Audio Cover Song Identification

Carlos Manuel Rodrigues Duarte

Thesis to obtain the Master of Science Degree in
Information Systems and Computer Engineering

Supervisors: Doctor David Manuel Martins de Matos

Examination Committee

Chairperson: Doctor João Emílio Segurado Pavão Martins
Supervisor: Doctor David Manuel Martins de Matos
Member of the Committee: Doctor Sara Alexandra Cordeiro Madeira

October 2015
Acknowledgements

I would like to thank my advisor Doctor David Martins de Matos for giving me the freedom to choose the way I wanted to work and for all the advices and guiding provided.

I would like to thank L2F and INESC-ID for providing me with the means I needed to produce my work.

I would like to thank Teresa Coelho for making the dataset that proved to be extremely useful for me to test all my work and also Francisco Raposo, for providing me summarization versions of that dataset in order for me to conduct my experiments.

Last but not least, I would like to thank my friends and family for all the support and strength given to keep me focus on this journey.

None of this would be possible without these people.

Lisboa, November 1, 2015

Carlos Manuel Rodrigues Duarte
For my family and friends
Resumo

A identificação de covers musicais é uma das principais tarefas na comunidade de Recuperação de Informação Musical e tem utilizações práticas como a identificação de violações de direitos de autor ou de estudos relativamente a tendências musicais. Os sistemas criados para a identificação de covers baseiam-se no conceito de similaridade musical. Para calcular essa similaridade, é necessário compreender os aspetos musicais adjacentes, como o timbre, ritmo ou instrumentação, que caracterizam uma música, contudo, esse tipo de informação nem sempre é fácil de identificar, interpretar e utilizar.

Esta tese começa por dar informações acerca dos possíveis aspetos musicais e como estes influenciam o processo de identificação. São estudadas as abordagens comuns seguidas que utilizam a informação fornecida por esses aspetos musicais e como se pode calcular um valor de similaridade entre músicas. Também se explica como é que se pode avaliar a qualidade de um sistema. Foi escolhido um sistema para servir como base e, com base no trabalho recente na área, foram elaboradas algumas experiências na tentativa de obter uma melhoria de resultados.

Na melhor experiência obteve-se um acréscimo de 5% Mean Average Precision e 109 novas covers identificadas, através do uso de descritores de melodia e voz em fusão com os resultados obtidos pelo sistema base.
Abstract

Audio cover song identification is one of the main tasks in Music Information Retrieval and has many practical applications such as copyright infringement detection or studies regarding musical influence patterns. Audio cover song identification systems rely on the concept of musical similarity. To compute that similarity, it is necessary to understand the underlying musical facets such as timbre, rhythm and instrumentation, that characterize a song but, since that kind of information is not easy to identify, interpret and use, it is not a straightforward process.

This thesis begins by giving information about the possible musical facets and how they influence the process of identifying a cover. The most common approaches to take advantage of those musical facets are addressed as well as how the similarity values between a pair of songs can be computed. There is also an explanation of how the system quality can be assessed. A system was chosen to serve as baseline and, based on recent work in the field, some experiments were made in order to try achieving an improvement in the results.

In the best experiment an increase of 5% Mean Average Precision and 109 more covers being identified was obtained, using the similarity values of melody and voice descriptors fused together with the results given by the baseline.
Palavras Chave

Características Chroma

Fusão de Distâncias

Identificação de Covers Musicais

Similaridade Musical

Recuperação de Informação Musical

Keywords

Chroma Features

Distance Fusion

Audio Cover Song Identification

Music Similarity

Music Information Retrieval
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Most common types of covers</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Musical facets</td>
<td>4</td>
</tr>
<tr>
<td>2.3</td>
<td>Possible changes in musical facets according to the type of cover</td>
<td>5</td>
</tr>
<tr>
<td>3.1</td>
<td>Related work results comparison</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>Rhythmic Features</td>
<td>37</td>
</tr>
<tr>
<td>4.2</td>
<td>Smith-Waterman weighted distance matrix</td>
<td>40</td>
</tr>
<tr>
<td>4.3</td>
<td>Results of all experiments</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>Statistical analysis of the changes in cover detection</td>
<td>42</td>
</tr>
<tr>
<td>4.5</td>
<td>Correlation coefficients results</td>
<td>43</td>
</tr>
</tbody>
</table>
Introduction

Technology is rapidly evolving and nowadays it is possible to access digital libraries of music anywhere and anytime, and have personal libraries that can easily exceed the practical limits to listen to them (Casey, Veltkamp, Goto, Leman, Rhodes, and Slaney 2008). These fast-pacing advances in technology also present new opportunities in the research area where patterns, tendencies and levels of influence can be measured in songs and artists. By having a way to compute a similarity measure between two songs, it is possible to provide new services such as automatic song recommendation and detection of copyright infringements.

These services can be achieved by identifying cover songs since, by their nature, cover songs rely on the concept of music similarity. In musical terms, a cover is a re-recording of an existing song that may, or may not, be performed by the original artist or have exactly the same features, but it has something that makes it recognizable once knowing the original. There exist many types of covers ranging from renowned bands that produce an existing music but in a style that corresponds to their identity, to unknown people who play music with the simple goal of trying to perform a song that they like.

The identification of cover songs is best made by humans. However, the amount of existing musical content makes manual identification of different versions of a song infeasible and, thus, an automatic solution must be used to achieve that even though it entails the issue of not knowing the exact way to represent human being’s cognitive process. With that in mind, it is important to know which are the musical facets that characterize a song and what are the exist-
ing cover types in order to understand how they can be explored to make cover identification possible and the difficulty of computing an accurate similarity value between two songs.

Knowing the musical facets and how they can be used to extract meaningful information, a cover detection system can be constructed. The goal of this work will be to use an existing system, analyze its results, and develop a way to improve them. In this case, the improvements will be guided towards the identification of covers that are the closest possible, in terms of lyrics or instrumentation, to the original version.

This thesis will begin by giving background information about musical facets and how they affect the process of identifying musical covers. It will review the most common approaches that audio cover song identification systems take in order to produce quality results and recent work in the area is addressed. A system was chosen to serve as baseline and based on the ideas of some recent work in the field, some experiments were conducted to improve the quality of its results. The improvement of the results was achieved using a heuristic for distance fusion between extracted melodies and the baseline, making possible the detection of covers that presented similar melodies and similar singing.

This document is organized as follows: Chapter 2 reviews the underlying musical facets that condition the process of identifying a cover and addresses the common approaches taken for audio cover song identification. Chapter 3 describes public datasets and related work, with all the results gathered in Table 3.1. Chapter 4 describes all the experimental setups followed by the discussion of the results obtained. Chapter 5 concludes this document and future work is discussed.
Musical Facets and Approaches for Cover Detection

The concept of cover is, in general, usually applied in the simplified sense of an artist reproducing the work of another artist, but it is not that straightforward. It is important to know what types of similarities exist between two songs and what they consist of. The type of cover can give us information, such as the changes that were applied or if the song was performed by the same artist or not. The most common types of covers (Serrà 2011) are presented in Table 2.1.

<table>
<thead>
<tr>
<th>Cover type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remaster</td>
<td>Reproduced by the original artist. Sound enhancement techniques are applied to an existing work.</td>
</tr>
<tr>
<td>Instrumental</td>
<td>Adaptation of a song without the vocal component.</td>
</tr>
<tr>
<td>Acapella</td>
<td>Adaptation of a song using only vocals.</td>
</tr>
<tr>
<td>Mashup</td>
<td>Song or composition created by blending two or more pre-recorded songs. The result is a single song that usually presents the vocals of one track over the instrumental part of another.</td>
</tr>
<tr>
<td>Live Performance</td>
<td>Live recording of the performance of the original artist or other performers.</td>
</tr>
<tr>
<td>Acoustic</td>
<td>Adaptation without electronic instruments.</td>
</tr>
<tr>
<td>Demo</td>
<td>Original version of a song that usually has the purpose of being sent to record labels, producers, or another artists, with the goal of trying to have someone’s work published.</td>
</tr>
<tr>
<td>Duet</td>
<td>Re-recording or performance of an existing song with more lead singers than the original.</td>
</tr>
<tr>
<td>Medley</td>
<td>Happens mostly in live performances. Several songs are played continuously without interruptions and a particular order.</td>
</tr>
<tr>
<td>Remix</td>
<td>Addition or removal of elements that compose a song or simply the modification of the equalization, pitch, tempo, or another musical facet. Sometimes the final product barely resembles the original one.</td>
</tr>
<tr>
<td>Quotation</td>
<td>Embedding of a brief segment of another song in an analogous way to quotations in speech or literature.</td>
</tr>
</tbody>
</table>

The type of cover can be useful to reveal what sort of resemblance we can expect between two songs. By knowing the most common possible types, one can expect a remasterization to
CHAPTER 2. MUSICAL FACETS AND APPROACHES FOR COVER DETECTION

be much more similar to the original song than a quotation or a remixed version. That is due to the large quantity of possible variations that complicate the process of associating two songs together. Even a live performance can display enough variations to make the two digital audio signals different. Those variations may be irrelevant for the human brain, but for a machine, that may have to work directly on the digital signal, it can make all the difference.

The variations that might be present can be associated to one or more musical facets and are very relevant to the process of identifying alternative versions of a song. Those changes must be taken into account at the time the audio signal is being processed and can be relative to variations in timbre, pitch or even the entire structure of the musical piece. Table 2.2 shows the musical facets that may contribute to the distinction of two different songs.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timbre</td>
<td>It is the property that allows us to recognize a sound’s origin. Timbre variations can be a result of different processing techniques (e.g. Equalization, microphones) that introduce texture variations or by way of instrumentation such as different instruments, configurations or recording procedures.</td>
</tr>
<tr>
<td>Pitch</td>
<td>The pitch can be low or high and it is related to the relative frequency of the musical note.</td>
</tr>
<tr>
<td>Tempo</td>
<td>The different in tempo execution can be deliberated (other performers may prefer a version with a different beat-rate) or unintentional (in live performance, for example, it is hard to perfectly respect the original tempo)</td>
</tr>
<tr>
<td>Timing</td>
<td>The rhythmical structure of a piece might change according to the intention or feelings of the performer.</td>
</tr>
<tr>
<td>Structure</td>
<td>The original structure can be modified to not include certain segments such as the introduction or the inclusion of new segments like the repetition of the chorus.</td>
</tr>
<tr>
<td>Key</td>
<td>The key can be transposed to the whole song or to a selected section so that it is adapted to the pitch range of a different singer or instrument.</td>
</tr>
<tr>
<td>Lyrics and language</td>
<td>Translation to another language or simply recording using different lyrics.</td>
</tr>
<tr>
<td>Harmonization</td>
<td>The relation created by using several notes and chords simultaneously. It is independent of the main key and may imply changes to the chord progression or the main melody.</td>
</tr>
<tr>
<td>Rhythm</td>
<td>The way that sounds are arranged, working as a “pulse” of the song.</td>
</tr>
<tr>
<td>Melodic line or Bassline</td>
<td>Combination of consecutive notes (or silences) derived from the existing rhythms and harmonies.</td>
</tr>
<tr>
<td>Noise</td>
<td>Interferences such as public manifestations (e.g. cheers, screaming, whispers)</td>
</tr>
</tbody>
</table>
Once knowing the possible type of covers and underlying musical facets, it is possible to establish a relation between the two domains, as Table 2.3 shows. The indicated relations are possible but not necessary. Table 2.3 is based on the content of Serrà (2011).

Table 2.3: Possible changes in musical facets according to the type of cover

<table>
<thead>
<tr>
<th></th>
<th>Timbre/Pitch</th>
<th>Tempo/Timing</th>
<th>Structure</th>
<th>Key</th>
<th>Lyrics &amp; Language</th>
<th>Harmony</th>
<th>Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remaster</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Instrumental</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Acapella</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Mashup</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Live</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Acoustic</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Demo</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Duet</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Medley</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Remix</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Quotation</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

So far, the existing solutions for automatic audio cover identification rely on several approaches that try to make the most (or ignore) the information obtained from this musical facets and every year new techniques and approaches are created in the area of Music Information Retrieval (MIR). There is even a audio cover song identification competition\(^1\) held every year by an annual meeting named Music Information Retrieval Evaluation eXchange (MIREX).

MIREX is an annual meeting that is organized and managed by the International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL) of the University of Illinois at Urbana-Champaign (UIUC) that has the objective of comparing state-of-the-art algorithms and systems in Music Information Retrieval (Serrà and Gómez 2008).

Among the several existing tasks, there is one exclusive to audio cover song identification that has a strict regulation that solutions who desire to compete replicate. In order to evaluate

---

\(^1\)http://www.music-ir.org/mirex/wiki/2014:Audio_Cover_Song_Identification
the quality of a system, a dataset is used as input. This dataset is composed by 30 tracks and another 10 different versions, or covers, each, making a total of 330 tracks. To add noise to the dataset, another 670 songs that are not considered covers of the other 330, are included. The result is a collection of 1,000 songs in a diverse variety of genres and styles.

To determine the quality of a system, each of the 330 covers are used as input in it and the system must return a 330x1000 matrix that presents the similarity value for each cross-relation between one cover and each and every song in the dataset, including the song itself. The similarity value inside each cell is computed by a distance metric applied to the two musical pieces and it will be used to assess the quality of the results.

Additionally, the computational performance of the system can also be evaluated by measuring the time required to provide an answer. A specific threshold of time must be met, and if it is not respected, the system is disqualified.

2.1 Approaches for Cover Detection

How the information is handled depends on the information itself and, to tackle the problems faced in MIR, there are three types of information that can be used: metadata, high-level descriptors, and low-level audio features.

- **Metadata.** Is data that provides information about other data. Metadata can be divided in two: factual and cultural metadata. Factual metadata states data such as the artists’ name, year of publication, name of the album, title of the track, and length, whereas cultural metadata presents data like emotions (transmitted by the song), genre, and style. The problem with metadata is that, for it be obtained, it has to be inserted by human judges, and thus, is subject to representation errors. However, the worse aspect about
it is the time required to create it. Since the data has to be inserted manually, creating metadata for a large collection demands a great amount of time (Casey, Veltkamp, Goto, Leman, Rhodes, and Slaney 2008).

- **High-level music content description.** The descriptors are musical concepts such as the melody or the harmony to describe the content of the music. Those are the features that a listener can extract from a song intuitively or that can be measured. Table 2.1 has already presented these musical facets.

- **Low-level audio features.** This strategy uses digital information present in the audio signals of a music file.

Although these types of information can be all used for cover detection, most of the systems use the low-level audio features strategy (Casey, Veltkamp, Goto, Leman, Rhodes, and Slaney 2008). The goal is to compute a similarity value between different renditions of a song by identifying the musical facets that they share or, at least, the ones that present fewer variations. Those facets (e.g. timbre, key, tempo, structure) are subject to variations that make the process of computing that value more complex and forces the cover identification systems to be robust in that sense. The most common invariations that they try to achieve are related to tempo, key, and structure, since they are, in general, the most frequent changes and, together with feature extraction, they constitute the four basic blocks of functionality that may be present in an audio cover identification system.

Figure 2.1 presents these blocks of functionality. They will be explained in detail in the following sections.
2.1.1 Feature Extraction

In this approach, there is an assumption that the main melody or the harmonic progression between two versions is preserved, independently of the main key used. That representation, or tonal sequence, is used as comparator in almost all cover identification algorithms. The representation can be the extraction of the main melody or at harmonic level with the extraction of the chroma features (also known as Pitch Class Profiles (PCP)).

PCP features are created based upon the energy found in certain ranges of frequency in short-time spectral representations of the audio signal. This means that for each short-time segment (typically 100ms) of a song, a histogram is created that represents the relative intensity of each of the semitones. There are 12 semitones in an equal-tempered chromatic scale (represented in Figure 2.2) and the histogram represents only the octave-independent semitones, meaning that all the frequencies that represent the octaves of a semitone are collapsed
2.1. APPROACHES FOR COVER DETECTION

into a single bin.

![Chromatic Scale](image)

In Fig. 2.3, a representation of the PCP features (or Chromagram) is illustrated. One can clearly see the energy of each semitone (derived by the original frequency) for each time segment. The PCP approach is very attractive because it creates a degree of invariance to several musical characteristics since they generally try to respect the following requirements:

- Pitch distribution representation of both monophonic and polyphonic signals.
- Harmonic frequencies inclusion.
- Noise and non-tonal sounds tolerance.
- Timbre and instrumentation independent.
- Frequency tuning independent (there is no need to have a reference frequency such as A=440Hz).

One variant of PCP is the Harmonic Pitch Class Profiles (HPCP) and it considers the presence of harmonic frequencies, thus describing tonality. In the work of (Serrà and Gómez 2008), they extract HPCP features of an 36-bin octave-independent histogram that represents each 1/3 of the existing 12 semitones and use those HPCP features to create a global HPCP for each song. Then, the global HPCP is normalized by its maximum value and the maximum resemblance between two songs is calculated.
Another strategy is to adopt the concept of chords into PCP features. In order to estimate chord sequences from PCP features, template-matching techniques can be used. The logic behind template-matching is to define a binary mask in which the pitch classes that correspond to the components that constitute a given chord are set to one, while the others that do not are set to zero. The number of bins is subjective and it is typically 12, 24, or 36. In 24 bins, for example, there is a division between the lowest 12 semitones and the highest 12 semitones in which the lowest 12 represent the bassline. More bins means more specificity and, with the proper techniques, better results, but it also means that more computational resources are needed.

### 2.1.2 Key Invariance

One of the most frequent changes between versions of one song is in its key. Not all systems explore key invariance but those that use tonal representation, like PCP, do. A key transposition is represented as a ring-shift in the Chromatic Scale. This means that a bin, in PCP, that is assigned to a certain pitch is transposed to the next pitch class. There are several strategies
to handle transposition. The most common one is to perform all possible transpositions and use a similarity measure to get the most probable transposition. This strategy is the one that guarantees the best result but, it has the drawback of having poor performance, since the data becomes larger and more expensive to search.

One way to speed up the process of computing all transpositions is the Optimal Transposition Index (OTI) (Serrà, Gomez, and Herrera 2008). The OTI represents the number of positions that a feature vector needs to be circularly shifted. In (Serrà and Gómez 2008), after computing the global HPCP for each song, the OTI is computed so that it will represent the number of bins that need to be shifted in one song, so that they get the maximum resemblance between each other.

Other strategies include estimating the main key or using shift-invariant transformations. Key estimation is a very fast approach but, in case of errors, they propagate rapidly and deteriorate the accuracy of the system. A workaround can be estimating the K most probable transpositions and it has been shown that near-optimal results have been reached with just two shifts (Serrà, Gomez, and Herrera 2008).

Some recent systems (Bertin-Mahieux and Ellis 2012; Humphrey, Nieto, and Bello 2013) achieve transposition invariance by using 2D Fourier Transform, a technique widely used in the image processing domain due to its ability to separate patterns into different levels of detail (thus compacting energy) and for matching in vector spaces (Bertin-Mahieux and Ellis 2012). By computing the 2D Fourier Transform, one can obtain not only key invariance but also achieve phase-shift invariance and local tempo invariance.

In Fig. 2.4, we can observe the results of applying the 2D Fourier Transform to several segments. (A) is the original segment while (B) is transposed by two semitones, (C) is shifted
by one beat, and (D) has been time-shifted by 8 beats resulting in a musically different fragment (Marolt 2008).

The representation of all but (D) are very similar. (D) is different from the others because the time-shift was such that it produced a completely different music segment. Typically, the segments have the length of 8 or 12 beats but, in order to prevent these cases from happening, longer segments can be constructed with, for example, 75 beats (Bertin-Mahieux 2013).

2.1.3 Tempo Invariance

In some cover songs, tempo can be altered in such a way that extracted sequences cannot be directly compared. If a cover has a tempo 2 times faster than the original, a frame might correspond to two frames in the original and that creates the need for a way to be able to correspond those frames effectively.

One way to achieve that is using the extracted melody line to determine the ratio of du-
ration between two consecutive notes and another one is to estimate the tempo by resorting to beat tracking (estimating the beat of a song). An alternative to the latter is to do temporal compressing and expansion that consists of re-sampling the melody line into several musically compressed and expanded versions that will be compared so that the correct re-sampling is determined. The 2D Fourier transform, as previously mentioned, can also be used to achieve tempo invariance.

Lastly, dynamic programming techniques can be employed to automatically discover local correspondences. Considering the neighboring constraints and patterns, one can determine the local tempo deviations that are possible. Dynamic Time Warping (DTW) algorithms are the typical choice because their main goal is exactly to align two sequences in time achieving an optimal match.

2.1.4 Structure Invariance

The classic approach to make a system structure invariant is summarizing a song into its most repeated or representative parts (Gomez, Herrera, Vila, Janer, Serra, Bonada, El-Hajj, Aussenac, and Holmberg 2008; Marolt 2006). In order to do that, the system has to be capable of segmenting the structure and determine what the most important segments are. Structural segmentation (to identify the key structural sections) is another active area of research within the MIR community and it also has its own contest every year in MIREX but, similarly to what happens in cover detection, the solutions are not perfect. One also has to consider that sometimes the most identifiable segment of a musical piece is a small segment like an introduction or bridge and not always the most repeated one, like a chorus.

Dynamic programming algorithms, in particular, local-alignment algorithms such as the Smith-Waterman algorithm (Smith and Waterman 1981), can also be used to deal with some
structural changes between two songs. What they do is compare only the best sub-sequence alignment found between the tonal representations of two songs.

### 2.1.5 Similarity Measures

The last step of an audio cover song identification system is to compute the similarity values between any two songs of the dataset. The resulting value will determine how similar two songs are and, after all the values are computed, they allow us to validate the quality of the system and of the underlying implemented approaches. If the representation of a track is made using an Euclidean Space, one can use the Euclidean Distance (2.1) equation. The distance value will serve as the similarity value, since similar songs will be represented close to each other.

$$e(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \quad (2.1)$$

The same principle can be employed by using the Cosine Distance equation (2.2).

$$c(A, B) = \cos(\theta_{A,B}) = \frac{A \cdot B}{\|A\| \|B\|} \quad (2.2)$$

Another approach is using dynamic programming algorithms in the representation of the two songs, as discussed in section 2.1.4. For that matter, one that can be used is the DTW algorithm that is a technique used to find the optimal path to align two sequences in time and returns the distance between the two sequences (i.e. the total alignment cost between two features). Figure 2.5 illustrates the process of aligning two sequences in time and the Algorithm 1 shows how to implement this solution. The two sequences construct a matrix and the optimal path describes the insertion, deletion and matching operations necessary to convert one
sequence into the other. What distinguishes the DTW algorithm from the Smith-Waterman algorithm is that the DTW tries to align two sequences in time as a whole, while the Smith-Waterman algorithm matches local alignments in order to find the optimal path.

Algorithm 1: Dynamic Time Warping algorithm

| **Data:** Q and C: feature vectors of two songs |
| **Result:** Distance value between Q and C |

```c
int DTWDistance(Q: array[1..n], C: array[1..m])
|
DTW := array[0..n, 0..m] for i := 1 to n do
| DTW[i, 0] := infinity
end
for i := 1 to m do
| DTW[0, i] := infinity
end
DTW[0, 0] := 0 for i := 1 to n do
| for i := 1 to m do
| cost := d(Q[i], C[j])
| DTW[i, j] := cost + minimum(DTW[i-1, j]/"insertion"/), DTW[i, j-1]/"deletion"/, DTW[i-1, j-1]/"match"/)
end
end
return DTW[n, m]
```

One way of storing the results, independently of the similarity measure used, is by building a matrix that represents all the relations between two songs and the corresponding value. This is the method employed in the MIREX competition and it provides a good way of evaluating
the accuracy of the results.

2.1.6 Evaluation Metrics

Once the system provides the results obtained from the computation of the similarity measures, there is the need to evaluate the accuracy and quality of them. In order to do so, we need to know beforehand which are the songs that make up the dataset and from those songs, which are covers of each other. This knowledge allows us to construct the confusion matrix for each song and its possible covers, and the elements of the confusion matrix such as the True Positives (TP) or False Positive (FP) are necessary to construct the evaluation metrics that are usually implemented. Some systems make use of basic statistical measures such as Precision (2.3)

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{2.3}
\]

that tells us how many of the songs that were correctly identified as covers are truly covers, or Recall (2.4) that informs us how many of the covers that exist were retrieved in the results.

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{2.4}
\]

However, most of the existing solutions replicate the evaluation metrics implemented in the MIREX competition which consist of the total number of covers identified in top 10, which is given by the precision equation, the Average Precision (AP) in top 10, given by (2.5)

\[
AP = \frac{\sum_{i=1}^{n}(P(k) \times rel(k))}{C} \tag{2.5}
\]
where \( n \) is the number of retrieved results, \( k \) is the rank of a element in the result list, \( rel \) is a function that returns 1 if the item at rank \( k \) is a cover or 0 otherwise, and \( C \) is the total number of covers of the song that was used as input for the query. The average precision can be used to compute the Mean (arithmetic) of Average Precision (MAP) which is given by equation (2.6)

\[
MAP = \frac{\sum_{i=1}^{N} AP(i)}{N}
\] (2.6)

where \( i \) represents a query and \( N \) is the number of queries made. Lastly, the mean of the first correctly identified cover is also measured by using the Mean Reciprocal Rank (MRR) (2.7).

\[
MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_{i}}
\] (2.7)

where \( \text{rank}_{i} \) is the rank of the correct response of query \( i \) in the returned response list (Downie, Ehmann, Bay, and Jones 2010).

2.2  Summary

This chapter reviewed the most common types of covers such as instrumental, live performance, and remix versions, as well as the underlying musical facets that characterize a song such as tempo, structure, and key. The most common approaches taken for cover song identification were addressed. The approaches presented were: feature extraction, key invariance, tempo invariance, and structure invariance. The chapter was concluded by presenting how the similarity values between songs can be computed, and how the quality of the results provided by the system can be assessed.
Over the last years, in the area of cover song identification, there has been a considerable amount of new approaches and techniques that try to handle different issues. The typical goal is to try new algorithms or combinations of them in order to improve the results in comparison to previous systems, but the recent main focus by most researchers has been towards scalable strategies. The most common way to calculate the similarity between two different songs is through the use of alignment-based methods and they have shown to be able to produce good results\(^1\) (75% MAP in MIREX’2009). However, these methods are computational expensive and, when applied to large databases, they can become impractical: the best performing algorithm (Serrà, Gómez, Herrera, and Serra 2008) in MIREX’2008 implemented a modified version of the Smith-Waterman algorithm and took approximately 104 hours to compute the results for 1,000 songs\(^2\). If applied to the Million Song Dataset (MSD) dataset, the estimated time to conclude would be of 6 years (Balen, Bountouridis, Wiering, and Veltkamp 2014).

In the following sections, existing public datasets will be addressed as well as some of the recent work made in the audio cover song identification area. After that, the achieved results and approaches used by each of them will be presented in Table 3.1, followed by a brief discussion of those results.

\(^1\)http://www.music-ir.org/mirex/wiki/2009:Audio_Cover_Song_Identification_Results
\(^2\)http://www.music-ir.org/mirex/wiki/2008:Audio_Cover_Song_Identification_Results#Run_Times
3.1 Datasets

Any cover identification system must use a dataset to confirm its ability to perform the actions that it was designed to. Some developers produce their own music collections or reproduce the one used in the MIREX competition (since it is not available) but there has been a recent effort to create datasets and provide them freely for any researcher that desires to use it. The main advantage is having a way to compare results with the work of other researchers that used the same dataset.

3.1.1 Million Song Dataset

The most used dataset is the MSD (Bertin-Mahieux, Ellis, Whitman, and Lamere 2011) that, as the name suggests, is composed of one million tracks of several different artists, genres, and styles. The main purpose of this dataset is to encourage research in a large-scale fashion by providing metadata and audio features extracted with the EchoNest API\(^3\) and stored in a single file with the HDF5\(^4\) format. The HDF5 format is capable of efficiently handling heterogeneous types of information such as audio features in variable array lengths, names as strings, similar artists, and the duration of the track. This means that the audio files are not provided with the dataset and thus the researchers are limited to the audio features extracted, such as the timbre, pitches and loudness max. One subset of the MSD is the Second Hand Songs (SHS)\(^5\) dataset that is a list of 18,196 covers songs and the corresponding 5,854 original pieces within the MSD that can be used to evaluate if a song is truly a cover of another.

\(^3\)http://developer.echonest.com/
\(^4\)http://www.hdfgroup.org/HDF5/
\(^5\)http://labrosa.ee.columbia.edu/millionsong/secondhand
3.2. BLAST FOR AUDIO SEQUENCES ALIGNMENT

3.1.2 Covers80

Another dataset commonly used is the Covers80\textsuperscript{6} dataset that is composed by a collection of 80 songs, each performed by two artists (thus having a total of 160 songs). Unlike the MSD, the audio files (32 kbps, 16 kHz mono) are available.

3.2 BLAST for Audio Sequences Alignment

Martin, Brown, Hanna, and Ferraro (2012), the authors claim that the existing dynamic programming techniques are slow to align subparts of two songs and so they propose using BLAST (Basic Local Alignment Search Tool) which is used in bioinformatics for sequence searching. The BLAST algorithm is a development of the Smith-Waterman algorithm that follows a time-optimized model contrary to the more accurate and expensive calculations (Altschul, Gish, Miller, Myers, and Lipman 1990). It assumes that only a small amount of good alignments are found when querying a large database for results and so it filters the database to avoid computing irrelevant alignments of unrelated sequences. To filter the database and creating regions of data with strong similarity, they use several heuristic layers of rules that serve to index the search space for later local alignments. The main heuristic that they use lies in the assumption that significant local alignments include small exact matches. To detect those small exact matches, they have to seed the search space and determining the best seed depends of the representation and application of the database.

Once the search space is seeded, some filtering must be performed to select only the best subsequences that correspond to true similarity, instead of just coincidence. To evaluate the performance of the system, they used two different datasets. One was the MSD dataset and

\textsuperscript{6}\url{http://labrosa.ee.columbia.edu/projects/coversongs/covers80/}
the other consisted of 2514 songs coming from their own personal music collection. In their dataset, they implemented the HPCP with a constant time frame and 36-bins. It was shown that the system was capable of providing results extremely fast (0.33 second per query against 129 seconds per query in an alignment-based system) but with inferior accuracy (30.11% MAP against 44.82%). They also experimented with the MSD dataset but the computing time for each query was of 12.2 seconds. This can be explained by the apparent limitation of the MSD chroma features regarding sequence alignment with only 12 dimensions.

3.3 Chord Profiles

Khadkevich and Omologo (2013) explore the use of two high-level features: chord progressions and chord profiles, for large-scale cover detection. Their approach to make their solution scalable is to use the Locality-Sensitive Hashing (LSH) by indexing chord profiles and thus avoiding to perform pair-wise comparisons between all the songs in the database. The chord profiles of a musical piece are a compact representation that summarizes the rate of occurrence of each chord and chord progressions are a series of musical chords that are typically preserved between covers.

In their approach, first they extract beat-synchronous chord progressions. For evaluation purposes, they used two datasets: the MSD and the raw audio files of SHS, which they name SHS-wav. In SHS-wav, they make the extraction of beats and chords through external software. In the MSD dataset, they resort to the template-matching techniques of Oudre, Grenier, and Févotte (2009).

Since the LSH algorithm indexes similar items in the same regions, it is used to retrieve the nearest neighbors of the queried song, according to the result given by the $L_1$ distance (3.1)
between the chord profiles of the songs, where $a$ and $b$ represent feature vectors of two songs.

$$L_1(a, b) = \|a - b\| = \sum_{i=1}^{n} |a_i - b_i|$$

Having the nearest neighbors, the results are re-ranked according to the score given by the Levenshtein distance measure (Levenshtein 1966) between chord progressions and the best $K$ results are selected.

The evaluation of their system revealed that the best results were given by the 24-bin chroma features generated with their own dataset in comparison to the results obtained with the 12-bin chroma features given by the MSD. They also compared their results with the work of Bertin-Mahieux and Ellis (2012) proving that their approach had better results, having had 20.62% MAP in their dataset, and 3.71% on the MSD.

### 3.4 2D Fourier Transform Magnitude

Bertin-Mahieux and Ellis (2012) adopt the 2D Fourier Transform Magnitude (2DFTM) to achieve key invariance in the pitch axis and for fast matching in the Euclidean space, making it suitable for large-scale cover identification. Each song is represented by a fixed-length vector that defines a point in the Euclidean space and to discover its covers, it simply has to find which points are closer.

The process is divided into six stages. The Chroma features and beat estimation are obtained from the MSD dataset, and resampled into several beat grids and a power-law expansion is applied to enhance the contrast between weak and strong chroma bins. Then, the PCPs are divided into 75-beat long patches and the 2DFTM is computed, keeping only the median for
each bin across all patches. Finally, the last step consists of using Principal Component Analysis (PCA) (Jolliffe 1986) to reduce dimensionality. This solution obtained a MAP of 1.99%, and needed 3 to 4 seconds to compute each query.

3.5 Data Driven and Discriminative Projections

Back in 2012, Bertin-Mahieux and Ellis (2012) solution improved the state-of-the-art on large-scale cover song recognition, when compared to existing solutions and it has served as baseline for more recent systems. The work of Humphrey, Nieto, and Bello (2013) was one of those systems and they suggest two modifications to improve the original work: a sparse, high-dimensional data-driven component to improve the separability of data and a supervised reduction of dimensions.

By resorting to a data-driven approach, they apply the same concepts of data mining. They learn a set of rules or bases from a training set and try to encode a behavior from a small number of active components that hopefully is present in new data. They perform three pre-processing operations, advocating that the 2DFTM, by itself, is not enough for a particularly good feature extraction. The operations were: logarithmic compression and vector normalization for non-linear-scaling and PCA, to reduce the dimensionality and discard redundant components, producing a sparsed out single patch of all the 2DFTM 75-beat segments.

The K-Means algorithm is applied to the sparse data to capture local features and embed summary vectors into a semantically organized space. After computing the aggregation, their next step was applying supervised dimensionality reduction, using LDA (Linear Discriminant Analysis), to recover an embedding where distance values could be computed. They evaluated their work with the MSD and SHS dataset and compared their results to the baseline. The MAP
obtained was 13.41\%, meaning that the results were highly improved, particularly at the top-
K results but it takes three times more to compute the results compared to the original work. 
Another drawback is the tendency to have overfitted learning, although the authors claim that 
it is alleviated by using PCA.

### 3.6 Cognition-Inspired Descriptors

Balen, Bountouridis, Wiering, and Veltkamp (2014) suggest the use of high-level musical 
features that describe the harmony, melody, and rhythm of a musical piece. They argue that 
these cognition-inspired audio descriptors are capable of effectively capturing high-level mu-
sical structures such as chords, riffs, and hooks that have a fixed dimensionality and some 
tolerance to key, tempo, and structure invariance. After extracting these descriptors, they are 
used to assess the similarity between two songs.

In their work, they propose three new descriptors: the pitch bihistogram, the chroma cor-
relation coefficients, and the harmonization feature. The pitch bihistogram expresses melody 
and it is composed by pitch bigrams that represent a set of two different pitch classes that occur 
less than a predefined distance apart across several segments. The correlation coefficients are 
related to the harmony of a song and it consists of a 12x12 matrix that contains in its cells the 
correlation value between two 12-dimensional chroma time series. The correlation value that 
is inserted in the cells is a representation of how many times a set of pitches appears simulta-
neously in a signal. Finally, the Harmonization feature is a set of histograms of the harmonic 
pitches as they accompany each melodic pitch. This way, the information about harmony and 
melody are combined. 12-dimensional melodic pitches and 12-dimensional harmonic features 
are used and, thus, the harmonization feature also has a 12x12 dimensionality.
CHAPTER 3. RELATED WORK

To make their system scalable, they also adopted the use of LSH, but they do not test their solution in a true large-scale environment. They obtained 0.563% using recall at the top 5, and the dataset that they used was the Cover80 dataset that is composed by only 160 songs. The SHS dataset, for example, was not used, because it does not provide all the information needed to produce the suggested high-level descriptors and thus, by not evaluating their work in the environment that it is supposed to work, there is no way of knowing the potential benefits of exploring these approach.

3.7 Chroma Codebook with Location Verification

Lu and Cabrera (2012) focus on detecting remixes of a particular song. Their strategy is based on the Bag-of-Audio-Words model, the audio codebook. An audio codebook is constituted by audio words, and audio words are the centroids of audio features. In their experiments, they extracted the Chroma features of the songs with the EchoNest API and use hierarchical K-means clustering to find centroids (i.e., audio words) in those features. The algorithm used was the K-means++ (Arthur and Vassilvitskii 2007) with 10 hierarchy levels and, once the audio words are detected, the audio features are quantized into them. This means that the representation of a song will no longer be by the beat-aligned chroma features but, instead, by its audio words.

With the resulting audio words, computing the similarity between two songs can be achieved by checking how many audio words they share. For them to be considered true matches, the audio words shared must preserve their order. In order to exclude false matches, a location coding map (L-Map) is constructed for each song by using each of its audio words to split the song in two parts and filling a matrix with a binary value that indicates if another
audio word is in the first or second part of the song. The L-Map representation is given by the following example:

\[
L_{\text{map}} = \begin{bmatrix}
  v_1 & v_2 & v_3 & v_i \\
  v_1 & 1 & 0 & 1 & 1 \\
  v_2 & 1 & 1 & 1 & 1 \\
  v_3 & 0 & 0 & 1 & 1 \\
  v_i & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\] (3.2)

This matrix, for each row, represents the selected splitting audio word. The features in the columns are put to 0 or 1 if they are before or after it (or itself). Once these matrixes are constructed for each song, to delete false matches, one must perform the XOR operation between the matrixes of two songs and, if there is any mismatching value (which will be shown as 1), then it is a false match and it will be not be taken into account in the similarity computation.

The performance of their solution is scalable since the chosen algorithm for the K-means algorithm is logarithmic in time and indexing the audio words is made though an inverted file structure in which SongID-Location features are related to an audio word. The suggested approach was tested with a dataset composed by 43,000 tracks plus 92 selected tracks (20 original tracks and 72 similar to them) and the obtained results revealed that the larger the codebook, the better the achieved results reaching a score of 80% average precision at the top 30 ranked songs.
3.8 **Approximate Nearest Neighbors**

A novel technique is explored by Tavenard, Jégou, and Lagrange (2013) where the Approximate Nearest Neighbors (ANN) are retrieved for one song. By using the ANN, accuracy is traded for efficiency and the search is performed in an indexing structure that contains the set of vectors associated with all the songs in the database. They use a recent method (Jégou, Tavenard, Douze, and Amsaleg 2011) to index a large quantity of vectors that it is believed to outperform the LSH algorithm and, after retrieving the neighbors, a re-ranking stage is used to improve the quality of the nearest neighbors. Once the set of neighbors is obtained, they are filtered out so that incoherent matches are discarded.

Their approach was compared to the LabROSA method (Ellis and Cotton 2007) that consists of pairwise comparisons with dynamic programming on the Covers80 dataset, and the results obtained revealed that it had worse, but comparable, scores available in far less time. The best result was 50% using recall at the top 50.

3.9 **Tonal Representations for Music Retrieval**

Salamon, Serrà, and Gómez (2013) explore the fusion of different musical features and so they construct descriptors to describe melody, bassline, and harmonic progression. The melody is extracted using the work of Salamon (2013) that won the MIREX'2011 Audio Melody Extraction task. A similar approach is used for the bassline but with different tuning, and the harmonic progression is represented by a 12-bin HPCP octave-independent with 100ms frames.

After retrieving the results of the melody extractor, the extracted frequencies were converted into cents (logarithmic unit of measure used for musical intervals) and the pitch values were quantized into semitones which are then mapped into a single octave. To reduce the
length of the sequence, the original hops of 2.9ms each are summarized into 150ms hops. This is accomplished by building a pitch histogram every 150ms and selecting the pitch bin with the most occurrences.

Matching is achieved by a dynamic programming algorithm named $Q_{\text{max}}$. The purpose of the $Q_{\text{max}}$ algorithm is to find the best subsequential match between two time series, making it a local alignment algorithm.

Every descriptor has a dissimilarity matrix and the relation between pairs of songs is represented by a 3-dimensional vector with the dissimilarities values (normalized between 0 and 1).

The quality of the system is assessed using classification algorithms in Weka\footnote{http://www.cs.waikato.ac.nz/ml/weka/}. The classifiers used were RandomForest, Support Vector Machines, simple logistic regression, k-star and Bayesian Network) and to train the classifiers, five different subsets of 10,000 songs randomly chosen were used. To evaluate their solution, they tested it in a dataset composed by 2125 tracks, and measured the accuracy of the results, that was 88.46%.

### 3.10 A Heuristic for Distance Fusion

Degani, Dalai, Leonardi, and Migliorati (2013) address the problem of having multiple possible ways to determine the similarity between two musical pieces and they purpose a way to fuse them together, producing a single ranking value. For the fusion to be possible, the results given by each of the distance measures must be in the form of a matrix identical to the MIREX format for result evaluation. All the values must be then normalized to [0, 1] and an N-Dimensional space, were N stands for the number of similarity measures used, is created.
for each and every relation between two songs. This means that there will be as much created N-Dimensional spaces as there are cells in one matrix. The derived fusion distance is calculated and, the closest it is to the origin of the dimensional space, the better.

They experimented two audio features: pitch salience function and HPCP 12-bins. The two of these were combined with two different ways of computing similarity values: $Q_{max}$ and DTW. In their experiments, the best result was 72% in the Covers80 dataset, given by all the combinations of techniques that they implemented. Although it has low computational cost to compute the final distance, one must bear in mind that there is a high computational cost when producing the results just for a single distance measure.

### 3.11 Results Comparison

The best results produced by the described solutions are gathered in Table 4.3. The evaluation metrics are not consensual, which makes the task of comparing all of them impossible. Some works have implemented two versions: one for the MSD and one for their own dataset. In those cases, the results obtained in their own datasets, where better on paper, but those datasets are smaller and so there is no way of proving that they would outperform the ones tested on the MSD. In a way, this shows that the MSD limits the researchers and forces them to produce different datasets to test their work. Some need the raw audio file to extract different features or prefer to use chroma features with more dimensions, which none are provided by the MSD.

Of all the systems that were tested with the MSD, the work of Humphrey, Nieto, and Bello (2013) was one that achieved the best MAP, but it was one of those that needed more time to perform a query. It is clear that a compromise must be made between quality and speed. In the
end, it is also clear that there is a greater focus on trying to produce results in the least amount of time instead of providing the best results.
<table>
<thead>
<tr>
<th>Section</th>
<th>Features</th>
<th>Similarity Computation</th>
<th>Accuracy</th>
<th>AP@n</th>
<th>MAP</th>
<th>AR</th>
<th>Recall@n</th>
<th>Tracks</th>
<th>Secs/Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAST for Audio Sequences Alignment</td>
<td>HPCP 36-bins</td>
<td>Altered</td>
<td>-</td>
<td>-</td>
<td>30.11%</td>
<td>-</td>
<td>-</td>
<td>2,514</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>PCP 12-bins</td>
<td>Needleman-Wunsch algorithm</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,000,000 (MSD)</td>
<td>16.9</td>
</tr>
<tr>
<td>Chord Profiles</td>
<td>Chord profiles</td>
<td>$L_1$ distance</td>
<td>-</td>
<td>-</td>
<td>20.62%</td>
<td>1,378</td>
<td>-</td>
<td>24,282</td>
<td>1.46</td>
</tr>
<tr>
<td></td>
<td>PCP 12-bins + Chord profiles</td>
<td>Levenshtein distance</td>
<td>-</td>
<td>-</td>
<td>3.71%</td>
<td>114,951</td>
<td>-</td>
<td>1,000,000 (MSD)</td>
<td>7.56</td>
</tr>
<tr>
<td>2D Fourier Transform Magnitude</td>
<td>PCP 12-bins</td>
<td>Euclidean Distance</td>
<td>-</td>
<td>-</td>
<td>1.99%</td>
<td>173,117</td>
<td>-</td>
<td>1,000,000 (MSD)</td>
<td>3~4</td>
</tr>
<tr>
<td>Data-Driven and Discriminative Projections</td>
<td>PCP 12-bins</td>
<td>Linear Discriminant Analysis</td>
<td>-</td>
<td>-</td>
<td>13.41%</td>
<td>343,522</td>
<td>-</td>
<td>1,000,000 (MSD)</td>
<td>9~12</td>
</tr>
<tr>
<td>Cognition-inspired Descriptors</td>
<td>HPCP 12-bins</td>
<td>Multiple Sequence Alignment</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.563% (n=5)</td>
<td>160 (Covers80)</td>
</tr>
<tr>
<td>A Heuristic for Distance Fusion</td>
<td>Salience function + HPCP 12-bins</td>
<td>DTW + $Q_{max}$</td>
<td>-</td>
<td>-</td>
<td>72%</td>
<td>-</td>
<td>-</td>
<td>160 (Covers80)</td>
<td>-</td>
</tr>
<tr>
<td>Approximate Nearest Neighbors</td>
<td>PCP 12-bins</td>
<td>Approximate Nearest Neighbors</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.500% (n=50)</td>
<td>160 (Covers80)</td>
</tr>
<tr>
<td>Chroma Codebook with Location Verification</td>
<td>PCP 12-bins centroids (audio words)</td>
<td>$Q_{max}$</td>
<td>-</td>
<td>-</td>
<td>80%</td>
<td>(n=30)</td>
<td>-</td>
<td>43,392</td>
<td>-</td>
</tr>
<tr>
<td>Tonal Representations for Music Retrieval</td>
<td>HPCP 12-bins + Melody + Bass line</td>
<td>$Q_{max}$</td>
<td>88.46%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2125</td>
<td>-</td>
</tr>
</tbody>
</table>

Not all solutions present the same evaluation metrics. AR stands for Average Rank and only informs of the average rank of the first correctly identified cover in all queries.
Experimental Setup

Taking into account the previous work and what has been done so far, an idea for an audio cover song identification system was created. The goal was to use a system as baseline and, given the results, determine what could be done in order to improve them. To evaluate the system, a dataset is required, and it would be best if it replicates the conditions of the MIREX task. To determine what improvements could be made, the results from the system would have to be analyzed manually and, given the patterns discovered, determine what is lacking in the baseline and figure out what are the best methods available to implement it.

4.1 Baseline

The system chosen to serve as baseline was developed by Ellis and Poliner (2007) that finished in first place in the MIREX’2006 Cover Song Identification Task\(^1\). This project is freely available online\(^2\) and it is implemented in MATLAB. This was the system chosen because it is the only one available online that participated, and won, the MIREX task.

The cover identification process begins by creating single beat-synchronous 12-bin chroma features with 2-4 columns per second. The beats are estimated from the autocorrelation of an "onset strength envelope" derived from a Mel-frequency spectrogram (Ellis 2007).

The second step consists of using the Fourier Transform with 93ms overlapping time win-

\(^1\)http://www.music-ir.org/mirex/wiki/2006:Audio_Cover_Song_Identification_Results
\(^2\)http://labrosa.ee.columbia.edu/projects/coversongs/
dows to emphasize energy from a couple of octaves above and below 400Hz, then filtered in order to select tonal concentrations, which can be found by looking for consistent instantaneous frequency across adjacent bins.

The similarity between songs is obtained by cross-correlating the entire beat-chroma representations since finding exact matches end-to-end is very unlikely due to structural alterations. The cross-correlations are normalized by dividing by the column count of the shorter matrix. The columns are normalized so that the correlation scores are between 0 and 1. After cross-correlation, the results pass through an high-pass filter to distinguish genuine matches from incidental high cross-correlations. To account for key transpositions, circular cross-correlation is used, which is simply applying cross-correlation to every possible transposition. The similarity value is found by using the largest raw peak cross-correlation found in all cross-correlations.

4.2 Dataset

The dataset chosen was the CODA dataset (Coelho, Duarte, and Matos 2015). CODA is a dataset created to replicate the ones used in the MIREX competition. Therefore, the dataset is composed by 30 original tracks, each with 10 covers, and another 670 with no relation with the originals. The songs are provided as MP3 files. This was the dataset chosen because it was the only one available that provided the files in a format that was possible to listen to, and since it is not legal to provide music due to copyright laws, there are no other alternatives online. Being able to listen to the files plays a major role because the songs need to be manually analyzed to determine the qualities and faults of the audio cover song identification system.
4.3 Experiments

Having the system and the dataset, results could be had. By checking the results in the Top10 of each query, it was possible to determine which were the covers that failed to be identified and the non-covers that were incorrectly identified. Some of those tracks were manually analysed and it was possible to observe some patterns. Most of the non-covers identified reflected the instrumentation used by the query. If, for example, the query song had mainly a classic guitar playing, the non-covers identified would most likely have a similar instrumentation with classic guitar. The covers that failed to be identified, on the other hand, showed that a large amount of them presented the exact same lyrics, rhythm, and the same melody, although most of them had different genre or instruments.

Considering these faults, some experiments were conducted that explored the use of the features present in the covers that failed to be detected. To merge the results from the experiments with the ones already had from the baseline, the metrics fusion method suggested in Section 3.10 was applied. The experiments conducted were related to rhythm, melody, and summarization, although the latter was made initially without checking the results from the baseline. The goal was to determine if summarizing a song would preserve its most important information so that the cover detection results would not be affected while being produced in a faster way.

4.3.1 Summaries

As previously mentioned, creating summaries of songs was explored to check if it was a good alternative to the original audio files. If the information held in the summaries was enough to identify the original song, they could be used as the dataset and provide results in
a much faster way. The summaries were created using the work of (Raposo 2014) that applied unsupervised algorithms for speech and text to music audio files. The algorithms used for this matter were GRASSHOPPER, LexRank, Latent Semantic Analysis (LSA) and Maximum Marginal Relevance (MMR).

The configuration parameters used were: 0.5 for Frame Size and Hop Size for all; the Vocabulary Size was 50 for MMR and 25 for the rest; Sentence Size was 5 for LexRank and 10 for the rest. The Weighting parameter was BINARY for GRASSHOPPER and LSA while LexRank and MMR had DAMPENED_TF. The \( \lambda \) value used in MMR was 0.7. Each of these algorithms were produced in two manners: either as chroma features or as 20-bin vector MFCC’s.

4.3.2 Rhythm

One of the observations made by manually analyzing the covers that failed to be in top 10 was that those tracks had similar rhythmic features with the original. Since that similarity was not detected, some sort of rhythmic features could be beneficial and improve the detection of those covers. This problem has already been addressed by some, such as Mitri, Ciesielski, and Uitdenboergerd (2004). In their work, an approach for rhythm extraction was presented that consisted in applying the Fourier Transform to the audio signal and then split it into low (0-100Hz) and high (12000-14000Hz) frequency bands. Twenty-six rhythmic features are then extracted.

The approach used was based on the rhythmic features extracted in Antunes, de Matos, Ribeiro, and Trancoso (2014) which is based of Mitri, Ciesielski, and Uitdenboergerd (2004). The main difference lies in the selection of 18 rhythmic features instead of 26. These features were extracted with the MIRToolbox created by Lartillot and Toiviainen (2007), a MATLAB framework with a large collection of musical feature extractors. The process was as follows:
• Compute the Fourier Transform for every 50ms frame, half-overlapping, for the low and high frequency bands. The frequency bands were the ones used in the original (Mitri, Ciesielski, and Uitdenbogerd 2004).

• Extract the rhythmic features presented in Table 4.1.

• Merge the vectors of the two bands into a single one.

• Construct a matrix with all similarity values between all rhythmic features. The similarity value was obtained by computing the Euclidean distance.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxampv_in</td>
<td>Highest average power along time</td>
</tr>
<tr>
<td>minampv_in</td>
<td>Lowest average power along time</td>
</tr>
<tr>
<td>count_80v_in</td>
<td>Number of times power of the band is above 80%. maxampv_in</td>
</tr>
<tr>
<td>count_15v_in</td>
<td>Number of times power of the band is above 15%. maxampv_in</td>
</tr>
<tr>
<td>count_maxv_in</td>
<td>Number of times the power values are above maxampv_in</td>
</tr>
<tr>
<td>count_minv_in</td>
<td>Number of times the power values are below minampv_in</td>
</tr>
<tr>
<td>meanv_indist</td>
<td>Mean Distance between peaks</td>
</tr>
<tr>
<td>stdv_indist</td>
<td>Standard deviation of distance of peaks</td>
</tr>
<tr>
<td>maxv_indist</td>
<td>Maximum distance of peaks</td>
</tr>
</tbody>
</table>

For example, the rhythmic features extracted for the song *Satisfaction* - Rolling Stones, had a similar representation to the following vector:

\[43.4][0][11921][43558][8198][0][2.8][8.5][359][0][8][0][2216][84679][1029][0][61.8][282.6][1850]\n
where the first nine elements are relative to the low frequency bands and the other remaining nine elements describe the rhythmic features of the higher frequency bands.

### 4.3.3 Melody

To improve the detection of covers that present the same lyrics or similar melodic line, the work of Salamon and Gómez (2012) was considered. The task of extracting the melody consists
of 1) calculate when the melody is present and when it is not, which can be referred to as voice
detection and 2) estimating the correct pitch value of the melody when it is present.

The approach used is comprised of four stages: sinusoid extraction, salience function, con-
tour creation, and melody selection. The goal of the first step is to analyze the audio signal and
find out which frequencies are present at every point in time and enhance frequencies that are
more perceptually sensitive to human listeners using an equal loudness filter. Next, the sig-
nal is split into small blocks that represent a specific moment in time and the Discrete Fourier
Transform (DFT) is applied to each block, which gives us the spectral peaks, that are the most
energetic frequencies at that moment and all the other frequencies are discarded.

The salience function stage, estimates how salient those selected frequencies are. In order
to so, harmonic summation is used, which is searching for harmonic series of frequencies that
would contribute to the perception of a correspondent pitch. The weighted sum of the energy
of the harmonic frequencies is the salience of the pitch. The result is a representation of pitch
salience over time, which is referred to as salience function.

The resulting salience function is used to track pitch contours. A pitch contour is a series of
consecutive pitch values continuous in both time and frequency. The contours are tracked by
taking the peaks of the salience function and use a set of cues based on auditory streaming to
group them into contours. These contours can have a duration of a single note or even a short
phrase.

The final stage is selecting the melody. This step does not consist of extracting all melody
contours but rather filter out all non-melody contours and this is done by having a set of filtering rules. The result is a sequence of fundamental frequency (F0) values that correspond to the perceived pitch of the main melody.
4.3. EXPERIMENTS

The aforementioned approach is available through MELODIA\(^3\), a Vamp\(^4\) plugin, which can be used in a Vamp host program, such as the case of Sonic Annotator\(^5\). All the melodies were extracted for the CODA dataset in the form of a Comma-Separated Values (CSV) file, with two fields: a timestamp for every 2.9ms and the corresponding fundamental frequency value in Hertz. The frequency values can be relative to voiced or non-voiced segments whereas the latter is represented with zero or negative values.

The first processing step was quantising all the frequencies into a semitone representation (C, V, D, W, E, F, X, G, Y, A, Z) where V stands for C#, W for D#, X for F#, Y for G# and Z for A#. The length of the sequence was then shortened to reduce the computation time of the matching algorithm and to reduce the influence of rapid pitch changes which are likely to be performance specific. To do so, every 150 frames of 2.9ms were summarized by producing a pitch class histogram and selecting the bin with the highest occurrences. This strategy was based on the work of Salamon, Serrà, and Gómez (2013). The result will be a string representation for every song that is sequence of the respective character of the semitones returned by the histogram until the end of the song.

Matching the melodies was accomplished with the Smith-Waterman algorithm with changes to the weighted distance matrix and the gap penalty, in order to be able to work with the melody representation. These changes were suggested by Frey (2008). The distance values of the weighted distance matrix were chosen based on how further the semitones are from each other. The weighted distance matrix is shown in Table 4.2 and the gap penalty used was -8. The distance value returned for each match was used to construct the distance matrix between all melodies.

\(^3\)http://mtg.upf.edu/technologies/melodia
\(^4\)Vamp is an audio processing plugin system that allows the use of plugins to extract information from audio data.
\(^5\)http://www.vamp-plugins.org/sonic-annotator/
Two melody representations were tested. One without negative values and another with both positive and negative. In both cases, the zero values were not considered, since it could result in high matching values due to long periods of silence.

### 4.4 Results and Discussion

Overall, three kinds of experiments were made with different aspects and approaches. The results obtained are presented in Table 4.3 with the corresponding evaluation metrics. The results obtained by the baseline are also included to serve as a basis for comparison. The results obtained by fusing different approaches, using the proposed solution discussed in section 3.10, are also shown.

The results relative to the summarization and rhythm experiments proved to produce poor results, with the best score obtained by a summarization method being 75 successfully identified covers and 44 by the rhythm approach. In some way, this was expected because these approaches create a short representation, meaning that information useful for cover detection can be lost while selecting the values. An interesting observation was made when the results
4.4. RESULTS AND DISCUSSION

Table 4.3: Results of all experiments

<table>
<thead>
<tr>
<th>Features</th>
<th>Total @Top10</th>
<th>Average @Top10</th>
<th>MAP</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (MIREX’2006)</td>
<td>761</td>
<td>2.310</td>
<td>-</td>
<td>0.490</td>
</tr>
<tr>
<td>Baseline (CODA)</td>
<td>1049</td>
<td>3.179</td>
<td>0.304</td>
<td>0.647</td>
</tr>
<tr>
<td>Baseline + GRASSHOPPER-Chroma</td>
<td>610</td>
<td>1.848</td>
<td>0.140</td>
<td>0.291</td>
</tr>
<tr>
<td>Baseline + Rhythm</td>
<td>1015</td>
<td>3.076</td>
<td>0.294</td>
<td>0.660</td>
</tr>
<tr>
<td>Baseline + Positive-only melody (PO-Melody)</td>
<td>1119</td>
<td>3.391</td>
<td>0.332</td>
<td>0.701</td>
</tr>
<tr>
<td>Baseline + All-values melody (AV-Melody)</td>
<td>1121</td>
<td>3.397</td>
<td>0.333</td>
<td>0.701</td>
</tr>
<tr>
<td><strong>Baseline + PO-Melody + AV-Melody</strong></td>
<td><strong>1158</strong></td>
<td><strong>3.509</strong></td>
<td><strong>0.348</strong></td>
<td><strong>0.732</strong></td>
</tr>
</tbody>
</table>

of these experiments were fused with the baseline results. Despite having had worse results by itself compared to most of the summarization methods, fusing the baseline with the rhythm resulted in a less significant loss of quality when compared to the best summarization method. This can mean that the rhythm representation is, in some way, connected to its original song but not in the best way. One can also deduce that the presence of similar rhythms on its own is not a good indicator of being a cover or not. Due to the poor quality of the results of these approaches, they were not included in further tests.

Melody-wise, as previously mentioned in section 4.3.3, two representations were made: one with only positive values and another considering positive and negative values. The results revealed that the positive-only representation is the best on its own. However, fusing with the baseline scores seems to favor, although slightly, the representation with all values. By combining these two representations and the baseline, the best result of the experiments was achieved, suggesting that the two melody representations interact well with each other. The difference between the two representations and why they concentrate on different characteristics is explained by the quantization made using histograms of 2.9ms. With longer length and more values on the positive and negative representation, the octave with most occurrences will most likely be different, therefore highlighting different characteristics than the positive-only
one.

To further understand the influence of each representation, data relative to the difference in covers being detected among approaches was collected. That data is shown in Table 4.4 and consists of counting what are the common correctly identified covers (common true positives), the shared non-covers that were incorrectly identified (common false positives), the number of covers that were initially identified and then were lost (lost true positives), the non-covers that initially were not present in the Top10 and then were incorrectly identified there (new false-positives), and the total number of new covers successfully identified (new true positives) with the improved similarity matrix. The tests made were:

- **T1**: Baseline results vs. fusion of the baseline with positive-only melody.
- **T2**: Baseline results vs. fusion of the baseline with all-values melody.
- **T3**: Baseline results vs. fusion of the baseline with all-values melody and positive-only melody
- **T4**: Comparison between all-values melody and only-positive melody.

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common covers (TP)</td>
<td>983</td>
<td>1003</td>
<td>947</td>
<td>236</td>
</tr>
<tr>
<td>Common non-covers (FP)</td>
<td>1582</td>
<td>1745</td>
<td>1207</td>
<td>461</td>
</tr>
<tr>
<td>New covers (new TP)</td>
<td>136</td>
<td>118</td>
<td>211</td>
<td>-</td>
</tr>
<tr>
<td>Lost covers (lost TP)</td>
<td>66</td>
<td>46</td>
<td>102</td>
<td>-</td>
</tr>
<tr>
<td>New non-covers (new FP)</td>
<td>41</td>
<td>1</td>
<td>79</td>
<td>-</td>
</tr>
</tbody>
</table>

Although T1 showed that the positive-only melody representation has a higher count of new covers identified, it also contributed with more false positives and had fewer correctly identified covers in common with the baseline. T2 in the other hand, despite having detected
fewer new covers, revealed that the all-values representation preserves the quality of the results of the baseline, with a higher count of common covers and only one new false positive. To further prove that effect, the correlation coefficient values were estimated between the similarity matrices obtained from each approach. These results are included in Table 4.5. This confirmed that the matrix obtained from the baseline has a higher correlation coefficient with the all-values melody representation, explaining why the fusion between it and the baseline reveals a high count of common covers.

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Positive-only Melody</th>
<th>All-values Melody</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>-</td>
<td>0.2463</td>
<td>0.3890</td>
</tr>
<tr>
<td>Final</td>
<td>0.8142</td>
<td>0.7170</td>
<td>0.7779</td>
</tr>
</tbody>
</table>

The combination of both melody representations and the baseline, as shown by T3, hinders the quality of the results provided by the baseline but, on the contrary, seems to strengthen the results of both melody approaches. Since the three matrices are fused together, all of them are considered equally important and, individually, the results from both melody representations are worse than the baseline. Therefore, it is expected that the number of common covers would decrease and the count of new true positives would increase, as well as the number of false positive. Such is the case, but the new covers vs. lost covers ratio is good enough to have better results compared to the baseline.

The correlation coefficient values for this 3-method fusion matrix and each of the composing matrices were also calculated in order to assess the contribution of each of them in the final result. The baseline results were the ones with higher correlation, as it would be expected, followed by the all-values melody representation and, with the lowest correlation coefficient, the positive-only melody representation.
With the objective of trying to understand the impact of fusing the results and what changes were made to those achieved with the baseline, some of the changes in the Top10s were analyzed. Specifically, some songs were listened to and compared to see if there was any pattern that could be identified. The songs that were listened were new covers being successfully identified, covers that were present in the baseline but were lost in the best method and songs that were included in the Top10 that are not covers.

The new covers being detected consist of songs that are, in general, very closely related and easily detected by a human judge. There were compatibility between songs of different genres and styles but possess a similar melodic line and/or lyrics. This means that the goal of using melody extraction was met. There was also cases of matches between pairs of songs in which one of them did not have any lyrical elements or completely different instruments. There was an interesting case between a song that was entirely instrumental and the other one was in acapella style that were correctly identified as covers due to its similar melody.

The covers that were lost by the fusion process presented two patterns. One was related to the singing style of the performer, that deviated from the query song and the other consisted of different tempos, either in singing or instrumentation. Since the melody is extracted every 2.9ms, differences in tempo will result in different frequencies being detected.

There was no clear pattern in the songs incorrectly identified. Since there is a high number of false-positives being detected by the melodies alone, it is understandable that some would benefit from it and have their scores mistakenly boosted.

Having found an improvement relative to the results in the baseline, it can be assumed that features about melody and voice are useful to the task of identifying audio cover songs by using a fusion heuristic. Using a fusion heuristic allows the inclusion of different audio features, since
it is a fast and scalable method that requires the matrices to be computed beforehand only once. One alternative version of the fusion heuristic suggested by Degani, Dalai, Leonardi, and Migliorati (2013) was made that had a minor change in how the distance was computed: instead of using the suggested refined distance value, the sum of the three distance values was computed; it was observed that the difference was very slim, with only three less covers being detected in total. While it can be argued that it is less complex and requires less computation effort, the original solution is still better and still fast to compute for, at least, three 1000x1000 matrices.

4.5 Summary

This chapter presented a description of the system chosen to serve as baseline, the dataset used, and the experiments made. The experiments conducted to improve the quality of the baseline were related to summarization, rhythm, and melody. The results obtained from those experiments determined that summarization and rhythm were detrimental to the baseline. Two types of melody descriptors were created. One taking into account only positive values (voiced segments) and another with all the values (voiced and non-voiced segments). Both of them increased the quality of the results but the best method was achieved by combining the baseline with both the melody descriptors, obtaining an increase of 5% in MAP and 109 more covers being detected.
This thesis reviewed the musical facets that condition the process of identifying an audio cover song and presented the most common approaches used to take advantage of those facets. A baseline was used and its results analyzed. In a general sense, it was able to correctly identify covers with similar instrumentation, and failed to detected those with the different instrumentation but with similar lyrics, rhythm, or melody. Experiments relative to summarization, rhythm, and melody, were realized in order to improve the identification of the covers that failed to be detected by the baseline. For the melody, two representations were made. One considering only voiced segments, and the other considering voiced and non-voiced. The matching of melody descriptors was accomplished using a modified Smith-Waterman algorithm to handle semitone representation.

While the summarization and rhythm experiments displayed no improvement, the fusion of the results from both melody descriptors with the results from the baseline, resulted in an increase of 5% in MAP and 109 more covers being detected. The new covers being detected displayed similar melodic lines and/or lyrics to the query song, while having a different instrumentation or genre. This proved that 1) the heuristic for distance fusion does enhance qualities from various methods, 2) melody is beneficial to cover song identification, and 3) voice related descriptors, such as the melody descriptors used, are useful to detect covers with similar lyrics.

Future work includes finding a way to improve the heuristic in order for it to not impair the good results of the baseline. Some of the covers were affected by the poor value returned
by the melody and so they were not detected in the improved version. The melody descriptors were also affected by different tempos, and thus, making them tempo invariant would improve their quality.


