Department of Computer Science
Series of Publications A
Report A-2016-1

Cover Song Identification Using Compression-based Distance Measures

Teppo E. Ahonen

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium CK112, Exactum, Gustaf Hällströmin katu 2b, on April 1st, 2016, at twelve o'clock noon.

University of Helsinki
Finland

Supervisors
Kjell Lemström, University of Helsinki, Finland
Esko Ukkonen, University of Helsinki, Finland

Pre-examiners
Juan Pablo Bello, New York University, USA
Olli Yli-Harja, Tampere University of Technology, Finland

Opponent
Petri Toiviainen, University of Jyväskylä, Finland

Custos
Esko Ukkonen, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: http://cs.helsinki.fi/
Telephone: +358 2941 911, telefax: +358 9 876 4314

ISSN 1238-8645
ISBN 978-951-51-2025-0 (paperback)
ISBN 978-951-51-2026-7 (PDF)
Computing Reviews (1998) Classification: H.3.3, E.4, J.5, H.5.5
Helsinki 2016
Unigrafia

Cover Song Identification Using Compression-based Distance Measures

Teppo E. Ahonen

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
teahonen@cs.helsinki.fi

PhD Thesis, Series of Publications A, Report A-2016-1
Helsinki, March 2016, 122+25 pages
ISSN 1238-8645
ISBN 978-951-51-2025-0 (paperback)
ISBN 978-951-51-2026-7 (PDF)

Abstract

Measuring similarity in music data is a problem with various potential applications. In recent years, the task known as cover song identification has gained widespread attention. In cover song identification, the purpose is to determine whether a piece of music is a different rendition of a previous version of the composition. The task is quite trivial for a human listener, but highly challenging for a computer.

This research approaches the problem from an information theoretic starting point. Assuming that cover versions share musical information with the original performance, we strive to measure the degree of this common information as the amount of computational resources needed to turn one version into another. Using a similarity measure known as normalized compression distance, we approximate the non-computable Kolmogorov complexity as the length of an object when compressed using a real-world data compression algorithm. If two pieces of music share musical information, we should be able to compress one using a model learned from the other.

In order to use compression-based similarity measuring, the meaningful musical information needs to be extracted from the raw audio signal data. The most commonly used representation for this task is known as chromagram: a sequence of real-valued vectors describing the temporal tonal content of the piece of music. Measuring the similarity between two chromagrams effectively with a data compression algorithm requires further processing to

extract relevant features and find a more suitable discrete representation for them. Here, the challenge is to process the data without losing the distinguishing characteristics of the music.

In this research, we study the difficult nature of cover song identification and search for an effective compression-based system for the task. Harmonic and melodic features, different representations for them, commonly used data compression algorithms, and several other variables of the problem are addressed thoroughly. The research seeks to shed light on how different choices in the scheme attribute to the performance of the system. Additional attention is paid to combining different features, with several combination strategies studied. Extensive empirical evaluation of the identification system has been performed, using large sets of real-world music data.

Evaluations show that the compression-based similarity measuring performs relatively well but fails to achieve the accuracy of the existing solution that measures similarity by using common subsequences. The best compression-based results are obtained by a combination of distances based on two harmonic representations obtained from chromagrams using hidden Markov model chord estimation, and an octave-folded version of the extracted salient melody representation. The most distinct reason for the shortcoming of the compression performance is the scarce amount of data available for a single piece of music. This was partially overcome by internal data duplication. As a whole, the process is solid and provides a practical foundation for an information theoretic approach for cover song identification.

Computing Reviews (1998) Categories and Subject Descriptors:

H.3.3	Information Search and Retrieval
E.4	Coding and Information Theory – data compaction and compression
J.5	Arts and Humanities
H.5.5	Sound and Music Computing – signal analysis, synthesis, and processing

General Terms:

information retrieval, similarity measuring, data compression, data quantization

Additional Key Words and Phrases:
music information retrieval, normalized compression distance, cover song identification

Acknowledgements

First and foremost, I need to address my supervisor, adjunct professor Kjell Lemström. This thesis would have never seen the light of day without his encouragement, and he never seemed to lose faith in my work, even when I myself had doubts.

Second, I wish to thank my other supervisor, professor Esko Ukkonen, who came in at the later stages of this work and had an immeasurable impact on the quality of the thesis.

The pre-examiners of this thesis, associate professor Juan Pablo Bello and professor Olli Yli-Harja, provided insightful feedback, for which I am very grateful.

During my years as a PhD student, I was funded by various sources. First, the Academy of Finland project Content-Based Retrieval and Analysis of Harmony and other Music Structures (C-Brahms), then the Helsinki Graduate School of Computer Science and Engineering (Hecse), and finally the Finnish Centre of Excellence for Algorithmic Data Analysis Research (Algodan). Thank you for making this research possible.

The Department of Computer Science has been a supportive and motivating environment for the course of my doctoral studies. I would like to thank especially Marina Kurtén for her valuable help with the grammar, with this thesis and elsewhere.

During my studies I have had the opportunity to have fruitful and interesting discussions with various bright minds. To name but a few, these include Väinö Ala-Härkönen, Mikko Apiola, Antti Laaksonen, Mika Laitinen, Simo Linkola, and Lari Rasku.

Finally, I would like to thank my parents Inkeri and Yrjö, my brothers Mika and Petri and their families, my other relatives, and all my dear friends, especially anyone whom I have had the pleasure and the privilege of playing music with. Last but definitely not the least, I must express my gratitude to Päivi for her endless love, support, patience, and grace during the time I have spent on this thesis.

Helsinki, March 2016
Teppo E. Ahonen

Contents

1 Introduction **1**
- 1.1 Content-Based Music Information Retrieval 2
- 1.2 Cover Song Identification 4
- 1.3 Research Questions 5
- 1.4 Thesis Outline 6

2 Tonal Features **7**
- 2.1 Chroma and Chromagram 7
  - 2.1.1 Chromagram Calculation 9
- 2.2 Chromagram and Musical Facets 11
  - 2.2.1 Beat-synchronization 11
  - 2.2.2 Key Invariance and Chromagram Data 12
- 2.3 Chromagram Data and Cover Song Identification 14
  - 2.3.1 Discrete Representations 14
  - 2.3.2 Continuous Representations 16
- 2.4 Large-scale Cover Song Identification 18

3 Compression-based Distance Measuring **21**
- 3.1 Normalized Information Distance 21
- 3.2 Normalized Compression Distance 22
  - 3.2.1 Observations on Normalized Compression Distance . 23
- 3.3 Compression-based Methods in MIR 24
  - 3.3.1 Symbolic Domain 24
  - 3.3.2 Audio Domain 26

4 Composition Recognition and Invariances **29**
- 4.1 Basis 29
- 4.2 Musical Examples 30
  - 4.2.1 Essential Musical Characteristics 30
- 4.3 Global Invariances 32
  - 4.3.1 Tempo Invariance 33

4.3.2 Key Invariance 35
4.3.3 Tuning Invariance 38
4.3.4 Structural Invariance 39
4.4 Local Invariances 42
4.4.1 Melodic Invariance 43
4.4.2 Arrangement Invariance 44
4.5 Similarity Values 46
4.5.1 Classification Experiment 47
4.5.2 Difficult Cases 50
4.6 Remarks 51
4.6.1 Compression-based Similarity 53

5 Retrieval Experiments 55
5.1 Evaluation setup 55
5.1.1 Test Data 55
5.1.2 Evaluation Measures in Identification and Retrieval 57
5.2 Identification Experiments 59
5.2.1 Effect of Compression Algorithm Selection 59
5.2.2 Effect of Chromagram Parameters 61
5.2.3 Invariances 64
5.2.4 Feature Representations 68
5.2.5 Internal Duplication 73
5.3 Summary of the Chapter 76
5.3.1 Computational Costs 77
5.3.2 Comparison with State of the Art 79

6 Feature Combination 85
6.1 Motivation 85
6.2 Melody Estimations 86
6.3 Combination Strategies 89
6.3.1 Strategy One: Combination of Features 89
6.3.2 Strategy Two: Combination of Representations . . . 91
6.3.3 Strategy Three: Combination of Distances 92
6.4 Details and Analysis of Feature Combination 93
6.4.1 Adding a Feature 93
6.4.2 Distance Calculation 94
6.4.3 The Overall Effect of Combination 96
6.5 Summary of the Chapter 101

7 Conclusions **103**
7.1 Contributions 104
7.2 Discussion 106
7.3 Future Work 108

References **111**

A Yesterday Dataset **123**

B Summertime Dataset **125**

C The Mixed Dataset **127**

D The Electronic Appendix **147**

Chapter 1

Introduction

Our world is filled with music, and we consume music on a daily basis for different purposes. Be it relaxation, partying, rituals, background music for some activity, or perhaps an area of study, music is listened to in enormous amounts by individuals worldwide. Unquestionably, music has a significant importance for us. And in consequence, we are always looking for new methods for accessing music.

During the last few years, the format of consuming music has undergone a dramatic change. Whereas music used to be purchased in physical media, such as vinyl albums and compact discs, the current trend favors online distribution: net stores offering downloads (such as iTunes Store[1] or Amazon Music MP3[2], to name but a few) and stream-based distribution services (Spotify[3] being one of the most known at the moment) are currently more and more favored by end-users as their choice for accessing music. In consequence, a huge amount of music is nowadays stored on hard drives in different appliances, ranging from large servers and personal computers to laptops and various mobile devices. This leads to large amounts of diverse musical data, and by reason of this, a demand for managing such vast data sets has emerged. Browsing through endless directories and playlists is far from practical, and end-users are presumably expecting more efficient methods for accessing their collections of music from a more in-depth point of view than just managing a group of files; in other words, accessing music in terms of the music itself. But where accessing textual data by means of textual information retrieval is somewhat straightforward, the task is far less self-evident with music data. As the old proverb goes, discussing music is equivalent to dancing architecture.

[1]http://itunes.com/
[2]http://www.amazon.com/digitalmusic
[3]http://www.spotify.com/

1.1 Content-Based Music Information Retrieval

Music information retrieval (MIR) [88, 36] is a relatively new discipline that studies how information contained in musical data can be meaningfully extracted, analyzed, and retrieved by computational means. The nature of MIR is inheritably interdisciplinary, and the area of study can be seen combining at least various subfields of computer science (algorithmics, artificial intelligence, machine learning, data mining), musicology, signal processing, music psychology, acoustics, mathematics, statistics, and library sciences. This addresses how complicated an area of study MIR can be. Music can be approached and analyzed from various points of view, and even the way music is experienced varies.

The history of MIR dates back to the first proposals of automatic information retrieval. One of the very first articles and possibly the first to use the term *musical information retrieval* was an article by Kassler from 1966 [56]. Arguably, back then some of the ideas were slightly ahead of their time [27], as the technical limitations prevented applying the ideas in practice. Due to the increase of computational power and storage capabilities, managing music with computers became gradually more and more general, but the progress in the research was rather slow overall. It has been only in the past ten-plus-some years that the area of study has grown to be a distinguished and attractive subfield of its own. Nowadays, the group of MIR researchers has grown to an active, ever-expanding global community [38].

Most music data collections are manipulated through the metadata connected to the piece of music. Such metadata consists of piece-related information including the name of the artist, the title of the piece, and possibly other relevant information such as the name of the album containing the piece, or the year when the piece of music was initially published. Also, more descriptional metadata can be added. Such data consists of text-based features such as lyrics, genre labels, or so-called tags, short terms that in a free form describe observed characteristics of the piece (for example, a set of tags for a piece could be “slow“, “live recording”, “’90s“, “atmospheric”, ”piano music”, and ”female voice“). Trivially, retrieving music can be based on the metadata features, but the weakness of the metadata lies in the unreliable nature of it, caused by the human factor behind the metadata. The metadata could be incorrect, unobjective, indefinable, or completely missing, making all kinds of music categorization and retrieval more or less impractical. Also, there are very few standards of music metadata. For example, symbolic music, such as MIDI files, contains metadata different from the standard of audio MP3 format metadata. And even though projects such as MusicBrainz[4] strive to produce open-source databases of reliable metadata, they still lack the ability to describe objective, musically informative qualities of the music.

The lack of reliable metadata creates a demand for advanced methods that are based on the content of the pieces of music. The subfield of MIR studying such methods is known as content-based music information retrieval (CBMIR). In CBMIR, the focus is in the information contained in the music itself, whether it is audio-based features such as spectral information extracted from the signal data, or semantic information extracted from symbolic representation such as the musical score or transcription of the audio data. Successful CBMIR allows developing applications that can be used by not only the common end-users of music, but also by music industry and copyright organizations, and musicologists and other academic researchers.

A typical retrieval task in CBMIR can be described as follows. Given an excerpt of a piece of music as a query, match the excerpt to a larger database of music, and return a list of likely candidates, possibly ranked according to their similarity with relation to the query. The task of *query by example* is a good example of such retrieval, and also one of the most commercially successful areas of CBMIR; the widely used application known as Shazam[5][115, 116] is a prime example of a query by example system. Shazam matches audio excerpts to a database using technique known as audio fingerprinting. As matching complete audio pieces would be too laborious, the audio signal is turned into a spectrogram, and then the spectrogram is reduced to a sparse representation of the peaks in the spectrogram, which in turn are processed into hash fingerprints that allow efficient matching between two pieces of music, and the similarity is based on discovering matches in the hash locations. This straightforward process is robust against noise as well as other minor differences between the audio signals.

However, the methodology presented above enables matching only between pieces of music taken from the same audio source. This might not be a case for the end user, as the original audio recording might not be available. More versatile methodologies allow different kinds of user inputs and provide more complex similarity measures that allow more variation between the query and the candidate pieces while maintaining distinguishing power. One of these techniques is known as *query by humming* [45], where,

[4] http://musicbrainz.org/
[5] http://www.shazam.com/

as the name states, the query is given as a hummed or sung input by the end user. Here, the task is to match a short (usually monophonic) query melody to a dataset of music. Query by humming systems require robust matching and sophisticated representations; the end user might provide a melody segment that is only briefly similar to the correct piece of music.

1.2 Cover Song Identification

Both query by example and query by humming techniques fall short when the task is to distinguish different versions of a recorded composition. This task, commonly known as cover song identification [37, 75, 102] takes as input a piece of music and strives to match the composition of the query recording to the compositions of the recordings in the database. Here, the term *cover* is slightly misleading, as a cover version might as well be a live recording, a remixed version, or any other kind of an interpretation or a re-work of the originally published performance.

Detecting the same composition among pieces of music is usually trivial for a human listener: only a short segment of melody, a distinctive chord change, or familiar lyrics might be enough to reveal the composition to the listener. However, for a computer the same task is notably more difficult. Cover versions might differ from the original performances in various ways; for example, different versions might have different keys, tempi, arrangements, structure, and language of the lyrics, resulting in highly different spectral information in the pieces. In order to identify a version, a cover song identification system needs to focus on discovering compositional characteristics of the piece, and calculating the similarities between pieces in a manner that allows a great deal of variation in such features. Whereas a human listener might require only a short melodic cue for the identification, this does not seem like a suitable starting point for automatic cover song identification, as the identification system does not know what might be that important melody for the pieces. Therefore, the matching process should consider longer segments; usually, cover song identification is based on full-length recordings of the pieces in order to detect the similar segments and sequences between the pieces.

The difficult nature of the task makes it also highly rewarding. Successful cover song identification yields information on how the essential compositional characteristics in music can be captured, represented, and measured. In other words, cover song identification provides an objective way to measure compositional similarity, instead of relying on subjective criteria of musical similarity. Because of this, cover song identification has potential areas of practical applications; most notably, a cover song identification system could be applied for detecting plagiarism and other violations of intellectual property rights. Also, scholars of musicology might benefit from such applications and information, as well as any music listener, who would like to discover cover versions from a large collection of music.

1.3 Research Questions

This thesis addresses the problem of cover song identification. Our work focuses on using data compression in order to measure the similarity between the versions, namely a compression-based similarity metric known as normalized compression distance (NCD) is applied for the task. This metric, that defines similarity as the amount of information shared between two objects, has been applied for various domains, including music (e.g. [34, 72]), with notable success. The motivation for using NCD for this particular task comes from various advantages of the metric: the parameter-free nature, the so-called quasi-universality [33], and overall robustness of the metric make it a highly interesting choice for the task.

This work is based purely on audio signal data, and it focuses completely on the tonal content of the music. Our work is based on the so-called mid-level features extracted from the audio signal. The low-level audio signal features, such as timbral characteristics, do not provide information that could be applied for successful cover song identification, neither do we assume any high-level semantic information to be included in the similarity measuring process (such as information on what instruments are present in the arrangement). Our key source of tonal information will be the chromagram [15], a mid-level representation obtained from the audio signal that describes how the spectral energy of the piece is temporally distributed between the pitch classes of the western chromatic scale. The chromagram, highly robust against timbral changes in audio signal, is the most commonly used feature in cover song identification.

The purpose of this thesis is to present answers to the following questions. The related previous work of the author is referenced.

- Can normalized compression distance be efficiently applied in cover song identification [4, 5, 7, 8, 9]? We want to discover whether NCD-based methodologies can provide identification accuracies in par or better than the state of the art.

- What features play a crucial role in cover song identification when NCD is used as the similarity measure [8, 5, 9]? The information in the chromagram can be approached in various different ways, and we are interested in whether some of the musical invariances needed in similarity measuring can be obtained by using NCD.

- How should chromagram features be represented for such similarity measuring [8, 4, 7]? Considering the byte-level nature of a standard data compression algorithm, the continuous chromagram data values are problematic, and quantization needs to be conducted; however, the quantization and compressibility should not be obtained at the expense of identification accuracy.

- What other issues should be noted when applying normalized compression distance to cover song identification [4]? The pros and cons of compression-based similarity measuring will be reported and analyzed.

1.4 Thesis Outline

The outline of the thesis is as follows. First, in Chapter 2, we observe the chromagram representation, how it is computed, various features that can be extracted from it, how they have been applied for different tasks in MIR, and how different quantized representations for the chroma features can be computed. In Chapter 3, we apply data compression to measure similarity, study the normalized compression distance and the information theoretic background of the metric, and present an extensive review on how NCD has been previously applied for different tasks in CBMIR. In Chapter 4, we observe the task of cover song identification and how different kinds of required invariances can be obtained, and how important these invariances are in order to successfully perform the task. In Chapter 5, we experiment with compression-based similarity measuring for chromagram data, present wide-range identification evaluations, analyze the results of the evaluations, and suggest optimal parameter settings and component choices for the task. In Chapter 6, using information gained from the experiments in Chapter 5, we observe methodologies for combining different chromagram features for a higher accuracy in identification tasks. Finally, conclusions and potential directions for future work are presented in Chapter 7.

Chapter 2

Tonal Features

In this chapter, we discuss the concept of chromagram and describe how it can be extracted from raw audio signal data. Motivation for using chromagram features in content-based music information retrieval is discussed, focusing on how chromagram data has been applied in cover song identification.

2.1 Chroma and Chromagram

The tonal content of an audio signal can be extracted and represented as a feature called *chromagram* [114], also known as a pitch class profile (PCP) [43]. Commonly 12-dimensional, thus representing the 12 semitone pitch classes of the equally-tempered western musical system, a chromagram extracted from a piece of music is a sequence of continuous-valued vectors that describe how the spectral energy of the audio signal is distributed to pitch classes temporally.

A visualization of a chromagram excerpt is depicted in Figure 2.1. In this example, each frame of the chromagram represents 0.3715 seconds of music, with no overlapping between the frames. Thus, the 160 frames in the visualization depict approximately one minute of music from the beginning of the piece, roughly corresponding to the first verse and chorus sections of the piece.

The basis of chroma is that the *pitch* is perceived by the human auditory system as a combination of two features: *tone height* and *chroma*, as described in the 1960s by cognitive psychologist Roger Shepard. Pitch (p), the Hertz (Hz) value, can be factored into chroma ($c \in [0, 1]$, also known as pitch class) and tone height ($h \in \mathbb{Z}$, also known as octave number) [114] as

$$p = 2^{c+h}.$$

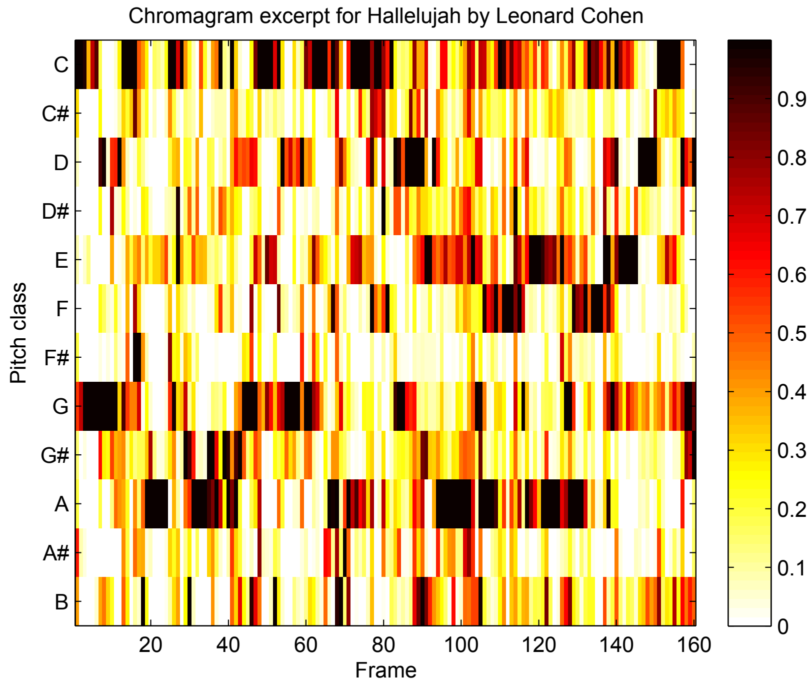

Figure 2.1: An illustration of a chromagram. Each frame depicts 0.3715 seconds of music. The darker the colour the higher the relative energy of the corresponding pitch class.

Pitch values share the same chroma class only if they are mapped to the same value of c. For example, 200, 400, and 800 Hz share the same chroma class as 100 Hz, but 300 Hz does not [114].

The chromagram extends the chroma with the dimension of time, thus describing the distribution of the chroma of the signal over time [114], resulting in a 12-dimensional time series. In addition to the 12 dimensions, a 24- or 36-dimensional chromagram can be used to capture the energy distribution in a finer resolution of $\frac{1}{2}$ or $\frac{1}{3}$ semitones. Using such representations has been shown to be able to provide better retrieval accuracies than the usual 12-dimensional representation [103], as they can help to manage slight tuning differences.

The chromagram captures the tonality of the piece: both the harmonic content and the melodic information is present in the chromagram, making it a highly applicable representation for various tasks in CBMIR. The list of CBMIR tasks where chromagram data is utilized (in addition to cover song identification) is vast, and to provide insight, here is just a brief list of such areas: genre-based audio classification (e.g. [92]), score alignment (e.g. [51]), key estimation (e.g. [113]) and a plethora of chord estimation algorithms (for a survey, see [89]).

2.1.1 Chromagram Calculation

Different approaches to produce the chromagram representation for a given audio signal exist, but the purpose and the basis of the method is similar: the audio signal is transformed to time-frequency domain using Fourier transform and the resulting components are mapped to the bins that correspond with the frequencies of the semitone pitches. In order to reduce the effect of noise and dynamics, the frames of the chromagram are usually normalized.

In [43], the definition for a pitch class profile is given as follows. Let $x(n)$ be a fragment of audio signal with a total length of N fragments, sampled with a frequency of f_s. The discrete Fourier transform X for the signal is calculated as

$$X(k) = \sum_{n=0}^{N-1} e^{-2\pi ikn/N} \cdot x(n),$$

where $k = 0, 1, \dots, N-1$ and $i = \sqrt{-1}$. Chromagram C is now calculated as

$$C(p) = \sum_{l \ s.t. \ M(l)=p} \|X(l)\|^2,$$

where $p = 0, 1, \dots, 11$ and M is a table which maps a spectrum bin index to the chromagram index:

$$M(l) = \begin{cases} -1 & \text{for } l = 0 \\ round(12 \log_2((f_s \cdot \frac{l}{N}/f_{ref})) \mod 12 & \text{for } l = 1, 2, \dots, N/2-1, \end{cases}$$

where f_{ref} is the reference frequency that falls into $C(0)$ and the term $(f_s \cdot \frac{l}{N})$ represents the frequency of the spectrum bin $X(l)$.

The chromagram can also be extracted using the constant Q transform [24], a close variant of the Fourier transform that uses logarithmically divided frequency bands instead of the linear bands of a common discrete time Fourier transform, thus dividing the spectrum to bands that correspond to the human ear [19]. An efficient implementation of the constant Q transform based on the Fast Fourier Transform exists [25]. In [19], the chromagram is obtained in the following manner. From audio signal x the constant Q transform X_{cq} is calculated as

$$X_{cq}(k) = \sum_{n=0}^{N(k)-1} w(n,k)x(n)e^{-2\pi i f_k n},$$

where w is the Fourier analysis window and N its length, both functions of the bin position k. Also, f_k is the center frequency of the k^{th} bin, defined

$$f_k = 2^{k/\beta} f_{\min},$$

where β is the number of bins per octave, and $f_{\min}$ the minimum frequency of the analysis. From X_{cq}, the chroma of a given frame is calculated as

$$C(p) = \sum_{m=0}^{M} |X_{cq}(p + m\beta)|,$$

where p is the chroma bin number, and M is the number of octaves.

Harmonic pitch class profile (HPCP) [46, 47] has been proposed as a more robust extension of the PCP representation. It allows a higher resolution than semitones, and the frequencies contribute not only to the nearest bin, but to several nearest bins, with a greater weight according to how near the bins are to the frequency. Also, as the name suggests, HPCP takes into consideration the harmonics of the pitches (i.e., for a pitch of frequency f, the harmonics of $2 \times f, 3 \times f$, and so on) that appear in the pitch class bins; to compensate, the harmonics of a pitch class are used to weight the values of its fundamental frequency. As a result, the HPCP is a more robust chroma representation for various chroma-related tasks.

Müller has suggested using frequency bands corresponding to musical notes for chromagram calculation [83]. The method decomposes the audio signal to 88 frequency bands that correspond to the pitches from notes *A0* to *C8*, thus describing the energies of these notes. Then, the octave-equivalent pitches are summed up, resulting in a 12-dimensional chroma representation. In addition, Müller has proposed further processing the chromagram in order to achieve more robustness; the Chroma Energy Normalized Statistics (CENS) method [85] quantizes the chroma bin values and smooths the data with statistical information, whereas the Chroma DCT-Reduced Log Pitch (CRP) method [84] removes timbral content from the chromagram data using discrete cosine transformation. Both have proved effective, CENS with classical music audio matching [85] and CRP with chord recognition and audio matching [84].

For our work, we use the implementation of MIRToolbox[1] [66, 67], version 1.3.4.

2.2 Chromagram and Musical Facets

The diversity of music is likely to be reflected in the extracted chroma features. To achieve a more robust chromagram representation, several steps of processing have been proposed.

2.2.1 Beat-synchronization

The chromagram is commonly calculated using a constant window length over an audio signal. This has several disadvantages, as different instruments, especially the percussive ones, create transients that appear as noise in the representations. When comparing chromagrams from different pieces of music, the problem becomes even more apparent, as the pieces in different tempi cause greatly different chroma profiles when extracted with a fixed window length, thus making matching and alignment highly difficult. This problem has been addressed using beat estimation methods. Beat estimation, also known as beat tracking, means analyzing the audio signal for an estimation of the location of the beats in the music, thus providing an estimate of the tempo of the piece. Several methods for the task exist; we refer an interested reader to an extensive survey of methods [62].

Although using beat-synchronous chroma features seems like a plausible idea for various chroma-related MIR tasks, this representation also has its own limitations. The beat-estimation can backfire and have an unwanted effect on the chromagram representations. This has been studied in cover song identification: although several results support the use of beat-synchronous chroma features [40], several other report for higher accuracies when using no beat estimation with chroma data [75, 16, 103, 5].

Whereas in chord detection the beat-synchronization is a highly workable idea, as the beat-synchronous chroma data does not suffer from the noise of chord transitions, beat-synchronous chord features in contrast fail to provide a higher accuracy in cover song identification [16]. This supports the notion made in [75] that the choice of the similarity measure is more important to the outcome of a cover song identification system than the selected feature robustness.

[1] https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox

2.2.2 Key Invariance and Chromagram Data

In pairwise similarity measuring between two chromagrams, the question of a common key between the pieces is important. Even two chromagrams of the same piece could be deemed highly dissimilar, if another of them would be transposed to a different key.

Estimating the main key from the chromagram data has been studied extensively, and several proposed unsupervised methods exist. These include various hidden Markov model (HMM) based techniques (e.g. [87, 91, 69]), a method that compares chroma profiles with key templates [54], and a technique that maps chroma-based tonality vectors to a coordinate of circle of fifths [53]. Therefore, when comparing two pieces of music, it would seem a practical idea to estimate the main keys of both pieces and then transpose the other respectively to achieve two chromagram profiles in a common key. Transposing a chromagram is trivial; all that is needed is to rotate the values of the chromagram bins.

However, key estimation is hardly a solved task. As with beat estimation, unsuccessful key estimation could lead to worse results. Considering that the best-performing key estimation method of the MIREX evaluation of 2014 reached a weighted key score of circa 0.83 [2], it would seem that the key estimation technique might still be unreliable.

Instead of estimating the keys and transposing, a method based on finding the optimal common key for two pieces has been proposed. The method is called Optimal Transposition Index (OTI) [103, 101], and it is calculated as a maximum dot product of semitone transpositions between global chromagrams. Formally, for a chromagram C, a feature called global chromagram G_C is calculated as

$$G_C = \frac{\sum_{i=0}^{N-1} C_i}{\max\{\sum_{i=0}^{N-1} C_i\}}, \tag{2.1}$$

where C_i is the frame i of the chromagram and N is the length of the chromagram. For two chromagrams, C_a and C_b, the OTI is now calculated as

$$OTI(C_a, C_b) = \underset{1 \le j \le M}{\arg\max}\{G_{C_a} \cdot \textsc{Circshift}(G_{C_b}, j-1)\}, \tag{2.2}$$

where M is the maximum of possible transpositions (for a semitone resolution chroma, this would be 12), and CIRCSHIFT is a function that rotates a vector j positions to the right. The OTI value is thus the amount of semitone transpositions needed to transpose one chromagram to the same key as the other.

[2] http://nema.lis.illinois.edu/nema_out/mirex2014/results/akd/

In [101], OTI was found to be a distinctly more suitable method for transposition than a state-of-the-art key estimation-based method, when applied for obtaining key invariance in pairwise chromagram similarity measuring in a cover song identification task. It is also more straightforward and computationally far less laborious than, for example, HMM-based key estimation techniques.

Several studies use key-invariant representations or measuring techniques in the process. With melody data, a common technique is to reduce the melody into melodic contour, or as a representation known as Parsons code [90]. Melodic contour describes the semitone difference between two subsequent notes, whereas Parsons code takes this even further by just describing whether the melody rises, descends, or stays in the same pitch. For chromagram data, an equivalent approach by Kim and Perelstein uses relative pitch changes obtained by taking for each frame a cross-correlation of 20 preceding frames with each 11 possible chroma intervals [61]. We proposed an OTI-based chroma contour in [7]. Other contour-like representations use chromagram data quantized as a chord sequence (see Subsection 2.3.1). Lee uses the most frequent chord of the sequence as an estimation of the key and transposes the chords accordingly [68]. In [8], we suggested a method that, after estimating the chord progression, represents the changes between subsequent chords; the changes are composed of the semitone differences between the root notes of the chords and whether there is a change from major to minor chord (or vice versa). As there are 12 possible semitone intervals and the possibility of the major/minor change, the chord sequence can thus be expressed with an alphabet of size 24.

Apart from these, there is always the possibility of applying a brute force approach by calculating the distances between each possible transposition. Such a method has been applied in several studies (e.g. [40, 60, 59]). The positive side is that the method will inevitably calculate the distance between the correct transpositions. The most obvious negative side is clearly a major growth in the computational time that will be required in the process. In [101], an observation was made that calculating only the two most likely OTI-based transpositions results in almost as good performance as the brute force approach, thus reducing the computational time needed by a factor of six.

2.3 Chromagram Data and Cover Song Identification

The task of cover song identification relies almost solely on chromagram data. Alongside chromagram similarity, the idea of applying mid-level melodic representations is suggested by several researchers. Marolt [77, 78] uses salient melodic fragments to describe pieces of music and measures the similarities between fragments using cosine similarity in [77] and locality-sensitive hashing in [78]. In [77] Marolt also uses chromagram data and a combination of both chromagram and melodic features, with the combined feature providing highest identification accuracy. In [78] it is stated that short melodic segments perform with better accuracies than short chroma segments, whereas longer chroma features might provide better accuracies as long as the song structures do not differ significantly. Tsai et al. [111] use estimation of the main melody of a piece of music as a feature in cover song identification, by first estimating and removing the non-vocal segments from the music, then selecting the strongest pitch of a time frame as a representative note and then using dynamic time warping to measure similarity between note segments. Apart from these, cover song identification is usually based on chromagram data or a feature extracted from chromagram (such as key templates in [55]); other spectral-based approaches are presented in [41, 117].

2.3.1 Discrete Representations

Several cover song identification techniques apply similarity measuring for sequences of symbols. Such methods require turning the multi-dimensional continuous chromagram data into a one-dimensional sequence of discrete symbols. Chromagram is essentially a multi-dimensional time series, and discretization of time-series data is a well-studied area of research. For chromagram data two methodologies stand out.

Vector quantization [44] (VQ) is a common technique for producing a symbolic representation from continuous data. When applied to chromagram data, the idea is to map the chroma vectors to prototype vectors, or codebook words, according to a distance metric (for example, Euclidian distance) and then represent the chromagram as a sequence of vector label characters, or in other terms, words of a codebook.

The k-means clustering method (for definition, see e.g. [97]) can be applied to the quantization procedure. In [28], k-means was used to turn the chromagram data to a string of characters. Then, the strings between different pieces of music were compared using exact string matching and edit distance. In addition, the symbol histograms and indicator strings (the unique symbols of the strings sorted lexicographically to short descriptors) were used. The k was experimented with values of 8, 16, 32 and 64. In [95], the authors report experimenting with several vector quantization methods, but k-means with $k = 500$ provided best results for their approach of text-based retrieval applied to chromagram-based retrieval. Further, in [23] online vector quantization was used to describe large amounts of audio data as codebook words for artist-based music retrieval. Perhaps not surprisingly, the authors discovered the most popular codebook words produced by the online VQ algorithm to represent the most common chords and single notes.

With cover song identification hidden Markov models (HMMs, see e.g [93, 97] for a tutorial) have been a widely used technique (e.g. [16, 69, 8, 4, 5]). The quantization is actually a chord estimation method suggested originally by Sheh and Ellis [106]. The chord estimation is based on using the chromagram frames as observations produced by states representing the 24 major and minor triad chords. Several ideas on HMM configurations and parameter selections exist: see [89] for a review and evaluation of several methods.

This thesis follows the methodology presented in [19]. Here, a 24-state fully connected HMM is used, with parameters initialized according to musical knowledge. The initial parameters are set as follows:

- Initial state distribution π: As there is no reason to favor any state before others, this is the same for each state (i.e. $\frac{1}{24}$).

- State transition matrix A: This is set according to a double-nested circle of fifths, meaning that triad chords that are closer to each other (i.e. share the same notes) are given higher probabilities. For C major chord, the highest transition probability is to the chord itself, $C \rightarrow C$. This value is $\frac{12+\epsilon}{144+24\epsilon}$. The next similar chords are A minor and E minor, both sharing two notes with C major, and the initial probabilities for both are $\frac{11+\epsilon}{144+24\epsilon}$. Eventually, the furthest chord for C major is $F\sharp$ major, with probability $\frac{0+\epsilon}{144+24\epsilon}$. The probabilities are set similarly to all states.

- Mean vector μ: The mean vectors are set by giving the value 1 to the pitch classes that are present in the corresponding chord, and 0 otherwise. For C major, the vector is $(1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0)$.

- Covariance matrix Σ: The covariance matrix for each state consists mainly of zeros. The diagonal is set to 0.2, apart from pitch classes present in the corresponding chord; these are set to 1. For non-diagonal matrix cells, the dominant of the root (i.e. the fifth) is set to 0.8, as is the dominant of mediant, whereas the mediant of the root (i.e. the third) is set to 0.6.

The model is then trained with the Expectation-Maximization algorithm, with only the initial state distribution and state transition matrix trained. After the model converges, the most likely path through the states is calculated using the Viterbi algorithm, this path thus presenting an approximation of the chord sequence of the piece. The 24-chord lexicon is likely too limited to produce an accurate chord transcription for a piece, but it is still a robust representation of the salient harmonic content of music [19]. Although more accurate chord transcription methods, such as [80], have been proposed, the limited representation would seem adequate for the identification process; too accurate transcriptions might even be restrictive, as they would not be robust against the tonal deviations in different versions.

The advantage of using HMMs as the quantization method is that they allow using musical knowledge in the process: for example, the state transition probabilities can be based on a double-nested circle of fifths, as the chord transitions are more likely to follow such musical regularities. Another positive aspect is that the HMM method does take note of the temporal structure of the chromagram data: the hidden state is dependent on the preceding state, and in music the sequential dependence is essential [28]. See Figure 2.2 for an example of how sequences produced with vector quantization and HMM differ. For both quantizations, the basis is a 24-chord lexicon; for vector quantization, the chord prototypes are used as the codebook, whereas the HMM is initialized as described above. The sequence produced by HMM is more stable, with far less oscillation between the chords. This results from the HMM favoring staying in one state, whereas vector quantization just maps the chromagram vector into the nearest codebook word.

2.3.2 Continuous Representations

Whereas quantization enables efficient and precise similarity measuring, it has the disadvantage of losing information in the quantization process. Because of this, many cover song identification approaches prefer to perform the similarity measuring directly to the chromagram data itself.

In [40], the chromagram similarity was calculated by cross-correlating complete beat-synchronous chromagrams. The distance between chroma-

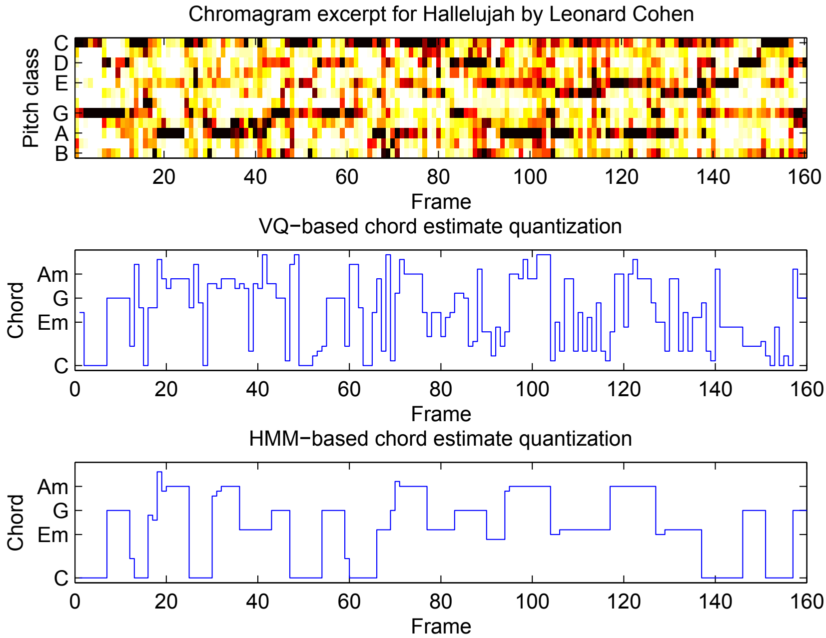

Figure 2.2: Chromagram excerpt and two quantized versions of it.

grams was measured as the peak value of the high-pass filter cross correlation, with a relatively high success rate. Later, this was improved with minor modifications of correlation normalization, tempo tracking, and temporal filtering of chromagram data [39].

In [103], a binary similarity matrix of two pieces of music was used for cover version identification. The binary similarity matrix is constructed by comparing the OTI values between two chroma frames. For chromagrams C_1 and C_2, with C_2 transposed to the most likely key of C_1, the matrix M cell (i, j) is set

$$M_{i,j} = \begin{cases} 1 & \text{if } OTI(C_1(i), C_2(j)) = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The authors report higher identification accuracies by using 36-dimensional chromagram frames and replacing the matrix values $[0, 1]$ with $[-0.9, 1]$. The similarity value is obtained by calculating a Smith-Waterman [108] alignment matrix H from the binary similarity matrix, and using the highest value of H (i.e. the best local alignment) as the similarity.

This was further extended in [104] where the self-similarity matrix used is a time series analysis tool called cross-recurrence plot. Here, the chromagram frames are first *embedded*, meaning that with embedding dimension m and time delay τ a 12-dimensional chromagram $C = c_1, c_2, \ldots, c_N$ of lenght N is turned into a sequence of state space vectors d,

$$d(i)=(c_{1,i},c_{1,i+\tau},\ldots,c_{1,i+(m-1)\tau},c_{2,i},c_{2,i+\tau},\ldots,c_{2,i+(m-1)\tau},\ldots,c_{12,i},c_{12,i+\tau},\ldots,c_{12,i+(m-1)\tau}),$$

where $i = 1, \ldots, N_d$, with $N_d = N-(m-1)\tau$. Then, for two state sequence vector sequences D_1 and D_2, the cross-recurrence plot R is constructed

$$R_{i,j} = \Theta(\epsilon_i^x - ||d_1(i) - d_2(j)||)\Theta(\epsilon_j^y - ||d_1(i) - d_2(j)||),$$

where $\Theta(v) = 0$ if $v < 0$ and $\Theta(v) = 1$ otherwise, and ϵ_i^x and ϵ_j^y are two threshold values chosen such that R has no more than κ percentage of nonzero elements for each row and column. In [104], the κ was empirically set to be 0.1.

From R, a cumulative matrix Q is calculated recursively (see [104]), and the maximum value of Q, describing the global similarity between D_1 and D_2, is chosen as the similarity value. The method presented in [104] is, to our best knowledge, the best-performing cover song identification method, based on the highest result of MIREX evaluation, obtained in the cover song identification task of 2009 [3].

2.4 Large-scale Cover Song Identification

Most cover song identification studies – including this thesis – are built on pairwise similarity measuring, with focus on extensively detecting similarities between the pieces. This naturally strives to lead to good identification results, but is hardly practical with genuinely large sets of music data due to the notable computational cost. Recently, cover song identification with so-called big data has gained a growing interest. The idea is to merge cover song identification ideas with computationally effective retrieval processes of such commercial systems as Shazam, which can search large-scale databases in a matter of seconds.

Pioneering work in large-scale cover song identification was conducted by Bertin–Mahieux and Ellis in [20], where the chromagram data was presented as fingerprints of differences in time and semitones between subsequent threshold-exceeding beat-scaled chromagram frames. The ideas were taken forward in [21], where two-dimensional Fourier transform was

[3] http://www.music-ir.org/mirex/wiki/2009:Audio_Cover_Song_Identification_Results

performed to the chromagram, resulting in representation that describes the music in a small fixed dimension space, similar to methods that are often used in digital image processing. Again, in [52], this was taken further with two-dimensional Fourier magnitude coefficients, producing a high-dimensional but sparse representation, which again was produced with dimension reduction into an even more efficient representation.

Other studies of large-scale cover song identification include [79] and [58]. In [79], the chromagram data was produced into a representation suitable for the BLAST algorithm, a near-linear sequence alignment algorithm developed originally for biosequence analysis. In [58], the retrieval process was two-phased, where the potential candidates were first filtered with a time-invariant global chord profiles hashes, before a more time-consuming but accurate retrieval was performed on the candidates with chord sequences. All in all, the large-scale algorithms make a tradeoff between the identification accuracies and computational costs.

As stated, the work presented in this thesis is not focused on fast retrieval, but emphasizes the identification accuracy. We will, nevertheless, pay some attention to the computational costs in Subsection 5.3.1.

Chapter 3

Compression-based Distance Measuring

In this chapter, compression-based distance measuring is introduced, mainly through the concept of normalized compression distance (NCD). The background of NCD in information theory is explained, and several observations on the performance of NCD are discussed. At the end of the chapter, a review of content-based music information retrieval approaches that utilize NCD or other compression-based distance measuring is presented.

3.1 Normalized Information Distance

Many similarity metrics are heavily parameter-dependent. Also, various are mostly applicable for a certain domain, utilizing a priori knowledge. In [71], a universal similarity metric based on Kolmogorov complexity was presented. Kolmogorov complexity $K(x)$ of string x is the length of the smallest binary program that produces x on a universal Turing machine. Denote the conditional Kolmogorov complexity, $K(x|y)$, as the length of the smallest binary program that produces x given y as an input. Using the conditional Kolmogorov complexity, information distance $E(x, y)$ between strings x and y is defined as [71]

$$E(x, y) = \max\{K(x|y), K(y|x)\}.$$

However, the information distance is absolute and as such does not consider the lengths of the objects, thus causing bias in the distance measuring. The authors of [71] give an example of measuring distance between the E.coli and H.influenza bacteria; the distance between H.influenza and some unrelated bacteria of the length of H.influenza would be deemed smaller simply due to the length factor. To overcome the distance bias caused by lengths of the objects a normalization factor is required. Adding the denominator $\max\{K(x), K(y)\}$ produces normalized information distance (NID), defined as

$$NID(x, y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}. \tag{3.1}$$

The normalized information distance is highly advantageous. First, it is parameter-free, requiring no background information from the domain it is applied for. NID satisfies all conditions required from a metric [71]; for objects x, y, and z, this means that all of the following requirements hold true:

1. identity: $NID(x, y) = 0$ iff $x = y$ and otherwise $NID(x, y) > 0$,
2. triangle equality: $NID(x, y) + NID(y, z) \geq NID(x, z)$, and
3. symmetry: $NID(x, y) = NID(y, x)$.

Perhaps most importantly, NID can be shown to be universal: if two objects x and y can be deemed similar according to some particular feature, they are at least as similar according to NID [71]. This would make NID usable in various distance measuring tasks. However, the non-computability of Kolmogorov complexity makes it impossible to apply the distance metric directly in practice.

3.2 Normalized Compression Distance

The normalized information distance of Equation 3.1 is non-computable, as the Kolmogorov complexity is non-computable in the Turing sense. However, the Kolmogorov complexity can be approximated using a standard lossless data compression algorithm. Kolmogorov complexity $K(x)$ can be approximated with $C(x)$, where $C(x)$ is the length of the string x when compressed using a fixed compression algorithm. The conditional Kolmogorov complexity $K(x|y)$ can be approximated as $C(x|y) = C(yx) - C(y)$, where yx is the concatenation of y and x. Thus, $\max\{K(x|y), K(y|x)\}$ can be approximated as $\max\{C(yx) - C(y), C(xy) - C(x)\}$. As $C(xy) = C(yx)$ within a compression algorithm dependent additive constant, the NID of Equation 3.1 can now be approximated as [33]

$$NCD(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}. \tag{3.2}$$

Like NID, normalized compression distance also satisfies all three requirements of a metric [33]. Similarly, it needs no background knowledge of the domain where it is used, as the compression algorithm mines the patterns from the data regardless of the domain – however, domain-specific compression algorithms such as GenCompress[1] can be applied. Normalized compression distance is not universal, though, but it can be shown to be quasi-universal [33]: it minorizes every computable similarity distance up to an error dependent on how well the compression algorithm approximates the Kolmogorov complexity.

3.2.1 Observations on Normalized Compression Distance

The domain-independence of NCD makes it applicable for various tasks. In addition, the characteristics of the metric itself have been under various studies. In [31], the robustness of NCD against noise was considered. The setup for the study was measuring the distance between a clean file x and a file with noise added y. The study proved that NCD is capable to detect the similarity even with a factor of 75 per cent of noise added. Hierarchical clustering could be adequately performed for noisy text and DNA sequences; the growth in the noise level results in worse clustering, but the quality drop is not directly proportional to the amount of noise. Also, it was observed that the noise has stronger effect when the alphabet of the strings is small. Further, in [49], the effect of different kinds of distortions (word elimination, character replacements) to the data were examined.

The differences between various compression algorithms and their peculiar features that should be taken into account in NCD-based distance measuring were analyzed in [30]. The study shows that both the dictionary-based Lempel-Ziv algorithm and the Burrows-Wheeler transformation-based block-sorting algorithm are less usable when the lengths of the compressed strings exceed the inner limitations of the compression methods. However, the algorithm based on Prediction by Partial Matching (PPM), a compression scheme that uses statistical information of the data in compressing, turned out to be robust against the file length. The most evident problem with PPM was its slow computational time.

Out of these observations, the robustness against noise is important for our work. We use noisy chromagram data, the methods used for quantizing the data produce noisy sequences, and considering the task at hand, the cover versions can be thought of as "noisy"[2] versions of the original perfor-

[1] http://www.cs.cityu.edu.hk/~cssamk/gencomp/GenCompress1.htm

[2] The term is used very loosely here; the cover versions are by no mean random versions.

mances. The file length pitfall is hardly a problem in the task of cover song identification, as the chromagram-based sequences are unlikely to exceed the length limitations of the compression algorithms, even when concatenated. Later, we will study the effect of compression algorithm selection in Subsection 5.2.1.

In [100], several variations of NCD ([32, 57, 100]) are discussed. The variations have all shown empirical success with different data and applications, and the authors prove that although all the variations of the original NCD seem diverse, they are in fact very similar, with differences mostly on the normalizing terms. Also, the practical performance of all the variations was detected to be highly similar, suggesting that the choice of the NCD variant plays a very small role.

3.3 Review of Compression-based Methods in Music Information Retrieval

Normalized compression distance and other compression-based distance measures have been widely applied for various tasks in music information retrieval. In fact, music was one of the first domains where normalized compression distance was applied [33]. In addition, NCD has been applied for distance measuring in various diverse domains such as genome data, natural language, programming languages, image data in tasks such as optimal character recognition, and many others. Although the ideas and observations made in the works of different domains are interesting and could provide insight for the work in music information retrieval, we refrain from a wider review of the state of the art, and instead refer an interested reader to the listing of NCD-based studies and applications provided in [74]. Next, we will present a review of content-based MIR methods utilizing NCD as the distance metric.

3.3.1 Symbolic Domain

With symbolic music data, it seems that NCD could be applied in a rather direct fashion; the data is already in a discrete representation (such as MIDI or musicXML), thus suitable for data compression. Straightforward distance measuring between, for example, two MIDI files is rarely practical, though, as the files themselves are likely to contain different kinds of added information (including metadata) that could easily cause bias in the distance measuring. Also, unprocessed data could be impractical in distance measuring, because, for example, raw MIDI data is not a key-independent representation.

One of the first and also most common tasks where NCD has been applied with symbolic music is genre classification [34, 33, 72, 29, 107]. Other common tasks are musical similarity measuring and retrieval [9, 12] and composer classification [34, 82]. Though not music information retrieval in the strictest sense of the word, we would also like to pay attention to the computational composition, where NCD has been used as a fitness measure for a genetic algorithm that produces melodies [10, 11, 3].

The choices for representations vary with different studies. In [34, 33], MIDI data was processed by quantizing the tracks to frames of 0.05 seconds and representing the notes in frames as semitone differences to the most frequent note of the track (a “modal” note), thus creating a key-independent representation, whereas in [72] only the highest notes of all tracks combined were preserved (also known as skyline reduction), and represented with either absolute pitch values or intervals of subsequent notes, with the latter performing better. Interval sequences and skyline reduction were also applied in [29]. In [9], we produced a binary chromagram from MIDI data by taking a time window of an estimated quarter note length, and turned the 12-dimensional data into six-dimensional tonal centroid vector representations by a method presented in [50]. The six-dimensional vectors were then labeled, thus using an alphabet of 2^6 symbols for the representation. Our work was extended in [12], where the tonal centroid transformation was excluded, and higher identification accuracy was obtained with a larger alphabet of 2^{12} symbols. Other experimented features and representations are bass melody interval histograms [107], graph-based key and tempo invariant representation [82], and differences in consecutive note lengths and pitches represented as a pair of integers [10, 11, 3].

In most studies, NCD provides rather successful results. The authors of [34, 33] report high clustering accuracies for genres and composers, but it should be noted that the studies are conducted with rather limited amounts of data, and in [34] the authors acknowledge that the results get worse as the data sets grow. Composer-based clustering was studied also in [82], with positive results. In [72], the compression-based nearest-neighbor classification method outperformed other evaluated systems (trigram-based statistical model and support vector machine). The method in [9] was evaluated with a dataset of classical variations, and it performed on a level with several state-of-the-art methods reported in [96]. The method in [12] provided even better results.

However, in some studies the compression-based distance measuring did not achieve the highest level of performance. In [107], authors report better results for k-nearest neighbor classification with Euclidean distance and Earth Mover's Distance than with NCD. In [29], the authors report that measuring distance with NCD for MIDI-based features provided subpar results, but also mention that a combination of NCD and an audio feature classifier resulted in a reasonable performance.

The studies above include several worthwhile notions. In [72], the size of the dictionary built by the Lempel-Ziv 78 compression algorithm [118] was used as an approximation of the Kolmogorov complexity, providing an interesting alternative for using the file lengths as approximations. In [12], the highest accuracies were obtained with the Lempel-Ziv based *gzip* algorithm, which deviates from various other studies where other algorithms provide better results. All in all, the studies show that the choice of the representation is crucial, but at the same time, the diversity of the works shows that there is no single choice of representation that would provide an efficient solution for all possible tasks.

3.3.2 Audio Domain

With audio data, applying NCD seems even more challenging than with symbolic music. The continuous audio signal in time-amplitude-domain is unlikely to compress efficiently (but see [48] for an interesting experiment and results on the subject). For a more practical retrieval and identification, relevant features need to be extracted from the signal and then represented in a suitable manner.

As with symbolic data, a popular task for utilizing compression-based similarity seems to be genre classification [73, 6, 48], with several studies conducted on structure analysis [17, 18] and cover song identification, the latter mostly our work [8, 4, 5, 7], but recently also with other ideas [42].

In genre classification, the work presented in [73] can be seen as a successor for the method of [72] that we discussed in the previous subsection; here, the focus was genre classification of audio data, with methodology based on a similar concept of using LZ78 dictionary size as an estimate of Kolmogorov complexity. The feature used was MFCC vectors, turned into one-dimensional symbol sequences via vector quantization; interestingly, the best results reported were obtained by using a rather large alphabet of size 1024. We used quantized MFCC vectors in [6], where the pairwise NCD was extended to lists of objects; in our work, the best results were in contrast obtained with a very small alphabet of size eight.

Compression-based measuring of structural similarity was first studied in [17], where NCD was used to cluster pieces of recorded music according to their structures. The choice of representation was uniformly quantized versions of self-similarity matrices. Later, in [18] this was extended by the recurrence plot approach of [104], with distance measuring between binary recurrence plot matrices.

Our work [8, 4, 5, 7] has been focused on the idea of using NCD for cover song identification. The motivations, approaches, features, and notions are discussed thoroughly in this thesis, with more experiments with larger sets of data and with more in-depth analysis of the results. To our best knowledge, our approach has not been proposed by other researchers until an interesting work that was published recently. Compression-based cover song identification has been proposed in [42] with focus on measuring the predictability of the time series. In this work different jazz standard renditions were detected based on chromagram data discretized with vector quantization using various codebook sizes. The paper also provides an interesting version of NCD where the concatenation as the estimation of $K(x|y)$ is replaced with an aligned version of the concatenation.

The results for the studies have mostly been in favor of using NCD, but there are several remarks. The notions in [17] suggest that the structural differences could be a pitfall for NCD-based distance measuring. This parallels the observations for symbolic music in [34], that classical movements of different symphonies were deemed more similar than different movements from the same symphonies due to the similarity in their structures. In [42] the authors introduce an entropy-based continuous distance measuring based on the normalized information distance, and even though the results are promising, the authors also note that the unquantized similarity measuring performed with a better accuracy. The genre classification experiments in [6] and especially in [73] provided satisfactory results, but both were conducted on the so-called GTZAN dataset [112]. Although this dataset has become a *de facto* benchmark used for evaluating genre classification methods, it has recently been a subject of criticism for various shortcomings (e.g. [109, 110]).

Chapter 4

Composition Recognition and Invariances

In this chapter, we take a closer look at alterations that are present in cover versions and explore what makes cover song identification such a difficult task. As a result, this chapter should provide insight on how to build a compression-based cover song identification system that is capable of achieving invariance to modifications and detect the essential compositional characteristics of the piece.

4.1 Basis

Work on cover song identification often seems to focus on developing a novel technique for the task and then providing in hindsight an analysis on how the method performs and what the strengths and weaknesses of the method are. Here, we take an alternative approach. We start out by observing what kind of variations and alterations are present in the cover versions.

As a starting point, we will use the list of musical invariances described in [70]. Although the purpose of [70] is to provide formal, set-theory based definitions for the musical invariances, the listing of the common invariances as presented in the paper is useful for our purposes also, as the differences in cover versions mostly fall into the categories of invariances described in the paper. In addition, [102] provides a list of possible changes in cover songs. Out of these, we are not interested in lyrics, as changes in language are unlikely detectable in chromagram data.

4.2 Musical Examples

In order to detect invariances, we experiment with two pieces of music and their cover versions. We chose two often covered pieces of popular music, and for both, we have 40 different cover versions, ranging from very similar to highly different and nearly obscure versions.

Yesterday

Yesterday is a popular song, originally performed and recorded by The Beatles, credited to John Lennon and Paul McCartney, and published in 1965. Yesterday is often noted as one of the most covered pieces in popular music; a total of 2200 recorded cover versions are known to exist[1]. Our dataset consists of versions from different genres and eras of popular music. For a detailed content of the Yesterday dataset, see Appendix A.

Summertime

Summertime is a jazz standard from 1935, composed by George Gershwin and originally included in the opera Porgy and Bess. Even more covered than Yesterday, there are over 25,000 different recordings of Summertime[2]. See Appendix B for information on the Summertime dataset. It should be noted that our dataset does not include the first recording by Abbie Mitchell. As the original canonical version is missing from our dataset, we will use the version by Billie Holiday as the canonical version. It was published in 1936, very soon after the version by Abbie Mitchell, and was the first version to appear in commercial charts.

4.2.1 Essential Musical Characteristics

Before taking a closer look at the alterations of the cover versions, we will first study the original performances and make observations on what are the most important and essential musical characteristics these pieces hold, and see how well such features are contained in the chromagram data and the quantized representations.

For both pieces, the lead melodies, usually performed by the vocalist or some soloist instrument, seem to be the most distinguishing feature and the most identifiable character for a human listener. Listening to our test material also proves that these salient melodies are present in all versions

[1] http://en.wikipedia.org/wiki/Yesterday

[2] http://en.wikipedia.org/wiki/Summertime_(song)

for both pieces, occasionally heavily varied but still easily distinguishable for a human listener; even with notable variations, there always seems to be enough cue to detect at least significant parts of the original melody.

Both pieces also have distinctive harmonic progressions. For visualizations of Yesterday chord sequence approximations and salient melodies that were extracted with a method presented in [63], and their ground truth comparisons, see Figures 4.1 and 4.2. The ground truth sequences are based on [2], transposed to the key of the original performance (F major). Note that the chord estimations, extracted with the method of [19], utilize a lexicon of only major and minor triad chords; in truth, the chords are more complex, and the estimated sequence cannot be considered as an adequate chord transcription. The ground truth sequence is similarly mapped to a 24-chord lexicon, with for example seventh chords mapped to the corresponding triad chords (e.g. Am7 is mapped to Am). With melodies, the ground truth melodies are transposed to the same octave as the estimated melodies; the actual vocal melody is in reality two octaves higher.

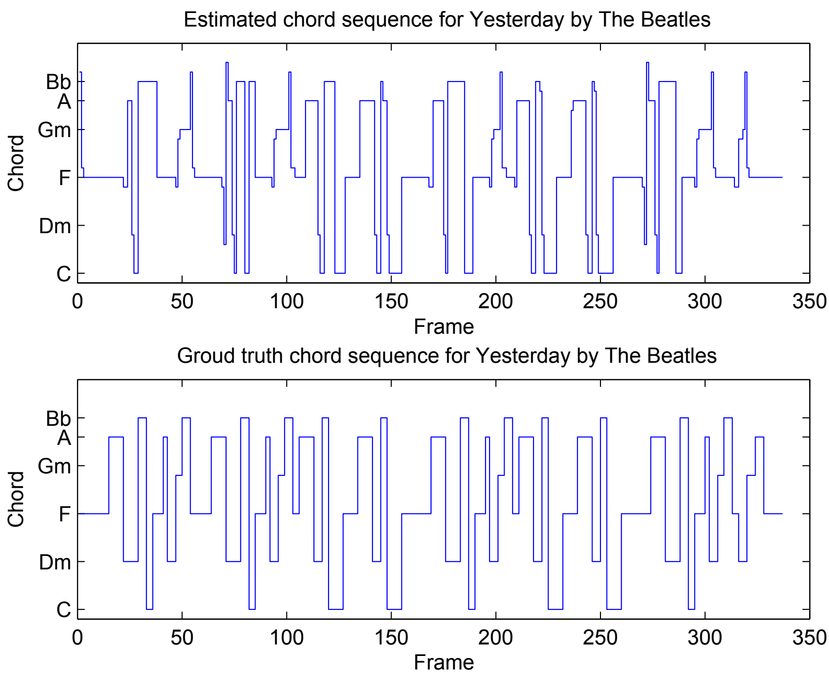

Figure 4.1: Comparison between estimated and ground truth chord sequences.

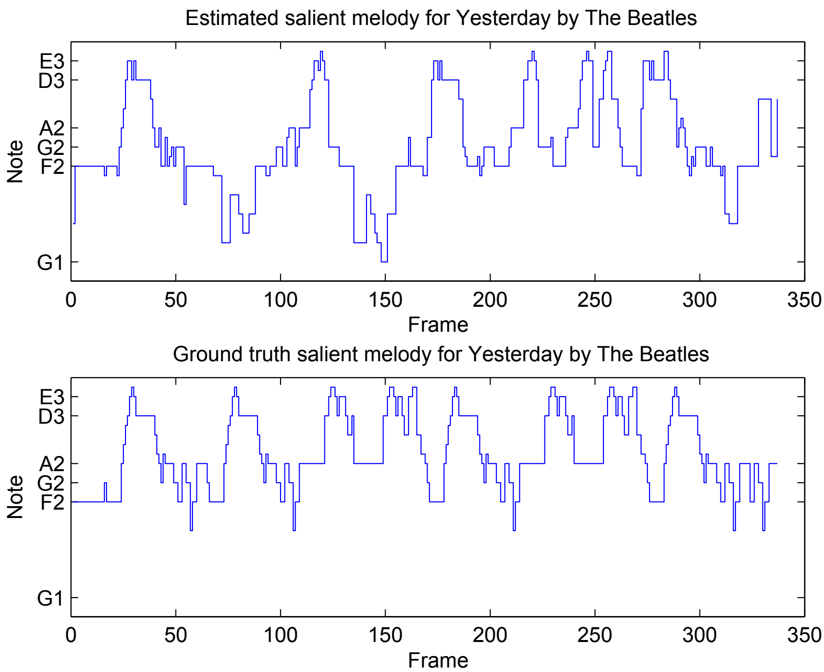

Figure 4.2: Comparison between estimated and ground truth salient melodies.

Similar chord and melody approximations with comparison to the ground truth are presented for Summertime in Figures 4.3 and 4.4, for chords and melody respectively. The ground truth melody is again transposed to the same key and octave as the estimated melody. In the melody estimation the third verse – an instrumental passage – is clearly visible as the melody leaps temporarily one octave higher. The ground truth is obtained from [1].

4.3 Global Invariances

First we take a look at invariances that are global in a piece of music, that is, invariances that hold true for most or all of the piece. The division into local and global invariances is not strict; some global invariances might appear only locally (for example, key modulations), and vice versa.

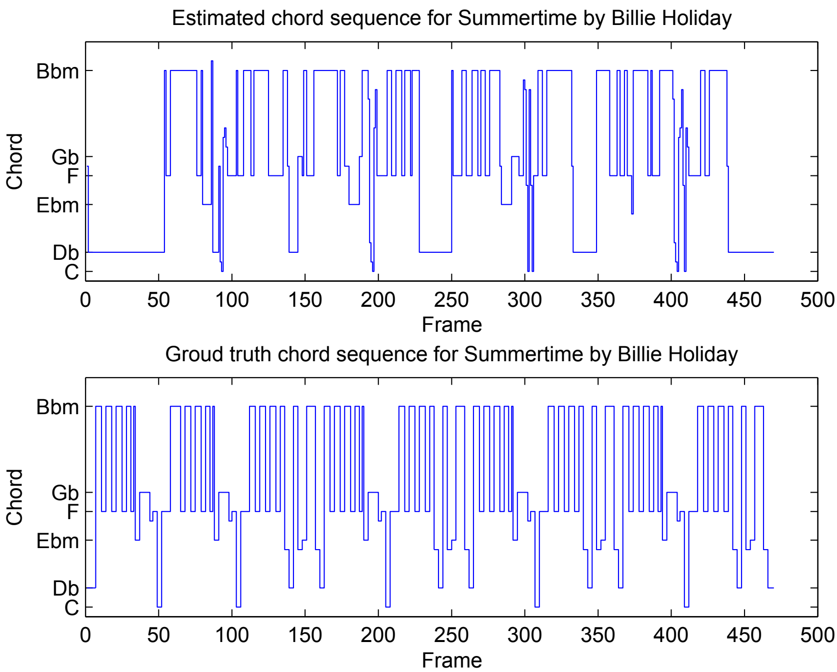

Figure 4.3: Comparison between estimated and ground truth chord sequences.

4.3.1 Tempo Invariance

The tempi of both original pieces are moderately slow, for Yesterday approximately 98 beats per minute (BPM) and for Summertime approximately 103 BPM. For a piece of music in a tempo of 100 quarter note BPM, a single chroma window of 16384 samples on audio of 44100 sample rate represents approximately a time of a bit over an eighth note (also known as a quaver).

For the two examples used here, the tempo variations in cover versions were mostly moderate. For both Yesterday and Summertime, most cover versions in our dataset are somewhat faster than the canonical version. Using the MIRToolBox implementation of a tempo estimation algorithm described in [65], we calculated the tempos for all pieces. The results show that the variations in tempi are quite modest in comparison to the original tempos, with standard deviations being 30.702 for Yesterday cover versions and 33.397 for Summertime covers. We noticed that several tempo estima-

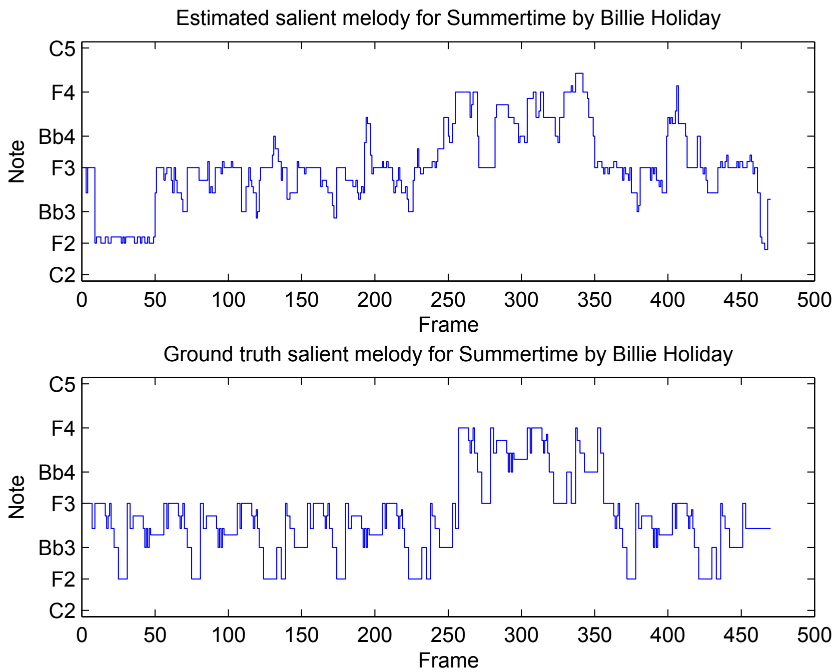

Figure 4.4: Comparison between estimated and ground truth salient melodies.

tions were clearly incorrect; in some cases, the algorithm gave the piece a tempo of circa 180 BPM, which is likely double the correct value. Thus, some distinct bias in the tempo values exists, making the actual tempo deviations even smaller.

Similarity in tempi does not automatically make two versions of the same composition easily distinguishable. A more interesting question is how much tempo changes confuse detection of otherwise similar pieces of music. Here, we performed a small experiment on the identity cases of the canonical versions. Using Audacity[3], version 2.0.0, we constructed alternative versions of the canonical versions by changing the tempo (without altering the pitch) by $-24, -18, -12, -6, 6, 12, 18,$ and 24 beats per minute, and then calculated the identity distance values between the original version and the tempo variations. We did this for both Yesterday and Summertime, and the changes in NCD values are depicted in Figure 4.5.

[3] http://audacity.sourceforge.net/

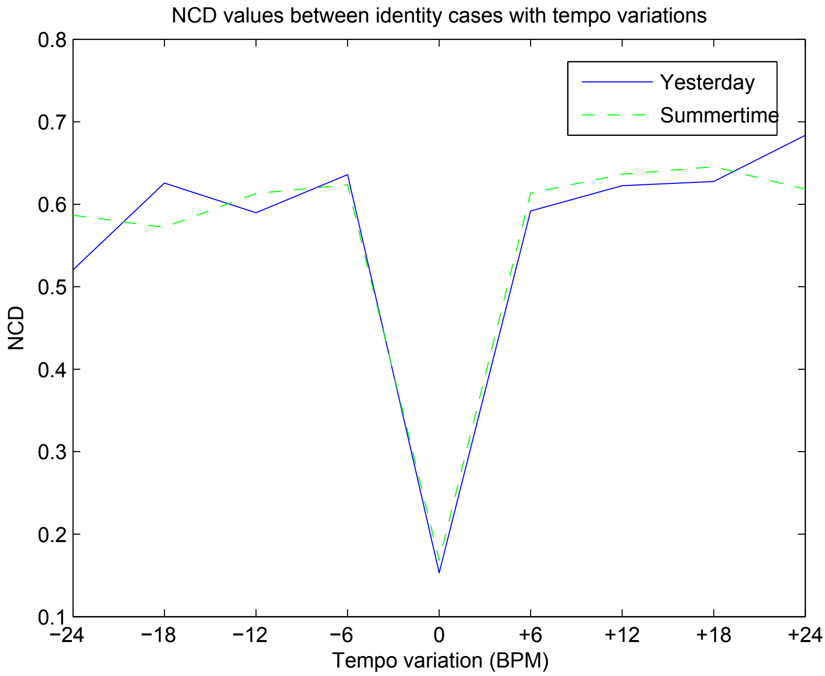

Figure 4.5: The effect of tempo changes in the NCD values.

The results show that the values do increase immediately when the tempo changes. Also, the change in NCD value does not follow linearly the tempo changes. On the other hand, the changes in NCD values are quite small altogether with different tempo variations, suggesting some tempo invariance in the distance measuring. Based on this, it seems that the question of tempo invariance does have importance in cover song identification and we will return to discussing tempo invariance in Section 5.2.3. It should be noted that the changes are similar with both Yesterday and Summertime; as we will notice, the Summertime dataset can be considered as a more difficult set than Yesterday, but consequently, this is not due to the changes in tempi.

4.3.2 Key Invariance

It is not unusual that cover versions of a piece are transposed into a different musical key. There are several reasons for this, namely finding a more suitable key for the vocalist. Detection of key has been studied extensively

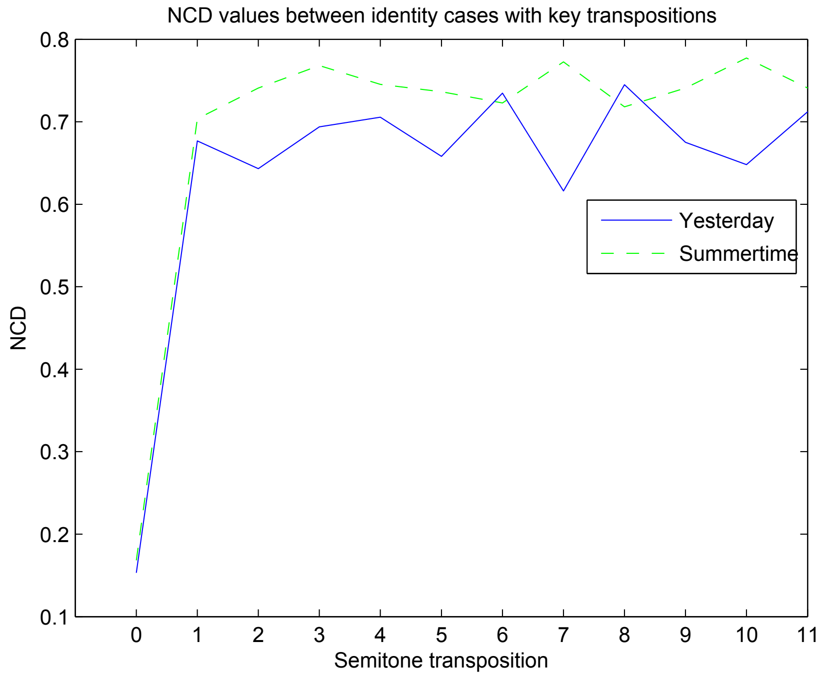

Figure 4.6: The effect of key transpositions in the NCD values.

in MIR literature (see Section 2.2.2) and several methodologies for obtaining key invariance when comparing pieces of music exist, although there still are no completely trustworthy solutions.

Despite the methodology, the key invariance is roughly obtained by either transposing the other chromagram or representation, calculating all possible transpositions (or a subsection of them). To review the effect of key invariance, we took a closer look at how the transposition of a piece of music affects the NCD values. We performed an experiment similar to the one in the previous subsection; we took the canonical versions, and produced all 12 key variations for the both of them, and then calculated the identity case values between the original performance and all the transpositions. The NCD values produced by this experiment are depicted in Figure 4.6.

The results confirm the presumption that key invariance is a highly important factor in cover song identification and should not be ignored when measuring distance in chromagram data. As with tempo invariance, we will return to the question of key invariance in retrieval in Section 5.2.3, where a more real-world experiment on cover song identification is performed. This experiment also shows that with a more complex chroma information of Summertime, the NCD values grow even higher with the wrong key.

Key Modulations

As the global key invariance can be achieved to a certain tolerance, we take a closer look at local key transpositions known as modulations. In pop music, modulations typically occur at the end of the piece, transforming the final section a semitone higher than the beginning of the piece. If a section of a piece is transposed, distance measuring based on global alignment is clearly affected, as the transposed section does not match the original performance.

If the chromagram representation is key-invariant (e.g. the relative changes), modulations will not cause major harm for the identification. However, key-invariant representations have other disadvantages (we will discuss this in Subsection 5.2.3), and thus they are not likely a solution for the problem. Instead, the distance measuring should be robust against key modulations.

Systems that measure the distance between the pieces using local optima, such as longest common subsequence between the pieces, can be considered to be somewhat robust against key modulations, as the longest common subsequence might appear before the modulation takes place. Normalized compression distance, on the other hand, measures a global distance between the pieces. With one piece of music including modulation and the other not, the data compression algorithm could not benefit from the modulated information when compressing the modulation-free version.

To observe this, we looked for some of the versions that include key modulations. For Yesterday, we took two versions with modulations under closer study. The version ID 25 can be deemed to be an easily distinguishable version, but it does include a semitone modulation in the last section of the piece. Version ID 9 is a slightly more difficult version to distinguish, and it also includes a transposition (one and a half semitones) at the end of the piece. Here, we constructed versions with no modulations straightforwardly by manually modifying the estimated chord sequences, and lowered the pitch of the modulated part of piece back to the key of the beginning of the performance. This gave us interestingly conflicting results; with the more difficult case of ID 9, the NCD value dropped from 0.7249 to 0.6648 when the modulation was removed. However, with the easier case ID 25, the distance actually *rose* from 0.6377 to 0.6569. The increase is quite small, but still it is an interesting notion on the behavior of NCD; even if the sequences should now be more similar, the measured distance is actually higher. The change in distance with the case of ID 9 suggests that the modulation might indeed affect the similarity measuring; however, the distance value with the modulation is still relatively low, so possible key modulation should not be a determining factor in the outcome of the identification process.

4.3.3 Tuning Invariance

A large proportion of recorded music is performed in the so-called pitch standard of 440 Hz. This refers to the frequency of the A4 note. This note is known as Concert A, and it is used as a reference to which instruments are tuned. The frequency value of Concert A has varied throughout times for various reasons. The 440 Hz value was standardized in 1955, and most of the recordings produced after this use it as the reference frequency. However, occasional differences exist. In addition to using a different Concert A frequency altogether, the recorded music might be processed, for example by manipulating an analog recording to a slightly faster tempo, which also affects the pitch of the recording. The differences in tuning could have an impact on the identification.

To obtain tuning invariance, the chromagrams can be tuned. To obtain a tuned 12-bin chromagram, the chromagram is first calculated using 36 frequency bins (i.e. each bin refers to a third of a semitone, or microtone). For each frame, the peak bins are calculated; a peak meaning the bin having a value higher than the values of the adjacent bins. Then, quadratic interpolation is applied in order to obtain peak positions and values. After locating the peaks, the chromagram is shifted (if necessary) so that the peak values now match the semitone center bins. Finally, the 36-bin chromagram is reduced to 12 bins by summing the values within a semitone. As with 12-bin chromagrams, the chromagram frame values are normalized. We take a shortcut, and instead of interpolation just calculate the peaks in the bins and select the center bins according to the peak histogram.

The effect of the tuning algorithm is illustrated in Figures 4.7 and 4.8. In Figure 4.7, a 36-bin chromagram is presented. Then, in Figure 4.8, the untuned 12-bin chromagram for the same audio excerpt is depicted, followed by the 12-bin tuned chromagram obtained from the 36-bin version. In this case, the piece of music is quite near to the Concert A pitch, but the 36-bin version still reveals that the pitch is not quite accurate; clearly, some of the spectral energy of pitch class F is spread between two microtone bins. The slight mistuning causes the values of tuned and untuned 12-bin chromagrams to be somewhat different, as visible in Figure 4.8.

From our point of view, however, this is mainly interesting in case it affects the quantized versions of the chromagram. In Figure 4.9, the chord

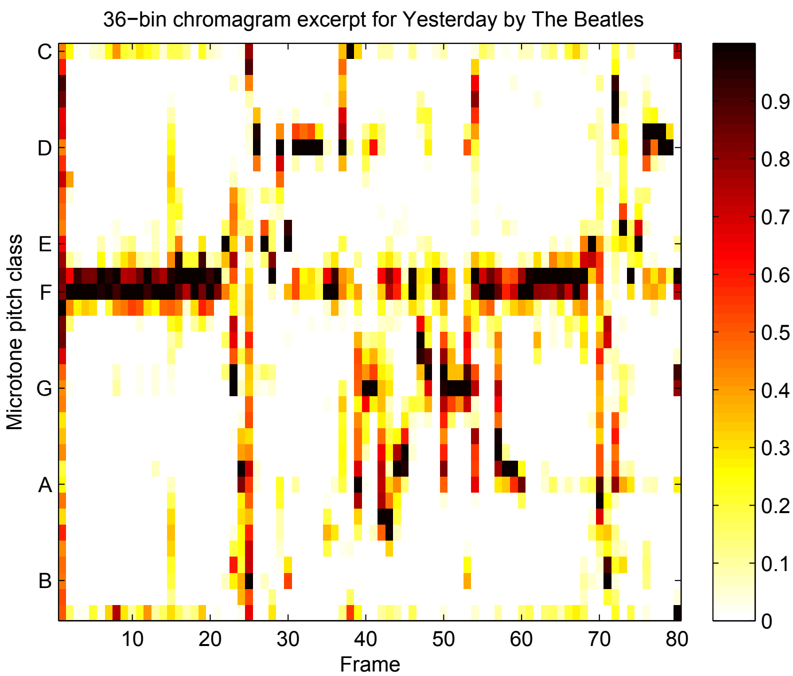

Figure 4.7: 36-bin chromagram.

sequence estimations for the untuned and tuned chromagram fragments of Figure 4.8 are presented. There seems to be a notable difference between the sequences, with the tuned version producing, as could be presumed, a cleaner version. This suggests that chromagram tuning should be applied in the process, not only to apply tuning invariance but also because the resulting sequences seem to approach correct transcriptions. However, the produced version is not only cleaner, but there are also major differences. Whereas the F major chord is notably common in both sequences, they both also contain estimated chords not present in the other sequence. Such differences will likely cause variation in the identification process. In Subsection 5.2.3, we will experiment with the tuning in order to empirically determine whether it actually is useful.

4.3.4 Structural Invariance

The original performance of Yesterday consists of two different sections (known as verse and chorus), both approximately eight bars long. Labeling

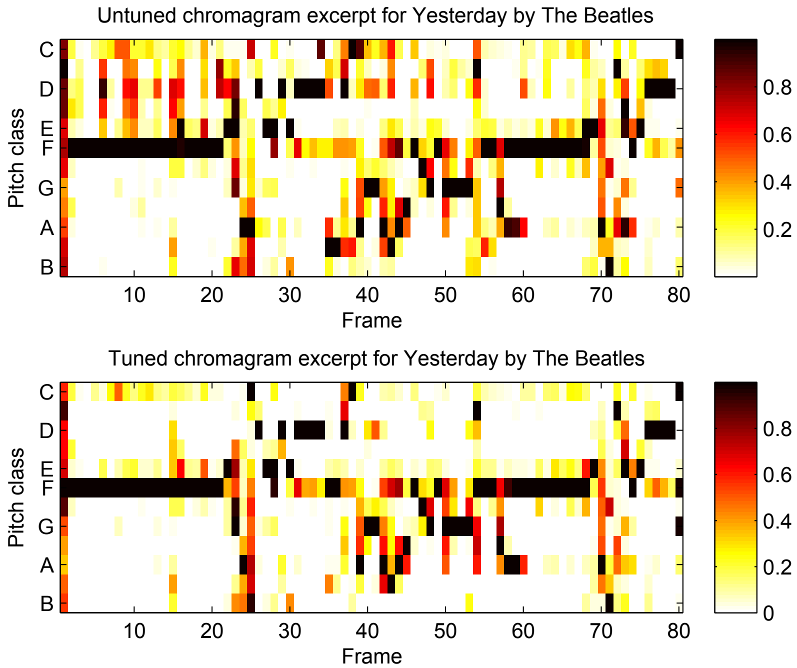

Figure 4.8: Comparison of chromagrams. The upper one is a version extracted directly from audio, whereas the lower is produced from a 36-bin chromagram.

the sections as A and B, the structure of Yesterday can be displayed as AABABA. Most cover versions of Yesterday follow this structure to an extent, occasionally adding an instrumental section somewhere, usually at the beginning, before the third A-section, or at the end. The versions that we consider to be the most dissimilar usually include a lengthy instrumental section at the end. Some performances include short (a few beats or at most a few bars) introductions or transitions between the A and B sections that are not present in the original composition, but all in all the structural variations with Yesterday cover versions can be considered moderate.

Summertime consists of a single repeating sequence of 16 bars. The version by Billie Holiday consists of this sequence repeating four and a half times; three times sung, one as an instrumental passage and an eight-bar length portion of instrumental introduction. Labeling these sections as A, B, and C, respectively, the structure of Summertime is CAABA. The cover versions of Summertime take much more artistic liberties with the

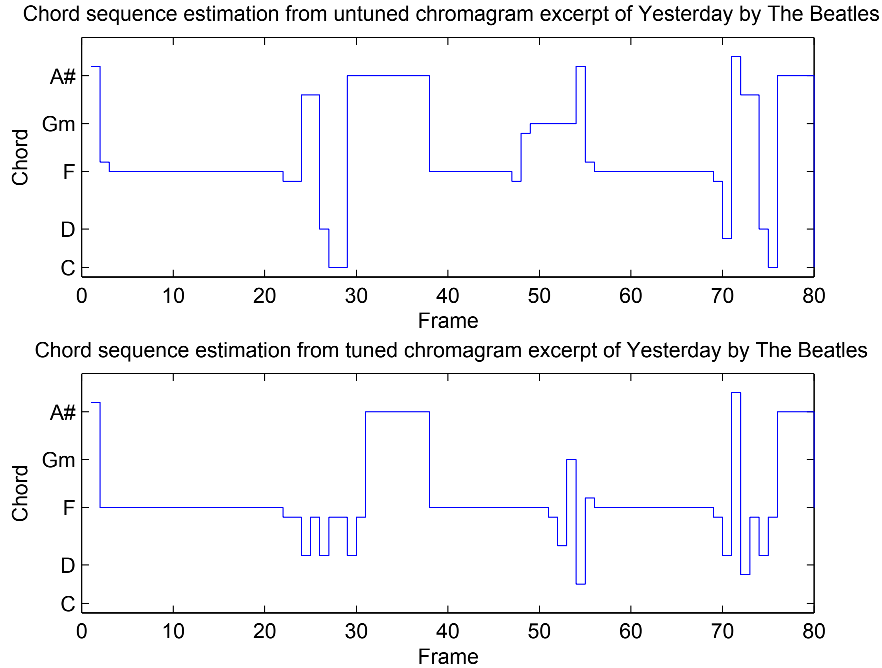

Figure 4.9: Comparison of chord sequences estimated from untuned and tuned chromagrams.

structure, but the 16-bar length sequence seems to be constant in more or less all versions; however, some performances do include short (i.e. a few bars of length) transitions or fills between the sections[4]. The versions of Summertime often contain extended soloistic instrumental passages, but in most cases they still seem to preserve the harmonic progressions underneath the solos. The actual chords themselves, though, seem occasionally to be greatly varied, making chord-estimation-based similarity measuring difficult.

Cover song identification applications do usually not pay additional attention to obtain invariance for the structural differences, such as attempt to label automatically the sections and then compare similarities between the detected sections. Most research is based on using a similarity metric that should be robust against such changes, and for our work, this is also one

[4]Actually, according to [1], the canonical version of Summertime is composed over an 18-bar length sequence, i.e. the Billie Holiday version is not a completely faithful rendition of the composition, whereas some of the other versions are.

of the motivations for using NCD; if the musical content of the additional sections in a cover version is similar the compression algorithm should be able to detect this and ignore the repetition. Considering our research, it should be noted that Bello states in [17] that one of the hindrances of NCD is the bias caused by structural differences. On the other hand, we noticed in [4] that structural differences had very little effect in overall identification; these results, however, were conducted with a small amount of data with only two versions per piece.

To evaluate the effect of structural variations, we took a closer look at the structural similarity of our test data with relation to the compressibility and NCD values. Not surprisingly, several of the Yesterday versions that have the smallest distance to the original recording are highly similar to the original performance, although out of the five nearest cover version, only two had a structure completely identical to the original version. For example, the version with the second smallest distance, ID 10, is not an identical version structurally; using the same notation as above, the structure of this version could be described as AABAA – that is, the version jettisons the second chorus section altogether. This has little effect on the performance of the NCD-based similarity measuring, suggesting that the structural differences are unimportant if the tonal content of the sections does not vary considerably. On the other hand, the versions with largest distance to the original include not only structures that have relatively little to do with the original version, but even with the corresponding sections, there is a significant amount of variations in melodies and arrangements, making the chroma profiles greatly different. This is the case with version ID 39, which has a closely similar structure to the original, but apart from that, there are very few similar elements between the pieces. Similar observations can be made with the Summertime versions.

In short, we feel free to state that the structural diversity is not a remarkable challenge for identification with compression-based similarity measuring, and the more important aspect is to build a representation that is robust against the changes in the chromagram information. No additional processing (i.e. structural analysis) is needed in the identification process.

4.4 Local Invariances

The small variations that occur in relatively small portions of the pieces, and may vary in both time and pitch, are often highly important in giving the cover version its own identity; again, for an example, small variations in salient vocal melody might result from a significant vocal style of the cover version performer.

4.4.1 Melodic Invariance

The lead melody is most likely the feature that will distinguish a human listener if a piece of music is a cover version. It is also subject to a great amount of variation; although the melodies of the cover versions are detectable for a human listener, they divert in both pitches and onset times. This makes detecting the melodies from chromagram data rather challenging; considering the length of the analysis window, each frame of the chromagram represents a very short time in music, and thus, the variations in melody occur over a number of frames altogether.

To actually identify the similarity despite changes in the lead melodies and other variations in pitch classes, the similarity measuring should be based on detecting longer melodic patterns, while allowing time-warping and ignoring occasionally pitch transformations that might occur in the versions. This issue has not been extensively studied in cover song identification, and most systems seem to rely on detecting the pairwise similarity by means of dynamic programming; for example, the dynamic time warping methodology allows aligning sequences that vary in time. However, the successful method of [104] utilizes a time series analysis technique called embedding; with embedding, the similarity measuring is based on matching longer pieces of chroma information. and thus allows to detect similarities in sequences of notes.

The small variations in the pitch of the melodies are ignored in cover song identification, and the similarity measuring is based on detecting the overall similarity between chromagram frames. This means that the similarity of the accompaniments between the pieces has a significant role in cover song identification, and again makes pieces of music with larger harmonic content and diverse arrangements far more difficult to identify.

For our work, the question of quantizing the chromagram data is highly relevant here. The chord estimation, as stated, is a rather crude representation, labeling chroma vectors with an alphabet of only 24 characters, a very small number considering the rich nature of tonal music. Still, there are motivating advantages; hypothetically, if the melody varies inside a bar of music accompanied by instruments playing a C major chord, the HMM-based estimation is likely to consider all frames of the bar to represent the C major chord, thus ignoring the small melodic variations. But this representation also has its downside. The chord estimation is prone to mislabelings, and the changes in chroma profiles due to the changes in the arrangements can possibly lead to incorrect chord estimations. Another downside is the crudeness of chord estimation, as unrelated pieces of music might still contain harmonic similarities, resulting into similar chord estimations. Because of this it would seem to be a good idea to use a larger or more complex alphabet for the representations. But there is a trade-off here; even if the representations should be capable of describing chroma frames in rich detail, they should also be able to maintain compressibility and the robustness on small variations. This problem will be considered in the following sections of this thesis.

4.4.2 Arrangement Invariance

The arrangement seems to be one of the most important features that changes in cover versions. With popular music, one can assume that a reason for recording a cover version of a piece of music is to produce an interpretation that highlights the distinct characteristics of the performers; thus, recorded cover versions rarely are note-to-note renditions of the original version.

The highly different arrangements make the task rather challenging, as the changes in spectral information eventually turn into chromagrams that can be greatly different from the chroma data of the original piece. In such cases, the most distinguishing feature – the lead vocal melody, for example – is "hidden" in the chromagram information, among the pitches played with accompanying instruments. In cases like this, the resulting chroma frames might be quite dense.

See Figure 4.10 for an illustrations with three different versions of Yesterday; each picture depicts approximately the first half of the first chorus section of the piece, all in the same key. Traces of the lead melodies are present in each chromagram example, but the overall chroma profiles have remarkable differences. The similarity between the first two chromagrams can be, to a certain extent, represented with the HMM-based quantization, whereas this is efficiently lost with the third version which bears little resemblance to the first two. For an example of this, see Figure 4.11.

In contrast to the complex arrangements, some of the versions included in the dataset have a remarkably light instrumentation, retaining only the lead melody and some accompaniment. Naturally, the chromagram data extracted from these versions is clean and almost sparse, making feature extraction notably more straightforward. We already mentioned receiving a relatively small distance value for a lightly arranged piano rendition of Yesterday; here, though, it needs to be mentioned that the original version of Yesterday is also rather sparsely arranged. This raises the question of whether the chromagram data should be processed into a more trivial version, by losing information considered as unimportant. This could be done by methods of dimensional reduction. However, occasionally the more

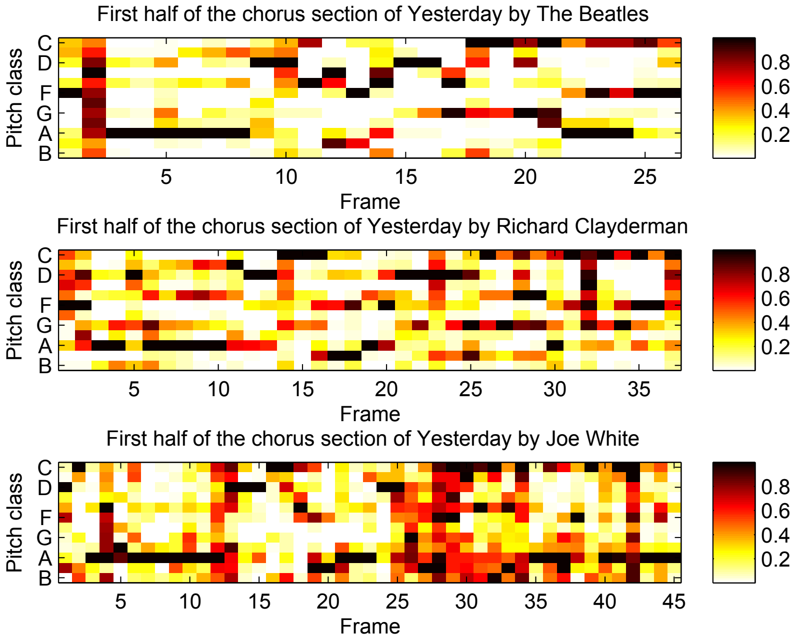

Figure 4.10: Chroma profiles for first halves of chorus sections from three different renditions of Yesterday. Notice the difference in lengths caused by tempo differences.

sparse arrangement might also be problematic; the most difficult version of Yesterday, ID 27, is indeed also a lightly arranged version with a relatively small number of instruments playing throughout the piece. But as the musical content of the version is greatly different, the sparse chroma information is equally unuseful. And vice versa, some of the easier versions of Yesterday include vast arrangements, but our methodology still discovers similarities between the versions. Similar observations could be made with the Summertime dataset; one of the versions deemed as most similar (ID 37) is an ascetic version featuring a singer with a single acoustic guitar accompaniment, but at the same time one of the most difficult versions (ID 3) is performed by a small group of musicians; it just contains far more musical information.

The remarks here suggest that a cover song identification system does not need to remove the external information, or “noise”, from the chromagram data, since such information might not even be present in the more

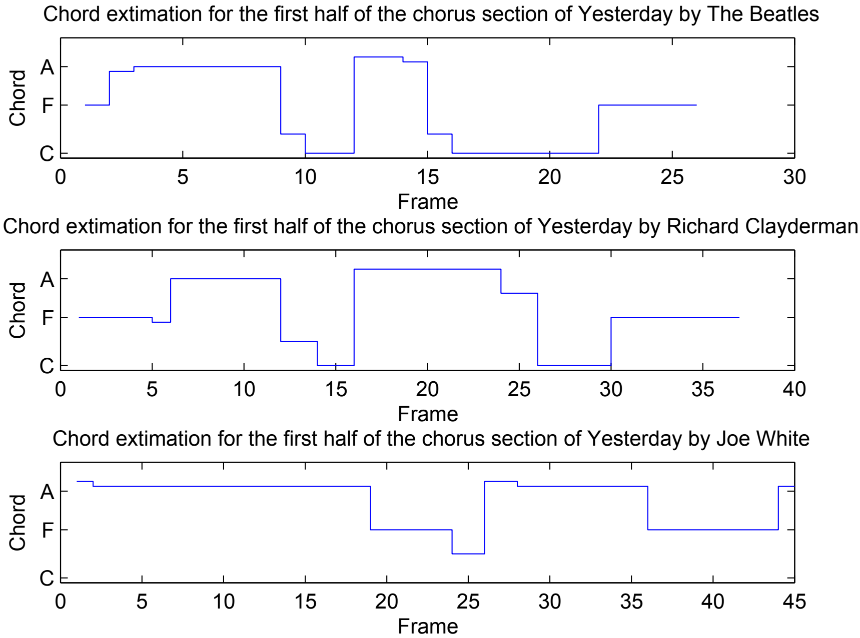

Figure 4.11: Chord sequence estimations for first halves of chorus sections from three different renditions of Yesterday. Notice the difference in lengths caused by tempo differences.

difficult versions. Again, the question lies more in the representation and similarity measuring.

4.5 Similarity Values

We calculated the NCD values between the original performances and their cover versions. We used chord sequences as representations, *bzip2* algorithm for compression, and OTI for key invariance. In order to observe the amount of confusion, we also calculated the distances between the original performances and the cover versions of the other piece of music. See Figure 4.12 for a visualization of sorted distances between the original Yesterday and all Yesterday and Summertime variations, and Figure 4.13 for similar visualization with the original Summertime performance. The visualizations reveal that with Yesterday, the performance is already decent; there are smaller distance values between the correct pairs than with the incorrect ones. The distance values with Yesterday are smaller than with Summertime; with Summertime, the highest distance values for correct pairs are nearly 0.9, meaning that the compression algorithm has not discovered many similarities between the two sequences.

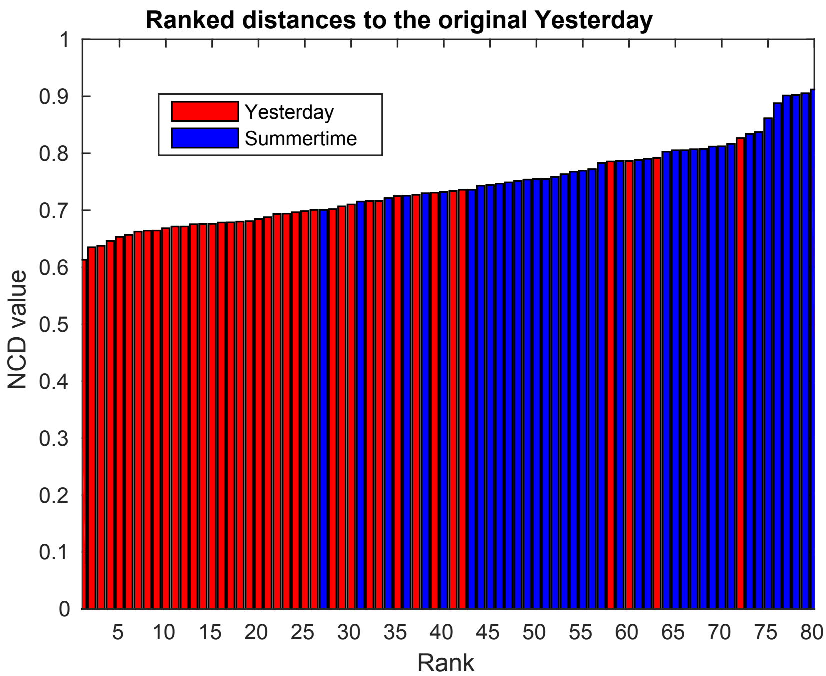

Figure 4.12: Normalized compression distances between the original version of Yesterday and all cover versions of both Yesterday and Summertime.

4.5.1 Classification Experiment

Next, we wish to consider the possibility of confusion between these two datasets. On this account, we performed a relatively straightforward classification task. Using the two canonical versions as the training data, we calculated for each of the total 80 cover version's distances to both canonical versions, and then classified them according to the nearest canonical version. Clearly, this test is far too trivial to be considered as an actual classification experiment, but it provides insight in how well the representations and compression-based similarity measuring performs in a case where

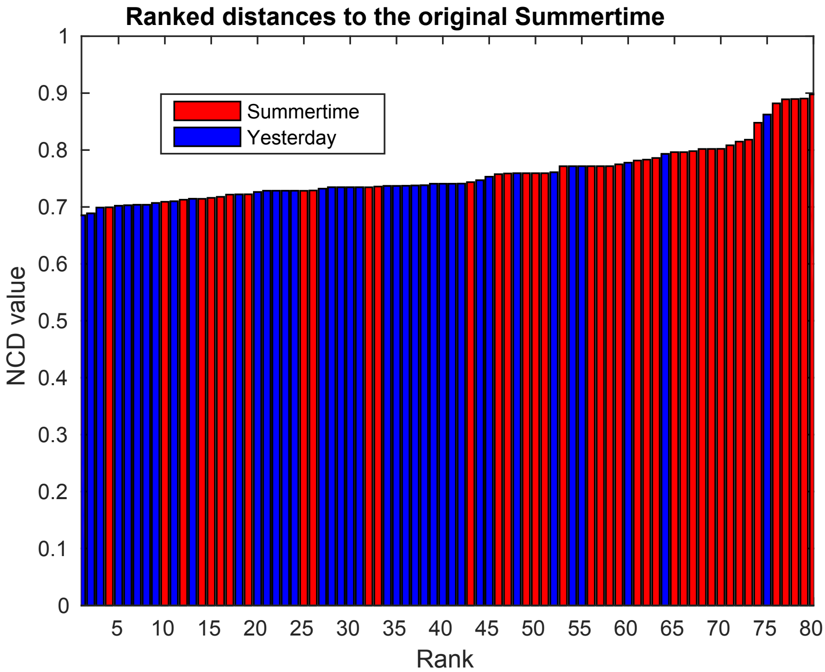

Figure 4.13: Normalized compression distances between the original version of Summertime and all cover versions of both Yesterday and Summertime.

the identification is a rather straightforward task of binary classification, suggesting that if the performance here has undeniable issues, there is very little chance that the actual identification task with vast amounts of music could be successful. The results of this experiment are presented in Table 4.1.

The results show that the classification of Yesterday is nearly perfect, with only two of the most dissimilar pieces deemed to be versions of Summertime, resulting thus in an accuracy of 0.925. On the other hand, with Summertime we identify far less versions correctly, achieving a rather limited accuracy of 0.600.

One could suggest that the poor accuracy with Summertime is due to the selection of the canonical versions; we would like to remind the reader that the version we refer to as canonical is not the very first recording of Summertime. We took a closer look at the pairwise distances between each pair of the Summertime dataset. The obtained distance matrix is presented in Figure 4.14. Here, we notice that some of the versions seem

		Predicted	
		Yesterday	Summertime
Actual	Yesterday	37	3
	Summertime	16	24

Table 4.1: Results for confusion experiment between the two datasets, with distances measured between HMM-based chord estimations.

to have overall smaller distances than others. By taking a mean value of the distances, version 41 is the one that can be considered to be the one that has most shared information between all versions of the data set. This version is by no means one of the earliest performances of Summertime, but instead a recording published in 2000. When working with NCD, it should be remembered that this means that if the representation from version 41 compresses most efficiently with other representations, it means that the quasi-universal similarity the compression algorithm detects is present the most in this version. By observing this version we notice that it shares similarities with both the "traditional" versions (the salient melody is highly similar to the original version) and the more "modern" versions (complex arrangement and structure with extended instrumental sections). We ran the experiment again, but this time using the version with ID 41 as the training data for Summertime; although this improved the identification of Summertime to an accuracy of 0.875, this had an adverse effect on identification accuracy, with accuracies being now only 0.700 for Yesterday.

These results, as stated, are based on the initial similarity values. However, we will here take a small sneak peek at the upcoming discoveries, and in the following, display results for these datasets using methodologies based on the remarks made in Chapters 5 and 6, using different steps of preprocessing and feature combination that will be discussed in the mentioned chapters. As previously, we depict the histograms for the sorted NCD for both datasets, with the histogram for Yesterday presented in Figure 4.15 and for Summertime in Figure 4.16. A similar classification experiment was performed with these distances, with the results depicted in Table 4.2. The results show an improvement with the classification of Summertime now achieving an accuracy of 0.725. On the other hand, the accuracy with Yesterday came down to 0.900.

Based on these notions, we can fairly denote Yesterday as an "easy" dataset altogether, and similarly denote Summertime as a difficult dataset. Next, we try to provide some insight into why this is the case; in other words, what actually makes similarity detection with Summertime versions

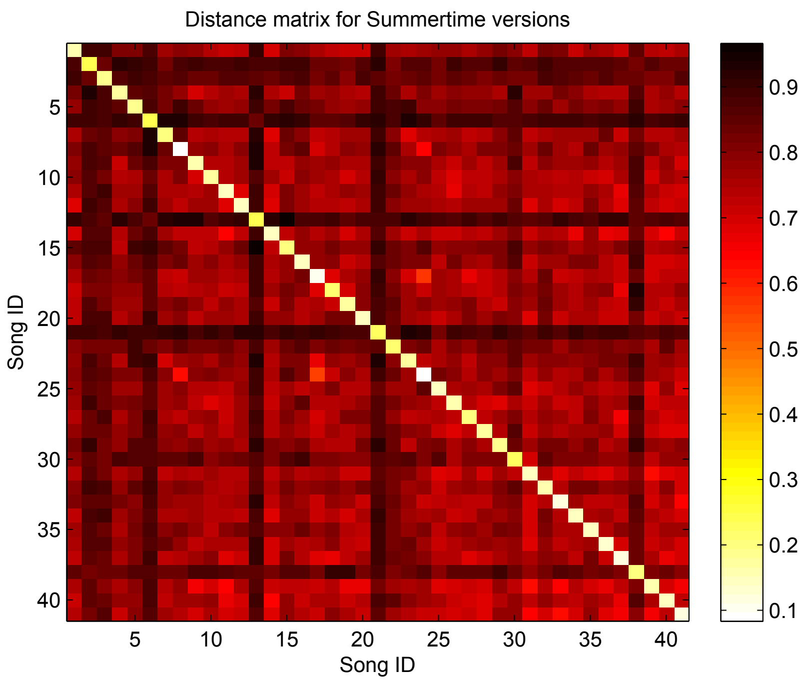

Figure 4.14: Distance matrix between the Summertime versions.

such a more difficult process, and is there anything that could be done better.

4.5.2 Difficult Cases

As stated, cases that are practically impossible to distinguish are likely to exist. With Yesterday, the few versions confused with Summertime can be considered difficult; the highly varied melodies, along with completely different arrangement, floating tempo, far more complex structure, and other modifications make the pieces a cover version of Yesterday only in name.

With Summertime, there are versions which are difficult to distinguish even by human listening; these versions include just a small fragment of melodies similar to the original piece, and the rest of the piece comprised soloistic performances, sharing only underlying similarities with the original performance. Here, the global similarity measuring of NCD seems to be problematic; the similarity between the fragments of the pieces might be

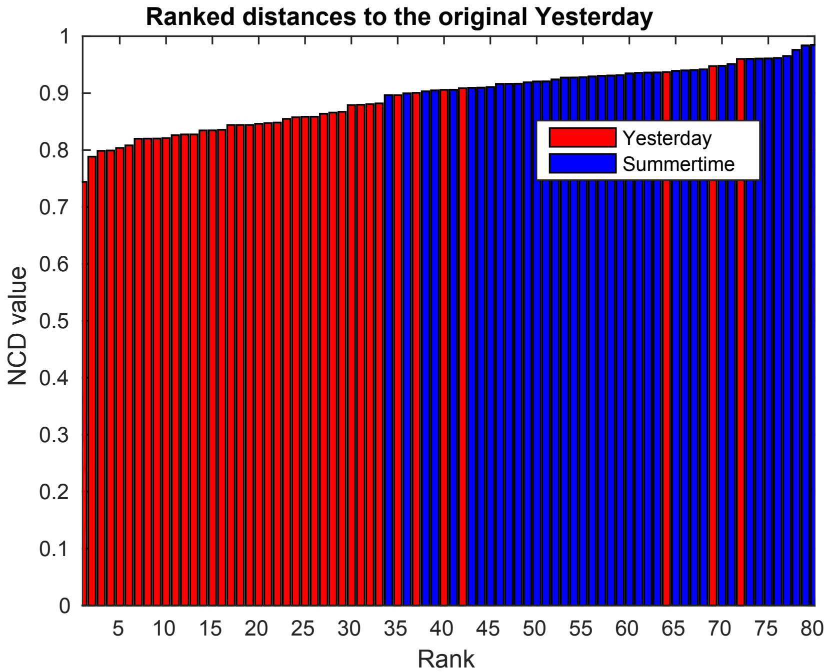

Figure 4.15: Normalized compression distances between the original version of Yesterday and all cover versions of both Yesterday and Summertime, using information from Chapters 5 and 6

high, but this local similarity is evidently lost in the global similarity.

We can also consider difficult cases not to be the ones that are difficult to identify, but instead cases that seem to be similar to various pieces of music. As stated in [103], pieces based on simple, repetitive harmonic structures are problematic, as they can be deemed similar to other pieces containing similar information. For our work, this is problematic, as pieces with long, repetitive sequences compress efficiently into small lengths and thus bias the measuring.

4.6 Remarks

After studying some of the most common variations in cover songs we now have several observations of what actually makes the identification such a difficult process. We can fairly state that several global invariances can be obtained through preprocessing (e.g. key) or with a suitable distance

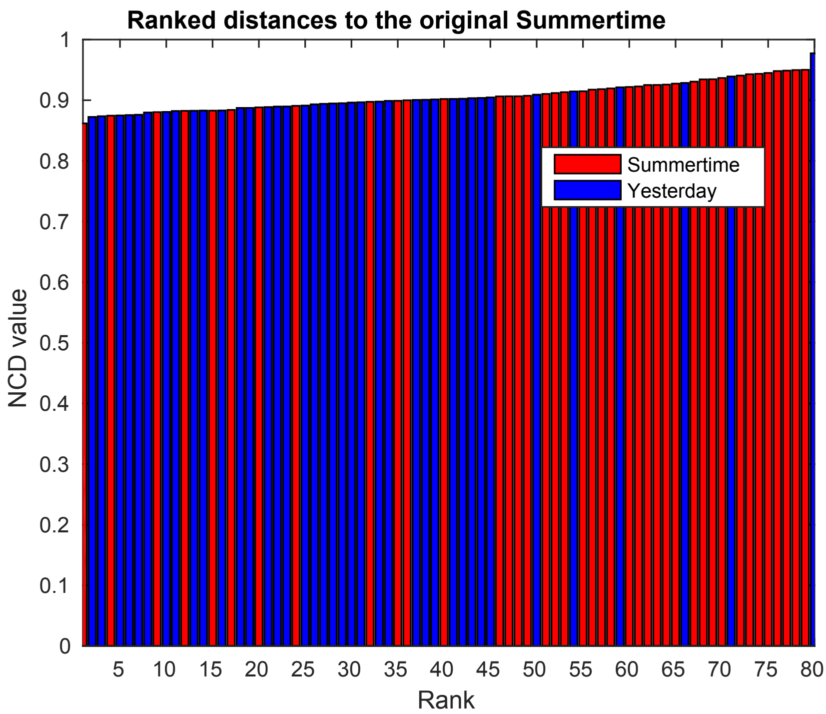

Figure 4.16: Normalized compression distances between the original version of Summertime and all cover versions of both Yesterday and Summertime, using information from Chapters 5 and 6

measure (e.g. structure), at least to a tolerable precision. The challenge lies more in the variations that hide the essential characteristics of the pieces, such as lead melodies. Such variations are mainly changes in arrangements, whereas smaller variations in lead melodies can be captured to some extent.

Observing some of the easiest and the most difficult cases of our experiment data has suggested that the easier versions do not need to be highly similar in their tempi, arrangements, and structures. As our methodology focuses strongly on harmony, it is often enough to detect the essential harmonic progressions from the pieces, and based on the results obtained by the more sophisticated methods from latter parts of this thesis, we are able to include even more information from the music into the identification, thus efficiently removing the confusion caused by unrelated pieces with similar harmonic progressions. With the more difficult versions, it seems that the identification cannot rely on lengthy harmonic similarities,

		Predicted	
		Yesterday	Summertime
Actual	Yesterday	36	4
	Summertime	11	29

Table 4.2: Results for confusion experiment between the two datasets, with distances measured with various features and several steps of preprocessing.

but instead should be able to detect the small musical cues that are present in both pieces.

After all, the question of cover song identification can be reduced into a process of robustly detecting similarities between multi-dimensional time series. However, considering the nature of the problem, including musical knowledge in the process would seem beneficial. In our work, this is addressed; we do not solely compare numerical values of the time series, but try to detect the essential musical characteristics from the chromagram data.

4.6.1 Compression-based Similarity

As the purpose of this thesis is to study the suitability of the compression-based similarity metric for the particular task of tonal similarity measuring, some observations need be to discussed here. Several issues with compression-based similarity have already been mentioned; most notably, the fact that compression-based similarity is a highly successful similarity metric with symbolic data, but might have performance issues with time-series data. Additionally, the length of the sequences might be problematic; in [64] it is stated that compression-based similarity measuring has been difficult with shorter time series. Another challenge was discovered with tempo invariance, we noticed that changes in tempo might have notable effects on the NCD value. This might not be an issue, though; several of the cases considered easy were not performed in the same or nearly same tempo as the original.

Still, the advantages are present. We noticed that NCD is structurally invariant, and several of the most notable challenges in cover song identification are questions of features and representations. In the following sections of this work, we will address the question of finding suitable representations for compression-based similarity measuring and offer possible solutions.

Chapter 5

Retrieval Experiments

In this chapter, we conduct a series of real-data cover song identification experiments with the compression-based similarity metric. We examine the effect of various parameters for both the features and the compression algorithms, and study the identification performance of different quantized chromagram features.

5.1 Evaluation setup

In our experiments, we follow a common cover song identification evaluation procedure. The evaluated system takes in two lists of audio files; that is, lists of n query (i.e. pieces of music we wish to find cover versions for) and m target (including pieces of music both relevant and irrelevant to the query) files, and produces a $n \times m$ matrix of pairwise distances. The performance is then evaluated as a retrieval task, by measuring how the relevant versions of the composition are returned for a query. For an illustration of the system components and how the query and target data are processed in the identification task, see Figure 5.1.

5.1.1 Test Data

We are aware of only two commonly available datasets for cover song identification. The *covers80* dataset[1] is the oldest and it has been used as a benchmark for various studies, but the size of the dataset is rather modest (only 80 pairs of original and cover versions), and more specifically, the low cardinality of the cover song sets could lead into unrealistic results [75, 102]. Another commonly available dataset is the more recent *SecondHandSongs*

[1] http://labrosa.ee.columbia.edu/projects/coversongs/covers80/

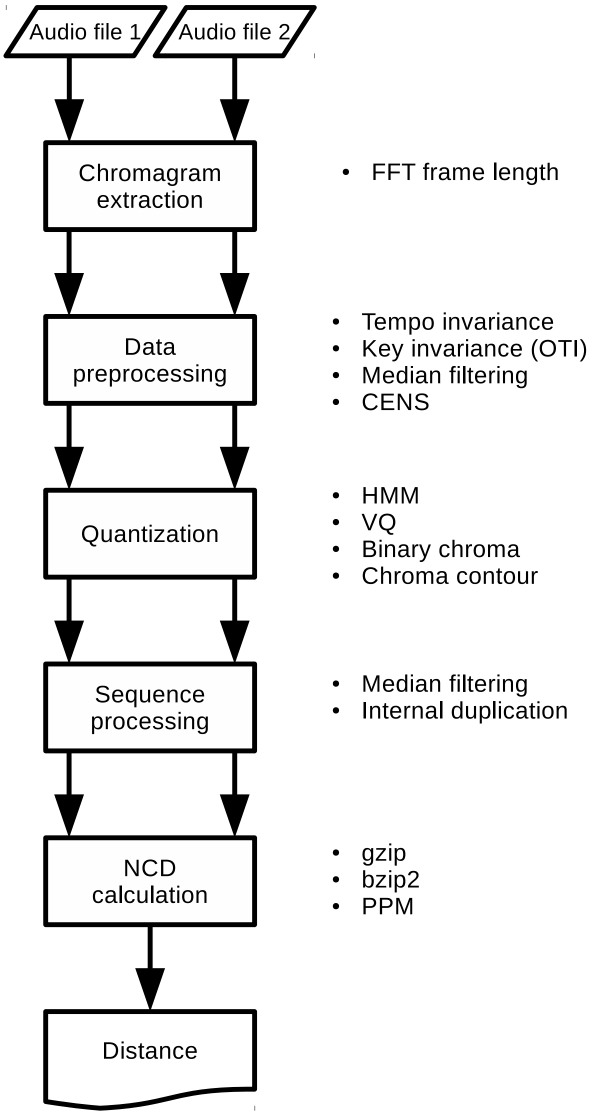

Figure 5.1: The data processing of NCD evaluation between two audio files illustrated.

dataset[2], a subset of the Million Song Dataset [22]. The SecondHandSongs dataset is vast (over 18 000 pieces of music) and diverse, but only provides access to the already extracted features, limiting the possibilities for the experiments conducted here.

Eventually, we chose to compile a new dataset for the evaluations. Named *Mixed*, the dataset was constructed in a similar fashion to the MIREX evaluation dataset [37]; it consists of 30 cover song sets, each comprising an original recording of the piece and 10 cover versions. All cover versions are performed by artists different from the original performance (thus, there are e.g. no live versions by the original artist), and a few versions are remixed versions of the original, containing audio segments taken directly from the original performance. See Appendix C for the detailed content of the Mixed dataset.

In addition to the cover sets, the dataset includes 670 "noise" tracks, pieces of music unrelated to the cover song sets. They are compositions performed by unrelated artists, and are mostly from the same genre as the original performances of the cover sets. We refer to the whole dataset as $Mixed_{1000}$, whereas a subset with no noise tracks is referred to as $Mixed_{330}$. Whereas the $Mixed_{330}$ yields information on how well the method distinguishes covers from other sets of covers, the $Mixed_{1000}$ is not only more difficult because it is larger, but also because it presumably includes a larger variety of chroma profiles.

Due to copyright restrictions, we are unable to distribute the original audio data we used for the Mixed dataset. However, for the sake of test reproduction, all extracted features, representations, source codes, and distance matrices are provided as an electronic appendix to this work. See Appendix D for information on how to obtain the electronic appendix.

5.1.2 Evaluation Measures in Identification and Retrieval

The elementary measures for performance of a retrieval scheme are *precision* ($Prec$) and *recall* (Rec). For a collection of documents R, with a subset of relevant documents R_a and a set of retrieved documents A, these are defined as $Prec = \frac{|R_a|}{|A|}$ and $Rec = \frac{|R_a|}{|R|}$. In other words, precision is the fraction of the retrieved documents which is relevant, whereas recall is the fraction of the relevant documents which have been retrieved. Combining both precision and recall into a single value is often useful and can be done several ways; one such is the harmonic mean, also known as the F-measure, $F = \frac{2}{\frac{1}{Prec}+\frac{1}{Rec}}$. [14]

[2] http://labrosa.ee.columbia.edu/millionsong/secondhand

The above measures are based on unordered answer sets. As our system returns pairwise distances, the answer set has a natural ranking. This makes it more convenient to measure the identification based on the order of the answer set. For this, we have chosen two measures commonly used in information retrieval.

Mean Average Precision (MAP) For a single query, average precision is the average of the precision values at the recall level of each relevant document [14]. MAP is thus the mean of average precisions over all queries. Formally,

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Prec(R_{jk}), \tag{5.1}$$

where Q is the set of queries (here $|Q| = 330$), m_j is the number of relevant documents for the query j (here, $m = 10$ for all queries), $Prec(R)$ is the precision value for the set R, and R_{jk} is the set of ranked retrieval results from the top results until document k [76]. MAP for a perfect answer set is 1, in our case this would mean that for every query all ten relevant cover versions would have the smallest distances. MAP has been shown to have good discrimination and stability, and it also pairs both precision and recall into a single measure, as MAP is roughly the average area under the precision-recall curve for a set of queries [76]. In addition, MAP has been widely applied in various evaluations of information retrieval, including the MIREX cover song identification task since 2007 [37]. According to [13], the expectation value of average precision is calculated as

$$E[AP] = \frac{1}{k} \sum_{i=1}^{N} (\frac{p_i}{i}(1 + \sum_{j=1}^{i-1} p_j)), \tag{5.2}$$

where p_i is the probability of seeing a correct document in rank i; for our work, this is $\frac{10}{999}$ for all i with the $Mixed_{1000}$ dataset. In our work, the expectation value of the average precision is the same for all queries, thus it is also the expectation value of MAP. For our data, these values are thus 0.0492 and 0.0174 for the $Mixed_{330}$ and the $Mixed_{1000}$ datasets, respectively.

Mean Reciprocal Rank (MRR) As the name implies, this measure is based on the reciprocal for the rank of the first correctly returned document of a query, and averaged over all queries. Whereas MAP measures the overall performance of the system, MRR yields additional information on how well the system is capable of identifying at least one correct version amid all target files. Formally, MRR is defined

$$MRR(Q) = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}, \tag{5.3}$$

where $rank_i$ is the rank of the first correct answer for query i. For perfect identification, the MRR value would be 1.

In addition, we will also calculate the mean distances and the standard deviations (sd) of the distances between queries and their both correct and incorrect pairs. This is to give an intuitive view on the distinguishing power of the measured scheme. Trivially, in order to be successful, the mean and sd values should be clearly lower for the correct pairs and higher for the incorrect. However, as the correct pair values are only calculated from ten pairs in contrast to the values of the 989 incorrect pairs, there is more bias caused by outliers in the correct pair values. Also, smaller mean values for correct pairs do not imply that the nearest neighbor for the query is one of the correct targets.

5.2 Identification Experiments

The purpose of the following identification experiments is to validate the selected methods and parameters that produce the optimal identification results. Unless otherwise stated, we use chromagrams extracted using a window of 0.3715 seconds (i.e. 16384 samples for audio signal with a sample rate of 44100 Hz), with no overlap between subsequent windows, and apply the bzip2 algorithm for data compression, with OTI used to obtain key invariance. The feature used here is the chord sequences obtained via 24-state HMMs.

We use all the original versions and their covers as queries one after another, thus totaling 330 different queries, and the whole 1000 pieces of the Mixed dataset as targets, and report values for both Mix_{1000} and the Mix_{330} subset. With the identity case (i.e. $NCD(x, x)$) ignored, we have 10 correct pieces for each query included in an answer set of 999 (329 for Mix_{330}) pieces, ranked according to their descending similarity values.

5.2.1 Effect of Compression Algorithm Selection

To begin our experiments, we start from the very basis of the compression algorithm selection. We experimented with three commonly used lossless data compression algorithms: *gzip*, *bzip2*, and *ppm*. These algorithms contain a good variety of most common data compression techniques; gzip is a Lempel-Ziv dictionary coder [119], bzip2 a hybrid compressor that applies Burrows-Wheeler transform for block-sorting compression [26], and ppm (prediction by partial matching) is a statistical compressor that applies arithmetic coding [35]. The results are presented in Table 5.1.

The results prove that the choice of the compression algorithm has a significant impact on the results. The gzip algorithm has a clearly weaker identification accuracy than the others, and the ppm algorithm in turn seems to be the best choice; we made a similar observation in [4]. During the following experiments, we will, however, show that after several steps of additional processing the bzip2 algorithm can provide even higher results than ppm. These steps do not have a similarly remarkable effect on the accuracy of ppm, and we will discuss this further in the thesis. From now on, the experiments are therefore conducted with bzip2 unless otherwise noted.

In Table 5.2 we provide the mean distance values and distance standard deviations (sd) for all correct (corr) and incorrect (incorr) pairs with all compression algorithms. The values show that there are clear differences in the performances of the compression algorithms; ppm seems to provide overall smaller distance values, but the relative difference between values of the correct and incorrect pairs is also smaller. With bzip2, the relative distance is wider, but with the correct pairs, there is a slightly larger variance in the distance values than with the incorrect pairs.

Dataset	Algorithm	MAP	MRR
Mix_{330}	gzip	0.1747	0.4311
	bzip2	0.2620	0.5478
	ppm	**0.2786**	**0.5973**
Mix_{1000}	gzip	0.1055	0.3098
	bzip2	0.1829	0.4547
	ppm	**0.1903**	**0.4646**

Table 5.1: Results for different compression algorithms.

Algorithm	Mean (corr)	sd (corr)	Mean (incorr)	sd (incorr)
gzip	0.7934	0.0275	0.8130	0.0303
bzip2	0.7296	0.0488	0.7828	0.0476
ppm	0.6400	0.0253	0.6541	0.0312

Table 5.2: Distance value statistics for different compression algorithms.

5.2.2 Effect of Chromagram Parameters

Chromagram Length

We consider the length of the chromagram extraction window to be important for two reasons. First, the window should be long enough to contain a meaningful amount of tonal information; too short a window length would likely result in noisy, uninformative chromagrams. Second, the length of the window affects the length of the chromagram (i.e. the longer the analysis window the shorter the extracted chromagram), and for a compression-based similarity measuring, we might assume that a longer chromagram is more advantageous; keeping in mind that the universality of NID holds true only for infinite sequences, the approximation is likely to be better with longer sequences (that is, extracted with a shorter window length). We experimented with chromagram windows of 0.7430, 0.3715, 0.1858, 0.0929, and 0.0464 seconds, with no overlap between subsequent frames; several experiments unreported here suggested that the overlap had very little effect on the identification accuracy. The results of the window length experiments are presented in Table 5.3.

Contrary to what could have been expected, the larger window size yields a higher identification accuracy, until the accuracy again drops rather steeply with the largest size experimented here. This is due to the sequences turning very short – with the largest window, a three-minute piece of music is only approximately 240 frames long, and with some short but complex cases, the file length of the compressed version of a single chord sequence file was actually *larger* than the uncompressed version. The best performance with a window of 0.3715 seconds is surprising, as it seems that the compression algorithm would benefit from the longer chroma sequences produced by the smaller window size. However, the results suggest otherwise. The longer chroma frames do, however, reduce the amount of transients and other noisy chroma frames in the sequence, thus representing the tonal content of the piece in a more robust way. Apparently, this length describes important musical characteristics.

Dataset	Window length (s)	MAP	MRR
Mix_{330}	0.7430	0.2207	0.4743
	0.3715	**0.2620**	**0.5478**
	0.1858	0.2323	0.5524
	0.0929	0.1620	0.4122
	0.0464	0.1269	0.3274
Mix_{1000}	0.7430	0.1391	0.3443
	0.3715	**0.1829**	**0.4547**
	0.1858	0.1570	0.4522
	0.0929	0.0935	0.3065
	0.0464	0.0694	0.2244

Table 5.3: Results for different window lengths of chromagram extraction.

Again, we took a look at the mean values and standard deviations for the distance values. These are reported in Table 5.4. Not surprisingly, the largest relative difference between correct and incorrect distances appears with the best performing chromagram window length. A more interesting notion is that the mean standard deviations are smaller with incorrect pairs for the larger window lengths. Also, the values show that smaller distance values do not imply higher distinguishing power.

Chromagram cleaning

The chromagram data is likely to contain noisy segments, transients, and outliers that harm the identification process. A common technique to remove such outliers is to apply median filtering to the chromagram data. The results with different lengths for the median filter window are depicted in Table 5.5.

The results provide a clear notion that the identification accuracy weakens as the filter window grows. However, filtering with a window of length 3 provided the highest MAP values for the smaller $Mixed_{330}$ dataset, but with only a relatively very narrow difference, and for MRR, the best results for both sets are obtained without any filtering. Based on this, it seems that the chromagram filtering is not required, and can even be harmful with too large filter window sizes, where identification power is lost as the chromagram data is stripped from its characteristics. With smaller window sizes, the differences are modest, suggesting that the HMM quantization process provides similar robustness against minor outliers in chromagram values. For different quantization methods, though, the chromagram filtering might

Window length (s)	Mean (corr)	sd (corr)	Mean (incorr)	sd (incorr)
0.7430	0.6782	0.0542	0.7259	0.0540
0.3715	0.7296	0.0488	0.7828	0.0476
0.1858	0.7652	0.0467	0.8193	0.0517
0.0929	0.8208	0.0386	0.8589	0.0419
0.0464	0.8548	0.0281	0.8828	0.0338

Table 5.4: Distance value statistics for different chromagram window lengths.

Dataset	Median window length	MAP	MRR
Mix_{330}	1	0.2620	**0.5478**
	3	**0.2641**	0.5463
	5	0.2510	0.5231
	7	0.2054	0.4294
	9	0.1793	0.4396
Mix_{1000}	1	**0.1829**	**0.4547**
	3	0.1793	0.4460
	5	0.1619	0.3929
	7	0.1288	0.3254
	9	0.1098	0.3368

Table 5.5: Results for median filtering of chromagrams. Median window length 1 means that no filtering is applied.

be more useful, as observed in [9].

Instead of median filtering, we also experimented with the CENS representation. CENS (Chroma Energy Normalized Statistics) [85] representation takes the chromagram information and in order to increase robustness post-processes the data with two steps. First, the frame-wise chromagram values are quantized, according to how the energy is distributed amongst the bins; by default, the quantization thresholds are 40, 20, 10, and 5 per cent of the total energy of the frame. Then, the quantized vectors are first convolved component-wise using a Hann window, and then the whole sequence is downsampled and the vectors are normalized. The purpose of the quantization is to reduce the noise caused by the note attacks, whereas calculating statistical information smooths the data and balances the differences between note groups such as arpeggios. Using CENS provided fair results for audio matching with classical music variations in [85]. We applied the quantization for the chromagram frame values, but did not apply the downsampling, as this would result in sequences far too short for compression-based similarity measuring. The results of the CENS representation experiment, in contrast to the unprocessed chromagram data, are presented in Table 5.6.

The CENS representation does not provide higher identification results. Again, the characteristics of the pieces are lost in the cleaning process, and the confusion in identification grows; the HMM-based quantized representations become too trivial and similar.

It seems that all attempts of chromagram cleaning actually make the identification less accurate. However, with individual queries, and even query sets of a certain cover song set, there is a minor improvement in the results. It seems that the median filtering or CENS representations could be applied with some data; however, with a larger sets of data, the identification should start from the premise of not applying any cleaning for the chromagram data.

5.2.3 Invariances

After we have discovered the best parameters for extracting chromagram data from the pieces of music, we will turn our focus into obtaining robustness over global differences between the pieces; namely, tempo and key invariance, which we already discussed in Section 2.2.

Tempo invariance

In order to achieve tempo invariance, we apply beat-synchronous chroma features. For beat-synchronous chroma features, we use the method described in [40] and apply the original implementation[3] to our chromagrams. The method estimates the beat locations from the audio signal and averages the chroma frames that belong in the same beat. The retrieval results of beat-synchronous chromagrams in comparison to the 0.3715 second sample window non-synchronous chromagrams of the previous experiment are presented in Table 5.7.

Again, the unprocessed chromagram data provides a slightly higher identification accuracy. It seems that the small deviations in tempi can be overcome with a suitable quantized representation and compression-based similarity measuring, and the overall similarity measuring between two sequences is more reliant on the large-scale similarities than the minor variations caused by tempo differences.

[3] http://labrosa.ee.columbia.edu/projects/coversongs/

Dataset	Algorithm	MAP	MRR
Mix_{330}	Chromagram	**0.2620**	**0.5478**
	CENS	0.2240	0.4745
Mix_{1000}	Chromagram	**0.1829**	**0.4547**
	CENS	0.1461	0.3660

Table 5.6: Results of the CENS representation, in contrast to the basic chromagram representation.

Dataset	Feature	MAP	MRR
Mix_{330}	Beat-synchronous	0.2235	0.5274
	Non-synchronous	**0.2620**	**0.5478**
Mix_{1000}	Beat-synchronous	0.1465	0.4351
	Non-synchronous	**0.1829**	**0.4547**

Table 5.7: Results of the tempo invariance estimation.

Key Invariance

Key invariance is clearly a highly important factor in cover song identification. Here, we will try several different methods for key invariance. The Optimal Transposition Index (OTI) used in other experiments is also included in this experiment. We will also apply our own representation-based key invariance used in [8, 4]. In this representation each chord transition is depicted as a symbol that represents the semitone difference between the root notes of the chords and implies whether there is a change between a major and a minor chord or not; thus, this is an alphabet of size 24. We also utilize the key estimation for the pieces with the method of MIRToolbox [66], where the chromagram data is compared against key templates, and transpose each piece to a common key of C major (or, in the case of a minor key, into the relative key of A minor). Finally, the brute force approach is applied; here, each query is matched with every possible transposition and the smallest distance is returned as the final distance between the pieces. The results are listed in Table 5.8.

Based on the results, it is clear that key invariance needs to be concerned, as the results with no aim for key invariance are clearly worse than the others. Nevertheless, the representation-based key invariance is only slightly better. This is most likely due to the fact that as chord changes occur only relatively seldom between chroma frames, there are long runs of

Dataset	Key Invariance	MAP	MRR
Mix_{330}	none	0.1681	0.4466
	OTI	**0.2620**	**0.5478**
	representation	0.2018	0.4597
	key estimation	0.2444	0.5198
	brute force	0.2498	0.5362
Mix_{1000}	none	0.1132	0.3651
	OTI	**0.1829**	**0.4547**
	representation	0.1219	0.3561
	key estimation	0.1681	0.4254
	brute force	0.1759	0.4464

Table 5.8: Results of the key invariance method selection.

"no change" -symbols in the sequences, and as all representations for the pieces of music consist of such long runs of similar symbols, there is a loss of identification accuracy. Key estimation performs second best, but it is likely that the method fails to find the correct key in some cases. A bit surprisingly, the brute force approach did not provide the best results.

We wanted to explore this more closely. In [101], the performance of OTI was evaluated, and it was also shown that using more than just one possible transposition provides better identification accuracy, and by using two most likely transpositions the results are in par with the brute force (i.e. all twelve transpositions). In our work, however, considering more possible transpositions lead to worse results. In Figure 5.2 the effect on both MAP and MRR is depicted while considering 1 to 12 most likely OTI transpositions. The trend is clear, and although there is some fluctuation, the range of changes in both MAP and MRR is very small, and such fluctuation can be caused by only a few different distance values. It seems that even though using several transpositions might give benefit in some correct cases, the overall effect is lost as more false positives are deemed to have a smaller distance. Also, the results suggest that several false positives have already been measured with the optimal transposition, and they have a distance value that is always smaller than that of the correct pair.

Tuning invariance

We described the need for tuning invariance in Subsection 4.3.3. We argued that applying the tuning algorithm using the 36-dimensional chromagrams seemed to produce a highly different kind of chord sequences, and because

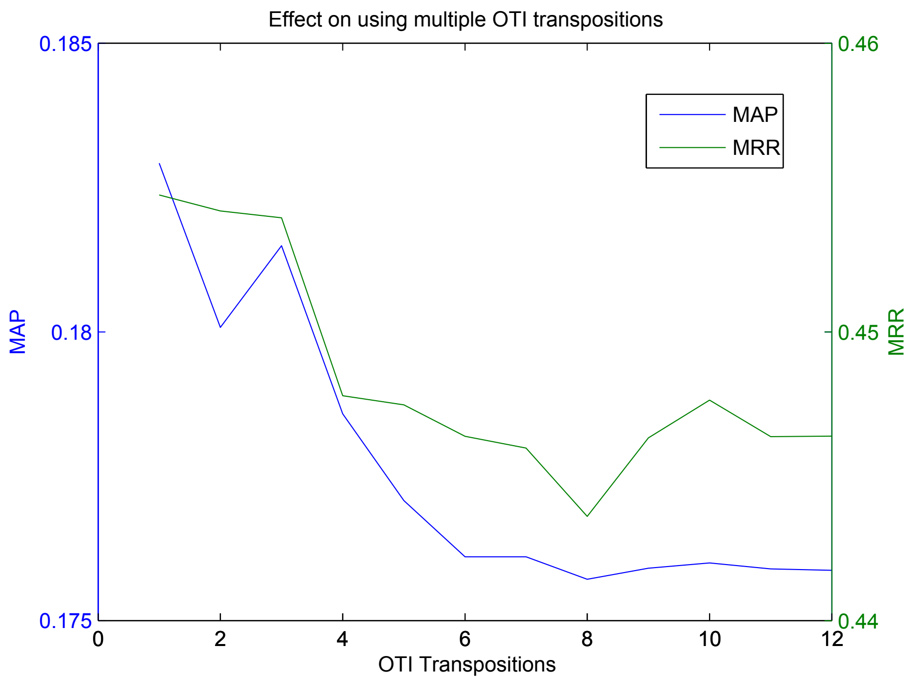

Figure 5.2: The effect of using more than one possible transposition value candidate.

of this, we felt it was necessary to run the full-scale evaluation using sequences from both tuned and untuned chromagram data. The results of this experiment are presented in Table 5.9.

Based on the experiment it seems that applying tuning invariance is not necessary, but in contrast harmful. The reason behind this is likely the observation made in Subsection 4.3.3: the tuning causes chord sequences to be cleaner. These cleaner sequences in turn are compressed more efficiently, and the more efficient compression reduces the distinguishing power. The absolute differences in identification accuracies are not significant, but the relative differences are rather high, and because of this, we will ignore the tuning in the following experiments.

We took a closer look at the *Mixed* dataset, and it seems that from the 1000 pieces of music 817 seem to be in the 440 Hz concert pitch (or at least near enough not to need tuning), whereas 112 were considered to be in a sharper pitch and needed tuning, and 71 were similarly considered to be flatter and tuned. These values seem rather high, considering that

Dataset	Algorithm	MAP	MRR
Mix_{330}	Tuned	0.2270	0.5169
	Untuned	**0.2620**	**0.5478**
Mix_{1000}	Tuned	0.1516	0.4002
	Untuned	**0.1829**	**0.4547**

Table 5.9: Results of chromagram tuning.

our data consists mostly of popular music recorded in the 1960s or later – apparently, the tuning algorithm might not be a fully reliable solution to begin with.

5.2.4 Feature Representations

We have used the Hidden Markov model-based chord estimation method to quantize the chromagram vectors to sequences of symbols representing an estimation of the triad chord sequences of the pieces. However, several other methodologies exist, and here, we will compare them.

Hidden Markov models

In addition to the 24-chord estimation of [19], we will apply the 12-chord estimation we suggested in [5]. This method is based on the similar HMM topology as in [19], but with chords that have only the root and fifth note (this will be discussed in detail in Subsection 6.4.1). The purpose of this representation is to eliminate the problems caused by the possible unclarity of the triad of the chord; this representation was originally invented as we noticed that with some pieces of music, we had confusion and oscillation between the major and minor chords of the same root note. The results for these two HMM-based representations are presented in Table 5.10; the 24-chord HMM sequences provide higher identification accuracies. However, the 12-chord HMM sequences do have their advantage in feature combination which we will discuss in the next chapter.

Vector quantization

We already discussed vector quantization in Subsection 2.3.1, and are aware that the choice of the codebook is crucial. Initially, we experimented with k-means clustering for the chromagrams in order to learn the codebook, but regardless of the size of the codebook, the amount of chromagram

Dataset	Number of states	MAP	MRR
Mix_{330}	24	**0.2620**	**0.5478**
	12	0.2191	0.5057
Mix_{1000}	24	**0.1829**	**0.4547**
	12	0.1383	0.3906

Table 5.10: Results for two HMM-based representations.

frames used for learning, or any other parameters, we ended up with rather dissatisfying identification results. We took a closer look at the learned codebooks and noticed that in most cases, they mainly comprised two kind of codewords; nearly binary codewords with only one chroma bin having a high value while the rest having considerably smaller values, or codewords with values almost the same for each bin. The first set is produced mostly due to the fact that the chroma frames are normalized according to their maximum value, thus all chroma frames have one peak, or occasionally a few peaks, whereas the second set mostly resulted from the amount of "flat" chroma frames, usually present at the beginning and the end of the pieces. Expanding the amount of learning data or the codebook size had very little effect, and produced mostly variants of the single peak and totally flat codewords. Also, the unequal distribution of different keys in the pieces makes it quite difficult to create representations that allow effective key invariance.

Eventually, we chose to use a manually constructed codebook, and similarly to the HMM, we chose to use musical knowledge for the codebook vectors. We experimented with several different binary codebooks that represent musical chromagram frames.

- Similar to the ones we learned, but binarized: we had 12 vectors with only dimensions with the value 1; naturally, the size of this alphabet is 12, and we refer to this as $12a$.

- Codebook consisting of binary vectors that reflect the root and fifth notes of chords; here, thus, each codebook has two dimension bins with value 1, for example bins 1 and 8. This also has a size 12 alphabet and is referred to as $12b$.

- Triad chord codebook, similar to the μ vectors of the HMM parameters; that is, codebook vector dimensions have value 1 according to the major or minor chord they represent; for example, a codebook vector has 1 on dimension bins 1, 5, and 8 (i.e. a major C chord). Here, the size of the alphabet is 24.

- Similar to the previous, but with a new set of chords added. Here, the included chords are suspended chords; again, for an example, a codebook vector has 1 on dimension bins 1, 6, and 8 (i.e. a C suspended 4th chord, identical to F suspended 2nd chord). The alphabet size is 36.

- Similar to the previous, but added with a set of chords that represent diminshed chords. An example codebook vector has the value 1 on bins 1, 4, and 7 (C diminshed chord). The alphabet size is 48.

- Similar to the previous, but added with a set of chords that represent augmented chords. An example codebook vector has the value 1 on bins 1, 5, and 9 (C augmented chord, identical to E augmented and G♯ augmented chord). Here, the codebook size is 52.

The results for this experiment are presented in Table 5.11. The pre-set codebook results are far better than any experiment where the codebook was learned, but it still fails to meet the level of HMM-based quantization.

We assume that vector quantization could possibly be applied with even higher results with more sophisticated codebooks. Similarly to the HMM-based representations, these VQ-based representations describe mostly harmonic content of the pieces, whereas a representation that would describe a richer tonal content might provide higher results. However, as the size of the codebook increases over 48, the identification accuracy drops slightly.

Binary chromagrams

In [86], similarity measuring between pieces of music was performed using binarized chromagrams. In their work, the chromagrams are binarized according to whether a pitch class is present in the frame. For our work, we needed to set a threshold to determine whether a pitch class is present. Here, we just chose to experiment with different values instead of attempting any kind of heuristics; if the value was above the threshold, the corresponding bin was set to 1, otherwise 0. This gives us a rather large alphabet as there are 2^{12} different binary chromagram frames, but in practice various note combinations never occur, making the actual alphabet smaller. The results with different threshold values are presented in Table 5.12.

The results show that the binary chromagram representation does not achieve the identification accuracy of the previously presented quantization

Dataset	Codebook	MAP	MRR
Mix_{330}	12a	0.1448	0.3789
	12b	0.1866	0.4663
	24	**0.1957**	0.4690
	36	0.1862	0.4690
	48	0.1934	**0.4933**
	52	0.1761	0.4361
Mix_{1000}	12a	0.0797	0.2687
	12b	0.1120	0.3572
	24	0.1158	0.3408
	36	0.1165	0.3522
	48	**0.1202**	**0.3635**
	52	0.1034	0.2932

Table 5.11: Results for vector quantization-based sequences. Codebook refers to the size of the codebook; 12a is the codebook with binary vectors with one dimension, 12b the codebook with two dimensionals with value 1.

methods. Likely, a more sophisticated method of binarization could provide better identification, but the poor level of results suggests that the representation is impractical for our task.

As an alternative approach, we also experimented with a representation we presented in [7]. This representation, called chroma contour, represents the chromagram as a sequence of values that describe the OTI transformation value between the frame and the global chromagram of the piece. A major advantage here is that the representation is completely key-invariant. Results for this representation, in comparison to the best-performing representations of the previously mentioned experiments and the HMM baseline representation, are presented in Table 5.13.

The HMM-based representation towers clearly above the other representations. As stated before, one of the major advantages of HMM is that it considers the temporal element of music; the subsequent symbols in a representation are not completely independent of each other, similarly as notes in a piece of music are not independent of the notes preceding them.

Sequence filtering

In contrast to filtering chromagram data, we did experiments with filtering the sequences produced by the hidden Markov model. We noticed in [9] that filtering sequences improved the identification results; this is mostly

Dataset	Threshold value	MAP	MRR
Mix_{330}	0.6	0.1420	0.3825
	0.7	**0.1463**	**0.4077**
	0.8	0.1425	0.3836
	0.9	0.1294	0.3651
Mix_{1000}	0.6	0.0816	0.2700
	0.7	**0.0861**	**0.2955**
	0.8	0.0823	0.2791
	0.9	0.0682	0.2545

Table 5.12: Results for binary chromagram sequences.

Dataset	Quantization	MAP	MRR
Mix_{330}	Hidden Markov model	**0.2620**	**0.5478**
	Vector quantization	0.1934	0.4933
	Binary chroma	0.1463	0.4077
	Chroma contour	0.1243	0.3154
Mix_{1000}	Hidden Markov model	**0.1829**	**0.4547**
	Vector quantization	0.1202	0.3635
	Binary chroma	0.0861	0.2955
	Chroma contour	0.0603	0.2033

Table 5.13: Comparison of results for best-performing different quantization techniques.

due to the fact that removing the outliers from the sequences increases the compressibility of the sequences. Naturally, this might also mean that overall filtering produces representations that lose their characteristics, and thus lead into sequences that have a small compression distance with various unrelated pieces of music.

As with the chromagram filtering, we did the median filtering with various values for the sequence representations before measuring their similarity with NCD. The results for different median filter window length values, in comparison to no filtering at all, are presented in Table 5.14.

Based on the results, the most efficient length of the median filtering window seems to be three for the larger dataset, and three or five for the smaller dataset. The overall effect, however, is rather modest.

Dataset	Median filter window length	MAP	MRR
Mix_{330}	1	0.2620	0.5478
	3	0.2668	**0.5649**
	5	**0.2670**	0.5553
	7	0.2405	0.5188
	9	0.2192	0.5187
Mix_{1000}	1	0.1829	0.4547
	3	**0.1863**	**0.4699**
	5	0.1805	0.4607
	7	0.1628	0.4378
	9	0.1464	0.4253

Table 5.14: Results for median filtering of sequence data.

5.2.5 Internal Duplication

One of the most considerable advances in recent years of chromagram similarity measuring has been the use of technique known as embedding. Presented by Serra et al in [104], the embedding of chromagram data is based on time series embedding, which has been found highly useful when analyzing time series data. We described the embedding in Subsection 2.3.2.

In our work, applying embedding in a similar sense as in [104], is hardly practical. Turning the 12-dimensional chromagram data into a representation of 120-dimensional state space vectors does not make our work any easier. Actually, it makes it downright harder, as we would need a method for quantization of these vectors, and concerning the difficulties of the 12-dimensional vector quantization, it seems that the most convenient way would be building a 120-dimensional HMM for the task, and this, on the other hand, can be considered to be a very challenging task.

By experimenting, we eventually came up with a method that provided more distinguishing power. Instead of such an approach, we turn our focus to the "embedding" the quantized chroma sequences, and for the lack of a better term, we refer to this as *internal duplication*, as it in practice duplicates our sequence data internally. This might sound a slightly trivial solution, but as the results show, it actually provides a remarkable improvement in identification accuracy. Formally, we will turn a quantized chromagram sequence $C = \{c_1, c_2, \ldots, c_n\}$ with an embedding dimension of D into a representation of

$$C^* = \{c_1, c_2, \ldots, c_D, c_2, c_3, \ldots, c_{D+1}, \ldots, c_{n-D}, c_{n-(D-1)}, \ldots, c_n\}.$$

The internal duplication does not “embed” the data in the sense of the time series analysis, but rather enhances the different subsequences in them. See Figure 5.3 for a visualized example of how a short sequence of characters turns with our internal duplication, with different values of D.

As with time series embedding, the internal duplication is highly dependent on the parameters used. We do not pay attention to the embedding step τ here (as it would make very little sense considering the subsequent nature of HMM-based chord sequences, and thus we fix $\tau = 1$), but the choice of the embedding dimension D is crucial. We experimented with several possible values of embedding, and the results are depicted in Table 5.15.

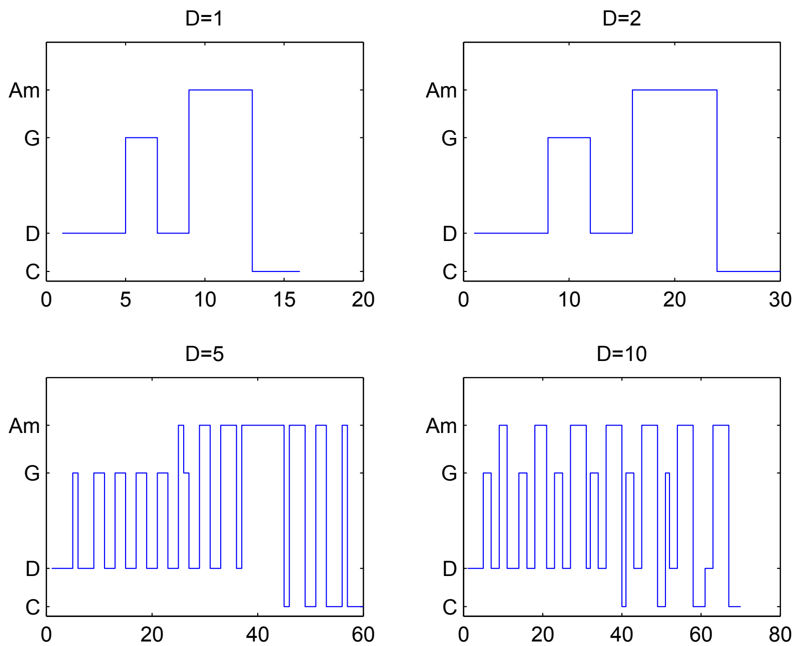

Figure 5.3: Internal duplication of a toy example chord sequence with different values of D. Case $D = 1$ is the original sequence.

Dataset	Duplication value	MAP	MRR
Mix_{330}	1	0.2620	0.5478
	2	0.2421	0.5449
	3	0.3285	0.6458
	4	0.3626	0.6737
	5	0.3648	0.6785
	6	0.3654	0.6750
	7	**0.3663**	0.6890
	8	0.3615	**0.6898**
	9	0.3600	0.6815
	10	0.3619	0.6734
Mix_{1000}	1	0.1829	0.4547
	2	0.1680	0.4636
	3	0.2401	0.5485
	4	0.2778	**0.5867**
	5	0.2734	0.5839
	6	**0.2783**	0.5788
	7	0.2761	0.5864
	8	0.2677	0.5839
	9	0.2639	0.5731
	10	0.2598	0.5578

Table 5.15: Results for internal duplication of sequence data. Duplication value 1 refers to unprocessed sequences.

The results of Table 5.15 suggest that the best value for the duplication seems to be six. This is likely a data-dependent value, but it should be noticed that the identification accuracy increases with almost every value of D. An interesting exception is the case $D = 2$, where the value actually drops from not using duplication at all. Also, there are no major differences in results between all values of D above 3.

The reason the internal duplication works can be addressed to the effect it has on the data compression.The increased amount of repetition in the data is clearly beneficial in order to learn a model from the data. Although the duplication does improve results with the bzip2 and gzip algorithm, it does not provide better results for the ppm algorithm. We assume that the first two algorithms benefit from the duplication because the algorithms strive to find repetitions from the data, and the duplication enhances the repetition greatly. However, the nature of the ppm algorithm is, as the name states, to *predict* the content of the string that is compressed, by learning from the contexts where the symbols appear. The duplication as applied here forces the algorithm to learn a very strict model, with high probabilities for the symbols in their given contexts. This makes the model overfit to the sequences, and this naturally is the opposite of the robust model that allows the different scales of variations to be included in the cover versions. Although the strict model of a piece of music is rather beneficial in the sense that it might eliminate the false positives, it can be unhelpful when turned into too limiting a model.

5.3 Summary of the Chapter

We have experimented with various compression algorithms, tonal features, representations, and their parameters. Out of all these, several stand out as useful and provide notably better results. In conclusion, we have found the following combinations to provide the highest values:

- Chromagram window of 16384 frames and a hop factor of 1, meaning no overlap between subsequent frames. (Unreported experiments suggest that the hop factor plays a rather insignificant role.)
- No beat synchronization or other techniques to obtain tempo invariance are required. Apparently, NCD seems to be robust against tempo invariances.
- No need for chromagram data filtering or other cleaning; actually, this seems to be harmful.
- Key invariance using OTI. Additional improvement could not be obtained by taking into consideration other likely transpositions; instead, this weakens the identification accuracy.
- HMM-based chord estimation as the quantized representation. HMM initialized with musical knowledge produces clearly the most useful quantized representations. The temporal element obtained with the HMM is also a clear advantage.
- Median filtering is not essential for the sequences. Nevertheless, in order to obtain the best results, filtering with a window of three frames is suggested for a slight improvement.
- Internal duplication of sequences with a value between four and ten; the highest MAP was obtained with the value six. This step can be

a major advantage with a block-sorting or dictionary-based compression algorithm; however, this provides a negative effect with prediction by partial matching compression algorithm.

- Using either ppm or bzip2 as the compression algorithm. Although bzip2 algorithm performs with an accuracy almost in par with ppm, it seems that the ppm scheme provides a more robust similarity measuring. However, the bzip2 is also more responsive to the improvements, and eventually provides highest accuracies.

Putting all this together, we now have a best-performing approach. Results for this combination of system components, invariance choices, parameters, and processing techniques are presented in Table 5.16. A major improvement is achieved with the help of the internal duplication; the advance caused by the sequence filtering is limited in comparison.

A significant notion is that the MRR value is rather low even with the best-performing combinations, suggesting that the system has shortcomings on identifying a correct cover version as the most similar piece in the dataset. Compression-based similarity measuring with the features and representations used seems to capture a broad-scale similarity between two pieces, but for a more successful identification, more attention should be paid to the smaller detail similarities between pieces. Achieving this by using a different representation is problematic, as this would require sequences with a larger alphabet, which in turn makes the compression more inefficient. One solution to this – feature combination – will be discussed in the next chapter.

5.3.1 Computational Costs

It is clear that the compression-based algorithm is hardly the most efficient solution for calculating similarities between pieces of music. Although compression algorithm implementations are usually optimized for a fast performance, the cumulative cost for pairwise similarity measuring for data amounts on scale of the $Mixed$ dataset used here grows large. And the similarity measuring is only a part of the process; feature extraction,

Dataset	MAP	MRR
Mix_{330}	0.3766	0.6902
Mix_{1000}	0.2891	0.6058

Table 5.16: Results for our best-performing approach.

representation production, and invariance calculation all add to the overall computational cost.

To provide an insight on the amount of computational labour required, we present example time requirements for a single query. We took the query with the median length of all the 330 queries in our dataset; the happens to be query ID 18, with 3:34 minutes duration in real time and 576 time frame representation with our commonly-used analysis window of length 0.3715 seconds . In Table 5.17 we first present the computational times required for the single file to be processed; extracting chromagram from the audio data, estimating the chord sequence from the chromagram with the HMM-based approach, and writing the chord sequence into a file. Then, we present pairwise similarity measuring times (calculated and averaged over all 1000 target files); first, the OTI calculation of most likely transposition, and then, the NCD value, calculated using the bzip2 compression algorithm. Finally, overall computational times are presented; first, a total sum of all parts of the process computed for a single query, and then multiplied in order to achieve the overall computational cost for all 330 queries. All run times in Table 5.17 were calculated on a standard desktop computer[4].

A good deal of the calculations presented in this work was performed on the computational cluster of the Department of Computer Science, University of Helsinki[5], using as much pre-calculated material (such as OTI transpositions) as possible and parallelizing computation into smaller subsets of the whole data. With such approach, we could perform the computation far more efficiently, at best in under half an hour wall-clock time. In practice, one of the most time-consuming parts was actually the constant reading, writing, and compression of the files.

We want to stress that the focus of this work has been on the retrieval accuracy instead of computational efficiency. Arguably, the computational time could be optimized without any loss of identification accuracy, for example by using more efficient programming languages. Even after the optimization, the system presented here is clearly not very scalable and might not be practical for a real-world application with large amounts of audio files and very strict time limits. However, we hope that the ideas presented here could be to some extend applied when producing a more robust, large-scale cover song identification system.

---

[4] Intel core i5-2400 3.1 GHz processor, 8 GB RAM

[5] https://www.cs.helsinki.fi/en/compfac/high-performance-cluster-ukko

5.3.2 Comparison with State of the Art

As we are using our own dataset for the evaluation, the results can not be directly compared to results reported in the works of other researchers. In order to be able to measure our performance, we need to run experiments with other algorithms to our data. We chose the algorithm presented in [104] as our comparison, and refer to it as *SSA*; the algorithm was explained in Subsection 2.3.2. SSA has so far obtained the highest results in the MIREX cover song evaluation task[6]. As we are unaware of any better performing cover song identification systems, we consider this to be the state of the art.

We conducted the evaluation with two versions of SSA. The first was done using parameters and values from [104], with the only exception being that we use chromagram data of 0.3715 second frame length. The second uses parameters and additional processing presented in [105]; here, the similarity estimation calculates two most likely OTI values (instead of just one), and then calculates the similarity between query and both transposed targets, and uses the higher similarity value as the final outcome. Additionally, the similarity value is considered as distance by using the target length as a normalizing factor. We refer to these versions as SSA and SZA, respectively.

The results for the Mixed dataset with the SSA and SZA are presented in Table 5.18 with comparison to the best-performing combination of our system components. We also take again a small sneak peek at the following chapter; the presented results are those obtained in this chapter (Chapter 5), and those obtained in the next (Chapter 6).

[6]http://www.music-ir.org/mirex/wiki/2009:Audio_Cover_Song_Identification_Results

Subprocess	Time required (in seconds)
Chromagram extraction	1.996
Chord estimation	0.867
Writing sequences	0.325
OTI calculation	0.961
Pairwise NCD computation	0.145
Complete processing of the files	3218
Total similarity matrix computing	368198

Table 5.17: Computational times for parts of our best-performing approach in wall-clock time.

Dataset	System	MAP	MRR
Mix_{330}	Chapter 5	0.3766	0.6902
	Chapter 6	0.4105	0.7275
	SSA	0.5432	0.8370
	SZA	**0.5752**	**0.8675**
Mix_{1000}	Chapter 5	0.2891	0.6058
	Chapter 6	0.3263	0.6583
	SSA	0.4803	0.7794
	SZA	**0.5029**	**0.8110**

Table 5.18: Results for our best-performing approach and cover song detector of [104, 105].

The comparison yields two major observations. First, there is a gap between the performances: both SSA and SZA perform better, and the gap is significant. Second, we want to stress that also the results for SZA are remarkably below the 0.75 MAP value obtained in the MIREX evaluation; this suggests that the evaluations performed here are conducted with a far more difficult dataset.

It is also notable that SZA clearly benefits from the additional steps of [105]. According to [101], using two potential OTI transposes provides accuracy nearly identical to the brute force approach, and all in all gives a significant boost in comparison to using only one OTI value. As witnessed before, this does sadly not provide a better identification accuracy for the NCD-based cover song identification, but instead causes more confusion in the set of all distance values. The normalization of the distance value, however, is something that NCD does automatically, although with sequences of only several hundred symbols, there is likely some bias caused by differences in sequence lengths.

We wish to pay more attention to the performance differences between our system and SSA. Mostly, we are interested in the cases where SSA performs better than our system, as this should reveal important information of what could be improved in our work. The differences between our work and SSA with relation to the query sets of the data are depicted in Figure 5.4.

Observing the values of Figure 5.4 reveals several interesting notions. It seems that various query sets (most notably the sets 21 and 30) of our dataset are nearly impossible to detect for SSA also. In addition, in one case (set 9), our work actually performs better. Still, there are clear cases

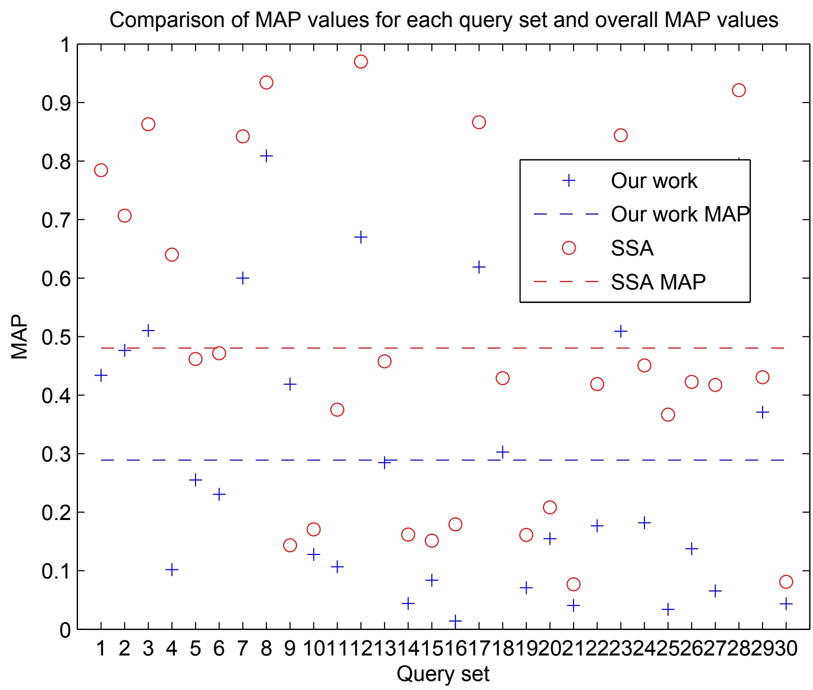

Figure 5.4: Comparison between our work and state of the art (entitled SSA). Mean of average precisions are presented for each 30 query sets, whereas the lines denote the overall MAP performance.

where SSA provides a notably higher identification accuracy. The most notable difference between the results occurs with query set ID 4. Also, with query sets $1, 3, 23, 25$, and 27 the difference in the performances is also rather big. None of the listed query sets seem to bear any significant difficulties or other quirky characteristics, but the performance with NCD is rather modest whereas SSA seems to detect distinguishing similarities from them.

The results raise the question of whether our choice of representation has been sound. SSA does not quantize the chromagram data until the last phases when the cross recurrence plot is binarized. To see if the quantization process is a significant issue, we experimented with chord sequences, binary matrices, and the *QMax* similarity measure of [104]. In this experiment, we constructed a $m \times n$ binary similarity matrix M_{XY} from chord sequences $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_n)$ by setting

$M(i, j) = 1$ if $x_i = y_j$ and $M(i, j) = 0$ otherwise, and then applied the QMax for the matrix. We used QMax with similar penalty values as in [104], and are aware that they probably are not optimal for our matrices. For an example of binary similarity matrices constructed by the system described in [104] and the method described above, see Figure 5.5. We ran this setup for our test data, and obtained MAP values of 0.3862 and 0.2967, and MRR values of 0.6858 and 0.5863 for the $Mixed_{330}$ and $Mixed_{1000}$ datasets, respectively. These values are clearly below the performance of SSA, but they are also above the results of the basic NCD-based method. This hints that the HMM-based representation, although not perfect, is still clearly workable, and with additional sequence processing and better parameter selection with QMax, might be even closer to the accuracy of SSA. By applying the parameters and settings of [105], we managed to get even higher results with chords and QMax: MAP values of 0.4520 and 0.3578 and MRR values of 0.7334 and 0.6510 for $Mixed_{330}$ and $Mixed_{1000}$ datasets, respectively.

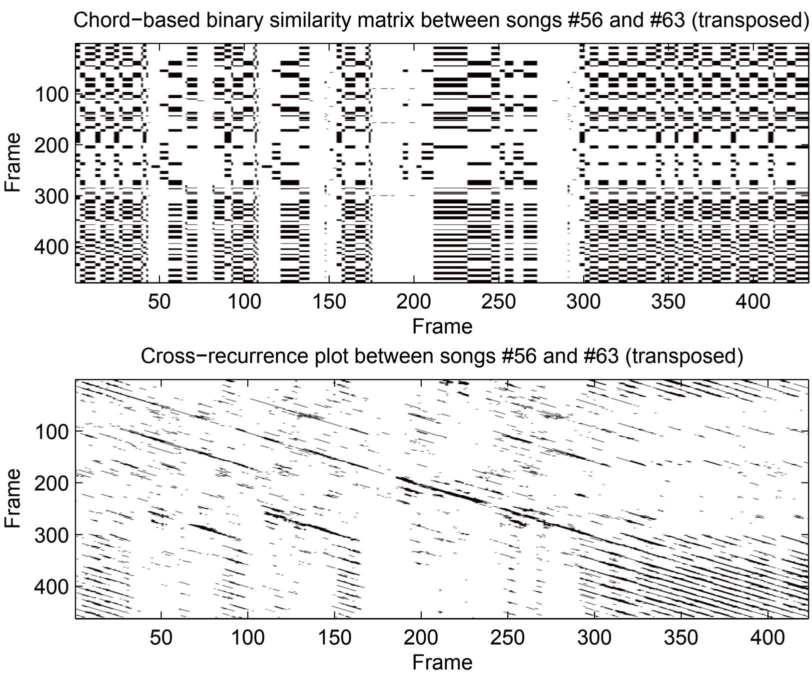

Figure 5.5: Binary similarity matrix examples between an original performance and a related cover. Black cell in visualization means value 1.

Based on this, we can now provide some light to what makes the state-of-the-art version work better than our proposition. It seems that one of the most important differences is that the system of [104] searches for the lengthtiest subsequence shared by the two sequences whereas our work measures global similarity between the sequences. The state-of-the-art algorithm emphasizes the similarity between the sequences only when the sequences share portions that have a very small distance between them, whereas NCD-based similarity measuring focuses on the overall similarity between two pieces of music, allowing several biases to be caused by sections of music that are considered unrelated. The foremost is closer to the way a human listener detects a cover version. Compressing full song-length sequences makes the focus move from small, but notable, nearly similar tonal characteristics to the more global song-level similarities. This could be highly problematic, as the global similarity measure between two sequences is likely to be small when most of the two sequences can be considered similar, and this is yet again highly dependent on the representation of the data; even though, for example, the chord sequences underlying in the pieces of music might be similar, they are easily confused by different pitches caused by the differences in the arrangements. However, at the same time the focus on global similarity can also be an advantage, as the quasi-universal nature of NCD should provide a distance value based on the most common similarity between the strings.

Chapter 6

Feature Combination

In this chapter, we observe how combination of different features provides better identification results than cover song identification based on a single chromagram feature. We propose three combination strategies and evaluate them using the same dataset as in the previous chapter.

6.1 Motivation

Combination of features is a rather commonly used method in CBMIR; for example, several well-performing methodologies in genre classification (e.g. [81]) combine different features in order to achieve a higher accuracy. This traces back to various methods and applications in machine learning, where combination of features, measures, and classifiers is used frequently.

For cover song identification, feature combination seems like a suitable idea, considering that the tonal similarity between pieces might be more likely captured in different representations; although the chromagram data contains a good deal of the tonal information of a piece of music, it is still, for example, an octave-folded representation, thus perhaps obscuring several important characteristics of the piece. Despite the potential though, this area has so far not been highly targeted in cover song identification. With chromagram data, we suggested this in [5]. Additionally, several other methods applying feature combination have appeared, with [99] providing notable results. Also, in [94] three different approaches were combined into a single version detection system.

6.2 Melody Estimations

We have so far used only quantized chromagram data, with the highest identification accuracies obtained with a method based on chord estimation and, as such, weighting the importance of harmonic information and similarity. This could be viewed as a hindrance; it is trivial that several pieces of music include similar harmonic progressions even though they are completely different compositions in every other way. Although the melody-based approach has not been highly successful (see Section 2.1) in cover song identification, it obviously seems to be a suitable complement for retrieval based on harmonic information. Various methods of melody extraction exist; see [98] for a comprehensive review.

For a mid-level melody feature, we utilize here a melody estimation system by Antti Laaksonen [63]. The method returns a one-dimensional sequence of MIDI note values that represent the salient melody of the piece, with a single note for a frame. In order to make the melody sequence lengths consistent with the chromagram lengths, we use the same 0.3715-second analysis window.

Similarly to the chromagrams, the melody sequences for different pieces are likely to have variations that need to be addressed. The key invariance is important, but unlike with chroma that is folded into one octave, we need to consider the octave invariance. Here, we experiment with four representations:

- **Absolute values** Here, we take the melody sequence as it is. For key invariance, we calculate the OTI from corresponding chromagrams and transpose the target melody up the required amount of semitones. Because of this, the transposition might produce melodies one octave apart, which leads to the next representation.

- **Absolute values with octave transposition** Here, the representation and key invariance is similar to the previous representation, but with an additional step where the melody sequence is transposed so that the most frequent note of the melody lies between $C3$ and $B3$.

- **Octave-folded melodies** Here, all note values are stripped from their octave information. Thus, the melody is similar to the 12-character melody taken from the chromagram, but extracted with a different methodology.

- **Melodic contour** Here, the melody is represented as the semitone difference between subsequent notes. The representation is thus both key and octave invariant, but as seen in Subsection 5.2.3, such representation might not be practical with compression-based similarity measuring.

The results of the four proposed representations are presented in Table 6.1. The octave-folded representation provided the highest results, although it loses the octave information; this has more to do with high compressibility of the sequences that are constructed. However, there still are enough distinctive qualities, whereas the similarly straightforward melodic contour is far too trivial for compression-based distance measuring. See Table 6.2 for the mean values and standard deviations for the distances.

Bass melody

Use of bass melody has provided efficient results in [99]. Although the bass lines themselves are likely to have variations between the cover versions, and as such being unsuitable as a single feature for the identification process, it could be a beneficial addition, as several cover versions might share a highly similar bass line.

We experimented with base melodies, again obtained using the algorithm of [63], but limiting the analysis range between MIDI notes corresponding to notes E1 and C3. As we learned from the previous experiment, the best representation for the melody is octave-folded, and we use it with bass melodies also.

The results for bass melody experiment are presented in Table 6.3 in comparison with the higher scale octave-folded melody of the previous experiment. Judging by the MAP values, the bass melody actually provides a slightly higher identification, whereas the MRR values are better for the higher scale melodies. However, looking at the average precision values of all individual queries (Fig. 6.1), it is evident that the two different melodies help to detect different pieces of music (while, in several cases, neither provide very little distinguishing power at all). For now, we will continue to use the higher scale melodies, but will return to using bass melodies later in this chapter.

Chromagram-based melodies

In [5], we applied a melody estimation taken from the chromagram data. Here, the chroma bin with the highest energy is selected as the note, thus creating a sequence of octave-folded notes with an alphabet of size 12. As the octave-folded melodies obtained with a more sophisticated method proved to be the best choice for the identification task, we would like to

Dataset	Melody representation	MAP	MRR
Mix_{330}	Absolute value	0.1437	0.3971
	Octave-transposed	0.1304	0.4053
	Octave-folded	**0.1763**	**0.4782**
	Melody contour	0.1073	0.3123
Mix_{1000}	Absolute value	0.0930	0.3138
	Octave-transposed	0.0802	0.3163
	Octave-folded	**0.1166**	**0.3930**
	Melody contour	0.0569	0.2193

Table 6.1: Results for different melody representations.

Melody representation	Mean (corr)	sd (corr)	Mean (incorr)	sd (incorr)
Absolute value	0.8423	0.0425	0.8723	0.0395
Octave-transposed	0.8496	0.0478	0.8766	0.0415
Octave-folded	0.7903	0.0373	0.8218	0.0376
Melody contour	0.8073	0.0332	0.8278	0.0356

Table 6.2: Distance value statistics for different melody representations.

see how well the chromagram-based melodies perform in comparison to the melody estimations. We took both the normal chromagram melody estimations, and also calculated a bass chroma by using chromagram extraction to frequency range of $[54, 110]$ Hz. The results of this experiment, in comparison to the melody estimations, are presented in Table 6.4. The straightforward chromagram-based melodies do not achieve the accuracies of the more advanced melody estimations, but they are only a small step behind. For now, though, we will retain the estimations produced by [63].

Dataset	Bass melody feature	MAP	MRR
Mix_{330}	Higher scale	0.1763	**0.4782**
	Bass melody	**0.1870**	0.4541
Mix_{1000}	Higher scale	0.1166	**0.3930**
	Bass melody	**0.1362**	0.3901

Table 6.3: Results for bass melody features.

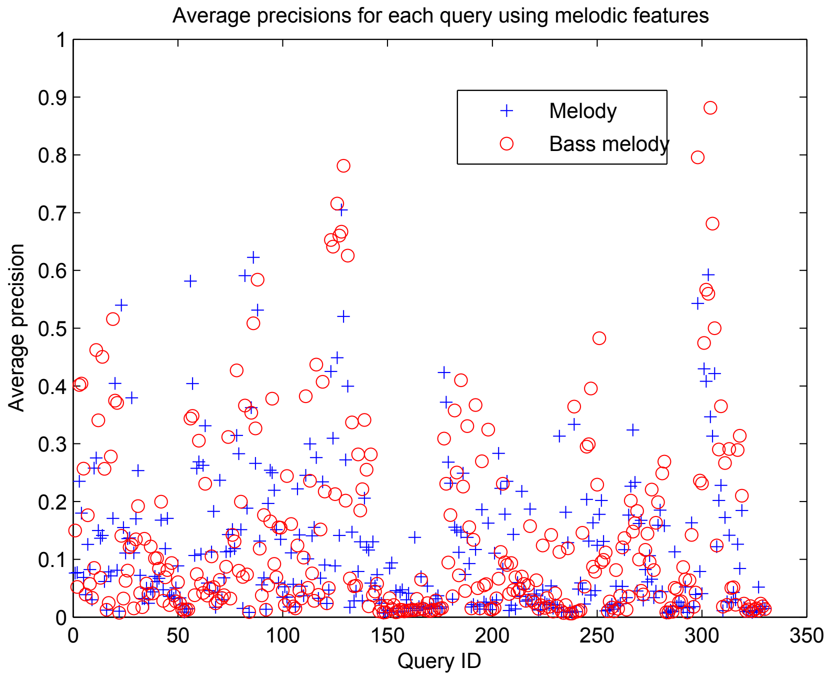

Figure 6.1: Average precisions for each query, using both higher scale and bass melodies.

6.3 Combination Strategies

We approach the feature combination with three different strategies; creating combined feature representations, combining different representations into single representations, and combining the distances calculated for individual features.

6.3.1 Strategy One: Combination of Features

The first strategy seems rather straightforward. Considering that the tonality of music consists of lead melodies and their accompaniment, an intuitive starting point can be seen as a combination of melody and chord estimations. To represent them in a single symbol, we take each frame of the same moment in time for both chroma and melody, and combine them by creating a tuple of $(note, chord)$. We label the tuples, so that each individual tuple has a distinctive label. As we use the octave-folded notes, this gives

Dataset	Melody representation	MAP	MRR
Mix_{330}	Melody estimations	0.1763	**0.4782**
	Melody estimations, bass	**0.1870**	0.4541
	Chromagram melodies	0.1634	0.4268
	Chromagram melodies, bass	0.1618	0.4000
Mix_{1000}	Melody estimations	0.1166	**0.3930**
	Melody estimations, bass	**0.1362**	0.3901
	Chromagram melodies	0.1073	0.3463
	Chromagram melodies, bass	0.1095	0.3213

Table 6.4: Results for melody estimations in comparison to chromagram-based melodies.

us a relatively large number of 24×12 different tuples. In order to reduce this even further, we propose four different representations with different alphabet sizes:

- Combining notes with relation to the chord. Here, the tuple receives a binary value label depending on whether the note in the tuple is present in the chord or not; that is, for example, a tuple with C major chord note values of c, e, and g would be labeled 1, and with other notes the tuple would be labeled 0. The size of this alphabet is thus 48.

- Combining notes with relation to the key related to the chord. Here the previous is extended by labeling notes that do not belong to the triad chord into two categories according to whether they are harmonically related to the chord. Here, we use the namesake key of the chord to determine the harmonic relation; if the note of the tuple belongs to this key, we label the tuple 2 and otherwise 0. A tuple with notes of the triad is again labeled 1. For example, a C major chord tuple with notes c, e, or g would be labeled 1, a tuple with notes d, f, a, or b would be labeled 2, and a tuple with other notes would be labeled 0. Here, the alphabet size is thus 72.

- Combining notes with relation to the notes of the chord and to the key related to the chord. Here the previous is extended by labeling tuples with notes that belong to the chord according to the note's position in the chord; if the note is the same as the root of the chord, the tuple is labeled 1, if the note is the same as the triad, the tuple is labeled 2, and if the note is the same as the fifth, the tuple is labeled

3. Again, if the note is related to the namesake key, the tuple would be labeled 4, and with any other note, the tuple would be labeled 0. For C major chord, the tuple with note c would be labeled 1, the tuple with note e as 2, the tuple with note g as 3, the tuple with note d, f, a, or b 4, and with other notes 0. Here, the alphabet size is thus 120.

- Combining notes individually regardless of the chord. Here, every tuple of a chord and a note would have a distinctive label, thus totaling the 288 different tuples. This representation, though rich in describing the music, has a rather large alphabet in contrast to the lengths of the pieces of music.

The results of this strategy, experimented with different tuple sizes, are presented in Table 6.5. In addition, we also report the previous results of the single features. The combination with alphabet size 48 provided the highest accuracies; however, the improvement in contrast to using only chord-based representations is limited.

6.3.2 Strategy Two: Combination of Representations

This strategy is based on combining the representations into one lengthy representation. Formally, we have a chord sequence $C = \{c_1, c_2, \ldots, c_n\}$ and a melody sequence $M = \{m_1, m_2, \ldots, m_n\}$, and these will be combined into new representation. We experiment with two different combination strategies.

- Concatenating the representations. Straightforwardly, C and M will be combined into $CM = \{c_1, c_2, \ldots, c_n, m_1, m_2, \ldots, m_n\}$.

- Merging the representations. Here, C and M would be combined by alternatively taking symbols from both, one at the time, $CM = \{c_1, m_1, c_2, m_2, \ldots, c_n, m_n\}$.

Both strategies allow trivial addition of novel features, and should boost the similarity by making the sequences longer and thus underlining the similarities in sequences with the compression algorithm; even if some of the features would not be deemed similar by the algorithm, the similarity of other features should compensate. Also, the alphabet will remain smaller than with the previous strategy. We begin again with a combination of chords and octave-folded note values. The results for this strategy are presented in Table 6.6, again with comparison results of the two features used alone.

Dataset	Feature	MAP	MRR
Mix_{330}	Chord estimations	0.2620	0.5478
	Melody estimations	0.1763	0.4782
	Tuple, $\|\Sigma\| = 48$	**0.2653**	**0.5997**
	Tuple, $\|\Sigma\| = 72$	0.2512	0.5608
	Tuple, $\|\Sigma\| = 120$	0.2476	0.5349
	Tuple, $\|\Sigma\| = 288$	0.2517	0.5459
Mix_{1000}	Chord estimations	0.1829	0.4547
	Melody estimations	0.1166	0.3930
	Tuple, $\|\Sigma\| = 48$	**0.1841**	**0.4948**
	Tuple, $\|\Sigma\| = 72$	0.1770	0.4590
	Tuple, $\|\Sigma\| = 120$	0.1798	0.4479
	Tuple, $\|\Sigma\| = 288$	0.1802	0.4613

Table 6.5: Results of combining features into a single representation, with comparison to using only single features.

Again, the results prove to be dissatisfying; both combinations are better than melody used alone, but when compared with the chord estimations, the identification accuracy is practically on par with the concatenation, and below with the merging. The concatenation does not make the process significantly slower, but the gained improvement is hardly worth it. As with strategy one, we will not continue further with this strategy, as there seems to be very few possibilities for improvement.

6.3.3 Strategy Three: Combination of Distances

This strategy is based on a strategy known as mixture of experts; here, the similarity between two pieces is obtained by calculating individual pairwise distances for each feature and then combining them into a final pairwise distance value. For combination, we take the mean of the distance values as the final value. To put it formally, this means that the final pairwise distance D between pieces of music x and y is calculated as

$$D(x, y) = \frac{\sum_{i=1}^{n} NCD_i(x, y)}{n}, \tag{6.1}$$

where n is the number of features and $NCD_i(x, y)$ is the normalized compression distance between x and y according to the feature i.

The results are presented in Table 6.7, again with comparison to using only single features. The combination of distances provides a higher identi-

Dataset	Feature	MAP	MRR
Mix_{330}	Chord estimations	**0.2641**	0.5463
	Melody estimations	0.1763	0.4782
	Concatenation	0.2513	**0.5698**
	Merging	0.2038	0.4866
Mix_{1000}	Chord estimations	0.1829	0.4547
	Melody estimations	0.1166	0.3930
	Concatenation	**0.1837**	**0.4848**
	Merging	0.1427	0.3943

Table 6.6: Results of combining features into a concatenated representation, with comparison to using only single features.

Dataset	Feature	MAP	MRR
Mix_{330}	Chord estimations	0.2641	0.5463
	Melody estimations	0.1763	0.4782
	Mean distance	**0.2821**	**0.5918**
Mix_{1000}	Chord estimations	0.1829	0.4547
	Melody estimations	0.1166	0.3930
	Mean distance	**0.2081**	**0.5111**

Table 6.7: Results of the combining features by using a mean distance value, with comparison to using only single features.

fication accuracy. The initial explanation seems to be that the normalized compression distance captures different similarities from different representations and eventually provides a satisfying result. However, this requires further investigation, and that will be focused on in the next section.

6.4 Details and Analysis of Feature Combination

6.4.1 Adding a Feature

In [5], we experimented with a chord estimation using a chord lexicon of only 12 chords, with the triads removed. We call this *power chord representation*, referring to the nickname of chords consisting only of the root and fifth notes. At first, the purpose of this representation was to overcome confusion between major and minor chords. However, we noticed it actually provides a different kind of representation that captures different characteristics of the pieces, and can be used in conjunction with 24-chord representations.

The initial HMM parameters for the 12 states are set as follows.

- Initial state distribution π: As there is no reason to favor any state before others, this is the same for each states (i.e. $\frac{1}{12}$).
- State transition matrix A: This is set according to a circle of fifths. For the C chord, the highest transition probability is to the chord itself, $C \to C$. This value is $\frac{6+\epsilon}{36+12\epsilon}$. The next similar chords are G and F chords, both sharing a note with the C chord, and the initial probabilities for both are $\frac{5+\epsilon}{36+12\epsilon}$. Eventually, the furthest chord from C is the $F\sharp$ chord, with probability $\frac{0+eps}{36+12\epsilon}$. The probabilities are set similarly to all states.
- Mean vector μ: The mean vectors are set by giving the value 1 to the pitch classes that are present in the corresponding power chord, and 0 otherwise. For the C chord, the vector is $(1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)$.
- Covariance matrix Σ: The covariance matrix for each state consists mainly of zeros. The diagonal is set to 0.2, apart from pitch classes present in the corresponding chord; these are set to 1. For non-diagonal matrix cells, the dominant of the root (i.e. the fifth) is set to 0.8.

Comparison of individual features

At the end of the previous chapter, we made notions that the identification accuracies can be improved by processing the sequences with median filtering and internal duplication. So far, we have not done this to the melody estimations or reduced chord sequences. So, we will now observe these effects with the features, and compare the combination with both basic and processed sequences. In Table 6.8 we present the results for each individual feature, both processed and unprocessed. The values of Table 6.8 display the effect of additional processing, as every feature clearly benefits from it.

6.4.2 Distance Calculation

Using mean distance as the combination strategy is clearly debatable. Using the minimum of the distances as the final outcome would seem a better idea; the NCD value for a correct pair should be very small for at least one feature. The inverse of this would of course be using the maximum distance as the ultimate distance, as this would likely reduce the amount

Dataset	Feature	MAP	MRR
Mix_{330}	Chord estimations	0.2620	0.5478
	Chord estimations, processed	**0.3766**	**0.6902**
	Power chord estimations	0.2191	0.5057
	Power chord estimations, processed	0.3336	0.6358
	Melody estimations	0.1763	0.4782
	Melody estimations, processed	0.2503	0.5624
Mix_{1000}	Chord estimations	0.1829	0.4547
	Chord estimations, processed	**0.2891**	**0.6058**
	Power chord estimations	0.1383	0.3906
	Power chord estimations, processed	0.2396	0.5411
	Melody estimations	0.1166	0.3930
	Melody estimations, processed	0.1748	0.4607

Table 6.8: Results of the combining features by different kind of arithmetics.

of possible fall positive cases. Also, mean values can easily be biased by outliers, making median distance a more sound solution. Using the three features listed above and distances calculated with them, we experimented with different distance combinations; in addition to mean we tried median, minimum, and maximum distances. The results are presented in Table 6.9.

The mean distance still provides the best results. Observing the results sheds some light on why this happens. In Figure 6.2 we depict the MAP values for each cover song set of the $Mixed_{1000}$ dataset with different distance selections. Using the minimum value as the final distance biases the detection by giving false positives a higher importance. For example,

Dataset	Feature	MAP	MRR
Mix_{330}	Mean distance	**0.4105**	**0.7275**
	Median distance	0.3749	0.6975
	Minimum distance	0.3792	0.6587
	Maximum distance	0.3226	0.6451
Mix_{1000}	Mean distance	**0.3263**	**0.6583**
	Median distance	0.2879	0.6113
	Minimum distance	0.2850	0.5740
	Maximum distance	0.2353	0.5451

Table 6.9: Results of the combined features by different kind of arithmetics.

with query set 27 the MAP value is rather worse when using the minimum distance. This is explained as the distance values with this dataset is at its minimum mostly with the 12-chord lexicon representation, and here it seems that this particular representation does not contain enough distinguishing power; similar sequences with long runs of a single chord are present elsewhere. Similar notions can be made with the maximum distance. With maximum distance, the overall average precision is usually behind the other alternatives; however, there are sets where the maximum would provide the highest MAP value. Closer observation suggests that the highest distance was in most cases the distance between the melodic sequences.

The slightly lower result of using median distance suggests that there are outliers in the correct pairwise distances. These outliers seem to be helpful and biasing identification in a more favorable direction. This might not be the case with different kinds of data, and thus we evade making any final conclusions on the suitability between the choice of using mean or median.

Figure 6.3 depicts an excerpt of a distance matrix obtained by the four different distance calculation methods. The excerpt depicts pairwise distances between pieces of music of three query sets (namely, query sets 11, 12, and 13). For these particular query sets, there is very little difference between the distance values; notably, with minimum distances, the matrix excerpt seems to be the most confused out of the four.

6.4.3 The Overall Effect of Combination

The mean of average precision values for different features and their combinations are presented in Table 6.10.

In addition to these results, the MAP values for each individual query set obtained with the best combination of features are depicted in Figure 6.4. Observing the values in Figure 6.4 reveals the query sets that benefited most from the combination, as well as some of sets with very little improvement. For some cases, the combination gives an additional boost to the already decent performance (and in some cases, may result in worse results than a single feature), while some of the sets remain near the values of a random baseline reference.

Case with most improvement

Several sets gained from the combination, and this is most notable with set 2, where the MAP value improved 26 per cent from using only the best

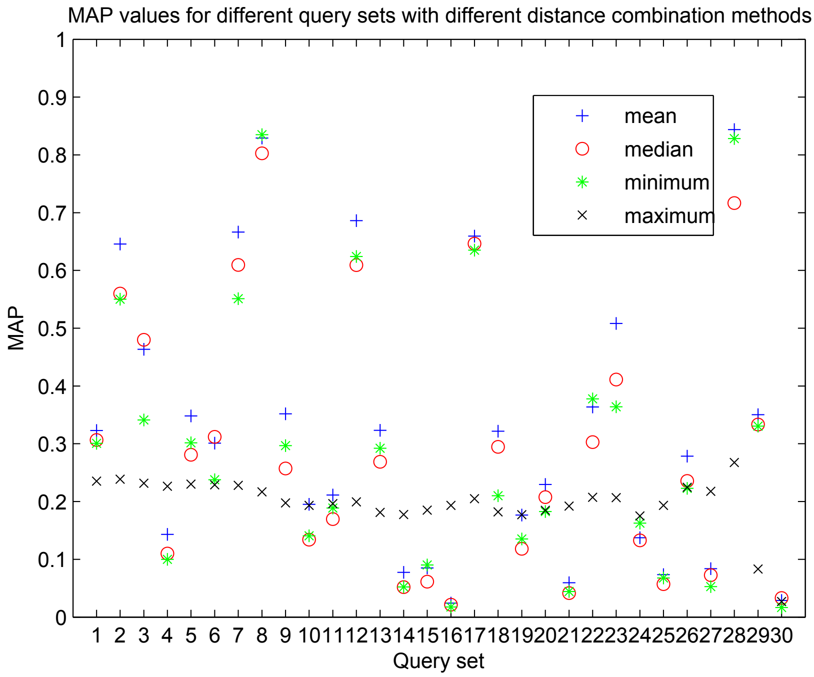

Figure 6.2: Mean of average precisions for query sets with different kind of combinations.

feature (power chords) and 35 per cent from using the basic features of Chapter 5. A closer look at this set suggests that this is due to the reason we have already proposed. For each unique feature, there is a rather limited amount of distinguishing power. The harmonic progressions and salient melodies are not particularly distinctive, but their combination makes the correct versions stand out from the false positives, as the false positives for melodic and harmonic features are to some extent distinct. And although the power chord representation works quite well with this set, it is not similarly useful with many other datasets, giving more motivation for combination.

The most difficult cases

Even with several combined features we can easily denote some datasets to be practically impossible to recognize. Most notably the query sets 16 and 30 seem to be greatly challenging with any feature or their combinations.

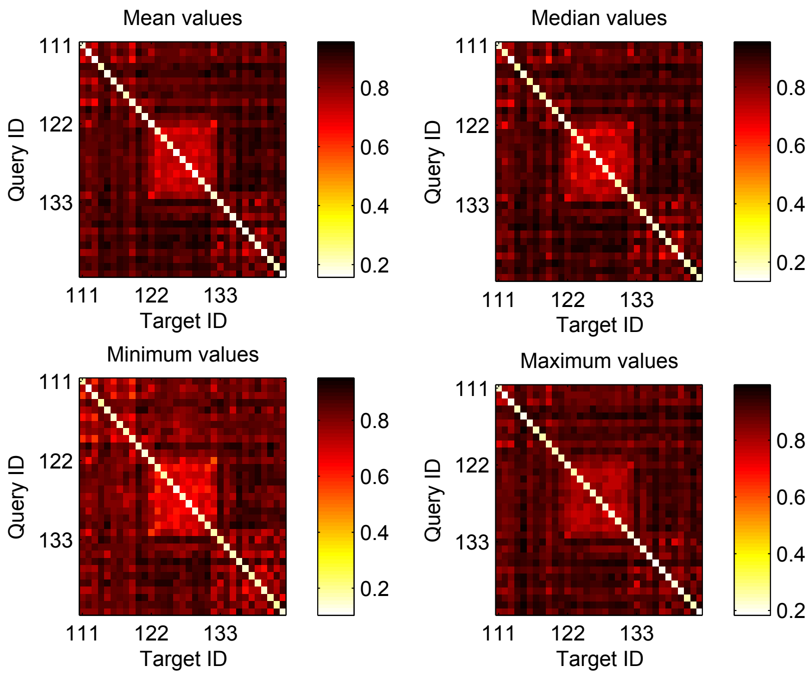

Figure 6.3: Excerpts from distance matrices with different distance calculation techniques.

Observing the pieces of the query set and the feature representations obtained from them reveals several reasons for this. We focus on set 30, as this was shown to be nearly impossible for the state of the art algorithm also.

The harmonic content of the original piece contains a rather meagre amount of variance; the main chord sequence can be denoted with only two chords, and the original version is driven by a guitar playing a power chord riff over this harmonic progression. The distinctive power chord riff is in some format shared with most of the cover versions, thus suggesting that the 12-state HMM quantization might provide a highly suitable representation for the pieces. But this does not always seem to be the case; the harmonic content is not very efficiently captured in the 12-chord representation. Whereas the 24-chord version sequences often mislabels chords due to their stripped-down nature of only two notes (i.e. occasionally the fifth of the power chord is denoted to be the root note, thus producing sequences that do not have as much in common as their actual tonal content

Dataset	Feature	MAP	MRR
Mix_{330}	Chord sequences	0.3766	0.6902
	Power chord sequences	0.3336	0.6358
	Melody estimations	0.2503	0.5624
	Combination	**0.4105**	**0.7275**
Mix_{1000}	Chord sequences	0.2891	0.6058
	Power chord sequences	0.2396	0.5411
	Melody estimations	0.1748	0.4607
	Combination	**0.3263**	**0.6583**

Table 6.10: Results for the best combination of features and their individual results.

would suggest), the 12-chord versions instead often stay in a single chord for lengthy periods of time, instead of moving shortly to the second chord; this happens especially with the original version. The main problem, however, lies in the repetitive nature of the chord sequences of the pieces that makes them highly compressable with various unrelated pieces of music. For the already highly compressable 12-chord sequence representations, this is an even more notable phenomenon.

For a human listener the pieces of music in query set ID 30 are nearly trivial to identify; in addition to the above-mentioned power chord riff, each of the pieces contains a distinctive, repetitive melodic pattern in the so-called verse section of the piece, and in combination with the riff, these make the different versions of the piece stand out from most of the material included in the complete dataset. However, apparently this melody is either not captured in the representations or it is too easily confused with other melodic representations of the dataset. Also, again the repetitive nature causes problems with the normalized compression distance, as the highly repetitive sequences compress very efficiently. The problem here thus lies in that both the harmonic and the melodic content is repetitive, and the combination does not provide any more distinguishing power. Also, some bias is likely caused by the prominent variance of the lengths of the pieces in the query set; the longest version of the pieces is nearly four times the length of the shortest version (which is also the original version). However, several of the versions are quite similar to the original in many aspects (such as tempo, structure, and key), but they are nevertheless considered different.

As stated, the set ID 30 is equally difficult for the state-of-the-art algo-

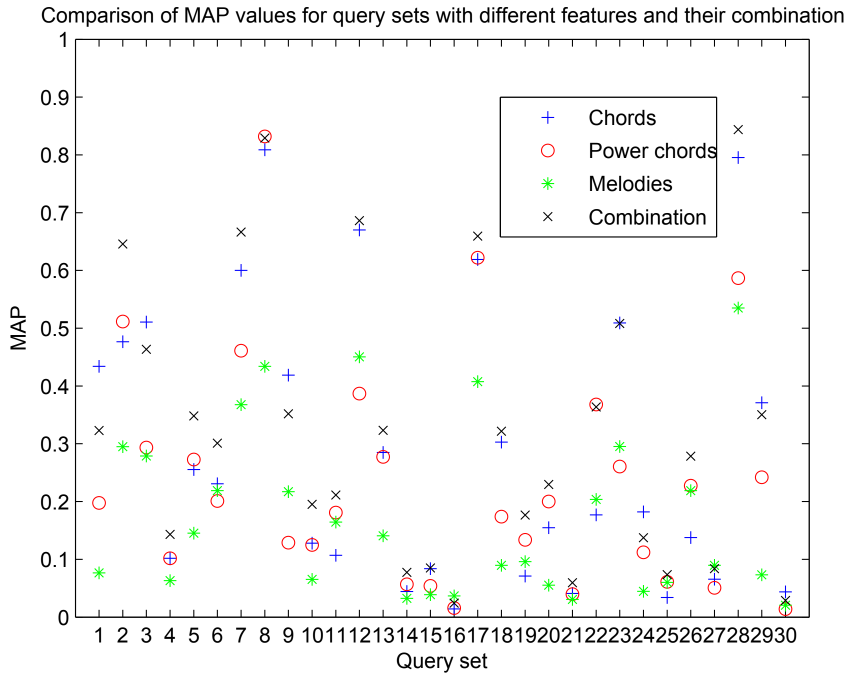

Figure 6.4: MAP values for each 30 different query sets, calculated individually for each feature and with the mean distance value combination strategy.

rithm (see Subsection 5.3.2). Apparently, the information contained in the chromagrams for these pieces of music is not adequate for successful identification. The pieces in query set ID 30 suggest that some novel features (and/or similarity measuring techniques) should be introduced in order to successfully capture the distinctive common features of the pieces in the set; however, it is unclear what these features might be, and perhaps more interestingly, whether these features could be beneficial with any other cover song queries. In any case, it is a highly interesting notion that a piece of music that can be easily identified by a human listener because its characteristics is nearly impossible to distinguish for an algorithm due to the very same characteristics.

6.5 Summary of the Chapter

The feature combination proved more difficult than it might have appeared. Combining features into single representations did not work very well, neither did a concatenated representation of basic representations. However, combination of computed distances provided better identification accuracies than using only single features. This is not unprecedented, as the mixture-of-experts strategy has been widely used in machine learning. Perhaps a slightly more surprising discovery was the notion that the best results were obtained by taking the mean of the distances, instead of more sophisticated methods. However, even the most sophisticated combinations we have used so far could not provide much help with some of the most challenging query sets in our data set.

As already mentioned, adding more features results in a tradeoff between identification accuracy and computational costs. Thus, the suitability of feature combination is left to the purpose of the practical application of the system. Also, as the identification improvement obtained via feature combination seems to be dependent on the data where it is applied, the practicality of combination is highly a matter of the implementation area. In cases where the focus is on accurate detection, the combination strategy is likely worth the growth in the computational time, whereas identification cases with very large amounts of data and/or lack of computational time are similarly likely hardly suitable for the feature combination.

A best of both worlds approach might be pipelining the features or distances. This means first filtering the possible candidates from a larger set by using a representation or distance metric that is likely to filter out the highly dissimilar pieces from the set, and then the more time-consuming higher definition similarity measuring could be carried out to a smaller subset of the pieces. This should provide a higher identification accuracy than a single-feature approach while still maintaining a reasonable computation cost.

Chapter 7

Conclusions

Chromagram data is a highly practical mid-level source of tonal information for various tasks of content-based music information retrieval. The question of similarity measuring between two chromagram sequences has been studied for different purposes, but one of the most interesting – and simultaneously most difficult – problems of chromagram similarity is the task of cover song identification. With different kinds of potential real-world application areas, successful cover song identification can lead to highly useful music information retrieval, but it can also provide interesting results in music-related research. The results of cover song identification can provide additional information on the unsolved question of what actually makes two compositions similar. The task of identifying a piece of music as a cover version is rather trivial for a human listener; however, this is all but true for an algorithm.

In this research we have studied how similarity between chromagrams can be measured using the compressibility of the data to define the distance between two chromagram sequences. We applied a methodology known as normalized compression distance, where the similarity between two objects is determined by measuring their mutual information via data compression. In short, when two objects contain similar information, we should be able to compress the second more efficiently given the information we have learned from the first, and the more the similar information is present, the more efficient the compression should be. The analogy here is evident; if we can learn the essential compositional features from a piece of music, we should be able to use these features to describe a cover version of the composition, in spite of the features that can be deemed unimportant in this regard (such as tempo, key, structure, arrangements, the language of the lyrics, and so on).

In order to compress the continuous chromagram data efficiently, we had to find a suitable quantization method to provide a representation that both contains essential tonal information of the piece but at the same time is still not too complex to be compressed with a real-world data compression algorithm. Ultimately, the best tradeoff between representational accuracy and demands of compression-based similarity measuring was obtained by training a hidden Markov model with the chromagram data and calculating a Viterbi path that provides an estimation of the chord changes of the piece, reduced to a set of major and minor triad chords of each twelve root note pitch classes [19].

Such representation is naturally quite limiting. Several pieces of music contain similar chord changes and sequences, even though those pieces might otherwise be highly dissimilar. Extending the representation did not provide a solution for the task, but instead, we noticed that combining several distances between features can have a positive effect on the identification accuracy. After including several features we came to the conclusion that at least for the data in our hands the best way to combine these distances is to take the mean value of the feature distances as the ultimate distance between the pieces of music.

As a whole, the performance of the NCD-based cover song identification system was relatively good, considering that our test data appears to be rather difficult. Still, the proposed system did not achieve the identification level of a state-of-the-art system. Observations suggest that this is likely not due to the features and representations used alone, but neither the similarity measure itself alone. Both have shortcomings, and for a dependable identification system, they should be throughly addressed.

7.1 Contributions

We proposed several questions in the introductory chapter of this thesis that formed the basis of the research work conducted here (see Subsection 1.3). After the studies, experiments, and analysis, we can now provide answers to these questions.

- Normalized compression distance can be effectively applied to chromagram similarity measuring, and more precisely, to the task of cover song identification. We proved this with a set of experiments and obtained results that fall somewhat behind the state of the art, but we are also quite positive that the optimal performance level of compression-based similarity measuring of chromagram data has not yet been reached. The results are mostly in line with our previous work [8, 4, 5, 9, 7], although as we conducted our experiments with a far larger and more difficult dataset here, the identification performance became lower.

- We discovered that quantization of the continuous features is most efficiently carried out with a hidden Markov model-based chord estimation. Although such mid-level representation is likely to be biased on the harmonic content of the two pieces of music, it still is both capable of expressing essential characteristics of the piece, while maintaining a reasonable level of compressibility that is likely unreached with sequences of a more complex alphabet. The representation causes two notable challenges. First, similar harmonic progressions can be found in pieces of music that are otherwise unrelated. Second, songs with very trivial and monotonous harmonic progressions are likely to cause difficulties in similarity measuring. Both of these problems can be to some extent overcome by using the chord representations in combination with additional features.

- After discovering that the chord estimation sequences are the most suitable quantized representation for the chromagram data, we studied the various parameters with relation to the given representation. We discovered several interesting notions on the length of the chromagram window used in extraction and on the representation of such sequences. We came to the conclusion that one of the key challenges for applying NCD to this task is the fact that the sequences are rather short. Borrowing ideas from the time series research technique known as embedding, we discovered that the compression-based identification for short sequences can be emphasized by extending the data length by duplicating moving window subsequences of the data. In addition, several methods of filtering the chromagram data and the obtained chord estimation sequences and their effect on identification was studied. Some of the observations made here differ from our previous work: whereas in [9] median filtering chromagram data and sequences were a useful addition, such discovery holds here only for the sequences.

- In order to apply normalized compression distance for tasks of content-based music information retrieval, one needs to focus on the data representation. This seems to be even more crucial than several other choices, such as the compression algorithm itself. As the literary review suggests, no standard representations for music features to be used exist. Our work might have shed some light on how the chromagram data can be represented for compression, but at the same time we must acknowledge that even the best quantizations lose information that needs to be compensated with additional features.

All in all, we have made an extensive overview on the task, with additional remarks based on the very fundamental challenges of the task known as cover song identification. We have also made several suggestions that have not provided the desired results, but we have nevertheless reported them here, in order to at least save future researchers interested in the topic the trouble.

7.2 Discussion

The purpose of this thesis was to provide insight into whether similarity measuring based on data compression could be efficiently applied for cover song identification. Initially, the results obtained from the very basic tests did not provide a high potential for success; although the level was already above a random baseline result, the identification accuracy could at best be described as modest. To explain such a low identification accuracy, we observed the results and made several suggestions on how the identification accuracy could be improved.

One of the most apparent drawbacks was the short lengths of the chromagram sequences. Even with a very short extraction window, the length of a typical three-minute piece of popular music would result in a sequence of only some hundreds of frames; clearly too short a sequence for a data compression algorithm that usually require a healthy amount of data in order to efficiently learn a model. Also, extending the length of the sequences with a shorter extraction window proved to have a negative effect on the identification accuracy; the chroma sequences simply became too noisy, with too short time frames to actually present musical features on a larger scale.

As the short length of the sequences is thus dictated, several other challenges ensue. Naturally, the entropy of a shorter sequence can be expected to be higher when the amount of different symbols in the string increases. Thus, in order to actually be able to compress a sequence, the sequence should be constructed from a rather small alphabet. And the smaller the alphabet, the less distinguishing power it is likely to contain; the chord sequence estimations, for example, are based on a lexicon of only 24 different triad chords. Trivially, this is too limiting to efficiently represent various musical characteristics. We solved this problem by the means of feature combination. The strategy of combining compression-based similarity values for different features did indeed have a positive effect on the identification, but at the same time, this came with a tradeoff of more computational resources needed. In practice this might still be applicable for several tasks; the real-world compression algorithms are often highly efficiently implemented, and the compression-based similarity measuring can be carried out in a less time-consuming manner than with a more sophisticated, task-wise similarity measuring algorithms.

Even with additional preprocessing, sequence duplication, and feature combination, several cases in our test data proved to be nearly impossible for the data compression algorithm to detect similarity, resulting into very low mean of average precision values for the particular query sets. Apparently, no suitable features could be extracted from these pieces of music in order to distinguish them from the other pieces in the dataset. However, based on the experiments conducted with our implementation of the state-of-the-art cover song identification algorithm, it seems that some of these difficult cases are more or less as difficult even with an overall more accurate identification system. However, the pieces contained in these most difficult sets are still quite easily distinguished by a human listener, raising a further question of how well the cover song identification algorithms are even able to perform and whether a glass ceiling on the accuracy exists, especially when the amount of data increases. This is a question beyond the scope of this thesis, but it is something that should be considered when studying automatic methodologies for cover song identification.

In comparison to the state of the art, we can denote that we have not reached its identification accuracy. We do not address NCD solely as the cause of our weaker results; the calculation of the similarity in [104] is based on a rather straightforward dynamic programming method with a binary similarity matrix. Using the dynamic programming similarity measuring on the sequences we used as features, we got results higher than the NCD-based distance measuring, suggesting that there are also shortcomings with NCD for this task. Still, the results were not significantly better than with NCD.

Based on the results, we denote compression-based cover song identification to be an interesting alternative for the task of chromagram similarity measuring. The robust similarity detecting nature of data compression, the quasi-universality of the normalized compression distance and its parameter-free simplicity, and the computational efficiency of standard data compression algorithms are all definitely advantages in the task of chromagram similarity measuring.

7.3 Future Work

Despite all the work presented here, it is still only a portion of compresion-based cover song identification. There is plenty of work that still awaits to be done, some of which we have conducted in small measures, while some are just distant ideas waiting to be taken into processing.

So far, we have used empirical discovering for selecting various parameters of our identification system. Although we have used a vast amount of real-world music data for our experiments and evaluations, there still is a high risk of overfitting the parameters to the data in question. For a more sophisticated solution it would be preferable to make several of the parameters used adaptive.

As stated in Section 6.5, the work of feature combination can still be extended with a pipelining strategy applied in order to reduce computational cost while maintaining a higher identification accuracy. In addition, we can also assume that different novel features could still be included into the process. So far, we have not explicitly used any structural information or other larger-scale features of the pieces.

As the quantization of the continuous chromagram data has been one of the major challenges in the process, it is a tempting idea to overcome this part completely and use compression-based similarity with continuous data. Even though compressing continuous values with a standard data compression algorithm is directly an unsuitable solution, the idea could be extended to algorithms purposely-built for continuous data. Several ideas exist; in [64] an approach of the Lempel-Ziv-based compression scheme for continuous data is presented, and the method is proven to work well with short time series. Also, in [42] an idea of continuous NCD is presented, in addition to a variation of NCD using alignment of sequences instead of crude concatenation for the estimation of $K(x|y)$. We have already produced some proportional research in this area.

Recently, estimating the predictability of music has been proposed as a cover song identification strategy [42]. The motivation here lies in the idea that a model learned from music can be applied to predict a new piece of music; when a piece of music is a cover, the prediction should be more precise than with unrelated pieces. Naturally, this is highly compatible with our compression-based scheme, as the compression algorithm indeed learns a model from the music. Using this model and a piece of music, we can predict a new sequence of chroma frames or quantized symbols, and then calculate the distance between the prediction and the real piece of music.

The task of cover song identification is far from being solved, and the compression-based approach to it still has plenty of interesting challenges and open questions. The insight provided by this thesis should be applicable as a starting point for future explorations in the world of cover song identification and musical similarity.

References

[1] *Ultimate Jazz Showstoppers*. Warner Bros. Publications, 2002.

[2] *The Beatles Easy Fake Book*. Hal Leonard, 2009.

[3] Abdel L. Abu Dalhoum, Manuel Alfonseca, Manuel Cebrián, Rafael Sanchez-Alfonso, and Alfonso Ortega. Computer-Generated Music Using Grammatical Evolution. In *Proceedings of the Middle-East Simulation Multiconference (MESM '08)*, 2008.

[4] Teppo E. Ahonen. Measuring Harmonic Similarity Using PPM-Based Compression Distance. In *Proceedings of the Workshop on Exploring Musical Information Spaces (WEMIS 2009)*, 2009.

[5] Teppo E. Ahonen. Combining Chroma Features for Cover Version Identification. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

[6] Teppo E. Ahonen. Compressing Lists for Audio Classification. In *Proceedings of the 3rd International Workshop on Machine Learning and Music (MML '10)*, 2010.

[7] Teppo E. Ahonen. Compression-Based Clustering of Chromagram Data: New Method and Representations. In *Proceedings of the 9th International Symposium on Computer Music Modeling and Retrieval (CMMR 2012)*, 2012.

[8] Teppo E. Ahonen and Kjell Lemström. Identifying Cover Songs Using Normalized Compression Distance. In *Proceedings of the 1st International Workshop on Machine Learning and Music (MML '08)*, 2008.

[9] Teppo E. Ahonen, Kjell Lemström, and Simo Linkola. Compression-Based Similarity Measures in Symbolic, Polyphonic Music. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011.

[10] Manuel Alfonseca, Manuel Cebrián, and Alfonso Ortega. Evolving Computer-Generated Music by Means of the Normalized Compression Distance. In *Proceedings of the 5th WSEAS International Conference on Simulation, Modelling and Optimization (SMO '05)*, 2005.

[11] Manuel Alfonseca, Manuel Cebrián, and Alfonso Ortega. A Simple Genetic Algorithm for Music Generation by Means of Algorithmic Information Theory. In *Proceedings of the IEEE Congress on Evolutionary Computation 2007 (CEC 2007)*, 2007.

[12] Yoko Anan, Kohei Hatano, Hideo Bannai, Masayuki Takeda, and Ken Satoh. Polyphonic Music Classification on Symbolic Data Using Dissimilarity Functions. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, 2012.

[13] Javed A. Aslam, Emine Yilmaz, and Virgiliu Pavlu. The Maximum Entropy Method for Analyzing Retrieval Measures. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*, 2005.

[14] Ricardo Baeza-Yates and Berthier Ribiero-Neto. *Modern Information Retrieval*. ACM Press, 1999.

[15] Mark A. Bartsch and Gregory H. Wakefield. To Catch a Chorus: Using Chroma-Based Representations for Audio Thumbnailing. In *Proceedings of the 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2001)*, 2001.

[16] Juan P. Bello. Audio-Based Cover Song Retrieval Using Approximate Chord Sequences: Testing Shifts, Gaps, Swaps and Beats. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, 2007.

[17] Juan P. Bello. Grouping Recorded Music by Structural Similarity. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, 2009.

[18] Juan P. Bello. Measuring Structural Similarity in Music. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2013—2025, September 2011.

[19] Juan P. Bello and Jeremy Pickens. A Robust Mid-Level Representation for Harmonic Content in Music Signals. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, 2005.

[20] Thierry Bertin-Mahieux and Daniel P. W. Ellis. Large-scale Cover Song Recognition Using Hashed Chroma Landmarks. In *Proceedings of the 2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2011)*, 2011.

[21] Thierry Bertin-Mahieux and Daniel P. W. Ellis. Large-scale Cover Song Recognition Using the 2D Fourier Transform Magnitude. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, 2012.

[22] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011.

[23] Thierry Bertin-Mahieux, Ron J. Weiss, and Daniel P. W. Ellis. Clustering Beat-Chroma Patterns in a Large Music Database. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

[24] Judith C. Brown. Calculation of a Constant Q Spectral Transform. *Journal of the Acoustical Society of America*, 89(1):425—434, January 1991.

[25] Judith C. Brown and Miller S. Puckette. An Efficient Algorithm for the Calculation of a Constant Q Transform. *Journal of the Acoustical Society of America*, 92(5):2698—2701, November 1992.

[26] Michael Burrows and David J. Wheeler. A Block-sorting Lossless Data Compression Algorithm. Technical Report 124, Digital Systems Research Center, May 1994.

[27] Donald Byrd and Tim Crawford. Problems of Music Information Retrieval in the Real World. *Information Processing and Management*, 38(2):249—272, March 2002.

[28] Michael Casey and Malcolm Slaney. The Importance of Sequences in Musical Similarity. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, 2006.

[29] Zehra Cataltepe, Yusuf Yaslan, and Abdullah Somnez. Music Genre Classification Using MIDI and Audio Features. *EURASIP Journal on Applied Signal Processing*, 2007(1):150—157, January 2007.

[30] Manuel Cebrián, Manuel Alfonseca, and Alfonso Ortega. Common Pitfalls Using the Normalized Compression Distance: What to Watch out for in a Compressor. *Communications in Information and Systems*, 5(4):367—384, 2005.

[31] Manuel Cebrián, Manuel Alfonseca, and Alfonso Ortega. The Normalized Compression Distance Is Resistant to Noise. *IEEE Transactions on Information Theory*, 53(5):1895—1900, May 2007.

[32] Xin Chen, Brent Francia, Ming Li, Brian McKinnon, and Amit Seker. Shared Information and Program Plagiarism Detection. *IEEE Transactions on Information Theory*, 50(7):1545—1551, July 2004.

[33] Rudi Cilibrasi and Paul M. B. Vitányi. Clustering by Compression. *IEEE Transactions on Information Theory*, 51(4):1523—1545, April 2005.

[34] Rudi Cilibrasi, Paul M. B. Vitányi, and Ronald de Wolf. Algorithmic Clustering of Music Based on String Compression. *Computer Music Journal*, 28(4):49—67, 2004.

[35] John G. Cleary and Ian H. Witten. Data Compression Using Adaptive Coding and Partial String Matching. *IEEE Transactions on Communications*, 32(4):396—402, April 1984.

[36] J. Stephen Downie. Music Information Retrieval. *Annual Review of Information Science and Technology*, 37(1):295—340, 2003.

[37] J. Stephen Downie, Mert Bay, Andreas F. Ehmann, and M. Cameron Jones. Audio Cover Song Identification: MIREX 2006–2007 Results and Analyses. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, 2008.

[38] J. Stephen Downie, Donald Byrd, and Tim Crawford. Ten Years of ISMIR: Reflections on Challenges and Opportunities. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, 2009.

[39] Daniel P. W. Ellis and Courtenay V. Cotton. The 2007 LabROSA Cover Song Detection System. In *Proceedings of the Music Information Retrieval Evaluation eXchange 2007 (MIREX 2007)*, 2007.

[40] Daniel P. W. Ellis and Graham E. Poliner. Identifying 'Cover Songs' with Chroma Features and Dynamic Programming Beat Tracking. In

Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007), 2007.

[41] Jonathan Foote. ARTHUR: Retrieving Orchestral Music by Long-Term Structure. In *Proceedings of the 1st International Conference on Music Information Retrieval (ISMIR 2000)*, 2000.

[42] Peter Foster, Simon Dixon, and Anssi Klapuri. Identification of Cover Songs Using Information Theoretic Measures of Similarity. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, 2013.

[43] Takuya Fujishima. Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music. In *Proceedings of the 1999 International Computer Music Conference (ICMC 1999)*, 1999.

[44] Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1991.

[45] Asif Ghias, Jonathan Logan, David Chamberlin, and Brian C. Smith. Query by Humming: Musical Information Retrieval in an Audio Database. In *Proceedings of the 3rd ACM International Conference on Multimedia (ACM Multimedia '95)*, 1995.

[46] Emilia Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, 2006.

[47] Emilia Gómez. Tonal Description of Polyphonic Audio for Music Content Processing. *INFORMS Journal on Computing*, 18(3):294—304, 2006.

[48] Antonio González-Pardo, Ana Granados, David Camacho, and Francisco de Borja Rodriguez. Influence of Music Representation on Compression-Based Clustering. In *Proceedings of the IEEE Congress on Evolutionary Computation 2010 (CEC 2010)*, 2010.

[49] Ana Granados, Manuel Cebrián, David Camacho, and Francisco de Borja Rodriguez. Evaluating the Impact of Information Distortion on Normalized Compression Distance. In *Proceedings of the 2nd International Castle Meeting (ICMCTA 2008)*, 2008.

[50] Christopher Harte, Mark Sandler, and Martin Gasser. Detecting Harmonic Change in Musical Audio. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia (AMCMM '06)*, 2006.

[51] Ning Hu, Roger B. Dannenberg, and George Tzanetakis. Polyphonic Audio Matching and Alignment for Music Retrieval. In *Proceedings of the 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2003)*, 2003.

[52] Eric J. Humphrey, Oriol Nieto, and Juan P. Bello. Data Driven and Discriminative Projections for Large-Scale Cover Song Identification. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, 2013.

[53] Takahito Inoshita and Jiro Katto. Key Estimation Using Circle of Fifths. In *Proceedings of the 15th International Multimedia Modeling Conference on Advances in Multimedia Modeling (MMM '09)*, 2009.

[54] Özgür Izmirli. Template Based Key Finding from Audio. In *Proceedings of the 2005 International Computer Music Conference (ICMC 2005)*, 2005.

[55] Özgür Izmirli. Tonal Similarity from Audio Using a Template Based Attractor Model. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, 2005.

[56] Michael Kassler. Toward Musical Information Retrieval. *Perspectives of New Music*, 4(2):59—67, 1966.

[57] Eamonn Keogh, Stefano Lonardi, and Chotirat A. Ratanamahatana. Towards Parameter-Free Data Mining. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, 2004.

[58] Maksim Khadkevich and Maurizio Omologo. Large-Scale Cover Song Identification Using Chord Profiles. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, 2013.

[59] Samuel Kim and Shrikanth Narayanan. Dynamic Chroma Feature Vectors with Applications to Cover Song Identification. In *Proceedings of the 10th IEEE International Workshop on Multimedia Signal Processing (MMSP '08)*, 2008.

[60] Samuel Kim, Erdem Unal, and Shrikanth Narayanan. Music Fingerprint Extraction for Classical Music Cover Song Identification. In *Proceedings of the 2008 IEEE International Conference on Multimedia and Expo (ICME 2008)*, 2008.

[61] Youngmoo E. Kim and Daniel Perelstein. MIREX 2007: Audio Cover Song Detection Using Chroma Features and a Hidden Markov Model. In *Proceedings of the Music Information Retrieval Evaluation eXchange 2007 (MIREX 2007)*, 2007.

[62] Anssi Klapuri and Manuel Davy, editors. *Signal Processing Methods for Music Transcription*. Springer, 2006.

[63] Antti Laaksonen. Orchestration Assumptions in Automatic Melody Extraction. Unpublished manuscript.

[64] Willis Lang, Michael Morse, and Jignesh M. Patel. Dictionary-Based Compression for Long Time-Series Similarity. *IEEE Transactions on Knowledge and Data Engineering*, 22(11):1609—1622, November 2010.

[65] Olivier Lartillot. MIRTempo: Tempo Estimation Through Advanced Frame-by-Frame Peaks Tracking. In *Proceedings of the Music Information Retrieval Evaluation eXchange 2010 (MIREX 2010)*, 2010.

[66] Olivier Lartillot and Petri Toiviainen. MIR in Matlab (II): A Toolbox for Musical Feature Extraction from Audio. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, 2007.

[67] Olivier Lartillot, Petri Toiviainen, and Tuomas Eerola. A Matlab Toolbox for Music Information Retrieval. In *Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation e.V. (GfKl 2007)*, 2007.

[68] Kyogu Lee. Identifying Cover Songs from Audio Using Harmonic Representation. In *Proceedings of the Music Information Retrieval Evaluation eXchange 2006 (MIREX 2006)*, 2006.

[69] Kyogu Lee and Malcolm Slaney. A Unified System for Chord Transcription and Key Extraction Using Hidden Markov Models. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, 2007.

[70] Kjell Lemström and Geraint A. Wiggins. Formalizing Invariances for Content-Based Music Retrieval. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, 2009.

[71] Ming Li, Xin Chen, Xin Li, Bin Ma, and Paul M. B. Vitányi. The Similarity Metric. *IEEE Transactions on Information Theory*, 50(12):3250—3264, December 2004.

[72] Ming Li and Ronan Sleep. Melody Classification Using a Similarity Metric Based on Kolmogorov Complexity. In *Proceedings of the 2004 Sound and Music Computing Conference (SMC'04)*, 2004.

[73] Ming Li and Ronan Sleep. Genre Classification via an LZ78-Based String Kernel. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, 2005.

[74] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, third edition, 2008.

[75] Cynthia C. S. Liem and Alan Hanjalic. Cover Song Retrieval: A Comparative Study of System Component Choices. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, 2009.

[76] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[77] Matija Marolt. A Mid-level Melody-based Representation for Calculating Audio Similarity. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, 2006.

[78] Matija Marolt. A Mid-Level Representation for Melody-Based Retrieval in Audio Collections. *IEEE Transactions on Multimedia*, 10(8):1617—1625, December 2008.

[79] Benjamin Martin, Daniel G. Brown, Pierre Hanna, and Pascal Ferraro. BLAST for Audio Sequences Alignment: A Fast Scalable Cover Identification Tool. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, 2012.

[80] Matthias Mauch and Simon Dixon. Approximate Note Transcription for the Improved Identification of Difficult Chords. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

[81] Cory McKay and Ichiro Fujinaga. Automatic Genre Classification Using Large High-Level Musical Feature Sets. In *Proceedings of the*

5th International Conference on Music Information Retrieval (ISMIR 2004), 2004.

[82] Bassam Mokbel, Alexander Hasenfuss, and Barbara Hammer. Graph-Based Representation of Symbolic Musical Data. In *Proceedings of the 7th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition (GbRPR 2009)*. Springer, 2009.

[83] Meinard Müller. *Information Retrieval for Music and Motion*. Springer Verlag, 2007.

[84] Meinard Müller, Sebastian Ewert, and Sebastian Kreuzer. Making Chroma Features More Robust to Timbre Changes. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, 2009.

[85] Meinard Müller, Frank Kurth, and Michael Clausen. Audio Matching via Chroma-Based Statistical Features. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, 2005.

[86] Hidehisa Nagano, Kunio Kashino, and Hiroshi Murase. Fast Music Retrieval Using Polyphonic Binary Feature Vectors. In *Proceedings of the 2002 IEEE International Conference on Multimedia and Expo (ICME 2002)*, 2002.

[87] Katy Noland and Mark Sandler. Key Estimation Using a Hidden Markov Model. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, 2006.

[88] Nicola Orio. Music Retrieval: A Tutorial and Review. *Foundations and Trends in Information Retrieval*, 1(1):1–96, 2006.

[89] Hélène Papadopoulos and Geoffroy Peeters. Large-Scale Study of Chord Estimation Algorithms Based on Chroma Representation and HMM. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing 2007 (CBMI '07)*, 2007.

[90] Denys Parsons. *The Directory of Tunes and Musical Themes*. S. Brown, 1975.

[91] Geoffroy Peeters. Musical Key Estimation of Audio Signal Based on Hidden Markov Modeling of Chroma Vectors. In *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, 2006.

[92] Carlos Pérez-Sancho, David Rizo, Stefan Kersten, and Rafael Ramirez. Genre Classification of Music by Tonal Harmony. In *Proceedings of the 1st International Workshop on Machine Learning and Music (MML '08)*, 2008.

[93] Lawrence R. Rabiner. A Tutorial on HMM and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257—286, 1989.

[94] Suman Ravuri and Daniel P. W. Ellis. The Hydra System Of Unstructured Cover Song Detection. In *Proceedings of the Music Information Retrieval Evaluation eXchange 2009 (MIREX 2009)*, 2009.

[95] Matthew Riley, Eric Heinen, and Joydeep Ghosh. A Text Retrieval Approach To Content-Based Audio Retrieval. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, 2008.

[96] David Rizo. *Symbolic Music Comparison with Tree Data Structures*. PhD thesis, Universidad de Alicante, 2010.

[97] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition, 2002.

[98] Justin Salomon, Emilia Gómez, Daniel P. W. Ellis, and Guilhem Richard. Melody Extraction from Polyphonic Music Signals: Approaches, Applications, and Challenges. *IEEE Signal Processing Magazine*, 31(2):118—134, 2014.

[99] Justin Salomon, Joan Serrà, and Emilia Gómez. Melody, Bass Line, and Harmony Representations for Music Version Identification. In *Proceedings of the 4th International Workshop on Advances in Music Information Research (AdMIRe 2012)*, 2012.

[100] D. Sculley and Carla E. Brodley. Compression and Machine Learning: A New Perspective on Feature Space Vectors. In *Proceedings of the Data Compression Conference (DCC '06)*, 2006.

[101] Joan Serrà, Emilia Gómez, and Perfecto Herrera. Transposing Chroma Representations to a Common Key. In *Proceedings of the IEEE CS Conference on The Use of Symbols to Represent Music and Multimedia Objects*, 2008.

[102] Joan Serrà, Emilia Gómez, and Perfecto Herrera. *Audio Cover Song Identification and Similarity: Background, Approaches, Evaluation, and Beyond*, volume 274 of *Studies in Computational Intelligence*, chapter 14, pages 307—332. Springer-Verlag Berlin / Heidelberg, 2010.

[103] Joan Serrà, Emilia Gómez, Perfecto Herrera, and Xavier Serra. Chroma Binary Similarity and Local Alignment Applied to Cover Song Identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6):1138—1151, August 2008.

[104] Joan Serrà, Xavier Serra, and Ralph G. Andrzejak. Cross Recurrence Quantification for Cover Song Identification. *New Journal of Physics*, 11, 2009.

[105] Joan Serrà, Massimiliano Zanin, and Ralph G. Andrzejak. Cover Song Retrieval by Cross Recurrence Quantification and Unsupervised Set Detection. In *Proceedings of the Music Information Retrieval Evaluation eXchange 2009 (MIREX 2009)*, 2009.

[106] Alexander Sheh and Daniel P. W. Ellis. Chord Segmentation and Recognition Using EM-Trained Hidden Markov Models. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, 2003.

[107] Umut Simsekli. Automatic Music Genre Classification Using Bass Lines. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR 2010)*, 2010.

[108] Temple F. Smith and Michael S. Waterman. Identification of Common Molecular Subsequences. *Journal of Molecular Biology*, 147(1):195—197, March 1981.

[109] Bob L. Sturm. An Analysis of the GTZAN Music Genre Dataset. In *Proceedings of the 2nd International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies (MIRUM '12)*, 2012.

[110] Bob L. Sturm. The GTZAN Dataset: Its Contents, Its Faults, Their Affects on Evaluation, and Its Future Use. *arXiv preprint arXiv:1306.1461*, 2013.

[111] Wei-Ho Tsai, Hung-Ming Yu, and Hsin-Min Wang. Using the Similarity of Main Melodies to Identify Cover Versions of Popular Songs

for Music Document Retrieval. *Journal of Information Science and Engineering*, 24(6):1669—1687, 2008.

[112] George Tzanetakis and Perry Cook. Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293—302, July 2002.

[113] Steven van de Par, Martin F. McKinney, and André Redert. Musical Key Extraction from Audio Using Profile Training. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, 2006.

[114] Gregory H. Wakefield. Mathematical Representation of Joint Time-Chroma Distribution. In *Proceedings of the SPIE Conference on Advanced Signal Processing Algorithms, Architectures, and Implementations IX*, 1999.

[115] Avery Wang. An Industrial Strength Audio Search Algorithm. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, 2003.

[116] Avery Wang. The Shazam Music Recognition Service. *Communications of the ACM*, 49(8):44—48, August 2006.

[117] Cheng Yang. Music Database Retrieval Based on Spectral Similarity. Technical Report 2001-14, Stanford InfoLab, 2001.

[118] Jacob Ziv. Coding Theorems for Individual Sequences. *IEEE Transactions on Information Theory*, 24(4):405—412, July 1978.

[119] Jacob Ziv and Abraham Lempel. A Universal Algorithm for Sequential Data Compression. *IEEE Transactions on Information Theory*, 23(3):337—343, May 1977.

Chapter A

Yesterday Dataset

The Yesterday dataset consists of 41 different variations of the song Yesterday. The composition is credited to John Lennon and Paul McCartney, and was first published in 1965. The content is listed in the table below. The table lists the performer of the version and the ID used as a reference throughout the experiments of Chapter 4. Also, for each piece the length of the pieces is indicated both in real time (mm:ss) and the length of the chroma sequences extracted with the window of 16384 samples, and the keys estimated by the MIRToolbox algorithm, as well as the OTI differences to the original piece, are given. The column "Title" displays the title of the published recording in case it is not Yesterday. The version with song ID 37 is taken from album The Panpipes Collection, and we are unaware of the name of the performer.

Song ID	Performer	Time	Frames	Key	OTI	Title
1	The Beatles	2:05	337	F major	0	
2	Markku Aro	2:44	440	F major	0	Eilinen
3	The Bar-Keys	3:22	544	D minor	0	
4	Count Basie	3:20	537	G♯ major	9	
5	Andrea Benzoni	3:12	518	G major	10	
6	Cathy Berberian	1:53	305	E major	1	
7	Cilla Black	2:27	396	A major	8	
8	Ray Charles	2:46	448	F major	0	
9	Cincinnati Pops Orchestra	3:35	578	F major	0	
10	Richard Clayderman	2:25	390	F major	0	
11	Perry Como	3:01	489	A minor	5	
12	Neil Diamond	3:31	569	B♭ major	7	
13	Placido Domingo	2:48	452	F major	0	
14	Marianne Faithfull	2:18	372	A major	8	
15	Chris Farlowe	2:29	400	F major	0	
16	The Flame All Stars	3:21	540	D minor	0	
17	Marvin Gaye	3:26	553	C major	5	
18	Jukka Gustavson	3:52	624	D minor	0	
19	Franz Hal‘asz	3:05	498	F♯ minor	8	
20	Dr. John	5:20	863	C major	5	
21	Linda Jones	2:32	409	G♯ major	9	
22	Tom Jones	2:56	474	F major	0	
23	Jormas	1:23	223	F major	0	
24	The King's Singers	2:34	416	D major	3	
25	Liberace	2:17	369	F major	0	
26	Max'C	3:19	535	F major	0	
27	The Modern Jazz Quartet	4:07	664	E♭ major	7	
28	Matt Monro	2:48	453	C major	5	
29	David Newman	4:03	655	F major	0	
30	Laura Närhi	2:16	366	G major	10	Eilinen
31	Poom	2:06	339	G♯ major	9	
32	Elvis Presley	2:27	396	C major	5	
33	LeAnn Rimes	3:10	512	A major	8	
34	The Saexophones	2:24	389	F major	0	
35	A Savage	2:39	427	F major	0	
36	Cyril Stapleton & His Orchestra	2:41	435	E♭ major	2	
37	Unknown	3:14	523	C major	5	
38	Wet Wet Wet	2:55	471	D minor	0	
39	Joe White	3:17	530	F major	0	
40	Andy Williams	2:50	457	D major	3	
41	Wings	1:49	294	F major	0	

Table A.1: Content of the Yesterday dataset.

Chapter B

Summertime Dataset

Similarly to the Yesterday dataset, the Summertime dataset consists of 41 variations of jazz standard Summertime. The composition is credited to George Gershwin, whereas the original lyrics are credited to DuBose Heyward. The composition was published in 1935 in the opera Porgy and Bess, and was soon recorded for the first time by Abbie Mitchell in the same year. We use Billie Holiday's version as the canonicical version; it was published in 1936, and was the first recording of the composition to gain commercial attention, appearing at position 12 in the US Pop Charts. The columns of the table below contain similar information to that of Appendix A.

Song ID	Performer	Time	Frames	Key	OTI	Title
1	Billie Holiday	2:54	470	B♭ minor	0	
2	Franco Ambrosetti	6:38	1070	D minor	8	
3	Peter Asplund	6:59	1128	A minor	8	
4	Chet Atkins	4:00	645	B minor	6	
5	Duck Baker	3:34	577	A minor	6	
6	Beat Function	9:04	1464	D minor	8	
7	Sidney Bechet	4:13	681	G minor	3	
8	George Benson	2:25	390	B♭ minor	7	
9	Michael Bolton	4:32	732	A minor	1	
10	Chanticleer	4:09	672	B♭ major	3	
11	Richard Clayderman	2:38	425	A minor	1	
12	Eddie Cochran	2:53	468	A minor	1	
13	John Coltrane	11:36	1874	D major	8	
14	Ray Conniff	2:41	433	B♭ minor	0	
15	Miles Davis	3:18	534	B♭ minor	5	
16	Djavos Heppes	3:18	534	G minor	3	
17	Ella Fitzgerald	2:57	477	B♭ minor	0	
18	Gerry & The Pacemakers	2:30	405	G minor	3	
19	The Go Getters	3:10	510	G major	6	
20	The Harmonie Ensemble NY	4:04	658	A minor	1	
21	Freddie Hubbard	10:08	1639	C♯ minor	2	
22	Johanna	5:30	889	A minor	1	
23	Jamppa Kääriäinen	3:52	623	C minor	10	Kesäyö
24	Barney Kessel	2:13	358	D minor	8	
25	Angelique Kidjo	4:21	703	B minor	11	
26	Laila Kinnunen	4:04	656	C minor	10	Kesäyö
27	Mat Mathews	2:20	377	A minor	1	
28	Gil Melle	4:03	653	A minor	1	
29	Nena	4:02	650	D minor	3	
30	Sonny Rollins	5:58	965	A minor	8	
31	Nina Simone	5:40	916	D minor	8	
32	Jimmy Smith	4:33	735	A minor	8	
33	Topi Sorsakoski	3:50	619	B♭ minor	0	Kesäyö
34	Toru Takemitsu	3:41	595	A minor	1	
35	McCoy Tyner	4:51	782	D minor	8	
36	Sarah Vaughan	3:18	532	F major	0	
37	Caetano Veleso	2:33	411	D minor	8	
38	Mads Vinding	8:12	1323	A minor	1	
39	The Walker Brothers	4:30	726	G minor	3	
40	Dinah Washington	2:27	395	E minor	1	
41	Brian Wilson	3:13	519	A minor	1	

Table B.1: Content of the Summertime dataset.

Chapter C

The Mixed Dataset

The dataset consists of 30 sets of 11 cover versions; for each set, the canonicical version is listed first. In addition, the dataset includes 670 unrelated "noise" pieces, thus totaling 1000 pieces of music and 330 potential queries.

Set 1: All I Have to Do Is Dream

Song ID	Performer	Time	Frames	Key	Title
1	The Everly Brothers	2:20	378	E major	
2	Paul Anka	2:07	342	G major	
3	Markku Aro	2:36	419	B♭ major	Elämäni on kuin suuri haave
4	Glen Campbell	2:35	417	E♭ major	
5	Eini	2:45	444	D major	Haaveissain
6	Barbara Jones	3:44	603	A minor	Dream, Dream, Dream
7	Barry Manilow	2:48	452	E♭ major	
8	Pimpline and the Definites	3:22	543	C major	
9	R.E.M.	2:38	424	E major	
10	Linda Ronstadt	3:31	566	major	
11	Teddy and the Tigers	2:21	381	D major	

Table C.1: Content of the Mixed dataset cover song set 1.

Set 2: Born to Be Wild

Song ID	Performer	Time	Frames	Key	Title
12	Steppenwolf	3:30	566	E minor	
13	Blue Oyster Cult	3:40	593	F♯ minor	
14	The Cult	3:55	632	E minor	
15	The Electric Screwdriver	2:59	482	E minor	
16	Fanfare Ciocărlia	3:11	515	F minor	
17	INXS	3:50	618	E minor	
18	Krokus	3:34	576	A minor	
19	Mass	4:21	703	E minor	
20	The Mooney Suzuki	3:54	629	E minor	
21	Pate Mustajärvi	3:31	568	E minor	Villiksi syntynyt
22	Slade	3:24	550	F minor	

Table C.2: Content of the Mixed dataset cover song set 2.

Set 3: Bridge Over Troubled Water

Song ID	Performer	Time	Frames	Key	Title
23	Simon and Garfunkel	4:55	796	E♭ major	
24	Franco Battiato	3:50	621	E♭ major	
25	Richard Clayderman	3:05	498	E♭ major	
26	Aretha Franklin	5:34	898	B♭ major	
27	Josh Groban	4:40	755	C major	
28	The Jackson 5	5:52	949	D major	
29	Tom Jones	3:03	492	D major	
30	The King's Singers	4:28	721	B major	
31	Markku Laamanen	4:43	761	C major	Silta yli synkän virran
32	Nana Mouskouri	4:17	692	A major	
33	Jessica Pilnäs	3:37	585	E major	

Table C.3: Content of the Mixed dataset cover song set 3.

Set 4: Can't Help Falling in Love

Song ID	Performer	Time	Frames	Key	Title
34	Elvis Presley	3:05	497	F♯ minor	
35	Neil Diamond	3:07	503	D minor	
36	Eels	2:08	343	G major	
37	Frederik	3:02	490	D major	Siellä on maailmain
38	Chris Isaak	3:01	486	D major	
39	Barry Manilow	3:38	588	F minor	
40	Al Martino	2:20	378	G♯ major	
41	Klaus Nomi	3:55	634	E major	
42	Stray Cats	3:22	544	D major	
43	UB40	3:28	561	D major	
44	Andy Williams	1:47	290	F major	

Table C.4: Content of the Mixed dataset cover song set 4.

Set 5: Enjoy the Silence

Song ID	Performer	Time	Frames	Key	Title
45	Depeche Mode	4:14	685	E♭ major	
46	Tori Amos	4:09	672	F major	
47	Ashaw featuring Mary F.	4:07	665	E♭ major	
48	Caater	3:16	528	E♭ major	
49	Gregorian	4:48	775	G major	
50	Janita	4:16	690	C major	
51	Lacuna Coil	4:05	661	D major	
52	Timo Maas	3;54	629	C♯ minor	
53	Nada Surf	3:21	541	F♯ major	
54	Matt Samuels featuring For The Masses	2:40	430	G♯ major	
55	Susanna & The Magical Orchestra	3:44	603	G♯ major	

Table C.5: Content of the Mixed dataset cover song set 5.

Set 6: God Only Knows

Song ID	Performer	Time	Frames	Key	Title
56	The Beach Boys	2:55	471	A major	
57	Clifters	2:49	455	A major	Pirun kaunis nainen
58	Holly Cole	4:27	720	E major	
59	Jormas	2:34	416	A major	Taivas vain tietää
60	Tapani Kansa	2:50	458	D major	Taivas vain tietää
61	The Langley Schools Music Project	3:05	497	G major	
62	The Manhattan Transfer	2:46	446	F major	
63	The Shadows	2:41	432	A major	
64	Luciana Souza	3:52	625	A minor	
65	Andy Williams	2:35	418	F major	
66	The Yellowjackets	5:25	875	E major	

Table C.6: Content of the Mixed dataset cover song set 6.

Set 7: Hallelujah

Song ID	Performer	Time	Frames	Key	Title
67	Leonard Cohen	4:38	748	A minor	
68	Chris Botti	3:00	485	C♯ major	
69	Susan Boyle	3:52	625	F major	
70	Jeff Buckley	6:55	1117	A minor	
71	Alexandra Burke	3:37	585	C minor	
72	Neil Diamond	4:10	675	G major	
73	Katherine Jenkins	4:47	772	B♭ major	
74	k.d. lang	5:08	830	E major	
75	Michael McDonald	5:01	810	B♭ major	
76	Molly Sanden	4:09	671	A major	
77	Amaury Vassili	6:12	1001	A minor	

Table C.7: Content of the Mixed dataset cover song set 7.

Set 8: Hotel California

Song ID	Performer	Time	Frames	Key	Title
78	Eagles	6:30	1050	D major	
79	Creol	5:30	889	D major	
80	Gipsy Kings	5:47	933	D major	
81	Jyrki Härkönen	5:11	838	C major	Yksinäisten kaupunki
82	James Last	5:44	926	D major	
83	Helmut Lotti	5:18	858	A minor	
84	Pat the Cat featuring Rachel Moreau	4:09	671	C major	
85	Rhythms del Mundo & The Killers	6:05	983	D major	
86	Rock Kids	6:03	977	D major	
87	Sly & Robbie	5:59	968	B minor	
88	Wilson Philips	8:52	1432	D major	

Table C.8: Content of the Mixed dataset cover song set 8.

Set 9: I Fought the Law

Song ID	Performer	Time	Frames	Key	Title
89	The Crickets	2:14	361	G major	
90	Bryan Adams	2:37	424	A major	
91	The Clash	2:40	430	D major	
92	The Jolly Boys	3:22	545	G minor	
93	Pelle Miljoona & Rockers	2:31	408	D major	Rikoin lakia
94	Mike Ness	2:49	456	G♯ major	
95	Roy Orbison	2:29	402	C major	
96	The Pogues	2:48	452	D major	
97	She Trinity	2:22	384	A major	He Fought the Law
98	Status Quo	3:07	504	G major	
99	Stray Cats	2:37	424	G major	

Table C.9: Content of the Mixed dataset cover song set 9.

Set 10: I Put a Spell on You

Song ID	Performer	Time	Frames	Key	Title
100	Screamin' Jay Hawkins	2:26	395	B♭ minor	
101	Natacha Atlas	3:42	598	A minor	
102	Jeff Beck featuring Joss Stone	2:59	484	B minor	
103	Joe Cocker	4:31	731	B♭ minor	
104	Creedence Clearwater Revival	4:32	732	?	
105	Demon Fuzz	3:55	632	C minor	
106	Eels	2:21	379	G♯ minor	
107	Buddy Guy featuring Carlos Santana	4:04	657	A minor	
108	Heinäsirkka	4:06	663	A minor	
109	Raney Shockne featuring Eddie Wakes	2:28	398	B major	
110	Nina Simone	2:35	418	F♯ major	

Table C.10: Content of the Mixed dataset cover song set 10. The key estimation algorithm could not determine key for song ID 104.

Set 11: I Walk the Line

Song ID	Performer	Time	Frames	Key	Title
111	Johnny Cash	2:45	443	B♭ major	
112	Rodney Crowell	3:51	621	G major	
113	Dion	3:13	519	A major	
114	The Everly Brothers	2:37	424	G♯ minor	
115	Honey B& T-Bones	4:30	729	A minor	
116	Chris Isaak	2:26	395	B♭ major	
117	Shelby Lynne	2:36	422	E minor	
118	Mad Dog Cole	2:02	330	C minor	Walk the Line
119	Pate Mustajärvi	2:54	469	B♭ major	Kaita tie
120	Leonard Nimoy	2:19	373	F minor	
121	Tapio Rautavaara	3:45	606	E major	Yölinjalla

Table C.11: Content of the Mixed dataset cover song set 11.

Set 12: I Will Always Love You

Song ID	Performer	Time	Frames	Key	Title
122	Dolly Parton	2:55	471	A major	
123	CC & Lee	4:29	725	G major	
124	Richard Clayderman	4:09	670	A major	
125	James Galway	3:22	542	F major	
126	Pentti Hietanen	4:16	690	C major	L'amore Sei Tu
127	Whitney Houston	4:24	711	A major	
128	Katherine Jenkins	4:20	702	A minor	L'amore Sei Tu
129	The King's Singers	4:36	743	E♭ major	
130	Hank Marvin	3:31	569	A major	
131	LeAnn Rimes	4:39	752	E major	
132	Linda Ronstadt	3:01	486	A major	

Table C.12: Content of the Mixed dataset cover song set 12.

Set 13: In the Midnight Hour

Song ID	Performer	Time	Frames	Key	Title
133	Wilson Pickett	2:33	411	E major	
134	The Chocolate Watch Band	4:29	724	A major	
135	The Commitments	2:24	388	G♯ major	
136	Echo & The Bunnymen	3:31	569	C major	
137	Chris Farlowe	2:19	373	A major	
138	Tom Jones	2:04	333	D minor	
139	Martha Reeves	2:19	376	E major	
140	Roxy Music	3:10	512	C major	
141	Voiceboys	3:02	490	G major	
142	The Walker Brothers	2:18	372	C major	
143	The Young Rascals	4:03	655	G major	

Table C.13: Content of the Mixed dataset cover song set 13.

Set 14: Light My Fire

Song ID	Performer	Time	Frames	Key	Title
144	The Doors	7:00	1130	A minor	
145	Brian Auger's Oblivion Express	5:38	911	D major	
146	Shirley Bassey	3:28	559	F minor	
147	David Benoit	4:00	647	A minor	
148	Erma Franklin	2:37	423	G♯ major	
149	Julie London	3:20	537	F major	
150	Nekromantix	3:15	524	A minor	
151	Minnie Riperton featuring Jose Feliciano	5:05	821	G major	
152	This Was	4:25	712	D major	
153	Train	3:43	602	G major	
154	Charles Wright & The Watts 103rd Street Rhythm Band	3:41	596	A major	

Table C.14: Content of the Mixed dataset cover song set 14.

Set 15: Mr. Tambourine Man

Song ID	Performer	Time	Frames	Key	Title
155	Bob Dylan	5:26	877	F major	
156	The Byrds	2:34	415	D major	
157	Judy Collins	5:26	877	B major	
158	Con-Funk-Shun	3:02	491	C minor	
159	Freud Marx Engels & Jung	4:26	718	D major	Hra Tampuurimies
160	Johnny Johnson & His Bandwagon	3:07	503	A major	
161	Jormas	2:12	356	A minor	
162	Melanie	4:24	711	C major	
163	Mountain	5:31	890	D major	
164	Odetta	10:44	1735	G major	
165	Bob Sinclair	4:59	804	C minor	

Table C.15: Content of the Mixed dataset cover song set 15.

Set 16: My Generation

Song ID	Performer	Time	Frames	Key	Title
166	The Who	3:19	534	G minor	
167	Count Five	3:06	501	A major	
168	Green Day	2:19	376	G♯ major	
169	Iron Maiden	3:37	584	A minor	
170	Manfred Mann	2:30	404	B♭ minor	
171	The Melvins	7:39	1235	F major	
172	Pelle Miljoona & 1980	3:05	497	A minor	
173	Rock Kids	3:19	535	G♯ major	
174	Patti Smith	3:20	537	G minor	
175	Sweet	3:56	635	C major	
176	Virtanen	3:05	499	D minor	Mun sukupolvi

Table C.16: Content of the Mixed dataset cover song set 16.

Set 17: My Heart Will Go On

Song ID	Performer	Time	Frames	Key	Title
177	Celine Dion	4:41	757	E major	
178	Michael Ball	4:39	751	A major	
179	Belfast Harp Orchestra	4:30	728	$B\flat$ major	
180	Saras Brightman	4:28	723	A major	Il Mio Cuore Va
181	Richard Clayderman	3:43	602	E major	
182	Neil Diamond	4:13	682	E major	
183	James Galway	4:50	782	F major	
184	Kaapo & Zetor	3:17	530	C major	Uskon sydämen totuuteen
185	Kenny G	4:23	709	$B\flat$ major	
186	Vicky Leandros	4:01	650	E major	Weil Mein Herz Dich Nie Mehr Vergisst
187	Paul Potts	4:27	721	A major	Il Mio Cuore Va

Table C.17: Content of the Mixed dataset cover song set 17.

Set 18: Oh, Pretty Woman

Song ID	Performer	Time	Frames	Key	Title
188	Roy Orbison	3:00	484	A major	
189	Agents	3:38	586	A minor	
190	Bad News	2:51	461	E major	
191	Al Green	3:25	552	A major	
192	Chris Isaak	2:52	464	A major	
193	Tapani Kansa	4:44	764	C minor	Kaunis nainen
194	John Mayall & The Bluesbreakers	3:40	592	$F\sharp$ minor	
195	Popeda	2:59	482	A major	Kaunis nainen
196	Sharleen Spiteri	2:17	370	E minor	
197	Van Halen	2:53	466	$G\sharp$ minor	
198	The Ventures	2:53	466	A major	

Table C.18: Content of the Mixed dataset cover song set 18.

Set 19: Paint It, Black

Song ID	Performer	Time	Frames	Key	Title
199	The Rolling Stones	3:45	608	C minor	
200	Africa	7:35	1225	C minor	
201	Vanessa Carlton	3:30	566	A minor	
202	Deep Purple	5:35	903	E minor	
203	Echo & The Bunnymen	3:15	523	E major	
204	Flamin' Groovies	3:02	490	G minor	
205	Chris Farlowe	3:30	565	E minor	
206	Frederik	3:21	541	B minor	Pikku musta
207	Popeda	3:29	564	E minor	Mustaa
208	Sixth Finger	4:08	668	A minor	
209	W.A.S.P.	3:29	562	E major	

Table C.19: Content of the Mixed dataset cover song set 19.

Set 20: Proud Mary

Song ID	Performer	Time	Frames	Key	Title
210	Creedence Clearwater Revival	3:08	505	D minor	
211	Solomon Burke	7:10	1158	G major	
212	Cagey Strings	3:06	501	D major	
213	Tom Jones	2:12	356	D minor	
214	Helmut Lotti	3:55	632	C major	
215	Leonard Nimoy	3:20	539	G major	
216	Number Nine	2:42	436	A minor	
217	Elvis Presley	2:47	450	G major	
218	Status Quo	3:33	575	D major	
219	Ike & Tina Turner	2:37	424	G♯ minor	
220	The Voices Of East Harlem	2:49	455	B♭ major	

Table C.20: Content of the Mixed dataset cover song set 20.

Set 21: (I Can't Get No) Satisfaction

Song ID	Performer	Time	Frames	Key	Title
221	The Rolling Stones	3:44	603	A major	
222	Pimpi Arrayo	4:57	801	C♯ major	
223	Devo	2:38	424	D minor	
224	Chris Farlowe	2:26	395	E minor	
225	Buddy Guy	3:41	596	F minor	
226	Tom Jones	2:09	348	G major	
227	Manfred Mann	2:51	459	G♯ major	
228	Otis Redding	2:46	446	E minor	
229	The Residents	4:31	730	G♯ minor	
230	Rhythms Del Mundo featuring Cat Power	3:01	488	A minor	
231	Charles Wright & The Watts 103rd Street Rhythm Band	3:11	515	E minor	

Table C.21: Content of the Mixed dataset cover song set 21.

Set 22: Smells Like Teen Spirit

Song ID	Performer	Time	Frames	Key	Title
232	Nirvana	5:01	812	G♯ major	
233	2Cellos	2:52	462	G minor	
234	Tori Amos	3:36	582	G♯ major	
235	The Bad Plus	5:57	961	G♯ major	
236	David Garrett	4:07	665	C major	
237	Robert Glasper Experiment	7:25	1199	F major	
238	Ituana	4:22	704	F♯ major	
239	Melvins featuring Leif Garrett	5:02	813	G♯ major	
240	The Muppet Barbershop Quartet	2:23	386	B♭ major	
241	Patti Smith	6:31	1053	C minor	
242	Warp Brothers	3:30	565	G♯ major	

Table C.22: Content of the Mixed dataset cover song set 22.

Set 23: Something

Song ID	Performer	Time	Frames	Key	Title
243	The Beatles	3:01	486	C major	
244	Gene Ammons	3:20	537	C major	
245	Chet Atkins, Jerry Reed & Suzy Bogguss	3:25	551	C major	
246	Count Basie	3:25	552	A minor	
247	Shirley Bassey	3:35	579	D minor	
248	Tony Bennett	3:19	536	B♭ major	
249	The Blues Busters	2:34	416	C major	
250	Joe Cocker	5:33	896	C major	
251	Perry Como	3:34	577	G major	
252	Leisure Society	3:18	532	G♯ minor	
253	The Shadows	2:45	445	C major	

Table C.23: Content of the Mixed dataset cover song set 23.

Set 24: Stand by Me

Song ID	Performer	Time	Frames	Key	Title
254	Ben E. King	2:55	472	A major	
255	Ry Cooder	3:43	602	G major	
256	John Lennon	3:31	569	A minor	
257	Pave Maijanen	3:56	635	F major	Jää mun luo
258	Quicksilver Messenger Service	3:35	579	C major	
259	Seal	4:06	662	A major	
260	The Searchers	3:34	577	G♯ minor	
261	Ike & Tina Turner	3:47	610	C major	
262	The Ventures	3:58	642	A major	
263	Voiceboys	4:00	645	F major	
264	The Walker Brothers	3:59	642	E♭ major	

Table C.24: Content of the Mixed dataset cover song set 24.

Set 25: Summertime Blues

Song ID	Performer	Time	Frames	Key	Title
265	Eddie Cochran	1:56	313	A minor	
266	The Beach Boys	2:10	351	E major	
267	The Boys	2:15	363	G major	Kesäduuni blues
268	Dion	3:12	517	E major	
269	Eläkeläiset	1:47	289	G major	Vaivasenluut
270	Robert Gordon & Link Wray	2:17	370	E major	
271	Joan Jett & The Blackhearts	2:17	370	A major	
272	Rush	3:41	596	A major	
273	The Brian Setzer Orchestra	3:07	504	E major	
274	James Taylor	2:39	430	A major	
275	The Who	2:35	418	A major	

Table C.25: Content of the Mixed dataset cover song set 25.

Set 26: The Weight

Song ID	Performer	Time	Frames	Key	Title
276	The Band	4:33	735	A minor	
277	Deana Carter	4:54	791	A major	
278	Joe Cocker	5:57	961	A major	
279	Shannon Curfman	5:26	877	A major	
280	John Denver	4:30	726	C major	
281	Jeff Healey	4:26	716	G major	
282	Little Feat featuring Bela Fleck	5:18	856	G major	
283	Joan Osborne	5:13	843	B♭ minor	
284	Rotary Connection featuring Minnie Riperton	3:26	555	G major	
285	Sweet Suzi & The Blues Experience	3:54	630	D major	
286	Cassandra Wilson	6:05	982	G major	

Table C.26: Content of the Mixed dataset cover song set 26.

Set 27: What's Going On

Song ID	Performer	Time	Frames	Key	Title
287	Marvin Gaye	3:53	627	E major	
288	Azymuth	5:27	879	B♭ major	
289	Joe Cocker	5:13	844	E♭ major	
290	Etta James	4:27	719	D major	
291	Cyndi Lauper featuring Chuck D	4:38	750	C minor	
292	Los Lobos	5:25	876	G♯ minor	
293	Mica Paris	3:21	541	G major	
294	A Perfect Circle	4:53	789	E major	
295	Seal	4:27	719	E♭ major	
296	Take 6 featuring Brian McKnight	4:13	682	E major	
297	Weather Report	6:28	1044	D major	

Table C.27: Content of the Mixed dataset cover song set 27.

Set 28: A Whiter Shade of Pale

Song ID	Performer	Time	Frames	Key	Title
298	Procol Harum	4:07	665	C major	
299	King Curtis	5:19	858	C major	
300	Keith Emerson, Glenn Hughes & Marc Bonilla	5:39	915	C major	
301	The Everly Brothers	4:52	788	C major	
302	Gregorian	4:58	802	A major	
303	Pentti Hietanen	4:51	783	C major	
304	Annie Lennox	5:18	857	C major	
305	Helmut Lotti	4:19	697	C major	
306	The Shadows	5:00	806	E major	
307	Shorty Long	2:59	483	C major	
308	Vikingarna	3:47	610	C major	

Table C.28: Content of the Mixed dataset cover song set 28.

Set 29: With a Little Help from My Friends

Song ID	Performer	Time	Frames	Key	Title
309	The Beatles	2:44	441	A major	
310	Count Basie	3:23	545	G minor	
311	Cheap Trick	2:37	422	A major	
312	Joe Cocker	5:11	838	A major	
313	Easy Star All Stars featuring Luciano	3:13	518	E major	
314	Sergio Mendes & Brasil '66	2:38	425	G major	
315	Puerto Muerto	2:34	415	E major	
316	Santana	4:10	672	A minor	
317	A Savage	2:47	448	A major	
318	Wet Wet Wet	2:37	424	E major	
319	Young Idea	2:32	409	F major	

Table C.29: Content of the Mixed dataset cover song set 29.

Set 30: You Really Got Me

Song ID	Performer	Time	Frames	Key	Title
320	The Kinks	2:17	368	G♯ major	
321	801	3:23	546	D major	
322	Pimpi Arroyo	5:19	859	D minor	
323	Jim Lea	2:48	453	D minor	
324	Pelle Miljoona & N.U.S.	2:19	373	F major	
325	Mott The Hoople	8:55	1441	A minor	
326	Oingo Boingo	4:37	745	E major	
327	Silicon Teens	3:00	485	G♯ major	
328	Sly Stone	3:46	609	D major	
329	Van Halen	2:38	425	G♯ major	
330	The West Coast Pop Art Experimental Band	3:07	503	G♯ major	

Table C.30: Content of the Mixed dataset cover song set 30.

Noise tracks

The following table contains the 670 "noise" tracks of $Mixed_{1000}$. For these, we report here only the lengths in frames and and estimated keys. Occasionally, the key estimation algorithm could not determine the key of the piece; for these pieces, the key is denoted with a question mark. In order to make the table fit the page, the titles of the pieces are occasionally truncated; the piece of music should still be possible to trace according to the name of the performer and the shortened title.

Song ID	Performer	Frames	Key	Title
331	10CC	855	?	Rubber Bullets
332	4Hero	883	F minor	Spirit in Transit
333	911	565	G major	A Little Bit More
334	ABC	618	F major	When Smokey Sings
335	Actified	696	A minor	Crucifixion
336	Adam & The Ants	503	A minor	Stand and Deliver
337	Ryan Adams	665	A major	Shallow
338	Cannonball Adderley	502	G minor	Mercy, Mercy, Mercy
339	Adele	562	E♭ major	Chasing Pavements
340	Adolescents	292	A major	LA Girl
341	Christina Aguilera	586	G♯ major	Genie in a Bottle
342	A-Ha	826	A minor	The Sun Always Shines on TV
343	Air	1155	B minor	La Femme D'Argent

Table C.31: Content of the Mixed dataset noise track set.

Song ID	Performer	Frames	Key	Title
344	Air Supply	628	C major	All Out of Love
345	Alessi's Ark	331	E major	Hand in the Sink
346	Alice In Chains	564	B♭ minor	Would
347	Alien Sex Fiend	1004	E minor	Now I'm Feelind Zombified
348	Lee Allen & His Band	421	C minor	Tic Toc
349	The Alleycats	522	G♯ major	Nothing Means Nothing Anymore
350	Mose Allison	738	G♯ minor	One of Those Days
351	Altered Images	566	C major	I Could Be Happy
352	Curtis Amy feat. Dupree Bolton	497	G♯ minor	Katanga
353	Anastacia	652	G♯ major	I'm Outta Love
354	Pernilla Andersson	465	G major	Dansa med dig
355	Peter Andre	582	E♭ major	Mysterious Girl
356	Aneka	635	D major	Japanese Boy
357	Angel	484	A major	Good Time Fanny
358	Johnny Angel	264	G minor	Teenage Wedding
359	Aphrodite's Child	943	A minor	The Four Horsemen
360	Tasmin Archer	665	F minor	Sleeping Satellite
361	Argent	650	F major	God Gave Rock'n'Roll to You
362	Art Of Noise	702	C♯ minor	Moments in Love
363	Asa	568	G♯ minor	No One Knows
364	Rick Astley	570	G♯ major	Never Gonna Give You Up
365	Aswad	648	G♯ major	Set Them Free
366	The Ataris	579	D major	The Night the Lights Went ...
367	Athlete	822	B♭ major	Shake Those Windows
368	Atomic	1113	F♯ minor	Pyramid Song
369	Atomic Kitten	565	A major	Right Now
370	Attack	710	A minor	Mr. Pinnodmy's Dilemma
371	Audion	2073	G♯ major	Mouth to Mouth
372	The Aurora Pushups	512	E major	Victims of Terrorism
373	The Avengers	431	E minor	We Are the One
374	Average White Band	644	G♯ major	Pick Up the Pieces
375	David Axelrod	876	G minor	Holy Thursday
376	Roy Ayers	776	A minor	I Love You Michelle
377	Babylon Zoo	642	C major	Spaceman
378	Tal Bachman	603	A major	She's So High
379	Baha Men	530	C major	Who Let the Dogs Out
380	Chet Baker Quartet	492	C♯ major	But Not for Me
381	Baltimora	550	F major	Tarzan Boy
382	The Band Of Holy Joy	789	D minor	Who Snatched the Baby
383	Bangles	595	B major	Going Down to Liverpool
384	Pato Banton	621	C minor	Baby Come Back
385	Ray Barretto	437	D major	El Watusi
386	John Barry	381	F major	Black Stockings
387	Bay City Rollers	604	E major	Yesterday's Hero
388	Beady Belle featuring Lech	768	D minor	Goldilocks
389	Jimmy Beasley	368	G♯ major	I'm So Blue
390	Beat Assailant	864	C♯ major	Hard Twelve (The Payout)
391	Robin Beck	530	C major	First Time
392	Bedrock featuring KYO	1044	C minor	For What You Dream of
393	Pierre Belmonde	626	A minor	Für Elise
394	Chuck Berry	472	G♯ major	Too Much Monkey Business
395	Richard Berry	369	C minor	Mad About You Baby
396	Big Star	443	B♭ major	Dony
397	Billie	809	G♯ major	Honey to the Bee
398	The Black Keys	742	A minor	Things Ain't Like They ...
399	Blackfoot Sue	635	E major	I'm Standing in the Road
400	Art Blakey	1933	C minor	Anthenagin
401	Mary J. Blige	721	C♯ minor	Family Affair
402	Blonde Redhead	776	C♯ minor	Elephant Woman
403	Blondie	537	D minor	Call Me
404	Barry Blue	621	D major	Do You Wanna Dance
405	Blur	970	A minor	Sing
406	Arthur Blythe	489	F minor	Autumn in New York (Part one)
407	Eddie Bo	399	E♭ major	I Love to Rock'n'Roll
408	The Boo Radleys	503	C♯ minor	Wake Up Boo!
409	Ken Boothe	599	B♭ major	Everything I Own
410	David Bowie	622	B♭ major	Life on Mars
411	Toni Braxton	776	C major	Spanish Guitar
412	Bread	526	E major	Make It with You
413	Bright Eyes	537	C major	Take It Easy (Love Nothing)
414	Meredith Brooks	636	A major	Bitch
415	Charles Brown	424	G♯ minor	I'll Always Be in Love with You
416	Roy Brown	376	G♯ minor	Diddy-Y-Diddy-O
417	Dave Brubeck Quartet	472	E♭ minor	Take Five
418	Ray Bryant	455	C major	Shake a Lady
419	Michael Buble	413	A minor	Peroxide Swing

Table C.31: Content of the Mixed dataset noise track set.

Song ID	Performer	Frames	Key	Title
420	Bernard Butler	853	A minor	Stay
421	The Buzzcocks	432	E major	Ever Fallen In Love . . .
422	Donald Byrd	922	C minor	Cristo Redentor
423	Jerry Byrne	324	E♭ major	Carry On
424	Calexico	557	C major	Across the Wire
425	Calling	558	D major	Wherever You Will Go
426	Candy	602	E major	Whatever Happened to Fun
427	Blu Cantrell	672	F minor	Hit 'Em Up Stype (Oops!)
428	Captain & Tenille	551	B minor	Love Will Keep Us Together
429	Belinda Carlisle	665	E major	Heaven is a Place on Earth
430	Kim Carnes	592	F minor	Bette Davis Eyes
431	Cartoons	495	C minor	Witch Doctor (Radio Mix)
432	Neko Case	541	D major	The Train from Kansas City
433	Catatonia	835	G♯ major	Road Rage
434	Serge Chaloff	906	E♭ major	Handful of Stars
435	Harry Chapin	612	F minor	Cats in the Cradle
436	Charles & Eddie	549	E minor	Would I Lie to You
437	Bobby Charles	378	G♯ major	I'll Turn Square for You
438	Ray Charles featuring Milt Jackson	877	C minor	The Genius After Hours
439	Cher	418	C major	Gypsys, Tramps & Thieves
440	Chicane featuring Máire Brennan	548	F minor	Saltwater
441	Chicory Tip	478	G major	What's Your Name
442	Christie	441	E major	Yellow River
443	June Christy	696	C♯ major	Something Cool
444	Chumbawamba	544	D major	Tubthumping
445	Jimmy Clanton	376	B♭ major	Ship on a Stormy Sea
446	Louis Clark	525	D major	Pachebel's Canon
447	Cockney Rebel	641	C major	Make Me Smile
448	Cozy Cole	579	D major	Topsy II
449	Ornette Coleman	1071	D major	Ramblin'
450	John Coltrane	1385	E major	Equinox
451	Roland Cook	375	B♭ major	I've Got a Girl
452	Sam Cooke	386	G♯ major	That's All I Need to Know
453	Alice Cooper	561	G major	School's Out
454	The Coral	415	A major	In the Morning
455	Jimmy Crawford	336	F major	I Love How You Love Me
456	Marshall Crenshaw	531	D major	Whenever You're on My Mind
457	Sonny Criss	806	G minor	West Coast Blues
458	The Crystals	370	B♭ major	Love You So
459	Jamie Cullum	720	G♯ major	It Ain't Necessarily So
460	Culture Club	539	C minor	Church of the Poison Mind
461	Cutting Crew	712	A major	(I just) Died in Your Arms
462	Dana	492	B♭ major	All Kinds of Everything
463	Johnny Dankworth & His Orchestra	383	B♭ minor	African Waltz
464	The Dandy Warhols	751	A minor	The Dope (Wonderful You)
465	Danse Society	812	D minor	We're So Happy
466	The Dark	597	D minor	The Masque
467	The Darkness	452	D minor	Get Your Hands Off My Woman
468	Dashboard Confessional	536	E♭ major	Vindicated
469	Daughter	532	E♭ major	Peter
470	Chris Davis	524	E♭ major	To a Wild Rose
471	Miles Davis	920	F minor	Frelon Brun
472	Taylor Dayne	585	G♯ major	Tell It to My Heart
473	The dB's	537	C major	Love is for Lovers
474	Matthew Dear	684	C minor	Fleece on Brain
475	Death in Vegas	628	A major	Aisha
476	The Decemberists	603	C major	Oh Valencia!
477	Deep Feeling	485	G minor	Pretty Colours
478	Deep Forest	625	A major	Sweet Lullaby
479	Deep Purple	470	A minor	Emmaretta
480	Delerium	1030	A minor	Silence
481	Gitane Demone	658	G minor	Incendiary Lover
482	Sandy Denny	859	A major	It'll Take a Long Time
483	Department S	443	B♭ minor	Is Vic There?
484	Descendents	328	D major	Myage
485	Destiny's Child	592	F♯ minor	Independent Women (Part one)
486	Dexy's Midnight Runners	534	C♯ minor	Geno
487	Diagrams	597	A minor	Night All Night
488	Lee Diamond	461	G♯ major	Hatti Malatti
489	Dick & Dee Dee	370	F minor	The Mountain's High
490	Dido	593	E major	Thank You
491	Digital Bled	806	A minor	Paciencia
492	Dirtmusic	1001	C♯ major	Morning Dew
493	Claire Diterzi	571	E♭ major	A Quatre Pattes
494	Divinyls	610	F major	I Touch Myself
495	Fats Domino	386	F major	Telling Lies

Table C.31: Content of the Mixed dataset noise track set.

Song ID	Performer	Frames	Key	Title
496	Jason Donovan	551	G major	Too Many Broken Hearts
497	Craig Douglas	410	C major	A Hundred Pounds of Clay
498	Big Al Downing	348	D minor	When My Blue Moon Turns ...
499	Duran Duran	703	C major	The Reflex
500	Baxter Dury	599	C♯ minor	Isabel
501	Ian Dury & The Blockheads	517	E major	Sex & Drusg & Rock & Roll
502	Earth Wind & Fire	461	G major	Shining Star
503	Kylie Eastwood	565	G minor	Big Noise (From Winnetka)
504	Dave Edmunds	451	E major	I Hear You Knocking
505	Teddy Edwards & Les McCann Ltd.	945	?	Our Love is Here to Stay
506	Lisa Ekdahl	610	D major	Öppna ditt fönster
507	Elbow	832	B major	Switching Off
508	Electro Deluxe & Cynthia Saint-Ville	975	G♯ major	Mister Freeze
509	Duke Ellington Orchestra	461	G minor	Minnie the Moocher
510	Don Ellis Orchestra	906	C minor	Alone
511	Embrace	677	A major	Hooligan
512	EMF	568	G♯ minor	Unbelievable
513	Brian Eno	635	E major	Deep Blue Day
514	Erland & The Carnival	473	D minor	Map of an Englishman
515	Europe	641	?	The Final Countdown
516	Eurythmics	727	F major	When Tomorrow Comes
517	Bill Evans	1082	C major	Peace Piece
518	Adam Faith	309	F major	Easy Going Me
519	Faithless	582	D minor	Drifting Away
520	Harold Faltermeyer	486	G♯ major	Axel F
521	The Farm	925	D major	All Together Now
522	John Farnham	826	B♭ major	You're the Voice
523	Fatboy Slim	1113	G major	The Rockafeller Skank
524	Paolo Fedreghini & Marco Bianchi	731	D major	Oriental Smile
525	The Victor Feldman Quartet	501	A minor	A Taste of Honey
526	The Felice Bros	662	F major	Frankie's Gun
527	Felt	595	A minor	Grey Streets
528	Shane Fenton & The Fentones	424	D major	I'm a Moody Guy
529	Fertile Ground feat. Navasha Daya	577	C minor	Yellow Daisies
530	Fiction Factory	568	E♭ major	(Feels like) Heaven
531	Fields Of The Nephilim	789	G major	Preacher Man
532	Neil Finn	717	C minor	Sinner
533	Tim Finn	676	D major	Fraction to Mutch Friction
534	Ella Fitzgerald	758	B major	Willow Weep for Me
535	The Five Corners Quintet	865	D minor	Trading Eights
536	Flaming Lips	1515	D minor	One Million Billionth ...
537	Fleet Foxes	740	C♯ minor	Mykonos
538	The Flesheaters	381	A major	Pony Dress
539	Johnny Flynn	459	B♭ major	In the Honour of Industry
540	Frankie Ford	443	E♭ major	It Must Be Jelly
541	Marcus Foster	743	E major	Circle in the Square
542	Four Tet	821	B minor	My Angel Rocks Back and Forth
543	The Four Tops	730	E major	Loco in Acapulco
544	Fox The Fox	645	E major	Precious Little Diamond
545	John Foxx	517	D minor	Burning Car
546	John Fred & The Playboys	311	B♭ minor	Shirley
547	Frankie Goes To Hollywood	632	D minor	Two Tribes
548	Glenn Frey	968	A major	Part of Me, Part of You
549	Fujiya & Miyagi	805	B minor	Ankle Injuries
550	Farley Jackmaster Funk	1111	C minor	The Acid Life
551	Nelly Furtado	655	B♭ major	I'm Like a Bird
552	Peter Gabriel	906	G major	Red Rain
553	Galaxie 500	606	G major	Tell Me
554	Gang Of Four	505	D minor	Natural's Not in It
555	Garbage	585	B♭ major	I Think I'm Paranoid
556	Paul Gayten	383	G major	Nervous Boogie
557	Generation X	371	A major	King Rocker
558	The Gentle Rain	627	C minor	Plastic Man
559	Geordie	545	C minor	Goodbye Love
560	Germs	503	B minor	Forming
561	Stan Getz	331	D minor	Desafinido
562	Joolz Gianni	793	B♭ major	Silver
563	Gigolo Aunts	618	G major	Cope
564	Dizzy Gillespie featuring Joe Caroll	472	F minor	Groovin' the Nursery Rhymes
565	Girls At Our Best	317	D major	Getting Nowhere Fast
566	Gary Glitter	565	A major	I'm the Leader of the Gang
567	Jimmy Gnecco	827	E major	Someone to Die for
568	Go West	572	F♯ minor	We Close Our Eyes
569	Gomez	631	F♯ minor	Bring It On
570	The Gondoliers	348	G major	You Call Everybody Darling
571	The Good, The Bad & The Queen	1013	G minor	The Good, The Bad & ...

Table C.31: Content of the Mixed dataset noise track set.

Song ID	Performer	Frames	Key	Title
572	Bob Gordon feat. Jack Montrose	396	F minor	Two Can Play
573	Dexter Gordon	959	B♭ minor	Body and Soul
574	Junior Gordon	378	C♯ major	Blow Wind Blow
575	Ellie Goulding	508	B♭ major	Your Song
576	Macy Gray	769	C minor	Sexual Revolution
577	Grays	652	D major	Same Thing
578	Great Buildings	613	A major	Hold on to Something
579	Norman Greenbaum	649	A minor	Spirit in the Sky
580	Greenberry Woods	550	G major	Trampoline
581	Nancy Griffith	704	C major	Good Night, New York
582	Groove Armada	683	G♯ major	At the River
583	Groove Master	356	E♭ major	Winter
584	The Vince Guaraldi Trio	504	G♯ major	Cast Your Fate to the Wind
585	Jimmy Guiffre	775	C♯ minor	Ironic
586	Josh Guru	655	B♭ major	Infinity
587	Woody Gutherie	357	G major	Hard Travelin'
588	Alice Hagenbrandt	548	B major	Samson
589	Haircut 100	482	C♯ major	Favourite Shirts
590	Half Man Half Biscuit	1015	A minor	National Shite Day
591	Alberta Hall	462	C minor	Oh, How I Need Your Lovin'
592	Daryl Hall & John Oates	684	A major	Maneater
593	Chico Hamilton Quintet	299	C minor	The Squimp
594	Herbie Hancock	472	F minor	Watermelon Man
595	Happy Mondays	857	A minor	Step On
596	Hard-Fi	459	B♭ minor	Watch Me Fall Apart
597	Paul Hardcastle	571	A minor	19
598	Harmonia	570	A major	Dino
599	Eddie Harris	453	A minor	Tampion
600	David Hasselhoff	632	C major	Looking for Freedom
601	Hampton Hawes Trio	557	E♭ major	I Hear Music
602	Chesney Hawkes	592	B minor	The One and Only
603	Isaac Hayes	530	C major	Theme from Shaft
604	Roy Haynes	1333	A minor	Quiet Fire
605	Hello	493	F major	Good Old USA
606	Hello Bye Bye	763	F minor	Don't Look at the Past
607	Clarence Henry	389	E♭ major	Baby, Baby Please
608	The Matthew Herbert Big Band	763	D major	Everything's Changed
609	The Hold Steady	499	D major	Chips Ahoy!
610	Nick Holder	688	G♯ minor	Sometime I'm Blue
611	Holly & The Italians	490	G major	Tell That Girl to Shut Up
612	Richard Holmes	317	G♯ major	Misty
613	Hoobastank	533	D minor	Did You
614	Hoodoo Gurus	516	G major	I Want You Back
615	Dr. Hook	475	C♯ major	When You're in Love With ...
616	Hooverphonic	643	B♭ major	Waves
617	Bruce Hornsby	886	B major	Look Out Any Window
618	Hot Chocolate	648	B♭ major	You Sexy Thing
619	Hothouse Flowers	652	A minor	Hard Rain
620	Ben Howard	571	F major	Three Tree Town
621	Howlin' Rain	956	A minor	Dancers at the End of Time
622	Freddie Hubbard	502	B♭ minor	Lonely Soul
623	The Human League	555	C minor	Love Action (I Believe in Love)
624	Humble Pie	518	A minor	Growing Closer
625	Mississippi John Hurt	446	B♭ major	Candy Man Blues
626	Susi Hyldgaard	688	D major	Blush
627	Idle Wilds	546	B♭ major	You're All Forgiven
628	Billy Idol	775	B minor	Rebel Yell
629	Imagination	592	A major	Just an Illusion
630	Natalia Imbruglia	762	D major	That Day
631	In Excelsis	776	D minor	The Sword
632	Inspiral Carpets	515	B major	This is How It Feels
633	Interpol	916	A minor	Pioneer to the Falls
634	Bon Iver	626	E minor	Skinny Love
635	Mahalia Jackson	509	C major	God's Gonna Separate ...
636	Michael Jackson	802	F minor	The Way You Make Me Feel
637	The Jam	471	D major	A Town Called Malice
638	James	616	A major	Destiny Calling
639	Jamie T	937	D major	Operation
640	Jane's Addiction	930	G♯ minor	The Riches
641	Jean Michel Jarre	508	G minor	Oxygene 2
642	Keith Jarrett	612	A major	Margot
643	The Jazz Crusaders	588	G minor	Young Rabbits
644	Jellyfish	690	A major	This Is Why
645	Jet	655	G major	Hold On
646	Joan As Police Woman	798	G minor	To Be Lonely
647	JoBoxers	572	G♯ major	Just Got Lucky

Table C.31: Content of the Mixed dataset noise track set.

Song ID	Performer	Frames	Key	Title
648	Billy Joel	577	E♭ major	The Longest Time
649	Elton John	857	E♭ major	Philadelphia Freedom
650	Ana Johnson	632	G♯ minor	We Are
651	Elvin Jones	951	C♯ minor	Elvin Elpus
652	Howard Jones	543	C♯ minor	What is Love
653	Joe Jones	377	E♭ major	A-Tisket A-Tasket
654	Norah Jones	533	C major	Come Away with Me
655	Sharon Jones	599	A minor	100 Days, 100 Nights
656	Journey	666	G major	Only the Young
657	Ernie Kador	366	G major	Eternity
658	Kajagoogoo	597	G♯ major	Too Shy
659	Ini Kamoze	670	E major	Here Comes the Hotstepper
660	Maria Kannegaard	438	G major	Hey Ya
661	Kasabian	582	C minor	Club Foot
662	Katrina & The Waves	448	E♭ major	Walking on Sunshine
663	Eddie Kendricks	576	G♯ major	Keep on Truckin'
664	Kenny	557	E major	Fancy Pants
665	Nik Kershaw	591	A major	The Riddle
666	Alicia Keys	584	F♯ major	Empire State of Mind (Part two)
667	Chaka Khan	931	E major	I Feel for You
668	Killing Joke	712	C minor	Love Like Blood
669	The Kills	538	G♯ major	Last Day of Magic
670	Earl King	481	F major	Well-O, Well-O, Well-O Baby
671	Kings Of Leon	548	?	On Call
672	Fern Kinney	675	B♭ major	Together We Are Beautiful
673	Kinny & Horn	710	G minor	Sacred Life
674	Kinobe	730	C minor	Slip into Something More ...
675	Rashaan Roland Kirk	584	C minor	Spirits Up Above
676	Kit	598	F minor	Mermaid
677	Michael Kiwanuka	650	A minor	Tell Me a Tale
678	The Knack	648	G minor	My Sharona
679	The Knife	773	E minor	Silent Shout
680	Beverly Knight	612	E♭ major	Greatest Day
681	Gladys Knight	751	C♯ major	Midnight Train to Georgia
682	The Moe Koffman Quartette	362	C major	The Swingin' Shepherd Blues
683	Komputer	661	C major	Like a Bird
684	Lee Konitz	576	G major	Five, Four and Three
685	Kraftwerk	621	B♭ minor	The Robots
686	Laleh	584	C♯ minor	Live Tomorrow
687	The La's	435	G major	There She Goes
688	Yusuf Lateef	1223	F minor	Like It Is
689	Lawrence Arabia	559	G major	Dream Teacher
690	LCD Soundsystem	1079	G minor	45:33
691	Harry Lee	362	A minor	Every Time I See You
692	Peggy Lee feat. George Shearing	610	F major	Do I Love You
693	Leftfield	526	G minor	A Final Hit
694	Benjamin Francis Leftwich	659	G major	More Than Letters
695	John Legend	759	B♭ major	Ordinary People
696	Leila	556	B minor	Little Acorns
697	Lemonheads	441	D major	Into Your Arms
698	Let's Active	469	E major	Every Word Means No
699	Huey Lewis & The News	759	E minor	Small World
700	The Ramsey Lewis Trio	495	D minor	The In Crowd
701	Smiley Lewis	364	E major	Someday (You'll Want Me)
702	The Lightning Seeds	652	C major	The Life of Riley
703	Marie Lindberg	491	A major	Trying to Recall
704	Jeanette Lindstrom	758	G♯ major	Here
705	Booker Little	404	G♯ major	Doin' the Hambone
706	Little Richard	351	G minor	Hey-Hey-Hey-Hey
707	Jennifer Lopez	658	G minor	Ain't It Funny
708	Lost Soul Division	690	E major	Castaway
709	Lostprophets	712	A minor	Lucky You
710	Louise	573	D major	Naked
711	Jon Lucien	430	E♭ major	A Sunny Day
712	Bascom Lamar Lunsford	482	G♯ major	Dry Bones
713	John Lytle	372	E♭ major	The Loop
714	Madness	531	G major	It Must Be Love
715	Magazine	643	C♯ minor	Shot by Both Sides
716	Magic Numbers	841	C♯ minor	The Mule
717	Bobby Mandolph	369	C♯ minor	Malinda
718	Manic Street Preachers	673	F major	My Little Empire
719	Herbie Mann	724	D minor	Consolacao
720	Bobby Marchan	370	G major	Chickee Wah Wah
721	Marillion	573	D major	Kayleigh
722	Hank Marr	503	G minor	The Greasy Spoon
723	Richard Marx	659	C major	Right Here Waiting

Table C.31: Content of the Mixed dataset noise track set.

Song ID	Performer	Frames	Key	Title
724	Willy Mason	455	G major	So Long
725	Massive Attack	889	A minor	Teardrop
726	Matthews' Southern Comfort	719	C minor	Woodstock
727	MC Hammer	685	G major	U Can't Touch This
728	Les McCann	462	D minor	The Shampoo
729	McCarthy	579	A major	Red Sleeping Beauty
730	Martine McCutcheon	616	C♯ major	Perfect Moment
731	Jimmy McGriff	418	F major	I've Got a Woman (Part one)
732	Don McLean	665	G major	American Pie (Part one)
733	George McRae	533	G♯ major	Rock Your Baby
734	Tom McRae	551	B minor	Ghost of a Shark
735	Meat Loaf	871	D major	I'd Do Anything for Love
736	Medicine Head	563	E major	One and One is One
737	Mekons	371	G major	Abernant 1984/5
738	Mel & Kim	792	G major	Showin' Out
739	Mercury Rev	833	G♯ major	Opus 40
740	Metropolitan Jazz Affair	752	G minor	Escapism
741	Miami Sound Machine	703	C major	Dr. Beat
742	Mike & The Mechanics	891	G♯ major	The Living Years
743	Amos Milburn	456	G minor	Chicken Shack Boogie
744	Charles Mingus	602	D minor	Moves
745	The Mississippi Sheiks	542	G major	The World is Going Wrong
746	Bobby Mitchell	353	B major	Try Rock and Roll
747	Robert Mitchum	412	C minor	The Ballad of Thunder Road
748	Moby	599	C major	Why Does My Heart Feel So Bad
749	Thelonious Monk	1006	B♭ minor	Blue Bolivar Blues
750	Gary Moore	696	D minor	Empty Rooms
751	Morcheeba	653	G major	World Looking In
752	Lee Morgan	508	G♯ major	The Sidewinder (Part one)
753	Motorhead	732	?	Hellraiser
754	Alison Moyet	591	G minor	All Cried Out
755	Mud	478	B major	Dyna-Mite
756	Gerry Mulligan feat. Shelly Manne	575	G♯ major	Black Nightgown
757	Mungo Jerry	568	E major	In the Summertime
758	Mark Murphy	364	E minor	My Favorite Things
759	Music Sculptors	532	A minor	X-Files
760	Myles & Dupont	403	C major	Loud Mouth Annie
761	Johnny Nash	450	G minor	I Can See Clearly Now
762	Kate Nash	659	C major	Foundations
763	The National	533	C major	Fake Empire
764	Sandy Nelson	370	G♯ major	Let There Be Drums
765	Nena	624	A major	99 Luftballons
766	Art Neville	423	A major	Cha Dooky Doo
767	New Edition	636	C♯ minor	Candy Girl
768	New Model Army	416	F major	51st State
769	New Order	846	C minor	True Faith
770	Joanna Newsom	1069	A minor	Colleen
771	Maxime Nightingale	518	?	Right Back Where ...
772	The Nightingales	544	D minor	My First Job
773	The Nightwatchman	698	C major	No One Left
774	Nikki O	771	G♯ major	Butterflies
775	Lisa Nilsson	579	?	Handens fem fingrar
776	Nits	706	D minor	Sketches of Spain
777	Oasis	785	A major	Cigarettes & Alcohol
778	Billy Ocean	652	E major	When the Going Gets Tough
779	Ocean Color Scene	799	E minor	The Riverboat Song
780	Odyssey	609	C minor	Use It Up and Wear It Out
781	The Offs	576	E major	624803
782	Ohio Players	621	C minor	Fopp
783	Oh Laura	499	G♯ major	Release Me
784	Alexander O'Neal	653	A major	Criticize
785	The Only Ones	492	E major	Another Girl, Another Planet
786	Declan O'Rourke	705	E♭ major	Sarah (Last Night in a Dream)
787	Orange Juice	509	E major	Lean Period
788	The Orb	1341	G♯ major	Outlands
789	William Orbit	1535	B♭ minor	Barber's Adagio for Strings
790	Our Theory featuring Erik Truffaz	778	G major	Our Theory
791	Outkast featuring Rosario Dawson	642	B minor	She Lives in My Lap
792	Panda Bear	651	B♭ major	Comfy in Nautica
793	Billy Paul	752	E♭ major	Me & Mrs. Jones
794	Art Pepper & Shorty Rogers Nine	545	F minor	Diablo's Dance
795	The Peppers	382	F♯ major	Pepper Box
796	Carolina Wallin Perez	702	G♯ minor	Utan dina andetag
797	Phosporescent	977	C♯ major	Cocain Lights
798	Pilot	496	G major	Magic
799	The Piltdown Men	391	E major	Piltdown Rides Again

Table C.31: Content of the Mixed dataset noise track set.

Song ID	Performer	Frames	Key	Title
800	Pink	518	?	Get the Party Started
801	Jay Jay Pistolet	475	E♭ major	Vintage Red
802	Placebo	673	B♭ minor	Without You I'm Nothing
803	Play Dead	517	E minor	Propaganda
804	Plimsouls	577	E minor	A Million Miles Away
805	The Pointer Sisters	616	G minor	I'm So Excited
806	Iggy Pop	843	A major	Lust for Life
807	Posies	548	F major	Solar Sister
808	Povo	924	E♭ minor	Uam Uam
809	Cozy Powell	583	E major	Dance with the Devil
810	Pravda	522	A minor	Tu Es à l'Quest
811	The Pretenders	685	G major	Middle of the Road
812	President	584	G major	You're Gonna Like It
813	Pretty Things	632	C minor	Baron Saturday
814	Andre Previn	528	E♭ major	Like Young
815	Lloyd Price	336	E♭ major	I'm Glad, Glad
816	Primal Scream	1708	C minor	Trainspotting
817	Professor Longhair	404	C minor	Look What You're Doing to Me
818	Public Image Ltd	678	E major	This is not a Love Song
819	Pulp	731	F major	Mile End
820	Suzi Quatro	632	G major	Crash
821	Jesse Quin & The Mets	481	D major	The Sculptor and the Stone
822	Gerry Rafferty	668	G♯ major	Baker Street
823	Bonnie Raitt	893	B♭ major	I Can't Make You Love Me
824	Ram Jam	404	G major	Black Betty
825	The Ramones	610	G♯ major	Baby, I Love You
826	The Randoms	658	A major	A-B-C-D
827	The Rapture	825	G♯ major	House of Jealous Lovers
828	Nathaniel Rateliff	650	D major	Early Spring Till
829	Ravens & Chimes	678	D major	Eleventh St.
830	Real McCoy	645	G major	Another Night
831	Redd Cross	542	G major	Lady in the Front Row
832	Helen Reddy	559	G minor	Angie Baby
833	Rednex	515	A major	Cotton Eye Joe
834	Redskins	621	F minor	Lev Bronstein
835	Jimmy Reed	373	A major	Take Out Some Insurance
836	Lou Reed	604	E♭ major	Perfect Day
837	Reef	590	G major	Place Your Hands
838	Martha Reeves	538	C♯ major	Wild Night
839	The Rembrants	720	A major	Rollin' Down the Hill
840	REO Speedwagon	647	G major	Take It on the Run
841	Ride	603	A major	Twistarella
842	Rilo Kiley	586	C major	Give a Little Love
843	Ritual	1062	E minor	Questioning the Shadow
844	Lester Robertson	429	F minor	My Girl Across Town
845	Robbie Robertson	805	C minor	Somewhere Down the . . .
846	The David Rockingham Trio	358	F major	Dawn
847	Jimmy Rodgers	449	D major	My Blue Eyed Jane
848	Romantics	479	A major	What I Like About You
849	Rooks	583	E major	Reasons
850	Rosetta Stone	784	F major	Deeper
851	The Royal Kings	337	F minor	Teachin' and Preachin'
852	Rubella Ballet	550	F major	Slant and Slide
853	The Ruby Suns	905	D major	Closet Astrologer
854	Alice Russel	1025	A minor	To Know This
855	Sade	552	D major	When Am I Going to . . .
856	Severed Heads	1045	G♯ minor	Dead Eyes Opened
857	Charlie Sexton	902	A minor	Badlands
858	Phil Seymour	492	C♯ minor	Baby It's You
859	Shaggy	668	G minor	Boombastic
860	Helen Shapiro	432	C major	You Don't Know
861	Shocking Blue	492	A minor	Venus
862	Showaddywaddy	559	A major	Rock'n'Roll Lady
863	Carly Simon	696	A minor	You're So Vain
864	Simple Minds	623	G♯ major	Promised You a Miracle
865	Sleeper	832	D minor	Atomic
866	Slik	668	E minor	The Kid's a Punk
867	Small Faces	497	D major	All or Nothing
868	The Sounds Of Tomorrow	413	C minor	Space Child
869	Spandau Ballet	579	C♯ major	Gold
870	Sparks	565	G major	Girl from Germany
871	Spear Of Destiny	689	A minor	Never Take Me Alive
872	Britney Spears	549	C minor	I'm a Slave 4 U
873	The Specials	592	G♯ major	Ghost Town
874	Spin Doctors	630	G minor	Little Miss Can't Be Wrong
875	Spongetones	402	E major	She Goes Out with Everybody

Table C.31: Content of the Mixed dataset noise track set.

Song ID	Performer	Frames	Key	Title
876	Lisa Stansfield	727	D major	8-3-1
877	The Staple Singers	719	C major	I'll Take You There
878	Edwin Starr	542	F♯ major	War
879	Starship	727	B major	Nothing's Gonna Stop Us Now
880	Stereophonics	541	E major	Have a Nice Day
881	Rod Stewart	844	D major	Maggie May
882	Stiff Little Fingers	482	A major	At the Edge
883	Angie Stone	721	C minor	Brotha
884	The Stone Roses	599	E major	I am the Resurrection
885	The Stranglers	406	F major	All Day and All of the Night
886	Stylistics	518	?	Can't Give You Anything ...
887	Suede	771	E♭ major	The Wild Ones
888	The Sundays	626	G major	Here's Where the Story Ends
889	Super Furry Animals	325	G♯ major	Do or Die
890	Supergrass	481	A minor	Alright
891	Survivor	668	C minor	Eye of the Tiger
892	Billy Swan	647	C major	I Can Help
893	Sweet	656	E minor	Ballroom Blitz
894	Matthew Sweet	589	D major	I've Been Waiting
895	Sweet Sensation	553	D major	Sad Sweer Dreamer
896	Swing Out Sister	552	E major	Breakout
897	Taking Back Sunday	678	D major	This Photograph is Proof
898	Talk Talk	828	D major	Today
899	Talking Heads	559	B minor	City of Dreams
900	Tame Impala	769	C major	Alter Ego
901	The Tamperer featuring Maya	524	G minor	Fell It
902	Tams	386	B♭ major	Hey Girl Don't Bother Me
903	Tangerine Dream	598	C minor	Rubycon (Part One)
904	Tavares	526	E minor	More Than a Woman
905	Tearaways	637	G major	Jessica Something
906	Technotronic featuring Felly	582	C minor	Pump Up the Jam
907	Teenage Fanclub	585	F major	Please Stay
908	Television Personalities	676	A major	Paradise Estate
909	Temperance Seven	495	B♭ major	Pasadena
910	Anna Ternheim	572	C major	Shoreline
911	Terrorvision	641	G♯ major	Tequila
912	Theatre Of Hate	539	D minor	Black Madonna
913	Thin Lizzy	723	G♯ major	The Boys are Back in Town
914	Thompson Twins	673	A minor	Love on Your Side
915	Tracey Thorn	339	G major	Plain Sailing
916	Three Blind Wolves	686	G major	Emily Rose
917	Three Degrees	473	A major	When I Will See You Again
918	Three Dog Night	534	G♯ major	Mama Told Me Not to Come
919	The Three Jonhs	431	A minor	The World of the Workers ...
920	The Thrills	804	B♭ major	Deckchairs and Cigarettes
921	Tin Tin Out & Emma Bunton	743	D major	What I Am
922	Cal Tjader	396	C minor	Soul Sauce
923	To Kill A King	565	A minor	Fictional State
924	Tok Tok Tok & Tokunbo Akinro	735	C major	About
925	The Tom Robinson Band	529	A major	2-4-6-8 Motorway
926	Tones On Tail	848	D minor	Burning Skies
927	Mel Torme	435	G minor	Comin' Home Baby
928	Tosca	596	G♯ major	Pyjama
929	Toto	804	A major	Africa
930	Allen Toussaint	382	F minor	Whirlaway
931	T'Pau	597	B major	China in Your Hand
932	Travis	716	E major	Why Does It Always Rain on Me
933	T-Rex	355	A major	I Love to Boogie
934	The Tropicals	393	F major	Sweet Sixteen
935	The Trost	526	C minor	Man on the Box
936	Bobby Troup	424	C minor	Route 66
937	Turkey Bones & The Wild Dogs	1377	A major	Raymond
938	Joe Turner	437	C minor	Honey Hush
939	Two Banks Of Four	1043	D major	One Day
940	UK Decay	501	D minor	Testament
941	Ultravox	603	F major	All Stood Still
942	Underworld	1572	E♭ major	Born Slippy
943	United Future Organisation	796	G♯ major	Loud Minority
944	Urinals	213	A major	Black Hole
945	Utopia	705	A major	Crybaby
946	Frankie Valli	572	A minor	My Eyes Adored You
947	Vanilla Ice	599	G minor	Ice Ice Baby
948	Bobby Vee	410	A major	Rubber Ball
949	Velocity Girl	532	E major	I Can't Stop Smiling
950	Velvet Crush	489	F♯ major	Hold Me Up
951	The View	579	?	Same Jeans

Table C.31: Content of the Mixed dataset noise track set.

Song ID	Performer	Frames	Key	Title
952	Gene Vincent	402	E major	Baby Blue
953	Virgin Prunes	557	C♯ minor	Pagan Love Song
954	Martha Wainwright	572	C major	Factory
955	John Waite	649	C♯ minor	Missing You
956	Ray Washington	333	G major	I Know
957	Waterboys	808	C major	The Whole of the Moon
958	Ben Watt	371	G major	North Marine Drive
959	Crystal Waters	597	D minor	Gypsy Woman (la de dee)
960	Ben Webster	535	C♯ major	There's No You
961	The Weirdos	379	C♯ minor	Life of Crime
962	Paul Weller	424	G major	Pink on White Walls
963	Bugge Wesseltoft	957	E minor	Min by
964	Wham	815	C major	Freedom
965	Whitesnake	745	C major	Is This Love
966	Chris Whitley	628	A major	
967	Jane Wieldin	658	E major	Rush Hour
968	Tomas Andersson Wij	398	G♯ major	Evighet
969	Kim Wilde	545	E minor	Kids in America
970	Charles Williams	363	F major	So Glad She's Mine
971	Hank Williams	436	D major	Lost Highway
972	Willard Grant Conspiracy	515	C major	Lost Hours
973	Kelly Willis	765	A major	Little Honey
974	Nicole Willis	584	C major	Feeling Free
975	Oscar Willis	355	B♭ minor	Flatfoot Sam
976	Edgar Winter Group	771	B♭ major	Frankenstein
977	The Wipers	665	A minor	D-7
978	Wire	461	A major	Outdoor Miner
979	Bill Withers	695	C major	Lean on Me
980	Wizzard	802	A major	See My Baby Jive
981	Wondermints	610	B minor	Proto-Pretty
982	Gloria Woods feat. Pete Candoli	452	G♯ major	Hey Bellboy!
983	World Party	689	F major	Ship of Fools
984	Xela	812	F major	Afraid of Monsters
985	X-Mal Deutschland	726	E minor	Incubus Succubus II
986	XTC	740	A minor	Senses Working Overtime
987	Yardbirds	385	F major	Shapes of Things
988	Yazoo	510	A major	Only You
989	Yellowcard	828	A major	Gifts and Curses
990	The Young Holt Trio	485	G♯ major	Wack Wack
991	Paul Young	797	A major	Love of the Common People
992	Robin Youngsmith	762	B major	The Flower Duet from Lakme
993	Frank Zappa	1457	A minor	Son of Mr. Green Genes
994	Thalia Zedek	1086	G major	Body Memory
995	Sophie Zelmani	463	A minor	Always You
996	The Zeros	419	A minor	Wimp
997	Hans Zimmer feat. Pete Haycock	657	A minor	Thunderbird
998	Muriel Zoe	552	C major	Bye Bye Blackbird
999	Zumpano	550	F major	The Party Rages On
1000	The Zutons	635	G♯ major	Valerie

Table C.31: Content of the Mixed dataset noise track set.

Chapter D

The Electronic Appendix

The source codes of the algorithms used in the evaluations can be found at `https://github.com/ahonenthesis/ncdcoversongs.git`. This repository also includes the chromagram data files for the pieces of music used in our experiments. In addition, all distance matrices and other results are provided.

TIETOJENKÄSITTELYTIETEEN LAITOS
PL 68 (Gustaf Hällströmin katu 2 b)
00014 Helsingin yliopisto

DEPARTMENT OF COMPUTER SCIENCE
P.O. Box 68 (Gustaf Hällströmin katu 2 b)
FI-00014 University of Helsinki, FINLAND

JULKAISUSARJA **A**

SERIES OF PUBLICATIONS **A**

Reports are available on the e-thesis site of the University of Helsinki.

A-2010-1 M. Lukk: Construction of a global map of human gene expression - the process, tools and analysis. 120 pp. (Ph.D. Thesis)

A-2010-2 W. Hämäläinen: Efficient search for statistically significant dependency rules in binary data. 163 pp. (Ph.D. Thesis)

A-2010-3 J. Kollin: Computational Methods for Detecting Large-Scale Chromosome Rearrangements in SNP Data. 197 pp. (Ph.D. Thesis)

A-2010-4 E. Pitkänen: Computational Methods for Reconstruction and Analysis of Genome-Scale Metabolic Networks. 115+88 pp. (Ph.D. Thesis)

A-2010-5 A. Lukyanenko: Multi-User Resource-Sharing Problem for the Internet. 168 pp. (Ph.D. Thesis)

A-2010-6 L. Daniel: Cross-layer Assisted TCP Algorithms for Vertical Handoff. 84+72 pp. (Ph.D. Thesis)

A-2011-1 A. Tripathi: Data Fusion and Matching by Maximizing Statistical Dependencies. 89+109 pp. (Ph.D. Thesis)

A-2011-2 E. Junttila: Patterns in Permuted Binary Matrices. 155 pp. (Ph.D. Thesis)

A-2011-3 P. Hintsanen: Simulation and Graph Mining Tools for Improving Gene Mapping Efficiency. 136 pp. (Ph.D. Thesis)

A-2011-4 M. Ikonen: Lean Thinking in Software Development: Impacts of Kanban on Projects. 104+90 pp. (Ph.D. Thesis)

A-2012-1 P. Parviainen: Algorithms for Exact Structure Discovery in Bayesian Networks. 132 pp. (Ph.D. Thesis)

A-2012-2 J. Wessman: Mixture Model Clustering in the Analysis of Complex Diseases. 118 pp. (Ph.D. Thesis)

A-2012-3 P. Pöyhönen: Access Selection Methods in Cooperative Multi-operator Environments to Improve End-user and Operator Satisfaction. 211 pp. (Ph.D. Thesis)

A-2012-4 S. Ruohomaa: The Effect of Reputation on Trust Decisions in Inter-enterprise Collaborations. 214+44 pp. (Ph.D. Thesis)

A-2012-5 J. Sirén: Compressed Full-Text Indexes for Highly Repetitive Collections. 97+63 pp. (Ph.D. Thesis)

A-2012-6 F. Zhou: Methods for Network Abstraction. 48+71 pp. (Ph.D. Thesis)

A-2012-7 N. Välimäki: Applications of Compressed Data Structures on Sequences and Structured Data. 73+94 pp. (Ph.D. Thesis)

A-2012-8 S. Varjonen: Secure Connectivity With Persistent Identities. 139 pp. (Ph.D. Thesis)

A-2012-9 M. Heinonen: Computational Methods for Small Molecules. 110+68 pp. (Ph.D. Thesis)

A-2013-1 M. Timonen: Term Weighting in Short Documents for Document Categorization, Keyword Extraction and Query Expansion. 53+62 pp. (Ph.D. Thesis)

A-2013-2 H. Wettig: Probabilistic, Information-Theoretic Models for Etymological Alignment. 130+62 pp. (Ph.D. Thesis)

A-2013-3 T. Ruokolainen: A Model-Driven Approach to Service Ecosystem Engineering. 232 pp. (Ph.D. Thesis)

A-2013-4 A. Hyttinen: Discovering Causal Relations in the Presence of Latent Confounders. 107+138 pp. (Ph.D. Thesis)

A-2013-5 S. Eloranta: Dynamic Aspects of Knowledge Bases. 123 pp. (Ph.D. Thesis)

A-2013-6 M. Apiola: Creativity-Supporting Learning Environments: Two Case Studies on Teaching Programming. 62+83 pp. (Ph.D. Thesis)

A-2013-7 T. Polishchuk: Enabling Multipath and Multicast Data Transmission in Legacy and Future Interenet. 72+51 pp. (Ph.D. Thesis)

A-2013-8 P. Luosto: Normalized Maximum Likelihood Methods for Clustering and Density Estimation. 67+67 pp. (Ph.D. Thesis)

A-2013-9 L. Eronen: Computational Methods for Augmenting Association-based Gene Mapping. 84+93 pp. (Ph.D. Thesis)

A-2013-10 D. Entner: Causal Structure Learning and Effect Identification in Linear Non-Gaussian Models and Beyond. 79+113 pp. (Ph.D. Thesis)

A-2013-11 E. Galbrun: Methods for Redescription Mining. 72+77 pp. (Ph.D. Thesis)

A-2013-12 M. Pervilä: Data Center Energy Retrofits. 52+46 pp. (Ph.D. Thesis)

A-2013-13 P. Pohjalainen: Self-Organizing Software Architectures. 114+71 pp. (Ph.D. Thesis)

A-2014-1 J. Korhonen: Graph and Hypergraph Decompositions for Exact Algorithms. 62+66 pp. (Ph.D. Thesis)

A-2014-2 J. Paalasmaa: Monitoring Sleep with Force Sensor Measurement. 59+47 pp. (Ph.D. Thesis)

A-2014-3 L. Langohr: Methods for Finding Interesting Nodes in Weighted Graphs. 70+54 pp. (Ph.D. Thesis)

A-2014-4 S. Bhattacharya: Continuous Context Inference on Mobile Platforms. 94+67 pp. (Ph.D. Thesis)

A-2014-5 E. Lagerspetz: Collaborative Mobile Energy Awareness. 60+46 pp. (Ph.D. Thesis)

A-2015-1 L. Wang: Content, Topology and Cooperation in In-network Caching. 190 pp. (Ph.D. Thesis)

A-2015-2 T. Niinimäki: Approximation Strategies for Structure Learning in Bayesian Networks. 64+93 pp. (Ph.D. Thesis)

A-2015-3 D. Kempa: Efficient Construction of Fundamental Data Structures in Large-Scale Text Indexing. 68+88 pp. (Ph.D. Thesis)

A-2015-4 K. Zhao: Understanding Urban Human Mobility for Network Applications. 62+46 pp. (Ph.D. Thesis)

A-2015-5 A. Laaksonen: Algorithms for Melody Search and Transcription. 36+54 pp. (Ph.D. Thesis)

A-2015-6 Y. Ding: Collaborative Traffic Offloading for Mobile Systems. 223 pp. (Ph.D. Thesis)

A-2015-7 F. Fagerholm: Software Developer Experience: Case Studies in Lean-Agile and Open Source Environments. 118+68 pp. (Ph.D. Thesis)