. The HPCP vector is defined by the following formula:

$$HPCP(n) = \sum_{i=1}^{nPeaks} w(n, f_i) \cdot a_i^2 \tag{2.6}$$

Where a_i corresponds to the magnitude (in linear units) and f_i corresponds to the frequency (in Hz) of a spectral peak. *nPeaks* corresponds to the total number of peaks considered. The contribution of each peak is defined to be the square of it's magnitude (a_i^2). Also, a cosine weighting function $w(n, f_i)$ is introduced, so that each frequency f_i contributes to the HPCP bin(s) that are contained in a certain window around this frequency value. For each of those bins, the contribution of the peak i (the square of the peak linear amplitude a_i^2) is weighted using a cos^2 function around the frequency of the bin. The value of the weight depends on the frequency distance between f_i and the center frequency of the bin n, f_n, measured in semitones, as follows:

$$w(n, f_i) = \begin{cases} cos^2(\frac{\pi}{2} \cdot \frac{d}{0.5 \cdot l}) & \text{if } |d| \leq l/2, \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

$$d = 12 \cdot log_2(\frac{f_i}{f_n}) + 12 \cdot m \tag{2.8}$$

Where m is the integer that minimizes the module of the distance ($|d|$) and l is the window size in semitones.

The contribution of other harmonics is also taken into account. In order to make harmonics contribute to the pitch class of its fundamental frequency, a weighting procedure is introduced: each peak frequency f_i has a contribution to the frequencies having f_i as a harmonic frequency (f_i, $f_i/2$, $f_i/3$, $f_i/4$,... , $f_i/nHarmonics$). This contribution decreases exponentially along frequency.

Finally, HPCPs are normalized dividing by their maximum value.

A more extensive explanation on the procedure of extracting HPCPs from audio can be found in [Gómez 06a]. We refer to it for further details on the computation of them. As well, a more extensive literature review is also done in the same reference.

2.1.2 Musical descriptors

We can also consider other 'musically meaningful' descriptors. These would be enclosed in the mid or high levels mentioned in section 2.1.1. There are many of them of interest to the MIR community, but here we only consider the most relevant for the task proposed, or the ones used in the cover song identification literature (section 2.5).

Beat

Although tapping while listening to music may seem natural for a human listener, automatic beat (or tempo) tracking is not a trivial task for a computer, especially in the case of expressive interpretations and tempo changes.

Rhythmic structures can be very complex, but three layers are mainly described in the literature. The *tatum* is a time quantum dependent on the musical context, and is the shortest time interval found between note onsets creating the perception of a rhythm. The *tatum* provides a fine grained rhythmical grid at around 40 ms to 100 ms resolution, related to the minimum perceptible inter-onset interval. The *tactus*, or most commonly referred to as beat period, is the foot taping rate, with typical values ranging from 200 ms to 1.5 seconds. The beat period is usually found to be at multiple values of the *tatum*. At a higher level, the measure is related to the time signature of the piece and often corresponds to harmonic changes and rhythmic variations. The metre is the phenomenon of entrainment which arises in a musical context from the combination of these patterns in rhythms. Music performances are not perfectly isochronous, and a system to extract beat locations from real musical audio must take into account important deviations from one beat to another.

A review of several beat detection algorithms is given in [Hainsworth 04], and more recently in [Gouyon 06]. Following these references, a major distinction between the algorithms can be

- Rule-based: Rule-based approaches were among the earliest used when computers were not capable of running complex algorithms. They tend to be simple and encode sensible music-theoretic rules. The most classic references are [Longuet-Higgins 82, Temperley 99].

- Autocorrelative: Autocorrelation seems a plausible method for finding periodicities in data and has hence been used in several studies. Without subsequent processing, it can only find tempo and not the beat phase. These methods are not causal as they use all the data before making any decisions. Also, they can be susceptible to poor performance on variable tempo examples as the autocorrelation function becomes highly spread out. An example of this can be [Davies 04, Brossier 07].

- Oscillating filters: Here, a series of oscillators is excited by a signal, and the filter which corresponds best to the frequency of the data receives the highest excitation. Beat location can be calculated by examining the phase of the oscillator. This method is particularly suited to causal analysis. Oscillating filters have also been used for tracking beat structures, once the tempo is known. Various people have suggested that this implementation has psychological plausibility and this is their motivation for its use. In [Scheirer 98], psychoacoustically motivated amplitude envelopes detected in several bands as the input of a bank of comb filters are used. The outputs of each filter are summed together and the function obtained is searched for peaks, corresponding to the best tempo estimates.

- Histogramming: Several approaches have focused on audio beat tracking using histogramming of inter-onset intervals. First, the signal is analysed to extract onsets before the subsequent processing takes place. This method has similarities to the autocorrelation approaches and produces a similar output, though with a discrete input. The *BeatRoot* algorithm described in [Dixon 01] uses an onset detection function to detect onset times in a first pass, then finds different beat period hypothesis by constructing a histogram of the interonset intervals. The locations of beats are inferred by looking for sequences of events which match one of the period hypothesis and align to the onset times.

- Multiple agent: Multiple 'agent' methods have been developed in the area of computer science architecture, but often the models underlying them bear significant resemblance to the more rigorously probabilistic approaches described below. The philosophy is to have a number of agents or hypotheses which track independently; these are scored with their match to the data, and low scoring agents are killed while high scoring ones may be branched to cover differing local hypotheses. When processing reaches the end of the signal, the 'agent' with the highest score wins and is chosen. [Goto 95, Goto 01] uses multiple 'agents' with different strategies to detect temporal features such as onset times, inter-onset intervals and pre-defined spectral templates. See also [Allen 90] or [Rosenthal 94].

- Probabilistic: As mentioned above, probabilistic approaches have similarities to multiple agent approaches with the difference being that the framework used is fully stochastic in nature. There is an underlying model specified for the rhythm process, the parameters of which are then estimated by the algorithm. This allows the use of standard estimation procedures such as the Kalman filter [Blackman 99] or Markov chain Monte Carlo methods [Gilks 96]. In [Raphael 01a, Raphael 01b] an automatic accompaniment system capable of following tempo changes is described, and in [Klapuri 03], a probabilistic modelling of musical rules is used to infer the three layers of rhythmic structure from acoustic signals: tatum, tactus and measure.

We further refer to [Gouyon 05] for an extensive literature review on different beat tracking algorithms and other rhythm-related issues.

Regarding the experiments done for this thesis with a beat tracker, we have used an implementation[1] of the algorithm proposed in [Davies 04, Davies 05b]. In these articles, an algorithm for efficient causal beat tracking of musical audio and further improvements are described. The algorithm is based on the calculation of an autocorrelative function. Correlation functions compare the similarity between two signals on a sample-by-sample basis. The autocorrelation function compares the signal with delayed versions of the same signal. Different versions of the ACF have been proposed. The modified autocorrelation of a discrete signal x_t may be defined as:

$$r_t(\tau) = \sum_{j=t+1}^{t+W} x_j \cdot x_{j+\tau} \tag{2.9}$$

Where $r_t(\tau)$ is the modified autocorrelation function of lag τ at time index t. W is the time length we want to consider. With a periodic input signal, this function produces peaks at integer multiple of the period.

A description of the forementioned algorithm and its software implementation in real-time is found in [Davies 05a, Brossier 07].

Chord

Chromagram or Pitch Class Profile (PCP, see section 2.1.1) based features have been almost exclusively used as a front end to the chord recognition or key extraction systems from the audio recordings.

A system that automatically extracts from audio recordings tonal metadata such as chord, key, scale and cadence information was presented in [Gómez 04]. They used HPCP as the feature vector, and correlated it with a chord or key model adapted from [Krumhansl 90]. Similarly, [Paws 04] used the maximum-key profile correlation algorithm to extract key from the raw audio data, where he averaged

[1] http://aubio.piem.org

the chromagram features over variable-length fragments at various locations, and correlated them with the 24 major/minor key profile vectors derived by [Krumhansl 90]. [Harte 05] used a 36-bin chromagram to find the tuning value of the input audio using the distribution of peak positions, and then derived a 12-bin, semitone-quantized chromagram to be correlated with the binary chord templates. In [Lee 06a], an Enhanced Pitch Class Profile (EPCP) was introduced for automatic chord recognition from the raw audio.

An alternative way of embedding the idea of *harmonic progression* (see section 1.6) into the estimation is by using Hidden Markov Models (HMM, see also section 2.2.3). The states in the HMM represent chord types, and they try to find the optimal path (chord sequence) in a maximum likelihood sense. The work by [Raphael 03] is a good example of successfully using HMMs for harmonic analysis. Another approach is shown in [Sheh 03], where an HMM is used on PCP features estimated from audio. Both the models for chords (this last work used 147) and transitions, are learned from random initializations using the expectation maximization (EM) algorithm [Dempster 77]. [Bello 05] also used the HMMs with the EM algorithm, but they incorporated musical knowledge into the models by defining a state transition matrix based on the key distance in a circle of fifths, and by avoiding random initialization of a mean vector and a covariance matrix of observation distribution, which was modeled by a single Gaussian. In addition, in training the model for parameter estimation, they selectively update the parameters of interest on the assumption that a chord template or distribution is almost universal, thus disallowing adjustment of distribution parameters. In [Lee 06c], another approach based on this latter work is presented.

Melody

As some methods for cover song identification are based on retrieving songs by melodic similarity, we review here some of the most recent literature on that.

Melody extraction is strongly related to pitch tracking, which itself has a long and continuing history (for reviews, see [Hermes 93, De Cheveigne 05]). In the context of identifying melody within multi-instrument music, the pitch tracking problem is further complicated because although multiple pitches may be present at the same time, at most just one of them will be the melody. Thus, in essence, all approaches to melody transcription face two problems: identifying a set of candidate pitches that appear to be present at a given time, and then deciding which (if any) of those pitches belongs to the melody. There are also two important tasks to be performed within melody extraction: detecting whether the melody is active or silent at each time, and the post-processing of a sequence of melody estimates (typically to remove spurious notes or otherwise increase smoothness).

All the fundamental frequency estimation algorithms give us a measure corresponding to a portion of the signal (analysis frame). According to [Hess 83], the fundamental frequency detection process can be subdivided into three main steps that are passed through successively: the preprocessor, the basic extractor, and the postprocessor. The basic extractor performs the main task of measurement: it converts the input signal into a series of fundamental frequency estimates. The main task of the pre-processor is data reduction in order to facilitate the fundamental frequency extraction. Finally, the post-processor is a block that performs more diverse tasks, such as error detection and correction, or smoothing of an obtained contour.

Getting a bit more into detail, the description of existing (and more recent) algorithms can be broken down into several key dimensions [Polliner 07]:

- Front-end: This concerns the initial signal processing applied to input audio to reveal the pitch content. The available algorithms can be classified in different ways. For example, it is useful

to distinguish them according to their processing domain and separate the *time-domain* from the *frequency-domain* algorithms. This separation is not always that clear, since some of the algorithms can be expressed in both (time and frequency) domains. This is the case of the Autocorrelation Function (ACF) method. Zero-crossing rate (ZCR) is among the first and simplest techniques for estimating the frequency content of a signal in time domain, and consists in counting the number of times the signal crosses the 0-level reference in order to estimate the signal period. This method is very simple and inexpensive but not very accurate when dealing with noisy signals or harmonic signals where the partials are stronger than the fundamental. Time-domain Autocorrelation function (ACF) based algorithms have been among the most frequently used fundamental frequency estimators [Paiva 04, Vincent 05]. The ACF of a sequence $x(n)$ of length K is dened as:

$$r(n) = \sum_{k=0}^{K-n-1} x(k) \cdot x(k+n) \tag{2.10}$$

Where k is an index that goes from 0 to the end of the sequence $x(n)$ (K).

The ACF of a sequence $x(n)$ can be computed in both time and frequency domains, and the maximum of this function corresponds to the fundamental frequency for periodic signals. A popular technique is to take the magnitude of the Short-Time Fourier transform (STFT) and visualize a spectrogram (the DFT of successive windowed consecutive excepts of audio). With this, pitched notes appear as more-or-less stable harmonics. It is also common to reduce the magnitude spectra of the STFT by keeping only track of the sinusoidal frequencies estimated as relating to prominent peaks in the spectrum [Goto 04, Marolt 04, Dressler 05]. Other techniques include envelope analysis [Meddis 91], Cepstrum analysis [Noll 67], or bandwise processing algorithms [Klapuri 04].

- Multi-pitch: This dimension addresses how the systems deal with distinguishing the multiple periodicities present in the polyphonic audio and how many simultaneous pitches can be reported at any time. For systems based on STFT, the problem is to identify the sets of harmonics and properly credit the energy or salience of each harmonic down to the appropriate fundamental even though there need not be any energy at that fundamental for humans to perceive the pitch. One weakness with this approach is its susceptibility to reporting a fundamental one octave too high, since if all the harmonics of a fundamental frequency f_0 are present, then the harmonics of a putative fundamental $2f_0$ will also be present. In [Goto 04], a technique for estimating weights over all possible fundamentals is proposed. It tries to jointly explain the observed spectrum, which effectively lets different fundamentals compete for harmonics, based on Expectation-Maximization (EM) re-estimation of the set of unknown harmonic-model weights; this is largely successful in resolving octave ambiguities. In [Marolt 04], this procedure is slightly modified in order to consider only fundamentals that are equal to, or one octave below, actual observed frequencies, and then integrates nearby harmonics according to perceptual principles. [Polliner 05] takes a different approach consisting of feeding the entire Fourier Transform magnitude at each time slice, after some local normalization, into a Support Vector Machine (SVM) classifier. This approach ignores prior knowledge about the nature of pitched sounds, on the principle that it is better to let the machine learning algorithm figure this out for itself, where possible. The classifier is trained to report only one pitch (the appropriate melody) for each frame, quantized onto a semitone scale.

- Onset events: Only some of the systems incorporate sets of distinct objects (individual notes or

short strings of notes, each with a distinct start and end time) internal to their processing. Most of them simply decide a single best melody pitch at every frame and do not attempt to form them into higher note-type structures. [Marolt 04] and [Dressler 05], however, take sets of harmonics similar to those in [Goto 04], but track the amplitude variation to form distinct fragments of more-or-less continuous pitch and energy that are then the basic elements used in later processing. [Paiva 04] goes further to carefully resolve his continuous pitch tracks into piecewise-constant frequency contours, thereby removing effects such as vibrato (pitch modulation) and slides between notes to get something closer to the underlying, discrete melody sequence. [Ryynanen 05] uses a Hidden Markov Model (HMM) providing distributions over features including an 'onset strength' related to the local temporal derivative of total energy associated with a pitch.

- Post-processing: Raw (multi) pitch tracks are further cleaned up to give the final melody estimates. In [Marolt 04, Paiva 04, Dressler 05], this involves choosing a subset of the note or note fragment elements to form a single melody line, including gaps where no melody note is selected. In each case, this is achieved by sets of rules that attempt to capture the continuity of good melodies in terms of energy and pitch (i.e. avoiding or deleting large, brief, frequency jumps). Rules may also include some musical insights, such as preference for a particular pitch range, and for the highest or lowest (outer) voices in a set of simultaneous pitches (a polyphony). [Goto 04] uses a set of interacting 'tracking agents' (alternate hypotheses of the current and past pitch) which compete to acquire the new pitch estimates from the current frame, and live or die based on a continuously-updated penalty that reflects the total strength of the past pitches they represent. In [Ryynanen 05, Vincent 05] HMMs are used to limit the dynamics of their pitch estimates (to provide a degree of smoothing that favors slowly-changing pitches). The first, simply connects his per-note HMMs through a third, noise/background, state, and also has the opportunity to include musicologically-informed transition probabilities that vary depending on an estimate of the current chord or key. The latter uses an HMM simply to smooth pitch sequences, training the transition probabilities as a function of interval size from chosen ground-truth melodies.

- Voicing: Finally, we have to consider how the systems distinguish between intervals where the melody is present and those where it is silent (gaps between melodies). [Goto 04, Vincent 05] simply report their best pitch estimate at every frame and do not admit gaps. [Polliner 05] basic pitch extraction engine is also continuous, but this is then gated by a separate melody detector; a simple global energy threshold over an appropriate frequency range was reported to work as well as a more complex scheme based on a trained classifier. The selection of notes or fragments in [Marolt 04, Paiva 04, Dressler 05] naturally leads to gaps where no suitable element is selected.

For further reviews, and also evaluations and experiments on melody extraction we refer to [Gómez 03, Klapuri 04, Polliner 07].

2.2 Techniques for sequence alignment

When we have a sequence of items, we can check their resemblance through an alignment process. This compares two sequences and tries to 'fit' one sequence into the other (the process of adjusting parts so that they are in proper relative position). With this, a similarity index between them can be obtained. Alignment algorithms are important not just because they allow us to arrange sequences or signals, but because they report a cost of this alignment. This cost can be seen as a distance or similarity value between the two sequences.

In next subsections we review some of the most popular algorithms that allow us to do this task: Dynamic Time Warping, string or edit distances, sequence alignment algorithms (form the Molecular Biology literature point of view), and Hidden Markov Models.

2.2.1 Dynamic Time Warping

Dynamic Time Warping (from now on DTW) is a technique aligning two sequences which may vary in time or speed and for measuring similarity between them. In general, DTW is a method that allows a computer to find an optimal match or alignment between two given sequences (i.e. time series) with certain restrictions. The two sequences are non-linearly 'warped' in the time dimension to determine a measure of their similarity independent of certain non-linear variations in their temporal evolution. It is a well known technique in the speech recognition comunity since the 1970's [Rabiner 93]. We now describe it in detail.

Suppose we have two time series: a sequence $Q = q_1, q_2, \dots, q_n$ of length n, and a sequence $C = c_1, c_2, \dots, c_m$ of length m. To align these two sequences using Dynamic Time Warping, we construct an $n \times m$ matrix D where the (i-th, j-th) element of it corresponds to the value of a local cost function (i.e. the squared distance, $d(q_i, c_j) = (q_i - c_j)^2$), which represents the difference between points q_i and c_j.

To find the best match between these two sequences, we can find a path through this matrix that minimizes the total cumulative distance between them. A warping path W is then defined to be a contiguous set of matrix elements that characterizes a mapping between Q and C. The k-th element of W is defined as $w_k = (i_k, j_k)$. So we have:

$$W = w_1, w_2, \dots, w_k, \dots, w_l \tag{2.11}$$

Where the length of the warping path (l) is comprised between $max\{n, m\}$ and $n + m - 1$. By definition, the optimal path W_0 is the path that minimizes the warping cost:

$$DTW(C, Q) = min\left\{ \sqrt{\sum_{k=1}^{l} w_k} \right\} \tag{2.12}$$

This path can be found using Dynamic Programming (DP) to evaluate the following recurrence, which defines the cumulative distance $D(i, j)$ as the distance $d(i, j)$ found in the current cell and the minimum of the cumulative distances of the adjacent elements:

$$D(i, j) = d(q_i, c_j) + min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} \tag{2.13}$$

The time and space complexity of DTW is $O(nm)$. In $D(n, m)$ (bottom right element of the $n \times m$ matrix) we find the accumulated distance cost through the path. An example of the resultant matrix can be seen in figure 2.4.

Note that the Euclidean distance between two sequences can be seen as a special case of DTW where the k-th element of W is constrained such that $w_k = (i_k, j_k)$, $i = j = k$. Observe that it is only defined in the special case where the two sequences have the same length.

To obtain the alignment path, we just have to backtrack starting from element (n, m) and choosing the smallest matrix position near it ($D(n-1, m)$, $D(n-1, m-1)$ or $D(n, m-1)$). We repeat this procedure until we get element $D(1, 1)$. We can keep track of the values and positions of the visited elements to obtain the final path W. The length of this path usually acts as a normalization factor for

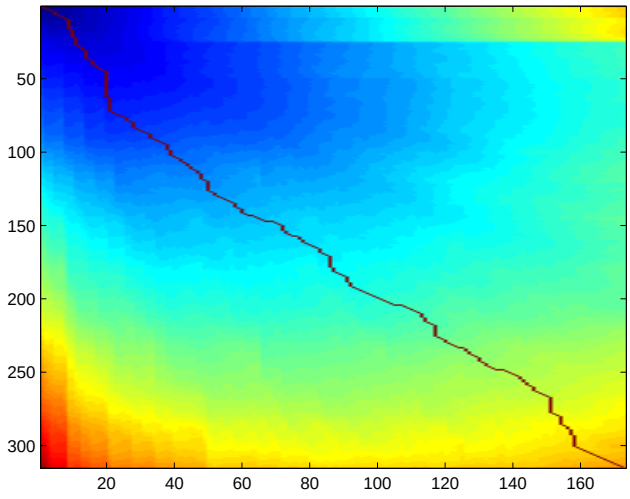

FIGURE 2.4 *DTW matrix example.*

the total accumulated distance $D(n, m)$. That is:

$$d_{norm} = \frac{D(n, m)}{l} \tag{2.14}$$

Where l corresponds to the total number of elements in W ($l = |W|$).

In practice, we do not evaluate all possible warping paths, since many of them correspond to pathological warpings. Instead, we consider the following constraints that decrease the number of paths considered during the matching process. This reduction of the number of paths considered also has the desirable side effect of speeding up the calculations, although only by a (small) constant factor.

In order to find the best path in the (n, m) plane (D), based on the parametric formulation above, several factors of the DTW algorithm must be specified [Myers 80b]:

1. Endpoint constraints on the path (boundary conditions). Usually the path starts at $w_1 = (1, 1)$ and ends at $w_l = (n, m)$, that is, the warping path starts at the top left and ends at the bottom right of the matrix. $(1, 1)$ and (n, m) represent the start and ending points of the two sequences.

2. Continuity conditions: Every point in the query (Q) and candidate (C) sequences must be used in the warping path, and both i and j indexes can only increase by 0 or 1 on each step along the path. In other words, if we take a point (i, j) from the matrix, the previous point must have been $(i - 1, j - 1)$, $(i - 1, j)$, or $(i, j - 1)$.

3. Monotonic condition: Given $w_k = (a, b)$ then $w_{k-1} = (a', b')$ where $a - a' \geq 0$ and $b - b' \geq 0$. The warping path cannot go backwards in time; both i and j indexes either stay the same or increase. They can never decrease.

4. Local continuity constraints (slope constraint condition). To further specify the optimal path, some local constraints must to be applied in order to guarantee that excessive compression or expansion of the time scales is avoided (i.e., the possible types of motion: directions, slopes of the path, ...) [Myers 80a].

5. Global path constraints (adjustment window condition). An intuitive alignment path is unlikely to drift very far from the diagonal. So, limitations on where the path can fall in the $n \times m$ plane are introduced. Some of the most known are the Sakoe-Chiba bound [Sakoe 78] or the Itakura parallelogram [Itakura 75].

6. Distance measures. Some formulations of DTW introduce various biases in addition to the slope constraints, by multiplying $d(i, j)$ by a weight which is dependent on the direction of the movement. In fact, the above formulation of $D(i, j)$ is biased towards diagonal steps: the greater the number of steps, the shorter the total path [Sankoff 83].

7. Axis orientation. In cases when both the local and global constraints are symmetric, and the distance metric is symmetric, there are no differences between the variable assignments of previous equations. However, when there is asymmetry in either local constraints, or in the distance metric, then the differences in variable assignments can be significant.

By applying these conditions, we can restrict the moves that can be made from any point in the path and therefore reduce the number of paths that need to be considered. An example of globally constrained DTW matrices is shown in figure 2.5.

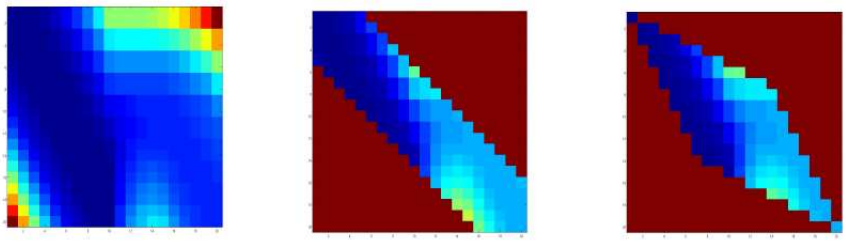

FIGURE 2.5 *Examples of an unconstrained DTW matrix (i), and Sakoe-Chiba (ii) and Itakura (iii) global constraints. Color gradient from dark blue to red indicates distance values starting at zero (dark blue). Dark burgundy corresponds to cells of the matrix whose value have not been calculated.*

An extension to DTW that also significantly speeds up the DTW calculation is a lower bounding technique based on the warping window (envelope) [Keogh 02]. Extensions of DTW and LCSS distance measures (section 2.2.2) to multiple dimensions have been proposed in [Vlachos 06]. Also, in [Ratanamahatana 04b], the authors introduce a new framework that learns arbitrary constraints on the warping path of the DTW calculation. Apart from improving the accuracy of classification, their technique speeds up DTW by a wide margin as well.

For further knowledge on DTW and related techniques we refer to [Myers 80a, Sankoff 83, Rabiner 93]. A review of some of the most popular statements about DTW is done in [Ratanamahatana 04a].

2.2.2 Edit-distances

The edit distance between two strings of characters is the number of operations required to transform one of them into the other. There are several different algorithms to define or calculate this metric and practically every algorithm implies a proper definition of the metric. We highlight some of them.

Levenshtein distance

Edit distance commonly refers to the Levenshtein distance [Levenshtein 66]. This is the basic edit distance function whereby the distance is given simply as the minimum edit distance which transforms a string into another. The Levenshtein distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. This operations, from now on called edit operations, must have some associated cost. For example:

- Delete a character $\Longrightarrow$ Cost 1
- Insert a character $\Longrightarrow$ Cost 1
- Substitute one character for another $\Longrightarrow$ Cost 1

The above edit cost scenario is called the *unit cost model*: the cost of edit operations does not depend on any characteristic of the elements under consideration, apart from their equality or inequality, in the case of replacement. Then, the Levenshtein distance between two strings $C = c_1, c_2, \ldots, c_n$ and $Q = q_1, q_2, \ldots, q_m$ is calculated through the recurrent relation:

$$D(i,j) = min \begin{cases} D(i-1,j-1) + d(c_i, q_j) & \text{(substitution or copy)} \\ D(i-1,j) + 1 & \text{(insertion)} \\ D(i,j-1) + 1 & \text{(deletion)} \end{cases} \tag{2.15}$$

Where $d(c_i, q_j)$ is a function whereby $d(c_i, q_j) = 0$ if $c_i = q_j$, 1 otherwise. Using this equation, a $n \times m$ matrix D can be filled. The edit-distance value for the entire candidate (C) and query (Q) sequences appears in the lower right corner of the matrix, and the optimal alignment is found by tracing back from cell (n, m) to $(1, 1)$ and recording which was the last applied operation that led to the distance value in each visited cell (i, j). This yields (in reverse order) the optimal sequence of edit operations, called *optimal alignment* between C and Q.

We now briefly point out some of the key advantages of the Levenshtein distance (and edit-distances in general):

1. The edit-distance is informative, in the sense that computing the edit-distance yields not only a distance value, but also the optimal alignment between two sequences, that displays which parts are resembling.
2. It is tolerant. It does not add any restriction on the length of the sequences.
3. The set of edit operations is not limited to the canonical set of insertion, deletion and replacement. Arbitrary edit operations may be defined. In this way, the edit-distance can be customized to fit the application domain.
4. Finally, the use of edit-distance for comparing sequences is not limited to sequences of symbols, nor to sequences of elements of the same type (see [Grachten 06] for an example application to melody).

There are many extensions to the Levenshtein distance function [Gusfield 97, Lemstrom 00b]. These typically alter the $d(i, j)$ function, but further extensions can be made, such as the Needleman-Wunsch distance.

Needleman-Wunsch distance

This approach is known by various names, Needleman-Wunsch, Needleman-Wunsch-Sellers, Sellers and the Improving Sellers algorithm, etc. [Needleman 70, Sellers 74]. It is similar to the basic edit-distance metric, Levenshtein distance, but adds a variable cost adjustment to the cost of a gap (i.e. insert/deletion), in the distance metric. So, the Levenshtein distance can simply be seen as the Needleman-Wunsch distance with $\delta = 1$.

$$D(i,j) = min \begin{cases} D(i-1,j-1) + d(c_i, q_j) & \text{(substitution or copy)} \\ D(i-1,j) + \delta & \text{(insertion)} \\ D(i,j-1) + \delta & \text{(deletion)} \end{cases} \tag{2.16}$$

Where δ equals to the 'gap cost', and $d(c_i, q_j)$ is again an arbitrary distance function on the characters c_i and q_j.

Smith-Waterman algorithm

Again similarly to the Levenshtein distance, the Smith-Waterman distance was developed to identify optimal alignments between related DNA and protein sequences.

When dealing with sequence similarity, an important distinction to make on the algorithmic level is whether a program calculates a global or a local alignment. The prototypical global algorithm is the classic Needleman-Wunsch algorithm (on previous section). The Smith-Waterman algorithm [Smith 81], on the other hand, is the best known local alignment algorithm. In cases where two sequences are similar over their entire lengths, the local algorithms should find this fact as well as the global algorithm, whereas when two sequences share only a limited region of similarity, only the local algorithm will discover this.

The Smith-Waterman algorithm is motivated by systems where scores for matches and mismatches have different signs, for instace, where matches increase the overall score of an alignment whereas mismatches decrease it. A good alignment then has a positive score and a poor alignment has a negative score. The local algorithm finds an alignment with the highest score by considering only alignments that score positive and picking the best one from those. The algorithm is a Dynamic Programming algorithm. For the comparison of two sequences, it basically requires setting a gap penalty ($\delta \geq 0$) in addition to the score for a match or identity, and the penalty for a mismatch ($\mu_- \geq 0$).

Let be two sequences $C = c_1, c_2, \ldots, c_n$ and $Q = q_1, q_2, \ldots, q_m$. The algorithm constructs an $(n+1) \times (m+1)$ similarity matrix H trough the recurrent formula[2]:

$$H(i,j) = max \begin{cases} H(i-1,j-1) + s(c_i, q_j) \\ H(i-1,j) - \delta \\ H(i,j-1) - \delta \\ 0 \end{cases} \tag{2.17}$$

Where $s(c_i, q_j)$ acts as a local cost function. Matrix H needs an initial assignment of:

$$H(i,0) = H(0,j) = 0 \tag{2.18}$$

[2]For convenience and coherence with next sections, the algorithm is formulated here as a similarity measure, but it can be easily formulated in a distance flavour [Gotoh 82].

For $0 \leq i \leq n$ and $0 \leq j \leq m$. Then, the desired local alignment score is defined to be the maximum value of $H(i,j)$ over the entire matrix:

$$max\{H(i,j) : 1 \leq i \leq n, 1 \leq j \leq m\} \tag{2.19}$$

Usually, one uses a similarity matrix that attributes a score to each possible pair. The score should be positive for desirable pairs and negative for dissimilar ones in order to ensure meaningful local alignments ($s(c_i, q_j) = \mu_+ > 0$ if $c_i = q_j$, and $s(c_i, q_j) = \mu_- \leq 0$ if $c_i \neq q_j$). Gaps are usually penalized using a linear gap function that assigns an initial penalty for a gap opening and extension gap penalty for each deleted or inserted sequence item increasing the gap length. In other words:

- The first term, $H(i-1, j-1) + s(c_i, q_j)$, corresponds to extending the alignment by one item of each sequence.
- The second term, $H(i-1, j) - \delta$, describes extending the alignment by including item j from sequence B, and inserting a gap of k items in length (aligned to end with element j of sequence B) into sequence A.
- The third term, $H(i, j-1) - \delta$, is the equivalent term for inserting a gap into sequence B.
- The fourth term, zero, is what distinguishes the Smith-Waterman from the Needleman-Wunsch algorithm. For a global alignment, the zero is eliminated [Needleman 70, Sellers 74]. That is, the partial scores within the table are allowed to become negative. Putting the zero in the recursion is saying that if the partial alignment score becomes negative during the calculations, we want to ignore that as well as ignore the preceding calculations and start over from a neutral score.

The score of a given sequence alignment can also be described by counting the number of matches, the number of mismatches, and the number of insertions or deletions. Then, each of these is multiplied by its corresponding parameter values:

$$Score = \mu_+ \cdot |matches| - \mu_- \cdot |mismatches| - \delta \cdot |insertions, deletions| \tag{2.20}$$

An optimal alignment is one that maximizes this expression. This equation also shows that the score can be represented as a linear function of two parameters: mismatch penalty ($\triangle\mu = \mu_+ - \mu_-$) and gap penalty ($\delta$). A review on sequence alignment and penalty choices can be found in [Vingron 94], and some proofs on the metricity of these similarity algorithms are developed in [Waterman 76].

Once a matrix H has been constructed, the pair of segments with maximum similarity is found by first locating the maximum element of it. The other elements leading to this maximum value are then sequentially determined with a traceback procedure ending with an element of H equal to zero. This procedure identifies the sequence as well as produces the corresponding alignment. The pair of segments with the next best similarity is found by applying the traceback procedure to the second largest element of H not associated with the first traceback [Waterman 87a].

An extension of the Smith-Waterman algorithm to speed-up computations was made in [Gotoh 82], which also allows affine gaps to be considered. Finally, we note that heuristics approximating the Smith-Waterman algorithm are commonly used in today's DNA and protein sequence comparison engines such as BLAST[3] [Altschul 90] or FASTA[4] [Lipman 85]. Both BLAST and FASTA place additional restrictions on the alignments that they report in order to speed up their operation. Because of this, Smith-Waterman is more sensitive than either any of them [Pearson 91].

[3] http://en.wikipedia.org/wiki/BLAST
[4] http://en.wikipedia.org/wiki/Fasta

Longest common subsequence

The longest common subsequence similarity measure (LCSS), is a variation of edit distance used in speech recognition and text pattern matching. The basic idea is to match two sequences by allowing some elements to be unmatched. The advantage of the LCSS method is that some elements may be unmatched or left out (e.g. outliers), whereas in Euclidean and DTW, all elements from both sequences must be used, even the outliers.

Let C and Q be two sequences of length n and m, respectively. As was done with DTW, we give a recursive definition of the length of the longest common subsequence of C and Q:

$$LCSS(C_{1,\ldots,i}, Q_{1,\ldots,j}) =$$

$$= \begin{cases} \phi & \text{if } i = 0 \text{ or } j = 0, \\ LCSS(C_{1,\ldots,i-1}, Q_{1,\ldots,j-1}) & \text{if } c_i = q_j \\ max\{LCSS(C_{1,\ldots,i-1}, Q_{1,\ldots,j}), LCSS(C_{1,\ldots,i}, Q_{1,\ldots,j-1})\} & \text{otherwise.} \end{cases} \tag{2.21}$$

Here '+' denotes concatenation, and $max\{\}$ gives the longest sequence. Since this problem has an optimal substructure property, it can be solved by dynamic programming [Hirschberg 75].

The dissimilarity between the two sequences is therefore defined as:

$$D(C, Q) = \frac{n + m - 2 \cdot |LCSS(C, Q)|}{n + m} \tag{2.22}$$

Where $|LCSS(C, Q)|$ is the length of the longest common subsequence. Intuitively, this quantity determines the minimum (normalized) number of elements that should be removed from and inserted into C to transform C to Q. As with DTW, the LCSS measure can be computed by dynamic programming in $O(nm)$ time. An algorithm combining LCSS and DTW for symbolic music retrieval (applied to query-by-humming systems) was proposed in [Guo 04]. Other works in this field with fast LCSS algorithms include [Ukkonen 03, Navarro 05].

Other edit-distances

The Hamming distance [Hamming 50] is defined as the number of bits which differ between two binary strings (i.e. the number of bits which need to be changed to turn one string into the other). For example, the bit strings "10011010" and "10001101" has a hamming distance of 4 (as four bits are dissimilar). Other variants of the edit distance include the Jaro metric [Jaro 89], the Jaro-Wrinckler distance [Wrinckler 99], the Wagner-Fischer edit-distance [Wagner 74], the Jaccard similarity (first mentioned in an early paper of 1912 [Jaccard 12]), and many more (including algorithms for transposition invariant string matching [Navarro 05]).

For a general discussion of string edit distances, see [Sankoff 83, Gusfield 97] and for particular applications to the music domain we refer to [McNab 96, Lemstrom 00a, Lemstrom 00b, Ukkonen 03, Grachten 06]. We also find many implemented algorithms on the web, for example in *Sam's string metrics* web page[5].

2.2.3 Hidden Markov Models

Hidden Markov Models (HMM) have shown to be a powerful statistical tool in speech processing. The theoretical aspects about HMMs are quite old but it is not until the sixties (when new parameter estima-

[5] http://www.dcs.shef.ac.uk/šam/stringmetrics.html

tion techniques appear [Baum 67]) that they become really important in mathematical developments. The exploitation of HMM for speech processing does not start until the ninety-seventies of past century. Nowadays, HMMs are included in many different research areas, and they have shown his robustness and elegance in exceed [Rabiner 89].

Main rationale

Nature is full of processes that are still unknown for humans. The unique knowledge one has about these processes is via observation. Observation is the way for discovering the real behavior of the system. At this point, one can choose between a deterministic or statistical description of the system. Deterministic descriptions are based on a perfect knowledge of the behavior of the system (i.e., a sine generator). On the other hand, statistical descriptions are used when this behavior is not fully controlled and only statistical estimations can be done. This would be the case for Gaussian models, Markov Models and Hidden Markov Models.

Hidden Markov Markov models are an extension of the Markov Models. The Markov Models are useful when each observation corresponds to a physical event, but this case is too restrictive to many problems of interest. HMMs are used when the observations are also probabilistic functions. Sometimes, these are referred as a double embedded stochastic process with an underlying stochastic process that is not observable (hidden) but can only be observed through another set of stochastic processes that produce the sequence of observations [Rabiner 89].

Elements of an HMM

Each nature phenomena can be represented as a sequence of vectors or observations O, defined as [Young 00]:

$$O = o_1, o_1 \dots o_T \tag{2.23}$$

Where o_t is the vector observed at time t. Then, the phenomena recognition problem can be resumed, from a mathematical point of view, as:

$$argmax_i \{P(w_i|O)\} \tag{2.24}$$

Where w_i is the i-th model previously defined in our 'vocabulary'. Now, by using the Bayes' rule, we get:

$$P(w_i|O) = \frac{P(O|w_i)\,P(w_i)}{P(O)} \tag{2.25}$$

Given a set of prior probabilities $P(w_i)$, the most probable model depends only on the likelihood $P(O|w_i)$.

Now, the unique problem is the estimation of the observations. Due to the high dimensionality of observation vectors, this process can't be done in a deterministic way. We will use a Markov model for that purpose.

Markov model A Markov model is a finite-state machine which changes state once every time unit. Each time t, the machine enters to a new state j and an observation vector o_t is generated by its probability density function $b_j(o_t)$. The transition between the previous state i and the actual state j is defined by the discrete probability value a_{ij}. Figure 2.6 shows a 6 state Markov model example.

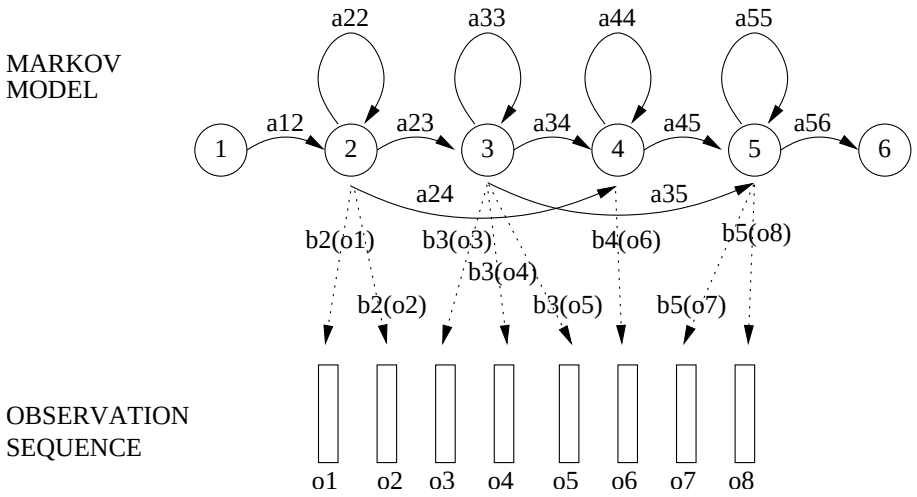

FIGURE 2.6 *Markov generation model.*

Here, the state sequence $X = X_1, X_2, \ldots, X_8$ moves through the Markov model in order to generate the observation vector $O = o_1, o_2, \ldots, o_8$. The joint probability that O is generated by this specific model is:

$$P(O, X|M) = a_{12} b_2(o_1) a_{22} b_2(o_2) a_{23} b_3(o_3) \ldots \tag{2.26}$$

Hidden Markov Models Note that, in fact, the underlying X sequence is unknown. That is what it is called Hidden Markov Models.

The full HMM system is a set of Markov models (like shown in figure 2.6), and the purpose of equation 2.24 is to try to find the model which best generates (that is, with the major probability) the observed sequence:

$$P(O|M) = \sum_X a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t)\, a_{x(t)x(t+1)} \tag{2.27}$$

Where $x(0)$ is the model entry state and $x(t+1)$ is the model exit state. Here, M is the set of Markov models ($M = M_1, M_2, \ldots$), in the same sense that in equation 2.24, w_i was presented as an specific model in our vocabulary.

It is obvious that equation 2.27 can not be computed directly. Some recursive techniques are needed. But whatever the recursive technique is, it supposes that $\{a_{ij}\}$ and $\{b_j(o_t)\}$ are known. We need some kind of training process for that.

Training process

Given a set of examples for each model, all these parameters ($\{a_{ij}\}$ and $\{b_j(o_t)\}$) can be automatically estimated by using some re-estimation techniques. Then, the global system can be summarized as follows:

- Each model is trained with a sufficient number of examples of that specific model.
- When a new unknown input sequence has to be recognized, the likelihood of each model generating that input sequence is calculated.

- The recognized sequence belongs to the model that best generates the input sequence.

Gaussian mixtures Before entering in detail with the re-estimation process, note that, sometimes, the output probabilities $b_j(o_t)$ are represented as an addition of S independent data streams. Gaussian mixture densities are commonly used for that purpose.

From a mathematical point of view, let $b_j(o_t)$ be:

$$b_j(o_t) = \prod_{s=1}^{S} \left[\sum_{m=1}^{M_s} c_{jsm} N(o_{st}; \mu_{jsm}, \Sigma_{jsm}) \right]^{\gamma_s} \tag{2.28}$$

Where M_s is the number of mixture components in the stream s, c_{jsm} is the weight of the m-th component, γ_s is the stream weight (it is usually a manual setting) and $N(\cdot; \mu, \Sigma)$ is a multivariate Gaussian with mean μ and covariance matrix Σ:

$$N(\cdot; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(o-\mu)' \Sigma^{-1} (o-\mu)} \tag{2.29}$$

Where n is the dimensionality of o.

Baum-Welch re-estimation In the training process, a good estimation for each model parameters is required. The Baum-Welch re-estimation algorithm improves these estimations providing more accurate results in our HMM system.

As seen previously, the output probabilities are usually represented as an addition of S mixture Gaussian components. For the study of the Baum-Welch re-estimation algorithm, only the case for one single Gaussian stream is considered. Note how multiple Gaussian mixtures can be interpreted as sub-states in each state in which the transition probabilities are exactly the mixture weights C_{jsm}. See figure 2.7 for details.

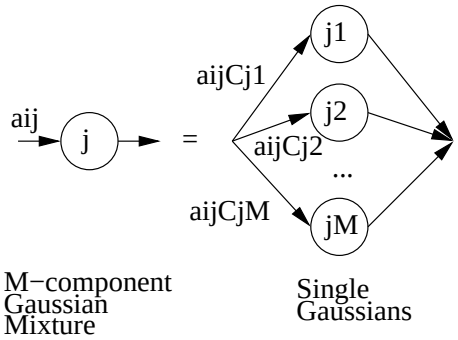

FIGURE 2.7 *Decomposition of Gaussian mixtures.*

Then, the problem can be reduced to a mean and variance estimation of the output probabilities of a single-Gaussian HMM system:

$$b_j(o_t) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} e^{-\frac{1}{2}(o_t-\mu_j)' \Sigma_j^{-1} (o_t-\mu_j)} \tag{2.30}$$

In the case of a single state HMM, the computation of these parameters is quite easy:

$$\hat{\mu}_j = \frac{1}{T} \sum_{t=1}^{T} o_t \tag{2.31}$$

And:

$$\hat{\Sigma}_j = \frac{1}{T}\sum_{t=1}^{T}(o_t - \mu_j)(o_t - \mu_j)' \tag{2.32}$$

But in real case, the HMM is not usually single state and, furthermore, it is not possible to assign each observation sequence to a specific state. Then, the expressions 2.31 and 2.32 can be interpreted only as a non-random initialization, but they have to be redefined.

In order to solve this problem, let each sequence be assigned to each of the states in proportion to the probability of the model being in a state when the vector was observed. Then:

$$\hat{\mu}_j = \frac{\sum_{t=1}^{T} L_j(t) o_t}{\sum_{t=1}^{T} L_j(t)} \tag{2.33}$$

And:

$$\hat{\Sigma}_j = \frac{\sum_{t=1}^{T} (o_t - \mu_j)(o_t - \mu_j)'}{\sum_{t=1}^{T} L_j(t)} \tag{2.34}$$

Where $L_j(t)$ is the probability of being in state j at time t and the denominators in both expressions represent the normalization factors.

At this point, the also called probability of state occupation $L_j(t)$ must be computed, and the so called *forward-backward algorithm* will be used for this task.

Forward-backward algorithm Let $\alpha_j(t)$ be the *forward probability*:

$$\alpha_j(t) = P(o1, \dots, o_t, x(t) = j|M) \tag{2.35}$$

Where M is one of the models and N is the number of states for that model.

The meaning of this forward probability is the joint probability of observing the first t vectors and being at state j at time t. From a mathematical point of view, it can be calculated recursively as:

$$\alpha_j(t) = \left[\sum_{i=2}^{N-1} \alpha_i(t-1)\alpha_{ij}\right] b_j(o_t) \tag{2.36}$$

Note that states 1 and N are non-emitting states, that is, no output probability can be calculated from them. They are useful for state-transition problems.

The initial conditions for recursion are:

$$\alpha_1(1) = 1 \tag{2.37}$$

And:

$$\alpha_j(1) = a_{1j} b_j(o_t) \qquad 1 > j > N \tag{2.38}$$

And the final condition is:

$$\alpha_N(T) = \sum_{n=2}^{N-1} \alpha_i(T)\alpha_{iN} \tag{2.39}$$

Then:

$$P(O|M) = \alpha_N(T) \tag{2.40}$$

In a similar way, let $\beta_j(t)$ be the *backward probability*:

$$\beta_j(t) = P(o_{t+1}, \ldots, o_t | x(t) = j, M) \tag{2.41}$$

This probability can be calculated as:

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(o_{t+1}) \beta_j(t+1) \tag{2.42}$$

The initial condition is:

$$\beta_i(T) = a_{iN} \qquad 1 < i < N \tag{2.43}$$

And the final condition is:

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{ij} b_j(o_1) \beta_j(1) \tag{2.44}$$

Then, we get the state occupation probability by multiplying the forward probability (joint probability) and the backward probability (conditional probability):

$$\alpha_j(t)\beta_j(t) = P(O, x(t) = j|M) \tag{2.45}$$

Finally,

$$\begin{array}{rcl} L_j(t) & = & P(x(t) = j|O, M) \\ & = & \frac{P(O, x(t)=j|M)}{P(O|M)} \\ & = & \frac{1}{P}\alpha_j(t)\beta_j(t) \end{array} \tag{2.46}$$

where $P = P(O|M)$. Note that these operations involves probability multiplications, and resolution can affects the final results. Hence, these values are usually computed in log arithmetic.

Viterbi Decoding

In the previous section, an efficient method for computing the forward probability was shown. When new data comes to the system and the observation vectors are extracted, the same algorithm could be applied successfully for recognition, that is, find the model which yields the maximum value of likelihood $P(O|M_i)$. But this method has a disadvantage: if one transition's probability is zero ($a_{i,j} = 0$), it is possible that the most probable state in a given time unit t belongs to an unexisting path.

This problem can be solved by computing the algorithm but calculating the maximum likelihood state sequence instead of the maximum probability state sequence. It is computed by using the same algorithm, but the summation is replaced by a maximum operation.

Let $\phi_j(t)$ be the maximum likelihood of observing vectors o_1 to o_t, and being at state j at time t. It can be computed, in a similar way than equation 2.36 does, as:

$$\phi_j(t) = \max_i \{\phi_i(t-1)a_{ij}\}\, b_j(o_t) \tag{2.47}$$

The initial conditions are:

$$\phi_1(1) = 1 \tag{2.48}$$

And:

$$\phi_j(1) = a_{1j} b_j(o_1) \qquad i < j < N \tag{2.49}$$

The final condition is:

$$\phi_N(T) = \max_i \{\phi_i(T) a_{iN}\} \tag{2.50}$$

Then, the maximum likelihood is:

$$\hat{P}(O|M) = \phi_N(T) \tag{2.51}$$

As for the Baum-Welch algorithm, these calculations lead to underflow. Log arithmetics are also used in this case:

$$\psi_j(t) = \max_i \{\psi_i(t-1) + \log(a_{ij})\} + \log(b_j(o_t)) \tag{2.52}$$

In figure 2.8 a graphical interpretation of the Viterbi algorithm is shown.

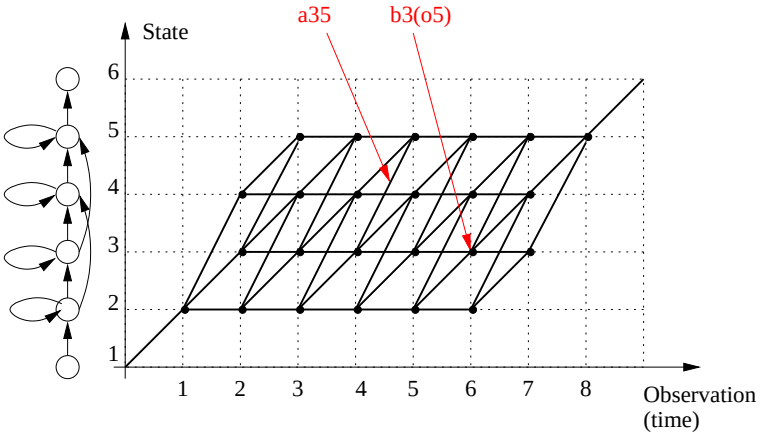

FIGURE 2.8 *Viterbi algorithm.*

2.3 Application contexts

The first part of this section is devoted to give a brief introduction and to cite some basic references of previous work related to audio content-based similarity. Concretely, two areas within the MIR community related to cover song identification (as we have seen in section 1.5) are overviewed: audio fingerprinting and genre classification.

2.3.1 Audio fingerprinting

Audio fingerprinting uses the inherent qualities of the music to uniquely identify it by comparing it against a database of known music. These systems extract a perceptual digest of a piece of audio

content (i.e., the fingerprint) and store it in a database. When presented with unlabeled audio (figure 2.9), its fingerprint is calculated and matched against those stored in the database. Using fingerprints and matching algorithms, distorted versions of a recording can be identified as the same audio content.

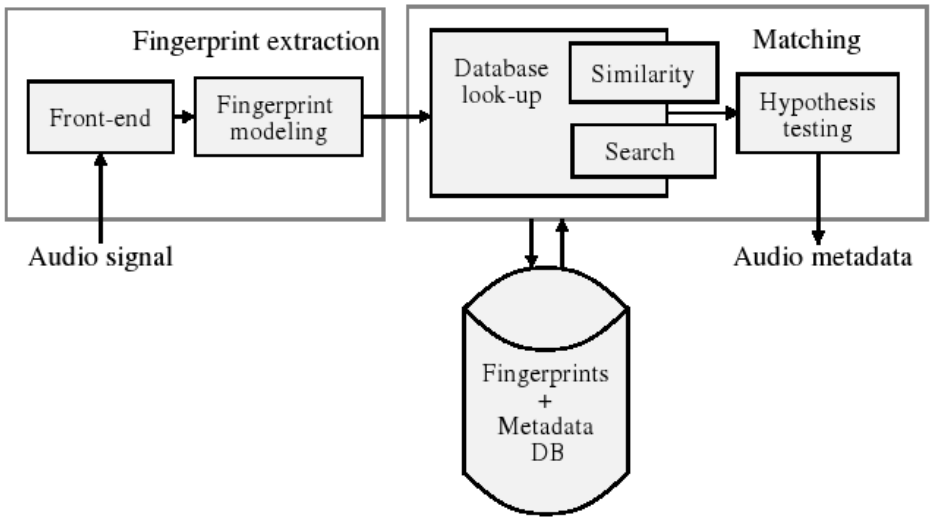

FIGURE 2.9 *Content-based audio fingerprint framework. Figure extracted from [Cano 07] with permission of the author.*

The design principles behind audio fingerprinting are recurrent in several research areas. Compact signatures that represent complex multimedia objects are employed in Information Retrieval for fast indexing and retrieval.

A typical fingerprint extraction process consists of a front-end and a fingerprint modelling block (a schema is shown in figure 2.10, where we can also see a schema and an enumeration of the techniques and features employed).

The front-end computes a set of measurements from the signal. It consists on pre-processing and framing, transforming the signal into the frequency domain or the wavelet domain, and extracting some features from this. The fingerprint model block receives a sequence of feature vectors calculated on a frame by frame basis and defines the final fingerprint representation (i.e.: a vector, a trace of vectors, a codebook, a sequence of indexes to HMM sound classes, a sequence of error correcting words or musically meaningful high-level attributes, etc) [Cano 07].

As we can see in figure 2.9, the database look-up consists of a matching procedure and an efficient search strategy. Regarding similarity (matching procedure), distance metrics are very much related to the type of model chosen. When comparing vector sequences, a correlation is common, but also Euclidean distances, cross entropy estimation and the Manhattan distance are used. In some systems, where fingerprints are compactly modeled as a codebook or a sequence of indexes to HMMs, distances are computed directly between the feature sequence extracted from the unknown audio and the reference audio fingerprints stored in the repository [Cano 02a]. For instance, in noisy broadcast audio, the identification system can be build on a stochastic matching technique such as HMMs [Batlle 02].

The search method depends on the fingerprint represention. One can use spatial access methods, a simpler distance to quickly eliminate many hypothesis, etc. Furthermore, heuristics similar to those used in computational biology for the comparison of DNA (see section 2.2.2) are used to speed up a search in a system where the fingerprints are sequences of symbols.

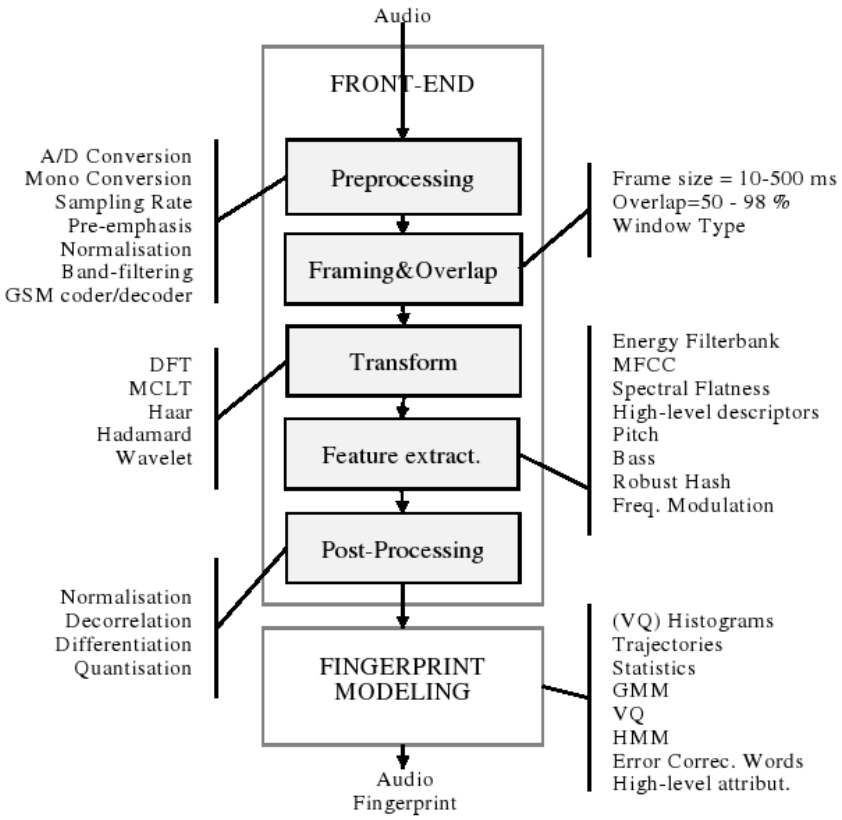

FIGURE 2.10 *Fingerprint extraction framework: front-end (top) and fingerprint modelling (bottom). Figure from [Cano 07] with permission of the author.*

Plenty of audio fingerprinting algorithms have been proposed. They differ mainly in the types of features being considered, the modelling of the fingerprint, the type of distance among fingerprints, and the indexing mechanisms for the database look-up [Batlle 03].

For a more detailed explanation on audio fingerprinting we further refer to the works cited previously in this subsection and, in addition, to [Herre 01, Cano 02b, Venkatachalam 04].

2.3.2 Genre/artist classification

Concerning the purely content-based musical similarity, the most significant amount of research has been devoted to audio-based genre or artist classification, with one of the first approaches presented in [Tzanetakis 01].

Since then, a big bunch of proposals to achieve such a task have been proposed, but, basically, they proceed in two general steps [Aucouturier 03]:

- Frame-based feature extraction: the music signal is cut into frames, and a feature vector of low-level descriptors related to timbre, rhythm, etc. is computed for each frame.
- Machine Learning/Classification: a classification algorithm is then applied on the set of feature vectors. The class models used in this phase are usually trained beforehand, in a supervised way.

Timbre descriptors are the most used ones. Numerous research efforts have been devoted to timbre similarity in music, and sometimes, each contribution is often yet another instantiation of the same basic pattern recognition architecture. The signal is cut into short overlapping frames, and, for each frame, Mel Frequency Cepstral Coefficients (MFCC) and the derivatives of them are computed. MFCCs have been applied successfully in speech processing, and their application to all audio types is straightforward [Logan 01]. The number of MFCCs is typically 12. Sometimes other spectral-based descriptors are used, such as spectral flatness, zero-crossing rate, etc. [Tzanetakis 02b].

Apart from the basic scheme related with timbre, other facets of music similarity are also incorporated. Although timbre is a very important aspect of music similarity, it is clear that it cannot define by itself the required similarity (or genre) shared between two pieces of music. Some papers that deal with rhythm or tempo similarity are [Paulus 02, Foote 02, Dixon 04, Vignoli 05].

Regarding classification itself, in general, a statistical model of the extracted features is computed. K-means and Gaussian Mixture Models (GMMs) are typically used. The number of K-mean or GMM centres is a discussed parameter. Finally, models are compared with different techniques: sampling, Earth Movers Distance (EMD), Asymptotic Likelihood Approximation, Kullback-Leibler divergence, etc. For more details we refer to [Berenzweig 03, Pampalk 03, Aucouturier 04]. Also, improvements on classification are well summarized at [Pampalk 05].

The huge amount of research done with genre similarity only deals with features describing the whole song (or a fragment of it), or with clusters of MFCCs or other features (the so called bag-of-features methods). When using these features, temporal information is kept out or mixed.

Very few research in genre classification has been done taking into account the succession of sequences of features (time series of descriptors) over the whole song (one rare exception would be [Shao 04]).

Finally, we highlight some reviews on the state-of-the-art regarding genre/artist classification: [Aucouturier 03, Tzanetakis 02b, Pampalk 05].

2.4 Descriptors' sequence alignment

The importance of sequences in musical similarity has been discussed in [Casey 06a]. Their results show a significant improvement in performance for audio similarity measures using temporal sequences of features.

This section introduces in much detail some early works about music alignment and the processing of sequences of descriptors. There are few works dealing with instantaneous features. So, before citing more recent works about cover song identification, one might also consider some earlier references dealing with the processing of descriptors sequences that are related to task.

An audio retrieval-by-example system for orchestral music was introduced in [Foote 00]. It was based on the variation of soft and loud passages. The envelope of audio energy versus time (RMS signal power across 1-second windows) was computed in one or more frequency bands, and similarity between energy profiles was calculated using Dynamic Programming (DP) and *Discrete Time Warping*. In this work, the author focused on classical music (specially symphonic music and piano concertos), where there exists a great variation in the dynamics of the piece, and solved the problem of small tempo variations by means of the mentioned DP algorithm. He already noted that this approach may not be useful for much popular music, which generally has much less dynamic range.

A similar approach was followed in [Yang 01]. This work focused on only spectral components over a short time period following temporal power peaks, rather than over the entire time period (making

tempo changes more transparent). Also, they refined the minimum distance matching obtained with Dynamic Programming with linearity filtering, making it more robust. For this, they fitted a straight line through the points obtained from the Dynamic Programming approach using least mean-square criteria, and checked and removed far away points. This was done iteratively until all of them had a small neighborhood to the fitted line.

[Hu 03] described a method that aligned polyphonic audio recordings of music to symbolic score information in standard MIDI files (without polyphonic transcription). They used chroma representation extracted both from MIDI and audioand Dynamic Time Warping (DTW) to perform the alignment.

A toolkit for aligning audio recordings of different renditions of the same piece of music automatically was presented in [Dixon 05]. It was based on an efficient implementation of a DTW algorithm and used some constraints for it in order to perform the alignment with linear time and space costs. They represented the audio data by positive spectral difference vectors and tested their method with Classical and Romantic piano music.

DTW has been the main algorithm used in the speech and music processing communities to align pieces of audio, but also several Hidden Markov Models (HMM) [Rabiner 89], string matching algorithms [Gusfield 97] such as approximate string matching, string edit-distances, the longest common subsequence algorithm or variants on that [Guo 04], have been successfully employed in MIR. For instance, some disciplines that have traditionally been using techniques dealing with sequences are Query by Singing/Humming [Ghias 95, McNab 96], Symbolic Melodic Similarity [Lemstrom 00a, Ukkonen 03, Typke 04] or Audio Fingerprinting [Cano 02b, Venkatachalam 04].

2.5 Audio cover song identification

A crucial aspect for audio cover song identification is the similarity between tonal sequences among pieces. This is considered in [Izmirli 05]. Here, the author uses Pitch Distribution Profiles obtained from labeled audio and from literature to create a template to approximate the distribution of pitches in major and minor key profiles. First, a chroma-based representation is used to capture tonality information, and templates are formed from labeled instrument sounds weighted according to Temperley and Diatonic profiles. Once templates are formed, these become part of the model to which incoming information is compared. More concretely, these profiles are used to weight chroma representation vectors extracted directly from audio frames of 2.5 seconds long (35% overlapping), and to conform a sequence of indices representing the mode and tonic values. So, precalculated templates serve as attractor points in tonal space to obtain a sequence of tonal center estimates that correspond to the trajectory of tonal evolution of the piece. Finally, Dynamic Time Warping (DTW) is employed to obtain a similarity measure between pieces. To perform DTW, a distance measure that models distances in tonal space (between forementioned indices) needs to be defined. The author solves this issue by using Lerdahl's regional distance (a tonal space distance measure defined to calculate distances between chords). Evaluation was carried out with a database of 125 recordings of Chopin Mazurkas, where only 12 of these recordings were unique, and the remaining were all played by multiple pianists.

An efficient method for audio matching is presented in [Müller 05]. In this article, and later in [Müller 06], a new type of chroma-based audio feature that is claimed to be robust against dynamics, timbre, articulation, and local tempo variations is introduced. Furthermore, they describe a matching procedure that allows to handle global and local tempo variations. To obtain said chroma-based audio features, they proceed in two phases: in a first stage, a small analysis window is used to investigate how the signal's energy locally distributes among 12 chroma classes. This 12-dimensional chroma

vectors are then normalized. In a second stage, a much larger statistics window is used, and chroma vectors are thresholded in order to make the features insensitive to noise components, and quantized (in a logarithmic fashion). Then, these vectors are convolved with a Hann window (component-wise), downsampled and re-normalized to create what they call CENS features (Chroma Energy distribution Normalized Statistics). To perform audio matching, they define a metric based on the inner product of two CENS vectors, and to account for global tempo variations, they create several versions of the query audio clip corresponding to different tempos (using different sizes of the statistics window and downsampling factors). Experiments are done with 1167 files reflecting solely some classical music pieces including different interpretations.

Gómez and Herrera have also focused on tonal similarity and its application to the identification of different versions of the same piece. Previously, they analyzed how tonal descriptors were useful to locate versions of the same song [Gómez 06a, Gómez 06b]. In these works, chroma representations (HPCP) and overall tonality are used to generate a transposed version of the features (THPCP) independent of the main key of the song. Then, two analysis are performed. First, they analyze how global tonal descriptors are similar for different pieces by considering only the first phrase of the song (manually detected) or the complete song. High correlation factors between THPCP can be seen. Second, they use similarity matrices between versions to assess if two songs come from the same *canonical song* or not. Finally, and looking at the structure of the piece, relevant results are achieved with DTW (used to determine a distance between songs). Correlation among HPCP feature vectors is employed as a measure of similarity between them. The forementioned method is refined in [Gómez 06c], where a structural analysis is performed to obtain automatically the most representative excerpts of the musical piece (two summaries are computed). In this work they also use a DTW algorithm to estimate the alignment cost between the extracted summaries, and therefore, to obtain a similarity measure. Promising results are provided with a relatively small collection of 90 versions from 30 different popular music songs.

[Marolt 06] uses Spectral Modelling Synthesis (SMS) with an EM approach, a beat tracker and a self-similarity matrix, to create a summarized description of the song for calculating similarity. First, SMS is used to extract partials from audio signal, and predominant pitches are extracted with an EM approach (which estimates the most likely pitches to have generated the observed series of partials). Then, using SMS partial tracking, pitches are linked in time in a series of pitch tracks. These, in the paper, are called *melodic fragments* because they represent different parts of melodic lines (lead and accompaniment). Besides, a beat tracker is used to perform beat detection. Then, beat boundaries are used to resample the representation. Finally, with a self-similarity matrix (obtained using the cosine similarity measure), the structure of the piece is inferred, and melodic patterns (i.e. parts of the melody) that are repeated several times in the song are extracted. The result is a description (a matrix) that incorporates melodic, rhythmic and structural information. These patterns are used as a summarized description of a song, and, therefore, for assessing similarity. This is calculated shifting the shorter pattern of a pair of songs beat-by-beat over the length of the longer pattern and calculating similarity of each shifted position. A key profile of each song is first calculated to compensate for difference in keys. For that, these key profiles are correlated in all shifted positions and the best match is taken to represent the difference in key between the two songs.

Different methods for efficiently assessing musical similarity are described and evaluated in [Casey 06a]. Apart from demonstrating a significant improvement in performance using temporal sequences of descriptors, they also show that quantizing the features to string-based representations also perform well, thus admitting efficient implementations based on string matching. They use two audio features: Log-Frequency Cepstral Coefficients (LFCC, which is a simplification of MFCC), and 12-bin chromagram

representation (PCP). For assessing similarity, they try with Matched Filters (an exact match over a given temporal window) and DTW. They also use Vector Quantization (VQ) to convert the signal into a string of symbols (each feature vector is assigned to the nearest of K cluster centers using Euclidean distance). This allows them to use text tools such as Exact String Matching, the String-Edit Distance (a good approximation to DTW for discrete symbols, also named Levenshtein distance metric), VQ State Histograms (a histogram of the symbols), and hash-table lookups. In [Casey 06b], the same features are used to model and recognize *remixes*. In the modelling step, they concatenate feature vectors into high-dimensional vectors (audio shingles of 800 dimensions for LFCC and 480 for PCP). That is, using a window of 4 sec. with a hop size of 0.1 sec. A distance between remixes is calculated summing the N minima of the pair-wise shingle distances between songs. These are obtained by sorting the distances in ascending order and summing the first N values. To classify, the define a scalar threshold on the remix distance. To speed-up computation they use Locality Sensitive Hashing (LSH) using the threshold as a search radius. In a very similar work ([Casey 07]), 12-dimensional chromagram features extracted from 30 frames are concatenated into a single 360 dimensional vector, and then, LSH is used to efficiently solve an approximate nearest neighbor retrieval in a high dimensional euclidean space. Although they present very good results based on experiments performed in a large database, we have to note that *derivative works* (the term they use explicitly in this last paper, but commonly called remixes) might preserve exactly the tempo, the basic structure and the main tonal progression of the entire song. Also, they just use a database of remixes of just two artists (Miles Davis and Madonna).

There are also some papers from the last Music Information Retrieval Evaluation eXchange[6] (MIREX). The ones specifically designed to detect cover song variants are cited below. The best performing algorithm was [Ellis 06]. We can take also [Ellis 07] as a reference, where a more detailed description of the method is done. Basically, it uses a beat tracker to generate a beat-synchronous representation with one feature vector per beat. With this, the authors try to overcome variability in tempo. The beat tracking algorithm is based on the first-order difference along time in a log-magnitude Mel-frequency spectrogram, and uses Dynamic Programming to find the set of beat times that optimize onset strength at each beat (to prefer strongest onsets as beats) and the spacing between beats (to reflect the global tempo parameter). To deal with variation in instrumentation, they use 12-dimensional chroma features, which are compared using the cosine distance. It is to note that instead of using a coarse mapping of FFT bins to chroma classes, they use the phase-derivative within each FFT bin (like in a sinusoid-modeling-based preprocessing step) to both identify strong tonal components in the spectrum and to get a higher-resolution estimate of the underlying frequency. The final similarity value is obtained by looking for sharp peaks (local maxima) in a high-pass filtered cross-correlation of the entire beat-by-chroma representation for two tracks.

[Lee 06b] uses as a feature set a chord sequence identified by an HMM trained with audio from symbolic data and computes a distance between two chord sequence pair by the use of a DTW algorithm. For chord recognition, first a quantized 12-bin chromagram from the raw audio is computed, and then, an automatic chord recognition algorithm based on HMM is applied to get the chord sequence. To account for transposition, the authors estimate the key of the song to be the most frequent chord in the chord sequence. They also implemented an alternative version, using the first chord to be the key. So, every song is transposed to a C major key before sending it to a distance computing algorithm. After frame-level chord sequence was obtained for each song, they used the DTW algorithm to find the minimum alignment cost between the two songs. As a distance measure between chords, they defined

[6] The Audio Cover Song Identification task was first proposed in 2006. Web (last access on July 2007): http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song

a combined cost by the sum of a cost of being in 'aligned states', and a transition cost. The former was obtained by computing the chord-to-chord distance from the HMM parameters obtained, and the latter by taking the transition cost directly from the transition probability matrix in the HMM.

The method from [Sailer 06] uses pitch detection and melody quantization to obtain an estimation of a discrete pitch for a note candidate. The algorithm consists of two parts: an indexing program that extracts the melody (predominant voices) from audio files, and a retrieval tool that calculates a distance matrix. For pitch detection, a multi resolution FFT is used, yielding a voiced/non voiced detection and a pitch line for the voiced parts. This pitch line is quantized using note boundaries estimated directly from the pitch line and further spectral information. Then, note candidates of too short duration or insufficient loudness are discarded, and a discrete pitch is estimated. The extracted melodies are split into what they call *relevant pieces* (between 3 and 8 seconds long) that consist of neither too few nor too many notes. The look-up is carried out as a string alignment process. The relative change of a melody over time (descriptions of note transitions represented by note intervals and the ratio of inter-onset intervals) is used as the search alphabet. The alignment is carried out as a semi-local string alignment process (the whole query must match any part of the reference string).

Finally, a brief summary overviewing the works mentioned in sections 2.4 and 2.5 is given in tables 2.1, 2.2 and 2.3. The aim of these is to constitute a general overview of the state-of-the-art of the cover song identification task.

There are 3 tables, each of one describing a particular aspect of the forementioned methods. Table 2.1 exposes the principal features extracted from audio. Table 2.2 tries to describe the most common pre-processing steps before a matching or alignment procedure is applied. Finally, in table 2.3, these procedures are highlighted.

	RMS	Spectral	LFCC/MFCC	Chroma/PCP	HPCP
[Foote 00]	√				
[Yang 01]	√	√			
[Dixon 05]		√			
[Izmirli 05]				√	
[Müller 05]				√	
[Gómez 06a, Gómez 06b]					√
[Gómez 06c]					√
[Marolt 06]		√		√	
[Casey 06a]			√	√	
[Casey 06b, Casey 07]			√	√	
[Ellis 07]				√	
[Lee 06b]				√	
[Sailer 06]		√		√	

TABLE 2.1
Features used in cited references.

	VQ	Segment.	Summariz.	Beat info.	Chord ext.
[Foote 00]					
[Yang 01]		√			
[Dixon 05]		√			
[Izmirli 05]					√
[Müller 05]	√	√		√	
[Gómez 06a, Gómez 06b]					
[Gómez 06c]			√		
[Marolt 06]		√	√	√	
[Casey 06a]	√	√	√	√	
[Casey 06b, Casey 07]	√	√		√	
[Ellis 07]				√	
[Lee 06b]					√
[Sailer 06]		√		√	

TABLE 2.2
Pre-processing used in cited references.

	Correl.	Edit-Dist.	DTW	State Hist.	Indic. String	Shingling & Hashing
[Foote 00]			√			
[Yang 01]	√					
[Dixon 05]			√			
[Izmirli 05]			√			
[Müller 05]	√					
[Gómez 06a, Gómez 06b]			√			
[Gómez 06c]			√			
[Marolt 06]	√					
[Casey 06a]	√	√		√	√	
[Casey 06b, Casey 07]						√
[Ellis 07]	√					
[Lee 06b]			√			
[Sailer 06]	√					

TABLE 2.3
Matching or alignment technique used in cited references.

2.6 Discussion

There are some concerns or limitations in the existing literature about cover song identification (mainly works commented in previous sections 2.4 and 2.5). We now briefly highlight them.

Regarding the descriptors used, tonality descriptors such as PCP have proven to be more useful than MFCCs [Casey 06a]. Energy descriptors have also been used [Foote 00, Yang 01], but for slightly different purposes and with very old systems. Furthermore, in the same articles, these are partially discarded for a most challenging scenario such as pop music [Foote 00]. Melodies also seem to be appropriate [Sailer 06]. But other sequences of descriptors might be appropriate for this task too (further research is needed).

When using tonality descriptors, some papers do not specify how a local distance between these feature vectors is computed. They are supposed to assess the similarity of chroma features as the rest of articles do: with an euclidean-based distance (like the cosine distance) [Müller 05, Gómez 06b, Ellis 07].

Since tonality features such as chroma vectors or PCPs are proven not to be in an euclidean space [Shepard 82, Lewis 87, Cohn 97, Chew 00], this assumption seems to be wrong. Furthermore, any method (i.e., a classifier) using distances and concepts just valid for an euclidean space (i.e., [Casey 07]) will have the same problem.

When testing the performance of a system, the databases used either were very small, or just covering a very concrete range of styles (i.e., classical music or 'remixes', when in section 1.4.4 we've seen that these might not be the most challenging scenarios) [Yang 01, Müller 05, Casey 06b]. The first big effort to evaluate cover song identification systems in a large-scale test with a relatively big corpus of songs and multiple genres was been done in MIREX 2006[7], but we feel that just using 330 songs (the number of songs they used) might be not enough.

Another common resource is to use some 'intermediate' techniques such as beat tracking systems, key estimation algorithms, and chord and melody extraction engines [Izmirli 05, Gómez 06c, Lee 06b, Sailer 06]. 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, they may decrease the performance of a system comprising (at least) one of them [Marolt 06]. The same argument can be applied to any audio segmentation, chorus extraction, or summarization technique. If we want some numbers, we can look at MIREX 2006[8] task evaluations, where performances had a ceiling of a P score of 0.407 for beat tracking systems and a 82.5% of overall accuracy for melody extraction. We can also take a lookt to state-of-the-art approaches. For example, common accuracy values for a chord recognition engine range from 75.5% [Bello 05] to 93.3% [Lee 06c]. Also, in this last case, once you obtain the chords, the distance between them is still a not completely solved issue, involving some perceptual concepts that are not fully undestood. So, errors in these 'intermediate' processes might be added (in case we are using more than one of them), and propagated to the final performance of the overall system. Furthermore, this 'intermediate' processes may increase computation time substantially (to the point of making completely unaffordable to retrieve a query in a very large database containing millions of tracks).

Finally, many systems for cover song identification use a global alignment technique such as DTW or entire song cross-correlation for determining similarity [Izmirli 05, Müller 05, Lee 06b, Ellis 07] (except the ones that use a summarization, chorus extraction or segmentation technique [Gómez 06b, Marolt 06], which would suffer from the problem of the 'weakest link', cited in the previous paragraph). In our opinion, a system that considers only similarity between excerpts of songs (local similarity or alignment methods) are the only way to cope with strong song structure changes (see section 1.4.4).

In the system described in this thesis we have tried to overcome all these issues. We will see how in the following chapters.

[7] http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song_Identification_Results
[8] http://www.music-ir.org/mirex2006/index.php/Main_Page

Chapter 3

Evaluation methodology

This chapter has five different parts. The first one is devoted to an introduction on evaluation of MIR systems. Then, some statistics about the music collection used in the experiments performed in this work are shown. These statistics about the covers employed for evaluating the implemented algorithms are important to qualitatively assess the significance of the results obtained. After that, the next part looks at some particularities of the cover song identification task proposed, and we enter into specific evaluation issues. A discussion on the suitability of some measures is presented, and those used in this thesis are explained. Finally, we present a base-line experiment thought to be a basic reference for future comparisons.

3.1 On evaluation

Before the final implementation of any Information Retrieval (IR) engine, we must carefully consider the quality of the end-product of our efforts. This step can be described as a performance evaluation of a proposed solution. IR techniques can be essentially seen as heuristics: we try to guess something as similar as possible to the right answer. So, we have to measure how close to it we can come. Furthermore, evaluation methods are used in a comparative way to measure whether certain changes lead to an improvement in system performance. In particular, when tuning algorithm parameters, it is important to choose the evaluation measure that rewards what we think is a right answer (choosing a valid measure).

But, what can we measure that will reflect the ability of the system in retrieving answers? In fact, the performance we want to evaluate can be decomposed into two terms: effectiveness and efficiency. The former relates to the performance as to the objectives we were pursuing (to what degree did a search accomplish what desired? how well done in terms of relevance?), while the later does to performance as to costs (at what cost and/or effort, time?). In a system designed for providing data retrieval, the response time and the memory/storage space required might be metrics of high importance in measuring its performance, but in a system designed for providing Information Retrieval, other metrics are also of interest. In an very early work on evaluation of Information Retrieval Systems [Cleverdon 70], there are listed up to six quantities that one could measure:

1. The *coverage* of the collection, that is, the extent to which the system includes relevant matter.
2. The *time lag*, that is, the average interval between the time the search request is made and the time an answer is given.
3. The form of *presentation* of the output.

4. The *effort* involved on the part of the user in obtaining answers to his search requests.
5. The *recall* of the system, that is, the proportion of relevant material actually retrieved in answer to a search request.
6. The *precision* of the system, that is, the proportion of retrieved material that is actually relevant.

We should note that the effort (4) relies partially on effectiveness and partially on efficiency. We can also see that only the last two items of the list (recall (5) and precision (6)) are the ones which attempt to measure the effectiveness of an IR system. There has been much debate as to whether precision and recall are in fact the appropriate quantities to use as measures of effectiveness [Baeza-Yates 99], but they remain to be a commonly used 'standard'.

In an evaluation we should measure the retrieval relevance, but relevance is a subjective and situational notion. Different users may differ about the relevance or non-relevance of particular documents to given questions. A document can be relevant if it answers precise questions precisely, if it partially answers a question, if it suggests a source for more information, if it gives background information, if it reminds the user of other knowledge, etc. In this thesis, we leave out the speculations about relevance and assume that we know if a retrieved document is relevant or not (binary relevance), which is a common assumption in the Information Retrieval communities. It also makes sense in the case of cover song identification, where we have an objective ground truth (see next paragraph and section 1.3). For further knowledge on relevance issues we refer to [Borlund 03], [Saracevic 75] and [Saracevic 06].

3.2 Music Collection

It is the objective of this section (jointly with *Appendix A*) to present the music corpus we have assembled for this thesis, and which has been used to test and evaluate the systems implemented. We have compiled a music collection of 2053 songs. To our knowledge this is the largest database employed ever in experiments related to cover song identification. We have added some 'outliers' to the database in order to make the task more challenging. In fact, 140 songs were chosen that were of similar artists and styles than the ones covered. So, in our music corpus there were 1913 covers and 140 non-covered songs.

Within this 1913 songs, there were 451 *canonical versions*, and 1462 covers. By *canonical version* or 'original' song we mean the track recorded and sold in a media (CD, vinyl, etc.) by the first performer of the song. So, we have 451 different groups of covers, each one with its *canonical version*. The average number of covers per song is 4.24, ranging from 2 (the 'original' song + 1 cover) to 20 (the 'original' song + 19 covers). In figure 3.1 we can see a distribution of the number of covers.

When compiling the database, a special emphasis was put in the variety of styles employed. There can be found artists and performers of a wide spectrum of genres and styles, ranging from classical music, *a Capella* songs or string quartet performances to electronic music productions or heavy metal bands, including contry and folk versions, jazz standards, musical soundtracks, Latin-American songs, and pop and rock music (for more details see *Appendix A*). The music collection included some of the forms where a cover song can be placed in (highlighted in section 1.4.3), including remaster, acoustic version, live performance, demo song, and remix.

Due to the elevated computational time costs of the cover song identification algorithms implemented, we have not used the entire music collection for preliminary experiments. Instead, we have compiled 3 different song databases. These were not overlapping, and were intended to be the most representative as possible of the entire corpus. We can see some statistics of them in table 3.1.

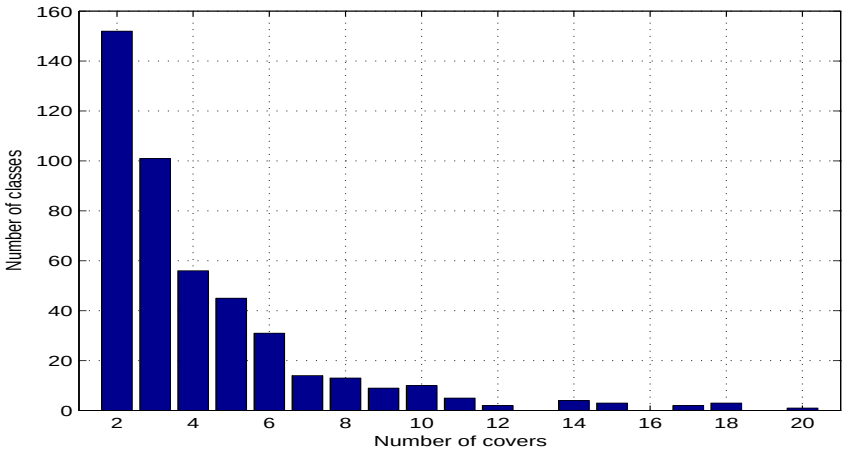

FIGURE 3.1 *Distribution of the number of covers. Plot of the number of classes (y axis) that contain x covers.*

Name of the database	Total number of songs	Groups of covers	Covers per group
DB75	75	15	5
DB330	330	30	11
DB2053	2053	451	4.24*

TABLE 3.1
Song compilations used in intermediate experiments. '' denotes average number of covers per group.*

In *DB75* and *DB330* there are no 'outliers'. *DB75* is mainly used for preliminary experiments. As the time complexity for cover song identification algorithms is quite high, some intermediate results are obtained with this database. As it will be shown in section 3.5, the results for the base-line experiment (random selection of pieces) are close to the ones achieved by *DB330*. This last was intended to be of similar dimensionality as the music compilation used in the MIREX 2006 contest[1]. *DB2053* is mainly used to compare effectiveness between best-performing methods.

A complete list of all the songs included in *DB2053* can be found in *Appendix A*, where songs are listed and grouped into *cover sets* (under the same song title). Artists are also listed and some details (such as if the song is a remaster or a live performance) are given.

3.3 Characteristics of the cover identification task

Going back to our application, we now provide some important details about the experiments done. We focus on the case where we have a database of songs (D, the music collection), and we have to evaluate an algorithm that given a song title (query q), yields the potential cover songs of it (A, a list of their titles, ranked from most similar to less similar). The query song is not retrieved (that is: $q \notin A$) and the 'outliers' are not queries (we do not have covers for the 'outliers', thus, the number of queries N_q is going to be the number of songs that have a cover).

As a ground truth we have listened and manually labeled[2] all the songs, indicating if they correspond

[1] http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song ; The Cover Song Identification task was first introduced in 2006.

[2] Do not confound the titles of the songs (to which we don't care), with the label we give them after listening.

to the same group (the same label is attached to the original song and covers of it) or not. Thus, our judgements are based on binary relevance (belonging to the desired group of covers or not).

For *DB75* and *DB330*, the length of the list of retrieved documents A is set to the the maximum number of covers per *canonical version* in order to be able to present all the relevant songs to a potential user in one output list. So, $|A| = 4$ for the former, and $|A| = 10$ for the latter (from now on, $|\cdot|$ will denote the number of items in a set). We also set $|A| = 10$ for *DB2053*. A cutoff like this is typically introduced in an Information Retrieval system because of the paged presentation of the search results.

Finally, we should highlight two aspects that we consider important for our system: ranking and recall. The order (ranking) in which the documents are presented in an answer set A should be relevant (as the algorithm attempts to partially define a similarity metric, and therefore, to provide the most similar songs at the beginning of the list). Another main objective of the desired algorithm is to maximize the amount of retrieved covers (recall). This is a shared objective with any audio identification task. Note that, in our experiments, all the documents in the collection where we want to search for are labelled (we know total recall values for each song).

3.4 Evaluation measures

We now present the evaluation measures considered, while discussing the suitability of them for evaluating a cover song identification system. Finally, all implemented measures for this task are reviewed.

3.4.1 MIREX 2006 evaluation measures

Audio cover song identification has been a very active topic of research within the last few years in the Music Information Retrieval (MIR) community, as it provides a direct way of evaluating music similarity algorithms. Some efforts are also being devoted to compare and evaluate different alternatives for this purpose (as MIREX[3]).

In order to compare our tests and algorithms to the existing ones, we, at first place, have considered the same evaluation measures that were used in the last MIREX 2006 cover song identification task. Altough we don't have the same database, in next chapters we will see that the values measuring performance results for our experiments coincide more or less with the ones obtained in the MIREX contest.

Used measures where: the *total number of covers identified* (*TNCI*), the *mean number of covers identified* (*MNCI*), the *mean of maxima* (*MM*) and the *mean reciprocal rank of the first correctly classified instance* (*MRR(1)*). Although some of these measures are very intuitive, we now explain them in detail.

The *total number of covers identified* (*TNCI*) was given in an absolute value, ranging from 0 to the maximum number of identified covers possible (assuming that we query all the covers, this value would be $N_q \times |A|$, where N_q represents the total number of queries).

The *mean number of covers identified* (*MNCI*) corresponds to the average number of correctly classified instances within the answer set.

$$MNCI = \frac{|R_a|}{|A|} \tag{3.1}$$

[3]Evaluation measures used for the cover song identification task in 2006 can be found at http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song_Identification_Results (last access on July 2007).

Where R_a is the set of relevant documents in a the answer set, and $|R_a|$ denotes the number of items in the set R_a (also known as the true positive rate). *MNCI* can be seen as an average performance measure.

The *mean of maxima* (*MM*) is an average of the best-case performance. For a given group of covers, only the best performance (highest *MNCI*) is kept. These values are then averaged over the number of cover groups.

With known item-search evaluation, one of the most used measures is the *Reciprocal Rank* (*RR*), where each relevant document contributes inversely proportionally to the rank of the document in the answer set. The *RR* is usually calculated for the first retrieved instance. This is useful for determining how far a user has to search to find the first relevant document.

$$RR(1) = \frac{1}{rank(1)} \tag{3.2}$$

Where $rank(1)$ is the rank of the first correctly classified instance. This measure is mostly used in systems answering questions (where usually just one answer is the correct one). *RR* can also can be extended to any number of instances of the answer set.

$$RR(k) = \frac{1}{k} \sum_{j=1}^{k} \frac{rel(j)}{rank(j)} \tag{3.3}$$

Here, k is the number of instances of the answer set we want to consider, rel_j is a binary variable indicating if the j-th document is relevant or not (we just set $rel(j) = 1$ in case it is relevant and $rel(j) = 0$ otherwise), and $rank(j)$ is the position of the j-th relevant document in the answer set. We have to note that calculating the RR until rank k, we discard documents below this rank.

The usual way to present results over several queries is computing the *Mean Reciprocal Rank* (*MRR*) averaging over all of them.

$$MRR(k) = \frac{1}{N_q} \sum_{i=1}^{N_q} RR(k)_i \tag{3.4}$$

Where N_q is the number of processed queries. So, the *mean reciprocal rank of the first correctly classified instance* is expressed as *MRR(1)*.

3.4.2 Other evaluation measures

A qualitative assessment of measures for the evaluation of a cover song identification system has been made in [Serrà 07], where other evaluation measures apart from the ones cited before are considered. There, the 'goodness' (construct validity) of an evaluation measure for a system like the one described in this thesis is discussed. Analyzed measures were *False/True Positives and Negatives* (*TP*, *FP*, *TN*, *FN*), *Sensitivity*, *Specificity*, the *Fallout Rate*, the *Receiver Operating Characteristic* (*ROC*) curve, and the *Lift Curve*. Furthermore, some popular IR measures were considered: *Precision*, *Recall*, the *Precision-Recall curve*, the *Break-even Point*, the *F-measure* and *Average Precision* (*AP*), *Reciprocal Rank* (*RR*), *Discounted Cumulative Gain* (*DCG*) and *Binary Preference-based measure* (*bpref* and *bpref-10*). We summarize here some of them while discussing some pros and cons.

Since we are assuming binary relevance, we start with some commonly used measures for binary classification. A simple 2-by-2 table (named contingency table) describing predictive performance can be done [Ye 03].

	Retrieved	Not retrieved
Relevant	*TP*	*FN*
Not relevant	*FP*	*TN*

TABLE 3.2
Error for binary relevance.

Thus, we have four combinations, two correct combinations in one diagonal and two incorrect combinations on the other one. The sets of *True Positives* (*TP*, hits) and *True Negatives* (*TN*, correct rejections) correspond to the correct answers, and the errors are broken into two sets: *False Positives* (*FP*, also called false alarms) and *False Negatives* (*FN*, misses). All the measures *TP*, *FP*, *FN* and *TN* provide us with important information but are not suitable for our task, as these features do not measure the rank (position) of the correctly classified instances. Furthermore, they do not consider the total number of relevant documents per query, so that they do not care, for instance, about the difference in retrieving the only possible item of a set, or one of the largest labelled group of covers (we want the former to have a higher reward than the latter).

An obvious alternative that may occur to the reader is to judge an Information Retrieval system by its *accuracy*, that is, the fraction of its classifications that are correct. This seems plausible, since there are two actual classes, relevant and not relevant, and an IR system can be thought of as a two class classifier which attempts to label them as such (it retrieves the subset of documents which it believes to be relevant). In terms of table 3.2:

$$Accuracy = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|} \tag{3.5}$$

We can conversely define the *Error Rate* (such that an *Error Rate* of 10% is the same as an *Accuracy* of 90%). Although *Accuracy* is the effectiveness measure which is usually used for evaluating machine learning classification problems, it is not an appropriate measure for Information Retrieval tasks. Here, in almost all circumstances, the data is extremely skewed: normally over 99.9% of the documents are in the not relevant category [Baeza-Yates 99]. In such circumstances, a system tuned to maximize accuracy will almost always declare every document to be not relevant.

The same four basic units of error and correctness are used in other disciplines, but the measures are slightly different:

$$Sensitivity = \frac{|TP|}{|TP| + |FN|} \tag{3.6}$$

$$Specificity = \frac{|TN|}{|TN| + |FN|} \tag{3.7}$$

$$Fallout = \frac{|FP|}{|FP| + |TN|} \tag{3.8}$$

Sensitivity is a measure identical to *Recall* (see equation 3.12 below), while *Specificity* is the inverse of *Sensitivity*. *Sensitivity* is a statistical measure of how well a binary classification test correctly identifies a condition, whether this be medical screening tests picking up on a disease or quality control in factories deciding if a new product is good enough to be sold. *Specificity* measures how well a binary classification test correctly identifies the negative cases, or those cases that do not meet the condition under study. The *Fallout rate* is also commonly named false alarm. *Accuracy*, *Specificity* and *Fallout rate* suffer from the problems cited above: data skewness, and not caring about the difference

in retrieving the only possible item of a set, or one of the largest labelled group.

The *Receiver Operating Characteristic* (*ROC*) curve (figure 3.2) is a plot of the true positive rate (or sometimes *Sensitivity*) against the false positive rate (or *1-Specificity*) for the different possible cutoff points of a ranked list (at rank 1, rank 2, etc.).

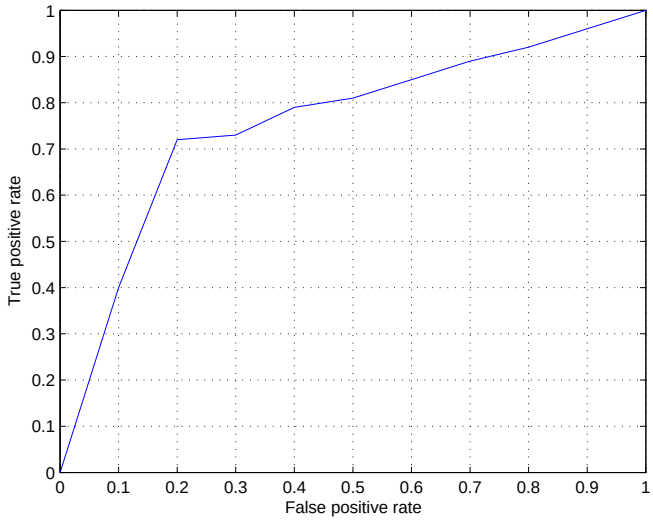

FIGURE 3.2 *Example of a ROC curve.*

The best possible prediction method would yield a point in the upper left corner or coordinate (0,1) of the *ROC* space, representing 100% *Sensitivity* (all true positives are found) and 100% *Specificity* (no false positives are found). For example, a *ROC* curve could make sense if looking over the full retrieval spectrum. In many fields, a common aggregate measure is to report the area under the *ROC* curve.

The *lift* curve (figure 3.3) plots cumulative true positive coverage (% of positive examples, y-axis) against the rank-ordered examples (% of examples, x-axis). A random ranking results in a straight diagonal line in this plot.

Unfortunately, if we plot the *ROC* and *lift* curves, we find the same problems as with *Accuracy*, *Specificity* and *Fallout rate*.

Binary relevance is a prototypical type of problem that occurs in many fields. For Information Retrieval applications there is usually a large amount of negative data. It would thus be useful to measure the classification performance by ignoring correctly predicted negative data. Two ratios have achieved particular prominence: *Precision* and *Recall* [Manning 07].

We define *Precision* as the probability that a retrieved object is relevant (or the ratio of relevant retrieved documents from the total of retrieved ones).

$$Precision = \frac{|R_a|}{|A|} \tag{3.9}$$

Here, $|R_a|$ is the number of relevant documents in the answer set A, and $|A|$ is the total number of documents in this answer set. Note that $|R_a| = |TP|$, and that $|A| = |TP| + |FP|$, so we can also define *Precision* in terms of *True Positives* and *False Positives*:

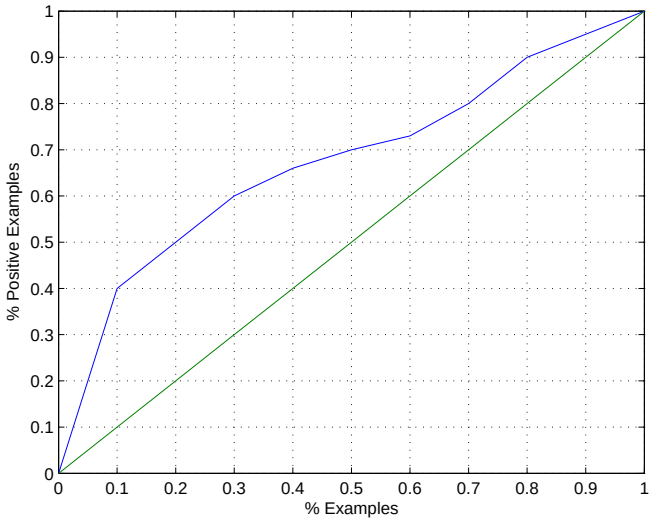

FIGURE 3.3 *Example of a Lift curve (blue upper line). Diagonal green line represents a random ranking.*

$$Precision = \frac{|TP|}{|TP| + |FP|} \tag{3.10}$$

Regarding *recall*, we define it as the probability that a relevant object is retrieved (fraction of relevant retrieved documents from the total of relevant documents). Therefore, the *Recall* measure is defined as follows:

$$Recall = \frac{|R_a|}{|R_q|} \tag{3.11}$$

Where $|R_a|$ is the number of relevant documents in the answer set (the same as before), and $|R_q|$ is the number of all relevant documents for the particular query in the entire test reference collection. Another time we have to note that $|R_q| = |TP| + |FN|$, so:

$$Recall = \frac{|TP|}{|TP| + |FN|} \tag{3.12}$$

Precision and *Recall* have been used extensively to evaluate the performance of retrieval algorithms. However, a more careful reflection reveals problems with these two measures [Baeza-Yates 99]. Although these seem to be good for evaluating a cover song identification system like the one presented in section 3.3 (*Recall* better than *Precision*, as we are attempting to identify the maximum number of covers possible), they fail in taking into account the position (rank) of the correctly retrieved items (*Precision* and *Recall* are set-based measures. They are computed using unordered sets of documents).

To evaluate ranked lists, *Precision* can be plotted against *Recall* after each retrieved document. To make such a curve, we only have to measure precision after 10 percent of documents retrieved, at 20%, 30%, etc. After interpolating these points we obtain the desired curve. With this, we obtain *Precision* at 11 *Recall* levels (0% is obtained by interpolation).

To take into account all the queries, as done with other measures, we average each *Precision* value over all queries to produce one curve:

$$P(r) = \frac{1}{N_q} \sum_{i=1}^{N_q} P_i(r) \tag{3.13}$$

Where r is the recall level, N_q is the number of queries that have been made, and $P_i(r)$ is the precision at the recall level r for the i-th query.

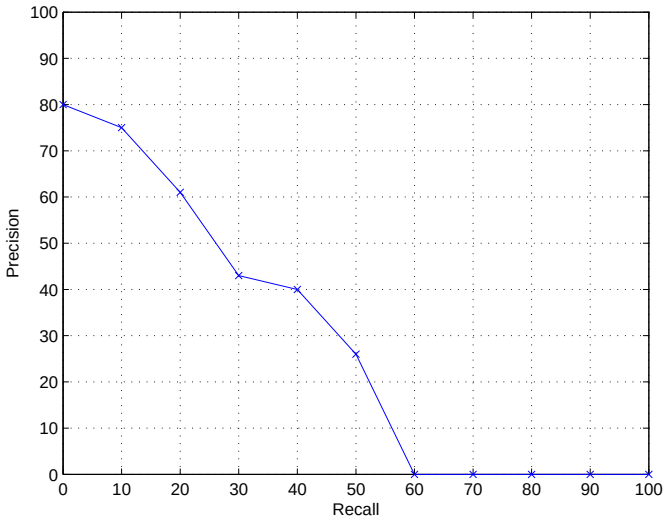

FIGURE 3.4 *Example curve: precision at 11 standard recall levels (%).*

A *Precision-Recall* curve provides a summary of the overall quality of the ranked list. The optimal graph would have a straight line (precision always at 100%), but typically, as recall increases, precision drops. This curve gives an idea about the ranking of the items, but it does not measure if we have retrieved all the possible elements. In addition, there are some problems when interpolating answers with just one relevant document. We also have to highlight some interpolation problems when there are equally spaced relevant documents in the answer set. Furthermore, the *Precision-Recall* curve has the problem that it is not just a value, and thus, it is a bit difficult to interpret it in some particular situations.

Precision at X (*P@X*) is the *Precision* achieved after X documents are retrieved. It counts the number of relevant items in the top X documents in the ranked list returned for a query. Commonly used values are *P@10* or *P@1*. The measure closely correlates with user satisfaction in tasks such as web searching (i.e., *P@10*), and is extremely easy to interpret. However, it is not a powerful discriminator among retrieval methods (the only thing that matters is a relevant document entering or leaving the top X), and does not average well (the constant cutoff represents very different recall levels for different queries, see [Buckley 00]).

Given a *Precision-Recall* curve, other measures can be easily computed in order to try to summarize the behaviour of such a curve into one single number. The most common ones are the *Break-even* point and the *Recall at X Precision* (*R@X*). The *Break-even* point is the value at which the *Precision* equals *Recall*, while *R@X* is the percentage of *Recall* we get when we reach a *Precision* of X. A common specification of this last one is *R@0.5*.

The idea behind *Average Precision* is to generate a single value summary of the ranking by averaging the *Precision* figures obtained after each new relevant document is observed. We just have to sum all

Precision values achieved when a relevant document is retrieved:

$$AP = \frac{1}{|R_a|} \sum_{j=1}^{|R_a|} P(j) \tag{3.14}$$

$|R_a|$ represents the number of relevant documents per query and $P(j)$ is the precision achieved when the j-th relevant document is retrieved. We use $AP = 0$ for non-retrieved documents. This measure can be interpreted as the area under the uninterpolated *Precision-Recall* curve, and provides a summary of the overall quality of the ranked list. It favors retrieval systems with relevant documents at the top of the ranked list (systems which retrieve documents quickly), and, of course, an algorithm might present a good average precision at seen relevant documents but have a poor performance in terms of overall recall.

AP is based on much more information than *P@10*, and is therefore a more powerful and more stable measure [Buckley 00]. Its main drawback is that it is not easily interpreted. For example, an *AP* score of 0.4 can arise in a variety of ways, whereas a *P@10* score of 0.4 can only mean that four of the top 10 documents retrieved are relevant.

For a query batch, the *Mean of Average Precision* (*MAP*) is often computed:

$$MAP = \frac{1}{N_q} \sum_{i=1}^{N_q} AP_i \tag{3.15}$$

MAP is the mean of the precision scores obtained after each relevant document is retrieved. Also, you can compute the *Geometric Mean of Average Precision* (*GMAP*).

$$GMAP = \sqrt[N_q]{\prod_{i=1}^{N_q} AP_i} \tag{3.16}$$

Where N_q is the number of queries and AP_i is the *Average Precision* for the i-th query. Alternatively, it can be calculated as an arithmetic mean of logs [Voorhees 06]. The *GMAP* measure is designed for situations where you want to highlight improvements for low-performing query-answers. It is the geometric mean of per-topic average precision, in contrast with MAP, which is the arithmetic mean. We can show these important differences in highlighting improvements for low-performing answers between both measures with an example. For instance, if a run doubles the average precision for q_k from 0.02 to 0.04, while decreasing $q_{k'}$ from 0.4 to 0.38, the arithmetic mean is unchanged, but the geometric mean will show an improvement.

AP seems better than *Precision* or *Recall* alone (we are able to distinguish between differently ranked answers), but we feel that ranking matters a lot. It also does not consider if we have retrieved all possible elements of a cover set.

F-measure is often used to measure the performance of a system when a single number is preferred instead of considering a *Precision-Recall* curve. This measure penalizes low *Precision* or *Recall*, which is good in most cases.

$$F\text{-}measure = \frac{Precision \cdot Recall}{(1-a) \cdot Precision + a \cdot Recall} \tag{3.17}$$

Inverse relationship between *Precision* and *Recall* forces general systems to go for compromise between them. But some tasks particularly need good *Precision* whereas others need good *Recall*. With high a, *Precision* is more important, whereas with low a, *Recall* becomes more important. Typically

$a = 0.5$ (which gives equal weights to them), what makes *F-measure* often to be defined as the harmonic mean of *Precision* and *Recall*. Some examples of precision-critical tasks would be the ones that: have little time available, a small set of relevant documents answers the information need or the ones that have potentially many documents that might fill the information need redundantly. On the other hand, recall-critical tasks could be the ones where: time matters less, one cannot afford to miss a single document or whereas one needs to see each relevant document.

With the *F-measure*, we use the harmonic mean because we can always get 100% *Recall* by just returning all documents, and therefore, we can always get a 50% arithmetic mean by the same process. This strongly suggests that the arithmetic mean is an unsuitable measure to use [Manning 07]. In contrast, if we assume that 1 document in 10000 is relevant to the query, the harmonic mean score of this strategy is 0.02%. The harmonic mean, the third of the classical Pythagorean means, is always less than or equal to the arithmetic mean and the geometric mean.

Finally, we have to note that the *F-measure* and other measures obtained by combining *Precision* and *Recall* also suffer from their drawbacks.

Another popular rank-based evaluation measure apart from the *Reciprocal Rank* (*RR*) is the *Discounted Cumulative Gain* (*DCG*). This measure is based on the intuition that after the first relevant document is found, the second is less useful.

$$DCG = rel(1) + \sum_{j=2}^{|A|} \frac{rel(j)}{\log(j)} \tag{3.18}$$

Here, $rel(j)$ indicates if the j-th retrieved document in the answer set A is relevant or not (0 or 1). $|A|$ is the number of answers for a given query and j stands for the rank of these answers. This measure can be adapted to different relevance levels just by weighting $rel(j)$ ($rel(j)' = w_j \cdot rel(j)$). A good example of this and a comparison with standard *precision* and *recall* measures is done in [Jarvelin 00].

Regarding both *RR* and *DCG*, we find again that ranking matters a lot. These measures also do not consider $|R_q|$. Also, considering only the rank of the first correctly classified instance (*RR(1)*) seems inappropriate since we are not dealing with an only-one-answer system.

Binary Preference-Based Measure (*bpref*) was introduced in [Buckley 04]. This measure is a function of how frequently relevant documents are retrieved before non-relevant documents.

$$bpref = \frac{1}{|R_q|} \sum_{j=1}^{|R_a|} \left(1 - \frac{N_{nr}(j)}{|R_q|}\right) \tag{3.19}$$

Where $|R_q|$ is the total number of relevant documents in the collection, $|R_a|$ is the number of relevant documents in the answer, and $N_{nr}(j)$ is the number of judged non-relevant documents ranked before the j-th relevant retrieved document in the ordered set R_a.

Bpref can be thought of as the inverse of the fraction of judged irrelevant documents that are retrieved before relevant ones. Unlike previous measures, it was originally designed for situations where relevance judgments are known to be far from complete, and it only uses information from judged documents. Of course it can also be used when we have judged all documents in our collection. In fact, when comparing systems over test collections with complete judgments, *MAP* and *bpref* are reported to be equivalent, but with incomplete judgments, *bpref* is shown to be more stable [Buckley 04].

Bpref works well most of the time in practice. However, it is excessively coarse when the number of relevant documents is very small (one or two) because the evaluation is then restricted to a very few document pairs. For this reason, the authors first introduced a variant called *bpref-10* where the

evaluation is guaranteed to use at least ten document pairs:

$$bpref\text{-}10 = \frac{1}{|R_q|}\sum_{j=1}^{|R_a|}\left(1 - \frac{N_{nr}(j)}{10 + |R_q|}\right) \tag{3.20}$$

Many variants for this measure are used, even for TREC [Voorhees 06], or for taking into account graded relevance judgements [De Beer 06].

We do not consider *Spearman's* ρ [Snedecor 89], *Kendall's* τ [Kendall 48] and *Maximum Marginal Relevance* (*MMR*) [Carbonell 98] because our data does not fit to the models they were thought of. Basically, we do not have a true measure of similarity for our ground truth (our cover songs are not ranked from more similar to less similar, we just only know if they are a cover of a given query of not (binary relevance)). Also, the *MMR* measure would not be suitable for our purposes because we do not want to introduce the concept of novelty in our answer lists, we want to gather relevant songs in a database and provide an answer set with minimum novelty (more focused, even, on detecting duplicates than on improving novelty).

For more details on evaluation measures for Information Retrieval systems we refer to [Baeza-Yates 99, Ye 03, Buckley 04, Manning 07, Voorhees 06].

3.4.3 Implemented measures

For evaluation purposes, we have implemented a set of algorithms performing different evaluation measures in order to see improvements in our tests. Implemented measures are: *TNCI*, *MNCI*, *MM*, *MRR(1)*, *P@1*, and *F-Measure*, average *Precision-Recall* curve. Furthermore, we have seen that *bpref** and normalized *lift* curves can be very informative for our task [Serrà 07]. So, we also have incorporated them in our evaluation procedure.

*Bpref** stands for a variant of *bpref* [Buckley 04] that fit our evaluation requirements regarding ranking and recall. The formulation of *bpref** is similar to *bpref-10*:

$$bpref^* = \frac{1}{|R_q|}\sum_{j=1}^{|R_a|}\left(1 - \frac{N_{nr}(j)}{|A| + |R_q|}\right) \tag{3.21}$$

Where $|R_q|$ is the total number of relevant documents in the collection, $|R_a|$ is the number of relevant documents in the answer set, and $N_{nr}(j)$ is the number of judged non-relevant documents ranked before the j-th relevant retrieved document in the ordered set R_a. Instead of the 10 introduced in *bpref-10* we use the value of $|A|$.

If we consider the fact that *bpref* and its variants have the additional property of dealing with unjudged information (not labelled elements), which allows us, for instance, to introduce outliers [4] to our database without affecting our evaluation measure [Buckley 04], *bpref** becomes our choice of preference.

A normalized *lift* curve was also found to be a suitable evaluation measure. It consists of plotting the percentage of positive examples normalized by the total number of relevant queries ($\%Pos.Ex/|R_q|$) versus the ratio of examples normalized by the length of the answer set ($\%Ex./|A|$).

In the following sections we will evaluate different experiments. Although the evaluation has been made with all the implemented measures cited in this section, in this thesis we will only show some of them for clarity and space reasons (most of them are redundant in many occasions). The measures

[4] Songs that we have not listened, streams of radio with excerpts of songs that we don't know if are relevant or not, etc.

chosen are: *bpref**, *F-measure*, and *MNCI*. As we have argued here and in [Serrà 07], *bpref** reflects the performance aspects that we consider most important; we will show the geometric mean of it. In addition, a very common measure used in many Information retrieval systems will be always used: the *F-measure*. Finally, an intuitive measure such as the *mean number of covers identified* (*MNCI*) will be also shown. When necessary, other measures will be mentioned just to compare with other experiments and existing methods (such as the MIREX 2006 evaluation measures explained in section 3.4.1). In certain occasions, a normalized *lift* curve might be plotted to provide more visual information.

3.5 Base-line experiment: random selection of similar pieces

As a first starting point, we performed a test in order to see the nature of our music collection, and to get an impression of how much easy was for a system to find a cover song in our database. The experiment consisted in randomly selecting songs for each query from a given database, and computing the evaluation measures. It was made for the three databases defined in section 3.2, and the average vales for 10 runs was taken. We first show the normalized *lift* curve for this experiment in figure 3.5, and then, in table 3.3, a summary of the results can be seen.

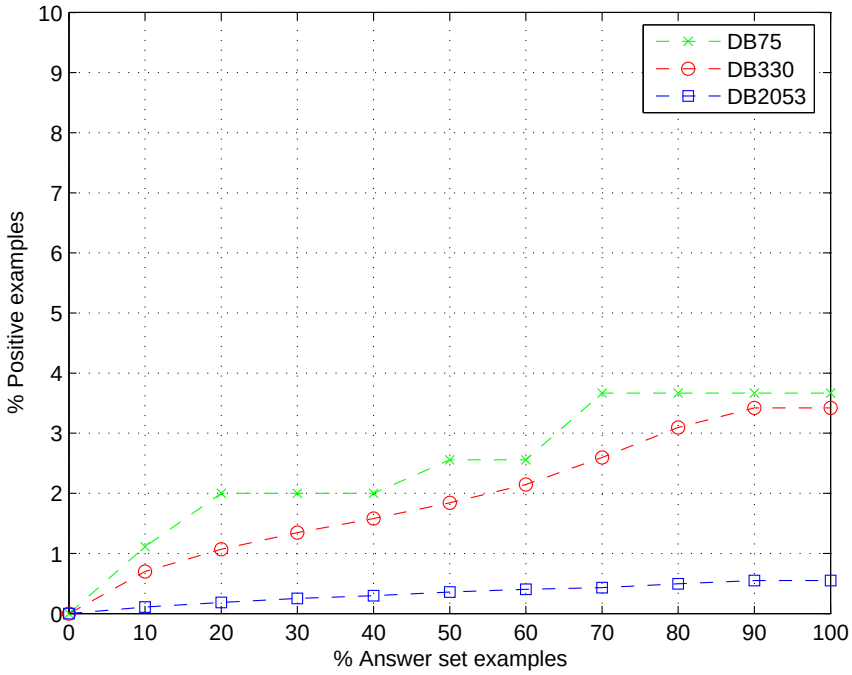

FIGURE 3.5 *Normalized lift curve for the base-line experiment. Note that the y-axis is zoomed to just 10%.*

For *DB75* and *DB330*, the length of the retrieved documents list A is set to the maximum number of covers per *canonical version*. Then, $|A| = 4$ for the former, and $|A| = 10$ for the latter. Also, $|A| = 10$ for *DB2053*. This specification remains constant for the rest of the experiments performed in this thesis (see section 3.3).

In general, we can see that these results are very unsatisfactory (all measures of table 3.3 range between 0 and 1 and the normalized lift curve y axis is zoomed to just 10%), what suggests that these song databases should be suitable for evaluation purposes. We can also see that all the implemented

Evaluation measure	DB75	DB330	DB2053
MNCI	0.035	0.031	0.003
F-measure	0.047	0.043	0.005
GM-bpref*	0.015	0.011	0.001

TABLE 3.3
Implemented evaluation measures results for the base-line experiment. GM corresponds to the geometric mean.

evaluation measures seem to scale linearly with the number of songs when using collections bigger than *DB330*. This allows us to work with this database and then extrapolate to bigger collections.

Chapter 4

Experiments with state-of-the-art methods

The goal of these experiments is to implement and evaluate existing state-of-the-art methods with the database presented in previous section 3.2, and to compute a performance reference value for new algorithms while testing several important parameters of them.

We have followed two approaches reported in the *Scientific Background* (section 2.5). We have chosen them because they represent in many ways the state-of-the-art. Their main characteristics are that they use global alignment techniques and that they operate in an euclidean feature space. In subsequent section titles, we differentiate these two methods by its alignment procedure (cross-correlation and Dynamic Time Warping), but, as we have noticed in sections 2.5 and 2.6, other procedures are characteristic for each one (such as the features used, the distance between feature vectors, etc.). Further justifications about the selection of these two methods are done in the corresponding sections.

In addition, several variants of the two main methods have been introduced for testing the suitability of some parameters that we had to fix, certain techniques, etc. This variants are summarized in the sections 4.2 and 4.4. The objective of these tests was to further improve the system performance, as well as to analyze which techniques or parameter values were relevant.

4.1 Cross-correlation approach

A quite straightforward approach is the one presented in [Ellis 06, Ellis 07]. This method finds cover versions by cross-correlating sequences of chroma vectors (the whole song) averaged beat-by-beat. Furthermore, it seems to be a good starting point because it was found to be significantly superior to the other methods presented in MIREX 2006 contest[1]. The system identified 761 cover songs (out of a theoretical maximum of 3300), and had a *MRR(1)* of 0.49. The next best performing system [Lee 06b] identified 365 covers with and *MRR(1)* of 0.22. Submissions included 4 systems specially designed for cover song detection (we have explained them in section 2.5) and 4 general music similarity systems (focused to genre or artist classification, and mainly aimed at capturing a listener's judgement of similarity between pieces of music).

[1] http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song_Identification_Results

4.1.1 Implementation

We have implemented a very similar version of the forementioned system. We have not used the implementation available on the web[2] because we wanted to use the same features for all the methods tested and because it was more straightforward for us to introduce new functionalities and improvements on an implementation made on our own. Furthermore, the original implementation is in Matlab^R, and it was too time consuming for performing several tests. We now describe the steps followed in our implementation.

First of all, 12-bin HPCP features are extracted [Gómez 06a]. For doing so, we cut the audio signal into short overlapping frames of 4096 samples of at 44.1 KHz sampling rate (f_s) with a hop size (T_{HPCP}) of 2048. Each one of these feature vectors is then normalized dividing by its maximum amplitude within the feature extraction process. For further information we refer to the reader to section 2.1.1, where this extraction process has been properly described.

In addition, beat timestamps are computed with an algorithm adapted from [Davies 05a, Brossier 07] using the *aubio* library[3] (explained in 2.1.2).

We then average HPCPs for each beat. If we have a sequence of beat timestamps $B_i = b_1, b_2, \dots, b_I$ and a sequence of HPCP vectors $HPCP = \overrightarrow{h_1}, \overrightarrow{h_2}, \dots, \overrightarrow{h_n}$, we get an averaged sequence of normalized HPCPs ($HPCP' = \overrightarrow{h'_1}, \overrightarrow{h'_2}, \dots, \overrightarrow{h'_I}$, and $I \ll n$). Each new HPCP vector ($\overrightarrow{h'_j}$) is computed as:

$$\overrightarrow{h'_j} = \sum_{i=k}^{k+\Delta k} \overrightarrow{h_i} \tag{4.1}$$

Where we sum all vectors in a given interval, and then:

$$\overrightarrow{h'_j} = \frac{\overrightarrow{h'_j}}{\|\overrightarrow{h'_j}\|} \tag{4.2}$$

Where we divide by the norm of obtained feature vector $\overrightarrow{h'_j}$. k and Δk correspond to the following equations:

$$k = b_j \cdot f_s \cdot \frac{1}{T_{HPCP}} \tag{4.3}$$

$$\Delta k = (b_{j+1} - b_j) \cdot f_s \cdot \frac{1}{T_{HPCP}} \tag{4.4}$$

With this, we get a beat-synchronous representation with one normalized feature vector h'_j per beat (a sequence of HPCP vectors (HPCP matrix) like in figure 4.1). So, variability in tempo among different tracks is overcome.

In addition, a *global HPCP* is computed by averaging all feature vectors in a song:

$$\overrightarrow{GlobalHPCP} = \sum_{j=0}^{I} \overrightarrow{h'_j} \tag{4.5}$$

$$\overrightarrow{GlobalHPCP} = \frac{\overrightarrow{GlobalHPCP}}{\|\overrightarrow{GlobalHPCP}\|} \tag{4.6}$$

Where I is the total number of beat timestamps, and $\|\cdot\|$ represents the norm of the vector.

[2] http://labrosa.ee.columbia.edu/projects/coversongs
[3] http://aubio.piem.org

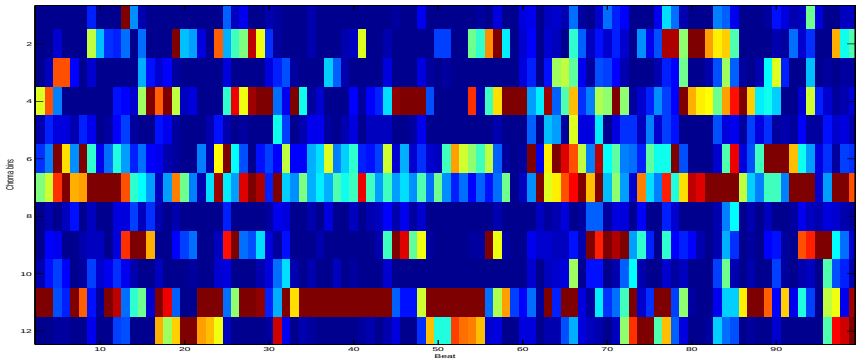

FIGURE 4.1 *12-bin HPCP matrix example averaging across beats.*

Then, when we want to compare two songs, they are transposed to a common key. This is done by shifting in just one song the bins of each HPCP vector by a defined index. We call this index the *optimal transposition index* (from now on *OTI*). For two songs A and B, this is calculated as:

$$OTI = argmax_{0 \leq id \leq N_B - 1}\{\overrightarrow{GlobalHPCP_A} \cdot circularshift(\overrightarrow{GlobalHPCP_B}, id)\} \tag{4.7}$$

Where '$\cdot$' indicates a dot product, N_B is the number of bins of the feature vector considered (in this case 12), and $circularshift(\overrightarrow{h}, id)$ is a function that rotates a vector (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. A circular-shift can also be expressed as:

$$circularshift(h[x], id) = h[((x + id))_{N_B}] \tag{4.8}$$

Where vector $\overrightarrow{h}$ is now represented by $h[x]$, N_B is again the number of components of the vector, and $((x + id))_{N_B}$ is the modulo N_B of $x + id$.

So, for each column of one of the two HPCP matrices we do:

$$\overrightarrow{h''_j} = circularshift(\overrightarrow{h'_j}, OTI) \tag{4.9}$$

And we get the transposed version of the HPCP matrix $HPCP''_B$ for one song (song B in this case).

Finally, a simple cross-correlation between the two HPCP matrices ($HPCP'_A = \overrightarrow{h'_{A,1}}, \overrightarrow{h'_{A,2}}, \dots, \overrightarrow{h'_{A,n}}$ and $HPCP''_B = \overrightarrow{h''_{B,1}}, \overrightarrow{h''_{B,2}}, \dots, \overrightarrow{h''_{B,m}}$) is done:

$$(HPCP'_A \star HPCP''_B)_i = \sum_j d(\overrightarrow{h'_{A,j}}^*, \overrightarrow{h''_{B,j+i}}) \tag{4.10}$$

Where the sum is over the appropriate values of j and a superscript asterisk indicates the complex conjugate. The HPCP distance used (d) is the cosine distance, thus, the feature vectors are assumed to be in an euclidean space. This might be a wrong assumption as we had reasoned in section 2.6. The cosine distance between two vectors is defined as:

$$d(\overrightarrow{h_i}, \overrightarrow{h_j}) = \frac{\overrightarrow{h_i} \cdot \overrightarrow{h_j}}{\|\overrightarrow{h_i}\| \cdot \|\overrightarrow{h_j}\|} \tag{4.11}$$

Where '$\cdot$' means the dot product between vectors. Note that if the vectors are divided by its norm (as we have done in equation 4.2), the dot product corresponds to the cosine distance.

The cross-correlation is similar in nature to the convolution of two functions. Whereas convolution involves reversing a signal, then shifting and multiplying it by another signal, correlation only involves shifting and multiplying it (no reversing). The cross-correlation is further normalized by the length of the shorter segment, so the correlation results are bounded to lie between zero and one.

In the end we get a cross-correlation vector of $n + m - 1$ components (figure 4.2). In cited references, 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.

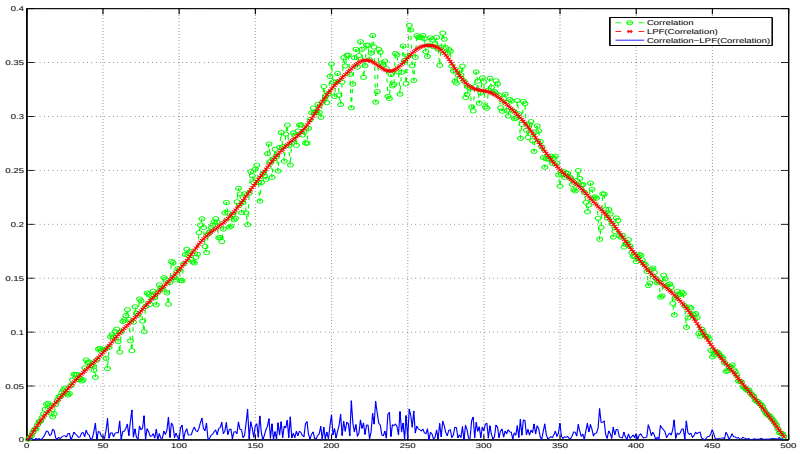

FIGURE 4.2 *Beat-by-beat cross-correlations between two songs. We can see the cross-correlation vector (green circles), the low-pass filtered cross-correlation (red crosses), and the difference between them (continuous blue line at the bottom). Peaks for cross-correlations and low-pass filtered cross-correlations are more or less in the middle of the song. This is an effect of the cross-correlation formula. The high-pass filtered cross correlation (difference between them) does not present this phenomena.*

The final distance measure representing the similarity between two songs is obtained with the reciprocal of the maximum peak value of this high-pass filtered cross-correlation.

4.1.2 Evaluation

We show in table 4.1 the results for the implemented approach. We can see that, with *DB75*, we practically reach 50% of good answers in a 4-items answer set per query (*MNCI*). This value corresponds more or less with the accuracy value reached with the "personal music collection" that was employed in [Ellis 07] to test a very similar approach (59%). This last collection was reported to be of 94 pairs of songs.

Evaluation measure	DB75
MNCI	0.477
F-measure	0.539
GM-bpref*	0.226

TABLE 4.1
Cross-correlation approach evaluation with DB75. Three different evaluation measures are shown.

4.2 Improvements to the cross-correlation approach

Starting from the method implemented previously (section 4.1), we tested several parameters which had implications for the final results that were not reported in the literature.

4.2.1 Implementation

In this stage of the experiments we were interested in measuring the influence of three parameters to the overall performance results: the number of HPCP bins used, the way key transposition was done, and the affection the HPCP distance measure used.

Number of HPCP bins Three values were tested: 12, 24 and 36 bins (that is, 1, 1/2 and 1/3 of a semitone resolution). These parameters were changed in the HPCP extraction method (section 2.1.1).

Key transposition To account for performances played in a different tonality, we calculated a global HPCP vector (equations 4.5 and 4.6) and we transposed (shifted) one HPCP matrix the desired number of bins (*OTI*, equations 4.7 and 4.9).

To check if we can really trust this concrete procedure, an alternative way of computing a transposed matrix has been introduced. This consists on calculating the main key for each piece using an algorithm for key estimation [Gómez 06a] (see also section 2.1.1). This algorithm is a recent state-of-the-art approach with an accuracy of 75% for real audio pieces [Gómez 04], and scoring among the firstly classified algorithms in the MIREX 2005 contest[4] with an accuracy of 86% with synthesized MIDI files.

In our procedure, once the main key is calculated, the whole song is transposed to a reference key.

HPCP distance In order to check the importance of the distance measure used between HPCPs, we introduced another similarity measure apart from the cosine distance used in section 4.1.1. This new similarity measure corresponds to the correlation between feature vectors.

Correlation has been used in [Gómez 06a, Gómez 06c], and is inspired in cognitive knowledge of musical pitch [Krumhansl 90]. Furthermore, in the first reference it was found to work better than a simple euclidean-based distance between HPCP vectors for key extraction purposes.

For sampled variables of n components correlation (ρ) is expressed as:

$$\rho = \frac{\sum_{k=1}^{n}(x_k - \mu_x)\cdot(y_k - \mu_y)}{(n-1)\cdot\sigma_x\cdot\sigma_y} \tag{4.12}$$

Where μ_x and μ_y are the sample means, and σ_x and σ_x are the standard deviations. When correlating two N_B-bin HPCPs (12, 24 or 36 for our experiments) the equation becomes:

[4] http://www.music-ir.org/mirex2005/index.php/Audio_and_Symbolic_Key_Finding

$$\rho(\overrightarrow{h_i}, \overrightarrow{h_j}) = \frac{\sum_{k=1}^{N_B}(h_i(k) - \mu_{h_i}) \cdot (h_j(k) - \mu_{h_j})}{(N_B - 1) \cdot \sigma_{h_i} \cdot \sigma_{h_j}} \tag{4.13}$$

Where again μ_{h_i} and μ_{h_j} are the HPCP means, and σ_{h_i} and σ_{h_j} are the HPCP standard deviation. Notice that calculating equation 4.13 with standardized feature vectors (substracting the mean of the vectors and dividing by their standard deviation) is similar to calculate the cosine distance with normalized vectors. It all leads to a dot product operation.

4.2.2 Evaluation

We have summarized the results for the three parameters explained in the previous section in a 'multiple measure' table (tables 4.2, 4.3 and 4.4). From these tables, some conclusions must be highlighted.

	d_{COS} + Key	d_{CORR} + Key	d_{COS} + Trans	d_{CORR} + Trans
12 bins	0.3500	0.3967	0.4767	0.4967
24 bins	0.4000	0.4166	0.4433	0.4800
36 bins	0.4166	0.4333	0.5333	0.5733

TABLE 4.2

Improved cross-correlation approach evaluation with DB75. Mean number of covers identified (MNCI) value for cosine distance (d_{COS}), correlation distance (d_{CORR}), OTI transposition (Trans) and key estimation algorithm (Key).

	d_{COS} + Key	d_{CORR} + Key	d_{COS} + Trans	d_{CORR} + Trans
12 bins	0.4173	0.4643	0.5392	0.5595
24 bins	0.4725	0.4973	0.5200	0.5531
36 bins	0.4867	0.5047	0.6035	0.6376

TABLE 4.3

Improved cross-correlation approach evaluation with DB75. F-measure value for cosine distance (d_{COS}), correlation distance (d_{CORR}), OTI transposition (Trans) and key estimation algorithm (Key).

	d_{COS} + Key	d_{CORR} + Key	d_{COS} + Trans	d_{CORR} + Trans
12 bins	0.1392	0.1719	0.2260	0.2461
24 bins	0.1796	0.1961	0.2340	0.2723
36 bins	0.2044	0.2209	0.3161	0.3812

TABLE 4.4

Improved cross-correlation approach evaluation with DB75. GM-bpref value for cosine distance (d_{COS}), correlation distance (d_{CORR}), OTI transposition (Trans) and key estimation algorithm (Key).*

The first conclusion is related to the resolution of the HPCP feature vectors. We show how, as the number of bins increases, we get better performance. With all the experiments, and being independent of the HPCP distance used and the transposition made, the greater the resolution of the HPCP, the better the performance we get.

Secondly, we see that the transposition method employed in section 4.1.1 outperforms the transposition to a reference key alternative. We can check in previous tables that this statement is independent of the number of bins and the HPCP distance used. So, it seems appropriate to transpose the songs according to the *OTI* of the global HPCP vectors.

Thirdly, we observe that HPCP distance plays a very important role. This aspect of the system implies more than a 13% of difference in the *mean number of covers identified* (*MNCI*) for some tests

(as an example, take a look at the 12-bin experiment, where we have $0.3967/0.35 = 1.1334\ldots$). In all trials done, correlation between HPCPs is found to be a better similarity measure than cosine distance. The former gives a mean improvement among the tested variants of a 6.5% in the *mean number of covers identified*.

4.3 Dynamic Time Warping approach

Another approach for detecting cover songs was implemented, which reflects the most used alignment technique in the literature: Dynamic Time Warping (DTW). The followed method has a very high resemblance with the one exposed in [Gómez 06a]. We now explain the first implementation made and it's evaluation results.

4.3.1 Implementation

We proceed by extracting HPCP features [Gómez 06a] (described in section 2.1.1) in the same way we did with the cross-correlation approach (section 4.1.1) except that now, 36-bin feature vectors are extracted, as we have seen that these may lead to a better performance (section 4.2.2). Then, these feature vectors are then normalized by dividing by its maximum amplitude as well.

In this implementation we do not use any beat tracking method because DTW is an algorithm specially designed for dealing with tempo variations (we will see a discussion on this matter in the *Improvements to the DTW approach* (section 4.4)). For speeding up calculations, and just for this first experiment, we average HPCP vectors in groups of 20, which corresponds to a time slot of 0.95 sec. This value corresponds to what we call the *averaging factor* in section 4.4 (see specially section 4.4.2 to check about the suitability of different values for this parameter). So, each new HPCP vector becomes:

$$\overrightarrow{h'_j} = \frac{\sum_{i=k}^{k+20} \overrightarrow{h_i}}{max\{\sum_{i=k}^{k+20} \overrightarrow{h_i}\}} \tag{4.14}$$

Where we divide by the maximum value to get a feature vector normalized between 0 and 1. k corresponds to the number (or index) of the desired HPCP vector ($HPCP' = \overrightarrow{h'_1}, \overrightarrow{h'_2}, \ldots, \overrightarrow{h'_k}, \ldots, \overrightarrow{h'_n}$):

$$k = (j-1) \cdot 20 \tag{4.15}$$

With this, we get an averaged HPCP matrix representation ($HPCP'$) such as the one in figure 4.3.

From here, we compute a global HPCP by averaging all HPCP vectors found in $HPCP'$:

$$\overrightarrow{GlobalHPCP} = \frac{\sum_{j=1}^{n} \overrightarrow{h'_j}}{max\{\sum_{j=1}^{n} \overrightarrow{h'_j}\}} \tag{4.16}$$

Where we also divide by the maximum value ($max\{\}$).

Then, we calculate an OTI as done in previous section 4.3.1 (equation 4.7). We also transpose one of the two songs being compared as explained in the same section by means of a circular shift (equations 4.8 and 4.9).

What we now do is to perform a DTW based on the two sequences of HPCP vectors. As explained in section 2.2.1, with DTW we find an optimal match between two given sequences of different length $HPCP_A = \overrightarrow{h_{A,1}}, \overrightarrow{h_{A,2}}, \ldots, \overrightarrow{h_{A,n}}$ and $HPCP_B = \overrightarrow{h_{B,1}}, \overrightarrow{h_{B,2}}, \ldots, \overrightarrow{h_{B,m}}$.

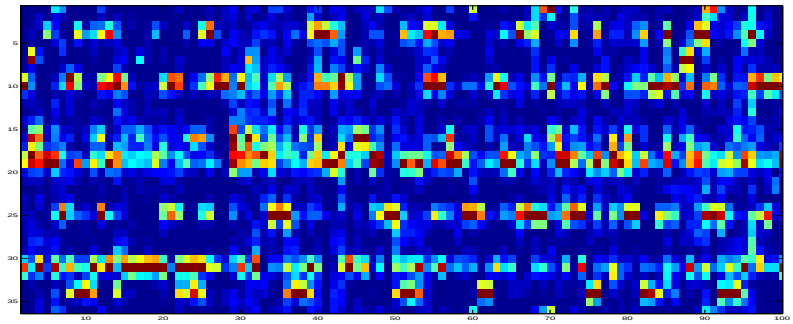

FIGURE 4.3 *36-bin (y-axis) HPCP matrix example (averaged HPCPs, x-axis). We show the same song as in figure 4.1 for comparison purposes.*

To align these two sequences, we construct an $n \times m$ matrix where the (i-th, j-th) element of the matrix corresponds to the value of a local cost function. This local cost function is determined to be 1 minus the correlation between HPCP vectors explained in section 4.1.1. So, the distance between two HPCP vectors is calculated as:

$$d(\overrightarrow{h_{A,i}}, \overrightarrow{h_{B,j}}) = 1 - \rho(\overrightarrow{h_{A,i}}, \overrightarrow{h_{B,j}}) \tag{4.17}$$

Where $\rho(\overrightarrow{h_i}, \overrightarrow{h_j})$ is the correlation between $\overrightarrow{h_i}$ and $\overrightarrow{h_j}$ defined in equation 4.13. As the value of the correlation ρ ranges from -1 to 1, the value of the distance d will be between 0 and 2.

As we were saying, an $n \times m$ matrix D is constructed, which follows the simple DTW recurrent relation (an example of the resultant matrix when comparing two cover songs can be seen in figure 4.4):

$$D(i,j) = d(\overrightarrow{h_{A,i}}, \overrightarrow{h_{B,j}}) + min\{D(i-1,j-1), D(i-1,j), D(i,j-1)\} \tag{4.18}$$

In section 2.2.1, we have seen that with this Dynamic Programming (DP) technique, we obtain the cumulative distance for the best alignment between two HPCP matrices in matrix element (i,j). Therefore, in element (n,m) we have the total alignment cost. We have also seen that we can obtain an alignment path just backtracking from element (n,m) and choosing the smallest matrix position near it ($D(n-1,m-1)$, $D(n-1,m)$, or $D(n,m-1)$) recursively. That is, we repeat this procedure until we get to element $D(1,1)$. If we keep track of the values and positions of the visited elements, we obtain the final path $W = w_l, w_{l-1}, \ldots, w_2, w_1$ in reverse order. The length of this path acts as a normalization factor, so the final dissimilarity value between two songs becomes:

$$d(HPCP_A, HPCP_B) = \frac{D(n,m)}{l} \tag{4.19}$$

Where l is the length of the alignment path.

4.3.2 Evaluation

We show the normalized *lift* curve for the implemented system (figure 4.5). In the plot, the DTW approach is compared with the previous implemented one (cross-correlation approach without any of the improvements of section 4.2.1). We also plot the results for the base-line experiment (section 3.5).

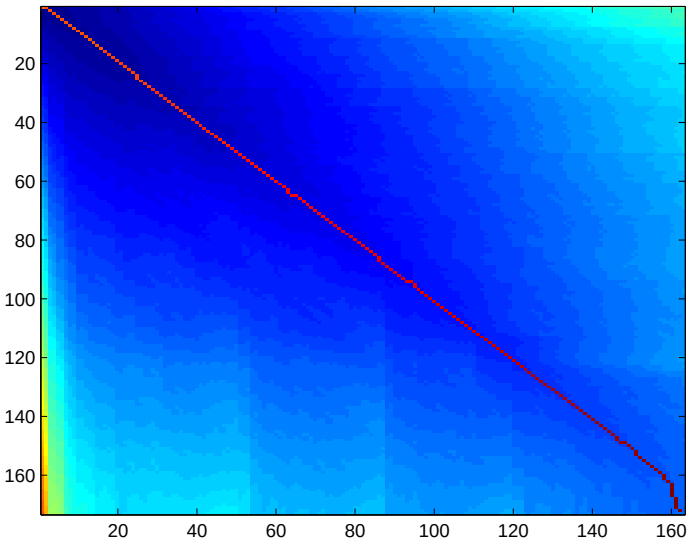

FIGURE 4.4 *DTW example of two versions of the song "Boys don't cry" performed by The Cure and by the group Happy Pills. We can see that these songs have slight different endings by looking at the end of the path found (burgundy line around position* $(160, 160)$*).*

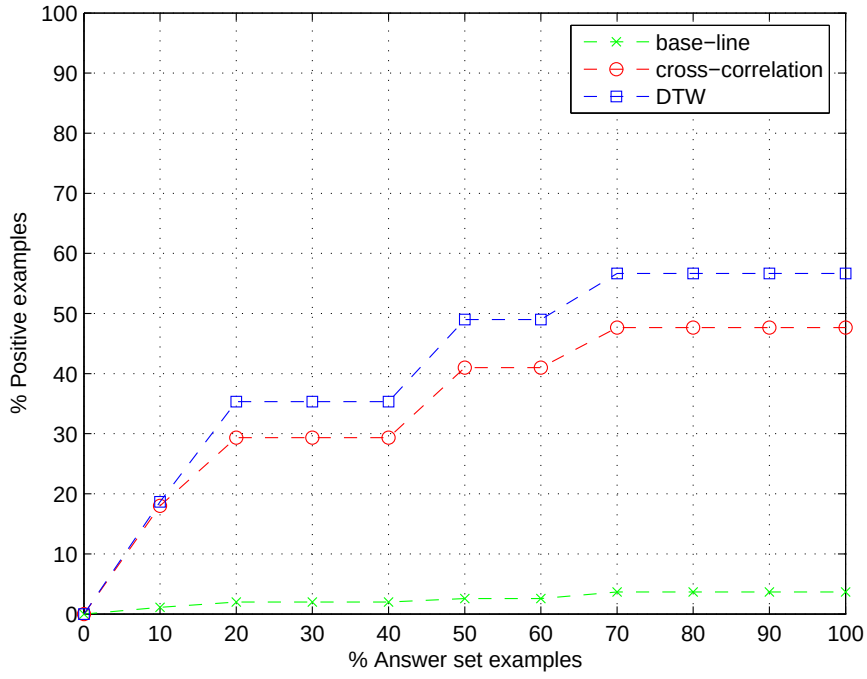

FIGURE 4.5 *Performance evaluation for the DTW approach with DB75. Normalized lift curves comparing 3 different experiments: base-line experiment (green line with crosses), cross-correlation approach (red line with circles), and DTW approach (blue line with squares).*

DTW approach seems to be slightly better than cross-correlation approach (this one without any improvements), and both are significantly better than the base-line experiment. In table 4.5 we summarize the results for DTW approach. In there, the same three evaluation measures that we are using throughout this thesis (section 3.4.3) are shown. In general, performance decreases as the database size increases. This is an expected effect that was also found in the base-line experiment (secion 3.5).

Evaluation measure	DB75	DB330
MNCI	0.567	0.385
F-measure	0.632	0.449
GM-bpref*	0.316	0.245

TABLE 4.5
DTW approach evaluation with DB75 and DB330. Three different evaluation measures are shown for two different databases.

4.4 Improvements to the DTW approach

When explaining the implementation method of the DTW approach in section 4.3.1, we have already used the characteristics that had been found to improve performance (section 4.2): 36-bin HPCPs, *OTI* key transposition, and HPCP correlation. We now focus on another particular aspect of the implementation of the DTW approach that we had not to take into account when improving the cross-correlation approach: the HPCP *averaging factor*. Also, we investigate the utility of a beat tracker for a DTW approach. In addition, as we have seen that we can apply several constraints to a DTW algorithm (section 2.2.1), some of these are tested.

4.4.1 Implementation

In section 2.2.1 we have seen that 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.

To test the effect of these constraints we have implemented 5 versions of our DTW algorithm: the DTW algorithm explained in previous section 4.3.1, two globally constrained DTW algorithms, and two locally constrained ones. We now comment them in detail.

Simple DTW This implementation corresponds to the one exposed in section 4.3.1, where no constraints are applied.

Globally constrained DTW Two implementations have been done. One corresponds to Sakoe-Chiba constraints [Sakoe 78] and the other one to the Itakura parallelogram [Itakura 75] (see section 2.2.1).

With Sakoe-Chiba global constraints, elements far from the diagonal of the $n \times m$ DTW matrix are not considered. A commonly used value for that in many speech recognition is 20% [Rabiner 93]. So, for implementing it, we just have to take a look at the values of i and j for the DTW recurrent relation and see if they fit the following condition:

$$min\{i \cdot s - p, m\} \leq j \leq min\{i \cdot s + p, 1\} \tag{4.20}$$

Where s is the slope defined by the lengths of the two songs being compared (i.e., $s = m/n$), and p is the percentage of far-from-the-diagonal elements that we want to consider (i.e., with 20%, $p = 0.2 \cdot m$).

Itakura global constraints operate in the same way as Sakoe-Chiba's, but we have different conditions for i and j that have to be fulfilled at the same time:

$$min\{(i - p_1) \cdot s, 1\} \leq j \leq min\{(i + p_1) \cdot s, m\} \tag{4.21}$$

$$min\{(i - p_2) \cdot s, 1\} \leq j \leq min\{(i + p_2) \cdot s, m\} \tag{4.22}$$

Where again, s is the slope defined by the lengths of the two songs being compared, and p_1 and p_2 are variable percentages depending on i and the parameter given (0.2 in our example):

$$p_1 = 0.2 \cdot (n - i) \tag{4.23}$$

$$p_2 = 0.2 \cdot i \tag{4.24}$$

Locally constrained DTW To further specify the optimal path, some local constraints must to be applied in order to guarantee that excessive compression or expansion of the time scales is avoided. In the tests performed we have specified two local constraints that were found to work in a plausible way with speech recognition [Myers 80a]. From this reference, *Type 1* and *Type 2* constraints are chosen (we will name them MyersT1 and MyertsT2 respectively).

We implement these two constraints by changing the recursive relation of equation 4.18. For MyersT1 it leads to:

$$D(i,j) = min \begin{cases} d(\overrightarrow{h_{A,i}}, \overrightarrow{h_{B,j}}) + D(i-1, j-1) \\ d(\overrightarrow{h_{A,i}}, \overrightarrow{h_{B,j}}) + d(\overrightarrow{h_{A,i-1}}, \overrightarrow{h_{B,j}}) + D(i-2, j-1) \\ d(\overrightarrow{h_{A,i}}, \overrightarrow{h_{B,j}}) + d(\overrightarrow{h_{A,i}}, \overrightarrow{h_{B,j-1}}) + D(i-1, j-2) \end{cases} \tag{4.25}$$

And for MyersT2:

$$D(i,j) = min \begin{cases} d(\overrightarrow{h_{A,i}}, \overrightarrow{h_{B,j}}) + D(i-1, j-1) \\ 2 \cdot d(\overrightarrow{h_{A,i}}, \overrightarrow{h_{B,j}}) + D(i-2, j-1) \\ 2 \cdot d(\overrightarrow{h_{A,i}}, \overrightarrow{h_{B,j}}) + D(i-1, j-2) \end{cases} \tag{4.26}$$

If we are in position (i, j), we are only paying attention to warpings $(i-1, j-1)$ (no tempo deviation), $(i-2, j-1)$ ($2x$ tempo deviation) and $(i-1, j-2)$ ($2x$ tempo deviation). So, with these local constraints, we allow a maximal tempo deviation of $2x$. This seems reasonable for us since, for instance, 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 the two types of constraints relies in the way we weight this warpings: considering intermediate distances for the former, and double-weighting $d(\overrightarrow{h_{A,i}}, \overrightarrow{h_{B,j}})$ for the latter. An intuitive drawing can be seen in figure 4.6.

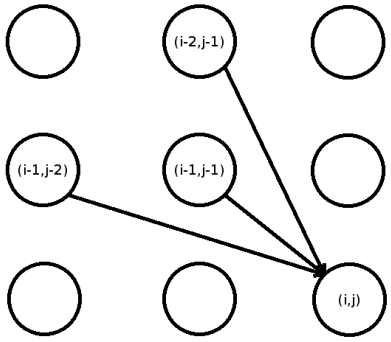

FIGURE 4.6 *Local DTW constraints example.*

Beat tracking and averaging factor In the cross-correlation approach, HPCP vectors were averaged beat-by-beat. With the DTW approach we have said that we did not need a beat tracker (and thus, a representation that was independent of tempo) because DTW was able to cope with tempo variations. We now demonstrate this affirmation while showing that a beat tracker can decrease performance in some situations.

If we do not use a beat tracking algorithm, the number of HPCP vectors to which we average becomes a critical parameter. In equation 4.14 we have blindly used a '20' value to group HPCPs (*averaging factor*). That is, 20 consecutive HPCP vectors are averaged. This corresponds to 0.93 seconds, which is an integration time that is musically sound for recognizing tonal harmonic content. To check if the forementioned choice was right, the *averaging factor* value has been implemented as an external parameter in the algorithm and we have tested several values for it while comparing performances with beat averaging.

4.4.2 Evaluation

Tables 4.6, 4.7 and 4.8 show the performances obtained with implemented variants mentioned in last subsection. Rows correspond to the different variants tested: a simple DTW algorithm, Sakoe-Chiba and Itakura global constraints, and MyersT1 and MyersT2 local constraints. The *Beat* column corresponds to the performance obtained using beat-by-beat averaging as in the first implemented approach in section 4.1.1. Other columns express performances depending on the *averaging factor* employed.

A first thing we can appreciate with these tables is that different evaluation measures have a different behaviour. For instance, the maximum performance reached with the MyersT2 variant is different for the *F-measure* (*averaging factor* of 5) than for the geometric mean of *bpref** (*averaging factor* of 25).

	Beat	5	10	15	20	25	30	40	50
Simple	0.493	0.460	0.546	0.557	0.567	**0.587**	0.580	0.540	0.537
Sakoe-Chiba	0.247	0.227	0.247	0.293	0.290	0.307	0.320	0.293	**0.323**
Itakura	0.240	0.217	0.260	0.327	0.313	0.323	0.353	0.337	**0.367**
MyersT1	0.560	0.593	**0.600**	0.593	0.593	0.573	0.573	0.523	0.520
MyersT2	0.543	**0.600**	0.590	0.557	0.553	0.543	0.510	0.477	0.487

TABLE 4.6

Improved DTW approach evaluation with DB75. Mean number of covers identified (MNCI). Columns correspond to different averaging factors tested.

	Beat	5	10	15	20	25	30	40	50
Simple	0.548	0.537	0.606	0.611	0.632	**0.638**	0.634	0.598	0.599
Sakoe-Chiba	0.289	0.259	0.282	0.327	0.332	0.342	0.355	0.331	**0.370**
Itakura	0.275	0.256	0.286	0.362	0.353	0.360	0.395	0.388	**0.417**
MyersT1	0.624	0.647	**0.651**	0.641	0.643	0.624	0.625	0.577	0.582
MyersT2	0.611	**0.651**	0.646	0.617	0.614	0.599	0.566	0.542	0.540

TABLE 4.7

Improved DTW approach evaluation with DB75. F-measure value. Columns correspond to different averaging factors tested.

	Beat	5	10	15	20	25	30	40	50
Simple	0.209	0.202	0.248	0.261	**0.316**	0.312	0.311	0.254	0.278
Sakoe-Chiba	0.066	0.048	0.060	0.084	0.085	0.090	0.101	0.091	**0.116**
Itakura	0.066	0.049	0.059	0.100	0.089	0.097	0.115	0.112	**0.134**
MyersT1	0.314	0.325	**0.327**	0.302	0.323	0.297	0.306	0.238	0.264
MyersT2	0.295	**0.331**	0.323	0.297	0.299	0.269	0.241	0.234	0.240

TABLE 4.8

Improved DTW approach evaluation with DB75. GM-bpref value. Columns correspond to different averaging factors tested.*

This is due to the fact that different evaluation measures reflect different things regarding effectiveness of the system. As we have argued in section 3.1, the measure that best reflects our interests (regarding ranking and recall of the answers) is *bpref**.

A second fact that can be stated is that we can reach better performances with averaging HPCPs with DTW, than just using beat-by-beat averaging. Values such as the ones of column 25 for the simple DTW approach, column 10 for MyersT1 DTW approach, or, for instance, column 30 for Itakura DTW state that.

We now elaborate some comments on figure 4.7. In it, we plot the same results that have been shown in table 4.8. Concretely, we have plotted the performance (measured with the geometric mean of *bpref**) of several methods: unconstrained DTW (dark blue line with circles), DTW with global constraints (triangles, light blue for Itakura and purple for Sakoe-Chiba) and DTW with local constraints (squares, red for MyersT1 and green for MyersT2)) versus the HPCP features framelength. We also denote the performance of these methods using the beat information into the respective curves (labels and arrows behind them).

With a simple DTW algorithm with no constraints (dark blue circles) we observe that there is peak around an *averaging factor* of 20 (near 1 sec), with performance decreasing on the left (as framelength decreases) and on the right (as framelength increases). This stability region might range from 17 to 32.

It can be seen that the performance decreases in all tested methods as groups of frames are larger (as framelength increases). This is due to the excessive averaging of frames, leading to a 'limit' situation where there would be only one frame for all the song. This globally averaged HPCP vector would be the key (tonality) profile. As we have previously transposed one song to the key of the other one with equation 4.9, both tonality profiles would be quite similar (all of them 'tuned' to the reference), and this would happen for each pair of songs being compared (so the similarity assessment would be more or less the same for each pair). As a consequence, in this 'limit' situation, we would reach the performance of a random method (*GM-bpref**=0.015).

We can feel that the performance of constrained methods keeps increasing until the end of the

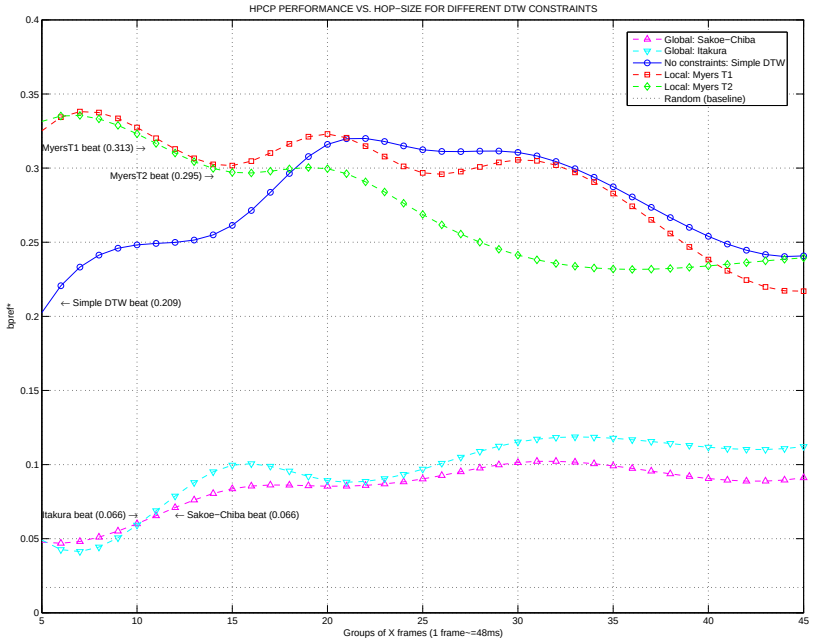

FIGURE 4.7 *Performance evaluation for the improved DTW approach with DB75. Interpolation of normalized lift curves comparing 4 differently constrained DTW algorithms with a simple one.*

x-axis of the graph, but this makes no sense, and, after this point, bpref* is expected to decrease for the mentioned reasons (previous paragraph). We corroborated this assumption by studying some further tests. These resulted in very low performance values (i.e., *GM-bpref**=0.201 for Itakura with an *averaging factor* of 75, and *GM-bpref**=0.215 for Sakoe-Chiba with the same factor value).

In general (except for the locally constrained methods), as the framelength decreases, it can be seen that performance 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 in the DTW matrix indicates several points of one sequence aligned just to one point of the other, left picture in figure 4.8). This makes the path length to increase, and since we are normalizing the final result by this value, the distance decreases. Then, false positives are introduced in the final answers of the algorithm. Figure 4.8 shows the matrices obtained after a simple DTW and locally constrained DTW approaches.

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 performance of locally constrained methods keeps increasing.

Finally, going back to the initial plot of figure 4.7, we observe that performance for globally constrained methods is significantly lower than others. This is due to the fact that by the use of these global constraints, we restrict the paths to be in the center of the DTW matrix. To understand that, as an example, we can consider a song composed of two parts that are the same ($S_1 = AA$) and another song with nearly half the tempo ($'$) and composed of only one of these parts ($S_2 = A'$). Next plots in figures 4.9 graphically explain this idea.

The first plot has no constraints. We can see that the best path (straight diagonal red line) goes

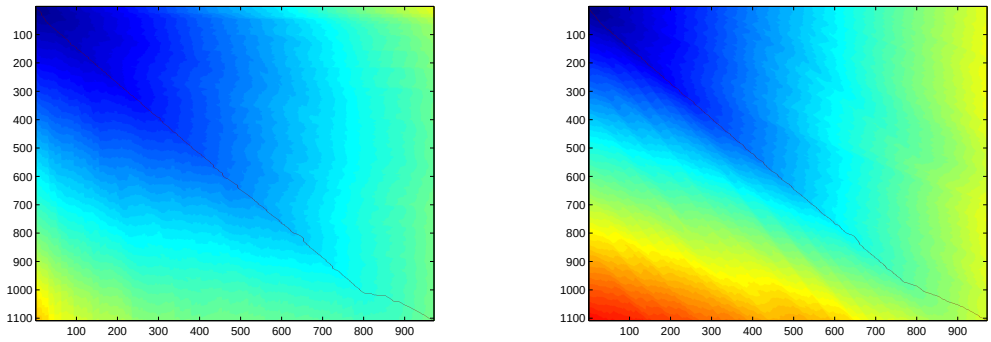

FIGURE 4.8 *Matrices obtained for a simple (left) and locally constrained (right) DTW approach (MyersT1). On the former we can observe some 'pathological' warpings, specially at points around* $(1000, 850)$. *On the latter, the same two songs are being compared. We can observe that 'pathological' warpings have disappeared.*

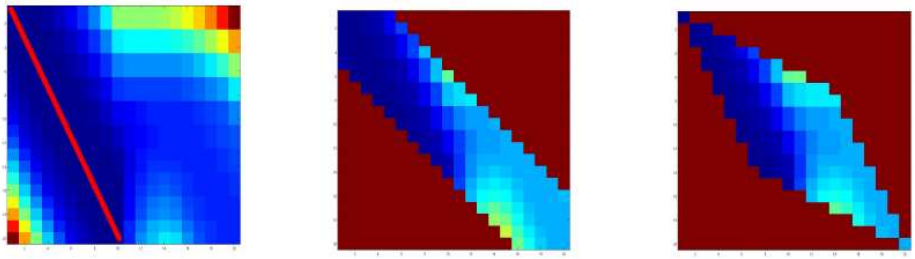

FIGURE 4.9 *Examples of an unconstrained DTW matrix (i), and Sakoe-Chiba (ii) and Itakura (iii) global constraints for* S_1 *(x-axis) and* S_2 *(y-axis).*

from $(1, 1)$ to more or less $(20, 10)$ (x-axis lower-half part). This is logical since one song has two parts (the same) and the other one just one part with half the tempo. The second plot corresponds to the same matrix with Sakoe-Chiba constraints. We can see that the 'optimal' path we could trace with the first plot has been broken by the effect of the global constraints. A same situation occurs with the third plot (Itakura constraints).

4.5 Discussion

In this chapter we have examined two state-of-the-art methods for cover song identification and several variants of them. We comment now some relevant aspects that have arisen within these experiments.

In section 4.1 we have presented a cross-correlation-based approach. This extracts a beat-synchronous chroma representation of the song and computes cross-correlation between pairs of songs. To account for differences in key, one song was transposed to the tonality of the other one by means of computing a global HPCP for each song (equations 4.5 and 4.6) and an *Optimal Transposition Index* (*OTI*, equation 4.7). This technique has been proven to be more accurate than transposing the song to a reference key by means of an algorithm for key estimation (section 4.2.2). This corroborates the statement we made in the *Scientific Background* chapter when discussing state-of-the-art methods related with cover song identification (section 2.6, 'intermediate' techniques that may be applied to achieve this task). In this case, the use of a not completely reliable key extraction algorithm lowers the overall performance.

In section 4.2.2 we have stated the importance of the metric used to compare chroma or HPCP vectors. Furthermore, we have shown that using a distance that better correlates with cognitive foundations of musical pitch can improve substantially the final system performance. As it was also discussed in section 2.6, an euclidean-based distance for HPCP similarity calculations can worsen performance. This is an important issue that will be dealt in detail in next chapter.

In the same section as well, we have shown that HPCP resolution is important both with cosine and correlation distances. We have tested 12, 24, and 36-bin HPCPs, and the results suggest that performance increases as the resolution does so. We have not tested a higher HPCP resolution, but this issue is partially addressed in next chapter, where a new distance measure between feature vectors that is quite robust to different HPCP resolutions (see section 5.1.2) is presented.

With the next implemented approach (DTW, section 4.3) we have seen that we can have similar performances to the cross-correlation method without taking the beat information into account. Furthermore, in the improved variants of the DTW approach (section 4.4) we have stated that this approach could lead to better performance results without such information. Tables 4.6, 4.7 and 4.8, figure 4.7 and the subsequent discussion have assessed that. This is another fact that makes us mistrust about the use of 'intermediate' processes such as key extraction algorithms or beat tracking systems (citing the two that have been tested in this thesis). As mentioned in section 2.6, errors in these 'intermediate' processes might be added and propagated to the final performance of the overall system.

DTW allows us to restrict the alignment (or 'warping') paths to our requirements (section 2.2.1). Consequently, we have tested four 'standard' constraints on these paths (two local and two global constraints) in section 4.4. With global constraints we are not considering paths (or alignments) that might be far from the main diagonal of the DTW matrix. A problem arises when this path can represent a 'correct' alignment (remember the example shown on figure 4.9). We also have seen that the performance decreases substantially with these constraints. As we have argued in section 1.4.4, song structure is often changed within covers. When this happens, the 'correct' alignment between two covers of the same *canonical song* may be outside of the diagonal. Therefore, the use of global constraints will dramatically decrease the performance of the system. In addition, these two facts make us think about the correctness of using a global alignment technique for cover song identification. In next chapter's discussion (section 5.3) when we have implemented and evaluated a local alignment-based approach, this aspect will be more highlighted. Regarding local constraints, we have seen that these can help us by reducing 'pathological' warpings that arise when using a small *averaging factor*, and, consequently allowing us to use much detail in our analysis, and, therefore, getting a better performance.

As a corollary, a comparative figure between improved methods explained in this chapter is shown. Also, a table comparing performance of implemented approaches and the ones presented in MIREX 2006 is given.

Firstly, in table 4.9, we show a summary of the performance results with the three measures used along this dissertation. Note that the DTW and the IDTW algorithms already incorporate the improvements of ICC (36-bin HPCP, HPCP correlation distance, etc.).

A normalized *lift* curve comparing improved versions of cross-correlation and DTW approaches (sections 4.2 and 4.4 respectively) is shown in figure 4.10. In there, we plot the normalized *lift* curve for the improved DTW approach (blue squares) using 36-bin HPCPs, HPCP correlation distance, transposing to the same key with a global HPCP (equation 4.16), MyersT1 local constraints and an HPCP *averaging factor* of 10. The values for the cross-correlation approach (red circles) are obtained with 36-bin HPCPs, HPCP correlation distance and global HPCP computation.

A plot like the one in figure 4.10 and the results shown along this chapter makes us realize that we

Evaluation measure	CC	ICC	DTW	IDTW
MNCI	0.477	0.573	0.567	0.600
F-measure	0.539	0.638	0.632	0.651
GM-bpref*	0.226	0.381	0.316	0.327

TABLE 4.9
Performance comparison for the experiments done (DB75). Cross-correlation approach (CC, section 4.1) and improved cross-correlation approach with correlation distance 36-bin HPCP and OTI transposition (ICC, section 4.2) are compared against a DTW approach (section 4.3) and an improved DTW algorithm with local constraints (MyersT1) and an averaging factor of 10 (section 4.4).

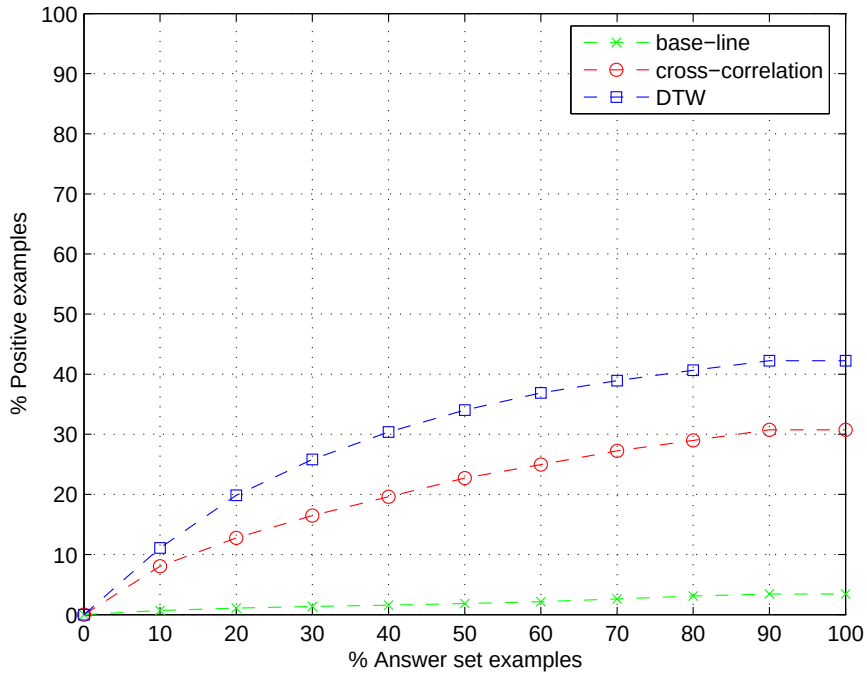

FIGURE 4.10 *Comparison between the two improved implemented methods. The base-line experiment is also shown (green crosses).*

can slightly outperform a simple cross-correlation method with a locally constrained DTW approach. However, an overall accuracy of 30-40% of correctly classified covers among the 10 firstly retrieved ones is still a poor result. We will see how do we improve it in next chapter.

Finally, in order to see if these results are far away of existing state-of-the-art methods, we also show a table which compares performances obtained with the two improved methods described in this chapter and the ones evaluated in the MIREX 2006 contest (table 4.10). Although we are not using the same databases (these results correspond to tests done with *DB330*), we can see that the values for used evaluation measures are quite equivalent.

We observe that results for *DE* and *ICC* columns are comparable. As we have commented in section 4.1, the cross-correlation approach has been inspired in [Ellis 06, Ellis 07]. Therefore, taking into account the improvements introduced in section 4.2, which might explain a slight better performance, we can consider MIREX 2006 song database and our music collection (*DB*330) to have some resemblance. Taking this into account, we can hypothesize that ICC and IDTW methods perform slightly better

	Range	CS	DE	KL1	KL2	ICC	IDTW
TNCI	[0..3300]	211	761	365	314	949	1306
MNCI	[0..1]	0.064	0.231	0.111	0.095	0.296	0.408
MM	[0..1]	0.213	0.453	0.250	0.227	0.527	0.633
MRR(1)	[0..1]	0.21	0.49	0.22	0.22	0.62	0.79

TABLE 4.10

Cross-database method comparison. The evaluation measures used in MIREX 2006 are shown. CS corresponds to the method explained in [Sailer 06], DE corresponds to [Ellis 06], and KL1 and KL2 to [Lee 06b]. In the two rightmost columns we show performances for the improved cross-correlation approach (ICC, section 4.2) and for the improved DTW approach (IDTW, section 4.4).

than considered state-of-the-art methods (CS, DE, KL1, KL2). Concretely, comparing DE and ICC, the improvement is around 25% in the total number of covers identified, and comparing DE and IDTW, it reaches more than 70% with the same evaluation measure.

Chapter 5

Local alignment method for tonal sequence similarity

Previous experiments with state-of-the-art methods and its proposed variants have served us in two ways. First, in the sense that now we have some methods to be compared, given our music collection. Furthermore, they are improved versions of existing methods, so we know where we are in terms of performance. Second, with implemented variants, we have been able to see some improvements that can be directly applied to a system developed entirely from scratch.

In next sections we extensively explain the implementation of a new cover song identification system (section 5.1) and its evaluation (section 5.2). In this chapter, the evaluation employs the same measures used in previous ones, but we also dedicate a subsection to a more 'musicological' evaluation (more focused on intuitive aspects of music, and trying to see the system from the user's perspective). Finally, section 5.3 is devoted to discuss all important aspects assessed by this method.

5.1 Implementation

In this section we present a tonal sequence local alignment method that can be applied to a cover song identification task. In next subsections we explain it in detail, but firstly, we give a brief overview of the system with a general block diagram and some comments on it. Details of the implementation are omitted in the first subsection. We explain them in subsection 5.1.2.

5.1.1 General system architecture

Figure 5.1 shows a functional block overview of the method. Rectangles indicate a module where some processing is done and parallelograms correspond to initial, intermediate or final data elements (or structures). The system comprises an *HPCP descriptors extraction* module (B1), an *HPCP matrix post-processing* module (B2), an *HPCP averaging* module (B3), a *transposition* module (B4), a *similarity matrix creation* module (B5), a *Dynamic Programming local alignment* module (B6), an *alignment analysis* module (B7), and a *score normalization* module (B8).

When comparing two songs, we first extract HPCP feature vectors for each one of them (module B1). With this, we obtain an HPCP matrix for each song, where columns represent HPCP vectors, and whose length (the number of columns) is proportional to the length of the audio file input. From this matrix, two parallel processes are applied. These processes are done for the two songs being compared in the same manner (and independently) in order to obtain the same kind of representation.

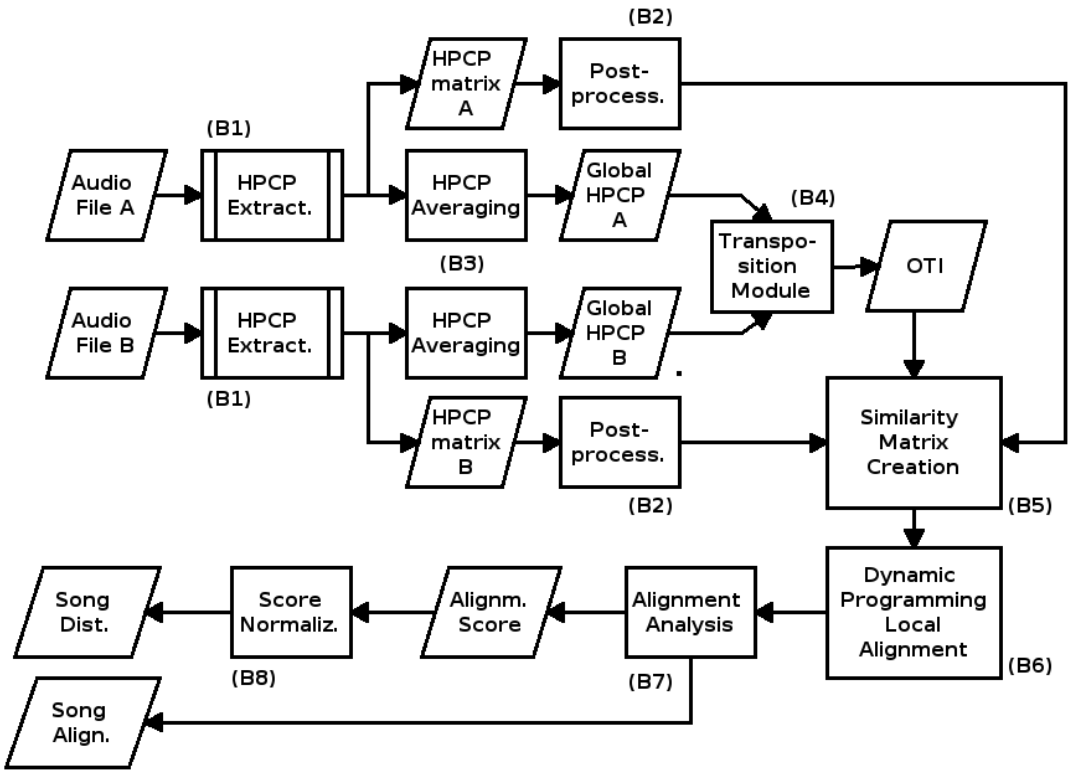

FIGURE 5.1 *General block diagram for the local alignment method proposed. Modules correspond to squares, and data elements correspond to parallelograms.*

Processes correspond to:

- HPCP matrix post-processing (module B2): We post-process the HPCP matrix for future calculations (frame averaging, normalization, etc.). The result from these operations (we will call it *refined HPCP matrix*) is used in the *similarity matrix creation* module (B5).
- Global HPCP calculation (module B3): we calculate a global HPCP, which corresponds to the average feature vector over all the matrix columns (audio frames). The global HPCP represents the main tonality (or key) profile of the song. These data elements are the only input for the *transposition* module (B4).

Then, information of the two songs being compared is combined. In module B4, we calculate an *Optimal Transposition Index* (*OTI*, previously defined in section 4.1.1) using global HPCPs obtained from both songs. This index will be used in the *similarity matrix creation* module (B5) to transpose one *refined HPCP matrix* into the same tonality as the other. So, module B5 has three inputs, two refined HPCP matrices, and an integer number representing the optimal transposition to be applied. The output of module B5 is a similarity matrix, where each element (i, j) represents the degree of resemblance between column i of HPCP matrix A and column j of HPCP matrix B.

A similarity matrix is the only necessary input for our Dynamic Programming local alignment algorithm (from now on, DPLA). This produces a matrix of local alignments that is then used to determine similarity between covers.

Through the *alignment analysis* module (B7), we obtain a score of the local similarity between the

two songs. This score is finally normalized in order not to introduce any bias with different song lengths in the *score normalization* module (B8).

5.1.2 Detailed description

We divide the description of the algorithm into 4 parts: pre-processing, similarity matrix creation, Dynamic Programming local alignment, and post-processing. These are intentionally chosen to be in some sense general, in order to accommodate any description of a system of this kind. In terms of figure 5.1, pre-processing comprises *HPCP descriptors extraction*, *HPCP matrix post-processing*, *HPCP averaging*, and *song transposition* (modules B1, B2, B3 and B4). The next two parts (similarity matrix creation and Dynamic Programming local alignment) are devoted to modules B5 and B6 respectively. Finally, post-processing details (modules B7 and B8) are addressed.

Pre-processing

As we have seen, in the early stages of the whole procedure each song is processed independently until we obtain a *refined HPCP matrix* and a global HPCP vector for each one. This part of the method is what we consider to be a pre-processing step, and it comprises mainly an HPCP extraction process, and some further processing issues.

To extract HPCP feature vectors, we first 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 frame overlapping of 50%. We consider a whitened frequency spectrum that ranges from 40 Hz up to 5 KHz, 8 harmonics, and a maximum of 10000 spectral peaks, that are summarized in a 36-bin octave-independent histogram. The HPCP extraction procedure employed here is the same that has been used in all the experiments of chapter 4 and in several approaches in the literature [Gómez 04, Gómez 06b, Gómez 06c]. For more details on the HPCP extraction process we refer to section 2.1.1, where they have been properly explained, or to [Gómez 06a].

From this frame-by-frame feature extraction process we end up with a sequence of HPCP vectors that describe the tonal evolution of the audio signal. Within the extraction process, each HPCP vector $\overrightarrow{u_i}$ is normalized as:

$$\overrightarrow{h_i} = \frac{\overrightarrow{u_i}}{max\{\overrightarrow{u_i}\}} \tag{5.1}$$

Where $\overrightarrow{u_i}$ represents an unnormalized HPCP vector, and $max(\overrightarrow{u_i})$ corresponds to the maximum value of vector $\overrightarrow{u_i}$. So, as there were no negative values of $\overrightarrow{u_i}$, each component is finally bounded between 0 and 1.

This sequence of HPCP vectors can be represented as a matrix, where each column represents an HPCP vector extracted for one frame. That is:

$$HPCP = [\overrightarrow{h_1}, \overrightarrow{h_2}, ..., \overrightarrow{h_l}] \tag{5.2}$$

As our HPCP resolution is $1/3$ of a semitone, this is going to be a $36 \times l$ matrix, where 36 (rows) is the number of HPCP bins considered. In previous experiments we have proven this resolution to work better than 1 or $1/2$ semitones (see section 4.1.2). The number of columns l would correspond to the length of the song in frames (the number of HPCP vectors extracted).

What we do next (still in the *HPCP matrix post-processing* module), is to remove all silent or highly inharmonic frames by pruning out vectors from the HPCP matrix to obtain a *refined HPCP matrix*.

These silent or highly inharmonic frames are simply detected by looking at very low variances of vectors $\overrightarrow{h_i}$. Note that since our method will perform a local alignment, we do not care about silences at beginnings, ends, or in the middle of songs.

In addition, consecutive vectors are averaged by summing matrix consecutive columns and dividing by the maximum value obtained (as done in section 4.3.1). New averaged HPCP vectors are calculated as:

$$\overrightarrow{h'_i} = \frac{\sum_{j=k}^{k+X} \overrightarrow{h_i}}{max\{\sum_{j=k}^{k+X} \overrightarrow{h_i}\}} \tag{5.3}$$

Where we divide by the maximum component of the resultant vector to get a normalized HPCP whose values are between 0 and 1. k corresponds to the number (or index) of the desired HPCP vector:

$$k = (i-1) \cdot X \tag{5.4}$$

If we choose larger groups, time latency of subsequent processes improves (as the number of frames or vectors to process decreases), but the accuracy of the method becomes poorer. In section 4.4.2, we have called X as the *averaging factor*, and we have seen that good choices for X, were $X = 5$ or $X = 10$, which results in a framelength near 0.25 and 0.5 seconds respectively. An empirical justification for these choices has been given in the same section.

So, with the mentioned two pre-processing steps, the sequence of tonal descriptors becomes a slightly different one. What we get is:

$$HPCP' = [\overrightarrow{h'_1}, \overrightarrow{h'_2}, \dots, \overrightarrow{h'_n}] \tag{5.5}$$

Where n will correspond the integer value closer to I/X.

In parallel with this *refined HPCP matrix* computation, a global HPCP vector is obtained in order to represent the main tonality (or key) profile for each song. This is simply done by averaging all HPCP vectors found in an HPCP matrix (as we have explained in section 4.3.1). The global HPCP vector is also normalized between 0 and 1. Therefore:

$$\overrightarrow{GlobalHPCP} = \frac{\sum_{i=1}^{I} \overrightarrow{h_i}}{max\{\sum_{i=1}^{I} \overrightarrow{h_i}\}} \tag{5.6}$$

Where I is the total number of HPCP vectors, and $max()$ gets the maximum value of the vector addition inside it (as in equations 5.1 and 5.3).

So, when we end the pre-processing of songs A and B (after modules B2 and B3), we get one refined HPCP matrix $HPCP'_A = [\overrightarrow{h'_{A,1}}, \overrightarrow{h'_{A,2}}, \dots, \overrightarrow{h'_{A,n}}]$ and a global HPCP vector $\overrightarrow{GlobalHPCP_A}$ for song A, and one refined HPCP matrix $HPCP'_B = [\overrightarrow{h'_{B,1}}, \overrightarrow{h'_{B,2}}, \dots, \overrightarrow{h'_{B,m}}]$, and a global HPCP vector $\overrightarrow{GlobalHPCP_B}$ for song B. These post-processed HPCP matrices look more or less like the ones shown in figure 4.3 (subsection 4.3.1).

To obtain an *Optimal Transposition Index* (OTI) between two songs, we proceed as in section 4.1.1 (equation 4.7):

$$OTI_{A,B} = argmax_{0 \le id \le N_B - 1}\{\overrightarrow{GlobalHPCP_A} \cdot circularshift(\overrightarrow{GlobalHPCP_B}, id)\} \tag{5.7}$$

Where '$\cdot$' indicates a dot product, N_B is the number of bins of the feature vector considered (in this case 36), and $circularshift(\overrightarrow{h}, id)$ is a function that rotates a vector (h) id positions to the right.

A similar formula has been used in equation 4.7, where we wanted to calculate an *OTI* for transposing the two songs being compared to a common key or tonality (section 4.1.1).

Similarity matrix creation

This part explains in detail module B5. Its inputs are, for each pair of songs being compared, only two refined HPCP matrices and one *Optimal Transposition Index* (*OTI*).

In the same manner as in the experiments in chapter 4, we now transpose one song (say song B) to the tonality of the other (song A). For doing so, we shift in just one of them all the bins of each HPCP vector by $OTI_{A,B}$. That is, for each vector $\overrightarrow{h'_{B,i}}$ of $HPCP'_B = [\overrightarrow{h'_{B,1}}, \overrightarrow{h'_{B,2}}, \dots, \overrightarrow{h'_{B,m}}]$ we calculate:

$$\overrightarrow{h''_{B,i}} = circularshift(\overrightarrow{h'_{B,i}}, OTI_{A,B}) \quad \forall\ 1 \leq i \leq m \tag{5.8}$$

We therefore obtain a transposed matrix $HPCP''_B = [\overrightarrow{h''_{B,1}}, \overrightarrow{h''_{B,2}}, \dots, \overrightarrow{h''_{B,m}}]$ for song B, and the one that we had for song A ($HPCP'_A = [\overrightarrow{h'_{A,1}}, \overrightarrow{h'_{A,2}}, \dots, \overrightarrow{h'_{A,n}}]$).

With $HPCP'_A$ and $HPCP''_B$ we are now ready to create a similarity matrix S. The values of each cell (i, j) of this matrix have the functionality of a local similarity measure between HPCP vectors $\overrightarrow{h'_{A,i}}$ and $\overrightarrow{h''_{B,j}}$. A cost function like this has been introduced in all the methods in section 4, where a cosine distance (for experiments in sections 4.1 and 4.2) and a simple correlation (for experiments in sections 4.3 and 4.4) were used.

As we have said in section 2.6, chroma feature vectors do not belong to an euclidean space, so, similarity judgements between them shouldn't be done with an euclidean-based distance (like the cosine distance). In addition, in section 4.2.2 we have seen that the function used to assess (dis)similarity between chroma or HPCP features is of crucial importance, with direct implications to the final performance of the system. In the same section we have also seen that correlation between HPCPs is a better (dis)similarity measurement that leads to improved results, but we still feel that this might be not the correct measure to use. Furthermore, chroma vectors distance and, more generally, tonal similarity, is a still far to be understood topic, with many of perceptual and cognitive issues that require lots of research.

So, we are faced to the problem of giving a similarity measure between two HPCPs or chroma feature vectors. We address this issue by considering only if an HPCP vector is similar or not to another one (binary similarity). We believe that this might be an easier (or at least more affordable) task to assess, than obtaining a reliable graded scale of resemblance between two HPCPs that correlates with (sometimes subjective) perceptions of similarity. In addition, considering binary similarity can have some advantages for further processing. For instance, the resultant similarity matrix gets very contrasted, leading us with a very clear notion of where the two sequences agree or not (intuitively we can think of an image where the different gray scales are transformed into black or white with an appropriate threshold that preserves what we want to see). Also, binary similarity allows us to operate such as many string alignment techniques do: just considering if two elements of the string are the same or not. With this, the range of techniques to be applied to an alignment of tonal sequences gets expanded with many methods borrowed from string comparison, DNA or protein sequence alignment, symbolic time series similarity, etc. (we have summarized some of them in section 2.2).

An intuitive idea to consider when deciding if two HPCP vectors refer to the same tonal root, is to keep circularly shifting one of them and calculate a resemblance index for all possible transpositions. Then, if the transposition that leads to maximal similarity corresponds to zero semitones, the two HPCP vectors are claimed to be the same. This idea can be formulated in terms of the *Optimal Transposition*

Index (OTI) explained before:

$$OTI(\overrightarrow{h'_{A,i}}, \overrightarrow{h''_{B,j}}) = argmax_{0 \leq id \leq N_B - 1}\{\overrightarrow{h'_{A,i}} \cdot circularshift(\overrightarrow{h''_{B,j}}, id)\} \tag{5.9}$$

Where $\overrightarrow{h'_{A,i}}$ and $\overrightarrow{h''_{B,j}}$ denote two HPCP feature vectors, and N_B is the number of bins of them.

After calculating $OTI(\overrightarrow{h'_{A,i}}, \overrightarrow{h''_{B,j}})$, the binary similarity measure between the two vectors is then obtained by:

$$s(\overrightarrow{h_i}, \overrightarrow{h_j}) = \begin{cases} \mu_+ & \text{if } OTI(\overrightarrow{h'_{A,i}}, \overrightarrow{h''_{B,j}}) = 0, \\ \mu_- & \text{otherwise.} \end{cases} \tag{5.10}$$

Where μ_+ and μ_- are two constants that indicate match or mismatch. These are usually set to 0 and 1, or to a positive and a negative value depending on the final application of them (see section 2.2.2). In next subsection we show the values taken in our experiments with a local alignment technique.

In our case, as we are using 36-bin HPCPs, equation 5.10 is extended to allow small transposition changes of 1/3 of a semitone:

$$s(\overrightarrow{h_i}, \overrightarrow{h_j}) = \begin{cases} \mu_+ & \text{if } OTI(\overrightarrow{h'_{A,i}}, \overrightarrow{h''_{B,j}}) = 0,1, \text{ or } N_B - 1, \\ \mu_- & \text{otherwise.} \end{cases} \tag{5.11}$$

So, after processing two HPCP sequences of lengths n and m, we end up with a $n \times m$ similarity matrix S whose (i-th, j-th) element corresponds to $s(\overrightarrow{h_i}, \overrightarrow{h_j})$. Two example regions of this similarity matrix are shown in figure 5.2.

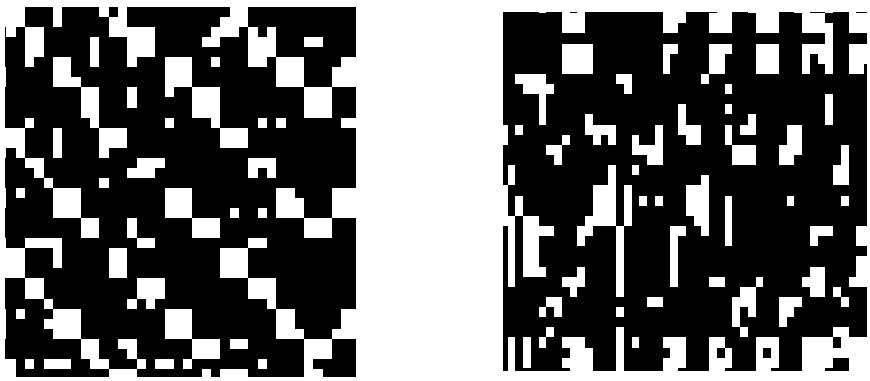

FIGURE 5.2 *Binary similarity matrices. Examples of comparing two covers of the same song (left) and two songs that not belong to the same cover set (right). We can see diagonal white lines in the former, while this pattern does not exist in the latter.*

We now show a new expression for a faster calculation of OTI. For that, it is necessary to note that the part inside *argmax* in the previous equation is a circular convolution. To see it, we use the notation of equation 4.8 in section 4.1.1. Then, a circular shift can be expressed as:

$$circularshift(h[x], id) = h[((x + id))_{N_B}] \tag{5.12}$$

Where N_B is the number of components of the vector, and $((x + id))_{N_B}$ is the modulo N_B of $x + id$.

Changing notation, $OTI_{i,j}$ becomes:

$$OTI(\overrightarrow{h'_{A,i}}, \overrightarrow{h''_{B,j}}) = argmax_{0 \leq id \leq N_B - 1}\Big\{ \sum_{n=1}^{N_B} h_{A,i}[x] \cdot h_{B,j}[((x + id))_{N_B}]\Big\} \tag{5.13}$$

Equation 5.9 is expensive to compute ($O(2N_BN_B)$ operations, being N_B the number of components of a vector). Alternatively, the Fast Fourier Transform (FFT) can be used, obtaining a computationally less expensive formula[1]. As the part inside *argmax* in the previous equation 5.13 is a circular convolution, it can be proved that [Oppenheim 99]:

$$\sum_{n=1}^{N_B} h_{A,i}[x] \cdot h_{B,j}[((x + id))_{N_B}] \propto FFT\{FFT\{h_{A,i}[x]\} \cdot FFT\{h_{B,j}[x]\}^*\} \tag{5.14}$$

Where N_B is the number of components of the vector, and $((x+id))_{N_B}$ is the modulo N_B of $x+id$. FFT is the Fast Fourier Transform and '*' indicates the complex conjugate. The value of $OTI(\overrightarrow{h'_{A,i}}, \overrightarrow{h''_{B,j}})$ can be obtained by the argument that leads to a maximum value of the result of both expressions in equation 5.14 (proportionality), while the latter is faster to calculate due to the speed of the FFT algorithm ($O(N_B log(N_B))$ operations). This formula is interesting because we have to calculate a huge amount of OTI expressions in our algorithm, so, the computational time gets significantly reduced.

Dynamic Programming local alignment

This part of the explanation is devoted to module B6 (*Dynamic Programming local alignment* block in figure 5.1). This module only takes one input: the similarity matrix S calculated before.

We have previously argued about the utility to perform a local alignment between sequences in the case of cover song identification in order to overcome changes in the structure of songs belonging to the same *cover set* (section 2.6). We have also stated that global constraints forcing warping paths to be around the alignment matrix diagonal had a detrimental effect in system performance (sections 4.4.2 and 4.5). Furthermore, in the same sections, we have seen that a Dynamic Programming algorithm (such as DTW) with local constraints was a powerful tool to deal with tempo variations. Now is time to add both things up: local alignment and Dynamic Programming.

For doing so, we started with a well known Dynamic Programming local alignment technique that has been properly reviewed in section 2.2: the Smith-Waterman algorithm [Waterman 76, Smith 81]. This algorithm creates a local alignment $(n+1) \times (m+1)$ matrix H based on a binary measure of local resemblance between items of a sequence. It starts by initializing the first row and column:

$$H(i, 0) = H(0, j) = 0 \tag{5.15}$$

For $0 \leq i \leq n$ and $0 \leq j \leq m$. Then, values for H are obtained from the relationship:

$$H(i,j) = max \begin{cases} H(i-1, j-1) + s(\overrightarrow{h_i}, \overrightarrow{h_j}) \\ H(i-1, j) - \delta \\ H(i, j-1) - \delta \\ 0 \end{cases} \tag{5.16}$$

For $1 \leq i \leq n$ and $1 \leq j \leq m$. In the expression, $s(\overrightarrow{h_i}, \overrightarrow{h_j}) = \{\mu_+, \mu_-\}$ acts as a local cost function.

[1] Thanks to César Alonso and Jordi Bonada for their ideas and help.

For the comparison of DNA sequences, the score for a match or identity is kept at 1 ($\mu_+ = 1$), and the penalty for a mismatch is set to zero or (preferably) a negative value ($\mu_- \leq 0$). δ is usually a linear function that assigns an initial penalty for a gap opening and extension gap penalty for each deleted or inserted sequence item increasing the gap length. We have given an extensive explanation of the Smith-Waterman algorithm in section 2.2.

In subsection 4.4.2 we have seen that certain local constraints [Myers 80a] can help us in reducing the number of alignments considered by just admitting certain relative tempo ranges. These prevent 'pathological' warpings of the signal as well (see also section 2.2.1). In order to introduce this beneficial aspects into our algorithm we have modified equation 5.16 in some ways:

1. We want just to consider 3 possible paths (the ones defined by the local constraints). In this sense we use the ones named MyersT2 that we had already employed section 4.4.1. An intuitive schema is shown in figure 5.3.

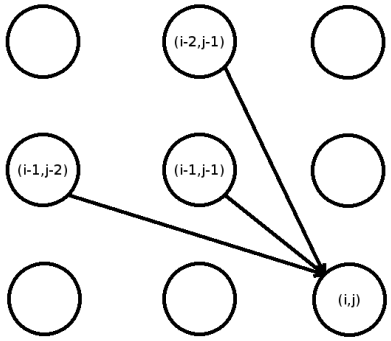

FIGURE 5.3 *Local constraints example.*

   Remember that with the Smith-Waterman algorithm we were just considering contiguous matchings. This is due to the fact that, in $H(i,j)$, we just look at $H(i-1,j-1)$ for a genuine match (first term of equation 5.16). The other terms are just for paths with insertions and deletions (see *Smith-Waterman* explanation in section 2.2).

2. In the Smith-Waterman algorithm, a gap opening and gap extension penalties are introduced. These are for taking into account insertions and deletions of different lengths. Such insertions and deletions introduce 'pathological' alignments by, for instance, allowing several elements of one sequence to be aligned to just one element of the other. An explanation of this phenomena has been given in section 4.4. As we do not want this to happen, we introduce this penalties directly in the 3 possible paths considered in (1).

3. As we introduce gap penalties δ directly in the paths considered (2), we have to set them in a proper way. We have to qualitatively assess all the possible cases where a gap have to be introduced, and when should it be an opening or an extension penalty. For instance, if element $(i-1,j-1)$ of the binary similarity matrix had a positive value but element (i,j) is negative, we have to add a gap opening penalty. An example of when to apply a gap extension penalty would be when $s(i-1,j-1) = s(i,j)$, and both are equal to zero or negative. No gap penalty should be applied in case $s(i,j) > 0$ (starting or continuing a similarity region).

So, in order to apply (1), we keep the first term of equation 5.16 and we introduce two more terms: $H(i-2,j-1) + s(\overrightarrow{h_i}, \overrightarrow{h_j})$ and $H(i-1,j-2) + s(\overrightarrow{h_i}, \overrightarrow{h_j})$. For accomplishing (2), the second and third

terms of equation 5.16 are removed and a gap penalty δ is added to the existing terms. As we have seen in (3), the value of δ has to obey some 'intuitive' rules. We summarize all of them in table 5.1, where we see that δ may have three values depending on the case being considered ($\delta = \{0, \delta_o, \delta_e\}$).

Path	$s(i-2, j-1)$	$s(i-1, j-2)$	$s(i-1, j-1)$	$s(i, j)$	δ	Comment
I,II,III	-	-	-	> 0	0	No gap
I	-	-	> 0	≤ 0	δ_o	Gap opening
I	-	-	≤ 0	≤ 0	δ_e	Gap extension
II	-	> 0	-	≤ 0	δ_o	Gap opening
II	-	≤ 0	-	≤ 0	δ_e	Gap extension
III	> 0	-	-	≤ 0	δ_o	Gap opening
III	≤ 0	-	-	≤ 0	δ_e	Gap extension

TABLE 5.1

Qualitative assessment of all possible cases for introducing gap penalties (a '-' sign denotes that we don't care about that value). The path column also corresponds to the respective elements inside the max{} operator in equation 5.17.

Therefore, with (1), (2) and (3), the final recurrent expression four our local alignment algorithm becomes:

$$H(i,j) = max \begin{cases} H(i-1, j-1) + s(\overrightarrow{h_i}, \overrightarrow{h_j}) - \delta \\ H(i-2, j-1) + s(\overrightarrow{h_i}, \overrightarrow{h_j}) - \delta \\ H(i-1, j-2) + s(\overrightarrow{h_i}, \overrightarrow{h_j}) - \delta \\ 0 \end{cases} \tag{5.17}$$

For $2 \leq i \leq n$ and $2 \leq j \leq m$.

The local similarity cost $s(\overrightarrow{h_i}, \overrightarrow{h_j})$ is either μ_+ or μ_-, and δ corresponds to the similarity dependent gap penalties shown in table 5.1 ($\delta = \{0, \delta_o, \delta_e\}$). Regarding μ_+ and μ_-, in the literature we find that for further calculations, what only matters is their difference $\Delta\mu = \mu_+ - \mu_-$ [Vingron 94]. So, we set $\mu_+ = 1$ and $\mu_- = 1 - \Delta\mu = \mu$. This is a commonly done trick when testing these parameters in DNA sequence comparison [Waterman 87b]. Then, the main parameters for tuning the algorithm become μ, δ_o and δ_e. After an exhaustive testing of these, we set them in the combination that resulted in a better performance, that is: $\mu = -0.9$, $\delta_o = 0.6$ and $\delta_e = 0.666$.

Notice that i, j go from 2 to n or m respectively. Thus, the initialization step for the $(n+1)\times(m+1)$ alignment matrix now becomes:

$$H(i,0) = 0; \quad H(0,j) = 0 \tag{5.18}$$

For $0 \leq i \leq n$ and $0 \leq j \leq m$. And:

$$H(i,1) = s(i,1); \quad H(1,j) = s(1,j) \tag{5.19}$$

For $1 \leq i \leq n$ and $1 \leq j \leq m$.

An example of the resultant matrix H can be seen in figure 5.4. We can see clearly two local alignment traces that correspond to two highly resemblant sections between two covers of the same *cover set*.

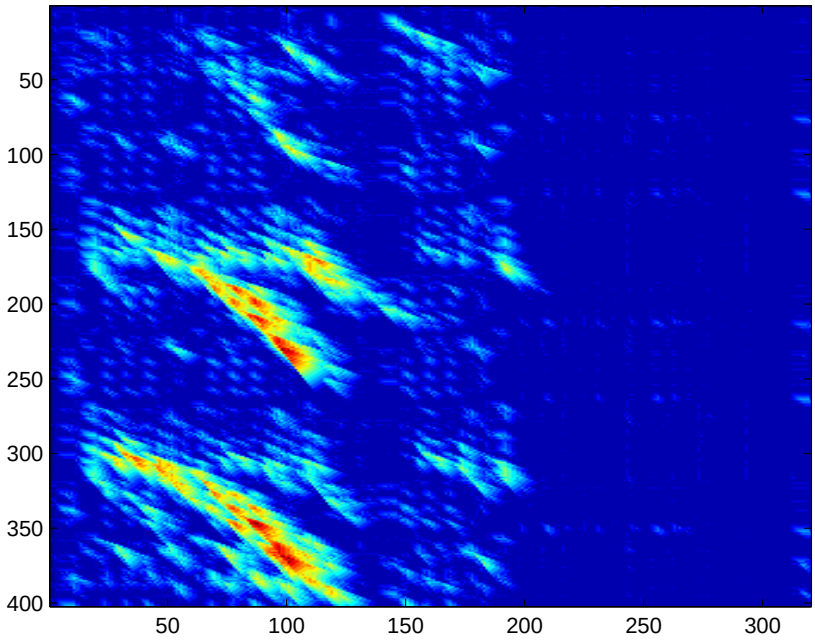

FIGURE 5.4 *Example of a local alignment matrix H with two songs belonging to the same cover set. It can be seen that the two songs do not coincide entirely (just in two fragments), and that, mainly, the second part between them is completely different.*

Finally, the score determining the local sequence similarity between two HPCP matrices, and, therefore, what we consider to be the similarity between two songs, corresponds to the value of the highest peak of H:

$$Score(HPCP'_A, HPCP''_B) = max\{H(i,j)\} \tag{5.20}$$

For any i, j such that $0 \leq i \leq n$ and $0 \leq j \leq m$.

Post-processing

For the application of cover song identification we just care about the highest local alignment peak of matrix H (around position $(370, 100)$ in figure 5.4). But for other applications, it might be of interest to track all the alignment paths in H. These applications may include song segmentation, chorus detection and cover song identification itself (maybe further improvements of the present algorithm, module B7).

If we just want to obtain the principal alignment we can follow a very simple backtrack procedure like the one shown in algorithm 1. This mainly consists in starting at position (i, j) of matrix H where the highest peak is located, and begin a traceback procedure selecting the maximum value of positions $(i-1, j)$, $(i-1, j-1)$ and $(i, j-1)$ recursively until we find a 0, which indicates that the local alignment has finished. We obtain the alignment path in reverse order.

Algorithm 1 Traceback procedure for obtaining an alignment between two sequences.

Require: matrix H
Ensure: alignment path P
  $i, j \leftarrow$ Get the position of the maximum peak of matrix H
  $P \leftarrow$ Record the position (i and j) and value visited ($H(i,j)$)
  while $H(i,j) > 0$ **do**
    $nextvalue \leftarrow \max(H(i-1,j),\ H(i-1,j-1), H(i,j-1))$
    if $nextvalue = H(i-1,j)$ **then**
      $i \leftarrow i-1$
    else if $nextvalue = H(i,j-1)$ **then**
      $j \leftarrow j-1$
    else
      $i \leftarrow i-1$
      $j \leftarrow j-1$
    end if
    $P \leftarrow$ Record the position (i and j) and value visited ($H(i,j)$)
  end while

Otherwise, if we want to have all optimal alignments, we can employ a procedure like the one used in [Waterman 87a]. In there, small modifications are introduced after the computation of matrix H in order to produce non-intersecting subsequent alignments. This briefly consists in selecting the maximum peak and employing a backtrack procedure that 'erases' the alignment path followed (and its neighbor positions) in matrix H. Then, the next maximum peak found is used to track another alignment.

Literature related with the Smith-Waterman algorithm shows that the similarity score obtained in equation 5.20 depends on the length of the sequences being compared [Vingron 94, Waterman 94b, Waterman 94a]. We have also seen this intuitively in section 2.2.2, equation 2.20. In order to check if this statement was also true for the algorithm proposed here, we performed some tests with the scores obtained. As we can see in figure 5.5 (blue straight line), the score increases as the length of songs does so. In there, a straight line is fitted to the data in a least minimum squares sense in order to see a general tendency of the score.

As we have said, this is an effect that is well known in the literature. Although in many disciplines such as DNA or protein comparison this is desirable effect (i.e., they might search for the longer DNA sequence that can be found in a database), it is not a good result for a cover song identification system, as it tries to deal with songs independently of their different length. So, in order to obtain a distance between two songs, in module B8 we normalize the score by the maximum path length possible:

$$d(HPCP'_A, HPCP''_B) = \frac{m+n-1}{Score(HPCP'_A, HPCP''_B)} \tag{5.21}$$

Where we take the inverse of the score, as we want to obtain a distance value, not a similarity measurement. As we can see in figure 5.6 (blue straight line), this overcomes satisfactorily the effect of increasing the score as the song lengths increases. This normalization factor, apart from being intuitive (the theoretically maximum alignment much be of a size lower than the sum of the lengths of the two songs being compared ($n+m$)), is commonly introduced in DTW algorithms [Rabiner 93].

So, the value obtained in equation 5.21 is the one returned by the algorithm when two songs are compared. This value is usually below 10 for highly resemblant covers, between 10 and 20 for covers that are not so similar (here we find some false positives) and higher than 20 for songs not belonging to the same cover set.

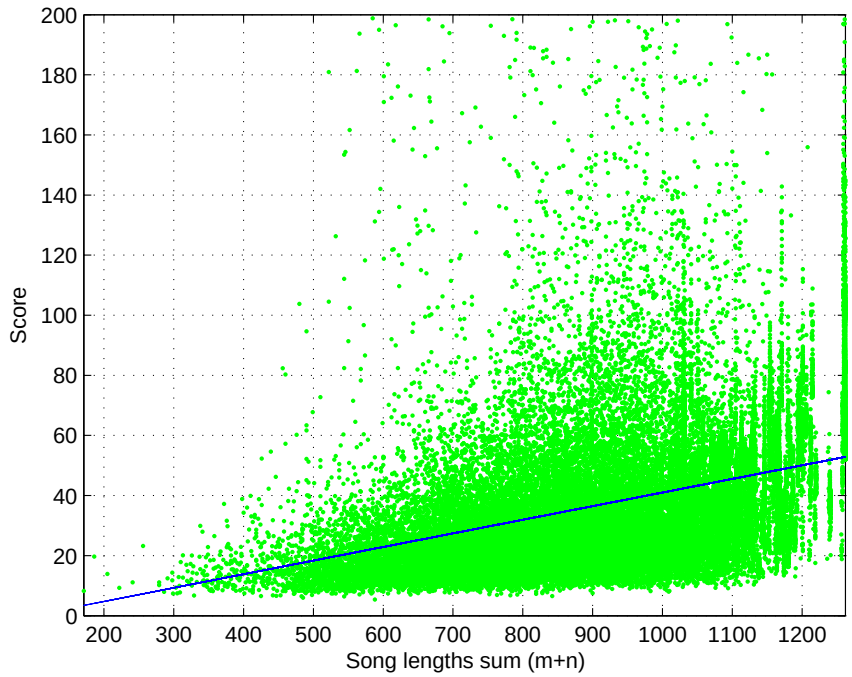

FIGURE 5.5 *Score study (unnormalized) for DB330. Obtained score increases as the length of the compared songs does so. We plot the score versus the sum of the lengths $n + m$ (green dots) and a straight line fitted in the least minimum squares sense (blue straight line).*

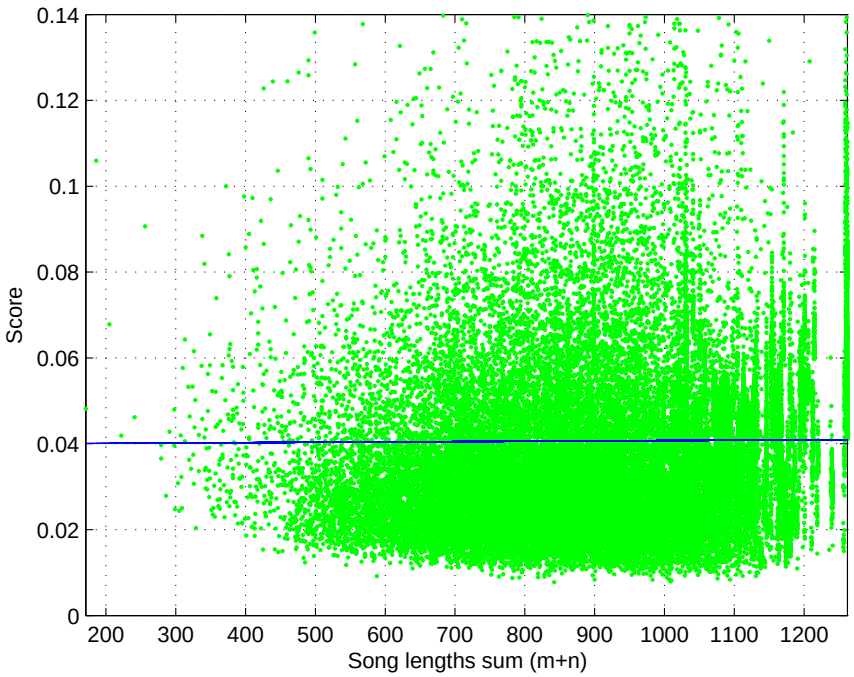

FIGURE 5.6 *Score study (normalized) for DB330. Obtained score after normalization does not increase with length.*

5.2 Evaluation

We now present the obtained results for the algorithm proposed in this section. Firstly, we carry out an evaluation with the same measures used along all this thesis. Secondly, we give more intuitive look at the results from a more 'musicological' perspective.

5.2.1 Performance evaluation

In a first experiment, we compared performance results for different *averaging factors* (module B2) as done in section 4.4.2 with the improved DTW approach. Results for the geometric mean of *bpref** are summarized in figure 5.7. These are similar to the ones obtained in table 4.8 of the forementioned section. We can see that performance improves as the framelength decreases (and therefore, employing more detail into our analysis) until a certain peak around 0.25 ms (an *averaging factor* of 5-7 in the plot). Also, we corroborate that using a beat tracker lowers the performance of our algorithm (black label next to the curves).

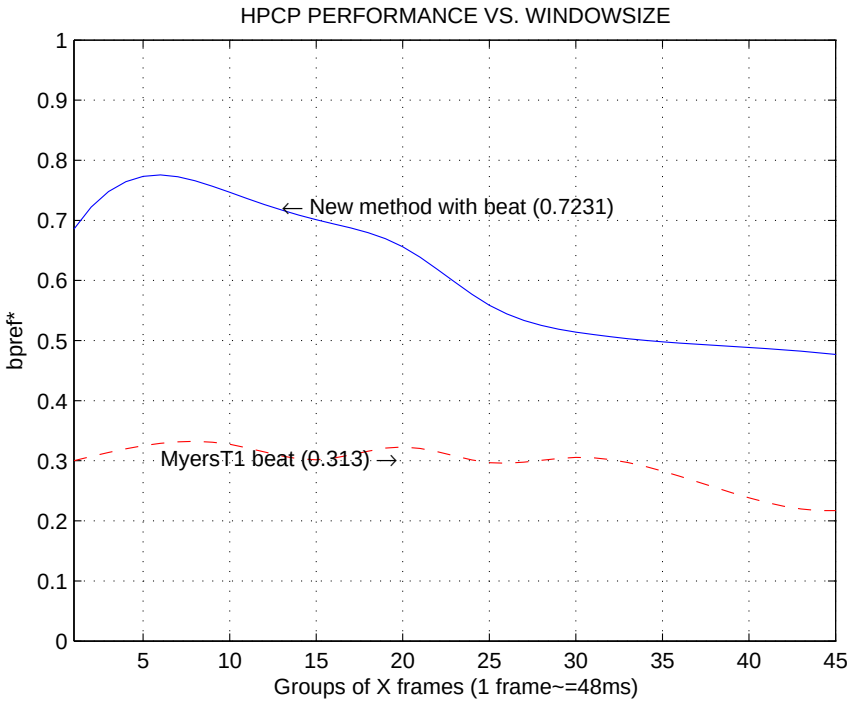

FIGURE 5.7 *Performance depending on the framelength. The proposed algorithm (continuous blue line) is compared with IDTW algorithm (dashed red line) with MyersT1 local constraints explained in section 4.4.1. We highlight the performance obtained by using a beat tracker with black labels next to the respective curves. Results correspond to experiments with DB75.*

As we have said in section 5.1.2, several tests have been done in order to tune the parameters μ, δ_o and δ_e for the Dynamic Programming local alignment (DPLA) algorithm. These values, unless being considerably different from the cited ones, have little effect on the final performance of the algorithm (for example, an increment of 25% in δ_e leads to a performance reduction around 4% in the geometric mean of *bpref**). As it have been said, an optimal configuration has been found to be: $\mu = -0.9$, $\delta_o = 0.6$ and $\delta_e = 0.666$.

We now draw a table that compares the performance obtained with the proposed DPLA algorithm and the ones discussed previously in chapter 4. *MNCI* is shown in table 5.2, the *F-measure* is shown in table 5.3 and the geometric mean of *bpref** in table 5.4. Compared methods are the ones that have lead to better performance for each type of experiment: base-line experiment (BLE) of section 3.5, improved cross-correlation approach (ICC) of section 4.2, improved DTW approach (IDTW) of section 4.4, and the Dynamic Programming local alignment method (DPLA) proposed here in this chapter.

Method	DB75	DB330	DB2053
BLE	0.037	0.033	0.003
ICC	0.573	0.296	0.080
IDTW	0.600	0.408	0.169
DPLA	0.823	0.619	0.272

TABLE 5.2
MNCI for compared methods for 3 different databases: DB75, DB330, and DB2053.

Method	DB75	DB330	DB2053
BLE	0.046	0.043	0.006
ICC	0.638	0.348	0.169
IDTW	0.651	0.485	0.399
DPLA	0.868	0.688	0.601

TABLE 5.3
F-measure for compared methods for 3 different databases: DB75, DB330, and DB2053.

Method	DB75	DB330	DB2053
BLE	0.017	0.015	0.011
ICC	0.381	0.175	0.046
IDTW	0.327	0.263	0.128
DPLA	0.747	0.469	0.288

TABLE 5.4
Geometric mean of bpref for compared methods for 3 different databases: DB75, DB330, and DB2053.*

We can appreciate that the final performance of DPLA is more or less twice higher than the ones achieved with other methods. Results are confirmed with different databases.

In addition, in figure 5.8, we plot a normalized *lift* curve for the previous methods. We can see that within the 10 first answers we reach an accuracy around 60% of correctly retrieved songs. This value is highly superior to the performances achieved for ICC and DTW methods (around 20 and 40 percent respectively), and is very far from the one achieved randomly (base-line experiment), which is near 0%. Note that, as explained in section 3.1, here we cannot trust much *MNCI* nor *F-measure* for a database with different number of covers per set (like *DB2053*), as they do not consider the total number of covers whithin the database (i.e., they do not care about the difference in retrieving the only possible item of a set, or one of the largest labelled group of covers).

An accuracy around 60% (figure 5.8) for such a big database (more than 2000 songs) is highly noticeable if we consider state-of-the art approaches evaluated in MIREX 2006 (with accuracies ranging from 6 to 23 percent with a music collection of only 330 songs, see section 4.5) and all the methods implemented in chapter 4 (with a maximum accuracy of 38% for IDTW approach with *DB2053*).

Finally, in order to test the robustness of the distance employed in DPLA approach (equation 5.9), and also for quantifying in some sense the improvement that it had in the overall performance without

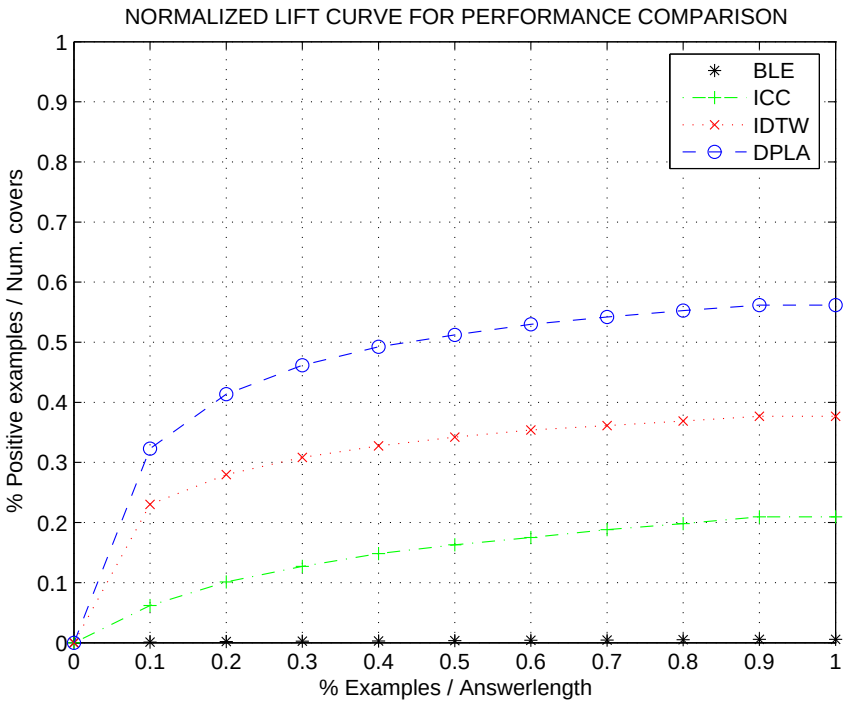

FIGURE 5.8 *Normalized lift curve comparing the DPLA approach (blue circles) with the ICC (red crosses) and IDTW (green sum signs) methods, and the base-line experiment (black bottom asterisks) for DB2053.*

taking into account the effect of the alignment procedure and vice versa, we tested IDTW and DPLA approaches with 12 and 36 bins, and with the *OTI*-based binary similarity measurement of equation 5.9 and the correlation between HPCPs employed in chapter 4. The results are shown in table 5.5.

Method	HPCP bins	HPCP Correlation	OTI Similarity
IDTW	12	0.327	0.344
IDTW	36	0.373	0.374
DPLA	12	0.504	0.742
DPLA	36	0.583	0.747

TABLE 5.5
OTI-based similarity measurement robustness. Geometric mean of bpref for different configurations of HPCP bins and distances employed. Experiments done with DB75.*

We realize that while not showing a significant improvement with IDTW, *OTI*-based binary similarity measurement has a deep impact in DPLA approach performance. This might be due to the fact that with IDTW method we are not exploiting the binarization of the similarity matrix S explained in section 5.1.2. Regarding the robustness against different HPCP resolution, we see that while with correlation distances we had differences around 15% in the *GM-bpref** measure (0.373/0.327 for the IDTW algorithm and 0.583/0.504 for the DPLA approach), we get differences of 7% for IDTW and 1% for DPLA.

5.2.2 'Musicological' evaluation

One way of qualitatively assessing performance of classifiers or retrieval systems is through a confusion matrix. This matrix usually shows the number of classified instances for each category. In there, each column represents the instances in a predicted class, while each row represents the instances in an actual class. One benefit of a confusion matrix is that it is easy to see if the system is confusing two classes (i.e. commonly mislabelling one as another).

For this type of evaluation we consider that a *canonical version* plus its covers belong to a single class (*cover set*), and that this is different from others comprising another *canonical song* plus its versions. We show the confusion matrix corresponding to *DB330* (table 5.6). This has been obtained just considering the 10 first retrieved answers for each query, and each row corresponds to one *cover set*. So, as in *DB330* there were 11 songs for each cover set and we just look at the 10 first retrieved answers, the total amount of elements for each row must be $11 \times 10 = 110$ (also the maximum number of instances classified in one category).

The way to interpret this confusion matrix is to look at each row and check whether the majority of songs have been classified in the correct cover set (a perfect classification with 100% accuracy would result in a diagonal matrix). For instance, if we want to know if the items labeled *nowomanno* (corresponding to covers of the song "No woman no cry" originally performed by Bob Marley) have been confused with the items labelled *across* ("Across the universe" by The Beatles), we just have to look at row *nowomanno* and column *ac*. For space reasons, in the confusion matrix columns we just write down the first two letters of the labels, so, for instance, *across* becomes *ac*.

By looking at table 5.6, we see that the majority of the classifications rely on the main diagonal of the matrix, which indicates us a relatively good performance. Covers better identified are *aforest* ("A forest", originally performed by The Cure) and *letitbe* ("Let it be", originally performed by The Beatles), with more than 100 correct classifications. Other correctly classified items are *yesterday*, *dontletme*, *wecanwork* (corresponding to "Yesterday", "Don't let me down" and "We can work it out" respectively, all originally performed by The Beatles) and *insensatez* ("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 labels associated with Beatles' songs (14 out of 30), but it can also be associated to the easy and well defined tonal progression, that, in comparison with other more elaborated ones (i.e., "Over the rainbow", *overrainbow*), leads to better identification.

If we observe the confusion matrix in more detail, we can appreciate that there are some songs, such as "Eleanor Rigby" and "Get Back", that cause 'confusion' more or less with all the queries made (columns *el* and *ge*). One explanation for this might be that these two songs are built over a very simple chord progression involving just two chords (or main tonality profiles): the tonic and the mediant (i.e., C and Em) for the former, and the tonic and the subdominant (i.e., C and F) for the latter. So, as they rely half of the time in the root chord, any song being compared to them will be share half of the tonal progression.

An interesting misclassification is done with the item *nowomanno* ("No woman no cry" originally performed by Bob Marley). These covers are associated more than 1/3 of the times with the song *letitbe* (*le* column), but the good surprise comes when we analyze the tonal progression and we discover that both share the same chords in different parts of the theme: C - G - Am - F. Thus, this might be a 'logical' misclassification.

	af	an	bo	co	cr	el	en	ge	he	hy	in	ip	le	lo	no	ov	so	st	ti	ye	ac	ba	di	do	mo	ni	sh	th	wa	we
aforest	101			2		5												2												
andilove	9	48	3	4	1	13	4		2		1			5				3			2	1		2	2		5	1	4	
boysdont			80			4		5					6	6	3	1					1	1							3	
cometoget	2			62		3	6	12						1				5				3			5				11	
corcovado		1	5		75	2	1		1		1				1			1	1	2	3	5					6	1	2	2
eleanor	1	1		10		46	7	7						9				8				2			4		1		14	
enjoysilen	1			6		17	60	6						1				4	2		1	3			1			1	7	
getback				2				83	1	1								4			3	1		3			1	1	10	
herecomes		1	1					9	59	7			7	2	6						5			2			3	4	2	2
heyjude								11	3	84			2		1						2	1		5					1	
insensatez	1				1		1		1		88				1			1		1	1	7		2			3		1	1
ipanema	1		2			1	2	3	6	4		60		3					1	1	5	6		6		1	4			4
letitbe													106		4															
lovesong			4	3		5	1						4	66	4	1		1			3	3			6	1	1		6	1
nowomanno													43	1	63	1		1				1								
overrainbow		2	3	3		1	2	17	7	2			8	3	12	11		2	2		13	2		10	1		2	3	4	
something				2	1	4		3	2	1				1			80			1	4	7			1		2		1	
stairway2h	6	3		5	1	16	2	12	2	2	1		1	5	6	1		32	1			1	1				2		8	2
ticotico	2		3	1		4	3	4	10					3		1			61		4	3			1		1	2	2	5
yesterday			1			2							1	1	4					90				2	2		7			
across								2	2	5			3		1						86	1								
battleepp	1		2	4		3	5	12	4	5					3			2	2		7	27		3		2	1	2	8	7
dieroboter				4		5	2	2											1				77				1	2	6	
dontletme										2											1			90				1	6	
money				12		5		1						2		1		5							68				6	
nightday			1	2		6	2	4	1	3				1	2			1	2		2	8	1	2	2	49	4	3	3	1
shelovesyou	11	4	1			1		6	6	2			9	1	2		1	1		6	6			5	3		31	2		2
thefool						1	2	7	9	2				1					3			2		8				64	1	
walkingin				2		15	3	10	1	1			1	2	3			6			1	3		6					45	1
wecanwork								7	1				2		1							1								88

TABLE 5.6
Confusion matrix of DPLA approach with DB330.

We can also appreciate other 'confusions', for instance, the label *boysdont* ("Boys don't cry", originally performed by The Cure), in six occasions is classified as *lovesong* ("Love song", originally performed by the same group). Such 'confusions' may be justified by the fact that the original artist or performer is the same, but since we are looking at tonal sequence similarity (and not to timbre or other musical aspects), this justification becomes not plausible (see also subsection 1.4.4).

Sometimes we can interpret that the same wrongly answered items for one set of covers is reciprocally answered when we query these answers. This would be the case of *eleanor* and *walkingin*: when you query covers of *eleanor*, you get 14 items in the *walkingin*, and when done reciprocally (querying *walkingin*), you get 15 elements with the *eleanor* label.

Other badly classified items are *overrainbow* ("Somewhere over the rainbow", originally performed by Judy Garland), *battleepp* ("The battle of Epping forest", Genesis) or *stairway2h* ("Stairway to heaven", Led Zeppelin).

In general, we have a fairly diagonal-dominated confusion matrix, with some particular cases that must be considered in detail when doing further improvements of the proposed algorithm.

Another way to visually see the output of the algorithm is to create a dendrogram starting from the distance matrix obtained. A dendrogram is a tree diagram frequently used to illustrate the arrangement of the clusters produced by a clustering algorithm, and is often used in computational biology to illustrate the clustering of genes and to see which sequences share similarity.

We are now going to perform an analysis of the results with two examples. The first one corresponds to the dendrogram obtained with the DPLA approach for all the available covers of the song "Bohemian Rhapsody", originally performed by Queen (figure 5.9).

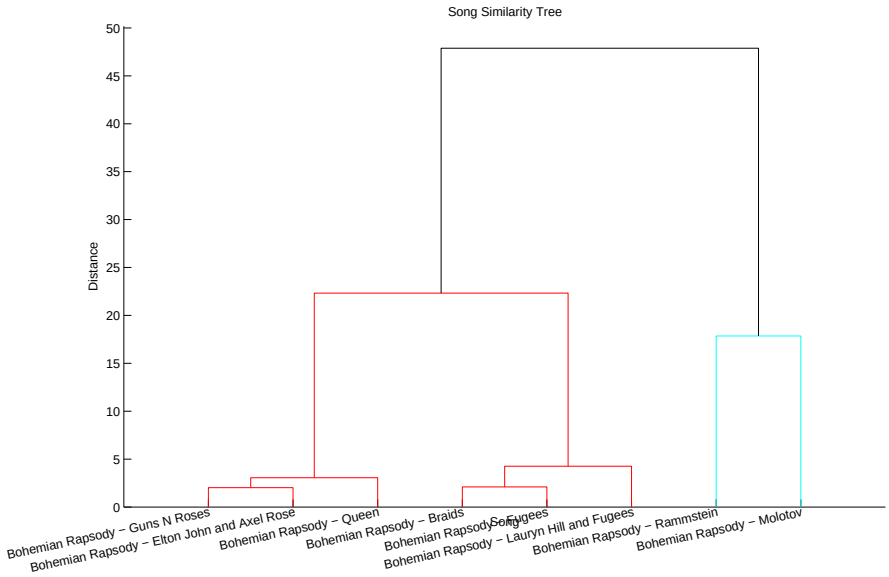

FIGURE 5.9 *Song dendrogram example computed from the distance matrix obtained with DPLA. All the available covers of the song "Bohemian Rhapsody" (originally performed by Queen) have been used.*

If we take a quick look to figure 5.9, we can see that the clustering obtained has some sense. The two first items correspond to the same version of the song (different remaster) performed by the same artists (Axel Rose and Elton John) with some members of the original band (Queen). These two songs, apart from sharing the highest resemblance, are very similar to the original one (third item). In them, the original song structure, tempo and tonal progressions are highly conserved. A same thing happens

with the next three clusters, where the tonal progression of the song is slightly changed. Finally, the two songs of Molotov and Rammstein are shown in a different cluster. They change substantially from the original one (tonal progression, structure, lyrics, central melody, instrumentation, etc.).

Finally, a second example is shown where three groups of covers are compared. The obtained dendrogram after processing the resultant distance matrix is shown in figure 5.10.

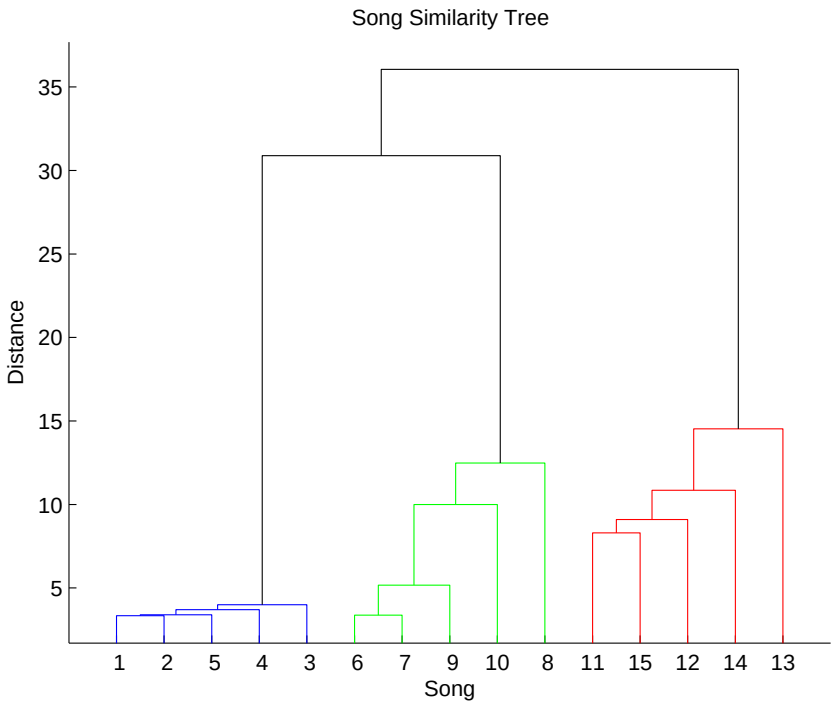

FIGURE 5.10 *Song dendrogram example computed from a distance matrix obtained with DPLA algorithm. Three different cover sets are compared.*

In there, songs 1 to 5 correspond to covers of the song "Angie" (originally performed by The Rolling Stones), songs 6 to 10 correspond to covers of the song "Billy Jean" (Michael Jackson) and songs 11 to 15 are versions of "Garota de Ipanema" (Antonio Carlos Jobim). As these are 3 clearly differentiated songs regarding its tonal progression, the algorithm does a good job in assigning distances between them. As a curiosity we have to remark that original songs are 1, 6 and 10, to which all covers refer. We can also see higher distances between songs of the third group. As these are, most of them, jazz variations of the same theme, tonal sequences have noticeable variations, and therefore, the distance between them gets higher.

5.3 Discussion

In this chapter we have presented a new method for detecting cover songs based on the similarity of tonal sequences. We have overviewed it and provided a general block diagram of the system in section 5.1.1. Next, in section 5.1.2, we have explained with detail all the parts of the method, with justifications on the decisions taken when necessary. Finally, in section 5.2.1, we have evaluated the performance of this new algorithm comparing it with the improved versions of some implemented state-of-the-art methods (chapter 4). In addition, a qualitative assessment of the results provided by this new approach has been

done in section 5.2.2.

The general system architecture presented (pre-processing, similarity matrix calculation, alignment and post-processing) is common in many of the approaches found in the literature. An example of that is also the DTW approach implemented in sections 4.3 and 4.4.

Pre-processing step is common in all systems: features are extracted (chroma vectors, PCP, HPCP, ...) and these further processed in order to lead to a better and more compact representation of the tonal behaviour of the song. In our case, we just normalize the vectors in order to overcome differences in energy, and we average them in (using an *averaging factor* of 5 or 10) to improve accuracy and speed (we have shown that this was so with an implementation of the improved DTW algorithm in section 4.4.2 and we have corroborated the result in the evaluation of the DPLA algorithm in section 5.2.1). As we had seen in section 4.2.2 that a good way to deal with key transpositions was to compute a global HPCP and then shift one song accordingly, we apply the same technique in the proposed system.

The main novelty of the algorithm described in this chapter consist in using a local alignment procedure, and using a non-euclidean distance for determining the similarity between two HPCPs. As we have argued in section 2.6, these two issues are poorly addressed in the literature. Furthermore, through the implementation of the systems in chapter 4, and, specially, in section 4.5, we have stated that they were important aspects that had a direct effect into the final performance of the system.

HPCP distance has been calculated in a binary manner with equation 5.10. As we have said in the *Similarity matrix creation* subsection (inside section 5.1.2), this has many advantages: having a highly contrasted similarity matrix, and allowing us to use a wider spectrum of techniques borrowed from other disciplines for processing text strings, DNA or protein sequences, etc. (being among this spectrum the local alignment technique that has inspired us out Dynamic Programming alignment). Furthermore, we have shown that this proposed HPCP binary distance leads to a significant increase in performance (table 5.5), which is more evident when further processing is specially designed to deal with it (like the proposed DPLA algorithm). In addition, we have provided an expression for a fast computation of this similarity measure (equation 5.14).

A local alignment approach have been followed. This was also another of the main desirable qualities debated in section 2.6 that few systems in the literature covered. We have started with an implementation of the Smith-Waterman algorithm, and we have developed a new recursive expression (equation 5.17) to compute a local alignment matrix with Dynamic Programming techniques (section 5.1.2, *Dynamic Programming local alignment* subsection). All this has been done while properly justifying the decisions and parameters chosen, and, where there was no objective justification for these, several tests have been performed to assess the values that lead to best performance (i.e., with μ, δ_o and δ_e).

Post-processing issues (*Post-processing* paragraph in section 5.1.2) that we have dealt with include the normalization of the final score, and a straightforward backtracking algorithm to obtain a local alignment between two sequences. Although this final facet has not been fully explored, it remains as an easy-to-do improvement for further implementations of the same algorithm, or for new applications based on the same scheme (such as song segmentation, chorus extraction, obtaining the common tonal sequence of two songs, etc.). The expression that normalizes the score (equation 5.21) has been obtained in an intuitive way, but it responds to the literature statements and, more importantly, has been comproved to work correctly for our application (figure 5.6).

We have performed an evaluation of the proposed algorithms' output in multiple aspects. Performance results have been obtained through an evaluation that has been carried with a big database comprising more than 2000 songs, thus giving high confidence on the values obtained. The same evaluation measures used along this thesis have been used, what let us easily compare between systems.

The proposed method has shown a significantly higher performance with all the evaluation measures tested than the approaches followed in chapter 4, which were improved versions of two state-of-the-art systems (tables 5.2, 5.3, 5.4, and figure 5.8). This performance reaches a top value of 60% of correctly detected covers whithin the first 10 answers for each query using the forementioned database. This is a high value if we take in consideration that cover song systems usually have a poor overall accuracy (see sections 2.5, 4.5 from previous chapters and 5.2.1 from this one).

Regarding time complexity, we have to note that the the similarity matrix creation and the DPLA processes result in an $O(2nm)$ algorithm. With the experiments performed we have stated that the average time latency is around 1.3 sec for each two songs being compared. This may not a brilliant aspect of the algorithm proposed, but, if we take a look at time latencies of the algorithms evaluated in the MIREX 2006 contest (between 0.01 and 1.5 seconds) and at the performance results of this section, it worths the accuracy achieved.

Finally, we have commented some other aspects relating to the evaluation of the proposed system in section 5.2.2. This more 'subjective' assessment of the output of the DPLA approach has been done based on a confusion matrix (table 5.6) and some specific dendrograms (figures 5.9 and 5.10). With this, we have seen that the results are coherent with the feelings that one might have when listening to the compared songs, or, if one knows them, when looking at their titles.

Chapter 6

Conclusions and future work

Along this thesis, we have addressed several issues that concern tonal sequence similarity from a computational point of view. We have focused on cover song identification, as this task provides a direct and objective way of evaluating tonal sequence similarity. Here we dedicate some lines to give a brief summary of the work done, and a list of some potential research issues for the future. Finally, a short general conclusions section closes the chapter.

6.1 Summary of achievements

We have started with an extensive introduction into the field of **music similarity**. We have explained the motivations for addressing this aspect, and we have given a brief insight into the interdisciplinary research area of Music Information Retrieval (MIR), where we think this research belongs mostly. We have overviewed some of the main techniques that are being used to determine music similarity, and we have then justified the **use of cover songs** for partially assessing it. From here, several **conceptual aspects about cover songs** have been reviewed: the term itself (with its origins, cultural and social implications), the types of covers that one may find (with a clear explanation on the connotations of them), the musical characteristics that might change in a cover song, an enumeration of the most covered themes, and a list of resources that emphasizes the increasing importance of versions in nowadays society.

We have **placed the cover song identification task** within other Music Information Retrieval areas where a bigger research effort has been done: audio fingerprinting and artist/genre classification. We have also given a short introduction on them, and we have discussed about the suitability for cover song identification to be considered between these two areas. Then, the **concept of tonal sequence similarity** has been introduced, and we have seen that this was a convenient facet of the music to consider when dealing with covers.

We have **reviewed** the main features, techniques and works relevant to the task of content-based music similarity, with a special **emphasis in audio cover song identification** systems. A review of the commonly used low-level descriptors has been done: energy descriptors, timbral features, and tonal descriptors. Also, we have explained the most relevant techniques for obtaining mid and high-level descriptors for the task being considered: beat tracking, chord identification, and melody extraction algorithms.

As the **alignment procedure** employed is an important part of any sequence similarity system, we have focused on the most relevant ones: Dynamic Time Warping (DTW), Edit-distances, and Hidden Markov Models (HMM) are introduced and explained.

We also have given a brief introduction and cited some basic references of **previous work related to**

audio content-based similarity. Furthermore, a special treatment has been given to works dealing with music alignment and the processing of sequences of descriptors. As a part of this, we have remarked specific systems designed to perform audio cover song identification. In addition, some important **shortcomings found in the literature** about cover song identification have been carefully highlighted.

Our **evaluation methodology** has been carefully described. After a proper introduction section, the evaluation measures considered along this thesis have been presented and commented. We also have presented our **music collection**, and given some statistics on it. We also explained a base-line experiment done in order to have some preliminary reference for future tests on this song database.

We have implemented two **state-of-the-art approaches**. These were chosen because they were well known in the MIR community and because they involved the main techniques used in the literature. First experiments with them have also provided a reference of the performance of such systems when dealing with our music corpus. Furthermore, we have experimented with several new **variants for the proposed methods** in order to improve accuracy and to see relevant aspects and processes of them. As we have stated, some of these improvements have been found to have a high impact on final performance.

Finally, we have approached and implemented a **new method for determining tonal sequence similarity** from a local point of view. We have given a detailed explanation on the approach followed. Furthermore, we have **confirmed that the main assumptions** we had made were right, and we have achieved a very **significant increment in performance** compared to previous implemented approaches. A proper **evaluation of the solution** has been done, and a more 'intuitive' insight about the goodness of the results has been presented.

6.2 Open issues and future work

Many improvements and considerations regarding the work exposed in this thesis can be done. In this section we will try to summarize some aspects that could be improved, and some future experiments and ideas that can be derived from this work. In order to organize the ideas, three lists are made. The first one discusses some additional issues and refinements that should be made to our music corpus in order to perform future experiments. The second one highlights several improvements that could be done to the method presented in chapter 5. Finally, the third subsection lists many of the future experiments based on the work of this thesis that we can think of.

6.2.1 Music collection refinements

First, we should do a revision of our song database, and, at the same time (if possible), increase the collection. There are plenty of resources available on the web (section 1.4.2) and, furthermore, our personal music collections keep increasing, so it is not difficult to think that we can find more covers in a straightforward way.

We also need perhaps to introduce some new tags to the database in order to perform a wide variety of experiments. Apart from trying to quantize the involved musical facets that change in every cover (section 1.4.4), we might obtain new data for the items that we already have in the database. These could include: the year where the cover was released, the country, some tags describing the mood transmitted by the song, etc. For instance, the year could be useful for trying to determine a 'historical' evolution of the tonal content, the country for possibly obtaining a geographical evolution, etc.

6.2.2 Cover song identification

Although we have carefully considered the values of the parameters employed in our implementation, it can always exist some possibility of improvement with the tuning of these. So, one thing to do would be to check the values assigned to important parameters.

We could study a more compact representation of the tonal sequence song. This might improve speed without loss of performance (even we could increase it). One first idea is to try to average HPCPs with a variable *averaging factor* depending on the similarity of the vectors (i.e., the HPCPs that are considered to be the same being averaged, and a number representing the original duration of the new vector added to it).

Also the extraction of the tonal features can be tested (subsection 2.1.1), as some experiments with quantizing and thresholding PCP features have been reported in the literature.

Still dealing with HPCP descriptors, we think there is much room for improvement regarding the distance measure being used between them. As we have said, this is an open issue that must be approached from an interdisciplinary point of view. One possibility that we have in mind is to compile a database of short musical sequences sharing a similar HPCP and perform a statistical survey to get binary resemblance judgements within humans. After that, we could learn a (binary) distance between HPCPs, for example, with Distance Metric Learning methods [Yang 06].

One of the important things to do in future cover song identification systems is the inclusion of new features describing other aspects or characteristics that might be relevant to accomplish such a task. We are thinking in the energy and timbre descriptors mentioned in the literature, but we also consider adding some rhythmic descriptors (i.e., Inter-Onset Interval (IOI)), tonal and timbral complexity descriptors, bass-line estimation algorithms, etc.

On the side of the alignment, the information that can provide us the study of the multiple alignments found in a local alignment matrix (*Dynamic Programming local alignment* paragraph, section 5.1.2) remains unexplored.

Also, we could exploit other local alignment techniques such as the Smith-Waterman algorithm itself, the Longest common subsequence (LCSS) distance, the use of Hidden Markov Models (HMM), and some other methods mentioned in section 2.2. It could also be interesting to consider systems from other disciplines that intensively research on sequence alignment (i.e., Data Mining, DNA sequence comparison, etc.).

6.2.3 Future insights

As we have mentioned in section 5.1.2, one straightforward output of the method proposed when comparing one song against itself could be to obtain a segmentation in different parts of the song. Also, perhaps the most remembered parts such as the chorus section could be segmented.

Although in previous subsection we have stated that the distance calculation between HPCPs needs additional research, it could be good to assess if the binary similarity measure proposed in chapter 5 has a beneficial effect into other tasks using chroma or PCP features (i.e., audio fingerprinting, song segmentation, genre classification, etc.).

An interesting direction to take would be to study the facilitations that might provide the existence of more than one cover within a database. We are going to explain it through an example: let's define A, B, C as versions of α. If the algorithm finds that A and B are very close to α, and C is very close to A and B, then you can infer (adding a post-processing function) that C could also be a version of α too [Ong 07].

We are also very interested in exploiting new manners of visualizing song similarity from a sequential point of view. In section 5.2.2 we have shown some dendrograms reflecting relations between songs, but also other techniques such as Self Organizing Maps (SOM) and some graph creation engines could be exploited in this context.

New visualization techniques perhaps could lead us to the concept of finding 'ancestors' of songs, or consider the tonal evolution (in a life-cycle sense) of songs. For that, it could be interesting to include some tags describing the year and country where the songs were released (as we have said in previous subsection 6.2.1).

A different line of research consists of quantifying the implications that tonal sequence similarity may have in mood determination. In this scenario, one first experiment to do would be to see if the similarity measure obtained classifies correctly previously manually labelled music excerpts.

Finding the most unusual sequence within a song can be of great profit by a cover song identification system if this retains the main characteristics of a track. Otherwise, it can be of great utility when discarding parts of the song to analyze.

A further idea consists in tying to predict musical sequences based on previous processed information. In these experiments, we could exploit some models to predict time series (i.e., time series forecasting).

6.3 Final conclusions

The processing of sequences is a poorly addressed topic in the research done until now in audio content-based music similarity. As music is a temporal succession of events, this should not be the case for future developments of systems dealing with it. There is a huge amount of approaches in the literature for musical analysis from score representations. This literature has been restricted until now to the available set of MIDI files. But nowadays, signal processing tools are able to automatically extract relevant information from audio, as for instance the distribution of pitches (tonal features). This fact opens the chance of using current symbolic audio sequence-oriented methods to analyze directly audio material. Furthermore, with the application proposed in this thesis, we have demonstrated that the processing of sequences of descriptors can be of high utility. So, we expect future similarity systems that extract relevant information from raw audio to look at music more 'sequentially'.

We have proven that temporal series of tonal descriptors are useful for a cover song identification task. Although the performance achieved is around 60%, we still have room for improvement with other sequences of features that might be of importance such as rhythm, timbre, complexity, etc. In addition, with an extensive literature review such as the one done in this thesis, we have been able to consider some conflictive blocks of the systems previously proposed. Furthermore, we have adressed these conflictive blocks in a constructive manner, bringing into light solutions that have proven to work.

Cover song identification is still a reduced area of research, but the results obtained in it can be of great importance for closely related fields such as artist, genre or mood classification, audio fingerprinting, song summarization and segmentation, music understanding, music cognition and perception, and many others.

We think that the use of sequences of descriptors, a better similarity assessment between tonal statistical features, and a local consideration of descriptors sequences, will have a strong effect on Music Information Retrieval systems, and much more research in the next future will be devoted to analyze and compare music according to these three main considerations.

Bibliography

[Adams 04] N. H. Adams, N. A. Bartsch, J. B. Shifrin & G. H. Wakefield. *Time series alignment for Music Information Retrieval*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2004.

[Allen 90] P. E. Allen & R. B. Dannenberg. *Tracking musical beats in real time*. Int. Computer Music Conference (ICMC), pages 140–143, 1990.

[Altschul 90] S. F. Altschul, W. Gish, W. Miller, E. W. Myers & D. J. Lipman. *Basic local alignment search tool*. Journal of Molecular Biology, no. 215, 1990.

[Aucouturier 02a] J. J. Aucouturier & F. Pachet. *Finding songs that sound the same*. IEEE Workshop on Model based Processing and Coding of Audio (MPCA), November 2002.

[Aucouturier 02b] J. J. Aucouturier & F. Pachet. *Music similarity measures: what's the use?* Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2002.

[Aucouturier 03] J. J. Aucouturier & F. Pachet. *Representing musical genre: a state of art*. Journal of New Music Research, vol. 32, no. 1, pages 83–93, 2003.

[Aucouturier 04] J. J. Aucouturier & F. Pachet. *Improving timbre similarity. How high is the sky?* Journal of Negative Results on Speech and Audio Sciences, 2004.

[Aucouturier 06] J. J. Aucouturier. *Ten experiments on the modelling of polyphonic timbre*. PhD thesis, University of Paris, France, May 2006.

[Baeza-Yates 99] R. Baeza-Yates & B. Ribeiro Neto. Modern Information Retrieval. ACM Press Books, 1999.

[Batlle 02] E. Batlle, J. Masip & E. Guaus. *Automatic song identification in noisy broadcast audio*. IASTED Signal and Image Processing Conference, 2002.

[Batlle 03] E. Batlle, J. Masip & P. Cano. *System analysis and performance tuning for broadcast audio fingerprinting*. Proc. of the Int. Conf. on Digital Audio Effects (DAFX), September 2003.

[Baum 67] L. E. Baum & J. A. Eagon. *An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology*. BAMS, pages 360–363, 1967.

[Bello 05] J. P. Bello & J. Pickens. *A robust mid-level representation for harmonic content in music signals*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2005.

[Berenzweig 03] A. Berenzweig, B. Logan, D. P. W. Ellis & B. Whitman. *A large scale evaluation of acoustic and subjective music similarity measures*. Int. Conf. on Music Information Retrieval, 2003.

[Blackman 99] S. S. Blackman & R. Popoli. Design and analysis of modern tracking systems. Artech House, 1999.

[Borlund 03] P. Borlund. *The concept of relevance in IR*. Journal of the American Society for Information Science, no. 54, pages 913–925, 2003.

[Brossier 07] P. Brossier. *Automatic annotation of musical audio for interactive applications*. PhD thesis, Queen Mary, London, 2007.

[Brown 91] J. C. Brown. *Calculation of a constant Q spectral transform*. Journal of the Acoustical Society of America, vol. 1, no. 89, January 1991.

[Brown 92] J. C. Brown & M. S. Puckette. *An efcient algorithm for the calculation of a constant-q transform*. Journal of the Acoustical Society of America, vol. 1, no. 89, pages 425–434, 1992.

[Buckley 00] C. Buckley & E. M. Voorhees. *Evaluating evaluation measure stability*. SIGIR'00, pages 33–40, 2000.

[Buckley 04] C. Buckley & E. M. Voorhees. *Retrieval evaluation with incomplete information*. SIGIR'04, no. 27, 2004.

[Cano 02a] P. Cano, E. Batlle, T. Kalker & J. Haitsma. *A review of algorithms for audio fingerprinting*. Int. Workshop on Multimedia Signal Processing, December 2002.

[Cano 02b] P. Cano, E. Batlle, H. Mayer & H. Neuschmied. *Robust sound modelling for song detection in broadcast audio*. Conv. of the Audio Engineering Society (AES), no. 112, May 2002.

[Cano 07] P. Cano. *Content-based audio search: from fingerprinting to semantic audio retrieval*. PhD thesis, MTG, Pompeu Fabra University, Barcelona, Spain, 2007.

[Carbonell 98] J. Carbonell & J. Goldstein. *The use of MMR, diversity-based reranking for reordering documents and producing summaries*. SIGIR'98, pages 335–336, 1998.

[Casey 06a] M. Casey & M. Slaney. *The importance of sequences in musical similarity*. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), May 2006. Toulouse (France).

[Casey 06b] M. Casey & M. Slaney. *Song intersection by approximate nearest neighbor search*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), October 2006.

[Casey 07] M. Casey & M. Slaney. *Fast recognition of remixed music audio*. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), May 2007.

[Chew 00] E. Chew. *Towards a mathematical model of tonality*. PhD thesis, Massachussets Institute of Technology (MIT), USA, 2000.

[Cleverdon 70] C. W. Cleverdon. *Evaluation of tests of information retrieval*. Journal of Documentation, no. 26, 1970.

[Cohn 97] R. Cohn. *Neo-riemannian operations, parsimonious trichords, and their tonnetz representations*. Journal of Music Theory, vol. 1, no. 41, pages 1–66, 1997.

[Dalla Bella 03] S. Dalla Bella, I. Peretz & N. Aronoff. *Time course of melody recognition: A gating paradigm study*. Perception and Psychophysics, vol. 7, no. 65, pages 1019–1028, 2003.

[Dannenberg 03] R. B. Dannenberg, W. P. Birmingham & G. Tzanetakis. *The MUSART testbed for query-by-humming evaluation*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2003.

[Davies 04] M. E. P. Davies & M. D. Plumbey. *Causal tempo tracking of audio*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), pages 164–169, 2004.

[Davies 05a] M. E. P. Davies & P. Brossier. *Beat tracking towards automatic musical accompainment*. Conv. of the Audio Engineering Society (AES), May 2005.

[Davies 05b] M. E. P. Davies & M. D. Plumbey. *Beat tracking with a two-state model*. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), vol. 3, pages 241–244, 2005.

[De Beer 06] J. De Beer & M. F. Moens. *Rpref: a generalization of bpref towards graded relevance judgments*. SIGIR'06, no. 29, pages 637–638, 2006.

[De Cheveigne 05] A. De Cheveigne. Pitch perception models. Springer-Verlag, New York, 2005.

[Dempster 77] A. P. Dempster, N. M. Laird & D. B. Rubin. *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the Royal Statistical Society, vol. 30, no. 1, pages 1–38, 1977.

[Dixon 01] S. Dixon. *Automatic extraction of tempo and beat from expressive performances*. Journal of New Music Research, vol. 1, no. 30, pages 39–58, March 2001.

[Dixon 04] S. Dixon, F. Gouyon & G. Widmer. *Towards characterization of music via rhythmic patterns*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2004.

[Dixon 05] S. Dixon & G. Widmer. *MATCH: A Music Alignment Tool Chest*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2005.

[Downie 03] J. S. Downie. *Music Information Retrieval*. Annual Review of Information Science and Technology, no. 37, 2003.

[Dressler 05] K. Dressler. *Extraction of the melody pitch contour from polyphonic audio*. MIREX extended abstract, 2005.

[Ellis 02] D. P. W. Ellis, B. Whitman, A. Berenzweig & S. Lawrence. *The quest for ground truth in musical artist similarity*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), pages 518–529, October 2002.

[Ellis 06] D. P. W. Ellis & G. E. Polliner. *Identifying cover songs with chroma features and dynamic programming beat tracking*. MIREX extended abstract, 2006.

[Ellis 07] D. P. W. Ellis & G. E. Polliner. *Identifying cover songs with chroma features and dynamic programming beat tracking*. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), April 2007. (submitted, 4pp) - See also MIREX'06 poster.

[Fingerhut 04] M. Fingerhut. *Music Information Retrieval: or how to search for (and maybe find) music and do away with incipits*. IAML-IASA Congress, August 2004.

[Foote 97] J. Foote. *Content-based retrieval of music and audio*. Proc. of SPIE Multimedia Storage and Archiving Systems II, vol. 3229, pages 138–147, 1997.

[Foote 00] J. Foote. *ARTHUR: Retrieving orchestral music by long-term structure*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), October 2000.

[Foote 02] J. Foote, M. Cooper & U. Nam. *Audio retrieval by rhythmic similarity*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2002.

[Fujishima 99] T. Fujishima. *Realtime chord recognition of musical sound: a system using common lisp music*. Int. Computer Music Conference (ICMC), pages 464–467, 1999.

[Ghias 95] A. Ghias, J. Logan, D. Chamberlin & B. C. Smith. *Query by humming: Musical Information Retrieval in an audio database*. Proc. ACM Multimedia, pages 231–236, 1995.

[Gilks 96] W. R. Gilks, S. Richardson & D. J. Spiegelhalter. Markov Chain Monte-Carlo in practice. Chapman and Hall, 1996.

[Gómez 03] E. Gómez, A. Klapuri & B. Meudic. *Melody description and extraction in the context of music content processing*. Journal of New Music Research, 2003.

[Gómez 04] E. Gómez & P. Herrera. *Estimating the tonality of polyphonic audio files: cognitive versus machine learning modelling strategies*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), pages 92–95, 2004.

[Gómez 06a] E. Gómez. *Tonal description of music audio signals*. PhD thesis, MTG, Pompeu Fabra University, Barcelona (Spain), 2006.

[Gómez 06b] E. Gómez & P. Herrera. *The song remains the same: identifying versions of the same song using tonal descriptors*. Int. Conf. on Music Information Retrieval, 2006.

[Gómez 06c] E. Gómez, B. S. Ong & P. Herrera. *Automatic tonal analysis from music summaries for version identification*. Conv. of the Audio Engineering Society (AES), October 2006.

[Goto 95] M. Goto & Y. Muraoka. *A real-time beat tracking system for audio signals*. Int. Computer Music Conference (ICMC), 1995.

[Goto 01] M. Goto & Y. Muraoka. *An audio-based real-time beat tracking system for music with or without drums*. Journal of New Music Research, vol. 2, no. 30, pages 159–171, June 2001.

[Goto 04] M. Goto. *A real-time music scene description system: predominant F0 estimation for detecting melody bass lines in real-world audio signals*. Speech Comunication, vol. 4, no. 43, pages 311–329, 2004.

[Gotoh 82] O. Gotoh. *An improved algorithm for matching biological sequences*. Journal of Molecular Biology, no. 162, pages 705–708, 1982.

[Gouyon 05] F. Gouyon. *A computational approach to rhythm description. Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing*. PhD thesis, MTG, Pompeu Fabra University, Barcelona, Spain, 2005.

[Gouyon 06] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle & P. Cano. *An experimental comparison of audio tempo induction algorithms*. IEEE Transactions on Speech and Audio Processing, vol. 14, no. 5, 2006.

[Grachten 06] M. Grachten. *Expressivity-aware tempo transformations of music performances using case based reasoning*. PhD thesis, MTG, Pompeu Fabra University, Barcelona, Spain, 2006.

[Guo 04] A. Y. Guo & H. Siegelman. *Time-warped longest common subsequence algorithm for music retrieval*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2004.

[Gusfield 97] D. Gusfield. Algorithms on strings, trees and sequences: computer sciences and computational biology. Cambridge University Press, 1997.

[Hainsworth 04] S. W. Hainsworth. *Techniques for the automated analysis of musical audio*. PhD thesis, University of Cambridge, UK, September 2004.

[Hamming 50] R. W. Hamming. *Error detecting and error correcting codes*. Bell System Technical Journal, vol. 2, no. 26, pages 147–160, 1950.

[Harte 05] C. A. Harte & M. B. Sandler. *Automatic chord identification using a quantized chromagram*. Conv. of the Audio Engineering Society (AES), pages 28–31, 2005.

[Hermes 93] D. J. Hermes. *Pitch analysis*. 1993.

[Herre 01] J. Herre, E. Allamanche & O. Helmuth. *Robust matching of audio signals using spectral flatness features*. Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pages 127–130, 2001.

[Hess 83] W. Hess. Pitch determination of speech signals. Algorithms and devices. Springer-Verlag, Berlin-Heidelberg, 1983.

[Hirschberg 75] D. S. Hirschberg. *A linear space algorithm for computing maximal common subsequences*. Communications of the ACM, no. 18, 1975.

[Hu 03] N. Hu, R. B. Dannenberg & G. Tzanetakis. *Polyphonic audio matching and alignment for music retrieval*. IEEE Workshop on Apps. of Signal Processing to Audio and Acoustics (WASPAA), 2003.

[Itakura 75] F. Itakura. *Minimum prediction residual principle applied to speech recognition*. IEEE Transactions on Acoustics, Speech and Signal Processing, no. 23, pages 52–72, 1975.

[Izmirli 05] Ö. Izmirli. *Tonal similarity from audio using a template based attractor model*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2005.

[Jaccard 12] J. Jaccard. *The distribution of the flora of the alpine zone*. New Phytologist, 1912.

[Jaro 89] M. A. Jaro. *Advances in record linking methodology as applied to the 1985 census of Tampa Florida*. Journal of the American Statistical Society, no. 64, pages 1183–1210, 1989.

[Jarvelin 00] K. Jarvelin & J. Kekalainen. *IR evaluation methods for retrieving highly relevant documents*. SIGIR'00, 2000.

[Kendall 48] M. Kendall. Rank correlation methods. Charles Griffin and Company Limited, 1948.

[Keogh 02] E. Keogh. *Exact indexing of Dynamic Time Warping*. Int. Conf. on Very Large Databases, pages 406–417, 2002.

[Klapuri 03] A. Klapuri. *Musical meter estimation and transcription*. Cambridge Music Processing Colloquium, 2003.

[Klapuri 04] A. Klapuri. *Signal processing methods for the automatic transcription of music*. PhD thesis, Tampere University of Technology, Finland, April 2004.

[Krumhansl 90] C. L. Krumhansl. Cognitive foundations of musical pitch. Oxford University Press, New York, 1990.

[Lambrou 98] T. Lambrou, P. Kudumakis, M. B. Sandler, R. Speller & A. Linney. *Classification of audio signals using statistical features on time and wavelet transform domains*. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), 1998.

[Lee 06a] K. Lee. *Automatic chord recognition using enhanced pitch class profile*. Int. Computer Music Conference (ICMC), 2006.

[Lee 06b] K. Lee. *Identifying cover songs from audio using harmonic representation*. MIREX extended abstract, 2006.

[Lee 06c] K. Lee & M. B. Sandler. *Automatic chord recognition using an HMM with supervised learning*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2006.

[Leman 91] M. Leman. *Een model van toonsemantiek: naar een theorie en discipline van de muzikale verbeelding*. PhD thesis, University of Ghent, 1991.

[Leman 95] M. Leman. *Music and schema theory: cognitive foundations of systematic musicology*. Information Science, no. 31, 1995.

[Lemstrom 00a] K. Lemstrom. *String matching techinques for music retrieval*. PhD thesis, University of Helsinki, 2000.

[Lemstrom 00b] K. Lemstrom & E. Ukkonen. *Including interval encoding into edit distance based music comparison and retrieval*. Proc. of the Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science, 2000.

[Lesaffre 05] M. Lesaffre. *Music Information Retrieval: conceptual framework, annotation and user behavior*. PhD thesis, Ghent University, Belgium, December 2005.

[Levenshtein 66] V. I. Levenshtein. *Binary codes capable of correcting deletions, insertions, and reversals*. Soviet Physics Doklady, no. 10, pages 707–710, 1966.

[Lewis 87] D. Lewis. *Generalized musical intervals and transformations*. 1987.

[Lipman 85] D. J. Lipman & W. R. Pearson. *Rapid and sensitive protein similarity searches*. Science, no. 227, pages 1435–1441, March 1985.

[Logan 00a] B. Logan. *Mel frequency cepstral coefficients for music modeling*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2000.

[Logan 00b] B. Logan & S. Chu. *Music summarization using key phrases*. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2000.

[Logan 01] B. Logan & A. Salomon. *A music similarity function based on signal analysis*. Proc. IEEE Int. Conf. on Multimedia and Expo (ICME), 2001.

[Longuet-Higgins 82] H. C. Longuet-Higgins & C. S. Lee. *The perception of musical rhythms*. Perception, vol. 2, no. 11, pages 115–128, 1982.

[Manning 07] C. D. Manning, R. Prabhakar & H. Schutze. An introduction to Information Retrieval. Cambridge University Press, Cambridge, England, preliminary draft edition, 2007. Online version at http://www.informationretrieval.org (last access: April 2007).

[Marolt 04] M. Marolt. *On finding melodic lines in audio recordings*. Proc. of the Int. Conf. on Digital Audio Effects (DAFX), 2004.

[Marolt 06] M. Marolt. *A mid-level melody-based representation for calculating audio similarity*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2006.

[McNab 96] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson & S. J. Cunningam. *Towards the digital music library: tune retrieval from acoustic input*. Proc. of the ACM Digital Libraries, pages 11–18, 1996.

[Meddis 91] R. Meddis & M. J. Hewitt. *Virtual pitch and phase sensitivity of a computer model of the auditory periphery*. Journal of the Acoustical Society of America, no. 89, pages 2866–2882, 1991.

[Müller 05] M. Müller, F. Kurth & M. Clausen. *Audio matching via chroma-based statistical features*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2005.

[Müller 06] M. Müller & F. Kurth. *Enhancing similarity matrices for music audio analysis*. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2006.

[Myers 80a] C. Myers. A comparative study of several Dynamic Time Warping algorithms for speech recognition. Master's thesis, Massachussets Institute of Technology (MIT), USA, 1980.

[Myers 80b] C. Myers, L. R. Rabiner & A. E. Rosenberg. *Performance tradeoffs in Dynamic Time Warping algorithms for isolated word recognition*. IEEE Transactions on Audio, Speech and Language processing, no. 6, 1980.

[Navarro 05] G. Navarro, V. Mäkinen & E. Ukkonen. *Algorithms for transposition invariant string matching*. Journal of Algorithms, no. 56, 2005.

[Needleman 70] S. B. Needleman & C. D. Wunsch. *A general method applicable to the search for similarities in the amino acid sequences of two proteins*. Journal of Molecular Biology, no. 48, pages 443–453, 1970.

[Noll 67] A. M. Noll. *Cepstrum pitch determination*. Journal of the Acoustical Society of America, no. 41, pages 293–309, 1967.

[Ong 07] B. S. Ong. *Structural analysis and segmentation of music signals*. PhD thesis, MTG, Pompeu Fabra University, Barcelona, Spain, 2007.

[Oppenheim 69] A. V. Oppenheim. *A speech analysis-synthesis system based on homomorphic filtering*. Journal of the Acoustical Society of America, no. 45, pages 458–465, 1969.

[Oppenheim 99] A. V. Oppenheim, R. W. Schafer & J. B. Buck. Discrete-Time Signal Processing. Prentice Hall, 2 edition, February 1999.

[Orio 06] N. Orio. *Music retrieval: a tutorial and review*. Foundations and Trends in Information Retrieval, vol. 1, no. 1, pages 1–90, 2006.

[Owen 00] H. Owen. Music Theory Resource Book. Oxford University Press, 2000.

[Paiva 04] R. P. Paiva, T. Mendes & A. Cardoso. *A methodology for detection of melody in polyphonic signals*. Conv. of the Audio Engineering Society (AES), 2004.

[Pampalk 03] E. Pampalk, S. Dixon & G. Widmer. *On the evaluation of perceptual similarity measures for music*. Proc. of the Int. Conf. on Digital Audio Effects (DAFX), 2003.

[Pampalk 05] E. Pampalk, A. Flexer & G. Widmer. *Improvements of audio-based music similarity and genre classification*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), pages 628–633, 2005.

[Pampalk 06] E. Pampalk. *Computational models of music similarity and their application to Music Information Retrieval*. PhD thesis, Vienna University of Technology, Austria, March 2006.

[Paulus 02] J. Paulus & A. Klapuri. *Measuring the similarity of rhythmic patterns*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2002.

[Paws 04] S. Paws. *Musical key extraction from audio*. Int. Conf. on Music Information Retrieval, 2004.

[Pearson 91] W. R. Pearson. *Comparison methods for searching protein sequences databases*. Protein Science, no. 4, pages 1145–1160, 1991.

[Polliner 05] G. E. Polliner & D. P. W. Ellis. *A classification approach to melody transcription*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2005.

[Polliner 07] G. E. Polliner, D. P. W. Ellis, A. Ehmann, E. Gómez, S. Streich & B. S. Ong. *Melody transcription from music audio: approaches and evaluation*. IEEE Transactions on Audio, Speech and Language processing, 2007.

[Purwins 00] H. Purwins, B. Blankertz & K. Obermayer. *A new method for tracking modulations in tonal music in audio data format*. Neural Networks (IJCNN), no. 6, pages 270–275, 2000.

[Purwins 05] H. Purwins. *Proles of pitch classes. Circularity of relative pitch and key: experiments, models, computational music analysis, and perspectives.* PhD thesis, Berlin University of Technology, Germany, 2005.

[Rabiner 89] L. R. Rabiner. *A tutorial on Hidden Markov Models and selected applications in speech recognition*. Proc. of the IEEE, 1989.

[Rabiner 93] L. R. Rabiner & B. H. Juang. Fundamental of speech recognition. Prentice, Englewood Cliffs, NJ, 1993.

[Raphael 01a] C. Raphael. *Automatic rhythm transcription*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), pages 99–107, 2001.

[Raphael 01b] C. Raphael. *Music plus one: a system for expressive and flexible music acconpainment*. Int. Computer Music Conference (ICMC), 2001.

[Raphael 03] C. Raphael & J. Stoddard. *Harmonic analysis with probabilistic graphical models*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), pages 177–181, October 2003.

[Ratanamahatana 04a] C. Ratanamahatana & E. Keogh. *Everything you know about Dynamic Time Warping is wrong*. Workshop on Mining Temporal and Sequential Data, in conjunction with the Tenth ACM SIGKDD, August 2004.

[Ratanamahatana 04b] C. Ratanamahatana & E. Keogh. *Making time-series classification more accurate using learned constraints*. SIAM International Conference on Data Mining (SDM), pages 11–22, April 2004.

[Rosenthal 94] D. Rosenthal, M. Goto & Y. Muraoka. *Rhythm tracking using multiple hypothesis*. Int. Computer Music Conference (ICMC), pages 85–87, 1994.

[Ryynanen 05] M. P. Ryynanen & A. Klapuri. *Polyphonic music transcription using note-event modeling*. IEEE Workshop on Apps. of Signal Processing to Audio and Acoustics (WASPAA), 2005.

[Sailer 06] C. Sailer & K. Dressler. *Finding cover songs by melodic similarity*. MIREX extended abstract, 2006.

[Sakoe 78] H. Sakoe & S. Chiba. *Dynamic Programming algorithm optimisation for spoken word recognition*. IEEE Transactions on Acoustics, Speech and Signal Processing, no. 26, pages 43–49, 1978.

[Sankoff 83] D. Sankoff & J. Kruskal. Time warps, string edits, and macromolecules. Addison-Wesley, New York, 1983.

[Saracevic 75] T. Saracevic. *Relevance: a review of and a framework for the thinking on the notion in information science*. Journal of the American Society for Information Science, no. 26, pages 321–343, 1975.

[Saracevic 06] T. Saracevic. *Relevance: a review of the literature and a framework for thinking on the notion in Information Science*. Advances in librarianship, no. 30, pages 3–71, 2006.

[Scheirer 98] E. D. Scheirer. *Tempo and beat analysis of acoustic musical signals*. Journal of the Acoustical Society of America, vol. 1, no. 103, pages 588–601, 1998.

[Schulkind 03] M. D. Schulkind, R. J. Posner & 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, pages 217–249, 2003.

[Sellers 74] P. H. Sellers. *On the theory and computation of evolutionary distances*. SIAM Journal on Applied Mathematics, no. 26, pages 787–793, 1974.

[Serrà 07] J. Serrà. *A qualitative assessment of measures for the evaluation of a cover song identification system*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), September 2007.

[Shao 04] X. Shao, X. Changsheng & M. S. Kankanhalli. *Unsupervised classification of music genre using Hidden Markov Model*. IEEE Int. Conf. on Multimedia and Expo, vol. 3, June 2004.

[Sheh 03] A. Sheh & D. P. W. Ellis. *Chord segmentation and recognition using EM-trained Hidden Markov Models*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2003.

[Shepard 82] R. N. Shepard. *Structural representations of musical pitch*. The Psychology of Music, 1982.

[Smith 81] T. F. Smith & M. S. Waterman. *Identification of common molecular subsequences*. Journal of Molecular Biology, no. 147, pages 195–197, 1981.

[Snedecor 89] G. W. Snedecor. Statistical methods. Blackwell Publishing Limited, 8 edition, August 1989.

[Stevens 37] S. S. Stevens, J. Volkmann & E. B. Newman. *A scale for the measurement of the psychological magnitude pitch*. Journal of the Acoustical Society of America, pages 185–190, January 1937.

[Temperley 99] D. Temperley & D. Sleator. *Modeling meter and harmony: a preference-rule approach*. Computer Music Journal, no. 23, pages 10–27, 1999.

[Typke 04] R. Typke, F. Wiering & R. C. Veltkamp. *A search method for notated polyphonic music with pitch and tempo fluctuations*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2004.

[Tzanetakis 01] G. Tzanetakis, G. Essl & P. Cook. *Automatic music genre classification of audio signals*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2001.

[Tzanetakis 02a] G. Tzanetakis. *Pitch histograms in audio and symbolic Music Information Retrieval*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2002.

[Tzanetakis 02b] G. Tzanetakis & P. Cook. *Musical genre classification of audio signals*. IEEE Transactions on Speech and Audio Processing, vol. 5, no. 10, pages 293–302, 2002.

[Ukkonen 03] E. Ukkonen, K. Lemstrom & V. Mäkinen. *Sweepline the music!* Comp. Sci. in Perspective, pages 330–342, 2003.

[Venkatachalam 04] V. Venkatachalam, L. Cazzanti, N. Dhillon & M. Wells. *Automatic identification of sound recordings*. IEEE Signal Processing Magazine, 2004.

[Vignoli 05] F Vignoli & S. Paws. *A music retrieval system based on user-driven similarity and its evaluation*. Proc. Int. Symposium on Music Information Retrieval (ISMIR), 2005.

[Vincent 05] E. Vincent & M. D. Plumbey. *Predominant F0 estimation using Bayesian harmonic waveform models*. MIREX extended abstract, 2005.

[Vingron 94] M. Vingron & M. S. Waterman. *Sequence alignment and penalty choice*. Journal of Molecular Biology, no. 235, pages 1–12, 1994.

[Vlachos 06] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos & E. Keogh. *Indexing multidimensional sime-series*. Very Large Databases Journal, no. 15, pages 1–20, July 2006.

[Voorhees 06] E. M. Voorhees & L. P. Buckland. *Common evaluation measures*. Proc. of Text Retrieval Conference, 2006. Appendix.

[Wagner 74] R. A. Wagner & M. J. Fischer. *The string-to-string correction problem*. Journal of the ACM, 1974.

[Waterman 76] M. S. Waterman, T. F. Smith & W. A. Beyer. *Some biological sequence metrics*. Advances in Mathematics, vol. 20, no. 3, June 1976.

[Waterman 87a] M. S. Waterman & M. Eggert. *A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons*. Journal of Molecular Biology, no. 197, pages 723–728, 1987.

[Waterman 87b] M. S. Waterman, L. Gordon & R. Arratia. *Phase transitions in sequence matches and nucleic acid structure*. Proc. of the National Academy of Sciences, vol. 84, pages 1239–1243, March 1987.

[Waterman 94a] M. S. Waterman. *Estimating statistical significance of sequence alignments*. Phil. Trans. of the Royal Society, no. 344, pages 383–390, 1994. Printed in Great Britain.

[Waterman 94b] M. S. Waterman & M. Vingron. *Rapid and accurate estimates of statistical significance for sequence data base searches*. Proc. of the Nat. Academy of Sciences, vol. 91, pages 4625–4628, May 1994.

[Wrinckler 99] W. E. Wrinckler. *The state of record linkage and current research problems*. Statistics of Income Division, Internal Revenue Service Publication, 1999.

[Yang 01] C. Yang. *Music database retrieval based on spectral similarity*. Stanford University Database Group Technical Report, 2001.

[Yang 06] L. Yang. *Distance metric learning: a comprehensive survey*. Technical report., 2006.

[Ye 03] N. Ye. The handbook of Data Mining. Lawrence Erlbaum Associates, 2003.

[Young 00] S. Young. The HTK book. July 2000. (for HTK version 3).

[Zheng 01] F. Zheng, G. Zhang & Z. Song. *Comparison of different implementations of MFCC*. Computer Science and Technology, vol. 6, no. 16, pages 582–589, September 2001.

Appendix A: Music Collection

It is the objective of this appendix to provide a list of the songs used in our tests. As we have said, these can be divided in two main groups: covers and outliers. For the first one, songs are grouped in their respective *cover sets*.

Covers

Song title: "One hundred years". Original artist: The Cure. Covered by: The Cure (Demo), Nosferatu. Database: DB2053.
Song title: "10:15 saturday night". Original artist: The Cure. Covered by: Candy Machine, The Cure (Demo). Database: DB2053.
Song title: "Seventeen seconds". Original artist: The Cure. Covered by: Lt No, Whispers in shadows. Database: DB2053.
Song title: "Dos gardenias". Original artist: Buena vista social club. Covered by: Ana Maria Gonzlez, Buena vista social club, Parrita and Lolita, Unknown. Database: DB2053.
Song title: "Six different ways". Original artist: The Cure. Covered by: Chisel. Database: DB2053.
Song title: "For absent friends". Original artist: Genesis. Covered by: Genesis (Remaster). Database: DB2053.
Song title: "Across the Universe". Original artist: Beatles. Covered by: Rufus Wainwright, Beatles (alternative version), Selway and Morpho Eugenia, Fiona Apple, Lydia, Yo la tengo, Roger Waters, Suede, Beatles (Remaster). Database: DB2053.
Song title: "A Day In The Life". Original artist: Beatles. Covered by: Phish (Live), Les Demerle (Instrumental), Chocolate snow (Instrumental), Frankie Valli, Jeff Beck (Instrumental), Sting (Live—Acoustic). Database: DB2053.
Song title: "Adios nonino". Original artist: Astor Piazzolla. Covered by: Quinteto da Paraiba (Instrumental). Database: DB2053.
Song title: "A forest". Original artist: The Cure. Covered by: Kriegsbereit (Remix), Carpathian forest, Ganymede, Madaski (Remix), Creaming. Jesus, Josh Rouse (Live—Acoustic), One, Death lies bleeding, Blank and Jones. (Remix), Children within, Dionysos, Floodland, Krauts, Rasca y pica, The Cure. (Acoustic), The Cure (Demo), Waltari. Database: DB2053.
Song title: "Agua de beber". Original artist: Astrud Gilberto. Covered by: Anna Lucia, Antonio Carlos Jobim, Astrud Gilberto (version 2), Astrud Gilberto. (version 3), Jobim Vinicius and Toquinho. Database: DB2053.
Song title: "A letter to Elise". Original artist: The Cure. Covered by: Blink 182, Kidsgofree. Database: DB2053.
Song title: "All by myself". Original artist: Eric Carmen. Covered by: Jamie O Neal, Tom Jones. Database: DB2053.
Song title: "All cats are grey". Original artist: The Cure. Covered by: Rollercoaster. Database: DB2053.
Song title: "All My Loving". Original artist: Beatles. Covered by: Los Manolos, Ricky Gianco, The 52 key organ. Database: DB2053.
Song title: "All You Need Is Love". Original artist: Beatles. Covered by: I Soliti Ignoti, London Symphony Orchestra, Neil Young, Oasis. Database: DB2053.
Song title: "Amazonas". Original artist: Joao Donato. Covered by: Lisa Ono. Database: DB2053.
Song title: "Amelitango". Original artist: Astor Piazzolla. Covered by: Astor Piazzolla (Remaster). Database: DB2053.
Song title: "And I Love Her". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Beatles (Remaster), Byron Lee and The. Dragonaires, Diana Krall, John Denver, John Pizzarelli, Lorenzo Santamaria, Massiel, Smokey Robinson and The Miracles, The 52 key organ. Database: DB2053.
Song title: "And Your Bird Can Sing". Original artist: Beatles. Covered by: The Jam. Database: DB2053.
Song title: "Cuando los angeles lloran". Original artist: Mana. Covered by: Miguel Rios. Database: DB2053.
Song title: "Angie". Original artist: Rolling Stones. Covered by: Bossa nova, Melua and Kubb, Pearl Jam, Rolling Stones (Acoustic), Stereophonics, Tori Amos. Database: DB2053.
Song title: "A night like this". Original artist: The Cure. Covered by: Hedtrip, Peter Hayes, Piano tribute to The Cure, Test Infection. Database: DB2053.
Song title: "Antenne". Original artist: Kraftwerk. Covered by: Psyche. Database: DB2053.
Song title: "Any colour you like". Original artist: Pink Floyd. Covered by: Pink Floyd (Remix), String Quartet, VVAA. Database: DB2053.
Song title: "Any dream will do". Original artist: Andrew Lloyd Webber. Covered by: The best of musicals. Database: DB2053.
Song title: "Anything goes". Original artist: Cole Porter. Covered by: Ella Fitzgerald (version 2), Ella Fitzgerald (version 3), Ella Fitzgerald. and Cole Porter. Database: DB2053.
Song title: "Astronomy domine". Original artist: Pink Floyd. Covered by: Pink Floyd (Live), Pink Floyd (Remaster). Database: DB2053.
Song title: "Take the A train". Original artist: Billy Strayhorn. Covered by: Duke Ellington (version 1), Duke Ellington (version 2), Duke Ellington. (version 3), Kei Kobayashi, Richard Davis, Woody Allen movie music. Database: DB2053.
Song title: "Autobahn". Original artist: Kraftwerk. Covered by: Buffalo Daughter, Kraftwerk (Remix), The Balanescu Quartet, Tragic comedy. Database: DB2053.
Song title: "Autumn in New York". Original artist: Chet Baker. Covered by: Kei Kobayahsi, Unknown. Database: DB2053.
Song title: "Back in the USSR". Original artist: Beatles. Covered by: Chriss and The Stroke, La orquesta Modragon. Database: DB2053.
Song title: "Bad to me". Original artist: Beatles. Covered by: Billy Kramer and The Dakotas. Database: DB2053.
Song title: "The battle of epping forest". Original artist: Genesis. Covered by: Genesis (version 2), Genesis (version 3), Genesis (Instrumental), Genesis. (version 4), Genesis (version 5), Genesis (Instrumental 2), Genesis. (Instrumental 3), Genesis (Instrumental 4), Genesis

(version6). Database: DB2053.
Song title: "Because". Original artist: Beatles. Covered by: Lynsey De Paul, Mystiquintet, Satelite Kingston. Database: DB2053.
Song title: "Begin the Beguine". Original artist: Cole Porter. Covered by: Bobby Morganstein, Ella Fitzgerald, Frank Sinatra, Glenn Miller, John Williams. and The Boston Pops, The best of musicals. Database: DB2053.
Song title: "In the beginning". Original artist: Genesis. Covered by: Mother Gong. Database: DB2053.
Song title: "Berimbau". Original artist: Vinicius de Moraes. Covered by: Astrud Gilberto, Sergio Mendes trio, Tamba trio. Database: DB2053.
Song title: "Billy Jean". Original artist: Michael Jackson. Covered by: Disco Galaxy (Remix), Eminem (Remix), Sisqo (Remix), Social Distorsion. Database: DB2053.
Song title: "Bim bom". Original artist: Joao Gilberto. Covered by: Astrud Gilberto (version 1), Astrud Gilberto (version 2), Mann Gilberto and. Jobim, Sergio Mendes, Stan Getz. Database: DB2053.
Song title: "Birth of the blues". Original artist: Frank Sinatra. Covered by: Johnny Hartman (version 1), Johnny Hartman (version 2). Database: DB2053.
Song title: "Black Bird". Original artist: Beatles. Covered by: Bonnie Pink, Bossa Rio, CSNY, Eros, Grateful dead, Los angeles, Rosalyn and. The Paragons, Sarah McLachlan. Database: DB2053.
Song title: "Black celebration". Original artist: Depeche Mode. Covered by: Monster magnet. Database: DB2053.
Song title: "Blasphemous rumors". Original artist: Depeche Mode. Covered by: Seega, Sexy Sadie, Sylvain Chauveau and Ensemble Nocturne. Database: DB2053.
Song title: "Blue Moon". Original artist: Billie Holiday. Covered by: Adrivalan Orchestra, Kenny Barron Trio. Database: DB2053.
Song title: "Bohemian Rapsody". Original artist: Queen. Covered by: Braids (Remix), Elton John and Axel Rose, Fugees, Guns N Roses, Lauryn Hill. and Fugees, Molotov, Rammstein. Database: DB2053.
Song title: "Born to be wild". Original artist: Steppenwolf. Covered by: AC-DC, Creedence Clearwater Revival, Muppets and Ozzy Osburne, Wilson Picket. Database: DB2053.
Song title: "Boys dont cry". Original artist: The Cure. Covered by: Another tale, CPM 22, David Ari Leon, Happy pills, Idlewild (Live), Lostprophets, Ninos mutantes, Obnoxious, Oleander (Live), Prozac plus, Reel big fish (Live), Sheer Terror, Sitar M, String Quartet, Terminal choice, The Cure (Acoustic), Tuscadero. Database: DB2053.
Song title: "Brain damage". Original artist: Pink Floyd. Covered by: London Philharmonic Orchestra, Piano tribute to Pink Floyd, Pink Floyd (Live), Pink Floyd (Remix), String Quartet. Database: DB2053.
Song title: "Breathe". Original artist: Pink Floyd. Covered by: London Philharmonic Orchestra, Pink Floyd (Live 1), Pink Floyd (Live 2). Database: DB2053.
Song title: "Bring the boys back home". Original artist: Pink Floyd. Covered by: Sherwood and Schellen. Database: DB2053.
Song title: "Broadway melody of 1974". Original artist: Genesis. Covered by: Controlled bleeding. Database: DB2053.
Song title: "Buenos Aires hora cero". Original artist: Astor Piazzolla. Covered by: Los tangueros. Database: DB2053.
Song title: "Bungalow bill". Original artist: Beatles. Covered by: Phish. Database: DB2053.
Song title: "Cafetin de Buenos Aires". Original artist: Astor Piazzolla. Covered by: Enrique Dumas. Database: DB2053.
Song title: "Candela". Original artist: Celia Cruz. Covered by: Buena Vista Social Club, Ibrahim Ferrer. Database: DB2053.
Song title: "Cant Buy Me Love". Original artist: Beatles. Covered by: John Pizzarelli, Los Idolos, Mocedades, The Kings singers. Database: DB2053.
Song title: "Can utility and the coastliners". Original artist: Genesis. Covered by: Band X II. Database: DB2053.
Song title: "Careful with that axe Eugene". Original artist: Pink Floyd. Covered by: Nik Turner. Database: DB2053.
Song title: "The carpet crawlers". Original artist: Genesis. Covered by: John Ford. Database: DB2053.
Song title: "Carry That Weight". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Dobby Dobson, The Bee Gees (2in1). Database: DB2053.
Song title: "Catch". Original artist: The Cure. Covered by: Eggs. Database: DB2053.
Song title: "Charlotte sometimes". Original artist: The Cure. Covered by: Madee, Psyche, Shamrain, Trampoline, Yell. Database: DB2053.
Song title: "Chega de saudade". Original artist: Antonio Carlos Jobim. Covered by: Gilberto Pereira de Oliveira, Jane Monheit, Joao Gilberto, Stan Getz. Database: DB2053.
Song title: "Chica de ayer". Original artist: Antonio Vega. Covered by: Antonio Vega basico, Julio Iglesias. Database: DB2053.
Song title: "Cinema Paradiso Love theme". Original artist: Ennio Morricone. Covered by: Henry Mancini Orchestra, Josh Groban, Monica Mancini, Pat Metheny. Database: DB2053.
Song title: "The cinema show". Original artist: Genesis. Covered by: The Flower Kings. Database: DB2053.
Song title: "Close to me". Original artist: The Cure. Covered by: Dismemberment plan, The Cure (Demo), The Cure (Acoustic). Database: DB2053.
Song title: "Come in number 51 your time is up". Original artist: Pink Floyd. Covered by: Pink Floyd (Film version). Database: DB2053.
Song title: "Come Together". Original artist: Beatles. Covered by: Alika and Nueva alianza, Azade Abi and more, Desmond Dekker and The Israelites, Diana Ross, Elton John (Live), Grateful dead, GunsNRoses (Live), Ike and Tina. Turner, Johnny Jones, Pushin, Richard groove Holmes, Soundgarden, The Supremes. Database: DB2053.
Song title: "Comfortably numb". Original artist: Pink Floyd. Covered by: Chillout tribute to PF, London Philharmonic Orchestra, Piano tribute to Pink. Floyd, Pink Floyd (Demo), Squire and White, VVAA. Database: DB2053.
Song title: "Computer love". Original artist: Kraftwerk. Covered by: Kraftwerk (Remix), Kraftwerk, Laura effect, Teruo Nakano, The Balanescu Quartet. Database: DB2053.
Song title: "Corcovado". Original artist: Antonio Carlos Jobim. Covered by: Astrud Gilberto (version 1), Astrud Gilberto (version 2), Astrud Gilberto. (version 3), Jackie and Roy, Joao Gilberto (version 1), Joao Gilberto (version 2), Jobim and Morais, Paquito D Rivera, Stan Getz (version 1), Stan Getz (version 2). Database: DB2053.
Song title: "Crumbling land". Original artist: Pink Floyd. Covered by: Pink Floyd (Film version), Pink Floyd (Remix). Database: DB2053.
Song title: "I have got a crush on you". Original artist: Frank Sinatra. Covered by: Barbara Streisand, Frank Sinatra 2, Stacey Kent.

Database: DB2053.
Song title: "Dont cry for me Argentina". Original artist: Andrew Lloyd Webber. Covered by: Olivia Newton John, The best of musicals. Database: DB2053.
Song title: "Ill cry instead". Original artist: Beatles. Covered by: Billy Joel. Database: DB2053.
Song title: "El cuarto de Tula". Original artist: Compay Segundo. Covered by: Buena Vista Social Club, Eliades Ochoa. Database: DB2053.
Song title: "Cucurrucucu Paloma". Original artist: Caetano Veloso. Covered by: Unkown. Database: DB2053.
Song title: "Dancing with the Moonlit knight". Original artist: Genesis. Covered by: Darxtar. Database: DB2053.
Song title: "So danco samba". Original artist: Antonio Carlos Jobim. Covered by: Antonio Carlos Jobim (version 2), Antonio Carlos Jobim and Vinicius de Morais, Milt Jackson, Sergio Mendes and Brasil, Stan Getz, Stan Getz and Gilberto Gil, Tamba trio. Database: DB2053.
Song title: "Das modell". Original artist: Kraftwerk. Covered by: Hikashu, Index, Kraftwerk (version 2), Nigra nebula, The Balanescu Quartet. Database: DB2053.
Song title: "Day Tripper". Original artist: Beatles. Covered by: Cleaners From Venus, Domain, Jimi Hendrix, Ocean Colour Scene, Sergio Mendes. and Brazil66, Sublime. Database: DB2053.
Song title: "Deaths door". Original artist: Depeche Mode. Covered by: Sylvain Chauveau and Ensemble Nocturne. Database: DB2053.
Song title: "Decarissimo". Original artist: Astor Piazzolla. Covered by: Los tangueros, Sergio and O Assad. Database: DB2053.
Song title: "Same deep water as you". Original artist: The Cure. Covered by: Another nothing. Database: DB2053.
Song title: "Se dejaba llevar". Original artist: Antonio Vega. Covered by: Antonio Vega basico. Database: DB2053.
Song title: "Dentaku". Original artist: Kraftwerk. Covered by: Kraftwerk (Remix), Kraftwerk (version 2), Satoru wono. Database: DB2053.
Song title: "Der telefon anruf". Original artist: Kraftwerk. Covered by: The Shining, Welle-Erdbal. Database: DB2053.
Song title: "Desafinado". Original artist: Antonio Carlos Jobim. Covered by: Gal Costa, Herbie Mann and Joao Gilberto, Joao Gilberto (version 1), Joao. Gilberto (version 2), Mann Gilberto and Jobim live, Nogueira and Jobim, Paquito. D Rivera, Stan Getz. Database: DB2053.
Song title: "Die roboter". Original artist: Kraftwerk. Covered by: Kraftwerk (version 2), Kraftwerk (version 3), Kraftwerk (Remix), Kraftwerk. (Remix 2), Kraftwerk (version 4), Kraftwerk (Remix 3), POD, Senor Coconuts, The Balanescu Quartet. Database: DB2053.
Song title: "Disintegration". Original artist: The Cure. Covered by: Converge, Decadence, Razed in black. Database: DB2053.
Song title: "Why Dont We Do It In The Road". Original artist: Beatles. Covered by: Fred James, Grateful dead. Database: DB2053.
Song title: "Dont leave me now". Original artist: Pink Floyd. Covered by: Shaw and Krieger. Database: DB2053.
Song title: "Dont let go". Original artist: Manhattan Transfer. Covered by: Tom Jones. Database: DB2053.
Song title: "Dont let me down". Original artist: Beatles. Covered by: Beatles and Billy Preston, Charles Walker, Doug Dillard and Gene Clarke, Harry J all stars, Ks Choice and Novastar, Marcia Griffiths, Matchbox 20, Planta. and Raiz, Stereophonics. Database: DB2053.
Song title: "I dreamed a dream". Original artist: Les miserables. Covered by: The best of musicals. Database: DB2053.
Song title: "Drive My Car". Original artist: Beatles. Covered by: Bobby McFerrin, RADD, Rainforest concert, Takako Minekawa, Tony Ronald. Database: DB2053.
Song title: "Echoes". Original artist: Pink Floyd. Covered by: Alien sex fiend, Pink Floyd (Live), Pink Floyd (Remix). Database: DB2053.
Song title: "Eight Days a Week". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, The mirza men. Database: DB2053.
Song title: "Eleanor Rigby". Original artist: Beatles. Covered by: Antidoping (Remix), B B Seaton, Caetano Veloso, Derek Enright MP, Godhead, Grateful dead (Live), John Pizzarelli, Kansas, Kings Singers, Mystiquintet, Nonpalidece, Rare earth, The four stops, Thrice, Vanilla Fudge, Wes Montgomery. Database: DB2053.
Song title: "Electric cafe". Original artist: Kraftwerk. Covered by: Kraftwerk (version 2), Xingu hill. Database: DB2053.
Song title: "E luxo so". Original artist: Joao Gilberto. Covered by: Getz and Gilberto, Lisa Ono, Stan Getz. Database: DB2053.
Song title: "Empty spaces". Original artist: Pink Floyd. Covered by: Sherwood and Krieger. Database: DB2053.
Song title: "Endless love". Original artist: Diana Ross and Lionel Richie. Covered by: Diana Ross, Tom Jones. Database: DB2053.
Song title: "The end of the world". Original artist: The Cure. Covered by: Piano tribute to The Cure. Database: DB2053.
Song title: "Enjoy the silence". Original artist: Depeche Mode. Covered by: Apoptygma Berzerk (Live), Depeche Mode (Remix 1), Depeche Mode (Remix 2), Depeche Mode (Remix 3), Depeche Mode (Remix 4), Depeche Mode (Remix 5), Depeche. Mode (Instrumental), Failure, HIM, Ninos mutantes, Sylvain Chauveau and Ensemble. Nocturne (version 1), Sylvain Chauveau and Ensemble Nocturne (version 2), Symphonic tribute to Depeche Mode, Talla 2XLC. Database: DB2053.
Song title: "Esperanza perdida". Original artist: Joao Gilberto. Covered by: Joao Donato, Joao Gilberto (version 2). Database: DB2053.
Song title: "Europe endless". Original artist: Kraftwerk. Covered by: Makoto Inoue. Database: DB2053.
Song title: "Everything counts". Original artist: Depeche Mode. Covered by: Deluxe, Meat Beat Manifesto, Soil and Eclipse, Yendri. Database: DB2053.
Song title: "The exploding boy". Original artist: The Cure. Covered by: Alkaline trio. Database: DB2053.
Song title: "Fascination street". Original artist: The Cure. Covered by: Godhead. Database: DB2053.
Song title: "Favela". Original artist: Sergio Mendes trio. Covered by: Antonio Carlos Jobim. Database: DB2053.
Song title: "I feel fine". Original artist: Beatles. Covered by: Nirvana, The Parasites. Database: DB2053.
Song title: "Fever". Original artist: Kylie Minogue. Covered by: Kylie Minogue (Live). Database: DB2053.
Song title: "The figurehead". Original artist: The Cure. Covered by: Acuarela, Escape, Sr Chinarro, The Cure (Demo). Database: DB2053.
Song title: "Fire in Cairo". Original artist: The Cure. Covered by: The Cure (Demo). Database: DB2053.
Song title: "Fire and rain". Original artist: James Taylor. Covered by: James Taylor (version 2). Database: DB2053.
Song title: "Firth of fifth". Original artist: Genesis. Covered by: Steve Hackett. Database: DB2053.
Song title: "Flaming". Original artist: Pink Floyd. Covered by: Pink Floyd (Live), Pink Floyd (Remaster). Database: DB2053.
Song title: "Fly on the windscreen". Original artist: Depeche Mode. Covered by: God lives underwater, Symphonic tribute to Depeche

Mode. Database: DB2053.
Song title: "Forever". Original artist: The Cure. Covered by: The Cure (Live). Database: DB2053.
Song title: "For No One". Original artist: Beatles. Covered by: Caetano Veloso, John Pizzarelli. Database: DB2053.
Song title: "Fountain of Salmacis". Original artist: Genesis. Covered by: Steve Hackett. Database: DB2053.
Song title: "Freelove". Original artist: Depeche Mode. Covered by: Begona, Sylvain Chauveau and Ensemble Nocturne. Database: DB2053.
Song title: "Friday Im in love". Original artist: The Cure. Covered by: David Ari Leon, Glo-Worm, Jet lag, Space rock revolution (Remix). Database: DB2053.
Song title: "From me to you". Original artist: Beatles. Covered by: Earl Green, Emi Bonilla, Fidel Nadal and Holy Piby. Database: DB2053.
Song title: "Fuga y misterio". Original artist: Astor Piazzolla. Covered by: Los Angeles Guitar Quartet, Los tangueros, Quinteto da Paraiba. Database: DB2053.
Song title: "Geiger counter". Original artist: Kraftwerk. Covered by: Axiome. Database: DB2053.
Song title: "Georgia on my mind". Original artist: Ray Charles. Covered by: Billie Holiday, Glenn Miller, James Brown, Ray Charles (Live), Unknown. Database: DB2053.
Song title: "Get Back". Original artist: Beatles. Covered by: Anonymously yours, Beatles (Remaster), Beatles and Billy Preston, Bon Jovi. and Van Halen (Live), Grateful dead, John Pizzarelli, Juniors, Paul Lamb, Rod Stewart, Shirley Scott. Database: DB2053.
Song title: "Get the balance right". Original artist: Depeche Mode. Covered by: Depeche Mode (Remix 1), Depeche Mode (Remix 2), Scaras. Database: DB2053.
Song title: "Getting Better". Original artist: Beatles. Covered by: Jeffrey Osbourne, Status Quo. Database: DB2053.
Song title: "The return of the giant Hogweed". Original artist: Genesis. Covered by: Spirits Burning. Database: DB2053.
Song title: "Gimme gimme gimme". Original artist: ABBA. Covered by: Sisters of mercy (Live), Yngwie Malmsteen, ABBA and Paul Johnson (Remix), Madonna (Remix), Madonna (Remix), VVAA, A-Teens, ABBA (Remix). Database: DB2053.
Song title: "Girl". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Peppino di Capri, St Louis Union. Database: DB2053.
Song title: "Give peace a chance". Original artist: Beatles. Covered by: The Maytals. Database: DB2053.
Song title: "Glass Onion". Original artist: Beatles. Covered by: Phish. Database: DB2053.
Song title: "Golden Slumbers". Original artist: Beatles. Covered by: Ben Folds, Eva Cassidy and Jackson Browne, The Bee Gees (2in1). Database: DB2053.
Song title: "Goodbye blue sky". Original artist: Pink Floyd. Covered by: Howe and Sherwood, Piano tribute to Pink Floyd. Database: DB2053.
Song title: "Goodbye cruel world". Original artist: Pink Floyd. Covered by: Sherwood and Levin. Database: DB2053.
Song title: "Good Day Sunshine". Original artist: Beatles. Covered by: The Tremeloes. Database: DB2053.
Song title: "Good Night". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, The moog beatles. Database: DB2053.
Song title: "Ive Got A Feeling". Original artist: Beatles. Covered by: Pearl Jam. Database: DB2053.
Song title: "Lets go to bed". Original artist: The Cure. Covered by: Crocodile shop, Dead Sexy Inc. Database: DB2053.
Song title: "Got To Get You Into My Life". Original artist: Beatles. Covered by: Cliff Bennett and The Rebel R, The four stops. Database: DB2053.
Song title: "The great gig in the sky". Original artist: Pink Floyd. Covered by: Chillout tribute to PF, London Philharmonic Orchestra, Pink Floyd (Remix), Pink Floyd (Live), String Quartet. Database: DB2053.
Song title: "The great pretender". Original artist: The Platters. Covered by: Queen. Database: DB2053.
Song title: "The hanging garden". Original artist: The Cure. Covered by: Afi, FGFC820 and Rexx Arkana, Mignight configuration, Moksha, Stone 588, Technova. Database: DB2053.
Song title: "The happiest days of out lives". Original artist: Pink Floyd. Covered by: Chillout tribute to PF, Sherwood and Colaiuta. Database: DB2053.
Song title: "Happiness is a Warm Gun". Original artist: Beatles. Covered by: Dream Theater, Sexy Sadie, The Breeders, U2. Database: DB2053.
Song title: "A Hard Days Night". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Beatles (Remaster), Billy Joel (Live), Diana Ross. and The Supremes, John Lennon, Pat Kelly, Ramsey Lewis, Shameless. Database: DB2053.
Song title: "Have a cigar". Original artist: Pink Floyd. Covered by: Piano tribute to Pink Floyd, VVAA. Database: DB2053.
Song title: "To have and to hold". Original artist: Depeche Mode. Covered by: Deftones, Deftones (Live). Database: DB2053.
Song title: "Heart beat pig meat". Original artist: Pink Floyd. Covered by: Pink Floyd (version 2). Database: DB2053.
Song title: "Hello Goodbye". Original artist: Beatles. Covered by: Bit-Nik, Don Carlos, Los modulos, Soulful strings. Database: DB2053.
Song title: "Help". Original artist: Beatles. Covered by: Bananarama, Baroque Chamber Orchestra, Henry Gross, Howie Day, Irvins 89 key. organ, Los mustang, The pug must die. Database: DB2053.
Song title: "Help me make it through the night". Original artist: Kris Kristofferson. Covered by: Henry Mancini, Joan Baez, Tom Jones. Database: DB2053.
Song title: "Helter Skelter". Original artist: Beatles. Covered by: Aerosmith, Husker Du, Motley Crue, Siouxsie and The Banshees, White Zombie. Database: DB2053.
Song title: "Here Comes The Sun". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Bon Jovi, Bruce Springsteen, Carmen Cuesta, Coldplay (Live), John Pizzarelli, Kings X, Miguel Rios, Miguel Rios, Monty. Alexander, Nina Simone, Phish, Riddim, Sergio Mendes, Sharon Forrester, Steve Harley and Cockney Rebel, Travis (Live), Travis, Vodoo glow skulls. Database: DB2053.
Song title: "Here, There And Everywhere". Original artist: Beatles. Covered by: Emmylou Harris. Database: DB2053.
Song title: "Hey Jude". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Chikocheeky, Donna Hightower, Edu Lobo, Grateful dead, John Holt, King Curtis, Marisol, The Brothers Johnson, The Dynamites, The London. Symphony Orchestra, The Temptations, Wilson Pickett. Database: DB2053.
Song title: "Hey you". Original artist: Pink Floyd. Covered by: Chillout tribute to PF, Furnance, Piano tribute to Pink Floyd, Wetton

Lukather. Shaw and White. Database: DB2053.
Song title: "Youve Got To Hide Your Love Away". Original artist: Beatles. Covered by: Eddie Vedder, Eddie Vedder, John Lennon, John Pizzarelli, Oasis, The Silkie, Travis. Database: DB2053.
Song title: "Higher love". Original artist: Depeche Mode. Covered by: Symphonic tribute to Depeche Mode. Database: DB2053.
Song title: "High hopes". Original artist: Frank Sinatra. Covered by: Frank Sinatra with kids. Database: DB2053.
Song title: "Home". Original artist: Depeche Mode. Covered by: Sylvain Chauveau and Ensemble Nocturne. Database: DB2053.
Song title: "Honey Pie". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Phish. Database: DB2053.
Song title: "Hot hot hot". Original artist: The Cure. Covered by: Inertia, String Quartet. Database: DB2053.
Song title: "I am what I am". Original artist: Gloria Gaynor. Covered by: Shirley Bassey, The best of musicals. Database: DB2053.
Song title: "I call you". Original artist: Beatles. Covered by: Mo indigo, The mamas and the papas. Database: DB2053.
Song title: "I feel you". Original artist: Depeche Mode. Covered by: Apollo 4 40, Placebo, Symphonic tribute to Depeche Mode. Database: DB2053.
Song title: "If I Fell". Original artist: Beatles. Covered by: Los diablos negros. Database: DB2053.
Song title: "If I Needed Someone". Original artist: Beatles. Covered by: Joe White. Database: DB2053.
Song title: "If only tonight we could sleep". Original artist: The Cure. Covered by: Deftones. Database: DB2053.
Song title: "I know what I like". Original artist: Genesis. Covered by: Genesis (version 2), Genesis (version 3), Steve Hackett. Database: DB2053.
Song title: "Imagine". Original artist: Beatles. Covered by: Chantal Kreviazuk (Live), David Bowie (Live), Diana Ross and The Supremes, Tony Ronald. Database: DB2053.
Song title: "Im a loser". Original artist: Beatles. Covered by: Eels, Marianne Faithfull. Database: DB2053.
Song title: "Im cold". Original artist: The Cure. Covered by: The Cure (Demo). Database: DB2053.
Song title: "Im down". Original artist: Beatles. Covered by: Fred James. Database: DB2053.
Song title: "I Me Mine". Original artist: Beatles. Covered by: Elliot Smith. Database: DB2053.
Song title: "Im looking through you". Original artist: Beatles. Covered by: Wallflowers. Database: DB2053.
Song title: "In between days". Original artist: The Cure. Covered by: Piano tribute to The Cure, The Cure (Demo), The Cure (Acoustic), The obligations. Database: DB2053.
Song title: "In My Life". Original artist: Beatles. Covered by: Bruno Lomas, Dave Matthews, Stephen Stills, The 52 key organ. Database: DB2053.
Song title: "Insensatez". Original artist: Vinicius de Moraes. Covered by: Antonio Carlos Jobim (version 1), Antonio Carlos Jobim (version 2), Astrud. Gilberto, Joao Donato, Joao Gilberto (version 1), Joao Gilberto (version 2), Mann Gilberto and Jobim, Passos and Carter, Robert Wyatt, Stan Getz, Wes Montgomery. Database: DB2053.
Song title: "Interstellar overdrive". Original artist: Pink Floyd. Covered by: Pink Floyd (Live), Pink Floyd (Remaster), Spiral realms. Database: DB2053.
Song title: "In the flesh". Original artist: Pink Floyd. Covered by: Belew White and Porcaro, Sherwood Porcaro and Colaiuta. Database: DB2053.
Song title: "In the mood". Original artist: Glenn Miller. Covered by: John Williams and The Boston Pops, Woody Allen movie music. Database: DB2053.
Song title: "In your room". Original artist: Depeche Mode. Covered by: Harshrealm, Sylvain Chauveau and Ensemble Nocturne. Database: DB2053.
Song title: "Garota de Ipanema". Original artist: Antonio Carlos Jobim. Covered by: Astrud Gilberto, Astrud Gilberto 2, Bobby Morganstein, Frank Sinatra, Gilberto. Gil, Jobim Vinicius and Morais, Jobim Vinicius and Toquinho, Najwajean (Remix), Najwajean, Rosa Passos and Ron Carter, Stan Getz, Tamba trio, The peeping toms, Vinicius de Moraes. Database: DB2053.
Song title: "I Saw Her Standing There". Original artist: Beatles. Covered by: Little Richard, Stan Webb. Database: DB2053.
Song title: "She is a Carioca". Original artist: Sergio Mendes trio. Covered by: Astrud Gilberto, Joao Gilberto. Database: DB2053.
Song title: "Is there anybody out there". Original artist: Pink Floyd. Covered by: Belew and Sherwood. Database: DB2053.
Song title: "Its more fun to compute". Original artist: Kraftwerk. Covered by: Takkyu Ishino. Database: DB2053.
Song title: "Its no good". Original artist: Depeche Mode. Covered by: Automatic, Orphans of infamy. Database: DB2053.
Song title: "Its not you". Original artist: The Cure. Covered by: Dead end, The Cure (Demo). Database: DB2053.
Song title: "I Wanna Be Your Man". Original artist: Beatles. Covered by: Brian Sewell. Database: DB2053.
Song title: "I want to hold your hand". Original artist: Beatles. Covered by: Balsara, Beatles (Remaster), Glen Adams, I against I, Mrs Yetta Bronstein. Database: DB2053.
Song title: "I Want You". Original artist: Beatles. Covered by: Jette Ives. Database: DB2053.
Song title: "I Will". Original artist: Beatles. Covered by: Alison Krauss, John Holt, Movimiento Urbeno and The Skatalites, Phish. Database: DB2053.
Song title: "Jesus Christ Superstar". Original artist: Andrew Lloyd Webber. Covered by: The best of musicals, The best of musicals 2, The James Taylor Quartet, Tony Ronald. Database: DB2053.
Song title: "Jugband blues". Original artist: Pink Floyd. Covered by: Eden. Database: DB2053.
Song title: "Jumping someone elses train". Original artist: The Cure. Covered by: Kill Switch Klick, Lukestar, Piano tribute to The Cure, The Ropers. Database: DB2053.
Song title: "Just cant get enough". Original artist: Depeche Mode. Covered by: C Project, Studio 99, Universal Circus. Database: DB2053.
Song title: "Just like heaven". Original artist: The Cure. Covered by: Piano tribute to The Cure, String Quartet, The Cure (Acoustic). Database: DB2053.
Song title: "Instant karma". Original artist: Beatles. Covered by: Nelly Furtado. Database: DB2053.
Song title: "I get a kick out of you". Original artist: Cole Porter. Covered by: Dinah Washington, Ella Fitzgerald, Ella Fitzgerald and Cole Porter, Frank. Sinatra, Johnny Hartman (version 1), Johnny Hartman (version 2), Tom Jones. Database: DB2053.
Song title: "Killing an arab". Original artist: The Cure. Covered by: Dwomo, Frodus, Superlemonade, The electric hellfire club. Database:

DB2053.
Song title: "Knocking on heavens door". Original artist: Bob Dylan. Covered by: Eric Clapton, Guns N Roses, U2 Bob Marley and Bob Dylan (Live). Database: DB2053.
Song title: "Do You Want To Know A Secret". Original artist: Beatles. Covered by: Billy Kramer and The Dakotas. Database: DB2053.
Song title: "Kyoto song". Original artist: The Cure. Covered by: Orders of Ellington. Database: DB2053.
Song title: "Lady Bird". Original artist: Tad Dameron. Covered by: Chet Baker. Database: DB2053.
Song title: "Lady Marmalade". Original artist: Labelle. Covered by: Aguilera Kim Mya Pink, All saints, Christina Aguilera (version 1), Christina. Aguilera (version 2), Disco fever collection, OT. Database: DB2053.
Song title: "The lady is a tramp". Original artist: Rodgers and Hart. Covered by: Ella Fitzgerald, Ella Fitzgerald and Frank Sinatra (version 1), Frank Sinatra. (version 2), Frank Sinatra (version 3), Low Rawls, Robbie Williams. Database: DB2053.
Song title: "Lagrimas negras". Original artist: Bebo y Cigala. Covered by: Unknown, Unknown 2, Unknown 3. Database: DB2053.
Song title: "Lament". Original artist: The Cure. Covered by: The Cure (Demo). Database: DB2053.
Song title: "La Tarara". Original artist: Camaron de la Isla. Covered by: Unknown, Unknown 2. Database: DB2053.
Song title: "Learning to fly". Original artist: Pink Floyd. Covered by: Leather strip. Database: DB2053.
Song title: "Lestaca". Original artist: Lluis Llach. Covered by: JM Serrat and Lluis Llach, Lluis Llach (Live), Orquesta cubana. Database: DB2053.
Song title: "Let It Be". Original artist: Beatles. Covered by: Basilio, Eros, Gladys Knight and The Pips, Leo Sayer, Marillion (Live), Mocedades, Nana Mouskouri, Nick Cave, Nicky Thomas, Patrick Samson set, Roscoe. Shelton, Shang shang Typhoon, The Upsetter. Database: DB2053.
Song title: "Lie to me". Original artist: Depeche Mode. Covered by: Psyche, Razed in black. Database: DB2053.
Song title: "Light my fire". Original artist: The Doors. Covered by: Jose Feliciano, The Black Mighty Orchestra. Database: DB2053.
Song title: "Little 15". Original artist: Depeche Mode. Covered by: Maga, Symphonic tribute to Depeche Mode. Database: DB2053.
Song title: "With A Little Help From My Friends". Original artist: Beatles. Covered by: Bobby Morganstein, Jeff Lyne (2in1), Joe Cocker, Los angeles, Sham69, Tori Amos. Database: DB2053.
Song title: "I say a little prayer". Original artist: Burt Bacharach. Covered by: Aretha Franklin, Diana King. Database: DB2053.
Song title: "Live and let die". Original artist: Beatles. Covered by: Byron Lee and The Dragonaires. Database: DB2053.
Song title: "Livin on a prayer". Original artist: Bon Jovi. Covered by: Dalimas (Remix), Heavydance, Karma, Nika. Database: DB2053.
Song title: "Los endos". Original artist: Genesis. Covered by: Patrick Moraz, Steve Hackett. Database: DB2053.
Song title: "The love cats". Original artist: The Cure. Covered by: Dj Bootius Maximus, String Quartet, Tricky. Database: DB2053.
Song title: "Lovely Rita". Original artist: Beatles. Covered by: Michelle Shocked, Roy Wood. Database: DB2053.
Song title: "Love Me Do". Original artist: Beatles. Covered by: The Beatle Barkers. Database: DB2053.
Song title: "Love me tender". Original artist: Elvis Presley. Covered by: Elvis Presley (version 2), Elvis Presley (version 3), Frank Sinatra. Database: DB2053.
Song title: "Love song". Original artist: The Cure. Covered by: 311, A perfect circle, David Ari Leon, Dommestix (Live), Emotep (Live), Jack off, Jill, Kiethevez, One last fix, Puppetland, Snake river conspiracy, String Quartet, Tanzwut, The Cure (Acoustic), Tool and A perfect circle, Tori Amos, Volumen cero. Database: DB2053.
Song title: "Lucy In The Sky With Diamonds". Original artist: Beatles. Covered by: Black Crows, Elton John, Grateful dead, Joseph Jaime, William Shatner. Database: DB2053.
Song title: "Lucifer Sam". Original artist: Pink Floyd. Covered by: Pink Floyd (Remaster), The electric hellfire club. Database: DB2053.
Song title: "Lullaby". Original artist: The Cure. Covered by: Buzzo, Leather strip, String Quartet, The Cure (Acoustic). Database: DB2053.
Song title: "Lady Madonna". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Gleemen, Gondwana, Jools Holland, The Crystalites. Database: DB2053.
Song title: "Magical Mystery Tour". Original artist: Beatles. Covered by: Ambrosia. Database: DB2053.
Song title: "Mamma mia". Original artist: ABBA. Covered by: ABBA musical. Database: DB2053.
Song title: "Manha de Carnaval". Original artist: Joao Gilberto. Covered by: Astrud Gilberto, Paquito D Rivera, Stan Getz. Database: DB2053.
Song title: "The man machine". Original artist: Kraftwerk. Covered by: XCR. Database: DB2053.
Song title: "Many rivers to cross". Original artist: Jimmy Cliff. Covered by: Joe Cocker, The Colgate Thirteen Cadence, UB40. Database: DB2053.
Song title: "Mas que nada". Original artist: Tamba trio. Covered by: Dom um Romao, Sergio Mendes and Brazil, VVAA. Database: DB2053.
Song title: "Master and servant". Original artist: Depeche Mode. Covered by: L-Kan, Locust, Los Acusicas, Studio 99. Database: DB2053.
Song title: "Maxwells Silver Hammer". Original artist: Beatles. Covered by: Frankie Laine. Database: DB2053.
Song title: "Maybe someday". Original artist: The Cure. Covered by: String Quartet. Database: DB2053.
Song title: "Meathook". Original artist: The Cure. Covered by: Jawbox, The Cure (Demo). Database: DB2053.
Song title: "Mediterraneo". Original artist: Joan Manuel Serrat. Covered by: Estopa, Lolita, Sabina, Sedajazz Big Band, Siempre asi. Database: DB2053.
Song title: "Memory". Original artist: Andrew Lloyd Webber. Covered by: Barbara Streisand, Michael Crawford, Sarah Brightman, The best of musicals. Database: DB2053.
Song title: "Message in a bottle". Original artist: The Police. Covered by: Incubus and No Doubt, John Mayer, Sting (Acoustic). Database: DB2053.
Song title: "Metall auf metall". Original artist: Kraftwerk. Covered by: Fading colours, Kraftwerk (Remix). Database: DB2053.
Song title: "Michelangelo". Original artist: Astor Piazzolla. Covered by: Los tangueros, Soledad. Database: DB2053.
Song title: "Michelle". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Richard Cocciante, The four stops, The

Overlanders. Database: DB2053.
Song title: "Money". Original artist: Beatles. Covered by: Barret Strong. Database: DB2053.
Song title: "Money". Original artist: Pink Floyd. Covered by: Chillout tribute to PF, London Philharmonic Orchestra, Piano tribute to Pink. Floyd, Pink Floyd (Live 1), Pink Floyd (Demo), Pink Floyd (Remix), Pink Floyd. (Live 2), String Quartet, VVAA. Database: DB2053.
Song title: "Monument". Original artist: Depeche Mode. Covered by: Alex Under, Gus gus. Database: DB2053.
Song title: "I am in the mood for love". Original artist: Charlie Parker. Covered by: Barbara Streisand, Charlie Parker (version 2). Database: DB2053.
Song title: "Moonlight bay". Original artist: Beatles. Covered by: Frank Sinatra. Database: DB2053.
Song title: "Let there be more light". Original artist: Pink Floyd. Covered by: Pink Floyd (Live), Pressurehear. Database: DB2053.
Song title: "Mother". Original artist: Pink Floyd. Covered by: Piano tribute to Pink Floyd, Wetton Belew and White. Database: DB2053.
Song title: "Mother Natures Son". Original artist: Beatles. Covered by: Phish, Sheryl Crow. Database: DB2053.
Song title: "Please Mister Postman". Original artist: Beatles. Covered by: The Marvelets. Database: DB2053.
Song title: "Musique non stop". Original artist: Kraftwerk. Covered by: Hajime Fukuma, Kraftwerk (Remix), Pierrepoint. Database: DB2053.
Song title: "My love". Original artist: Beatles. Covered by: Jr Walker, Ken Boothe. Database: DB2053.
Song title: "My Sharona". Original artist: The Knack. Covered by: Eldritch, Moritz y Leo, The Hormonauts, The Kinks, The Ramones. Database: DB2053.
Song title: "My sweet lord". Original artist: Beatles. Covered by: Boy George, Chiffons, Edwin Starr, Julio Iglesias, Keith Lynn and The. Dragonaires, Larry Norman, Megadeth, The Rudies. Database: DB2053.
Song title: "My way". Original artist: Frank Sinatra. Covered by: Adrivalan Orchestra, Keely Smith. Database: DB2053.
Song title: "You make me feel like a natural woman". Original artist: Aretha Franklin. Covered by: Celine Dion. Database: DB2053.
Song title: "You Never Give Me Your Money". Original artist: Beatles. Covered by: Sarah Vaughan, Wil Malone and Lou Reizner. Database: DB2053.
Song title: "Never let me down again". Original artist: Depeche Mode. Covered by: Depeche Mode (Remix), Depeche Mode (Remix), Depeche Mode (Remix), Digital 21, Smashing Pumpkins, Sylvain Chauveau and Ensemble Nocturne, Symphonic tribute. to Depeche Mode, Tina Root. Database: DB2053.
Song title: "Never say goodbye". Original artist: The Communards. Covered by: Gloria Gaynor (version 1), Gloria Gaynor (version 2), Michael Jackson. Database: DB2053.
Song title: "I will never smile again". Original artist: Frank Sinatra. Covered by: Frank Sinatra (Live), Frank Sinatra (version 2), Keely Smith. Database: DB2053.
Song title: "New York New York". Original artist: Frank Sinatra. Covered by: Bobby Morganstein, Frank Sinatra (version 2), Keely Smith, Liza Minnelli, Unknown. Database: DB2053.
Song title: "Night and day". Original artist: Cole Porter. Covered by: Billie Holiday, Ella Fitzgerald (version 1), Ella Fitzgerald (version 2), Frank Sinatra, Fred Astaire, Oscar Peterson, Sergio Mendes, The best of. musicals, Tony Bennett. Database: DB2053.
Song title: "The nile song". Original artist: Pink Floyd. Covered by: Farflung. Database: DB2053.
Song title: "Nobody home". Original artist: Pink Floyd. Covered by: London Philharmonic Orchestra. Database: DB2053.
Song title: "No me importa nada". Original artist: Luz Casal. Covered by: Lolita. Database: DB2053.
Song title: "Norwegian Wood (This Bird Has Flown)". Original artist: Beatles. Covered by: Herbie Hancock, I Camaleont, P M Dawn, Paul Lamb, Willie Lindo. Database: DB2053.
Song title: "Nothing else matters". Original artist: Metallica. Covered by: Metallica (Live), Lucie Silvas, Staind (Live), Metallica and London Symphony. Orchestra, Metallica (Live). Database: DB2053.
Song title: "Nothing compares to you". Original artist: Sinnead O Connor. Covered by: London Symphony Orchestra. Database: DB2053.
Song title: "But not tonight". Original artist: Depeche Mode. Covered by: Pseudocipher. Database: DB2053.
Song title: "Nowhere Man". Original artist: Beatles. Covered by: Jeff Lyne (2in1), Natalie Merchant, Paul Westenberg, Three good reasons. Database: DB2053.
Song title: "No woman no cry". Original artist: Bob Marley. Covered by: Ben Harper, Boney M, Carmen Consoli, Erikah Badu and Peter Tosh, Fugees, Gilberto Gil, Rancid, Sublime, Wyclef Jean (Remix), Ziggy Marley and. The Fugees. Database: DB2053.
Song title: "Nummern". Original artist: Kraftwerk. Covered by: Aiboforcen, Kraftwerk (version 2). Database: DB2053.
Song title: "O amor em paz". Original artist: Joao Gilberto. Covered by: Kai Winding, Tamba trio, Toquinho and Horta. Database: DB2053.
Song title: "O barquinho". Original artist: Joao Gilberto. Covered by: Joao Donato, Joao Gilberto (version 2), Lorez Alexandria, Mann Gilberto and. Jobim. Database: DB2053.
Song title: "Ob-La-Di, Ob-La-Da". Original artist: Beatles. Covered by: Dino, Ken Lazarus, Los Javaloyas, No Doubt (Live), Phish, Ribelli, The Marmalade. Database: DB2053.
Song title: "Oh! Darling". Original artist: Beatles. Covered by: Earl gaines, John Pizzarelli. Database: DB2053.
Song title: "Ohm sweet ohm". Original artist: Kraftwerk. Covered by: Apoptygma berzerk. Database: DB2053.
Song title: "One". Original artist: U2. Covered by: Pearl Jam and U2 (Live), REM and U2 (Live), Robbie Williams (Live), Johnny. Cash, Bono and Mary J Blidge. Database: DB2053.
Song title: "One of my turns". Original artist: Pink Floyd. Covered by: Tommy Shaw. Database: DB2053.
Song title: "One of these days". Original artist: Pink Floyd. Covered by: Spahn Ranch. Database: DB2053.
Song title: "Im only sleeping". Original artist: Beatles. Covered by: Blues Motel, Noel Gallager and Stereophonics, The Vines. Database: DB2053.
Song title: "Only you". Original artist: The Platters. Covered by: Adrivalan Orchestra, The Platters (Remaster). Database: DB2053.
Song title: "On the run". Original artist: Pink Floyd. Covered by: Din, Pink Floyd (Remix), String Quartet. Database: DB2053.
Song title: "O Pato". Original artist: Vinicius de Moraes. Covered by: Joao Gilberto (version 1), Joao Gilberto (version 2), Lisa Ono, Stan Getz, Stan Getz and Gilberto Gil. Database: DB2053.
Song title: "Somewhere over the rainbow". Original artist: Judy Garland. Covered by: Barbara Streisand, Eva Cassidy, Glenn Miller

and his orchestra, Guns N Roses. and Steve Vai, Meet Joe Black, Mishka Adams, Norah Jones, Ray Charles, Tom. Waits, Tori Amos. Database: DB2053.
Song title: "Your own special way". Original artist: Genesis. Covered by: John Wetton, Steve Hackett. Database: DB2053.
Song title: "Paperback writer". Original artist: Beatles. Covered by: Augusto Righetti, Baroque Chamber Orchestra, Kris Kristofferson, Lefty. in the night. Database: DB2053.
Song title: "Para la libertad". Original artist: Joan Manuel Serrat. Covered by: Joan Manuel Serrat (version 2). Database: DB2053.
Song title: "Penny Lane". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, David Bowie, Los tonks, Sting (Live), The Wilson. Malone voice band. Database: DB2053.
Song title: "People are people". Original artist: Depeche Mode. Covered by: Australian Blonde, Basic Implant, Studio 99. Database: DB2053.
Song title: "Personal Jesus". Original artist: Depeche Mode. Covered by: Johnny Cash, Marilyn Manson, Studio 99. Database: DB2053.
Song title: "The phantom of the opera". Original artist: Andrew Lloyd Webber. Covered by: The best of musicals. Database: DB2053.
Song title: "Photographic". Original artist: Depeche Mode. Covered by: Carnage, Freezepop, Poupue Fabrikk. Database: DB2053.
Song title: "Pictures of you". Original artist: The Cure. Covered by: David Ari Leon, Joy Electric, My life in rain, Piano tribute to The Cure, String Quartet, The Scaries. Database: DB2053.
Song title: "Piggy in the mirror". Original artist: The Cure. Covered by: Cinnamon Toast. Database: DB2053.
Song title: "Pigs on the wing". Original artist: Pink Floyd. Covered by: Helios creed, Piano tribute to Pink Floyd, Pink Floyd (version 2). Database: DB2053.
Song title: "Plainsong". Original artist: The Cure. Covered by: Black heaven, Cave in, Jaime sin tierra, Migala, Yuppie flu. Database: DB2053.
Song title: "Plastic passion". Original artist: The Cure. Covered by: Edsel, Eva, The Cure (Demo). Database: DB2053.
Song title: "Play for today". Original artist: The Cure. Covered by: Noise box, The Cure (Demo). Database: DB2053.
Song title: "Pleasure Little Treasure". Original artist: Depeche Mode. Covered by: Dirty princess. Database: DB2053.
Song title: "Point me at the sky". Original artist: Pink Floyd. Covered by: Melting euphoria. Database: DB2053.
Song title: "Policy of truth". Original artist: Depeche Mode. Covered by: Depeche Mode (Remix 1), Depeche Mode (Remix 2), Dishwalla, Disown, Studio 99, Sylvain Chauveau and Ensemble Nocturne, Symphonic tribute to Depeche Mode, Terry Hoax. Database: DB2053.
Song title: "Polythene Pam". Original artist: Beatles. Covered by: Roy Wood. Database: DB2053.
Song title: "Pornography". Original artist: The Cure. Covered by: Wreckage. Database: DB2053.
Song title: "The power of love". Original artist: Celine Dion. Covered by: London Symphony Orchestra. Database: DB2053.
Song title: "Oh pretty woman". Original artist: Roy Orbison. Covered by: David Alfaro, The Djangoes. Database: DB2053.
Song title: "Primary". Original artist: The Cure. Covered by: Bell Book and Candle. Database: DB2053.
Song title: "Dear Prudence". Original artist: Beatles. Covered by: Alanis Morissette, Siouxsie and The Banshees, U2. Database: DB2053.
Song title: "Push". Original artist: The Cure. Covered by: Bloomington. Database: DB2053.
Song title: "Whats new pussycat". Original artist: Burt Bacharach. Covered by: Tom Jones. Database: DB2053.
Song title: "Radioactivity". Original artist: Kraftwerk. Covered by: Data bank A, Fiction 8, Hikashu, Kraftwerk, Trylok. Database: DB2053.
Song title: "Rain". Original artist: Beatles. Covered by: Augusto Righetti, Bushmen, Grateful dead, The psychedelic filbert. Database: DB2053.
Song title: "Reach out I will be there". Original artist: Gloria Gaynor. Covered by: Gloria Gaynor (version 2), London Symphony Orchestra. Database: DB2053.
Song title: "You really got me". Original artist: The Kinks. Covered by: The Clash, Van Halen. Database: DB2053.
Song title: "Retrato em branco e preto". Original artist: Antonio Carlos Jobim. Covered by: Joao Gilberto, Lisa Ono. Database: DB2053.
Song title: "Revolution 1". Original artist: Beatles. Covered by: Grandaddy, INXS (Live), Kelly Jones and Stereophonics, Stone Temple Pilots, Thompson Twins and Madonna. Database: DB2053.
Song title: "Baby Youre A Rich Man". Original artist: Beatles. Covered by: Kula Shaker, Los canarios. Database: DB2053.
Song title: "The Long and Winding Road". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Beatles (Remaster), Dancing Mood, Diana Ross and. The Supremes, John Pizzarelli, Leo Sayer, Paloma San Basilio, The Corrs. Database: DB2053.
Song title: "Rock around the clock". Original artist: Chuck Berry. Covered by: Bill Haley and The Comets, Buddy Holly, Elvis Presley, Sex Pistols, Tritons. Database: DB2053.
Song title: "Rocky Raccoon". Original artist: Beatles. Covered by: Andy Fairweather Low. Database: DB2053.
Song title: "Roxanne". Original artist: The Police. Covered by: Ewan McGregor Jose Feliciano, Fall out boy, George Michael, Incubus, La Susi, P Daddy and Sting and Fugees (Remix), Tom Waits. Database: DB2053.
Song title: "Run For Your Life". Original artist: Beatles. Covered by: Earl Green. Database: DB2053.
Song title: "Run like hell". Original artist: Pink Floyd. Covered by: Chefe Zappa and Kaye, VVAA. Database: DB2053.
Song title: "Sacred". Original artist: Depeche Mode. Covered by: Moonspell. Database: DB2053.
Song title: "Samba de uma nota so". Original artist: Antonio Carlos Jobim. Covered by: Jobim Vinicius and Toquinho, Paulinho Nogueira, Stan Getz. Database: DB2053.
Song title: "Samba da Bencao". Original artist: Vinicius de Moraes. Covered by: Bebel Gilberto, Vinicius de Moraes and Odette Lara. Database: DB2053.
Song title: "Black Orpheus". Original artist: Vinicius de Moraes. Covered by: Bob Brookmeyer, Joao Donato, Unknown. Database: DB2053.
Song title: "Samba da minha terra". Original artist: Tamba trio. Covered by: Joao Gilberto, Joao Gilberto 2, Lucio Alves, Mann Gilberto and Jobim. Database: DB2053.
Song title: "Satisfaction". Original artist: Rolling Stones. Covered by: Aretha Franklin, Britney Spears, Fatboy slim, Ottis Redding, Rolling Stones. (Remaster). Database: DB2053.
Song title: "Saturday night". Original artist: The Cure. Covered by: Soylent green. Database: DB2053.
Song title: "A saucerful of secrets". Original artist: Pink Floyd. Covered by: EXP, Pink Floyd (Live). Database: DB2053.
Song title: "Scarborough fair". Original artist: Simon and Garfunkel. Covered by: Celtic chillout album, Sergio Mendes and Brasil.

Database: DB2053.
Song title: "Seek and destroy". Original artist: Metallica. Covered by: Metallica and Pantera (Live), Skazi (Remix), Testament, Tobias Luke. Database: DB2053.
Song title: "Ive just seen a face". Original artist: Beatles. Covered by: John Pizzarelli, Pearl Jam (Live). Database: DB2053.
Song title: "Send in the clowns". Original artist: Stephen Sondheim. Covered by: Andre Rieu, The best of musicals, Tom Jones. Database: DB2053.
Song title: "Set the controls for the heart". Original artist: Pink Floyd. Covered by: Pink Floyd (Live 1), Pink Floyd (Live 2), Psychic TV. Database: DB2053.
Song title: "Sexy Sadie". Original artist: Beatles. Covered by: Paul Weller, Phish. Database: DB2053.
Song title: "Sgt. Peppers Lonely Hearts Club Band". Original artist: Beatles. Covered by: Beatles (Remix), The new apocalypse. Database: DB2053.
Song title: "Shake the disease". Original artist: Depeche Mode. Covered by: Brave new world, Hooverphonic, Luxury, Studio 99. Database: DB2053.
Song title: "Shake dog shake". Original artist: The Cure. Covered by: Ex-Voto, Shudder to think. Database: DB2053.
Song title: "She Came In Trough The Bathroom Window". Original artist: Beatles. Covered by: Mimi Maura, The Bee Gees. Database: DB2053.
Song title: "Sheep". Original artist: Pink Floyd. Covered by: Chillout tribute to PF, Pink Floyd (Remix). Database: DB2053.
Song title: "She loves you". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Beatles (Remaster), Duo Kramer, Emi Bonilla, Fausto. Leali, Irvins 89 key organ, Lennon McCartney, Little Boys, Stan Webb. Database: DB2053.
Song title: "She Said She Said". Original artist: Beatles. Covered by: Yeah yeah noh. Database: DB2053.
Song title: "Shes Leaving Home". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Brian Ferry, Syreeta, Tori Amos. Database: DB2053.
Song title: "Shine on you crazy diamond". Original artist: Pink Floyd. Covered by: Chillout tribute to PF, Piano tribute to Pink Floyd, Pink Floyd (Remix), Pink. Floyd (version 2), VVAA. Database: DB2053.
Song title: "The show must go on". Original artist: Pink Floyd. Covered by: Belew and Colaiuta. Database: DB2053.
Song title: "Showroom dummies". Original artist: Kraftwerk. Covered by: Kraftwerk (Remix), Leaether strip, Senor Coconuts, Thee Hyphen. Database: DB2053.
Song title: "Siamese twins". Original artist: The Cure. Covered by: Curious. Database: DB2053.
Song title: "Smoke on the water". Original artist: Deep Purple. Covered by: Black Sabbath, Dream Theater and Bruce Dikinson, Led Zeppelin, Metallica, Yngwie Malmsteen. Database: DB2053.
Song title: "Solsbury hill". Original artist: Peter Gabriel. Covered by: Peter Gabriel (version 2). Database: DB2053.
Song title: "Somebody". Original artist: Depeche Mode. Covered by: Veruca Salt. Database: DB2053.
Song title: "Something". Original artist: Beatles. Covered by: Bob Dylan, James Brown, Joe Cocker, Karina, Larry Coryell, Martha Reeves and. The Vandellas, Natty Combo, Pavement, Radiohead, Selway, The Blues Busters. Database: DB2053.
Song title: "Sometimes". Original artist: Depeche Mode. Covered by: Vienna 1. Database: DB2053.
Song title: "So nice". Original artist: Bebel Gilberto. Covered by: Astrud Gilberto, Astrud Gilberto and Walter Wanderley, Bebel Gilberto. (version 2), Sergio Mendes trio. Database: DB2053.
Song title: "Spacelab". Original artist: Kraftwerk. Covered by: Mental conquest. Database: DB2053.
Song title: "Spanish Harlem". Original artist: Aretha Franklin. Covered by: Adrivalan Orchestra, Tom Jones. Database: DB2053.
Song title: "Speak to me breathe". Original artist: Pink Floyd. Covered by: Pink Floyd (Remix), String Quartet. Database: DB2053.
Song title: "Stairway to heaven". Original artist: Led Zeppelin. Covered by: Beatnix, Del pueblo del barrio, DJ Earworm, Dolly Parton, Final Fantasy X, Foo Fighters, Gregorian Masters of Chant, Helloween, Iron Maiden, Leningrad. cowboys and Red Army, Orchestre National de Jazz, Rodrigo y Gabriela, Rvd. Billy C Wirtz (Live), Sisters of Mercy, The String Quartet, Those Darn. Accordians, Vienna Symphonic Orchestra. Database: DB2053.
Song title: "Stand by me". Original artist: Ben E King. Covered by: John Lennon. Database: DB2053.
Song title: "Stop". Original artist: Pink Floyd. Covered by: Billy Sherwood. Database: DB2053.
Song title: "Stormy weather". Original artist: Ethel Waters. Covered by: Frank Sinatra, Quincy Jones, The best of musicals. Database: DB2053.
Song title: "A strange day". Original artist: The Cure. Covered by: Apoptygma berzerk. Database: DB2053.
Song title: "Strange love". Original artist: Depeche Mode. Covered by: Studio 99. Database: DB2053.
Song title: "Strangers in the night". Original artist: Frank Sinatra. Covered by: Adrivalan Orchestra, Orquesta cubana. Database: DB2053.
Song title: "Strawberry Fields Forever". Original artist: Beatles. Covered by: Ben Harper, Grateful dead, Peter Gabriel, Signs. Database: DB2053.
Song title: "Street fighting man". Original artist: Rolling Stones. Covered by: Rolling Stones (Remaster). Database: DB2053.
Song title: "Stripped". Original artist: Depeche Mode. Covered by: Rammstein (Remix), Rammstein, Shiny Toy Guns, Sylvain Chauveau and Ensemble. Nocturne, Tiamat. Database: DB2053.
Song title: "Yellow Submarine". Original artist: Beatles. Covered by: Akiko Kanazawa, Baroque Chamber Orchestra, Derek Enright MP, Duo dinamico, Klaus Beyer and Band, Los Fernandos, NoMataras. Database: DB2053.
Song title: "Sun King". Original artist: Beatles. Covered by: The Bee Gees. Database: DB2053.
Song title: "The sun and the rainfall". Original artist: Depeche Mode. Covered by: After coma, Marry Thoughts. Database: DB2053.
Song title: "You are the sunshine of my life". Original artist: Stevie Wonder. Covered by: Tom Jones. Database: DB2053.
Song title: "Suspicious minds". Original artist: Elvis Presley. Covered by: Best of reggae. Database: DB2053.
Song title: "Sweet home Alabama". Original artist: Lynyrd Skynyrd. Covered by: Body Pump 59, Dave Matthews Band, Elvis Presley,

Led Zeppelin, Leningrad. Cowboys. and Russian Red Army Choir, Steve Miller Band. Database: DB2053.
Song title: "Tanto tempo". Original artist: Bebel Gilberto. Covered by: Bebel Gilberto (Remaster). Database: DB2053.
Song title: "Taschenrechner". Original artist: Kraftwerk. Covered by: Black wedding, Dhiva. Database: DB2053.
Song title: "Taxman". Original artist: Beatles. Covered by: Resistencia suburbana. Database: DB2053.
Song title: "Smells like teen spirit". Original artist: Nirvana. Covered by: Tori Amos. Database: DB2053.
Song title: "That will be the day". Original artist: Beatles. Covered by: The Djangoes. Database: DB2053.
Song title: "The End". Original artist: Beatles. Covered by: George Benson, London Symphony Orchestra. Database: DB2053.
Song title: "Here comes the Flood". Original artist: Peter Gabriel. Covered by: Peter Gabriel (version 2). Database: DB2053.
Song title: "The Fool On The Hill". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Beatles (Live), Edmundo Ros and Catarina Valente, Helen Reddy, Junior, Sergio Mendes and Brasil, Sergio Mendes and Brazil66, Shirley Bassey, The four stops. Database: DB2053.
Song title: "Theres A Place". Original artist: Beatles. Covered by: Les Surfs. Database: DB2053.
Song title: "The trial". Original artist: Pink Floyd. Covered by: Malcolm McDowell. Database: DB2053.
Song title: "The walk". Original artist: The Cure. Covered by: Bellatronica, Piano tribute to The Cure, The Cure (Acoustic). Database: DB2053.
Song title: "The Word". Original artist: Beatles. Covered by: Harvey Averne. Database: DB2053.
Song title: "The things you said". Original artist: Depeche Mode. Covered by: Sylvain Chauveau and Ensemble Nocturne. Database: DB2053.
Song title: "Things We Said Today". Original artist: Beatles. Covered by: John Pizzarelli, The Sandpipers. Database: DB2053.
Song title: "The thin ice". Original artist: Pink Floyd. Covered by: Anderson and Levin. Database: DB2053.
Song title: "Ticket To Ride". Original artist: Beatles. Covered by: Asfalto, Cathy Berberian. Database: DB2053.
Song title: "Tico tico". Original artist: Perez Prado. Covered by: 101 Strings, Charlie Parker, Esquivel, Henry Mancini, Hollywood Bowl Symphony. Orchestra, Les Paul and Mary Ford, Liberace, Microscopic Septet, Paco de. Lucia, Philharmonic Jazz, SHS jazz choir, The Harmonicats, Xavier Cugat. Database: DB2053.
Song title: "Time". Original artist: Pink Floyd. Covered by: Chillout tribute to PF, London Philharmonic Orchestra (Remix), London. Philharmonic Orchestra, Pink Floyd (Remix), Pink Floyd (Remix), Pink Floyd. (Live), String Quartet. Database: DB2053.
Song title: "As time goes by". Original artist: Louis Armstrong. Covered by: Bobby Morganstein, Johan Stengard, Natalie Cole, Royal philharmonic Orchestra, Unknown, Unknown 2, Woody Allen movie music. Database: DB2053.
Song title: "Tomorrow Never Knows". Original artist: Beatles. Covered by: Grateful dead, Jay Atwood and Susan Maccorkle, Our Lady Peace, Phil Collins, Sheila Chandra and Monsoon. Database: DB2053.
Song title: "Tour de France". Original artist: Kraftwerk. Covered by: Dhiva and Skibby, Inertia, Ionic vision, Kraftwerk (version 2), Kraftwerk. (version 3), Kraftwerk (version 4). Database: DB2053.
Song title: "Trans Europa express". Original artist: Kraftwerk. Covered by: Kraftwerk (Remix), Kraftwerk, Limbo, Senor Coconuts. Database: DB2053.
Song title: "Triste". Original artist: Antonio Carlos Jobim. Covered by: Antonio Carlos Jobim (version 2), Oscar Peterson, Paulinho Nogueira, Sergio. Mendes and Brasil. Database: DB2053.
Song title: "Trust". Original artist: The Cure. Covered by: El agente naranja. Database: DB2053.
Song title: "Twist And Shout". Original artist: Beatles. Covered by: Beatles (Remaster), Los Sirex, Nirvana Sonic Youth and REM, Revolver. Database: DB2053.
Song title: "Two of Us". Original artist: Beatles. Covered by: Aimee Mann and Michael Penn, Skylab2000. Database: DB2053.
Song title: "You have got a friend". Original artist: James Taylor. Covered by: Carole King, Paul Carrack. Database: DB2053.
Song title: "I have got you under my skin". Original artist: Cole Porter. Covered by: Bobby Morganstein, Diana Krall, Dinah Washington, Ella Fitzgerald, Frank. Sinatra, Gloria Gaynor, Keely Smith, Oscar Peterson. Database: DB2053.
Song title: "Uran". Original artist: Kraftwerk. Covered by: Battery, Kraftwerk (version 2). Database: DB2053.
Song title: "Us and them ". Original artist: Pink Floyd. Covered by: Chillout tribute to PF, London Philharmonic Orchestra, Piano tribute to Pink. Floyd, Pink Floyd (Live), Pink Floyd (Remix), String Quartet, VVAA. Database: DB2053.
Song title: "Valley of the kings". Original artist: Genesis. Covered by: Steve Hackett. Database: DB2053.
Song title: "Vera". Original artist: Pink Floyd. Covered by: Shaw and Howe. Database: DB2053.
Song title: "Visions of angels". Original artist: Genesis. Covered by: David Allen and Solid Space. Database: DB2053.
Song title: "Vivir asi es morir de amor". Original artist: Camilo Sesto. Covered by: Alaska and Camilo Sesto, El canto del loco, Javier Parra, Miguel Nandez y. Marey. Database: DB2053.
Song title: "Dance on a volcano". Original artist: Genesis. Covered by: Steve Hackett. Database: DB2053.
Song title: "Waiting for the worms". Original artist: Pink Floyd. Covered by: Sherwood Levin and Colaiuta. Database: DB2053.
Song title: "Waiting for the night". Original artist: Depeche Mode. Covered by: Rabbit in the moon. Database: DB2053.
Song title: "Waiting room only". Original artist: Genesis. Covered by: Steve Hackett. Database: DB2053.
Song title: "Walking in my shoes". Original artist: Depeche Mode. Covered by: Depeche Mode (Remix 1), Depeche Mode (Remix 2), Depeche Mode (Remix 3), Depeche Mode (Remix 4), Depeche Mode (Remix 5), Depeche Mode (Remix 6), Simo, Studio 99, Symphonic tribute to Depeche Mode. Database: DB2053.
Song title: "I Am The Walrus". Original artist: Beatles. Covered by: John Otway, Leo Sayer, Men without hats, XTC. Database: DB2053.
Song title: "Watcher of the skies". Original artist: Genesis. Covered by: Steve Hackett. Database: DB2053.
Song title: "Watching the weels". Original artist: John Lennon. Covered by: John Lennon instrumental. Database: DB2053.
Song title: "All along the watchtower". Original artist: Bob Dylan. Covered by: Bruce Springsteen and Bob Dylan (Live), Dave Matthews and Ben Harper, Jimi. Hendrix, Santana Howe Krieger West and more (Live), U2. Database: DB2053.
Song title: "Wave". Original artist: Antonio Carlos Jobim. Covered by: Antonio Carlos Jobim and Vinicius de Morais, Camargo Mariano and Lubambo, Henry Mancini, Joao Gilberto, Sedajazz Big Band, Sergio Mendes and Brasil, The Sandpipers. Database: DB2053.
Song title: "We can work it out". Original artist: Beatles. Covered by: Baroque Chamber Orchestra, Four Seasons, Heather nova, Massiel,

Mike Liddell. and Gli Atomi, Sandro, Shelter, Stevie Wonder, The Beatle Barkers. Database: DB2053.
Song title: "Welcome to the machine". Original artist: Pink Floyd. Covered by: VVAA. Database: DB2053.
Song title: "We will rock you". Original artist: Queen. Covered by: Britney Pink Beyonce, Genesis, Moritz, Robbie Williams. Database: DB2053.
Song title: "What the world needs now". Original artist: Burt Bacharach. Covered by: Sergio Mendes. Database: DB2053.
Song title: "When Im Sixty-Four". Original artist: Beatles. Covered by: John Pizzarelli, Keith Moon. Database: DB2053.
Song title: "While My Guitar Gently Weeps". Original artist: Beatles. Covered by: Holy Piby, Joe Louis Walker, Kenny Lattimore, Michael Hedges, Peter Frampton, Spineshank, The natural 1. Database: DB2053.
Song title: "White Christmas". Original artist: Irving Berlin. Covered by: Barry Manilow, Dion Jones Flack and more, Elvis Presley and Bing Crosby, Louis. Armstrong, Palast orchester and Max Raabe, Rondo Veneziano. Database: DB2053.
Song title: "Why cant I be you". Original artist: The Cure. Covered by: Half foot outside. Database: DB2053.
Song title: "Walk on the wild side". Original artist: Lou Reed. Covered by: Lou Reed 2. Database: DB2053.
Song title: "I will survive". Original artist: Gloria Gaynor. Covered by: Celia Cruz, Gloria Gaynor (version 2), Gloria Gaynor (version 3), Gloria Gaynor. (version 4). Database: DB2053.
Song title: "Days of wine and roses". Original artist: Henry Mancini. Covered by: Jose Carreras, Marian McPartland, Quartette Tres Bien. Database: DB2053.
Song title: "Wish you were here". Original artist: Pink Floyd. Covered by: Chillout tribute to PF, Piano tribute to Pink Floyd. Database: DB2053.
Song title: "What a wonderful world". Original artist: Louis Armstrong. Covered by: Adrivalan Orchestra, Nick Cave, The Flaming Lips. Database: DB2053.
Song title: "It Wont Be Long". Original artist: Beatles. Covered by: Redd Kross. Database: DB2053.
Song title: "World in my eyes". Original artist: Depeche Mode. Covered by: Depeche Mode (Remix), Studio 99, The Cure. Database: DB2053.
Song title: "Yer Blues". Original artist: Beatles. Covered by: Clapton Lennon Richards Mitchell (Live). Database: DB2053.
Song title: "Yes it is". Original artist: Beatles. Covered by: Don Henley (Live). Database: DB2053.
Song title: "Yesterday". Original artist: Beatles. Covered by: Bob Dylan, Boys II Men, David Essex, Eva Cassidy, I Trappers, James Taylor, Joe White, Lisa Ono, Los modulos, Marvin Gaye, Miguel Rios, The Flame all. stars, The Templeton twins, Wilco (Live). Database: DB2053.
Song title: "You Cant Do That". Original artist: Beatles. Covered by: Diana Ross and The Supremes, Ruby Turner. Database: DB2053.
Song title: "Young lust". Original artist: Pink Floyd. Covered by: Glenn Hughes, Penal colony, VVAA. Database: DB2053.
Song title: "You Wont See Me". Original artist: Beatles. Covered by: Ernest Ranglin, Ernie Smith. Database: DB2053.

Outliers

"I Dont Want to Spoil the Party" by Beatles, "Think For Yourself" by Beatles, "Good Morning Good Morning" by Beatles, "Honey Dont" by Beatles, "Hold Me Tight" by Beatles, "Wild Honey Pie" by Beatles, "Within You Without You" by Beatles, "Maggie Mae" by Beatles, "Any Time At All" by Beatles, "Long Long Long" by Beatles, "Every Little Thing" by Beatles, "Tell Me Why" by Beatles, "Her Majesty" by Beatles, "For You Blue" by Beatles, "Words of Love" by Beatles, "Fixing A Hole" by Beatles, "I Need You" by Beatles, "What Youre Doing" by Beatles, "Martha My Dear" by Beatles, "Mean Mr Mustard" by Beatles, "Julia" by Beatles, "Love You To" by Beatles, "Till There Was You" by Beatles, "Flying" by Beatles, "Little Child" by Beatles, "The Night Before" by Beatles, "I Want To Tell You" by Beatles, "I Should Have Known Better" by Beatles, "Mr. Moonlight" by Beatles, "Misery" by Beatles, "Its Only Love" by Beatles, "Ill Be Back" by Beatles, "Im So Tired" by Beatles, "Chains" by Beatles, "Octopuss Garden" by Beatles, "Anna (Go To Him)" by Beatles, "Act Naturally" by Beatles, "Everybodys Trying to Be My Baby" by Beatles, "Another Girl" by Beatles, "Birthday" by Beatles, "One After 909" by Beatles, "What Goes On" by Beatles, "All Ive Got To Do" by Beatles, "Everybodys Got Something To Hide Except Me and M" by Beatles, "You Like Me Too Much" by Beatles, "Please Please Me" by Beatles, "Boys" by Beatles, "Savoy Truffle" by Beatles, "P. S. I Love You" by Beatles, "A Taste Of Honey" by Beatles, "Piggies" by Beatles, "When I Get Home" by Beatles, "No Reply" by Beatles, "Not A Second Time" by Beatles, "Dig It" by Beatles, "Doctor Robert" by Beatles, "Wait" by Beatles, "Dig a Pony" by Beatles, "Dont Bother Me" by Beatles, "Kansas City- Hey, Hey, Hey, Hey" by Beatles, "You Really Got A Hold On Me" by Beatles, "Dont Pass Me By" by Beatles, "Your Mother Should Know" by Beatles, "Babys In Black" by Beatles, "Ill Follow the Sun" by Beatles, "Ask Me Why" by Beatles, "Im Happy Just To Dance With You" by Beatles, "Baby Its You" by Beatles, "Devil In Her Heart" by Beatles, "Cry Baby Cry" by Beatles, "Youre Going to Lose That Girl" by Beatles, "Blue Jay Way" by Beatles, "Tell Me What You See" by Beatles, "The meaning of love" by Depeche Mode, "Macro" by Depeche Mode, "See you" by Depeche Mode, "Damaged people" by Depeche Mode, "Its called a heart" by Depeche Mode, "Dreaming of me" by Depeche Mode, "I want it all" by Depeche Mode, "A pain that Im used to" by Depeche Mode, "The darkest star" by Depeche Mode, "Nothing is impossible" by Depeche Mode, "New life" by Depeche Mode, "Precious" by Depeche Mode, "Leave in silence" by Depeche Mode, "The sinner in me" by Depeche Mode, "Introspectre" by Depeche Mode, "Lillian" by Depeche Mode, "Suffer well" by Depeche Mode, "John the revelator" by Depeche Mode, "Love in itself" by Depeche Mode, "Another brick in the wall part 1" by Pink Floyd, "Love scene take 4" by Pink Floyd, "Love scene take 1" by Pink Floyd, "Another brick in the wall part 2" by Pink Floyd, "Keep smiling people" by Pink Floyd, "Love scene take 2" by Pink Floyd, "Another brick in the wall part 3" by Pink Floyd, "Eclipse" by Pink Floyd, "Love scene take 3" by Pink Floyd, "The promise" by The Cure, "I dont know whats going on" by The Cure, "Cold colours" by The Cure, "Us or them" by The Cure, "Going Nowhere" by The Cure, "I want to be old" by The Cure, "Lost" by The Cure, "Labyrynth" by The Cure, "Before Three" by The Cure, "Alt End" by The Cure, "Anniversary" by The Cure, "Faded smiles" by The Cure, "See the cildren" by The Cure, "Never" by The Cure, "Need myself" by The Cure, "Listen" by The Cure, "Taking off" by The Cure, "Doo doo doo la la la" by The Police, "Dont stand so close to me" by The Police, "I cant stand loosing you" by The Police, "Walking on the moon" by The Police, "Everything you do is magic" by The Police, "We wish you a merry christmas" by Barry White, "Moonlight in Vermont" by Billie Holiday, "Another brick in the wall" by Chillout tribute to PF, "Another brick in the wall" by Controlled bleeding, "Behind the wheel" by Cut rate box, "My joy" by Depeche Mode, "My joy" by Depeche Mode, "Moonlight

in Vermont" by Frank Sinatra and Ella Fitzgerald, "Another brick in the wall" by London Philharmonic Orchestra, "Another brick in the wall" by Lukather and Levin, "Another brick in the wall" by Morse and Sherwood, "Extranos ateos" by Orquesta cubana, "Extranos ateos" by Orquesta cubana, "Another brick in the wall" by Pink Floyd, "Another brick in the wall" by Pink Floyd, "Another brick in the wall" by VVAA, "Another brick in the wall" by Waybill Montrose and Porcaro.

Appendix B: Relevant publications by the author

Here we summarize the relevant publications or technical reports that are associated with the work done in this thesis.

J. Serrà & E. Gómez. *A cover song identification system based on sequences of tonal descriptors*. Music Information Retrieval EXchange (MIREX) extended abstract. In Proceedings of the 8th International Symposium on Music Information Retrieval (ISMIR). Vienna, Austria. 2007.

J. Serrà. *A qualitative assessment of measures for the evaluation of a cover song identification system*. In Proceedings of the 8th International Symposium on Music Information Retrieval (ISMIR). Vienna, Austria. 2007.

J. Serrà & E. Gómez. *Music similarity method based on instantaneous sequences of tonal descriptors*. United States provisional patent application number 60/946860. Filed June 28, 2007.