UNIVERSITAT
POMPEU FABRA

Identification of Versions of the Same Musical Composition by Processing Audio Descriptions

Joan Serrà Julià

TESI DOCTORAL UPF / 2011

Director de la tesi:

Dr. Xavier Serra i Casals
Dept. of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona, Spain

Dissertation submitted to the Deptartment of Information and Communication Technologies of Universitat Pompeu Fabra in partial fulfillment of the requirements for the degree of

DOCTOR PER LA UNIVERSITAT POMPEU FABRA,

with the mention of European Doctor.

Music Technology Group (http://mtg.upf.edu), Dept. of Information and Communication Technologies (http://www.upf.edu/dtic), Universitat Pompeu Fabra (http://www.upf.edu), Barcelona, Spain.

Als meus avis.

Acknowledgements

I remember I was quite shocked when, one of the very first times I went to the MTG, Perfecto Herrera suggested that I work on the automatic identification of versions of musical pieces. I had played versions (both amateur and professionally) since I was 13 but, although being familiar with many MIR tasks, I had never thought of version identification before. Furthermore, how could they (the MTG people) know that I played song versions? I don't think I had told them anything about this aspect...

Before that meeting with Perfe, I had discussed a few research topics with Xavier Serra and, after he gave me feedback on a number of research proposals I had, I decided to submit one related to the exploitation of the temporal information of music descriptors for music similarity. Therefore, when Perfe suggested the topic of version identification I initially thought that such a suggestion was not related to my proposal at all. However, subsequent meetings with Emilia Gómez and Pedro Cano made me realize that I was wrong, up to the point that if now I had to talk about the work in this thesis I would probably use some of the words of my original proposal: "temporal information", "music descriptors", and "music similarity".

Being in close contact with these people I have mentioned has been extremely important, not only for the work related to this thesis, but also for my education as a researcher in general (not to mention the personal side!). I am really happy to have met them. And I am specially grateful to Xavier for giving me the opportunity to join the MTG.

One day, while talking with Xavier, he mentioned a course on time series analysis given in the UPF by some guy called Ralph, who had quite an unpronounceable surname (Andrzejak). My research at that time was already pivoting around nonlinear time series analysis tools, so I managed to attend to Ralph's course and off-line told him about my research. This turned out to be the starting point of a very fruitful collaboration between Ralph and myself. I must confess I have learned A LOT from him.

Another day, at Ralph's office, I saw quite a deteriorated (by use) copy of a book by some guys called Kantz & Schreiber. Ralph told me that this was "the bible", so I bought it and started reading. It was himself who, after seeing that my Kantz & Schreiber book was nearly as deteriorated as his, suggested doing a research stay abroad. We decided to contact Holger Kantz and, to my surprise, he agreed on a collaboration. So I went to work at the MPIPKS for four months under Holger's supervision. That was a great experience!

Some time before, Pedro had invited Massimiliano Zanin to give a talk at the MTG. I do not remember if we had already had a short conversation at that time, but for the subsequent months he remained being just “the complex networks guy with very very long hair”, that is until I had some research problem related to complex networks. Then I contacted him and we started collaborating (and furthermore became friends). Now “the complex networks guy with very very long hair” has been substantially reduced to “Max”.

All the people I have mentioned are just a small part of the relevant interactions that have shaped this thesis. There are many more people from the MTG that I would like to acknowledge, and whose work, advice and frienship I really appreciate. These are Vincent Akkermans, Eduard Aylon, Dmitry Bogdanov, Jordi Bonada, Òscar Celma, Graham Coleman, Maarten de Boer, Ferdinand Fuhrmann, Jordi Funollet, Cristina Garrido, Enric Guaus, Salvador Gurrera, Martín Haro, Jordi Janer, Markus Koppenberger, Cyril Laurier, Oscar Mayor, Ricard Marxer, Owen Meyers, Hendrik Purwins, Gerard Roma, Justin Salamon, Mohamed Sordo, and Nicolas Wack (sorry if I am forgetting someone!). In addition, I have been in contact with people outside the MTG, specially with Josep Lluís Arcos, Juan Pablo Bello, Mathieu Lagrange, Matija Marolt, and Meinard Müller. I would also like to acknowledge Jean Arroyo for proofreading this thesis.

Last, but not least, I want to mention my friends and my family, who have supported me in all aspects.

This thesis has been carried out at the Music Technology Group of Universitat Pompeu Fabra (UPF) in Barcelona, Spain from Sep. 2007 to Jan. 2010 and from Jun. 2010 to Dec. 2010, and at the Max Planck Institute for the Physics of Complex Systems (MPIPKS) in Dresden, Germany from Feb. 2010 to May 2010. This work has been supported by an R+D+I scholarship from UPF, by the European Commission projects CANTATA (FIT-350205-2007-10), SALERO (IST-2007-0309BSCW) and PHAROS (IST-2006-045035), by the project of the Spanish Ministry of Industry, Tourism and Trade MUSIC 3.0 (TSI-070100-2008-318) and by the project of the Spanish Ministry of Science and Innovation DRIMS (TIN-2009-14247-C02-01). The research stay at the MPIPKS was funded by the German Academic Exchange Service (DAAD; A/09/96235) and the MPIPKS.

Abstract

Automatically making sense of digital information, and specially of music digital documents, is an important problem our modern society is facing. In fact, there are still many tasks that, although being easily performed by humans, cannot be effectively performed by a computer. In this work we focus on one of such tasks: the identification of musical piece versions (alternate renditions of the same musical composition like cover songs, live recordings, remixes, etc.). In particular, we adopt a computational approach solely based on the information provided by the audio signal. We propose a system for version identification that is robust to the main musical changes between versions, including timbre, tempo, key and structure changes. Such a system exploits nonlinear time series analysis tools and standard methods for quantitative music description, and it does not make use of a specific modeling strategy for data extracted from audio, i.e. it is a model-free system. We report remarkable accuracies for this system, both with our data and through an international evaluation framework. Indeed, according to this framework, our model-free approach achieves the highest accuracy among current version identification systems (up to the moment of writing this thesis). Model-based approaches are also investigated. For that we consider a number of linear and nonlinear time series models. We show that, although model-based approaches do not reach the highest accuracies, they present a number of advantages, specially with regard to computational complexity and parameter setting. In addition, we explore post-processing strategies for version identification systems, and show how unsupervised grouping algorithms allow the characterization and enhancement of the output of query-by-example systems such as the version identification ones. To this end, we build and study a complex network of versions and apply clustering and community detection algorithms. Overall, our work brings automatic version identification to an unprecedented stage where high accuracies are achieved and, at the same time, explores promising directions for future research. Although our steps are guided by the nature of the considered signals (music recordings) and the characteristics of the task at hand (version identification), we believe our methodology can be easily transferred to other contexts and domains.

Resum

Racionalitzar o donar significat de manera automàtica a la informació digital, especialment als documents digitals de música, és un problema important que la nostra societat moderna està afrontant. De fet, encara hi ha moltes tasques que, malgrat els humans les puguem fer fàcilment, encara no poden ser realitzades per un ordinador. En aquest treball ens centrem en una d'aquestes tasques: la identificació de versions musicals (interpretacions alternatives d'una mateixa composició de música tals com 'covers', enregistraments en directe, remixos, etc.). Basant-nos en un enfocamen computacional, i utilitzant únicament la informació que ens proporciona el senyal d'àudio, proposem un sistema per a la identificació de versions que és robust als principals canvis musicals que hi pot haver entre elles, incloent canvis en el timbre, el tempo, la tonalitat o l'estructura del tema. Aquest sistema explota eines per a l'anàlisi no linial de sèries temporals i mètodes estàndard per a la descripció quantitativa de la música. A més a més, no utilitza cap estratègia de modelat de les dades extretes de l'àudio; és un sistema 'lliure de model'. Amb aquest sistema obtenim molt bons resultats, tant amb les nostres dades com a través d'un entorn d'avaluació internacional. De fet, d'acord amb aquestes últimes avaluacions, el nostre sistema lliure de model obté a dia d'avui els millors resultats d'entre tots els sistemes avaluats. També investiguem sistemes basats en models. A tal efecte, considerem un seguit de models de sèries temporals, tant linials com no linials. D'aquesta manera veiem que, encara que els nostres sistemes basats en models no aconsegueixen els millors resultats, aquests presenten certs avantatges relatius a la complexitat computacional i a l'elecció de paràmetres. A més a més, també explorem algunes estratègies de post-processat per a sistemes d'identificació de versions. Concretament, evidenciem que algoritmes d'agrupament no supervisats permeten la caracterització i la millora dels resultats de sistemes que funcionen a través de 'preguntes per exemple', tals com els d'identificació de versions. Amb aquest objectiu construim i estudiem una xarxa complexa de versions i apliquem tècniques d'agrupament i de detecció de comunitats. En general, el nostre treball porta la identificació automàtica de versions a un estadi sense precedents on s'obtenen molt bons resultats i, al mateix temps, explora noves direccions de futur. Malgrat que els passos que seguim estan guiats per la natura dels senyals involucrats en el nostre problema (enregistraments musicals) i les característiques de la tasca que volem solucionar (identificació de versions), creiem que la nostra metodologia es pot transferir fàcilment a altres àmbits i contextos.

Preface

When this thesis started, there had been very few attempts to automatically identify musical piece versions from audio. A quick look at the literature review of this thesis for works done before 2007 corroborates this assertion. However, in the course of this thesis, many interesting studies have appeared, changing and shaping the task at hand. This thesis makes a valuable contribution with the compilation of all this specific literature.

Automatic version identification has rapidly evolved from a quite incipient topic to a well-established and partially solved one, from quite low accuracies to salient results. We are very proud to say that our work from 2007 to 2010, which is reported in this thesis, jointly with our preliminary work from 2006 to 2007, has been essential and key to such a rapid evolution of the topic, developing a leading role within our scientific community. At the same time we hope that our work will remain inspirational for forthcoming research in both related and unrelated scientific areas.

The outcomes of this research have been published in a number of international conferences, journals, and a book chapter. Some of these publications have been featured in the media. Our approaches have participated in several editions of an international evaluation campaign, obtaining the highest accuracies in each edition where we participated, and the highest accuracies among all editions up to the moment of writing this thesis. Furthermore, part of this research has been incorporated into a commercial media broadcast monitoring service, and the author has patented two of his inventions separately.

Contents

List of figures

List of tables

List of abbreviations and symbols

Abbreviations

Abbreviation	Description
AR	Autoregressive
CA	Clustering algorithm
CL	Complete linkage
CRP	Cross recurrence plot
DP	Dynamic programming
DTW	Dynamic time warping
FFT	Fast Fourier transform
HC	Harmonic change
HMM	Hidden Markov model
HPCP	Harmonic pitch class profile
IDF	Inverse document frequency
IR	Information retrieval
KM	K-medoids
MAP	Mean of average precisions
MC	Music collection
MIDI	Musical instrument digital interface
MIR	Music information retrieval
MIREX	Music information retrieval evaluation exchange
MLSS	Most likely sequence of states
MO	Modularity optimization
MST	Minimum spanning tree
NCD	Normalized compression distance
OTI	Optimal transposition index
PBFV	Polyphonic binary feature vector
PCP	Pitch class profile
PM	Proposed method
RBF	Radial basis function
RP	Recurrence plot
RQA	Recurrence quantification analysis
SL	Single linkage
STFT	Short-time Fourier transform
TAR	Threshold autoregressive

Abbreviation	Description
TC	Tonal centroid
TF	Term frequency
UPGMA	Group average linkage
WPGMA	Weighted average linkage

Mathematical symbols

General

Example	Symbol type	Description
$\mathcal{A}, \mathcal{B}, \mathcal{C}$	Calligraphy letters	Matrices, bidimensional arrays.
A, B, C	Uppercase letters	Single numbers: constants, fixed values, etc.
$\mathbf{a}, \mathbf{b}, \mathbf{c}$	Bold lowercase letters	Vectors, unidimensional arrays.
a, b, c	Lowercase letters	Single numbers: indices, variables, etc.

Specific

Symbol	Description
$\mathcal{A}$	Model coefficients' matrix
a	Model's coefficient. Element of $\mathcal{A}$
$\mathbf{b}$	Cluster center
b	Cluster center component. Element of $\mathbf{b}$
$\mathcal{C}$	Sequence of tonal centroids
$\bar{\mathcal{C}}$	Sequence of downsampled tonal centroids
C	Cardinality
$\mathbf{c}$	Tonal centroid. Element of $\mathcal{C}$
$\bar{\mathbf{c}}$	Downsampled tonal centroid. Element of $\bar{\mathcal{C}}$
c	Tonal centroid component. Element of $\mathbf{c}$
$\mathcal{D}$	Dissimilarity matrix
$\mathcal{D}'$	Symmetrized dissimilarity matrix
$\hat{\mathcal{D}}$	Refined dissimilarity matrix
d	Dissimilarity value. Element of $\mathcal{D}$
$\hat{d}$	Refined dissimilarity value. Element of $\hat{\mathcal{D}}$
d_{Th}	Distance threshold
F	F-measure
f_k	Frequency (in bins) of the k-th spectral peak
$\mathbf{g}$	Sequence of harmonic changes
$\bar{\mathbf{g}}$	Sequence of downsampled harmonic changes
g	Harmonic change. Element of $\mathbf{g}$

Symbol	Description
$\bar{g}$	Downsampled harmonic change. Element of $\bar{\mathbf{g}}$
$\mathcal{H}$	Sequence of pitch class profiles
$\breve{\mathcal{H}}$	Sequence of normalized pitch class profiles
$\bar{\mathcal{H}}$	Sequence of downsampled pitch class profiles
$\mathbf{h}$	Pitch class profile. Element of $\mathcal{H}$
$\breve{\mathbf{h}}$	Normalized pitch class profile. Element of $\breve{\mathcal{H}}$
$\bar{\mathbf{h}}$	Downsampled pitch class profile. Element of $\bar{\mathcal{H}}$
$\dot{\mathbf{h}}$	Transposed pitch class profile.
h	Pitch class magnitude. Element of $\mathbf{h}$
i	Index
j	Index
K	Number of clusters
k	Index
k'_{Th}	Ranking threshold
$\mathcal{L}$	Cumulative recurrence matrix
$L_{\max}$	Maximum value found in matrix $\mathcal{L}$
l	Cumulative recurrence value. Element of $\mathcal{L}$
M	Constant
m	Embedding dimension
N	Total number of descriptors in a time series
$\hat{N}$	Total number of descriptors in an embedded time series
N_{T}	Number of trials in a test
$N^{\mathrm{F-}}$	Number of false negatives
$N^{\mathrm{F+}}$	Number of false positives
$N^{\mathrm{T+}}$	Number of true positives
$N_{\triangledown}$	Number of complete triangles
$N_{\vee}$	Number of incomplete triangles
n	Index
$\mathbf{o}$	Transposition index array
$\mathring{\mathbf{o}}$	Sorted array of transposition indices
O	Number of (optimal) transposition indices
o	Magnitude of a transposition index
$\mathcal{P}$	Probability matrix (transition matrix)
P	Precision (version groups)
p	Probability. Element of $\mathcal{P}$
$\mathcal{Q}$	Cummulative recurrence matrix
$Q_{\max}$	Maximum value found in matrix $\mathcal{Q}$
$Q^{*}_{\max}$	Post-processed version of $Q_{\max}$
q	Cumulative recurrence value. Element of $\mathcal{Q}$
$\mathcal{R}$	Cross recurrence plot
R	Recall (version groups)
r	Recurrence. Element of $\mathcal{R}$

Symbol	Description
$\mathcal{S}$	Cumulative recurrence matrix
$S_{\max}$	Maximum value found in matrix $\mathcal{S}$
s	Cumulative recurrence value. Element of $\mathcal{S}$
t	Time step for predictions (prediction horizon)
U	Total number of songs
U_{N}	Number of (unrelated) added songs in a music collection
U_{S}	Number of version sets in a music collection
u	Song index
v	Song index
$2W$	Total number of elements of the windowing function
$\mathbf{w}$	Windowing function
w	Element of $\mathbf{w}$
$\mathcal{X}$	Time series of descriptors
$\hat{\mathcal{X}}$	Embedded time series of descriptors
$\mathbf{x}$	Descriptor. Element of $\mathcal{X}$
$\hat{\mathbf{x}}$	Embedded descriptor. Element of $\hat{\mathcal{X}}$
x	Descriptor component. Element of $\mathbf{x}$
$\hat{x}$	Embedded descriptor component
$\mathcal{Y}$	Spectrogram
Y	Total number of windows of the spectrogram
$\mathbf{y}$	Magnitude spectrum. Element of $\mathcal{Y}$
y	Magnitude of the spectrum. Element of $\mathbf{y}$
$y^{(f_k)}$	Magnitude of the k-th spectral peak
$\bar{y}^{(f_k)}$	Whitened magnitude of the k-th spectral peak
Z	Total number of samples of the audio signal
$\mathbf{z}$	Audio signal
z	Audio sample. Element of $\mathbf{z}$
$\alpha_{\mathrm{A}}, \alpha_{\mathrm{B}}$	Constants
β	Constant
γ_o, γ_e	Gap penalties
ϵ	Arbitrary distance
ε	Distance threshold
ζ	Support variable
η	Minimum number of neighbors
θ	Model's parameter
ι	Constant
κ	Percentage of nearest neighbors
λ	Embedding window
μ	Mean value
$\boldsymbol{\mu}$	Mean vector
ν	Averaging factor
ξ	Normalized mean squared error (prediction error)

Symbol	Description
ρ	Distance
ϱ	Objective function
σ	Variance
ς	Constant
τ	Time delay
υ	Logarithmic mapping function
χ	Random value
ψ	Precision (query-by-example)
$\bar{\psi}$	Average precision
$\left\langle\bar{\psi}\right\rangle$	Mean of average precision
ω	Cosine weighting function
$\varGamma$	Relevance function
Δ	Relative accuracy increase
Θ	Heaviside step function
$\varLambda$	Ranked list of candidates
ϕ	Radial basis function
Φ	Transformation matrix
$\varPhi$	Vector basis. Element of Φ
$\varOmega$	Set of nearest neighbors

CHAPTER 1
Introduction

1.1 Motivation

1.1.1 Automatic version detection

To relate and compare musical pieces is a very complex task. Musical pieces usually collapse multiple information sources (e.g. multiple instruments) and exhibit several degrees of inner structure (e.g. syntactic structure; Lerdahl & Jackendorff, 1983). Moreover, a number of complex multifaceted interactions can be established between pieces (e.g. concept-sharing; Zbikowski, 2002). However, in spite of such degrees of complexity, we humans are outstandingly good at performing certain musical judgments, some of them requiring very little conscious effort (Dowling & Harwood, 1985). A prominent example is the ability to assess whether or not two audio renditions correspond to the same underlying musical piece.

Think for instance in the song[1] "Happy birthday to you"[2]. If somebody sings its melody, even if some parts are out of tune, we can easily recognize this musical piece. This recognition ability is present in any listener, provided that he/she is familiar with the piece, and it could grow with increased exposure to music (Bailes, 2010; Dalla Bella et al., 2003). Moreover, this ability is not restricted to human beings. In particular, research has been conducted with whales (Frankel, 1998) and birds (Comins & Genter, 2010; Marler & Slabbekoorn, 2004), showing that certain species present comparable capabilities.

Neither is the recognition of a musical piece is restricted to a specific audio rendition. In fact, we group together variations of the same musical composition. This grouping is inherent in our music experiences and can be explained

[1] In this thesis we loosely employ the term *song* to refer to any rendition of a musical piece, independently of the fact of whether there is any singing or not. Strictly speaking, a song is "a piece of music for voice or voices, whether accompanied or unaccompanied, or the act or art of singing" (Chew et al., 2010).

[2] http://en.wikipedia.org/wiki/Happy_birthday_song (all Internet links were checked at the time of submission of this thesis).

in terms of categorization (Zbikowski, 2002). Returning to the example above, the fact of whether it is Marylin Monroe's or The Ramones' performance[3] does not prevent us identifying the "Happy birthday" song. Notice however that there are numerous objective differences between the two performances. The first one is sung 'a cappella', with a slow and varying tempo. The second one is rendered in punk style, including electric guitars, bass and drums, and has a fast and strict tempo. Despite these important differences, we are able to tell unequivocally that the two performances correspond to the same musical piece. In other words, we recognize that the two songs are *versions*. Furthermore, we group them under the category *versions of Happy Birthday*, where other performances of this particular musical piece may also be found (in the case of knowing more of them).

An interesting way to investigate version recognition is through computational resources. Even before Turing (1950), researchers had already been interested in determining whether a computer can *imitate* a human (Saygin et al., 2000). This question is an essential concept in artificial intelligence (Russell & Norvig, 2003). Indeed, relevant knowledge can be gained from such imitations, both with theoretical and practical consequences. Our research, framed in the context of machine listening and music computing (Polotti & Rocchesso, 2008) also follows this approach.

Think of a computer that could make decisions as a human would. In particular, imagine that you provide a computer with a pair of music items and it tells you if they are the same or not. Moreover, imagine that the two items do not correspond to the same interpretation, but to two different versions of the same underlying musical piece, such as our "Happy birthday" example. If we add the fact that the machine should perform such a judgment without any prior information of the music items, just by analyzing two audio waveforms at a time, we are facing quite a challenging task (Fig. 1.1). This thesis deals with such a task.

1.1.2 Music information retrieval

In what regards to research around music and computers, developments within the music information retrieval (MIR) community have a fundamental role. MIR is an interdisciplinary research field that aims at automatically understanding, describing, retrieving and organizing musical contents (Casey et al., 2008b; Downie, 2008; Lesaffre, 2005; Orio, 2006). In particular, the MIR community has invested much effort in automatically assessing *music similarity* from an audio content-based perspective (e.g. Berenzweig et al., 2004; Pampalk, 2006; Pohle et al., 2009; West & Lamere, 2007). Music similarity is a key feature for searching and organizing today's million-track digital music collec-

[3] Due to copyright issues we cannot provide a link to listen to music items. In case the reader may be interested in listening to the cited items we suggest searching for them by artist and title on the web, e.g. in YouTube (`http://www.youtube.com`).

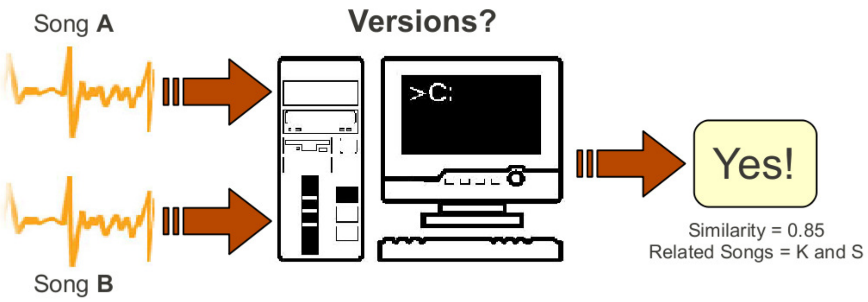


Figure 1.1: Illustration of automatic version detection from the audio signal.

tions (Pachet, 2005), and developing automatic ways to quantify it addresses part of a more general problem our modern society is facing: making sense of digital information (Ratzan, 2004).

Music similarity, however, is an ambiguous term. Apart from involving different musical facets such as timbre, tonality or rhythm, it also depends on cultural (or contextual) and personal (or subjective) aspects (Harwood, 1976; Lynch et al., 1990). There are many factors involved in music similarity judgments, and some of them, maybe the most relevant ones, are difficult to measure (Berenzweig et al., 2004). Therefore, it is not surprising that current efforts to develop a computational music similarity measure based on the audio content crash against the so-called "glass ceiling" (Aucouturier & Pachet, 2004). Indeed, average user scores[4] for such current approaches for music similarity do not surpass a value of 6 in a scale from 0 to 10.

To further proceed in assessing the similarity between music documents, some MIR researchers have devoted their efforts to the related task of *version identification*. Remarkably, and in contrast to music similarity, the relation between versions is context-independent and can be qualitatively defined and objectively measured. In addition, research on this task can yield valuable clues on how music similarity can be modeled. As Downie et al. (2008) indicate, considering the task of version identification "motivates MIR researchers to expand their notions of similarity beyond acoustic similarity to include the important idea that musical works retain their identity notwithstanding variations in style, genre, orchestration, rhythm or melodic ornamentation, etc".

1.2 Versions

1.2.1 Terms

In previously published work (e.g. Serrà et al., 2010a) we pragmatically used the term *cover songs* to refer to "different renditions of the same underlying

[4] http://www.music-ir.org/mirex/wiki/2010:Audio_Music_Similarity_and_Retrieval_Results

musical piece". This was motivated by the term's extended usage within the MIR community, including the MIR evaluation exchange (MIREX), an international initiative for the quantitative evaluation of MIR systems[5] (Downie, 2008; Downie et al., 2008).

One should note that, strictly speaking, the term cover song may carry a lot of ambiguities (Mosser, 2010). Many authors limit the term to popular music, in particular pop and rock genres, and to the period after 1950s (Coyle, 2002; Mosser, 2010; Solis, 2010; Weinstein, 1998; Witmer & Marks, 2010). In addition, they highlight its commercial, marketing and industrial connotations. Indeed, cover songs were originally part of a strategy to profit from 'hits' that had achieved significant commercial success. Record companies obtained important economic benefits by releasing alternative versions in other commercial or geographical areas without remunerating the original artist or label. Little promotion, different recording media and highly localized distribution in the middle of the 20th century favored these practices[6] (Plasketes, 2010; Weinstein, 1998; Witmer & Marks, 2010).

One may think about employing the term *variation*. Quoting the Grove Music Online, variation is a musical form "in which a discrete theme is repeated several or many times with various modifications" (Sisman, 2010). Although variation forms can be written as 'free-standing' pieces, the term commonly refers to the repetition of musical motifs within a piece. Moreover, in our view, the term has some restrictions with regard to music style (classical and contemporary music) and epoch (from 16th century on). To avoid any of these connotations we opt for not using it in this thesis.

Another term that is usually employed in this context is *plagiarism* (Posner, 2007). According to the online Merriam-Webster dictionary[7], plagiarizing implies "to steal and pass off (the ideas or words of another) as one's own" and also "to use (another's production) without crediting the source". With these definitions we can already see that the term clearly involves some sort of law infringement. Besides, plagiarism might be used in a provocative way. There are many artists who, without hiding the source, create art around the plagiarism concept by taking one or more existing audio recordings and altering them in some way to make a new composition. An example of this practice is found in the artist John Oswald and his project "Plunderphonics"[8] (Oswald, 1985). Anyway, the term plagiarism leaves out many renditions of music that do not conform to the above in its definition. Thus, in our opinion, plagiarism is an even more restrictive term than cover song or variation.

In this thesis, instead of cover songs, cover versions, plagiarisms or variations, we simply employ the term *versions*. We feel that this is a better way to de-

[5] We will introduce MIREX in more detail in Sec. 2.3.3.

[6] For additional information the reader may consult `http://en.wikipedia.org/wiki/Cover_version`

[7] `http://www.merriam-webster.com/dictionary/plagiarize`

[8] `http://en.wikipedia.org/wiki/Plunderphonics`

nominate the music material we consider for our experiments. Moreover, we think it is the best term to be associated with the motivations that drive our research (Sec. 1.1). With this term we aim to get rid of the economical, geographical, historical and social connotations outlined previously. In particular, we would like to stress that our research is not particularly focused nor biased to cover songs or plagiarisms.

We think about music versions as a term that globally encompasses *any* rendition or recording of the same musical piece, independently of the motivations for performing it, the historical period or whether it is sung or not. Reuse of music material has been a common practice for centuries, or even since the beginning of human history (Mithen, 2007). An example of an ancient reuse practice is the traditional Gregorian melody of “Dies Irae”, which has been used as a ‘musical quotation’ in requiems and a number of other classical compositions[9] (see Caldwell & Boyd, 2010, and references therein). In general, musicians can play versions simply as a homage or tribute to the original performer, composer or band. But there are many more reasons to play a version (c.f. Plasketes, 2010; Solis, 2010): to translate a song to another language, to adapt a musical piece to a particular country or regional tastes, to contemporize an old piece, to introduce a new artist, to parody or just for the simple pleasure of playing a familiar song. In addition, one must not forget that versions represent the opportunity for beginners and consolidated artists to perform a radically different interpretation of a musical piece, incorporating then a large amount of ‘creativity’ and ‘originality’.

Plasketes (2010) summarizes the last paragraph in one (long) sentence: “standardization, interpretation, incorporation, adaptation, appropriation and appreciation have been manifest in a multitude of musical manners and methods, including retrospectives and reissues, the emergence of rap and sampling as commercially dominant pop styles, karaoke, and a steady flow, if not stream, of cover compilations and tribute recordings which revisit a significant cross section of musical periods, styles, genre and artists and their catalogs of compositions”.

1.2.2 Types

Many distinctions between versions can be made. The majority of these come from musicology (e.g. Coyle, 2002; Mosser, 2010; Plasketes, 2010), although few have been made from an MIR perspective (Gómez, 2006; Tsai et al., 2008; Yang, 2001). In general, but specially true for the MIR-based ones, these distinctions aim at identifying different situations where a song was performed in the context of mainstream popular music. In this context, one can find a huge amount of tags, terms and labels related to versions, many of them being just buzzwords for commercial purposes.

[9] For a list the reader may consult `http://en.wikipedia.org/wiki/Dies_irae`

In Serrà et al. (2010a) we provided some examples of tags associated to versions, which we now briefly extend.

Remaster Creating a new master for an album or song generally implies some sort of sound enhancement to a previously existing product (e.g. compression, equalization, different endings or fade-outs).

Instrumental Sometimes, versions without any sung lyrics are released. These might include karaoke versions to sing or play along with, alternative versions for different record-buying public segments (e.g. classical versions of pop songs, children versions, etc.) or rare instrumental takes of a song in CD-box editions specially made for collectors.

Mashup It is a song or composition created by blending two or more prerecorded songs, usually by overlaying the vocal track of one song seamlessly over the instrumental track of another.

Live performance A recorded track from live performances. This can correspond to a live recording of the original artist who previously released the song in a studio album or to other performers.

Acoustic The piece is recorded with a different set of acoustical instruments in a more intimate situation. Sometimes "unplugged" is used as synonym.

Demo It is a way for musicians to approximate their ideas on tape or disc, and to provide an example of those ideas to record labels, producers or other artists. Musicians often use demos as quick sketches to share with band mates or arrangers. In other cases, a music publisher may need a simplified recording for publishing or copyright purposes, or a songwriter might make a demo in order to be sent to artists in the hope of having the song professionally recorded.

Standard In jazz music, there are compositions that are widely known, performed and recorded. Musicians usually maintain the main melodic and/or harmonic structure but adapt other musical characteristics to their convenience. There is no definitive list of jazz standards though this might change over time. Songs that can be considered standards may be found in the *fake book* (Kernfeld, 2006) or the *real book*[10] (Hal Leonard Corp., 2004).

Medley Mostly in live recordings, and in the hope of catching listeners' attention, a band performs a set of songs without stopping between them and linking several themes. Usually just the more memorable parts of the music work are included.

[10] See also http://www.myrealbook.com/home.htn or http://www.realbook.us

Remix This word can be very ambiguous. From a 'traditionalist' perspective, a remix implies an alternate master of a song, adding or subtracting elements or simply changing the equalization, dynamics, pitch, tempo, playing time or almost any other aspect of the various musical components. But some remixes involve substantial changes to the arrangement of a recorded work and barely resemble the original one. A remix may also refer to a re-interpretation of a given work such as a hybridizing process simultaneously combining fragments of two or more works.

Quotation The incorporation of a relatively brief segment of existing music in another work, in a manner akin to quotation in speech or literature. Quotation usually means melodic quotation, although the whole musical texture may be incorporated. The borrowed material is presented exactly or nearly so, but is not part of the main substance of the work. Incorporating samples of other songs into one's own song would fall into this category.

Of course all this terminology is defined in the context of (mainstream, commercial, popular) Western music. However, the near-duplicate repetition of musical items and phrases is a global phenomena. Each culture might label near-duplicate repetitions in a different manner and might apply different criteria to distinguish between them. For instance, in the Japanese culture there is a long and continuing tradition in *enka*, a sentimental ballad form that through patterned repetition derives authenticity over time (Yano, 2005). In general, one should be cautious in finding versions in other cultures because many misinterpretations could arise. For example, it would be misleading to consider two performances to be versions just because they are part of the same raga[11] (Bor, 2002; Daniélou, 1968).

1.2.3 Modifiable characteristics

According to our definition of the term version, we advocate a distinction based on musical characteristics instead of using geographical, commercial, subjective or situational tags like the ones above. The main musical characteristics that can change in a version are listed below. For completeness we also include an additional characteristic not strictly related to 'musical variations'. Noticeably, many of the listed characteristics may occur simultaneously in the same version.

1. Timbre: many variations changing the general color or texture of sounds might be included in this category. Two predominant groups are:

[11] Quoting Bor (2002), "a raga is far more precise and much richer than a scale or mode, and much less fixed than a particular tune". It can be regarded as a "tonal framework for composition and improvisation" that has "a particular scale and specific melodic movements".

   a) Production techniques: different sound recording and processing techniques introduce texture variations in the final audio rendition (e.g. equalization, microphones or dynamic compression).

   b) Instrumentation: the fact that the new performers can be using different instruments, configurations or recording procedures can confer different timbres to the version.

2. Tempo: as it is not as common to strictly control the tempo in a concert, this characteristic can change or fluctuate even in a live performance of a given song by its original artist. In fact, strictly following a predefined beat or tempo might become detrimental for expressiveness and contextual feedback. Even in classical music, small tempo fluctuations are introduced for different renditions of the same piece. In general, tempo changes abound, sometimes on purpose, with different performers.

3. Timing: in addition to tempo, the rhythmical structure of the piece might change depending on the performer's intention or feeling. Not only by means of changes in the drum section, but also including more subtle expressive deviations by means of swing, syncopation, accelerandos, ritardandos or pauses.

4. Structure: it is quite common to change the structure of the song. This modification can be as simple as skipping a short introduction, repeating the chorus where there was no such repetition, introducing an instrumental section or shortening one. On the other hand, such modifications can be very elaborated, usually implying a radical change in the musical section ordering.

5. Key: the piece can be transposed to a different key or main tonality. This is usually done to adapt the pitch range to a different singer or instrument, for aesthetic reasons or to induce some mood changes in the listener. Transposition is usually applied to the whole song, although it can be restricted just to a single musical section.

6. Harmonization: independently of the main key, the chord progression might change (e.g. adding or deleting chords, substituting them by relatives, modifying the chord types or adding tensions). The main melody might also change some note durations or pitches. Such changes are very common in introduction and bridge passages. Also, in instrumental solo parts, the lead instrument voice is practically always different from the original one.

7. Lyrics and language: one purpose for recording a version is to translate it to other languages. This is commonly done by high-selling artists to become better known in large speaker communities.

Tag	Timbre	Tempo	Timing	Struct.	Key	Harm.	Lyrics	Noise
Remaster	√							√
Instrumental	√						√	√
Mashup	√			√			√	√
Live	√	√	√					√
Acoustic	√	√	√		√	√		√
Demo	√	√	√	√	√	√	√	√
Standard	√	√	√	√	√	√	√	√
Medley	√	√	√	√	√			√
Remix	√	√	√	√	√	√	√	√
Quotation	√	√	√	√	√	√	√	√

Table 1.1: Musical changes that can be usually observed within different version tags. The '√' mark indicates that the change is possible, but not necessary.

8. Noise: in this category we consider other audio manifestations that might be present in a recording. Examples include audience manifestations such as claps, shouts or whistles, speech and audio compression and encoding artifacts.

We can of course relate music characteristics with the version-related 'types' or tags presented above (Table 1.1). In spite of the qualitative difference between both, music characteristics and version-related tags nowadays coexist. As an example, consider Beethoven's 5$^{\text{th}}$ symphony. If we randomly choose two classical music versions of it, we may see that one is tagged as, e.g. "instrumental" and "acoustic", while the other is only tagged as "live". However, none of these tags provide effective musical information for comparison. Indeed, when listening to such versions we may notice several musical variations (usually changes in instrument configurations, overall equalization, reverberation, tempo and loudness are noticeable). If we then listen, e.g. to the also "instrumental" Yngwie Malmsteen version, we will easily spot more changes (e.g. employing a full rock instrument set, a faster tempo, some structure changes, etc.). Finally, if we take a hip-hop remix by, e.g. 50 Cent, we may realize that nearly all original characteristics of the song are gone, except a lick or a phrase that is in the background. It is in this scenario where version identification becomes a very challenging task.

1.2.4 Social interest

'Versioning' is a phenomenon that clearly captures social attention. People have an increasing interest in versions of musical pieces, specially in versions of popular pieces. We can get an impression of this interest by having a look at the Internet. For instance, we can search for videos in YouTube that contain song version related terms. The result is a list of around 380000 videos[12], some of them having a play-count in the range of millions. These videos are not only from more or less consolidated artists, but also from amateurs and semi-professional bands.

If we perform the same search with Google we obtain around 3.5 million pages. These web pages range from comprehensive editorial or metadata collections (e.g. Second Hand Songs[13]) to social community portals where users can upload, listen and chat about their own song versions (e.g. Midomi[14]); from podcasts and radio programs (e.g. Coverville[15]) to news portals (e.g. BBC[16]); from personal blogs (e.g. Cover Me[17]) to research pages (e.g. LabROSA[18]). One of these web pages, Second Hand Songs, provides some statistics that, although being "heavily biased by the preferences of the editors and visitors"[13] (popular music, from 1950 on), give interesting indicators such as the "most covered songs", "most covered authors", "year statistics" or the "longest cover chain" (some of these indicators are highlighted in Table 1.2). To the present, their metadata collection contains "32009 works, 126427 performances, 2347 samples and 38629 artists (performers and songwriters)".

Social interest in versions is not only visible in the Internet. Song versions feature in many radio shows and even some of these shows are completely dedicated to them. Documentaries in musical television channels discuss or highlight different aspects of music versioning. Bands play versions in any kind of event: from weddings to big concerts. Amateur musicians perform versions. Indeed, nowadays easy access to music, instruments and recording techniques has greatly facilitated the repetition and modification of musical themes (Kotska, 2005), reaching a volume of version material that was unthinkable some decades ago.

1.2.5 Versions in other arts

The action of performing the same underlying 'production' despite numerous relevant changes in its characteristics is not restricted to the music nor the audio domains. Interestingly, we can straightforwardly draw some close analogies within other artistic domains. The most obvious domain where 'versions' are present is in literature (and, in general, in almost all kinds of writing activities). In fact, the term *quotation* we have introduced before is directly borrowed from there. Furthermore, if we think of a restatement of a text giving the meaning in another form, we talk about a *paraphrase*, another common practice in all

[12] The data was obtained on Sep. 13, 2010, by searching for ≪ "cover song" OR "cover songs" OR "cover version" OR "cover versions" OR "song version" OR "song versions" ≫.

[13] http://www.secondhandsongs.com

[14] http://www.midomi.com

[15] http://coverville.com

[16] http://news.bbc.co.uk/2/hi/7468837.stm

[17] http://www.covermesongs.com

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

“Most covered author”	John Lennon (3581), Paul McCartney (3416), [Traditional] (1980), Bob Dylan (1801), Ira Gershwin (1377), George Gershwin (1294), Richard Rodgers (1285), Cole Porter (1002), Burt Bacharach (964), Hal David (894), ...
“Most covered performer”	The Beatles (3541), Bob Dylan (1593), Elvis Presley (1005), Duke Ellington (782), The Rolling Stones (770), Hank Williams (757), The Ramones (730), David Bowie (533), Stevie Wonder (515), Chuck Berry (515), ...
“Most covering performer”	Johnny Mathis (327), Frank Sinatra (288), Elvis Presley (283), Ella Fitzgerald (281), Cliff Richard (267), Johnny Cash (229), Willie Nelson (225), Andy Williams (219), Tony Bennett (207), Jerry Lee Lewis (206), ...
“Most covered song”	Summertime (311), Body and soul (257), St. Louis Blues (207), Yesterday (184), Eleanor Rigby (160), Stille nacht! Heilige nacht! (156), Unchained melody (154), Silent night! Holly night! (146), Cry me a river (140), Over the rainbow (137), ...
“Cover year statistics”	Majority of originals performed from 1955 to 1985, majority of covers performed from 1985 to 2010.

Table 1.2: Indicators from Second Hand Songs at Dec. 9, 2010. The rank of elements in the table is the same as in the web.

kinds of writing. Also the notion of plagiarism is very present in written texts (Posner, 2007).

Specially relevant is the notion of *intertextuality* (Agger, 1999; Allen, 2000), which implies the shaping of texts' meanings by other texts. This practice is more or less clear in what could be considered old or ancient literature. A prominent example are popular stories. In many stories, the main theme can be kept while other contextual facets change (e.g. characters' features, action details or parts of the plot). These changes may be due to historical or geographical circumstances, or just due to the storyteller's taste. Another example can be found in the New Testament, where some passages quote from the Old Testament, and in Old Testament books such as Deuteronomy, where the prophets refer to the events described in the Exodus (Porter, 1997). Other more modern examples of intertextuality include[19] “East of Eden” (Steinbeck, 1952), which constitutes a retelling of the story of Genesis, set in the Salinas Valley of Northern California, or “Ulysses” (Joyce, 1918), a retelling of Homer's Odyssey set in Dublin.

Forms of intertextuality and ‘versioning’ are very present in painting, sculpture and photography. A portion of the history of both Eastern and Western visual art is dominated by motifs and ideas that reoccur, often with striking similarities. Religious paintings are examples of these recurrences. They range from artwork depicting mythological figures to Biblical scenes, scenes from the

[19] http://en.wikipedia.org/wiki/Intertextuality

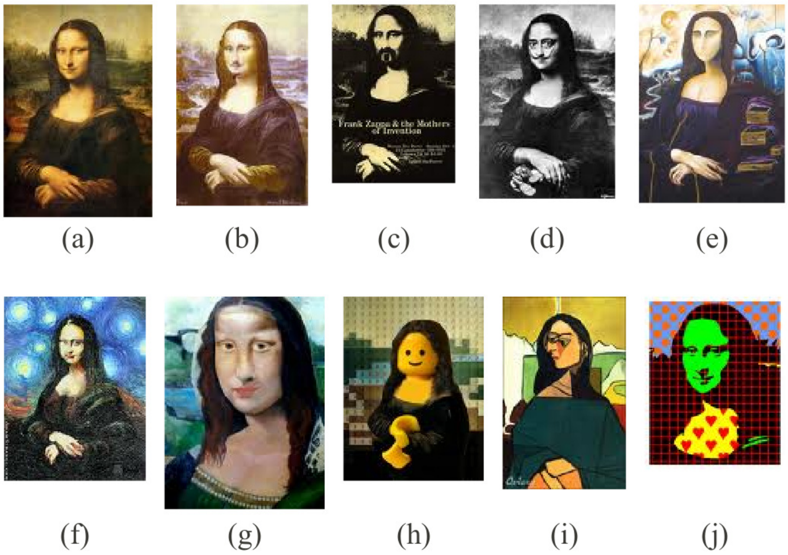


Figure 1.2: Examples of different versions of the “Mona Lisa” painting (see text).

life of Buddha or other scenes of Eastern religious origin.

Alternative renditions of existing paintings may be done as a homage, or motivated by important conceptual or technical changes. Furthermore, sometimes a painting may strongly influence other paintings. That would be the case of, for instance, “Las Meninas” (Velázquez, 1656), which has led to a number of ‘versions’ from the most famous artists, among them Picasso, who produced 44 interpretations of the painting[20]. Another example of a highly replicated painting is the “Mona Lisa” (Da Vinci, 1519). A simple search through the Internet can serve us to compile several renditions of it (Fig. 1.2). Some of them vary in small details (Fig. 1.2a-d), while others constitute a more radical reinterpretation of the picture (Fig. 1.2e-j). A few may even be a forgery or a parody (e.g. Fig. 1.2b-d,h).

Still in the visual domain, we find another avenue for versioning: movies. Of course here we find the obvious movie versions and remakes but, behind these, it is worth noticing that many movies make small ‘references’ to older movies. These references can be somewhat hidden or readily obvious, and reveal influences, imitations or restatements of other authors’ works. Importantly, these references can go beyond textual phrases[21]. Such is the case with entire sequences that remind the viewer of a previous film. These sequences are usually ‘versioned’ on purpose, even within current mainstream films. We can find some examples in many of Tarantino’s movies, where characters, scenes

[20] http://www.museupicasso.bcn.cat/meninas/index_en.htm

[21] For a compilation of quoted textual phrases see http://en.wikipedia.org/wiki/AFI%27s_100_Years%E2%80%A6100_Movie_Quotes

or frames are taken from other films that he considers inspiring. Another example would be the film “Wall-E”[22] (Stanton, 2008), which somehow reminds us of the film “Dumbo” (Disney, 1941) and which incorporates clear references to the musical “Hello Dolly!” (Merrick, 1964) or to the film “2001: A Space Odyssey” (Kubrick, 1968). Noticeably, this ‘sequence versioning’ is not solely done within movies. Just think about some episodes of “The Simpsons” series. To the best of our knowledge, existing technologies do not specifically address the problem of version identification within these ‘affine arts’. Song version is a very characteristic concept in music and therefore it is difficult to compare approaches from other arts. Nevertheless, one finds relevant works on authorship attribution and plagiarism detection, both with text (Juola, 2008; Stamatatos, 2009) and paintings (Hughes et al., 2010; Taylor et al., 2007). Further relevant research is found within automatic recognition of image objects and faces (Roth & Winter, 2008; Zhao et al., 2003) and movie sequences (Antani et al., 2002). In general, and roughly speaking, these approaches are conceptually similar to what could be applied to music versions: one tries to extract and compare features that are invariant towards common changes in the characteristics of the object of study (see Sec. 2.3).

1.3 Version identification: application scenarios

As mentioned, version identification can be directly exploited in a music retrieval scenario, where there is a need for searching and organizing musical pieces. One of the most basic paradigms of information retrieval, and by extension of music retrieval, is the query-by-example task: a user submits a reference query and the system returns a list of potential candidates that *match* the query. According to Casey et al. (2008b), we could talk about a “sense of match”, which implies different degrees of specificity. A match can be exact, retrieving candidates with specific musical content, or approximate, retrieving near neighbors in a musical space where proximity encodes different senses of music similarity. Following these directives, one could think of an imaginary “specificity axis” where music retrieval tasks with different match specificities can be placed, version identification being one of them (Fig. 1.3).

Currently, audio identification or fingerprinting techniques (Cano et al., 2005) are used to identify a particular recording with a high match specificity (exact duplicate detection). These techniques are applied in different contexts such as audio integrity verification or broadcast radio monitoring and tracking [see Cano et al. (2005) and references therein]. On the other side, we find e.g. the genre classification task (Scaringella et al., 2006), which corresponds to a low match specificity (category-based grouping). Version identification would be placed somewhere in the middle of the specificity axis (near duplicate detection, Fig. 1.3).

[22] `http://armchairc.blogspot.com/2010/04/walle.html`

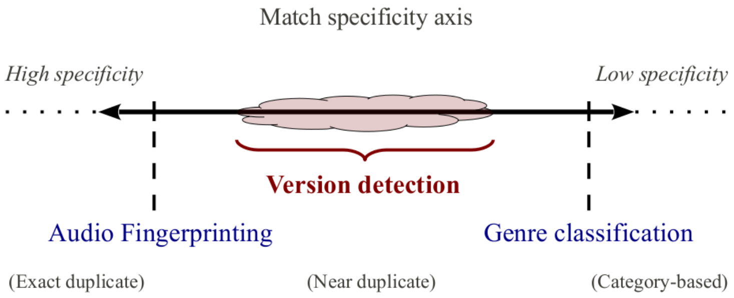


Figure 1.3: Picture of an hypothetical query match specificity scale.

We can see intuitively that both audio fingerprinting and category-based retrieval would fail to detect versions that incorporate some of the musical variations outlined above (Sec. 1.2.3). Thus version identification has its own application scenario. In addition, version identification systems have the potential to eventually replace and extend audio fingerprinting techniques by allowing less specificity in the match of music documents. At the same time, version identification systems represent a more specific retrieval that goes beyond genre or categorical associations. Furthermore, version identification can provide insights both in exact duplicate detection and category-based grouping (e.g. important musical aspects, new matching techniques or relevant algorithm features). One should bear in mind that such a specificity axis is not limited by strict boundaries: there is no well-defined point where something stops being a version and becomes a different piece of music.

Apart from the retrieval scenario, it may be readily apparent to the reader that algorithms for the automatic assessment of versions of musical pieces have direct implications to musical rights' management and licenses. For instance, a quantitative assessment of the similarity between two versions could be extremely helpful in court decisions with regard to music copyright infringement. To this extent, it is worth noting that lists of reference material are being collected and made public. For example, the Copyright Infringement Project[23] (Cronin, 2002) has the goal "to make universally available information about U.S. music copyright infringement cases from the mid-nineteenth century forward". Such ground truth could be used to train future systems on the specifics of plagiarism demands. Interestingly, and going further into some possible future applications, one could even think of a system assisting judges and juries in this aspect. The pioneering work by Müllensiefen & Pendzich (2009) suggests that court decisions can be predicted on the basis of statistically informed version similarity algorithms.

But not everything must be tied to commercial or economic purposes. Indeed, there exist more creative application contexts than the ones presented above.

[23] http://cip.law.ucla.edu

We can think for example of a musician who is composing a new piece. A version similarity algorithm could assess him on the originality of his ideas, providing a more informed compositional process. Musicologists can take advantage of such algorithms too. Automatic similarity measures could be used, among other things, to facilitate the analysis of related compositions, to trace the evolution of a musical piece, to establish relationships between performances, to compare passages or to quantify tempo deviations. From a simple user perspective, finding versions of a musical piece can be valuable and fun. This is easy to anticipate given the current interest in song versions (Sec. 1.2.4).

1.4 Objectives and outline of the thesis

The main goal of this thesis is to develop methods for automatically assessing whether two recordings are versions of the same musical piece. Our main starting point is the audio signal (e.g. an MP3 file), which we use as the unique source of information. Therefore most of the techniques we employ and propose are placed within the fields of signal processing and time series analysis. However, other techniques such as the ones derived from complex networks are also used. As general guidelines for our research we strive for simplicity, accuracy and generality. We focus overall on simple yet powerful approaches that can yield outstanding accuracies and that furthermore can be applied to signals and sources of a distinct nature. A further consideration with regard to the present work is that we aim at using unsupervised techniques, in the sense that no explicit learning is done on the basis of a pool of labeled examples.

In Chapter 2 we proceed with a comprehensive literature review focused on the specific topic of version identification. Since this topic is relatively new, we first position it within the wider context of MIR research. In particular, we place the task of version identification within both audio and symbolic music processing scenarios (Secs. 2.2.1 and 2.2.2). Some words about relevant research in music cognition are also given (Sec. 2.2.3). The remainder of the chapter is devoted to reviewing approaches specifically designed for version identification (Sec. 2.3). This review is organized around what we consider the main functional blocks of a version identification system (Sec. 2.3.1), which seek to tackle the aforementioned musical variations between song versions. Apart from functional blocks, we review some pre- and post-processing strategies for these systems (Sec. 2.3.2). The evaluation of version identification systems is also reviewed, with emphasis on the music material, the evaluation measures and the efforts to develop a common framework for the accuracy assessment of such systems (Sec. 2.3.3).

In Chapter 3 we present our main approach for version identification. We follow the major trend in the literature and devise a model-free approach, i.e. no strong assumptions are made about the nature of the signals involved in the process of identifying a version. The approach goes from the raw audio signal to a single measure reflecting version similarity. First, tonality-based descriptors are computed from audio using a state-of-the-art methodology (Sec. 3.2.2). Importantly, at this early stage we deal with timbre, noise and language invariance, three important characteristics that can change in versions (recall we have presented them previously in Sec. 1.2.3). Next, we propose a novel strategy for tackling different transpositions (Sec. 3.2.3). The two previous steps yield time series of music descriptors, which are then compared on a pairwise basis in order to obtain a version similarity measure. For that, nonlinear time series analysis concepts are employed. First, cross recurrences between a pair of songs are assessed in order to see which parts of the corresponding time series match (Sec. 3.2.5). Then, these cross recurrences are quantified (Sec. 3.2.6) and a dissimilarity measure is obtained (Sec. 3.2.7). These two stages specially focus on achieving structure, tempo as well as timing invariance. The approach is evaluated with a large in-house music collection and a common information retrieval methodology (Sec. 3.3). As a main result, we show that our approach yields a high accuracy with such a music collection (Sec. 3.4.2). This high accuracy is confirmed through an independent international evaluation framework allowing the comparison between existing approaches (Sec. 3.4.3).

Chapter 4 is devoted to post-processing stages for version identification systems. In particular, we explore the relation between songs that are inferred from such a system. To this end, we first study the network of version similarities obtained with our approach and show that different groups (clusters or communities) of songs are formed (Sec. 4.2.2). Such groups are detected in an unsupervised way (Sec. 4.2.3) and this information is subsequently exploited to enhance the accuracy of the original system (Sec. 4.2.4). Results prove the feasibility and effectiveness of this idea (Sec. 4.4). To close the chapter, we present a pioneer study on the role of the original song within its versions (Sec. 4.5). In particular, we show that the original song tends to occupy a central position within the group containing all possible versions of a musical piece.

In Chapter 5 we return to the development of dissimilarity measures for version identification. However, this time we take a radically different approach and explicitly model descriptor time series. More specifically, we study how common linear and nonlinear time series models can be used for the task at hand (Sec. 5.2.4). A prediction-based framework is proposed in order to obtain a suitable dissimilarity measure (Sec. 5.2.5). We base such a measure on the predictions of the models and evaluate them through a standard error measure (Sec. 5.2.6). Although the results for the model-based strategy are worse than the ones for the model-free strategy (Sec. 5.4), we show that such a model-based approach is very promising, specially with reference to computational costs and user parameter settings (Sec. 5.5). We also comment on further developments that could lead to a very competitive version identification system (Sec. 5.6).

Chapter 6 concludes this thesis. It provides a summary of contributions and discusses future perspectives for version identification.

CHAPTER 2

Literature review

2.1 Introduction

This literature review is divided into two main sections. The first briefly highlights the scientific background around automatic version detection. In particular, we focus on three areas of research: audio-based retrieval, symbolic music processing and music cognition. In audio-based retrieval, we place the task of version identification within music retrieval, focusing on audio content-based approaches. With the section on symbolic music processing we stress the importance of research done in the symbolic domain[1] and briefly discuss its applicability to the problem at hand. In the section devoted to music cognition we review relevant knowledge for version detection coming from this discipline. The second provides a comprehensive summary of version identification systems. The summary is based on a functional block decomposition of these systems. Apart from the core blocks, some pre- and post-processing strategies are relevant. We therefore give an outline of those that have been applied to version identification. Finally, the evaluation of version identification systems is discussed. In this second main section we only focus on methods that work in the audio domain and explicitly consider versions of musical pieces as primary music material. We furthermore restrict the review to methods specifically designed to achieve invariance to the characteristic musical changes among versions[2] (Sec. 1.2.3).

[1] As symbolic domain we refer to the approach to music content processing that uses, as starting raw data, symbolic representations of musical content (e.g. data extracted from printed scores). In contrast, the audio domain processes the raw audio signal (e.g. data from real-time recordings).

[2] Even considering these criteria, it is difficult to present the complete list of methods and alternatives. We apologize for any possible omissions/errors and, in any case, we assert that these have not been intentional.

2.2 Scientific background

2.2.1 Audio-based retrieval

Approaches for music retrieval can use multiple information sources, e.g. the raw audio signal, symbolic music representations, audio metadata, tags provided by users or experts or music and social networks data (Lesaffre, 2005; Orio, 2006). In the case of version identification, a metadata or tag-based approach would become trivial and would separate us from our initial motivation, namely that the computer 'hears' two musical pieces and determines if they are versions of the same composition[3]. Therefore, in our work we select an approach with the raw audio signal as its primary and only source of information.

In general, music retrieval is organized around use cases defined through the type of query, the sense of match and the form of the output (Casey et al., 2008b; Downie, 2008). In particular, in Sec. 1.3 we discussed that the sense of match implies different degrees of specificity and that version identification would be positioned somewhere in the middle of an hypothetical match specificity axis (near-duplicate detection, Fig 1.3). However, it must be noted that some systems that do not strictly focus on song versions approximate this intermediate match specificity region. This section provides a brief overview of these systems.

In audio content-based MIR, much effort has been focused on extracting information from the raw audio signal to represent certain musical aspects such as timbre, melody, main tonality, chords or tempo. This information is commonly called music description or *descriptors*. The computation of these descriptors is usually done in a short-time moving window either from a temporal, spectral or cepstral representation of the audio signal. The result is a descriptor time series (or sequence) reflecting the temporal evolution of a given musical aspect. The introduction and refinement of tonality descriptors, i.e. numeric quantities reflecting the tonal content of the signal, has broadened the match specificity of some music retrieval systems, specially those which can be placed near the two extremes of high and low match specificity. Indeed, a common extension of audio fingerprinting algorithms for achieving a lower match specificity consists of using tonal descriptors instead of the more routinely employed timbral ones[4] (e.g. Casey et al., 2008a; Miotto & Orio, 2008; Riley et al., 2008; Unal & Chew, 2007). The adoption of tonal descriptors adds an extra degree of timbre/noise invariance to audio fingerprinting algorithms, which are usually invariant with respect to song structure changes. Despite this, many of these fingerprinting algorithms may still have a low recall in a version identification task. One

[3] Furthermore, in the case of versions that completely change the title and the lyrics, there might be no clues to identifying them using only textual information.

[4] These approaches may also be termed audio identification, audio matching, or simply, polyphonic audio retrieval.

reason for this could be that, since these systems focus on retrieval speed, they usually employ some kind of descriptor quantization. This quantization may be excessively coarse for version identification (Riley et al., 2008). Another reason for a low version recall could come from the lack of invariance with respect to tempo variations or to key transpositions, which are frequent musical changes between song versions. The importance of these and other invariance characteristics in a version identification scenario may become evident through the thesis. Further evidence was shown as work prior to this document (Serrà et al., 2008b).

Like audio fingerprinting algorithms, many systems stemming from category-based grouping or from music similarity may also fall into the aforementioned region of intermediate match specificity. These systems, in general, differ from traditional systems of their kind in the sense that they also incorporate tonal information (e.g. Mardirossian & Chew, 2006; Pickens, 2004; Tzanetakis, 2002; Yu et al., 2008). However, they can fail in identifying recordings with a different key or with strong structure modifications. Furthermore, since these systems focus on timbre and this feature can radically change between versions (Sec. 1.2.3), wrong groupings could be made. In general, they do not consider full sequences of musical events, but just statistical summarizations of them, which might blur and distort valuable information for version retrieval.

2.2.2 Symbolic music processing

Although our focus is on the audio domain, one should note that relevant ideas for version identification can be also drawn from the symbolic domain. As symbolic domain we refer to the approach to music content processing that uses, as starting raw data, symbolic representations of musical content (e.g. MIDI[5] or **kern[6] files, which are data extracted from printed scores). Approaches using symbolic information are quite scattered among different disciplines. In particular, MIR researchers have proposed many quantitative approaches to symbolic similarity and retrieval. Good general resources are the works by Lemstrom (2000), Pickens (2004), Typke (2007) and Van Kranenburg (2010).

Of particular interest are query-by-humming systems (Dannenberg et al., 2007) and extensions of these to the polyphonic and to the audio domains (Pickens et al., 2003). In query-by-humming systems, the user sings or hums a melody and the system searches for matches in a musical database. Thus, this query-by-example situation is analogous to retrieving versions from a music collection without any other prior information. Another very active area of research is symbolic music similarity and matching (Grachten et al., 2005; Mäkinen et al., 2005; Rizo et al., 2009; Robine et al., 2007). Generally speaking, symbolic melodic similarity can be approached from very different points of view (Ur-

[5] http://www.midi.org

[6] http://wiki.humdrum.org

bano et al., 2010): some techniques are based on geometric representations of music, others rely on classic n-gram representations to calculate similarities and others use editing distances and alignment algorithms.

All these techniques are relevant for version identification. However, the kind of musical information that the systems above manage is symbolic (usually MIDI files). Therefore, if considering audio, the query, as well as the music material, must be transcribed into the symbolic domain. This would have the additional advantage of removing some expressive trends from the performer (c.f. Arcos et al., 1997; Juslin et al., 2002; Molina-Solana et al., 2010; Todd, 1992), thus potentially benefiting version detection systems. Unfortunately, transcription systems of this kind do not yet achieve a significantly high accuracy on real-world music signals. Current state-of-the-art algorithms for polyphonic transcription yield overall accuracies below 75%, and melody estimation approaches are within the same accuracy range[7]. Consequently, we argue that research in the symbolic domain cannot be directly applied to audio domain systems without incurring several estimation errors in the early processing stages of these systems. These errors, in turn, may have dramatic consequences in the final systems' accuracy.

2.2.3 Music cognition

Identification

The problem of version identification is also challenging from the point of view of music cognition, but apparently it has not attracted much attention by itself. Intuitively, in order to recognize versions, each individual needs to rely on some invariant representation of the whole song or, at least, its critical features. Currently we have little knowledge of which are the specific mechanisms that give rise to this level of abstraction.

One might hypothesize that abstract representations are grounded on physical neural templates that are shared across individuals (Schaefer et al., 2010). But we still do not know what is the essential information that our brains encode for solving this particular problem. Some knowledge has been gained about the relevance of melody statistics for music similarity (Eerola et al., 2001) and the sensitivity or insensitivity to certain melodic and rhythmic transformations (Dalla Bella et al., 2003; Kuusi, 2009; Schulkind et al., 2003). Timbre cues might provide important information, even from very short snippets of audio (Schellenberg et al., 1999), but recent studies with noise excerpts suggest that a rapid formation of auditory memories could be perfectly independent of timbre (Agus et al., 2010).

In this quest to know the essential information that is preserved, one might hypothesize that such 'essence' is not the same for all versions of a musical

[7] Recent results for these tasks can be found at the MIREX wiki: `http://www.music-ir.org/mirex/wiki/2010:MIREX2010_Results`

piece. From a perceptual or cognitive point of view, a musical work or song can be considered as a category (Zbikowski, 2002), one of the basic devices to represent knowledge either by humans or by machines (Rogers & McClelland, 2004). Usually, categories are taken to rely on features that are common to all items covered by them. Sometimes, a prototype for the whole category can be established (prototype-based categorization). This way, all members of the category can be compared against the prototype (Rosch & Mervis, 1975). However, we usually see that abstraction can still take place in the absence of a single common feature. This can be justified by the concept of *family resemblance* (Wittgenstein, 1953). The concept states that things which may be thought to be connected by one essential common feature may, in fact, be connected by a series of overlapping similarities. Therefore, in the end, it can easily happen that no one feature is common to all these connected entities. A widely used example is with family members[8]: all of them share some traits but maybe a common denominator does not exist.

Besides knowing which essential information to retain, there is the additional issue of the memory representation of songs in humans. It could either be the case that the similarity between two musical pieces is computed in their encoding step, or that all the songs are stored in memory and their similarity is computed at the retrieval phase. For example, Levitin (2007) discusses the possibility of absolute and detailed coding of song-specific information. On the other hand, Deliege (1996) discussed the possibility of encoding processes that abstract and group certain musical cues by similarity.

Furthermore, music is a sequential process, and as such, it poses the question of storage and retrieval of serial-order information in human working memory. And again we find some controversies, specially with regard to the use of absolute (hierarchically structured) or relative (associatively structured) position information. Two general theoretical frameworks exist: chaining models (Henson, 2001), which propose that individual items are coded in association with their preceding and/or succeeding elements, and ordinal position models (Conrad, 1965), which suggest that each individual item is coded by its absolute or relative position within a sequence.

Some insights from version identification

In general, version identification systems rarely pay attention to cognitive aspects (nor cognitive scientists pay attention to MIR systems). However, if one draws intuitive cross-domain analogies, some interesting reasonings can be made.

We find a first example with the essential information that we as humans need to encode in order to recognize a song. We have seen that studies on music cognition have put much emphasis on melodies. However, automatic version identification systems may use other tonal representations such as chords or

[8] Wittgenstein (1953) also used games as an example.

the so-called tonal profiles (see forthcoming Sec. 2.3.1). The fact that version identification systems are able to perform their task in a reliable manner suggests that the melody is not the only essential property to retain, and that other tonal representations as well could be useful for song recognition in the human brain.

A second example is found with regard to categorization aspects. If we consider a group of versions forming a category, family resemblance mechanisms may apply (in the sense of getting abstractions in the absence of a single common feature). However, from our point of view, some characteristic must be retained by all versions in the category. We believe that tonal sequences are so powerful that, in the case of song recognition, hardly any other boundary between *version groups* can be established. Therefore, in such a scenario where some feature is common to all items in the category, prototype-based categorization may take place. We provide evidence for that in Chapter 4 when we briefly study the relationships between versions and their originals.

A third example can be given with regard to the issue of memory representation. In this aspect, all version identification systems advocate the same: song representations are stored in memory and their similarities are computed at the retrieval stage. This might be due to pragmatic reasons, since similarity computation at the encoding step intuitively seems hard to implement.

Finally, with regard to absolute and relative encoding of sequential elements, we see that version identification systems use both strategies (Sec. 2.3.1). Importantly, by looking at version identification systems, the usage of these encodings seems to be independent of the song representation. Although one has to note that maybe the best performing systems are based on absolute encodings.

2.3 Version identification: state-of-the-art

2.3.1 Functional blocks

The standard approach to version identification is essentially to exploit the musical facets that are shared between multiple renditions of the same piece. We have seen that several important characteristics are subject to variation among versions: timbre, key, harmonization, tempo, timing, structure and so forth (Sec. 1.2.3). An ideal version identification system must be robust against these variations.

Usually, extracted music descriptors are in charge of overcoming the majority of musical changes outlined above. However, special emphasis is put on achieving tempo, key or structure invariance, as these are very frequent changes that are not usually managed by music descriptors themselves. Therefore, one can group the elements of existing version identification systems into five basic functional blocks (Fig. 2.1): descriptor extraction, key invariance, tempo invariance, structure invariance and similarity computation. We now elaborate

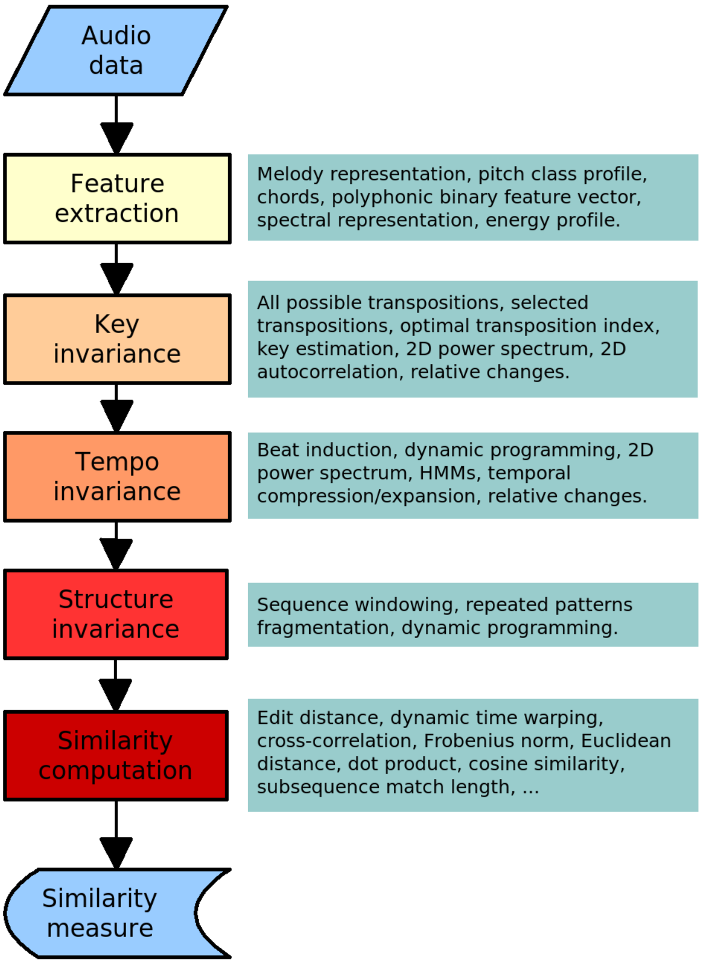

Audio data
Feature extraction
Melody representation, pitch class profile, chords, polyphonic binary feature vector, spectral representation, energy profile.
Key invariance
All possible transpositions, selected transpositions, optimal transposition index, key estimation, 2D power spectrum, 2D autocorrelation, relative changes.
Tempo invariance
Beat induction, dynamic programming, 2D power spectrum, HMMs, temporal compression/expansion, relative changes.
Structure invariance
Sequence windowing, repeated patterns fragmentation, dynamic programming.
Similarity computation
Edit distance, dynamic time warping, cross-correlation, Frobenius norm, Euclidean distance, dot product, cosine similarity, subsequence match length, ...
Similarity measure

Figure 2.1: Building blocks of a version identification system. The vertical arrows in the intermediate blocks do not necessarily imply the sequential application of these, except for the feature extraction and the similarity computation blocks, which are usually at the beginning and end of the chain, respectively.

on these blocks based on Serrà et al. (2010a). A summary table for several state-of-the-art approaches and the different strategies they follow in each functional block is provided at the end of the section (Table 2.1).

Descriptor extraction

In general, one assumes that versions of the same piece preserve the main melodic line and/or the harmonic progression, regardless of its main key. Therefore, tonal or harmonic content is the most employed characteristic in version identification. The term tonality is commonly used to denote a system of relationships between a series of pitches, which can form melodies and harmonies, having a tonic or central pitch class as its most important or stable element (Hyer, 2010). In its broadest possible sense, the term refers to the arrangements of pitch phenomena. Tonality is ubiquitous in Western music, and most listeners, whether musically trained or not, can identify the most stable pitch while listening to tonal music (Dalla Bella et al., 2003). Furthermore, this process is continuous and remains active throughout the sequential listening experience (Schulkind et al., 2003).

A tonal sequence can be understood, in a broad sense, as a sequentially-played series of different note combinations. These notes can be unique for each time slot (a melody) or can be played jointly with others (chord or harmonic progressions). That temporal and sequential information is important for retrieval is also evident in many other fields such as speech recognition (Nadeu et al., 2001) or string matching (Baeza-Yates & Perleberg, 1996). From an MIR point of view, clear evidence on the importance of tonal sequences for music similarity and retrieval exists (Casey & Slaney, 2006; Ellis et al., 2008; Hu et al., 2003). In fact, almost all version identification systems exploit tonal sequence representations extracted from the raw audio signals. More specifically, they either estimate the main melody, the chord sequence or the harmonic progression. Only what would be considered early version identification systems are an exception. For instance, Foote (2000a) worked with the audio signal's energy and Yang (2001) worked with spectral-based timbral features.

Melody is a salient musical descriptor of a piece of music (Selfridge-Field, 1998). Therefore, a number version identification systems use melody representations as a main descriptor (Marolt, 2006, 2008; Sailer & Dressler, 2006; Tsai et al., 2005, 2008). As a first processing step, these systems need to extract the predominant melody from the raw audio signal (Gómez et al., 2006b; Poliner et al., 2007). Melody extraction is strongly related to pitch perception and fundamental frequency tracking, both having a long and continuing history (De Cheveigne, 2005; De Cheveigne & Kawahara, 2001). However, in the context of complex mixtures, the perception and tracking issues become further complicated because, although multiple fundamental frequencies may be present at the same time, at most just one of them will be the melody. This and many other facets make melody extraction from real-world audio signals a difficult task.

To refine the obtained melody representation, version identification systems usually need to combine a melody extractor with, e.g. a singing voice detector, or other post-processing modules in order to achieve a more reliable representation (Sailer & Dressler, 2006; Tsai et al., 2005, 2008). Another possibility is to generate a so-called 'mid-level' representation for these melodies. The emphasis then is not only on melody extraction, but also on the feasibility to describe audio in a way that facilitates retrieval (Marolt, 2006, 2008). The level of abstraction (or smoothing) of a representation is an important issue that compromises the discriminatory power (see e.g. Grachten et al., 2004; Serrà et al., 2008b).

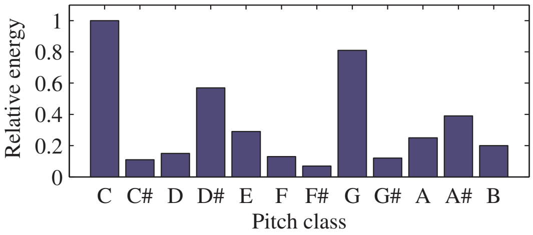


Figure 2.2: Example of a PCP descriptor. This may correspond to a C minor chord environment (it mostly contains C, D# and G pitch classes), where the root pitch class (C) is predominant.

Alternatively, version identification can be assessed by harmonic sequences, rather than melodic ones. Harmonic sequences, as they are nowadays estimated in MIR, might already incorporate melody information. The most straightforward way to carry out such an estimation is by means of so-called pitch class profiles (PCP) or chroma descriptors (Fujishima, 1999; Gómez, 2006; Leman, 1995; Purwins, 2005). These mid-level descriptors can provide a more complete, reliable and straightforward representation than melody estimation, as they do not need to tackle the pitch selection and tracking issues outlined above. PCP-based descriptors are widely used in the MIR community (Bartsch & Wakefield, 2005; Gómez & Herrera, 2004; Goto, 2006; Lee, 2008; Müller, 2007; Müller & Ewert, 2008; Ong, 2007; Sheh & Ellis, 2003).

PCP descriptors are derived from the energy found within a given frequency range (usually from 50 to 5000 Hz) in short-time spectral representations (typically 100 ms) of audio signals extracted on a frame-by-frame (or window) basis. This energy is usually collapsed into a 12-bin octave-independent histogram representing the relative intensity of each of the 12 semitones of an equal-tempered chromatic scale (the 12 pitch classes, Fig. 2.2). According to Gómez (2006), reliable PCP descriptors should, ideally, (a) represent the pitch class distribution of both monophonic and polyphonic signals, (b) consider the presence of harmonic frequencies, (c) be robust to noise and non-tonal sounds, (d) be independent of timbre and instruments played, (e) be independent of loudness and dynamics and (f) be independent of tuning, so that the reference frequency can be different from the standard A 440 Hz.

This degree of invariance with respect to several musical characteristics make PCP descriptors very attractive for version identification systems. Hence, the majority of systems use a PCP-based descriptor the primary source of information (Di Buccio et al., 2010; Egorov & Linetsky, 2008; Ellis & Cotton, 2007; Ellis & Poliner, 2007; Gómez & Herrera, 2006; Gómez et al., 2006a; Jensen et al., 2008a,b; Kim & Narayanan, 2008; Kim et al., 2008; Kim & Perelstein, 2007; Kurth & Müller, 2008; Müller et al., 2005; Nagano et al., 2002; Serrà et al., 2008b, 2010c, 2009a). Enhanced PCP information might also be considered, either with relative (or delta[9]) representations (Kim & Narayanan, 2008; Kim et al., 2008), or directly including multiple frame values in the analysis [e.g. the state space reconstruction in Serrà et al. (2010c, 2009a) that we will explain in the next chapter]. Distances between successive PCP vectors can also be considered, as well as adding information of the strongest pitch class (Ahonen, 2010).

An interesting variation of using raw PCP descriptors for characterizing the tonal content of song versions is proposed by Casey & Slaney (2006). In this work, PCP sequences are collapsed into symbol sequences using vector quantization, i.e. summarizing several PCP vectors by 8, 16, 32 or 64 representative symbols via the K-means algorithm (Xu & Wunsch II, 2009). Nagano et al. (2002) perform vector quantization by computing binary PCP vector components in such a way that, with 12 dimensional vectors, a codebook of $2^{12} = 4096$ symbols is generated (named polyphonic binary feature vectors). On the other hand, Di Buccio et al. (2010) use a hashing function of the rank of the elements in a PCP vector. Sometimes, the lack of interpretability of the produced sequences and symbols makes the addition of some musical knowledge to these systems rather difficult. This issue is addressed by Kurth & Müller (2008) who, instead of quantizing in a totally unsupervised way, generate a codebook of PCP descriptors based on musical knowledge (with a size of 793 symbols).

In general, vector quantization, indexing and hashing techniques result in highly efficient algorithms for music retrieval (e.g. Casey et al., 2008a; Di Buccio et al., 2010; Kurth & Müller, 2008; Nagano et al., 2002; Riley et al., 2008), even though their accuracy has never been formally assessed for the specific version identification task. It would be very interesting to see how these systems perform on a well-established benchmark collection in comparison to specifically designed approaches. More specifically, it is still an issue if PCP quantization strongly degrades version retrieval (see below). Some preliminary results suggest that this is the case (Riley et al. 2008; c.f. Di Buccio et al. 2010).

Depending on how we look at it, another form of PCP quantization consists of using chord or key template sequences (Ahonen & Lemstrom, 2008; Bello, 2007; Izmirli, 2005; Lee, 2006). Estimating chord sequences from audio data has been a very active research area in recent years (Bello & Pickens, 2005; Cho et al., 2010; Fujishima, 1999; Lee, 2008; Papadopoulos & Peeters, 2007; Sheh & Ellis, 2003). The common process for chord estimation consists of two steps: preprocessing the audio into a descriptor vector representation, usually a PCP, and approximating the most likely chord sequence from these vectors, usually done via template-matching or expectation-maximization trained hidden Markov models (Rabiner, 1989).

Usually, 12 major and 12 minor chords are used, although some studies incorporate more complex chord types, such as 7th, 9th, augmented and diminished

[9] By delta representations we mean the component-wise differences between consecutive descriptors.

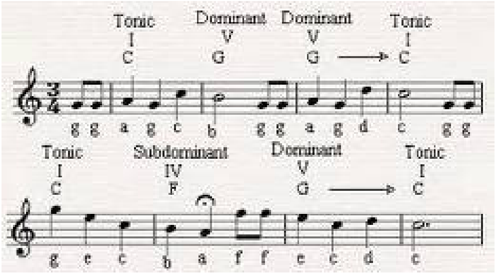


Figure 2.3: "Happy birthday" song score. Retrieved from http://www.piano-play-it.com.

chords (Fujishima, 1999; Harte & Sandler, 2005). This way, the obtained strings have a straightforward musical interpretation. Ahonen (2010) experiments with a 12-symbol representation, i.e. what would correspond to a 'power chord' representation[10]. He reports some accuracy increase with the addition of this reduced-symbol codebook to the standard 24-chord one.

In general, chord-based representations may be too coarse for version detection, and are also error-prone. Think for instance in the chord progression of the example we used in the previous chapter, the "Happy birthday" song (Fig. 2.3). There are just three chords, these being C, G and F (tonic, dominant and sub-dominant, respectively). If one makes a query with this specific chord progression, the answer would contain not only versions of "Happy birthday", but also lots of other songs that can be substantially different in terms of melody and arrangements. Thus, behind potential errors in their estimation, we conjecture that chord representations alone might be too ambiguous for version retrieval. Analogous reasonings may be derived for alternative 'tonal quantizations' in the case they do not use enough representative symbols.

Key invariance

As stated in Sec. 1.2.3, versions may be transposed to different keys. Transposed versions are equivalent to most listeners, as pitches are perceived relative to each other rather than in absolute categories (Dowling, 1978). Transposition to a common key has been elucidated as a very important feature for any version identification system, providing a deep impact on final system's accuracy [e.g. up to 17% difference in standard evaluation measures, depending on the method chosen, see Serrà et al. (2008a,b)]. In spite of being a common change between versions, some systems do not consider transposition. This is the case for systems that do not specifically focus on versions, or that do not use a tonal representation (Foote, 2000a; Izmirli, 2005; Müller et al., 2005; Yang, 2001).

[10] The so-called power chord is a chord with just its fundamental and the fifth, with possible multiple octaves of these pitches.

Several strategies can be followed to tackle transposition, and their suitability may depend on the chosen descriptor. In general, transposition invariance can be achieved by relative descriptor encoding, by key estimation, by shift-invariant transformations or by applying different transpositions. We now briefly comment on them.

The most straightforward way to achieve key invariance is to test all possible transpositions (Ellis & Cotton, 2007; Ellis & Poliner, 2007; Jensen et al., 2008a; Kim & Narayanan, 2008; Kim et al., 2008; Kurth & Müller, 2008; Marolt, 2008; Nagano et al., 2002). In the case of an octave-independent tonal representation, this implies the computation of a similarity measure for all possible circular or ring-shifts in the 'pitch axis' for each test song. This strategy usually guarantees a maximal retrieval accuracy (Serrà et al., 2008a) but, on the other hand, it increases the time and the size of the database to search in.

Instead of testing all possible transpositions, one can select certain 'preferred' transpositions (Ahonen, 2010; Egorov & Linetsky, 2008; Serrà et al., 2008b, 2010c, 2009a). This way, version identification approaches can be computationally faster. The trick consists in computing a sort of probability index for all possible relative transpositions and testing just those that are more likely to produce a good match. This technique corresponds to the so-called *optimal transposition indices* (Serrà et al., 2008a). The process for computing these indices is very fast, since a pre-computed global representation of the signal's tonal content is used (e.g. a simple averaging of the PCP features over the whole song, therefore reducing the whole PCP series to just a vector of numbers). Our results suggest that, for 12 bin PCP representations, a near-optimal accuracy can be reached with just two shifts, thus reducing the computational load by six (further details on this strategy are presented in the next chapter).

It should be mentioned that some systems do not follow the aforementioned strategy, although they predefine a certain number of transpositions to compute. In these cases, the number and the transpositions themselves are chosen either arbitrarily (Tsai et al., 2005, 2008), or based on some musical and empirical knowledge (Bello, 2007; Di Buccio et al., 2010). Decisions of this kind are very specific for each system and, most of all, for the specific descriptor being used.

A further approach is to off-line estimate the main key of the song and then apply transposition accordingly (Gómez & Herrera, 2006; Gómez et al., 2006a; Marolt, 2006). In this case, errors propagate faster and can dramatically worsen retrieval accuracy (e.g. if the key for the original song is not correctly estimated, no versions will be retrieved as they might have been estimated in the correct one). However, it must be noted that a similar procedure to choosing the most probable transpositions could be employed: one could compute an optimal *key* transposition index.

In the case of using a more symbolic representation such as chords or melodies, one can usually modify it in order to describe relative information changes, such as pitch or chord intervals (Ahonen & Lemstrom, 2008; Lee, 2006; Sailer &

Dressler, 2006). This way, a key-independent descriptor sequence is obtained. This idea, which is grounded in existing research on symbolic music processing (Sec. 2.2.1), has been recently extended to PCP sequences (Kim & Narayanan, 2008; Kim & Perelstein, 2007) by encoding such sequences using the optimal (or minimizing) transposition indices introduced above (see also Müller, 2007). A very interesting approach to achieving transposition invariance is to use a two-dimensional power spectrum (Marolt, 2008) or a two-dimensional autocorrelation function (Jensen et al., 2008b). Autocorrelation is a well-known operator for converting signals into a delay or shift-invariant representation (Oppenheim et al., 1999). Therefore, the power spectrum, which is formally defined as the Fourier transform of the autocorrelation, is also shift-invariant. As Marolt (2008) notes, other two-dimensional transforms could be also used, specially shift-invariant operators derived from higher-order spectra (Heikkila, 2004). Such transforms are very common in the image processing domain (Chandran et al., 1997; Klette & Zamperoni, 1996), and one can easily foresee a future usage of them in the audio domain.

Tempo invariance

Different renditions of the same piece may vary in the speed they have been played (Sec. 1.2.3), and any frame-based descriptor sequence will reflect this variation. For instance, in case of doubling the tempo, frames $i, i+1, i+2, i+3$ might correspond to frames $j, j, j+1, j+1$, respectively. As a consequence, extracted sequences cannot be directly compared.

Some version identification systems do not include a specific module to tackle tempo fluctuations (Ahonen, 2010; Ahonen & Lemstrom, 2008; Di Buccio et al., 2010; Kim & Narayanan, 2008; Kim et al., 2008; Yu et al., 2008). The majority of these systems generally focus on retrieval efficiency and treat descriptor sequences as statistical random variables. Thus, they discard much of the sequential information that a given representation can provide (e.g. a representation consisting of a 4-symbol pattern like ABABCD, would yield the same statistical values as AABBCD or ABCABD, which is indeed a misleading oversimplification of the original data).

A first option for achieving tempo invariance is again relative encoding. A symbolic descriptor sequence can be encoded by considering the ratio of durations between two consecutive notes (Sailer & Dressler, 2006). This strategy is commonly employed in query-by-humming systems (Dannenberg et al., 2007) and, combined with the relative pitch encoding of the previous section, leads to a representation that is key and tempo-independent. However, for the reasons outlined above, extracting a symbolic descriptor sequence is not straightforward and may lead to important estimation errors. Therefore, one needs to look at alternative tempo-invariance strategies.

Another way of achieving tempo invariance is to first estimate the tempo and then aggregate the information contained within comparable units of time. In this manner, the usual strategy is to estimate the beat (Gouyon et al., 2006) and then to aggregate the descriptor information corresponding to the same beat. This can be done independently of the descriptor used. Some version identification systems based on PCP descriptors (Ellis & Poliner, 2007; Nagano et al., 2002) or melody estimations (Marolt, 2006, 2008) use this strategy, and extensions with chords or other types of information could be easily worked out. If the beat does not provide enough temporal resolution, a finer representation might be employed (e.g. half-beat or quarter-beat; Ellis & Cotton, 2007). However, several studies suggest that systems using beat-averaging strategies can be outperformed by others (see below).

An alternative to beat induction is to do temporal compression and expansion (Kurth & Müller, 2008; Müller et al., 2005). This straightforward strategy consists of re-sampling the descriptor sequence into several musically plausible compressed and expanded versions, and then comparing all of them in order to discover the correct re-sampling empirically. Another interesting way to achieve tempo independence is again the two-dimensional power spectrum or the two-dimensional autocorrelation function (Jensen et al., 2008a,b; Marolt, 2008). These functions are usually designed for achieving both tempo as well as key independence (Sec. 2.3.1).

If one wants to perform direct comparisons of descriptors, a sequence alignment or similarity algorithm must be used to determine the correspondences between two distinct frame-based representations. Several alignment algorithms for MIR have been proposed (e.g. Adams et al., 2004; Dixon & Widmer, 2005; Grachten et al., 2004; Müller, 2007) which, sometimes, are derivations from general string and sequence alignment/similarity algorithms (Baeza-Yates & Perleberg, 1996; Gusfield, 1997; Rabiner & Juang, 1993; Sankoff & Kruskal, 1983).

In version identification, dynamic programming (Gusfield, 1997) is a routinely employed technique for aligning two representations and automatically discovering their local correspondences (Bello, 2007; Egorov & Linetsky, 2008; Foote, 2000a; Gómez & Herrera, 2006; Gómez et al., 2006a; Izmirli, 2005; Lee, 2006; Marolt, 2006; Nagano et al., 2002; Serrà et al., 2008b, 2009a; Tsai et al., 2005, 2008; Yang, 2001). Overall, one reiteratively constructs a cumulative distance matrix (Fig. 2.4) considering the optimal alignment paths that can be derived by following some neighboring constraints or patterns (Myers, 1980; Rabiner & Juang, 1993). These neighboring constraints determine the allowed local temporal deviations and they have been shown to be an important parameter in the system's final accuracy (Myers, 1980; Serrà et al., 2008b). One might hypothesize that this importance relies on the ability to track local timing variations between small parts of the performance.

A number of studies suggest that systems using dynamic programming can outperform those following a beat-averaging strategy (Bello, 2007; Liem & Hanjalic, 2009; Serrà et al., 2008b). The most typical algorithms for dynamic programming alignment and similarity are dynamic time warping algorithms

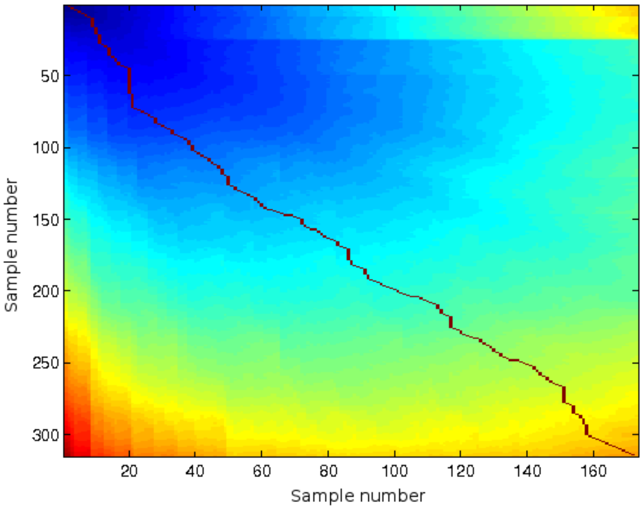


Figure 2.4: Example of a cumulative distance matrix, computed with dynamic programming, and its optimal alignment path.

(Rabiner & Juang, 1993; Sankoff & Kruskal, 1983) and edit distance variants (Gusfield, 1997). Their main drawback is that they are computationally expensive (quadratic in the length of the song representations), although several fast implementations may be derived (Gusfield, 1997; Mäkinen et al., 2005; Ukkonen et al., 2003).

Structure invariance

The difficulties that a different song structure may pose in the detection of musical piece versions are very often neglected. However, this has been demonstrated to be a key factor (Serrà et al., 2008b) and, in fact, recent version identification systems thoughtfully consider this aspect, specially many of the best-performing ones.

A classic approach to structure invariance consists in summarizing a song into its most repeated or representative parts (Gómez et al., 2006a; Marolt, 2006). In this case, song structure analysis is performed in order to segment sections from the song's representation used (Chai, 2005; Goto, 2006; Müller & Kurth, 2006b; Ong, 2007; Peeters, 2007). Usually, the most repetitive patterns are chosen and the remaining ones are disregarded. This strategy might be prone to errors since structure segmentation algorithms still leave much room for improvement (see references above). Furthermore, sometimes the most identifiable or salient segment of a musical piece is not the most repeated one, but the introduction, the bridge and so forth.

It must be noted that some dynamic programming algorithms are able to deal with song structure changes. These algorithms are basically the so-called local alignment algorithms (Gusfield, 1997). In particular, they have been successfully applied to the task of version identification (Egorov & Linetsky, 2008; Serrà et al., 2008b, 2009a; Yang, 2001). These algorithms solely consider the best[11] *sub*sequence alignment found between two tonal representations for similarity assessment, what has been evidenced to yield very satisfactory results (e.g. Serrà et al., 2008b). This is the approach followed in this thesis.

However, the most common strategy for achieving structure invariance consists of windowing the descriptors representation (so-called sequence windowing; Di Buccio et al., 2010; Kurth & Müller, 2008; Marolt, 2008; Müller et al., 2005; Nagano et al., 2002). The whole descriptor sequence is cut into short segments and the similarity measure is computed based on matches between these. Sequence windowing can be performed with a small hop size in order to faithfully represent any possible offset in the representations. However, this hop size has not been found to be a critical parameter for accuracy, as near-optimal values are found for a considerable hop size range (Marolt, 2008). Sequence windowing is also used by many audio fingerprinting algorithms using tonality-based descriptors (e.g. Casey et al., 2008a; Miotto & Orio, 2008; Riley et al., 2008).

Similarity computation

The final objective of a version identification system is, given a query, to retrieve a list of versions from a music collection. This list is usually ranked according to some similarity measure so that the topmost songs are the most similar to the query. Therefore, version identification systems output a similarity (or dissimilarity[12]) measure between pairs of songs. This similarity measure operates on the obtained representation after the main building blocks of feature extraction, key invariance, tempo invariance and structure invariance.

Common dynamic programming techniques used for achieving tempo invariance already provide a similarity measure as an output (Gusfield, 1997; Rabiner & Juang, 1993; Sankoff & Kruskal, 1983). Accordingly, the majority of systems following a dynamic programming approach use the similarity measure these methods provide. This is the case for systems using edit distances (Bello, 2007; Sailer & Dressler, 2006) or dynamic time warping algorithms (Foote, 2000a; Gómez & Herrera, 2006; Gómez et al., 2006a; Izmirli, 2005; Lee, 2006; Tsai et al., 2005, 2008). These similarity measures usually contain an implicit normalization depending on the lengths of the representations, which can generate some conflicts with versions of very different durations. In the case of the local alignment techniques, the similarity measure usually corresponds to the length

[11] By "best" we mean the longest most stable aligned subsequence.

[12] For the sake of generality, we use the term similarity to refer to both the similarity and the dissimilarity. In general, a distance measure can also be considered a dissimilarity measure, which, in turn, can be converted into a similarity measure.

of the found subsequence match (Egorov & Linetsky, 2008; Nagano et al., 2002; Serrà et al., 2008b, 2009a; Yang, 2001). This is the approach favored in this thesis, jointly with the new approach based on tonal sequence modeling (Serrà et al., 2010c). In the latter, a similarity measure is obtained by means of the prediction error made by a model trained on the query song when predicting the candidate song's tonal sequence.

Conventional similarity measures are also used, in particular cross-correlation (Ellis & Cotton, 2007; Ellis & Poliner, 2007; Marolt, 2006), the Frobenius norm (Jensen et al., 2008a), the Euclidean distance (Jensen et al., 2008b; Marolt, 2008), set intersection (Di Buccio et al., 2010) or the dot product (Kim & Narayanan, 2008; Kim et al., 2008; Kurth & Müller, 2008; Müller et al., 2005). These similarity measures are sometimes normalized depending on compared lengths of representations. In the case of adopting a sequence windowing strategy for dealing with structure changes, these similarity measures are usually combined with multiple subsequent steps such as threshold definition (Kurth & Müller, 2008; Marolt, 2008; Müller et al., 2005), TF-IDF[13] weights (Marolt, 2008), term pruning (Di Buccio et al., 2010) or mismatch ratios (Kurth & Müller, 2008). Less conventional similarity measures include the normalized compression distance (Ahonen, 2010; Ahonen & Lemstrom, 2008), and the hidden Markov model-based most likely sequence of states (Kim & Perelstein, 2007).

A summary table for several state-of-the-art approaches and the different strategies they follow in each functional block is provided in the next page (Table 2.1). A similar table with evaluation issues and results is given in the next section (Table 2.2).

2.3.2 Pre- and post-processing strategies

In Sec. 1.1.2 we mentioned the existence of a "glass ceiling" in the accuracy of music similarity approaches. However, the truth is that one can observe such phenomenon in many other MIR tasks (Downie, 2008). Depending on the task, several research directions can be considered for tackling this issue. Here we focus on the version detection task but, as noted in Lagrange & Serrà (2010), "most of the argumentation may be transferred to more general similarity tasks involving a query-by-example system".

One option to boost the accuracy of current query-by-example systems is to use an enhanced description of the musical stream using the segregation principle (Bregman, 1990). Intuitively, much can be gained if an audio signal is available for each instrument. This way, one can easily focus on the stream of interest for each MIR task. In this line, Foucard et al. (2010) show that considering a dominant melody removal algorithm as a pre-processing step is a promising

[13] The TF-IDF weight (term frequency-inverse document frequency) is a weight often used in information retrieval and text mining. For more details we refer to Baeza-Yates & Ribeiro-Neto (1999).

Reference(s)	Extracted feature	Key invariance	Tempo invariance	Structure invariance	Similarity computation
Foote (2000a)	Energy + Spectral		DP		DTW
Yang (2001)	Spectral		DP	Linearity filtering	Match length
Nagano et al. (2002)	PBFV	All transp.	Beat + DP	Seq. windowing + DP	Match length
Izmirli (2005)	Key templates		DP		DTW
Müller et al. (2005)	PCP		Temporal comp./exp.	Sequence windowing	Dot product
Tsai et al. (2005, 2008)	Melodic	K transp.	DP		DTW
Gómez & Herrera (2006)	PCP	Key estim.	DP		DTW
Gómez et al. (2006a)	PCP	Key estim.	DP	Repeated patterns	DTW
Lee (2006)	Chords	Key estim.	DP		DTW
Marolt (2006)	Melodic	Key estim.	DP	Repeated patterns	Cross-correlation
Sailer & Dressler (2006)	Melodic		Relative		Edit distance
Bello (2007)	Chords	K transp.	DP		Edit distance
Ellis & Cotton (2007); Ellis & Poliner (2007)	PCP	All transp.	Beat		Cross-correlation
Kim & Perelstein (2007)	PCP	Relative	HMM		MLSS
Ahonen & Lemstrom (2008)	Chords	Relative			NCD
Egorov & Linetsky (2008)	PCP	OTI	DP	DP	Match length
Jensen et al. (2008a)	PCP	All transp.	Fourier transform		Frobenius norm
Jensen et al. (2008b)	PCP	2D autocorr.	2D autocorrelation		Euclidean distance
Kim & Narayanan (2008); Kim et al. (2008)	PCP + Delta PCP	All transp.			Dot product
Kurth & Müller (2008)	PCP	All transp.	Temporal comp./exp.	Sequence windowing	Dot product
Marolt (2008)	Melodic	2D spectrum	Beat + 2D spectrum	Sequence windowing	Euclidean distance
Serrà et al. (2008b, 2009a)	PCP	OTIs	DP	DP	Match length
Ahonen (2010)	Chords + Other	OTI			NCD
Serrà et al. (2010c)	PCP	OTIs			Prediction error
Di Buccio et al. (2010)	PCP	K transp.		Sequence windowing	Set intersection

Table 2.1: Summary table. Version identification methods and their ways of overcoming changing musical characteristics. A blank space denotes no specific treatment for these changes. Abbreviations for extracted features are PBFV for polyphonic binary feature vector, and PCP for pitch class profile. Abbreviation for key invariance is OTI for optimal transposition index. Abbreviations for tempo invariance are DP for dynamic programming, and HMM for Hidden Markov Models. Abbreviations for similarity computation are DTW for dynamic time warping, MLSS for most likely sequence of states, and NCD for normalized compression distance.

approach for observing more robustly the harmonic progression and, in this way, achieve a higher accuracy in the version identification task. However, it may be a long time until such pre-processing based on segregation is beneficial for managing medium to large-scale music collections.

Related to the option of considering different streams is the consideration of different descriptors extracted from the same song. In this context, a first step has been done by Ahonen (2010). He extends the usual chord-based PCP quantization (24 symbols, Sec. 2.3.1) by including 'power chord' information (12 symbols), distances between successive PCP representations and the index of the strongest pitch class. The similarity measures obtained by these features separately are combined by averaging. This process may be computationally costly but shows some improvement in overall accuracy.

Regarding post-processing strategies, an efficient alternative is to consider approaches exploiting the regularities found in the results of a query-by-example system for a given music collection. Indeed, music collections are usually organized and structured at multiple levels. In the case of version detection, songs naturally cluster into so-called version sets[14] (Serrà et al., 2009b). Therefore, if those version sets can be approximately estimated, one can gain significant retrieval accuracy (Egorov & Linetsky, 2008; Serrà et al., 2010d, 2009b). A different and very interesting post-processing alternative is the general classification scheme proposed by Ravuri & Ellis (2010), where they employ the output of different version detection algorithms and a z-score normalization scheme to classify pairs of songs. In general, we believe that the combination of supervised and unsupervised methods could yield the most interesting approach for music retrieval (c.f. Baeza-Yates et al., 2006).

2.3.3 Evaluation

Music collection

A relevant issue when dealing with the evaluation of MIR systems is the music material considered. In the case of version identification, both the complexity of the problem and the selected approach largely depend on the studied music collection and the types of versions we want to identify. These might range from remastered tracks to radically different songs (Sec. 1.2.2). In this sense, it is very difficult to compare two systems evaluated in different conditions and designed to solve different problems.

Some works solely analyze classical music (Izmirli, 2005; Kim & Narayanan, 2008; Kim et al., 2008; Kurth & Müller, 2008; Müller et al., 2005), and it is the case that all of them obtain very high accuracies. However, classical music versions might not present strong timbral, structural or tempo variations. Therefore, one might hypothesize that, when only classical music is considered, the complexity of the version identification task decreases. Other works use a

[14] We originally termed it *cover sets* in Serrà et al. (2009b).

more variated style distribution in their music collections, but it is often still unclear which types of versions are used. These are usually mixed and may include remastered tracks (which might be easier to detect), medleys (where invariance towards song structure changes may be a central aspect), demos (with substantial variations with respect to the finally released song), remixes or quotations (which might constitute the most challenging scenario due to their potentially short duration and distorted harmonicity). In our view, a large variety in genres and version types is the only way to ensure the general applicability of the method being developed.

Besides the qualitative aspects of the music material considered, one should also take care with the quantitative aspects of it. The total number of songs and the distribution of these can strongly influence final accuracy values. To study this influence, one can decompose a music collection into version sets (i.e. each original song is assigned to a separate song set). Then, their cardinality (number of versions per set, i.e. the number of versions for each original song) becomes an important parameter.

In Serrà et al. (2010a) we performed a simple test with the system described in this thesis in order to assess the influence of these two parameters (number of version sets and their cardinality) on the system's final accuracy. Based on a collection of 2135 songs, 30 random selections of songs were carried out for a number of combinations of the previous two parameters. Then, an average for the mean average precision of all runs was computed and plotted (Fig. 2.5a). We can see that considering less than 50 version sets, or even just a cardinality of 2, yields unrealistically high results. Higher values for these two parameters at the same time all fall within a stable accuracy region[15]. This effect can also be seen if we plot the standard deviations of the evaluation measure across all runs (Fig. 2.5b). In particular, it can be observed that using less than 50 version sets introduces a high variability in the evaluated accuracy, which may then depend on the chosen subset. This variability becomes lower as the number of version sets and their cardinality increase.

With this small experiment we can see that an insufficient size or particular configurations of the music collection could potentially lead to abnormally high accuracies, as well as to parameter overfitting (in the case that the system required a training procedure). Unfortunately, many reported studies use less than 50 version sets (Foote, 2000a; Gómez & Herrera, 2006; Gómez et al., 2006a; Izmirli, 2005; Nagano et al., 2002; Tsai et al., 2005, 2008). Therefore, one cannot be confident about the reported accuracies. This could even happen with the so-called *covers80* dataset[16] (Ellis & Cotton, 2007), a freely available dataset composed of 80 version sets with a cardinality of 2 that many researchers use to test system's accuracy and to tune their parameters (Aho-

[15] It is not the aim of the experiment to provide explicit accuracy values. Instead, we aim at illustrating the effects that different configurations of the music collection might have for final system's accuracy.

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

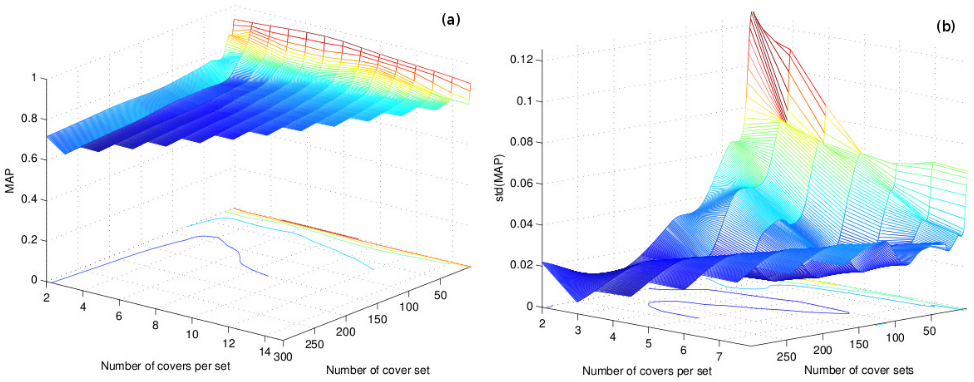


Figure 2.5: Mean accuracy (a) and accuracy variability (b) of a version identification system depending on the number of version sets, and the number of versions per set.

nen, 2010; Ahonen & Lemstrom, 2008; Ellis & Cotton, 2007; Ellis & Poliner, 2007; Jensen et al., 2008a,b).
When the music collection is not large enough, one may try to compensate the potential variability in final accuracies by adding so-called ‘noise’ or ‘control’ songs (Bello, 2007; Downie et al., 2008; Egorov & Linetsky, 2008; Marolt, 2006, 2008). The inclusion of these songs in the retrieval collection might provide an extra dose of difficulty to the task, as the probability of getting relevant items within the first ranked elements becomes then very low (numbers are given by Downie et al., 2008).

Evaluation measures

A further issue to be considered when evaluating the quantitative aspects of version identification is the evaluation measure to employ. In general, the quantitative evaluation of version identification systems is usually set up as a typical information retrieval ‘query and answer’ or query-by-example task, where one submits a query song and the system returns a ranked list of answers retrieved from a given collection (Baeza-Yates & Ribeiro-Neto, 1999; Voorhees & Harman, 2005). Therefore, several standard information retrieval measures have been employed for evaluating the accuracy of version identification systems: the R-precision (Bello, 2007; Izmirli, 2005), variants of precision or recall at different rank levels (Ellis & Cotton, 2007; Ellis & Poliner, 2007; Foote, 2000a; Jensen et al., 2008a,b; Kim & Narayanan, 2008; Kim et al., 2008; Kurth & Müller, 2008; Tsai et al., 2005, 2008; Yang, 2001), the average of precision and recall (Nagano et al., 2002), the F-measure (Gómez & Herrera, 2006; Gómez et al., 2006a; Serrà et al., 2008b) and the mean of average precisions (Ahonen, 2010; Ahonen & Lemstrom, 2008; Egorov & Linetsky, 2008; Serrà et al., 2010c, 2009a).
Since each of these evaluation measures focus on specific aspects of the retrieval

task (c.f. Serrà, 2007b), the quantitative comparison between systems of the same kind becomes difficult. In addition, the above measures only provide an overall accuracy for each system. A valuable improvement would be to implement independent evaluations for the different functional blocks outlined in this chapter, in order to analyze their contributions to the global system behavior.

MIREX

The only existing attempt to find a common methodology for the evaluation of MIR systems is the music information retrieval evaluation exchange (MIREX) initiative. MIREX is an international community-based framework for the formal evaluation of MIR systems and algorithms (Downie, 2008). Among other tasks, MIREX allows the comparison of different algorithms for artist identification, genre classification or music transcription[17].

Since 2006, MIREX allows for an objective assessment of the accuracy of different version identification algorithms (the so-called "audio cover song identification task"; Downie et al., 2008). For that purpose, participants can submit their algorithms and the MIREX organizers determine and publish the algorithms' accuracies and runtimes. The underlying music collections are never published or disclosed to the participants, either before or after the contest. Therefore, participants cannot tune their algorithms to the music collections used in the evaluation process.

The main MIREX test collection is composed of 30 version sets, each set being of cardinality 11. Accordingly, the total collection contains 330 songs. Another 670 individual songs, i.e. version sets of cardinality 1, are added to make the identification task more difficult. This music collection is meant to include "a wide variety of genres" (e.g. classical, jazz, gospel, rock, folk-rock), and a sufficient "variety of styles and orchestrations" (Downie et al., 2008). However, beyond this general description, no further information about the test collection is published or disclosed to the participants. In particular, only the MIREX organizers know what actual musical pieces are contained in the test collections. Since 2006, the same music collection has been used (so-called 'mixed collection').

For obtaining an accuracy value, each of the collection's versions are used as queries, and the submitted algorithms are required to return a distance matrix with one row for each query (i.e. a 330×1000 matrix must be returned for the 'mixed collection'). From this distance matrix, a number of evaluation measures are computed by the MIREX organizers. A number of evaluation measures have been computed for all editions of the MIREX version identification task (Downie et al., 2008): the total number of identified versions, the mean number of identified versions, the mean of maxima, the mean reciprocal

[17] http://www.music-ir.org/mirex/wiki/MIREX_HOME

rank and the mean of average precisions. Among all of these, the mean of average precisions is used as the principal accuracy measure for reporting results (Downie, 2008).

In 2009 a new music collection was introduced (the so-called 'mazurka collection'). This collection consists of 539 pieces corresponding to 11 selected versions from 49 Chopin mazurkas from the Mazurka Project[18]. Strictly speaking, this collection is a version compilation. However, the variability of the set is very reduced. All of them are Chopin's mazurkas, all of them are classical versions, and none of them present important variations with regard to song structure. Furthermore, the recordings that conform the entire Mazurka Project's collection are known. Therefore, one could overfit a system to it[19].

Overall, one might consider the mazurka collection as more of a "music identification" collection rather than a representative version collection. It is worth noting that all the systems submitted to MIREX that have been evaluated with the mazurka collection have achieved particularly high accuracies (e.g. the system we present in this thesis achieved a mean average precision of 0.96). With this view, high accuracies highlight the good performance that version identification algorithms can have in tasks such as music identification or audio fingerprinting (Sec. 1.3).

A summary table of the evaluation strategies and accuracies reported for the version identification systems outlined in the previous sections is shown in next page (Table 2.2).

[18] http://www.mazurka.org.uk

[19] However, the specific 539 pieces that are used for the MIREX evaluation are not known.

Reference(s)	Version sets	Card.	Total	Musical styles	Types of versions	Eval. meas.	Accuracy	MIREX MAP
Foote (2000a)	28		82	C, P	A, I, L,	P@3	0.80	
Yang (2001)			120	C, P, R		P@2	0.99	
Nagano et al. (2002)	8	27$^{\sim\wedge}$	216		L	Avg PR	0.89	
Izmirli (2005)	12		125	C		R-Prec	0.93	
Müller et al. (2005)			1167	C		P@15	0.93	
Tsai et al. (2005, 2008)	47	2	794			P@1	0.77	
Gómez & Herrera (2006)	30	3.1$^{\sim}$	90			Fmeas	0.39	
Gómez et al. (2006a)	30	3.1$^{\sim}$	90			Fmeas	0.41	
Lee (2006)								0.13
Marolt (2006)	8	4.5$^{\sim}$	1820	P, R		P@5	0.22	
Sailer & Dressler (2006)								0.07$^{\wedge}$
Bello (2007)	36	4.4$^{\sim}$	3208	P, R,	L,	R-Prec	0.25	0.27
Ellis & Cotton (2007); Ellis & Poliner (2007)	80	2	160	B, CO, M, P, R	A, DU, I, L	P@1	0.68	0.33
Kim & Perelstein (2007)								0.06
Ahonen & Lemstrom (2008)	80	2	160	B, CO, M, P, R	A, DU, I, L	MAP	0.18	
Egorov & Linetsky (2008)	30	11	1000	C, CO, E, HH, MT, P, R	A, DU, I, L, RR,	MAP	0.72	0.55
Jensen et al. (2008a)	80	2	160	B, CO, M, P, R	A, DU, I, L	P@1	0.38	0.24
Jensen et al. (2008b)	80	2	160	B, CO, M, P, R	A, DU, I, L	P@1	0.48	0.23
Kim & Narayanan (2008); Kim et al. (2008)	1000$^{\wedge}$	2$^{\sim}$	2000	C		P@1	0.79	
Kurth & Müller (2008)			1167	C		R@1	0.97	
Marolt (2008)	34	4.3$^{\sim}$	2424	P, R		MAP	0.40	
Serrà et al. (2008b, 2009a)	523	4.1$^{\sim}$	2125	B, C, CO, E, J, M, P, R, W	A, DE, DU, I, L, M, RX, Q	MAP	0.66	0.66
Ahonen (2010)	25	6	600	C, M, P, R, E, W	L, RX	MAP	0.41	
Serrà et al. (2010c)	523	4.1$^{\sim}$	2125	B, C, CO, E, J, M, P, R, W	A, DE, DU, I, L, M, RX, Q	MAP	0.44	
Di Buccio et al. (2010)	70	7.1$^{\sim}$	10000		L	MAP	0.32	0.15

Table 2.2: Summary table. Version identification methods and their evaluation strategies. Accuracies (including MIREX) correspond to best result achieved. Blank space, '$^{\wedge}$' and '$^{\sim}$' denote unknown, approximate and average values, respectively. Key for genres is (B) blues, (C) classical, (CO) country, (E) electronic, (J) jazz, (HH) hip-hop, (M) metal, (P) pop, (R) rock and (W) world. Key for types of versions is (A) acoustic, (DE) demo, (DU) duet, (I) instrumental, (L) live, (M) medley, (RR) remaster, (RX) remix and (Q) quotation. Key for evaluation measures is (MAP) mean of average precisions, (R-Prec) R-precision, (P@X) precision at rank X, (R@X) recall at rank X, (Fmeas) F-measure and (Avg PR) average precision-recall.

CHAPTER 3

Model-free version detection

3.1 Introduction

In our literature review we realize that, if there is a general shared characteristic among version identification systems, this is the lack of a model of the song or its descriptor sequences. This is specially true for the similarity computation stage, since all approaches simply try to 'match' or align data of some sort without making strong assumptions on the model that could generate or represent such data[1] (Sec. 2.3.1). In this chapter we propose an approach which also follows this outline, hence the "model-free" term in the title. We explore modeling strategies in Chapter 5.

Before we enter into the details of our model-free approach, it is convenient to contextualize the research done within this thesis, in particular with regard to our publications and our submissions to the MIREX "audio cover song identification task" (Sec. 2.3.3). In 2007, prior to the work of this thesis, we made a MIREX submission that we subsequently described in Serrà et al. (2008b). This 2007 algorithm, which used a specifically designed similarity measure for PCP features and a local alignment method, yielded the highest accuracy of all algorithms submitted in 2007 or in earlier editions. For the 2008 edition we submitted a qualitatively novel approach. The version identification measure that we derived from this approach ($Q_{\max}$) and a composition of this measure with a simple post-processing step ($Q^{*}_{\max}$) yielded the two highest accuracies of all algorithms submitted in 2008 or in earlier editions. In particular, the accuracy of both $Q_{\max}$ and $Q^{*}_{\max}$ clearly surpassed our earlier 2007 submission. Remarkably, the accuracies obtained by these two approaches remain the highest accuracies in the MIREX "audio cover song identification task" to date (this includes the 2008, 2009[2] and 2010 editions).

[1] The only exception is the approach by Kim & Perelstein (2007), who use hidden Markov models.

[2] In 2009 we resubmitted the same algorithm in order to see how it performed in a new test collection (Sec. 2.3.3). This new submission just included some minor software modifications and parameter adjustments, but the method remained exactly the same.

The work in this thesis started right after the MIREX 2007 submission[3]. Therefore $Q_{\max}$ and $Q^*_{\max}$ are direct products of this work. The remainder of the chapter provides a complete explanation of the $Q_{\max}$ measure and it is mostly based on our publications Serrà et al. (2008a) and Serrà et al. (2009a). Details about $Q^*_{\max}$, its post-processed version, are given in Chapter 4, which is mostly based on our publications Serrà et al. (2009b) and Serrà et al. (2010d).

The $Q_{\max}$ algorithm shares some pre-processing steps with the MIREX 2007 submission (Serrà et al., 2008b). However, the crucial difference is that it involves techniques derived from nonlinear time series analysis (Kantz & Schreiber, 2004). More specifically, $Q_{\max}$ is a recurrence quantification analysis (RQA; Marwan et al., 2002b; Webber Jr. & Zbilut, 1994; Zbilut & Webber Jr., 1992) measure that is extracted from cross recurrence plots (CRP; Zbilut et al., 1998), which are the bivariate generalization of classical recurrence plots (RP; Eckmann et al., 1987).

Repetition, or recurrence, is an important feature of music (Patel, 2008), and is also a key property of complex dynamical systems and of a wide variety of data series (Marwan et al., 2007). The framework of nonlinear time series analysis offers a number of techniques to quantify similarities and recurrences between signals measured from dynamical systems. Among these techniques, the CRP seems to be the most suitable to analyze pairs of music descriptor time series since it is defined for pairs of signals of different lengths and can easily cope with variations in the time scale and non-stationarities of the dynamics (Facchini et al., 2005; Marwan et al., 2002a). CRPs are constructed using delay coordinates (Takens, 1981), a tool routinely employed in nonlinear time series analysis (Kantz & Schreiber, 2004) that we will formally introduce in Sec. 3.2.4. For obtaining quantitative information of the structures present in a CRP, one uses RQA measures. These are actually measures of complexity that assess the number and duration of the recurrences (Marwan et al., 2007). Intuitively, when comparing two songs, we are specially interested in the duration of their shared recurrences.

CRPs and RQA measures are known as very intuitive and powerful tools in various disciplines such as astrophysics, earth sciences, engineering, biology, cardiology or neuroscience [see Marwan et al. (2007) and references therein]. However, to the best of our knowledge, there are no previous applications of CRPs and RQA measures to music-related signals. In general, only a few studies apply nonlinear time series analysis to these signals. Gerhard (1999) and Reiss & Sandler (2003) apply delay coordinates to raw audio signals with regard to audio analysis and visualization. Mierswa & Morik (2005) and Mörchen et al. (2006a,b) apply delay coordinates to music descriptor time series with regard to genre classification, user preferences and timbre modeling. Hegger et al. (2000) apply delay coordinates to human speech signals for the purpose of local projective noise reduction. Subsequently, Matassini et al. (2002) de-

[3] Specifically in September 2007. The MIREX submission was in August.

fined an RQA measure to automatically adjust the best neighborhood size for this local projection.

It should be noted that RPs and CRPs have certain analogies with commonly used MIR methods. In particular, it is worth recalling the so-called self similarity matrix, introduced by Foote (1999), to visualize music and audio tracks. It was later used by Foote (2000b) for song structure segmentation and by Casey & Westner (2000) for identifying components of an audio piece. Currently, self similarity matrices are used for diverse tasks such as song structure analysis (see references in Sec. 2.3.1) or music meter detection (Gainza, 2009). Cross similarity matrices are used, either directly or indirectly, in audio matching and synchronization algorithms (Müller, 2007), a task closely related to version identification (Sec 2.2.1). However, in contrast to CRPs, these similarity matrices do not apply any delay coordinate state space representation and are, in general, not thresholded. Although the quantification of structures in self or cross similarity matrices has received some attention from the MIR community (the references cited in this paragraph provide some examples), the usage of RQA measures as such is, to the best of the author's knowledge, unprecedented within the MIR literature.

On a more musical side, we can draw some analogies between the application of delay coordinates and the smoothing of self-similarity matrices in MIR. Delay coordinates allow us to bring together the information about both current and previous samples. In addition, by evaluating vectors of sample sequences, delay coordinates allow one to assess recurrences of systems more reliably than by using only the scalar samples (Marwan et al., 2007). Noticeably, the use of note sequences rather than isolated notes is essential in music (Huron, 2006) and is important for melody perception and recognition (Schulkind et al., 2003). Indeed, the concept of delay coordinates recalls some strategies that have been used in MIR[4]. In particular, the smoothing of self-similarity matrices along the main diagonal, sometimes referred to as the "incorporation of contextual information", has been used by Foote (2000b), Peeters et al. (2002), Peeters (2007), Bartsch & Wakefield (2005) and Müller & Kurth (2006a). In addition, Casey & Slaney (2006) discuss the importance of sequences, and use this fact in their "shingling" framework (Casey et al., 2008a). Evidence about the benefits of smoothing for self-similarity matrices has been reported within some MIR tasks other than version identification, in particular in the context of structure analysis (Müller & Kurth, 2006b) and partial music synchronization (Müller & Appelt, 2008). Notably, Müller & Appelt (2008) also applied some thresholding to their matrices.

[4] The author thanks M. Müller for providing insight on this aspect (M. Müller, personal communication, September 2010).

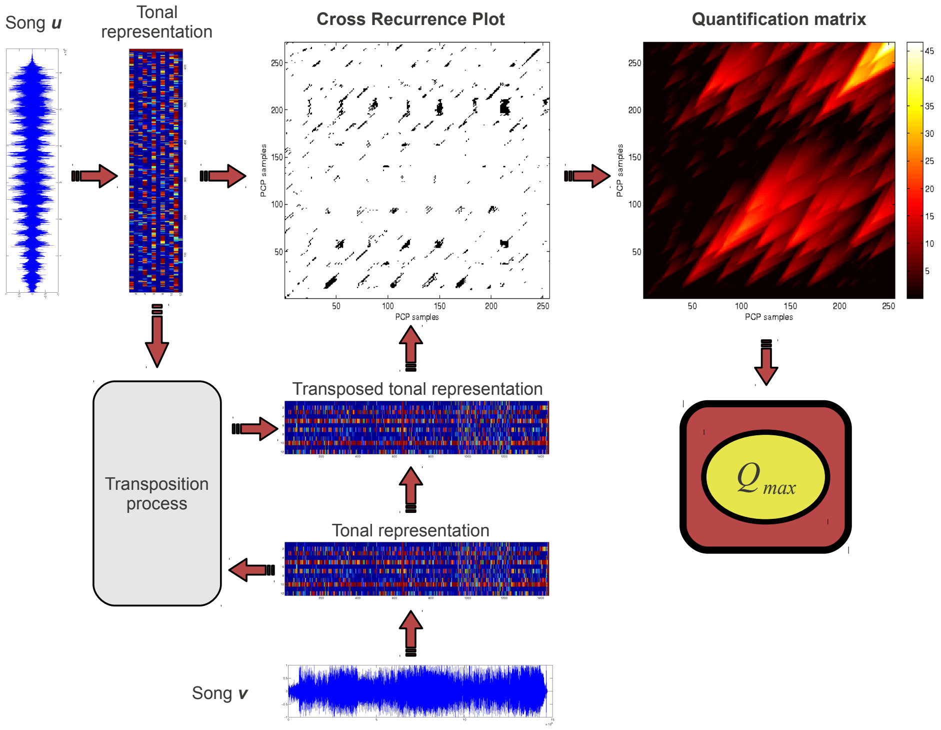


Figure 3.1: General block diagram of the $Q_{\max}$ approach.

3.2 Method

3.2.1 Overview

A brief overview of the $Q_{\max}$ algorithm and the resulting structure of this chapter can be outlined as follows (Fig. 3.1). Given two songs, we first extract descriptor time series (Sec. 3.2.2) and transpose one song to the main tonality of the other (Sec. 3.2.3). From this pair of multivariate time series, we form state space representations of the two songs using delay coordinates involving an embedding dimension m and time delay τ (Sec. 3.2.4). From this state space representation, we construct a CRP using a fixed maximum percentage of nearest neighbors κ (Sec. 3.2.5). Subsequently, we use $Q_{\max}$ to extract features that are sensitive to characteristics of song version CRPs, which results in two additional parameters γ_o and γ_e. In particular, we derive $Q_{\max}$ from a previously published RQA measure ($L_{\max}$; Eckmann et al., 1987), but adapt it in two steps (via $S_{\max}$) to the problem at hand (Sec. 3.2.6).

We evaluate our approach using a large collection of music recordings (Sec. 3.3.1) and a standard information retrieval evaluation measure (Sec. 3.3.2). We use a subset of this music collection to, first, study our transposition methodology, and then, to perform an in-sample optimization of parameters m, τ, κ, γ_o and γ_e (Sec. 3.4.1). We subsequently report the out-of-sample accuracy with optimized parameters of $L_{\max}$, $S_{\max}$ and $Q_{\max}$ (Sec. 3.4.2). All these steps were carried out before we submitted the resulting algorithm to MIREX as a further out-of-sample validation. We review results of all MIREX editions to date (Sec. 3.4.3) and provide an error analysis of our system (Sec. 3.4.5) before we draw some conclusions on the work presented in this chapter (Sec. 3.5).

3.2.2 Descriptor extraction

Pitch class profiles

Tonal information, and specially tonal hierarchies, are at the basis of human musical conception (Krumhansl, 1990; Lerdahl, 2001). Moreover, there is evidence that tonal hierarchies are primarily involved in important tasks related to music understanding such as music prediction, memorization or interpretation (Huron, 2006). Therefore, it seems reasonable to think that tonal information is one of the characteristics (if not the only one) that remains more or less invariant among different versions. The majority of version detection systems extract quantitative information related to this musical characteristic (Sec. 2.3.1).

In order to exploit the tonal information that is present in the audio signal we use pitch class profile (PCP) descriptors. In general, PCPs are robust against non-tonal components (e.g. ambient noise or percussive sounds) and independent of timbre and the specific instruments used (Gómez, 2006). Furthermore, they are usually independent of the loudness fluctuations in a musical piece.

PCPs are derived from the frequency dependent energy in a given range (typically from 50 to 5000 Hz) in short-time spectral representations (e.g. 100 ms) of audio signals computed in a moving window. This energy is usually mapped into an octave-independent histogram representing the relative intensity of each of the 12 semitones of the Western music chromatic scale (12 pitch classes). To normalize with respect to loudness, this histogram can be divided by its maximum value, thus leading to values between 0 and 1, or by the sum of its elements, thus leading to probability values for each pitch. A PCP example has been depicted in Fig. 2.2.

In our method we use an enhanced PCP descriptor: the so-called harmonic pitch class profile (HPCP; Gómez, 2006). HPCPs share the aforementioned PCP properties, but are based only on the peaks of the spectrum within a certain frequency band, thereby diminishing the influence of noisy spectral components. Furthermore, HPCPs are tuning independent, so that the reference tone can be different from the standard tone A at 440 Hz. In addition, they take into account the presence of harmonic frequencies. A general block diagram of the HPCP extraction process is provided in Fig. 3.2.

We now explain the process for obtaining HPCP descriptors, although some

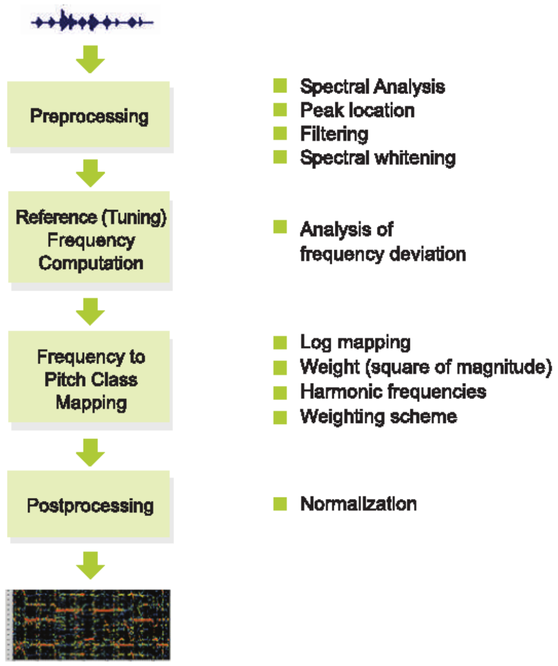


Figure 3.2: Basic block diagram for HPCP computation.

details are just summarized for the sake of brevity. For further information we refer to Gómez (2006) and the citations within the text. For an additional, more technical reference the reader may consult Gómez et al. (2008).

The employed HPCP extraction starts by converting each audio signal to a mono signal with a sampling rate of 44100 Hz. Stereo to mono conversion is done through channel averaging. We proceed with a moving window analysis and compute a spectrogram by means of the short-time Fourier transform (STFT; Smith III, 2010b). Let vector $\mathbf{z} = [z_1, \dots z_Z]^{\mathrm{T}}$ be the raw audio signal containing Z samples (since $^{\mathrm{T}}$ denotes transposition, $\mathbf{z}$ is a column vector). Then the spectrogram $\mathcal{Y} = [\mathbf{y}_1 \cdots \mathbf{y}_Y]^{\mathrm{T}}$ is obtained by means of the fast Fourier transform (FFT; Smith III, 2010a). For successive windows with 75% overlap, a magnitude spectrum $\mathbf{y}_i$ is calculated as

$$y_{i,k} = \left| \sum_{n=1}^{2W} z_{(i-1)\frac{W}{2}+n} w_n e^{-j\pi(k-1)\frac{n-1}{W}} \right| \tag{3.1}$$

for $k = 1, \dots W$, where $\mathbf{w} = [w_1, \dots w_{2W}]^{\mathrm{T}}$ is a windowing function of length $2W$ and j here corresponds to the imaginary number. For $\mathbf{w}$ we use a 92 dB Blackman-Harris function (Smith III, 2010b) and $2W = 4096$ (i.e. 93 ms). The last window is discarded due to possible insufficient length, therefore $Y = \lfloor 2Z/W \rfloor$ (recall that we use a 75% overlap, therefore the hop size in samples is $W/2 = 1024$). Notice that Eq. (3.1) takes the magnitude of the result of the FFT and that therefore discards phase information. Spectrum symmetries are also discarded ($k = 1, \dots W$).

Once the spectrogram $\mathcal{Y}$ is computed, a peak detection process is applied, i.e. the local maxima of all spectra $\mathbf{y}_i$ are extracted. The same procedure of parabola fitting used in sinusoidal modeling synthesis is followed (Serra, 1997) and only the 30 highest peaks found between 40 and 5000 Hz are taken. We indicate these peaks as $y_i^{(f_k)}$, f_k being the frequency of the k-th peak found in the i-th window.

With all $y_i^{(f_k)}$ for $i = 1, \dots Y$ and $k = 1, \dots 30$, a reference tuning frequency f_{ref} is computed for the whole song. First, a reference frequency is estimated for each window i by analyzing the deviations of $y_i^{(f_k)}$, for $k = 1, \dots 30$, with respect to the frequencies of an equal-tempered chromatic scale with A4 tuned at 440 Hz. Then, a histogram incorporating the deviations found in all windows $i = 1, \dots Y$ is used to estimate f_{ref}. Our approach is the same as the one employed by Gómez (2006).

An important part of the HPCP extraction is the spectral whitening process applied to each peak $y_i^{(f_k)}$. In particular, each $y_i^{(f_k)}$ value is normalized with respect to the corresponding value of the i-th spectral envelope at frequency f_k. The spectral envelope represents a crude approximation of the timbre information. Therefore, with such a timbre normalization, notes on high octaves contribute equally to the final HPCP vector as those on the low pitch range. This way, one gains robustness to different instrument configurations and equalization procedures (Gómez, 2006). To estimate a spectral envelope we use the same approach as Röbel & Rodet (2005).

After obtaining whitened peak magnitudes $\bar{y}_i^{(f_k)}$, we add their contributions to an octave-independent histogram $\mathbf{h}_i$ representing the relative intensity of the 12 semitones of the Western chromatic scale. Not only the contributions from peak values $\bar{y}_i^{(f_k)}$ are considered, but also the contributions of the frequencies having f_k as harmonic frequency. Apart from f_k, we consider 7 such frequencies, i.e. $f_n = f_k, f_k/2, \dots f_k/8$.

Developing from Gómez (2006), and considering the aforementioned 30 peaks, 12 semitones and 8 harmonics, the computation of an HPCP vector $\mathbf{h}_i = [h_{i,1}, \dots h_{i,12}]^{\mathrm{T}}$ can be expressed as

$$h_{i,j} = \sum_{k=1}^{30} \sum_{n=1}^{8} {\alpha_A}^{n-1} \left[\omega\left(j, \frac{f_k}{n}\right) \bar{y}_i^{(f_k)} \right]^2, \tag{3.2}$$

where α_A is a constant, ${\alpha_A}^{n-1}$ is a harmonic weighting term, and $\omega\left(j, f_n\right)$ is a cosine weighting function such that

$$\omega\left(j, f_n\right) = \begin{cases} \cos\left(\frac{\pi}{2} \frac{\upsilon(j,f_n)}{\alpha_B}\right) & \text{if } \left|\upsilon\left(j, f_n\right)\right| \leq \alpha_B, \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

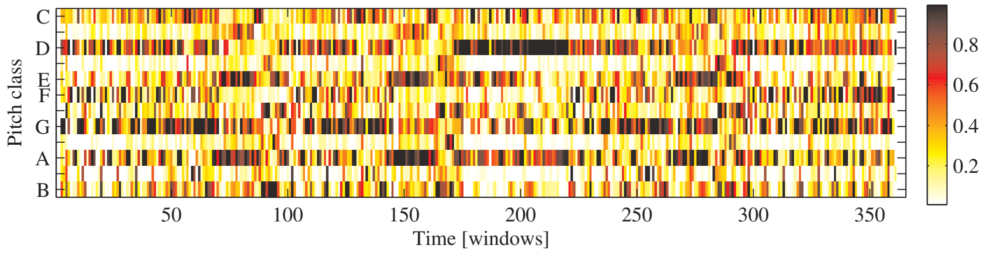


Figure 3.3: Example of an HPCP time series extracted using a moving window from the song "Day Tripper", as performed by The Beatles.

where α_B is a constant and

$$v\left(j, f_n\right) = 12\left[\log_2\left(\frac{f_n}{f_{\text{ref}} 2^{\frac{j}{12}}}\right) + \beta\right], \tag{3.4}$$

β being the integer that minimizes $|v(j, f_n)|$. Constants α_A and α_B are both experimentally set to 2/3. The HPCP of a given window is normalized by its maximum value such that

$$\breve{\mathbf{h}}_i = \frac{\mathbf{h}_i}{\max\left(\mathbf{h}_i\right)}. \tag{3.5}$$

We denote a multidimensional time series of normalized HPCP vectors by $\breve{\mathcal{H}} = [\breve{\mathbf{h}}_1 \cdots \breve{\mathbf{h}}_Y]^{\mathrm{T}}$. An example is depicted in Fig. 3.3.

The HPCP extraction procedure employed here is the same that has been used in Gómez & Herrera (2004, 2006); Gómez et al. (2006a); Serrà et al. (2008c) and Serrà et al. (2008b), and the parameters mentioned in this section have been proven to work well for key estimation, chord extraction, tonal profile determination and version identification, respectively, in the previously cited references. Exhaustive comparisons between 'standard' PCP features and HPCPs have been presented in Gómez (2006), Ong et al. (2006) and Serrà et al. (2008c).

Tonal centroid

From a PCP representation it is quite straightforward to derive other tonal representations. Of particular interest is the tonal centroid (TC) representation proposed by Harte et al. (2006). In the line of Chew (2000), and inspired by other well-known representations of pitch relations such as the *tonnetz* (Cohn, 1997), Harte et al. (2006) proposed an equal-tempered model for pitch space that is specially suitable for data derived from audio. In their implementation, PCP features are mapped to the interior space of a 6-dimensional polytope, where perceptually close harmonic relations appear as small Euclidean distances. Then, in the same manner that Chew (2000) defines her "center of

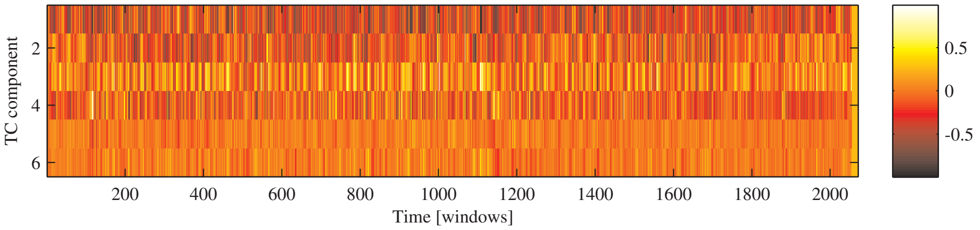


Figure 3.4: Example of a TC time series extracted using a moving window from the song “All along the watchtower”, as performed by Jimi Hendrix.

effect” on the *spiral array* model, the coordinates inside the proposed polytope correspond to the actual TC descriptor. Since a 6-dimensional model is nearly impossible to visualize, Harte et al. (2006) proposed imagining it “as a projection onto the three circularities in the equal tempered tonnetz: the circle of fifths, the circle of minor thirds and the circle of major thirds”. For the sake of brevity, here we only provide the explicit formulae of the TC descriptor. Further details, including explanatory pictures, are given in the cited work. Given the i-th analysis window, the TC descriptor $\mathbf{c}_i = [c_{i,1}, \dots c_{i,6}]^{\mathrm{T}}$ is obtained by multiplying the PCP vector $\mathbf{h}_i$ by a suitable transformation matrix Φ and then normalizing such that

$$c_{i,j} = \frac{1}{\|\bar{\mathbf{h}}_i\|_1} \sum_{k=1}^{12} \phi_{j,k} \bar{h}_{i,k}, \tag{3.6}$$

where $\| \cdot \|_1$ is the L_1 norm. The transformation matrix Φ represents the basis of the 6-dimensional space and is defined as $\Phi = [\mathit{\Phi}_1 \dots \mathit{\Phi}_{12}]^{\mathrm{T}}$, where each column vector

$$\mathit{\Phi}_j = \begin{bmatrix} \phi_{j,1} \\ \phi_{j,2} \\ \phi_{j,3} \\ \phi_{j,4} \\ \phi_{j,5} \\ \phi_{j,6} \end{bmatrix} = \begin{bmatrix} \sin\left((j-1)\frac{7\pi}{6}\right) \\ \cos\left((j-1)\frac{7\pi}{6}\right) \\ \sin\left((j-1)\frac{3\pi}{2}\right) \\ \cos\left((j-1)\frac{3\pi}{2}\right) \\ \frac{1}{2}\sin\left((j-1)\frac{2\pi}{3}\right) \\ \frac{1}{2}\cos\left((j-1)\frac{2\pi}{3}\right) \end{bmatrix}. \tag{3.7}$$

We denote a multidimensional time series of TC vectors as $\mathcal{C} = [\mathbf{c}_1 \cdots \mathbf{c}_Y]^{\mathrm{T}}$. An example is depicted in Fig. 3.4.

Harmonic change

Harte et al. (2006) also define the harmonic change (HC) descriptor: a measure “for detecting changes in the harmonic content of music audio signals”. This descriptor is simply computed as the Euclidean distance between pairs of consecutive TC samples. Therefore, it yields a unidimensional descriptor time

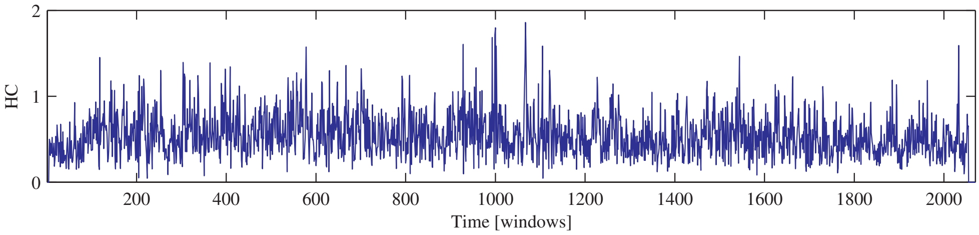


Figure 3.5: Example of an HC time series extracted using a moving window from the song “All along the watchtower”, as performed by Jimi Hendrix.

series $\mathbf{g} = [0, g_2, \ldots g_Y]^{\mathrm{T}}$, $g_i = \|\mathbf{c}_i - \mathbf{c}_{i-1}\|_2$, where $\|\cdot\|_2$ is the L$_2$ or Euclidean norm. An example of $\mathbf{g}$ is depicted in Fig. 3.5.

Downsampling

After the extraction process above, we are left with a descriptor time series (or sequence) of length Y. The HC descriptor is a unidimensional time series $\mathbf{g}$. The TC and PCP descriptors are multidimensional time series $\mathcal{C}$ and $\breve{\mathcal{H}}$ of 6 and 12 components, respectively. For further processing, these three time series are downsampled according to a pre-specified averaging factor ν such that the new length $N = \lfloor Y/\nu \rfloor$. The downsampled time series $\bar{\mathcal{H}}$, $\bar{\mathcal{C}}$ and $\bar{\mathbf{g}}$ are computed as

$$\bar{\mathbf{h}}_n = \frac{\sum_{i=1}^{\nu} \breve{\mathbf{h}}_{i+\nu(n-1)}}{\max\left(\sum_{i=1}^{\nu} \breve{\mathbf{h}}_{i+\nu(n-1)}\right)}, \tag{3.8}$$

$$\bar{\mathbf{c}}_n = \frac{\sum_{i=1}^{\nu} \mathbf{c}_{i+\nu(n-1)}}{\nu}, \tag{3.9}$$

and

$$\bar{g}_n = \frac{\sum_{i=1}^{\nu} g_{i+\nu(n-1)}}{\nu}, \tag{3.10}$$

for $n = 1, \ldots N$, respectively. Alternatively to Eqs. (3.8)-(3.10), the median can be taken. Preliminary analysis shows that it leads to a marginal improvement. The downsampling above obviously favors computational speed since less windows are used for further processing ($N < Y$). Moreover, and for particular values of ν, such downsampling has been proven to be beneficial for version retrieval (Serrà, 2007a; Serrà et al., 2008b). According to these references, we empirically set $\nu = 20$. Since the previous hop size of $\mathcal{Y}$ was 23.2 ms [$W/2 = 1024$ samples with a sampling rate of 44100 Hz, Eq. (3.1)] we now obtain a hop size of 464 ms. Therefore, our resulting descriptor time series have a sampling rate of approximately 2.1 Hz (e.g. a song of 4 minutes yields a music descriptor time series of 516 samples). The resulting window size is 534 ms [$2W + (\nu - 1)W/2$ samples, see also Eq. (3.1)].

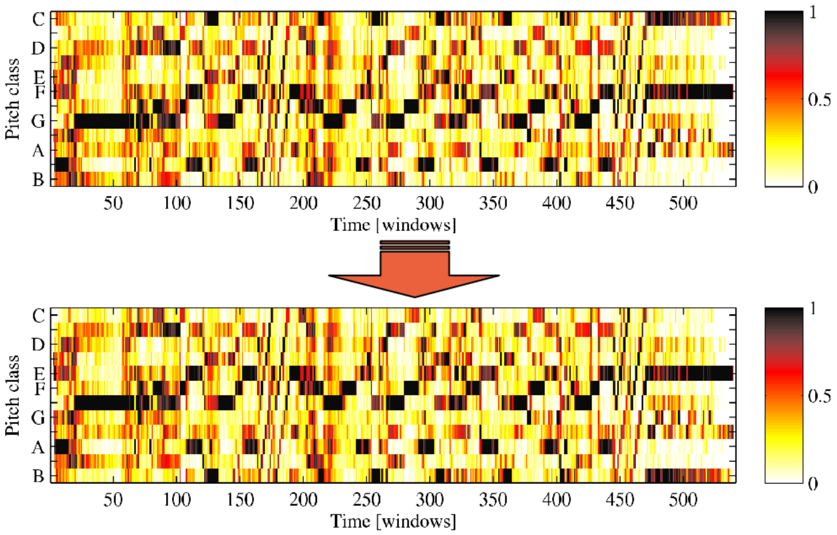


Figure 3.6: Example of a circular shift of the pitch class components by one position along the vertical axis of a PCP time series.

3.2.3 Transposition

A change in the main tonality or key is a common alteration when musicians perform song versions (Sec. 1.2.3). This change in tonality is usually done to adapt the original composition to a different singer or solo instrument, or just for aesthetic reasons. In PCP descriptors, a change in the main tonality is represented by a circular pitch class shift (Purwins, 2005). Accordingly, one can reverse this change using an appropriate circular shift of the pitch class components along the vertical axis of a PCP time series (Fig. 3.6).

To determine the number of shifts, we use the *optimal transposition index* procedure. We first compute a so-called global PCP $\bar{\mathbf{h}}_{\text{glo}}$ by averaging all descriptor vectors in a sequence and normalizing:

$$\bar{\mathbf{h}}_{\text{glo}} = \frac{\sum_{i=1}^{Y} \breve{\mathbf{h}}_i}{\max\left(\sum_{i=1}^{Y} \breve{\mathbf{h}}_i\right)}. \tag{3.11}$$

We do it for the two songs being compared, say u and v, resulting in $\bar{\mathbf{h}}_{\text{glo}}^{(u)}$ and $\bar{\mathbf{h}}_{\text{glo}}^{(v)}$, respectively.

With the global PCPs for the two songs, we calculate a list of 'transposition likelihoods' $\mathbf{o}^{(u,v)} = [o_1^{(u,v)}, \dots o_{12}^{(u,v)}]$. Intuitively, if we test the likelihood between two global representations of the tonal content for all 12 possible shifts, we can have a first guess of which shift is more likely to produce a good match when comparing the two descriptor sequences. Mathematically, and using the dot product $(\cdot)$ as a measure of likelihood,

$$o_k^{(u,v)} = o_k\left(\bar{\mathbf{h}}_{\mathrm{glo}}^{(u)}, \bar{\mathbf{h}}_{\mathrm{glo}}^{(v)}\right) = \bar{\mathbf{h}}_{\mathrm{glo}}^{(u)} \cdot \left[\bar{\mathbf{h}}_{\mathrm{glo}}^{(v)} \trianglerighteq (k-1)\right], \tag{3.12}$$

for $k = 1, \dots 12$. The operation $\mathbf{h} \trianglerighteq k$ implies the application of k circular shifts to the right to vector $\mathbf{h}$. For example, a circular shift to the right of one position is a permutation of the entries in a vector where the last component becomes the first one and all the other components are shifted to the right. More formally, for an arbitrary shift k, $\mathbf{h} \trianglerighteq k = \dot{\mathbf{h}} = [\dot{h}_1, \dots \dot{h}_{12}]^{\mathrm{T}}$, where each $\dot{h}_i$ for $i = 1, \dots 12$ is obtained using the modulo operation:

$$\dot{h}_{(i+k)-12\left\lfloor \frac{i+k}{12} \right\rfloor} = h_i. \tag{3.13}$$

Notice that, instead of 12, any number of components could be used in Eqs. (3.12) and (3.13). Notice also that the aforementioned operations can be performed by means of the circular convolution properties of the FFT, as we demonstrated in Serrà (2007a). This option is interesting in that which regards computational speed, specially when more than 12 PCP bins are considered.

Once we have $\mathbf{o}^{(u,v)}$, we sort its elements in descending order and obtain

$$\mathring{\mathbf{o}}^{(u,v)} = \mathrm{argsort}\left(\left[o_1^{(u,v)}, \dots o_{12}^{(u,v)}\right]\right). \tag{3.14}$$

With this ordered list, one can choose transposition indices in a more informed way. In particular, indices that are more likely to produce a good match of the PCP sequences might occupy the first positions of $\mathring{\mathbf{o}}^{(u,v)}$. Thus, the preferred options would be $\mathring{o}_1^{(u,v)}$, then $\mathring{o}_2^{(u,v)}$, then $\mathring{o}_3^{(u,v)}$ and so forth. Moreover, supposing that indices close to 12 yield a poor match, some of these transpositions may be skipped. We denote with O the maximum number of transposition indices considered. In our experiments we use $O = 2$, thus reducing 6 times the computational costs of the overall system. The effect of parameter O is studied in Sec. 3.4.1. A comparison with a key normalization strategy (Sec. 2.3.1) is also presented. Insights on the internal organization of $\mathring{\mathbf{o}}^{(u,v)}$ were provided in Serrà et al. (2008a).

To effectively transpose the PCPs of song v to the k-th most likely transposition we do

$$\dot{\mathbf{h}}_i^{(v)} = \bar{\mathbf{h}}_i^{(v)} \trianglerighteq \mathring{o}_k^{(u,v)} \tag{3.15}$$

for $i = 1, \dots N$. In case of using the TC or HC descriptors, the above procedure is done for the corresponding PCP time series and then TC and HC are computed, i.e. we do it with $\bar{\mathcal{H}}$ before Eq. (3.6).

To close the section, we should highlight that the above procedure [Eqs. (3.12)-(3.14)] has also been used as part of a PCP binary similarity measure, termed optimal transposition index (OTI) similarity. Basically, one considers two PCP descriptors to be similar if the index $\mathring{o}_1^{(i,j)}$ corresponds to a shift smaller than a semitone (e.g. in the case of 12-bin PCPs, $\mathring{o}_1^{(i,j)}$ must be zero in order to consider that PCPs i and j are the same). The OTI similarity measure has been employed in a number of studies in the context of version detection (e.g. Foucard et al., 2010; Liem & Hanjalic, 2009; Ravuri & Ellis, 2010), including our previous system (Serrà et al., 2008b). Furthermore, it has been used in other studies not strictly related to version identification (e.g. Müller & Ewert, 2010).

3.2.4 State space embedding

The preceding steps yield a descriptor sequence which reflects the temporal evolution of a given song's musical aspect, in our case tonality aspects. Such sequence can be viewed as a multidimensional time series[5] $\mathcal{X}$. From this perspective, one can resort to the existing literature on time series analysis in order to exploit the information contained in $\mathcal{X}$ (e.g. Box & Jenkins, 1976; Lütkepohl, 1993). In particular, we resort to techniques from nonlinear time series analysis (Kantz & Schreiber, 2004). "Nonlinear time series analysis is a practical spin-off from complex dynamical systems theory and chaos theory. Among others, it comprises a variety of techniques to characterize the nonlinearities that give rise to a complex temporal evolution" (Andrzejak, 2010).

We consider a time series $\mathcal{X}$ to be a representation of a succession of system *states* (for our purposes, the system might be associated to the musical composition and the states to a particular musical quality, e.g. instantaneous tonal characteristics). In general, the information about a concrete state is not fully contained in a single sample from a time series (Sauer, 2006). Therefore, to achieve a more comprehensive characterization of such state, one can take into account samples from the recent past[6]. This is formalized by the concept of time delay embedding (Takens, 1981), also termed delay coordinate state space embedding.

The construction of a state space by means of delay coordinates technically solves the problem of the reconstruction of a succession of system states from a single time series measured from this succession (Hughes, 2006). In particular, it specifies a vector space "such that specifying a point in this space specifies

[5] Many of the procedures below are not specific for tonal descriptor time series or sequences, but can be applied to *any* time series. Therefore, to emphasize this generality, a new variable is introduced: we denote a multidimensional time series as a matrix $\mathcal{X} = [\mathbf{x}_1 \cdots \mathbf{x}_N]^{\mathrm{T}}$, where N is the total number of samples and $\mathbf{x}_i$ is a column vector with X components representing an X-dimensional sample at window i. In particular, $\mathcal{X}$ may indistinctly refer to time series of descriptors $\bar{\mathcal{H}}$, $\dot{\mathcal{H}}$, $\bar{\mathcal{C}}$, $\dot{\mathcal{C}}$, $\bar{\mathbf{g}}$ or $\dot{\mathbf{g}}$. Element $x_{i,k}$ of $\mathcal{X}$ represents the magnitude of the k-th descriptor component of the i-th window.

[6] The previous sentences can be illustrated as follows: think of a discrete (sufficiently sampled) sinusoidal signal whose amplitude is between -1 and 1, and suppose we are told that, at a certain moment of time, the signal has an amplitude of 0.85. If this is the only information we have, we are unable to tell if the next sample will be higher or lower than 0.85, i.e. we are unable to tell if we are in the ascending or the descending part of the sinusoidal (ascending or descending state). However, if we know the value of the previous sample, the solution becomes straightforward.

the state of the system, and vice versa" (Kantz & Schreiber, 2004). Thus we can then "study the dynamics of the system by studying the dynamics of the corresponding [vector/state] space points". Most commonly, the construction of delay coordinate state space vectors is done from a unidimensional signal. Nevertheless, extensions to multidimensional signals can be derived [see Vlachos & Kugiumtzis (2008) and references therein].

In our case, for multidimensional samples $\mathbf{x}_i$, we straightforwardly construct delay coordinate state space vectors $\hat{\mathbf{x}}_i$ by vector concatenation such that

$$\hat{\mathbf{x}}_i = \left[\mathbf{x}_i^{\mathrm{T}} \;\; \mathbf{x}_{i-\tau}^{\mathrm{T}} \;\; \cdots \;\; \mathbf{x}_{i-(m-1)\tau}^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{3.16}$$

where m is the embedding dimension and τ is the time delay. The sequence of these reconstructed samples yields again a multidimensional time series $\hat{\mathcal{X}} = [\hat{\mathbf{x}}_{\lambda+1} \ldots \hat{\mathbf{x}}_N]$ of $\hat{N} = N - \lambda - 1$ elements, where $\lambda = (m-1)\tau$ corresponds to the so-called embedding window. Notice that Eq. (3.16) still allows for the use of the raw time series samples (i.e. if $m = 1$ then $\hat{\mathcal{X}} = \mathcal{X}$).

For nonlinear time series analysis, an appropriate choice of m and τ is crucial to extract meaningful information from noisy signals of finite length. Recipes for the estimation of optimal fixed values of m and τ exist, e.g. the false nearest neighbors method and the use of the auto-correlation function decay time (Kantz & Schreiber, 2004). However, here we opt to first study the accuracy of the proposed approach under variation of these parameters and then select the best combination (Sec. 3.4.1).

One should note that the concept of delay coordinates has originally been developed for the reconstruction of stationary deterministic dynamical systems from single variables measured from them (Takens, 1981). Certainly, a music descriptor time series does not represent a signal measured from a stationary dynamical system which could be described by some equation of motion. Nonetheless, delay coordinates, a tool that is routinely used in nonlinear time series analysis (Kantz & Schreiber, 2004), can be pragmatically employed to facilitate the extraction of information contained in descriptor time series $\mathcal{X}$ (c.f. Hegger et al., 2000; Matassini et al., 2002). Analogies between music, MIR and delay coordinates have been discussed in Sec. 3.1.

3.2.5 Cross recurrence plot

A recurrence plot (RP) is a straightforward way to visualize characteristics of similar system states attained at different times (Eckmann et al., 1987). For this purpose, two discrete time axes span a square matrix which is filled with zeros and ones, typically visualized as white and black cells, respectively. Each black cell at coordinates (i, j) indicates a recurrence, i.e. a state at time i which is similar to a state at time j. Thereby, the main diagonal line is black.

A cross recurrence plot (CRP) allows one to highlight equivalences of states between two systems attained at different times. CRPs are constructed in the same way as RPs, but now the two axes span a rectangular, not necessarily square matrix (Zbilut et al., 1998). When a CRP is used to characterize distinct systems, the main diagonal is, in general, not black, and any diagonal path of connected black cells represents similar state sequences exhibited by both systems (Marwan et al., 2007).

Let $\hat{\mathcal{X}}^{(u)}$ and $\hat{\mathcal{X}}^{(v)}$ be two different signals representing two songs u and v of lengths $\hat{N}^{(u)}$ and $\hat{N}^{(v)}$, respectively. To analyze dependencies between these two signals we compute a CRP $\mathcal{R}$ from

$$r_{i,j} = \Theta\left(\varepsilon_i^{(u)} - \left\|\hat{\mathbf{x}}_i^{(u)} - \hat{\mathbf{x}}_j^{(v)}\right\|\right)\Theta\left(\varepsilon_j^{(v)} - \left\|\hat{\mathbf{x}}_i^{(u)} - \hat{\mathbf{x}}_j^{(v)}\right\|\right) \tag{3.17}$$

for $i = 1, \ldots \hat{N}^{(u)}$ and $j = 1, \ldots \hat{N}^{(v)}$, where $\hat{\mathbf{x}}_i^{(u)}$ and $\hat{\mathbf{x}}_j^{(v)}$ are state space representations of songs u and v at windows i and j, respectively, $\Theta(\cdot)$ is the Heaviside step function [$\Theta(\zeta) = 0$ if $\zeta < 0$ and $\Theta(\zeta) = 1$ otherwise], $\varepsilon_i^{(u)}$ and $\varepsilon_j^{(v)}$ are two different threshold distances and $\|\cdot\|$ is any norm. Here we use the Euclidean (L_2) norm.

The thresholds $\varepsilon_i^{(u)}$ and $\varepsilon_j^{(v)}$ are adjusted such that a maximum percentage of neighbors κ is used for $\hat{\mathbf{x}}_i^{(u)}$ and $\hat{\mathbf{x}}_j^{(v)}$. In this way, the total number of non-zero entries in each row and column never exceeds $\kappa\hat{N}^{(u)}$ and $\kappa\hat{N}^{(v)}$, respectively. In-line with studies on the identification of deterministic signals in noisy environments (Zbilut et al., 1998), in pre-analysis we found the use of a fixed percentage of neighbors κ to yield superior accuracies compared to the use of a fixed threshold ε. We study the influence of the parameter κ in Sec. 3.4.1.

Notice that by Eq. (3.17) $r_{i,j} = 1$ if and only if $\hat{\mathbf{x}}_i^{(u)}$ is a neighbor of $\hat{\mathbf{x}}_j^{(v)}$ and at the same time $\hat{\mathbf{x}}_j^{(v)}$ is a neighbor of $\hat{\mathbf{x}}_i^{(u)}$. When dealing with multiple CRPs, a fixed threshold ε is difficult to choose. This is specially true when we have data at different scales. Contrastingly, using a fixed percentage of neighbors can connect points on different scales. However, with a fixed percentage of neighbors, regions with a high density of points are usually connected with regions of low density (c.f. Von Luxburg, 2007). The mutual nearest neighbor strategy of Eq. 3.17 tends to connect points within regions of constant density but, at the same time, does not connect regions of different densities with each other. Therefore, this strategy can be considered as being 'in between' a fixed absolute threshold and a fixed percentage of neighbors. For a similar discussion in the context of spectral clustering see Von Luxburg (2007).

In general, pairs of unrelated songs result in CRPs that exhibit no evident structure (Fig. 3.7b), while CRPs constructed for two song versions show distinct extended patterns (Fig. 3.7a). These extended patterns usually correspond to similar sections, phrases or progressions between both music pieces u and v.

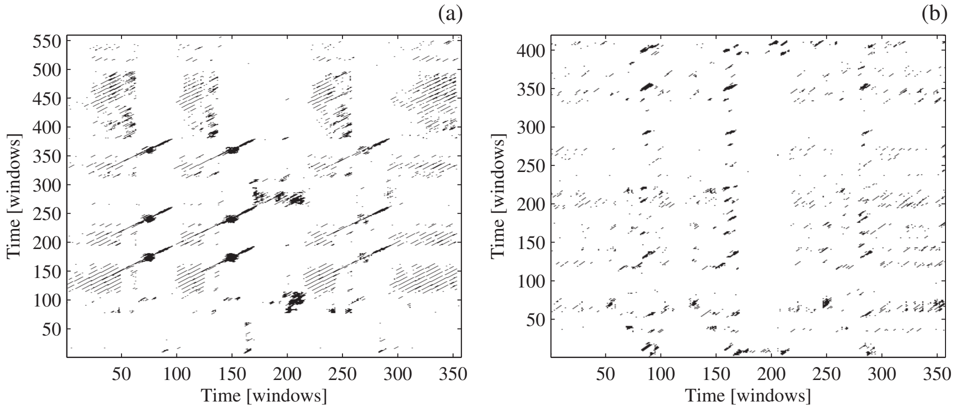


Figure 3.7: CRPs for the song "Day Tripper" as performed by The Beatles, taken as song u, versus two different songs, taken as song v. These are a version made by the group Ocean Colour Scene (a) and the song "I've got a crush on you" as performed by Frank Sinatra (b). Black dots represent recurrences (see text). Parameters are $m = 9$, $\tau = 1$ and $\kappa = 0.08$.

3.2.6 Recurrence quantification measures for version identification

Given a CRP representation of two songs, we need a quantitative criterion to determine whether they are versions or not. In pre-analysis, we tested different measures for recurrence quantification analysis (RQA; Marwan et al., 2007) as input for binary classifiers such as trees or support vector machines in combination with several feature selection algorithms[7] (Witten & Frank, 2005). This analysis showed that the maximal length of diagonal lines ($L_{\max}$) feature yielded by far the highest discriminative power between CRPs from versions and non-versions. All other RQA measures that we tried were found to have no or very low discriminative power. In particular, we tried with the recurrence rate, determinism, average diagonal length, entropy, ratio, laminarity, trapping time, maximal length of horizontal or vertical lines and combinations of them (Marwan et al., 2007).

Despite not being the standard way to compute it, the $L_{\max}$ measure introduced by Eckmann et al. (1987) can be expressed as the maximum value of a cumulative matrix $\mathcal{L}$ computed from the CRP. We initialize $l_{1,j} = l_{i,1} = 0$ for $i = 1, \ldots \hat{N}^{(u)}$ and $j = 1, \ldots \hat{N}^{(v)}$, and then recursively apply

$$l_{i,j} = \begin{cases} l_{i-1,j-1} + 1 & \text{if } r_{i,j} = 1, \\ 0 & \text{if } r_{i,j} = 0, \end{cases} \qquad (3.18)$$

[7] For that we used the data mining software Weka (Hall et al., 2009): http://www.cs.waikato.ac.nz/ml/weka

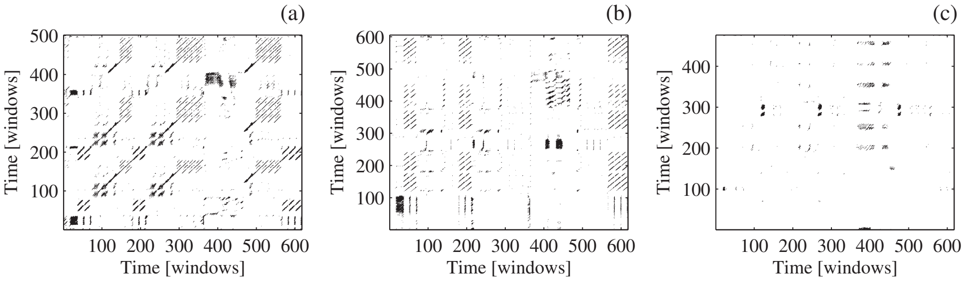


Figure 3.8: CRPs for the song “Gimme, gimme, gimme” as performed by the group ABBA, taken as song u (horizontal axis), versus three different songs, taken as song v (vertical axis). These three different songs are a version made by the group A-Teens (a), a techno performance of the song “Hung up” by Madonna (b) and the song “The robots” by Kraftwerk (c). In (a) $L_{\max} = 43$ starting at windows (118,121), in (b) $L_{\max} = 34$ starting at windows (176,130) and in (c) $L_{\max} = 16$ starting at windows (373,245). Parameters are the same as in Fig. 3.7.

for $i = 2, \dots \hat{N}^{(u)}$ and $j = 2, \dots \hat{N}^{(v)}$ [recall that $r_{i,j}$ was defined in Eq. (3.17)]. Then we can define $L_{\max} = \max\{l_{i,j}\}$ for $i = 1, \dots \hat{N}^{(u)}$ and $j = 1, \dots \hat{N}^{(v)}$.

To understand why $L_{\max}$ performs so well we depict some example CRPs (Fig. 3.8), where we use the same song for u (horizontal axis) and three different songs for v (vertical axis). A high $L_{\max}$ value is obtained when u and v are versions (Fig. 3.8a), whereas a low value is obtained when that is not the case (Fig. 3.8c). An intermediate value is obtained for two songs that share a common tonal progression, but only for brief periods (Fig. 3.8b). It turns out that this particular example of Fig. 3.8b is a border case where one could consider the two songs to be versions or not. The two songs are very different even in terms of main melody and tonality, but still they share a very characteristic (short) sample featuring a flute hook that forms the basis of both songs[8].

Diagonal patterns are clearly discernible in Figs. 3.8a and 3.8b, and the longest of these diagonals corresponds to the maximum time that u and v evolve together without disruptions, i.e. the maximal length of their continuously shared tonal sequence ($L_{\max}$). Notice that only in Fig. 3.8a the longest diagonal is found close to the main diagonal. However, that is not a necessary criterion of v being a version of u (e.g. Fig. 3.8b). In general, this depends on the musical structure of the versions. Often, new performers add, delete or change the introduction, solo sections, endings, verses and so forth (Sec. 1.2.3). Thus, to account for structure changes, it is necessary to consider any diagonal regardless of its position in the CRP. This allows one to detect passages of a recording that have been inserted in any part of another recording.

However, while $L_{\max}$ can account for such structural changes, it cannot account for tempo changes. When versioning a music piece, musicians often adapt the

[8] http://news.bbc.co.uk/2/hi/entertainment/4354028.stm

tempo to their needs and, even in a live performance of the original artist, this feature can change with respect to the original recording (Sec. 1.2.3). Tempo deviations between two song versions result in the curving of CRP diagonal traces.

To quantify the length of curved traces we therefore extend Eq. (3.18) and compute a cumulative matrix $\mathcal{S}$ from the CRP. We initialize $s_{1,j} = s_{2,j} = s_{i,1} = s_{i,2} = 0$ for $i = 1, \ldots \hat{N}^{(u)}$ and $j = 1, \ldots \hat{N}^{(v)}$, and then recursively apply

$$s_{i,j} = \begin{cases} \max\left([s_{i-1,j-1}, s_{i-2,j-1}, s_{i-1,j-2}]\right) + 1 & \text{if } r_{i,j} = 1, \\ 0 & \text{if } r_{i,j} = 0, \end{cases} \tag{3.19}$$

for $i = 3, \ldots \hat{N}^{(u)}$ and $j = 3, \ldots \hat{N}^{(v)}$. Here, the maximum value $S_{\max} = \max\{S_{i,j}\}$ for $i = 1, \ldots \hat{N}^{(u)}$ and $j = 1, \ldots \hat{N}^{(v)}$ corresponds to the length of the longest curved trace in the CRP. This formulation is inspired by common alignment algorithms (Gusfield, 1997; Rabiner & Juang, 1993), but constrains the possible alignments by excluding horizontal and vertical paths. We should note that these particular path connections $(s_{i-1,j-1}, s_{i-2,j-1}, s_{i-1,j-2})$, which are only one aspect of Eq. (3.19), were used before in the available literature. They were found to work well for speech recognition in application to distance matrices (Myers et al., 1980), and for version identification in application to the so-called optimal transposition index-based binary similarity matrices (Serrà et al., 2008b).

Apart from tempo deviations, musicians might skip some chords or part of the melody when performing song versions (Sec. 1.2.3). This practice leads to short disruptions in otherwise coherent traces (see e.g. Fig. 3.7a). Moreover, such disruptions can also be caused by the fact that the considered tonal descriptors might contain some energy not directly associated to tonal content.

To account for disruptions, we therefore extend Eq. (3.19) and compute a cumulative matrix $\mathcal{Q}$ from the CRP. We initialize $q_{1,j} = q_{2,j} = q_{i,1} = q_{i,2} = 0$ for $i = 1, \ldots \hat{N}^{(u)}$ and $j = 1, \ldots \hat{N}^{(v)}$, and then recursively apply

$$q_{i,j} = \begin{cases} \max\left([q_{i-1,j-1}, q_{i-2,j-1}, q_{i-1,j-2}]\right) + 1 & \text{if } r_{i,j} = 1, \\ \max([0, q_{i-1,j-1} - \gamma(r_{i-1,j-1}), & \\ \qquad q_{i-2,j-1} - \gamma(r_{i-2,j-1}), & \\ \qquad q_{i-1,j-2} - \gamma(r_{i-1,j-2})]) & \text{if } r_{i,j} = 0, \end{cases} \tag{3.20}$$

for $i = 3, \ldots \hat{N}^{(u)}$ and $j = 3, \ldots \hat{N}^{(v)}$, with

$$\gamma(r) = \begin{cases} \gamma_o & \text{if } r = 1, \\ \gamma_e & \text{if } r = 0. \end{cases} \tag{3.21}$$

Hence γ_o is a penalty for a disruption onset and γ_e is a penalty for a disruption extension. The zero inside the second max clause in Eq. (3.20) is used to prevent that these penalties lead to negative entries of $\mathcal{Q}$. Notice that for

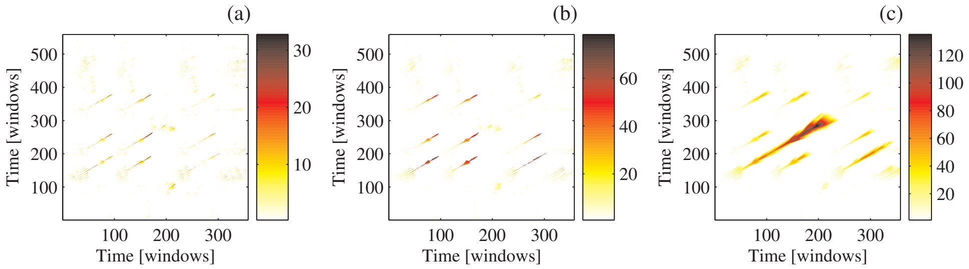


Figure 3.9: "Day Tripper" as performed by The Beatles, taken as song u (horizontal axis), versus an Ocean Colour Scene performance, taken as song v (vertical axis). Example plots of $\mathcal{L}$ (a), $\mathcal{S}$ (b) and $\mathcal{Q}$ (c). Notice the increase in the maximum values (color scales). In (a) $L_{\max} = 33$ starting at windows (140,232), in (b) $S_{\max} = 79$ starting at windows (216,142) and in (c) $Q_{\max} = 136$ starting at windows (14,118). CRP parameters are the same as in Fig. 3.7. Parameters for (c) are $\gamma_o = 3$ and $\gamma_e = 7$.

$\gamma_o, \gamma_e \to \infty$, Eq. (3.20) becomes Eq. (3.19). For $\gamma_o = \gamma_e = 0$, $q_{i,j}$ becomes a cumulative value indicating global similarity between two time series starting at sample 0 and ending at samples i and j, respectively. Note that this has certain analogies with classical dynamic time warping algorithms (Myers, 1980; Rabiner & Juang, 1993). Instead of setting γ_o and γ_e a priori, we study their influence on the accuracy of our version identification system (Sec. 3.4.1). Analogously to $L_{\max}$ and $S_{\max}$, we take $Q_{\max} = \max\{Q_{i,j}\}$ for $i = 1, \ldots \hat{N}^{(u)}$ and $j = 1, \ldots \hat{N}^{(v)}$ to quantify the length of the longest curved and potentially disrupted trace in the CRP.

For illustration we depict some examples for the three quantification measures discussed in this section (Fig. 3.9). The $L_{\max}$ measure (Fig. 3.9a) characterizes straight diagonals regardless of their position. The $S_{\max}$ measure can account for tempo fluctuations resulting in curved traces (Fig. 3.9b). Furthermore, the $Q_{\max}$ measure allows for disruptions of the tonal progression (Fig. 3.9c).

3.2.7 Dissimilarity value

At the end of the process, we are interested in a notion of dissimilarity between song versions. To obtain a dissimilarity value $d_{u,v}$ between songs u and v we simply take the inverse of $Q_{\max}$,

$$d_{u,v} = \frac{1}{\max\left(\left[1, Q_{\max}^{(u,v)}\right]\right)}, \tag{3.22}$$

where $Q_{\max}^{(u,v)}$ denotes the maximal $Q_{\max}$ value for songs u and v after O transpositions have been applied (Sec. 3.2.3). Optionally, one can weight $Q_{\max}$ by the length of the candidate song:

$$d_{u,v} = \frac{\sqrt{N^{(v)}}}{\max\left(\left[1, Q_{\max}^{(u,v)}\right]\right)}. \tag{3.23}$$

Such a weighting scheme is motivated by traditional information retrieval approaches (Baeza-Yates & Ribeiro-Neto, 1999; Manning et al., 2008) and it is intuitively justified by the fact that $Q_{\max}$ is dependent on the length of the descriptor time series. Therefore, one might compensate this dependency by multiplying by a value proportional to the length of one of them. In the case where a symmetric measure is needed, $\sqrt{\min(N^{(v)}, N^{(v)})}$ or $\sqrt{N^{(v)} + N^{(v)}}$ may be used. Nevertheless, in pre-analysis, all normalizations turned out to be somehow equivalent, leading to very similar accuracies. In our experiments we employ Eq. (3.23), which provided a marginal accuracy increment. Further justification of length weighting terms can be found in our previous work (Serrà, 2007a).

3.3 Evaluation methodology

3.3.1 Music collection

To test the effectiveness of the implemented approach, we analyze a music collection comprising a total of 2125 commercial recordings. In particular, we use an arbitrarily selected compilation of versions. This music collection includes 523 *version sets*, where version set refers to a group of versions of the same piece. The average cardinality of these version sets (i.e. the number of performances per set) is 4.06, ranging from 2 to 18. To the best of our knowledge, this is the largest version collection ever employed in MIR version identification experiments. A complete list of the recordings in this music collection can be downloaded from the author's website[9].

The collection spans a variety of genres, with their corresponding sub-genres and styles: pop/rock (1226 songs), electronic (209), jazz/blues (196), world music (165), classical music (133) and miscellaneous (196). The recordings have an average length of 3.6 minutes, ranging from 0.5 to 8 minutes. Apart from genre, additional editorial information has been compiled. A tag cloud of the versioned artists and the 100 most versioned titles has been rendered (Figs. 3.10 and 3.11, respectively). Quantitative information is provided in Fig. 3.12.

The histogram of cardinalities has a very fast decay from 2 to 8 (Fig. 3.12a). The versions with higher cardinalities are "Here comes the sun" (18 versions), originally performed by The Beatles, "A forest" and "Boys don't cry" (18 versions), originally performed by The Cure, "Stairway to heaven" (17 versions),

[9] `http://mtg.upf.edu/files/personal/songList.pdf`

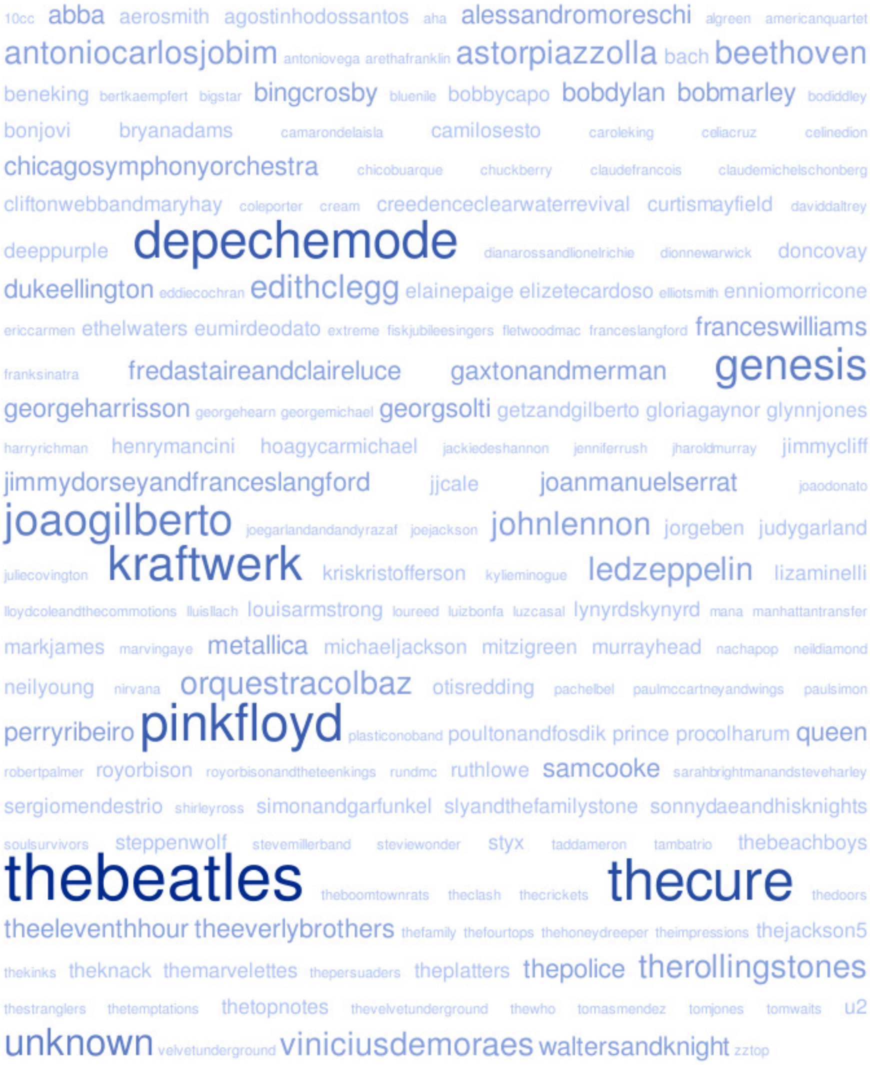


Figure 3.10: Tag cloud of versioned artists in our music collection. The tag clouds were rendered with http://tagcrowd.com.

originally performed by Led Zeppelin, "Eleanor Rigby", "We can work it out" and "Yesterday" (16 versions), originally performed by The Beatles, and "Love song" (16 versions), originally performed by The Cure. In the histogram of most versioned artists we see that, on one hand, there are few artists with more than 10 originals in our collection (Fig. 3.12b). These are The Beatles (121 originals), Pink Floyd (52 originals), The Cure (51 originals), Depeche Mode (36 originals), Kraftwerk (21 originals) and Genesis (15 originals). On the other hand, in Fig. 3.12b we also see that there is a large number of artists who are just represented by one original version in the collection.

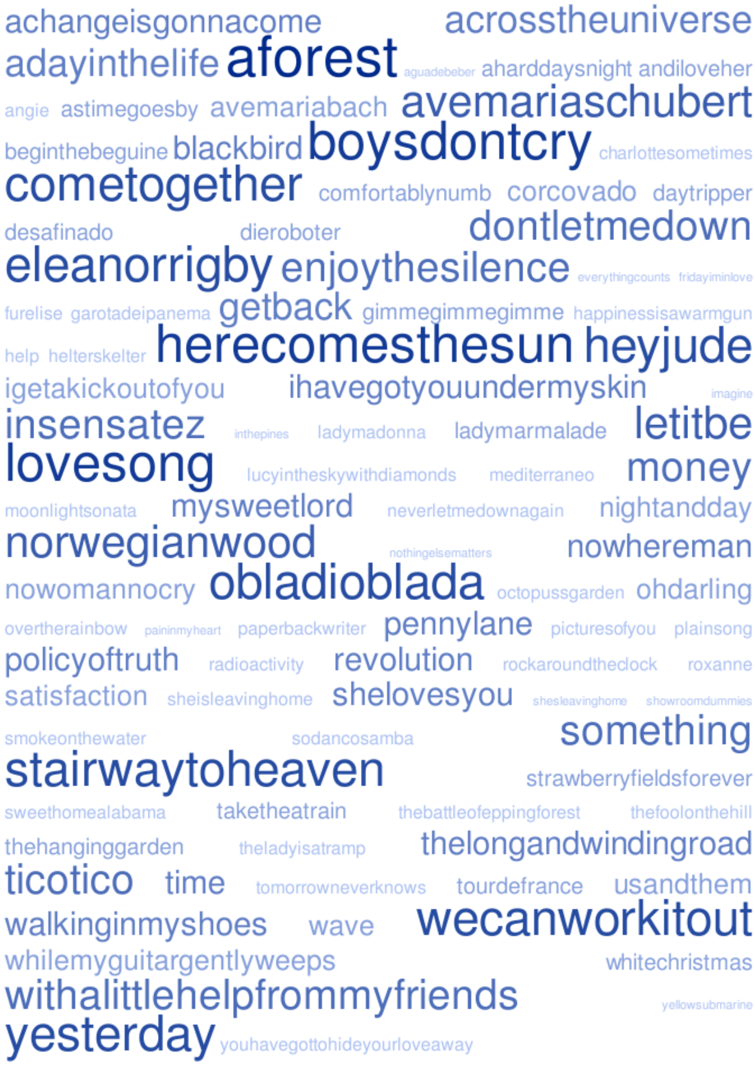


Figure 3.11: Tag cloud of the 100 most versioned song titles in our music collection.

Despite this additional editorial information, the only information we use for evaluation purposes is the version set and the original tag. The version set is a textual description of the underlying composition a music piece is a version of[10]. The original tag is a boolean variable indicating whether a recording corresponds to the original performance (understanding as original the first recorded version). In this chapter, solely the version set is used for quantitative evaluation, while the original tag is used for error analysis. In Chapter 4 we use the original tag for providing quantitative results. Further discussion on the original tag can be found there.

[10] For example, the title of the original recording.

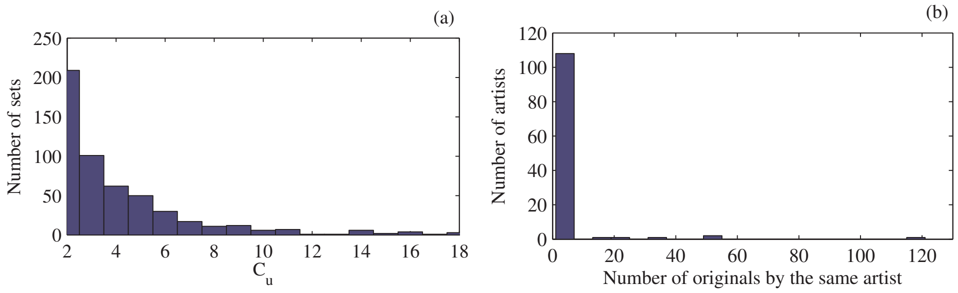


Figure 3.12: Cardinality (a) and original artist (b) histograms. In the cardinality histogram (a) we can see the distribution of version sets as a function of their cardinality. In the original artist histogram (b) we can see the distribution of the number of artists as a function of the original songs in the collection.

For training purposes (parameter optimization), a music collection composed of 17 version sets with cardinality 6 is used. For testing purposes (report of out-of-sample accuracies), another music collection of 30 version sets with cardinality 11 is used. These collections are taken as subsets of the whole collection with no particular preference for specific version sets. Furthermore, both collections are non-overlapping, i.e. there are no version sets shared between them. We denote each music collection by their total number of songs. This way, the first subset is denoted as MC-102, the second subset as MC-330 and the whole collection as MC-2125.

3.3.2 Evaluation measure

To evaluate the accuracy in identifying song versions we proceed as follows. Given a music collection with U songs, we calculate $d_{u,v}$ [Eq. (3.23)] for all $U \times U$ possible pairwise combinations and then create a dissimilarity matrix $\mathcal{D}$. Once $\mathcal{D}$ is computed, we can use standard information retrieval measures to evaluate the discriminative power of this information. We use the mean of average precision measure (MAP), which we denote as $\langle\overline{\psi}\rangle$.

To calculate $\langle\overline{\psi}\rangle$, the rows of $\mathcal{D}$ are used to compute a list Λ_u of $U-1$ songs sorted in ascending order with regard to their dissimilarity to the query song u. Suppose that the query song u belongs to a version set with cardinality C_u+1 (i.e. the set comprises C_u+1 songs). Then, the average precision $\overline{\psi}_u$ is obtained as

$$\overline{\psi}_u = \frac{1}{C_u}\sum_{k=1}^{U-1}\psi_u(k)\Gamma_u(k), \qquad (3.24)$$

where $\psi_u(k)$ is the precision of the sorted list Λ_u at rank k,

$$\psi_u(k) = \frac{1}{k}\sum_{i=1}^{k}\Gamma_u(i), \qquad (3.25)$$

and $\mathit{\Gamma}_u(j)$ is the so-called relevance function: $\mathit{\Gamma}_u(j) = 1$ if the song with rank j in the sorted list is a version of song u, and $\mathit{\Gamma}_u(j) = 0$ otherwise. Hence $\overline{\psi}_u$ ranges between 0 and 1. If the C_u versions of song u take the first C_u ranks, we get $\overline{\psi}_u = 1$. If all versions are found towards the end of $\mathit{\Lambda}_u$, we get $\overline{\psi}_u \approx 0$. The mean of average precision $\langle\overline{\psi}\rangle$ is calculated as the mean of $\overline{\psi}_u$ across all queries $u = 1, \ldots U$. Using Eqs. (3.24) and (3.25) has the advantage of taking into account the whole sorted list where correct items with low rank receive the largest weights.

In addition to the reported results, we estimate the accuracy level expected under the null hypothesis that the dissimilarity matrix $\mathcal{D}$ has no discriminative power with regard to the assignment of versions. For this purpose, we separately permute $\mathit{\Lambda}_u$ for all u and keep all other steps the same. We repeat this process 99 times, corresponding to a significance level of 0.01 of this Monte Carlo null hypothesis test (Robert & Casella, 2004), and take the average, resulting in $\langle\overline{\psi}\rangle_{\text{null}}$. This $\langle\overline{\psi}\rangle_{\text{null}}$ is used to estimate the accuracy of all considered approaches under the specified null hypothesis.

3.4 Results

3.4.1 Parameter optimization

Number of transpositions

Before testing the approach presented here we experimented with transposition and the previous approach of Serrà et al. (2008b). In particular, we compared different transposition strategies. These strategies consisted of (i) transposing with the optimal transposition index as done in Serrà et al. (2008b), (ii) trying all possible transpositions and (iii) transposing by key normalization. Furthermore we tested (iv) the effect of no transposition and (v) the effect of a random transposition. Notice that (i) implies taking only the most likely transposition index and (ii) implies taking all possible indices, i.e. $O = 1$ and $O = 12$ in Eq. (3.14), respectively. Transposing by key normalization (iii) consists in using a key estimation algorithm and then transposing the song to a predefined key (C major or A minor). Then, no further processing step needs to be done when comparing descriptor time series (Sec. 2.3.1). To automatically estimate the key we use the algorithm by Gómez & Herrera (2004), also explained in Gómez (2006), which had an accuracy of 75% for real audio pieces, and scored among the first classified algorithms in the MIREX 2005 key estimation contest[11], with an accuracy of 86% with synthesized MIDI files.

In Table 3.1 we show the general accuracies for the different transposition variants tested. We can appreciate that all transposition methods improve the accuracy of the version identification system up to relative values higher

[11] http://www.music-ir.org/mirex/2005/index.php/Audio_and_Symbolic_Key_Finding

Transposition method	$\langle\overline{\psi}\rangle$
Random transposition	0.16
No transposition	0.51
Key estimation	0.53
$O=1$	0.69
$O=12$	0.73

Table 3.1: Effect of different transposition strategies with MC-102 and the algorithm by Serrà et al. (2008b).

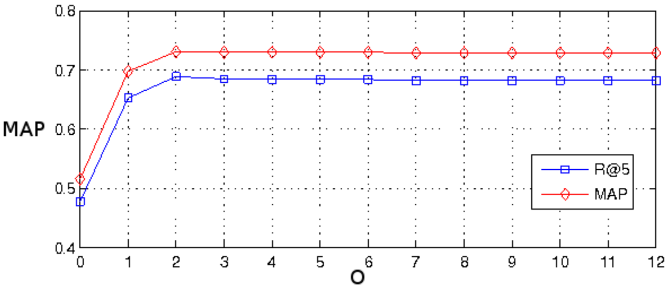


Figure 3.13: Accuracy $\langle\overline{\psi}\rangle$ (MAP) for different number of transposition indices O with MC-102 and the algorithm by Serrà et al. (2008b). Not applying any transposition is depicted as $O=0$. An additional evaluation measure (recall at rank 5) is shown for completeness.

than 40% compared with simply not considering any transposition. The key estimation method performs the worst among the three transposition methods tested (i-iii). This might be due to the fact that automatic key estimation algorithms are not completely reliable, which, surely, introduces errors in the version identification system. Furthermore, as we query all songs against all, these errors might be propagated among queries. For example, if we fail in determining the key of one song, we will neither retrieve its versions nor retrieve it as a version of others. As expected, trying all possible transpositions presents the best accuracy, followed by the method based on the first optimal transposition index (i.e. with $O=1$).

Additionally, we tested the possibility of considering multiple transposition indices, i.e. $O\in[1,12]$ (Fig. 3.13). Note that just considering two transpositions ($O=2$), we are able to achieve the same accuracy as with all of them ($O=12$). This implies that, instead of computing all possible CRPs and $\mathcal{Q}$ matrices, we only have to compute those corresponding to the two most probable or optimal transposition indices. This is quite remarkable as we decrease by a factor of 6 the number of operations carried out by the system. For the remaining experiments we set $O=2$.

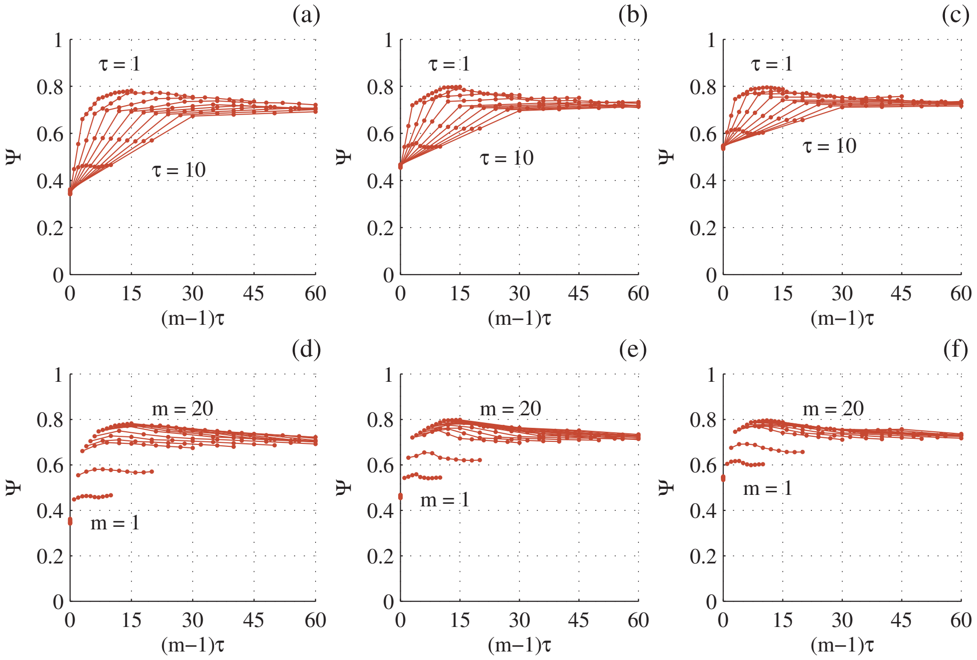


Figure 3.14: Accuracies $\langle\overline{\psi}\rangle$ for different state space reconstruction parameters: $Q_{\max}$ iso-τ (a-c) and iso-m (d-f) curves for $\kappa = 0.05$ (a,d), $\kappa = 0.1$ (b,e) and $\kappa = 0.15$ (c,f).

State space reconstruction

We also use the MC-102 collection to study the influence of the embedding parameters m and τ and the percentage of nearest neighbors κ on our accuracy measure $\langle\overline{\psi}\rangle$. Results for $Q_{\max}$ (Fig. 3.14) illustrate that the use of an embedding ($m > 1$) improves the accuracy of the algorithm as compared to no embedding ($m = 1$). A broad peak of near-maximal $\langle\overline{\psi}\rangle$ values is established for a considerable range of embedding windows λ [approximately $7 < \lambda < 17$, recall that $\lambda = (m-1)\tau$, Sec. 3.2.4]. From these near-maximal values, $\langle\overline{\psi}\rangle$ decreases slightly upon further increasing of the embedding window. Optimal κ values are found between 0.05 and 0.15. Therefore, within these broad ranges of the embedding window λ and κ values, no fine tuning of any of the parameters is required to yield near-optimal accuracies. In the following we use $m = 10$, $\tau = 1$ and $\kappa = 0.1$.

Gap penalties

While accuracies shown in Fig. 3.14 are computed for a disruption onset $\gamma_o = 2$ and disruption extension $\gamma_e = 2$ penalties, the influence of these penalty parameters is further studied in Fig. 3.15. Recall that γ_o and γ_e are introduced

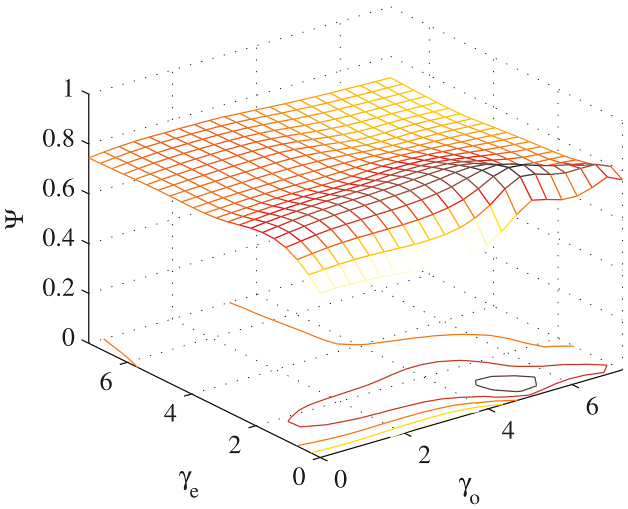


Figure 3.15: Accuracy for different gap penalties: $\langle\overline{\psi}\rangle_{Q_{\max}}$ depending on γ_o and γ_e values.

only in the definition of $Q_{\max}$ and that for $\gamma_o, \gamma_e \to \infty$, the measure $Q_{\max}$ [Eq. (3.20)] reduces to $S_{\max}$ [Eq. (3.19)]. Using finite values of these terms generally increases the accuracy, revealing the advantage of $Q_{\max}$ over $S_{\max}$. Optimal $Q_{\max}$ accuracy values are found for $\gamma_o = 5$ and $\gamma_e = 0.5$.

3.4.2 Out-of-sample accuracy

The same parameter optimization for state space reconstruction described above for $Q_{\max}$ was carried out separately for $L_{\max}$ and $S_{\max}$ and $m = 10$, $\tau = 1$ and $\kappa = 0.1$ yielded near-optimal accuracies as well. Furthermore, no fine tuning was needed since iso-τ and iso-m curves for different κ values had similar shapes as the ones depicted for $Q_{\max}$ in Fig. 3.14. For the MC-102 collection, this in-sample parameter optimization leads to the following accuracies: $\langle\overline{\psi}\rangle_{L_{\max}} = 0.64$, $\langle\overline{\psi}\rangle_{S_{\max}} = 0.73$ and $\langle\overline{\psi}\rangle_{Q_{\max}} = 0.83$ (Fig. 3.16a). The accuracy for MC-330 using the parameters determined by the optimization on MC-102 is shown in Fig. 3.16b. The exact values are $\langle\overline{\psi}\rangle_{L_{\max}} = 0.48$, $\langle\overline{\psi}\rangle_{S_{\max}} = 0.61$ and $\langle\overline{\psi}\rangle_{Q_{\max}} = 0.74$.

The good out-of-sample accuracies achieved with MC-330 indicate that our results cannot be explained by parameter over-optimization. The accuracy increase gained through the derivation from $L_{\max}$ via $S_{\max}$ to $Q_{\max}$ is substantial. Most importantly, this increase in accuracy is reflected in the test collection as well. Moreover, all values for $L_{\max}$, $S_{\max}$ and $Q_{\max}$ are outside the range of $\langle\overline{\psi}\rangle_{\text{null}}$. Therefore, our accuracy values are not consistent with the null hypothesis that the dissimilarity matrices $\mathcal{D}$ have no discriminative power.

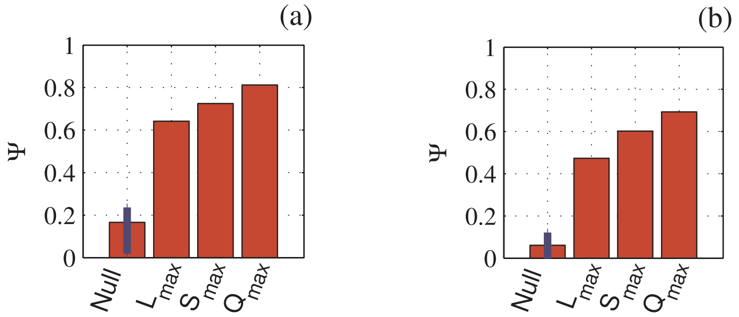


Figure 3.16: Mean average precision $\langle\overline{\psi}\rangle$ for the MC-102 (a) and the MC-330 (b) collections. Error margins in the leftmost bars correspond to the randomizations described in Sec. 3.3.2.

Collection	$\langle\overline{\psi}\rangle_{\text{null}}$	Descriptor		
		PCP	TC	HC
MC-102	0.18	0.83	0.77	0.30
MC-330	0.08	0.74	0.72	0.22
MC-2125	<0.01	0.70	0.64	0.13

Table 3.2: Accuracies $\langle\overline{\psi}\rangle_{Q_{\max}}$ for the different descriptors tested.

The same procedure of in-sample optimization has been carried out for the other two descriptors introduced in Sec. 3.2.2, namely the TC and HC descriptors. Again, no important differences for m, τ, κ, γ_o and γ_e were observed. The final in-sample and out-of-sample accuracies for the three descriptors are reported in Table 3.2. We see that the PCP descriptor performs best, followed by the TC descriptor. We see that the HC descriptor is much less powerful than the other two. This is to be expected, since HC compresses tonal information to a univariate value. Furthermore, tonal change might be less informative than tonal values themselves, which already contain the change information in their temporal evolution. However, the HC accuracy is still higher than the random baseline $\langle\overline{\psi}\rangle_{\text{null}}$.

3.4.3 Comparison with state-of-the-art: MIREX submissions

As stated in the introduction of this chapter, the $Q_{\max}$ algorithm was submitted[12] to MIREX, as well as our previous approach of Serrà et al. (2008b) and a post-processed version of $Q_{\max}$ that will be explained in the next chapter (we denote the latter as $Q^{*}_{\max}$). We now report on the results for all submissions to the "audio cover song identification task" (Table 3.3). These are

[12] For MIREX submissions we only used PCP descriptors, as these were found perform the best (Table 3.2).

Edition	Method	Accuracy (absolute)	Accuracy $\langle\overline{\psi}\rangle$
2006	Sailer & Dressler (2006)	211	-
	Lee (2006)-2	314	-
	Lee (2006)-1	365	-
	Ellis & Poliner (2007)	761	-
2007	Kim & Perelstein (2007)	190	0.06
	Lee (2007)-2, *unpublished*	291	0.09
	Lee (2007)-1, *unpublished*	425	0.13
	Jensen et al. (2008a)	762	0.24
	Bello (2007)	869	0.27
	Ellis & Cotton (2007)	1207	0.33
	Serrà et al. (2008b)	1653	0.52
2008	Jensen et al. (2008b)	763	0.23
	Cao & Li (2008)-1, *unpublished*	1056	0.34
	Cao & Li (2008)-2, *unpublished*	1073	0.34
	Egorov & Linetsky (2008)-1	1762	0.55
	Egorov & Linetsky (2008)-3	1778	0.56
	Egorov & Linetsky (2008)-2	1781	0.56
	Serrà et al. (2009a)-$Q_{\max}$	**2116**	**0.66**
	Serrà et al. (2009a, 2010d)-$Q^*_{\max}$	**2422**	**0.75**
2009	Ahonen & Lemstrom (2008)	646	0.20
	Ravuri & Ellis (2010)	2046	0.66
	Serrà et al. (2009a, 2010d)-$Q^*_{\max}$	**2426**	**0.75**
2010	Di Buccio et al. (2010)	471	0.15
	Martin et al. (2010), *unpublished*	780	0.24
	Rocher et al. (2010), *unpublished*	908	0.29

Table 3.3: MIREX accuracies for the "audio cover song identification task" from 2006 (first edition) to 2010. For completeness, and because the mean of average precisions $\langle\overline{\psi}\rangle$ was not used in 2006, we also report the absolute number of identified versions in top 10 ranks (it ranges from 0, worst case, to 3300, best case). Furthermore, we have skipped 2006 submissions that were not specifically designed for the task (they obtained even lower accuracies than those reported here). The numbers behind the references indicate different algorithmic variants. The term *unpublished* means that, to the author's knowledge, the algorithm has not been published nor disclosed previously.

comprised from the first edition of 2006 until 2010. All this data is available in the MIREX wiki[13] and, for editions before 2008, also in Downie (2008) or Downie et al. (2008).

By looking at the table, we see that our previous algorithm of Serrà et al. (2008b) was found to be the most accurate in the 2007 edition. However, the

[13] `http://www.music-ir.org/mirex/wiki/MIREX_HOME`

two most accurate algorithms in all editions to date are based on $Q_{\max}$. The raw $Q_{\max}$ algorithm as presented here reached an accuracy of $\langle\overline{\psi}\rangle_{Q_{\max}} = 0.66$. It was only outperformed by another algorithm from us which included $Q_{\max}$ as described here, plus one additional post-processing step applied to the dissimilarity matrix derived from $Q_{\max}$, which we denote as $Q^*_{\max}$ ($\langle\overline{\psi}\rangle_{Q^*_{\max}} = 0.75$). The post-processing step consists of detecting version sets instead of isolated songs, and will be explained in detail in the next chapter. The approach by Ravuri & Ellis (2010) also achieves a similar accuracy to $Q_{\max}$, although with a smaller number of identified versions (Table 3.3). Notice however that this approach is not based on a single dissimilarity measure but on a composition of them, one being our previous approach of Serrà et al. (2008b). Furthermore, it uses a supervised train/test post-processing step (Sec. 2.3.2), therefore being more comparable to $Q^*_{\max}$ than to $Q_{\max}$. The approach by Egorov & Linetsky (2008) is also based on Serrà et al. (2008b).

Importantly, the $\langle\overline{\psi}\rangle_{Q_{\max}}$ value obtained for the MIREX music collection is very close to the $\langle\overline{\psi}\rangle_{Q_{\max}}$ values reported for the testing collections used here (0.66 and 0.70, respectively). This provides evidence that the out-of-sample accuracy values reported in Sec. 3.4.2 are not related to any hidden in-sample optimization or overfitting which could have been introduced involuntarily, for example, by a biased selection of songs for the testing collections.

3.4.4 Computation time

The average time spent in the dissimilarity assessment of two recordings is around 0.34 s on an Intel(R) Pentium(R) 4 CPU 2.40GHz with 512M RAM. The $Q_{\max}$ algorithm is quadratic in the length N of the descriptor time series, with an execution speed that also depends on m. If we consider a descriptor dimensionality of X, the algorithm is basically $O\left(N^2 m X\right)$. Such figure could potentially be improved by considering fast methods for searching in metric spaces (Andoni & Indyk, 2008; Chávez et al., 2001) or by approximating the alignment step (Baeza-Yates & Perleberg, 1996; Ukkonen et al., 2003). One should note that N is not very large, since a downsampling step is applied to the time series (Sec. 3.2.2). Such downsampling, in turn, has been shown to be beneficial for version retrieval (Serrà et al., 2008b).

The primary focus of this thesis was on accuracy (Sec. 1.4). Therefore, execution speed was not one of the main objectives of our research. In general, there seems to be an important trade-off between accuracy and speed. Best-performing approaches are computationally expensive, while computationally cheap ones do not achieve competitive accuracies. This trade-off can be easily seen by studying the approaches submitted to MIREX and their accuracies (Table 3.3). In Chapter 5 we propose an alternative strategy that results in substantially faster algorithms with competitive accuracies.

3.4.5 Error analysis

It is always interesting to analyze the errors of an information retrieval system. However, an exhaustive analysis of this kind is, in many cases, unfeasible due to the amount of data and the complexity of the task. In our case, we opt for a general non-exhaustive analysis, which can nevertheless provide valuable information both about the task at hand and our specific approach.

The intended error analysis has two aims: assessing the main characteristics of misidentified versions, and assessing the algorithmic reasons for this misidentification. To narrow down the scope in the search for misidentifications in MC-2125, we concentrate on a particular use-case consisting of querying for the original song u and looking at the retrieved answer Λ_u. Furthermore, we focus on what we call *outstanding false negatives*, i.e. versions that are not detected to be close to the first C_u positions of Λ_u. This approach is motivated by the fact that in the majority of cases we do not observe *outstanding false positives*, i.e. there are no important misidentifications found between the first C_u retrieved items. We only performed the error analysis with the system using the PCP descriptors, as these were found to perform the best in the previous section.

Following the above criteria and restrictions resulted in the manual analysis of 198 false negatives. Such analysis was done with the help of an online demo of the system (see Appendix A). For each of the false negatives, we carried out an assessment of which "type of version" was involved (Sec. 1.2.2), which the musical variations with respect to the original song were (Sec. 1.2.3) and which stage of the system which was, most probably, providing an unreliable output (Sec. 3.2). With these data and the correlations between them we can qualitatively derive some conclusions.

Firstly, we correlate the abovementioned data with the versions' genre information and normalize with respect to the total number of items of each genre. With this process we see that one third (relative) of the analyzed errors correspond to the electronic genre. We hypothesize that this is due to song remixes and quotations that are usually done within the electronic context. Indeed, a look at the version types present in our false negative analysis confirms this hypothesis. Remixes and quotations are intuitively the most difficult versions to detect due to the large amount of musical changes involved and the reduced presence of the essential element of the underlying musical piece (Table 1.1), and this was confirmed by our observations. Another third of the analyzed errors are split between the classical and jazz/blues genres. A possible reason for this is that some of the classical versions in our music collection are highly arranged pieces with much ornamentation, therefore making the match between tonal descriptors more difficult. With regard to jazz/blues genres, we see that these are usually versions of jazz standards, which inherently include improvisation and important changes in both melodies and harmonies (Table 1.1).

Musical variation	Count	%
Timbre (a)	198	100
Timbre (b)	178	90
Tempo	158	80
Timing	143	72
Structure	142	72
Key	109	55
Harmony	116	58
Lyrics	85	43
Noise	36	18

Table 3.4: Distribution of false positives broken down into the changed musical facet (% corresponds to the percentage of outstanding false negatives we analyzed, see text). The row labels are those used in Sec. 1.2.3.

In fact, the number of musical characteristics that change between versions seems to be a key aspect in our error analysis. From a total of 9 characteristics (including the two sub-categories for timbre we underlined in Sec. 1.2.3), few false negatives contained less than 4 musical changes at the same time. On the other hand, many of the observed false negatives had 7 or 8. The mean number of musical changes in the same false negative recording was found to be 5.93, with a standard deviation of 1.74. This value of around 6 in a scale between 0 and 9 reinforces the (commonly held) belief that versions with more changes are more difficult to detect. Table 3.4 provides the absolute and relative (%) error counts distributed by musical facet.

With regard to the critical algorithm stages, we find a clear tendency of unreliable outputs at the very first stages of the system, in particular for descriptor extraction (around two thirds of the analyzed false positives). This does not mean that PCPs could be sensitive to timbre or other facets of the musical pieces. On the contrary, we are able to detect many versions that have a radical change in the instrumentation, which we think it is due to the capacity of PCPs to filter timbre out. However, some errors come from remixes and very percussive pieces. Therefore, one may hypothesize that these are specially challenging cases for such descriptor extraction procedure.

Furthermore, the tonal sequence might not always be, on its own, a valid descriptor for a song version. In particular, our error analysis suggests the consideration of alternative descriptions. Two relevant cases exemplify this: the versions of "We will rock you", originally performed by Queen, and a version of The Rolling Stones' "Satisfaction" performed by P. J. Harvey and Björk. The former hardly contains any melodic or harmonic references. The latter is performed just with a simple and common chord progression and is sung with a forced plain melodic contour (mostly the same note all the time). Apart from the description extraction process, some transposition errors were also found in medleys and versions that incorporated one or several key modulations that were not in the original piece.

Finally, we correlated the three parameters of our error analysis (version type, musical variation and algorithm stage) on a pairwise basis, i.e. version type vs. musical variation, version type vs. algorithm stage and musical variation vs. algorithm step. Nevertheless, no remarkable correlation was obtained. This reveals that, for instance, no particular version type causes problems in a concrete algorithm stage, or that no particular variation seriously affects an algorithm stage. Marginal and quite low correlations (around 0.15) were found between remixes and jazz standards and the descriptor extraction process, and between structure and key changes and the transposition process. Overall, these correlations corroborate what has been said in the rest of the present section.

3.5 Discussion and conclusion

In the present chapter we combine concepts from music signal processing, nonlinear time series analysis, machine learning and information retrieval to build a system that successfully identifies versions of musical pieces. The composition of concepts from these different disciplines naturally results in a modular organization of our model-free approach. Given two audio signals we, at first, use techniques from music signal processing to extract descriptor time series representing their tonal progression. These time series are then used for multivariate embedding by means of delay coordinates. To assess equivalences of states between both systems attained at different times, we use cross recurrence plots and recurrence quantification measures derived from them. In pre-analysis, existing recurrence quantification measures were evaluated using machine learning techniques. The obtained result motivated us to introduce new cross recurrence quantification measures $S_{\max}$ and $Q_{\max}$. Using standard information retrieval evaluation measures we quantify the accuracy for the task at hand. A qualitative error analysis is also done.

We show here that our algorithm leads to a high accuracy for a version identification task on a comprehensive music collection compiled prior to and independently from the study we did in Serrà et al. (2009a). This music collection is divided into non-overlapping testing and training collections. We adjust the parameters on the training collection and then determine the accuracy out-of-sample on a testing collection. Nonetheless, in such a study design, one could still overestimate the true accuracy of the algorithm by involuntarily introducing biases in the compilation of the music collection. However, the close match of accuracy reported here for our music collection and the one obtained in the MIREX campaigns support the generality of the reported results (the music collection used here was compiled prior to and independently from our participation in MIREX). Furthermore, the proposed algorithm has reached the highest accuracies in the MIREX "audio cover song identification task" up to the moment of writing these lines. It has only been surpassed by further developments based on $Q_{\max}$ (details in the forthcoming chapter). This illustrates its superiority in respect to current state-of-the-art algorithms, including our previous approach of Serrà et al. (2008b).

One should note that the concept of delay coordinates has originally been developed for the reconstruction of stationary deterministic dynamical systems from single variables measured from them (Kantz & Schreiber, 2004; Takens, 1981). Also, the identification of coherent traces within the cross recurrence plot is connected to the notion of deterministic dynamics [see Marwan et al. (2007) and references therein]. Certainly, music pieces do not represent the output of a stationary deterministic dynamical system, and therefore, one could argue that applying concepts developed for deterministic systems to such signals is inappropriate. However, if we consider a song as the output of some 'complicated system' evolving in time and a descriptor sequence as a multivariate time series measured from it, we can use the method of delay coordinates to facilitate the extraction of the information characterizing the underlying system. In fact, we find that the accuracy of our version identification system is significantly improved using an embedding, compared to not using it. In conclusion, our work provides a further example for an application of nonlinear time series analysis methods to experimental time series where the assumption of some underlying deterministic dynamics is not fulfilled in a strict sense, but which nonetheless allows one to successfully characterize the system underlying the time series.

In closing, we would like to indicate that the $S_{\max}$ and $Q_{\max}$ measures are not restricted to MIR nor to the particular application of version identification. In Serrà et al. (2009a) we provided evidence for that. Curved structures have been reported in RPs and CRPs of artificial and experimental signals. Artificial signals include frequency modulated periodic signals (Facchini et al., 2005; Groth, 2005; Marwan et al., 2002a) or time series derived from Rössler dynamics with bidirectional couplings close to the onset of phase synchronization (Groth, 2005). Experimental data include signals with nonlinearly re-scaled or distorted time axes such as geophysical data of sediment cores subjected to different compressions (Marwan et al., 2002a), symbolic dynamic representations of electroencephalographic recordings from the onsets of epileptic seizures (Groth, 2005) or acoustic signals from calls of primates (Facchini et al., 2005). Far beyond these particular examples, it can be conjectured that important features of further experimental signals, e.g. from bioinformatics (Aach & Church, 2001), physiology (Webber Jr. & Zbilut, 1994), human speech processing (Rabiner & Juang, 1993) or climatology (Marwan & Kurths, 2002), are reflected in curved and disrupted traces in RPs and CRPs. A quantitative assessment of these traces by means of the proposed measures can thus help to characterize a multitude of systems from different scientific disciplines.

CHAPTER 4

Characterization and exploitation of version groups

4.1 Introduction

Traditionally, version identification has been set up as a standard information retrieval task based on the query-by-example framework (Sec. 2.3.3). In this framework, the user submits an example query (a song u) and receives an answer (a list of songs Λ_u, ranked by their relevance to the query). Such a setup has remarkably conditioned the way of implementing and evaluating version identification systems. This assertion can be contrasted by looking at the literature review of Sec. 2.3. In particular, with regard to the implementation of version identification systems, we have seen that the efforts have been concentrated in achieving a metric that faithfully captures pairwise version similarities (Sec. 2.3.1). With regard to the evaluation of version identification systems, we have seen that the cited references always resort to common measures from information retrieval that serve to quantify the accuracies of query-by-example systems (Sec. 2.3.3). In fact, we ourselves have also used the query-by-example framework in the previous chapter for evaluating our system (Sec. 3.3.2).

In this chapter we consider a new approach that goes beyond query-by-example to achieve a more complete characterization of a music collection composed of song versions. In particular, instead of isolated songs, our approach focuses on groups[1] of songs. Therefore we identify, given a music collection, coherent groups of versions of the same piece. This way one can exploit the regularities found in the results of a query-by-example system for a given music collection. Indeed, music collections are usually organized and structured at multiple levels. In the case of version detection, songs naturally cluster into so-called

[1] Throughout the chapter we will use the words group, set, community or cluster interchangeably.

version sets (we have already used this terminology in the presentation of our evaluation framework, Sec. 3.3.2).

For automatically identifying the versions sets of a music collection we employ a number of grouping algorithms on top of the $Q_{\max}$ measure explained in the previous chapter. As grouping algorithms we consider unsupervised clustering (Jain et al., 1999; Xu & Wunsch II, 2009) and community detection algorithms (Danon et al., 2005; Fortunato, 2010). Notice that the task of version set detection can naturally be placed within a typical clustering framework. According to Jain & Dubes (1988) and Jain et al. (1999), the "typical pattern clustering activity involves the following steps: (1) pattern representation, optionally including feature extraction and/or selection, (2) definition of a pattern proximity measure appropriate to the data domain, (3) clustering or grouping" and, if needed, "(4) data abstraction" and "(5) assessment of output". We now observe that steps 1 and 2 have already been performed to obtain $Q_{\max}$ in the previous chapter. Therefore, only step 3, the grouping step, remains to be made. This is the main focus of this chapter. Nevertheless, we also explore steps 4 and 5 with the detected groups of versions. In particular, we study the version 'prototypes' found within a group and their relation to the original piece.

Apart from the typical clustering framework above, the detection of version sets can be formulated from a complex network perspective. Complex networks[2] are a well-established way to represent the interactions between a number of elements (Boccaletti et al., 2006; Costa et al., 2008; Newman, 2003; Strogatz, 2001), from proteins (Jeong et al., 2001) to web pages (Baeza-Yates et al., 2007). The interaction between elements usually gives rise to certain structures in the network. In fact, one of the most relevant features of networks is community structure (or clustering), i.e. the organization of vertices in clusters, with many edges joining vertices of the same cluster and comparatively few edges joining vertices of different clusters (Danon et al., 2005; Fortunato & Castellano, 2009). Thus, detecting communities is of enormous importance in disciplines where interacting elements are represented through networks, and many successful approaches for community detection have been proposed, specially in biology, sociology and computer science [for an overview see Fortunato (2010)].

The reader may easily see the resemblance between the detection of version sets and a more classical community detection task. This way, a set of vertices $\mathbf{u} \equiv \{u_1, \dots u_U\}$ represents the U recordings being analyzed, and the elements of the $U \times U$ weight matrix $\mathcal{D}$ represent the dissimilarity between any couple of nodes. Provided that the weights of this matrix are assigned with the help of a suitable version dissimilarity metric (recall that $\mathcal{D}$ was obtained from $Q_{\max}$, Sec. 3.2.7), communities inside this complex network will represent version

[2] In this thesis we use the terms network and graph interchangeably. Node and vertex, and link and edge are also used interchangeably.

sets. Although complex networks and community detection algorithms have been used in many problems involving complex systems (Boccaletti et al., 2006; Costa et al., 2008), and more specifically in studying musical networks (Buldú et al., 2007; Cano et al., 2006; Teitelbaum et al., 2008), to the best of our knowledge they have never been applied to a retrieval task before. We also use the framework of complex networks to study the characteristics of associations between song versions, in particular to assess their clustering properties and their relationships.

As we have noted in Serrà et al. (2009b), and subsequently in Serrà et al. (2010d), there are many intuitive advantages behind the aforementioned change of paradigm, namely going from specific query answers Λ_u to the detection of coherent groups of items. Importantly, one should bear in mind that these advantages are not specific for the version detection task, and that they hold for any information retrieval (IR) system operating through query-by-example (Baeza-Yates & Ribeiro-Neto, 1999), including analogous systems such as recommendation systems (Resnick & Varian, 1997). First, given that current systems provide a suitable metric to quantify the similarity between single query items, several well-researched options exist to exploit this information in order to detect inherent groups of items (we have outlined them above and present specific ones below). Second, focusing on groups of items may help the system in retrieving more coherent answers for isolated queries. In particular, the answers to any query belonging to a given group would coherently contain the other songs in the group, an advantage that is not guaranteed by query-by-example systems alone. Third, music collections are usually organized and structured on multiple levels (e.g. the version sets in our case). Thus we can infer and exploit these regularities to increase the overall accuracy of traditional version identification systems. Note that the two previous advantages specifically aim to achieve higher user satisfaction and confidence in IR systems, as they can be perceived as rational or intelligent agents or assistants (Russell & Norvig, 2003). Finally, once groups of coherent items are correctly detected, one can study these groups in order to retrieve new information, either from the individual communities or from the relations between these.

4.2 Method

4.2.1 Overview

In this chapter, we use the $Q_{\max}$ measure and our version collection (MC-2125, Sec. 3.3.1) in order to build a complex network. More specifically, the dissimilarity matrix $\mathcal{D}$ is used as a weighted adjacency matrix for a complex network. First, this complex network is analyzed in order to confirm that communities of versions are present (Sec. 4.2.2). We study both the topology of the network and the characteristics of the percolation process, i.e. how the network properties change with the threshold used to define the links. Subsequently, several

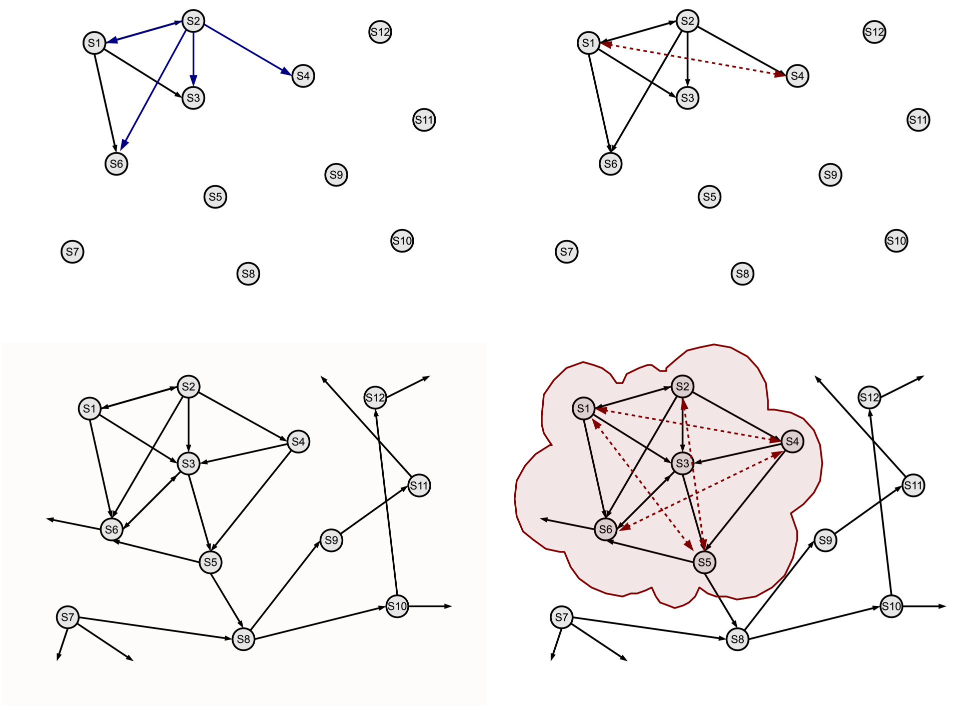


Figure 4.1: Example of the idea of inferring version similarities by exploiting group detection. In the top row the relation between S1 and S4 is inferred (red broken arrow) from the results of querying S1 (directed black arrows) and S2 (directed blue arrows). In the bottom row all queries are available, allowing the detection of a coherent group of items (red cloud) and to infer new relationships between the elements of this group (red broken arrows).

strategies to correctly detect groups of versions are presented (Sec. 4.2.3). In particular, we consider 4 clustering algorithms and 4 community detection approaches. Three of the community detection approaches are original ideas. In addition, we show how the $Q_{\max}$ measure can be post-processed to include the information gained by a group detection algorithm (Sec. 4.2.4, Fig. 4.1 exemplifies these ideas). This yields $Q^{*}_{\max}$, a measure that improves the results of a query-by-example system by exploiting the information obtained through the detection of groups of versions. Finally, we investigate the organization of these groups of versions (Sec. 4.5). In particular, we present a study on the role that original songs play within a group of versions. To the author's knowledge, this work constitutes the first reported attempt in this direction.

4.2.2 Analysis of the version network

As mentioned, the consideration of elements $d_{u,v}$ of $\mathcal{D}$ as link weights between vertices representing the songs of MC-2125 results in a complex network repre-

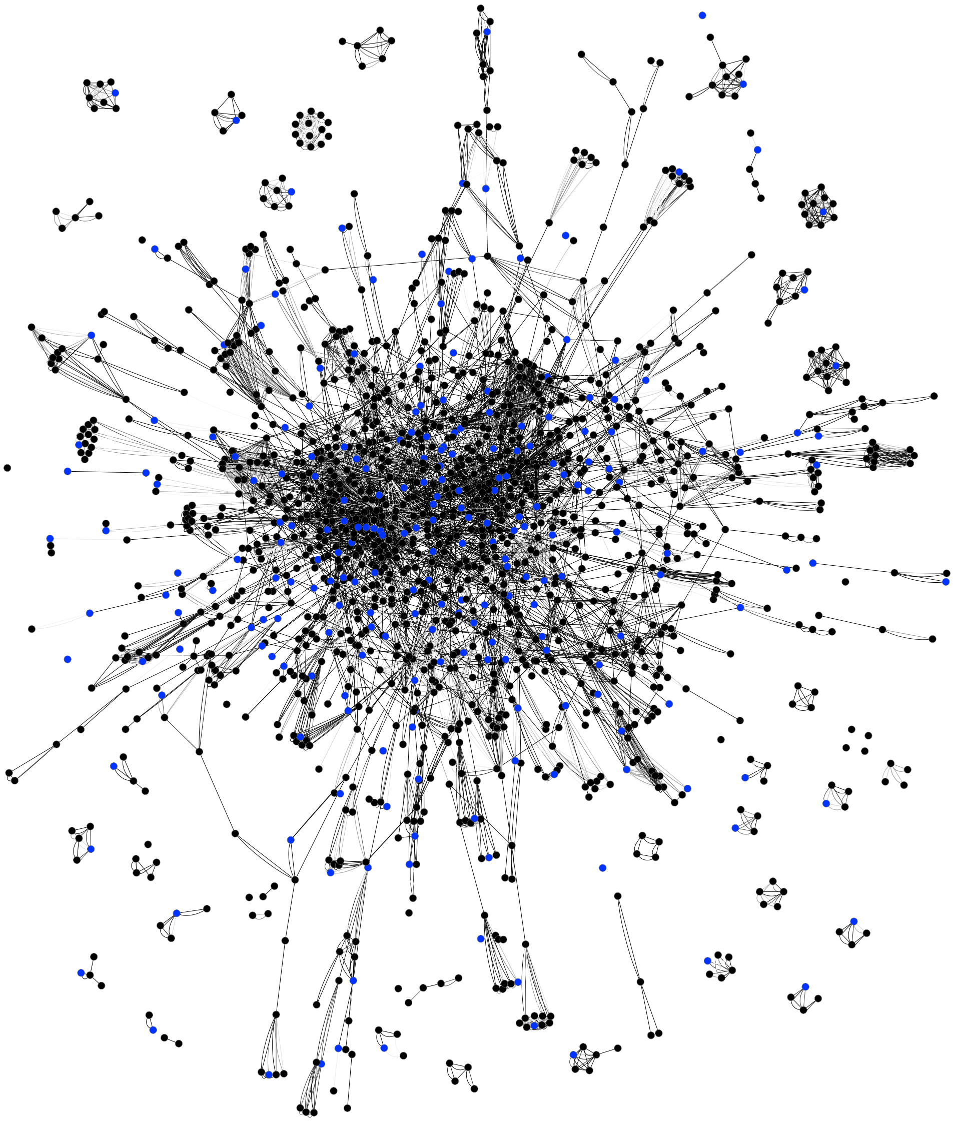


Figure 4.2: Graphical representation of the version network when a threshold of 0.2 is applied. Original songs are drawn in blue, while versions are in black. In Sec. 4.5, the role of original songs inside each community will be further studied.

sentation. This resulting network is depicted in Fig. 4.2. A threshold has been applied so that only links with $d_{u,v} \leq 0.2$ are drawn. Some clusters, i.e. sets of versions, are already visible, especially in the external zones of the network. In order to understand how the network evolves when the threshold is modified, we study six different classical metrics as a function of the threshold (Boccaletti et al., 2006; Costa et al., 2007):

1. Graph density: the number of existing edges, normalized by the total number of possible edges between all vertices.

2. Number of independent components: alternatively called number of connected components. A connected component of a graph is a sub-graph in which any two vertices are connected, and which is connected to no additional vertices [a directed sub-graph is called (strongly) connected if there is a path from each vertex in the graph to every other vertex].

3. Size of the strong giant component: a giant component is the connected component that contains the majority of the entire graph's nodes. The reported value corresponds to the proportion of nodes that belong to this component.

4. Number of isolated nodes: the number of nodes that do not have any link. We report the proportion of these nodes relative to the total number of nodes.

5. Efficiency (Latora & Marchiori, 2001): the harmonic mean of geodesic lengths, where geodesic length corresponds to the number of edges in the shortest path connecting two nodes. Efficiency is an indicator of the traffic capacity of a network.

6. Clustering coefficient: the fraction of connected triples of nodes (triads) which also form triangles[3] (three nodes that are fully connected). Clustering coefficient is a measure of degree to which nodes in a graph tend to cluster together.

4.2.3 Detection of version groups

We assess the detection of version groups by evaluating several methods either based on clustering or on complex networks. Since standard implementations of clustering algorithms do not operate with an asymmetric dissimilarity measure, in this and in the subsequent section we use a symmetric dissimilarity matrix $\mathcal{D}'$. This matrix is obtained by simply taking the new values $d'_{u,v} = d'_{u,v} = (d_{u,v} + d_{v,u})/2$.

K-medoids

K-medoids (KM) is a classical technique to group a set of objects inside a previously known number of K clusters (Theodoridis & Koutroumbas, 2006; Xu & Wunsch II, 2009). The most common realization of KM clustering is as follows (Theodoridis & Koutroumbas, 2006):

1. Randomly select K data points as the medoids.

[3] To calculate this value we removed the directionality of the links.

2. Associate each data point to the closest medoid (in our case this closeness is determined using $\mathcal{D}'$).

3. For each medoid u_{M}:

    a) For each non-medoid data point u_{D}:

        i. Swap u_{M} and u_{D} and compute the total cost of the configuration.

4. Select the configuration with the lowest cost.

5. Repeat steps 2 to 4 until there is no change in the medoid or a suitable number of iterations has been reached.

The K-medoids algorithm is a common choice when the computation of means is unavailable (as it solely operates on pairwise distances). Such is the case that fits better with the version identification task. Furthermore, the K-medoids algorithm can exhibit some advantages compared to the standard K-means algorithm, in particular when dealing with noisy samples (Xu & Wunsch II, 2009). The main drawback for its application is that, as well as with the K-means algorithm, the K-medoids algorithm needs to set K, the number of expected clusters. However, several heuristics can be used for that purpose (Theodoridis & Koutroumbas, 2006).
In our experiments we employ the K-medoids implementation of the TAMO package[4], which incorporates several heuristics to achieve an optimal K value[5]. We use the default parameters and try all possible heuristics provided in the implementation.

Hierarchical clustering

Hierarchical clustering creates a hierarchy of clusters which may be represented in a tree structure called a dendrogram (Jain et al., 1999; Xu & Wunsch II, 2009). The root of the tree consists of a single cluster containing all observations, and the leaves correspond to individual observations. Hierarchical clusterings can be agglomerative (bottom-up, each observation starts in its own cluster) or divisive (top-down, all observations start in one cluster). A generic realization of an agglomerative hierarchical clustering algorithm is as follows (Jain et al., 1999):

1. Compute the dissimilarity matrix containing the distance between each pair of observations (in our case the observations are recordings and we use the dissimilarity matrix $\mathcal{D}'$). Treat each observation as a cluster.

[4] `http://fraenkel.mit.edu/TAMO`

[5] `http://fraenkel.mit.edu/TAMO/documentation/TAMO.Clustering.Kmedoids.html`

2. Find the most similar pair of clusters using the dissimilarity matrix and merge these two clusters into one cluster.

3. Update the dissimilarity matrix to reflect this merge operation. A linkage criterion is used to determine the distance between sets of observations as a function of the pairwise distances between observations. Common linkage criteria are single linkage (the minimum distance in a set of observations is taken), complete linkage (the maximum is taken) or mean average linkage (a linear combination of the distances in a set of observations is done).

4. Repeat steps 2 and 3 until all patterns are in one cluster.

In our experiments we consider four representative agglomerative hierarchical clustering methods: single linkage (SL), complete linkage (CL), group average linkage (UPGMA) and weighted average linkage (WPGMA). We use the HCLUSTER implementation[6] with the default parameters and, in order to cut the dendrogram at a suitable layer we try different cluster validity criteria such as checking descendants for inconsistent values or considering the maximal or the average inter-cluster cophenetic distance[7]. In the end, all clustering algorithms rely only on the definition of a distance threshold d'_{Th}, which is set experimentally.

Modularity optimization

This method (MO), as well as the next three algorithms, is designed to exploit a complex network collaborative approach. In particular, MO extracts the community structure from large networks based on the optimization of the network modularity (Danon et al., 2005; Fortunato, 2010; Fortunato & Castellano, 2009). The modularity of a partition is a scalar value between -1 and 1 that measures the density of links inside communities as compared to links between communities. Although it may have some shortcomings, the maximization of the network modularity is, by far, the most popular way to detect communities in graphs (Fortunato, 2010). The standard implementation for optimizing modularity as proposed by Clauset et al. (2003) consists of recursively merging communities that optimize the production of such quantity (analogously to hierarchical clustering algorithms). To merge links between nodes of the same community one usually sums their weights.
In our experiments we use the method proposed by Blondel et al. (2008), with the implementation by Aynaud[8]. This method first looks for 'small' communities by optimizing modularity in a local way and then aggregates nodes of

[6] http://code.google.com/p/scipy-cluster

[7] http://www.soe.ucsc.edu/~eads/cluster.html

[8] http://perso.crans.org/~aynaud/communities/index.html

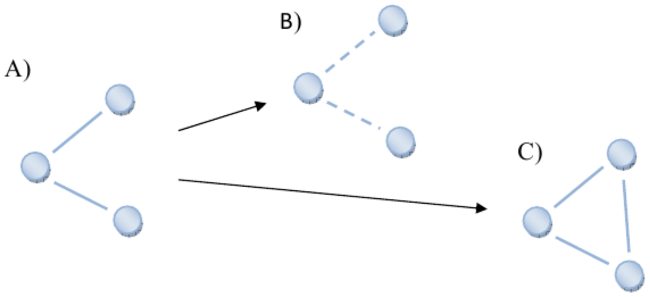


Figure 4.3: Example of the process of reinforcing the triangular coherence of the network. The sub-network on the left (A) can be improved by either deleting a link (B) or by adding a third link between the two nodes that were not originally connected (C).

the same community and builds a new network whose nodes are the communities. These steps are repeated reiteratively until a maximum of modularity is attained. The method proposed by Blondel et al. (2008) is reported to outperform all other known community detection algorithms in terms of computational time while still maintaining a high accuracy. Importantly, this method has the capacity to manage networks containing millions of nodes and links.

Proposed method 1

Our first proposed method (PM1) applies a threshold to each network link in order to create an unweighted network where two nodes are connected only if their weight (dissimilarity) is less than a certain value d'_{Th}. In addition, for each row of $\mathcal{D}'$ (each node), we only allow a maximum number of connections, considering only the lowest values of the thresholded row as valid links. That is, we only consider the first k'_{Th} nearest neighbors for each node, where k'_{Th} is a threshold rank (i.e. top k'_{Th} items). Values d'_{Th} and k'_{Th} are set experimentally. Finally, each connected component is assigned to be a group of versions. Although this is a very naïve approach, it will be shown that, given the considered network and dissimilarity measure, it achieves a high accuracy level at low computational costs.

Proposed method 2

The previous approach could be further improved by reinforcing triangular connections in the complex network before the last step of checking for connected components. In other words, proposed method 2 (PM2) tries to reduce the 'uncertainty' generated by triplets of nodes connected by two edges and to reinforce coherence in a triangular sense. This idea can be illustrated by the following example (Fig. 4.3).

Suppose that three vertices in the network, e.g. u_i, u_j and u_k, were versions: the resulting subnetwork should be triangular, so that every vertex is connected with the two remaining ones. On the other hand, if u_i, u_j and u_k were not versions, no edge should exist between them. If couples u_i, u_j and u_i, u_k are respectively connected (Fig. 4.3A), we can induce more coherence by either deleting one of the existing edges (Fig. 4.3B), or by creating a connection between u_j and u_k (i.e. forcing the existence of a triangle, Fig. 4.3C). This coherence can be measured through an objective function ϱ which considers complete and incomplete triangles in the whole network. We define ϱ as a weighted difference between the number of complete triangles $N_{\bigtriangledown}$ and the number of incomplete triangles $N_{\vee}$ (three vertices connected by only two links) that can be computed from a pair of vertices:

$$\varrho(N_{\bigtriangledown}, N_{\vee}) = N_{\bigtriangledown} - \iota N_{\vee}. \tag{4.1}$$

The constant ι, which weights the penalization for having incomplete triangles, is set experimentally.

The implementation of this idea sequentially analyzes each pair of vertices u_i, u_j by calculating the value of ϱ for two situations: (i) when an edge between u_i and u_j is artificially created and (ii) when such an edge is deleted. Then, the option which maximizes ϱ is preserved and the adjacency matrix is updated as necessary. The process of assigning version sets is the same as with PM1.

Proposed method 3

The computation time of the previous method can be substantially reduced by considering for the computation of ϱ only those vertices whose connections seem to be uncertain. This is what proposed method 3 (PM3) does: if the dissimilarity between two songs is extremely high or low, this means that the version identification system has clearly detected a match or a mismatch. Accordingly, we only consider for ϱ the pairs of vertices whose edge weight is close to d'_{Th} (a closeness margin is empirically set).

4.2.4 Accuracy improvement: from $Q_{\max}$ to $Q^*_{\max}$

Once a coherent group of versions is detected by means of the methods explained above, we can straightforwardly improve the overall accuracy of a query-by-example system. The idea is to modify the original dissimilarity measure of the system by means of the information obtained through the detection of version sets.

Given the dissimilarity matrix $\mathcal{D}$ and a solution for the cluster or community detection problem, one can calculate a refined dissimilarity matrix $\hat{\mathcal{D}}$ by setting its elements

$$\hat{d}_{u,v} = \frac{d_{u,v}}{\max(\mathcal{D})} + \varsigma_{u,v} \tag{4.2}$$

for $u, v = 1, \ldots U$, where $\varsigma_{i,j} = 0$ if songs u and v are estimated to be in the same community and $\varsigma_{i,j} = M$ otherwise. To ensure that the songs in the same community have $\hat{d}_{u,v} \leq 1$ and the others have $\hat{d}_{u,v} > 1$ we use a constant $M > 1$. Importantly, this refined matrix $\hat{\mathcal{D}}$ can be used to rank the query answers Λ_u again and, consequently, to evaluate the achieved accuracy improvement (for that we only have to compare the accuracies achieved with $\mathcal{D}$ and $\hat{\mathcal{D}}$, see below). The resulting measure from this process has previously been denoted as $Q^*_{\max}$, in contraposition to $Q_{\max}$.

4.3 Evaluation methodology

4.3.1 Music collection

For the evaluation of the approaches in this chapter we use the results obtained by query-by-example for PCP descriptors (matrix $\mathcal{D}$). Furthermore, we employ the MC-2125 music collection (Sec. 3.3.1) and its different (possibly overlapping) subsets. These subsets are organized into different setups. Each setup is defined by different parameters: the total number of songs U, the number of version sets U_{S} the collection includes, the cardinality C of the version sets (i.e. the number of songs in the set) and the number of added noise songs U_{N} (i.e. songs that do not belong to any version set, which are included to add difficulty to the task). Because some setups can lead to wrong accuracy estimations (Sec. 2.3.3), it is safer to consider several of them, including fixed and variable cardinalities.

In our experiments we use the setups summarized in Table 4.1. The whole MC-2125 collection corresponds to setup 3. For other setups we randomly sample version sets from setup 3 and repeat the experiments N_{T} times (number of trials, average accuracies reported). We either sample version sets with a constant cardinality ($C = 4$, the expected cardinality of setup 3, Sec. 3.3.1) or with a variable cardinality ($C = \chi$, a random value between 2 and 18 taken from an exponential distribution[9] with an expected mean of 4).

4.3.2 Evaluation measures

To quantitatively evaluate version set detection we resort to the classical F-measure with even weighting (Baeza-Yates & Ribeiro-Neto, 1999),

$$F = \frac{2\bar{P}\bar{R}}{\bar{P} + \bar{R}}, \tag{4.3}$$

which goes from 0 (worst case) to 1 (best case). In Eq. (4.3), $\bar{P}$ and $\bar{R}$ correspond to precision and recall, respectively. For this evaluation, we compute

[9] We found the exponential function to be the best candidate to model the distribution of version set cardinalities shown in Sec. 3.3.1.

Setup	Parameters				
	U_S	C	U_N	U	N_T
1.1	25	4	0	100	20
1.2	25	χ	0	$\langle 100\rangle$	20
1.3	25	4	100	200	20
1.4	25	χ	100	$\langle 200\rangle$	20
2.1	125	4	0	500	20
2.2	125	χ	0	$\langle 500\rangle$	20
2.3	125	4	400	900	20
2.4	125	χ	400	$\langle 900\rangle$	20
3	523	χ	0	2125	1

Table 4.1: Experimental setup summary. The $\langle\cdot\rangle$ delimiters denote expected value.

P_u and R_u independently for each song u and average afterwards with all U songs. Unlike other clustering evaluation measures, F is not computed on a per-cluster basis, but on a per-song basis through the averaging of P_u and R_u across all songs. This way, and in contrast with the typical clustering F-measure or other clustering evaluation measures like Purity, Entropy or F-Score (e.g. Sahoo et al., 2006; Zhao & Karypis, 2002), we do not have to blindly choose which cluster represents a given version set.

The process for obtaining F is as follows. For each song u, we count the number of true positives $N_i^{\mathrm{T+}}$ (i.e. the number of actual versions of song u estimated to belong to the the same community as u), the number of false positives $N_i^{\mathrm{F+}}$ (i.e. the number of songs estimated to belong to the same group as u that are not actual versions of u) and the number of false negatives $N_i^{\mathrm{F-}}$ (i.e. the number of actual versions of u that are not detected as belonging to the same group as u). Then we define

$$P_u = \frac{N_i^{\mathrm{T+}}}{N_i^{\mathrm{T+}} + N_i^{\mathrm{F+}}} \tag{4.4}$$

and

$$R_u = \frac{N_i^{\mathrm{T+}}}{N_i^{\mathrm{T+}} + N_i^{\mathrm{F-}}}. \tag{4.5}$$

These two quantities [Eqs. (4.4) and (4.5)] are finally averaged across all U songs ($u = 1, \ldots U$) to obtain $\bar{P}$ and $\bar{R}$, respectively.

To quantitatively evaluate the improvements in retrieval accuracy we again use the mean of average precision measure $\left\langle\overline{\psi}\right\rangle$ [Eqs. (3.24) and (3.25), Sec. 3.3.2]. We define the relative improvement in mean average precision as

$$\Delta = 100\left(\frac{\left\langle\overline{\psi}(\hat{\mathcal{D}})\right\rangle}{\left\langle\overline{\psi}(\mathcal{D})\right\rangle} - 1\right), \tag{4.6}$$

where $\langle\overline{\psi}(\mathcal{D})\rangle$ denotes the mean of average precisions for a dissimilarity matrix $\mathcal{D}$. Notice that $\langle\overline{\psi}(\mathcal{D})\rangle \in [0,1]$. Therefore, it could be the case that Δ would be undetermined or tending towards infinity. However, in our experiments, $\langle\overline{\psi}\rangle$ never reaches a value of zero. For that to happen, the list Λ_u should not contain any version at all [see Eqs. (3.24) and (3.25)].

4.4 Results

4.4.1 Analysis of the version network

In order to understand how the network evolves when the threshold is modified, we represent six different classical metrics as a function of the threshold (Fig. 4.4). In the same plots, we also draw the values for the last five measures as expected in random networks with the same number of vertices and links (i.e. with the same graph density).

By looking at the evolution of these metrics, some interesting knowledge about the network and its structure can be inferred. Notice that, by reducing the threshold (and therefore increasing the deleted links), the network splits into a higher number of clusters than expected, which represents the formation of version communities (Fig. 4.4, top right plot). This process begins around a threshold of 0.5 (see for instance the evolution of the size of the strong giant component, Fig. 4.4, middle left). When these communities are formed, they maintain a high clustering coefficient, i.e. sub-networks of versions tend to be fully connected, with triangular coherence (Fig. 4.4, bottom right plot, between 0.3 and 0.5). It is also interesting to note that the number of isolated nodes remains lower than expected, except for high thresholds (Fig. 4.4, middle right plot). This suggests that most of the songs are connected to some cluster while a small group of them are different, with unique musical features. Overall, the above analysis reports evidence for the formation of version sets from the output of $Q_{\max}$, and suggests a successful detection of these through some clustering or community detection algorithm such as the ones presented above.

4.4.2 Detection of version sets

To assess the grouping algorithms accuracy we independently optimized the highlighted parameters for each algorithm on setups 1.1 to 1.4. Within this optimization phase, we saw that the definition of a threshold d'_{Th} was, in general, the only critical parameter for all algorithms (for our proposed methods we used k'_{Th} between 1 and 3). The different heuristics used for the clustering algorithms were found to yield equivalent accuracies. Besides d'_{Th}, all other parameters turned out to be uncritical for obtaining near-optimal accuracies. Methods that had specially broad ranges of these near-optimal accuracies were KM, PM2 and all hierarchical clustering algorithms considered.

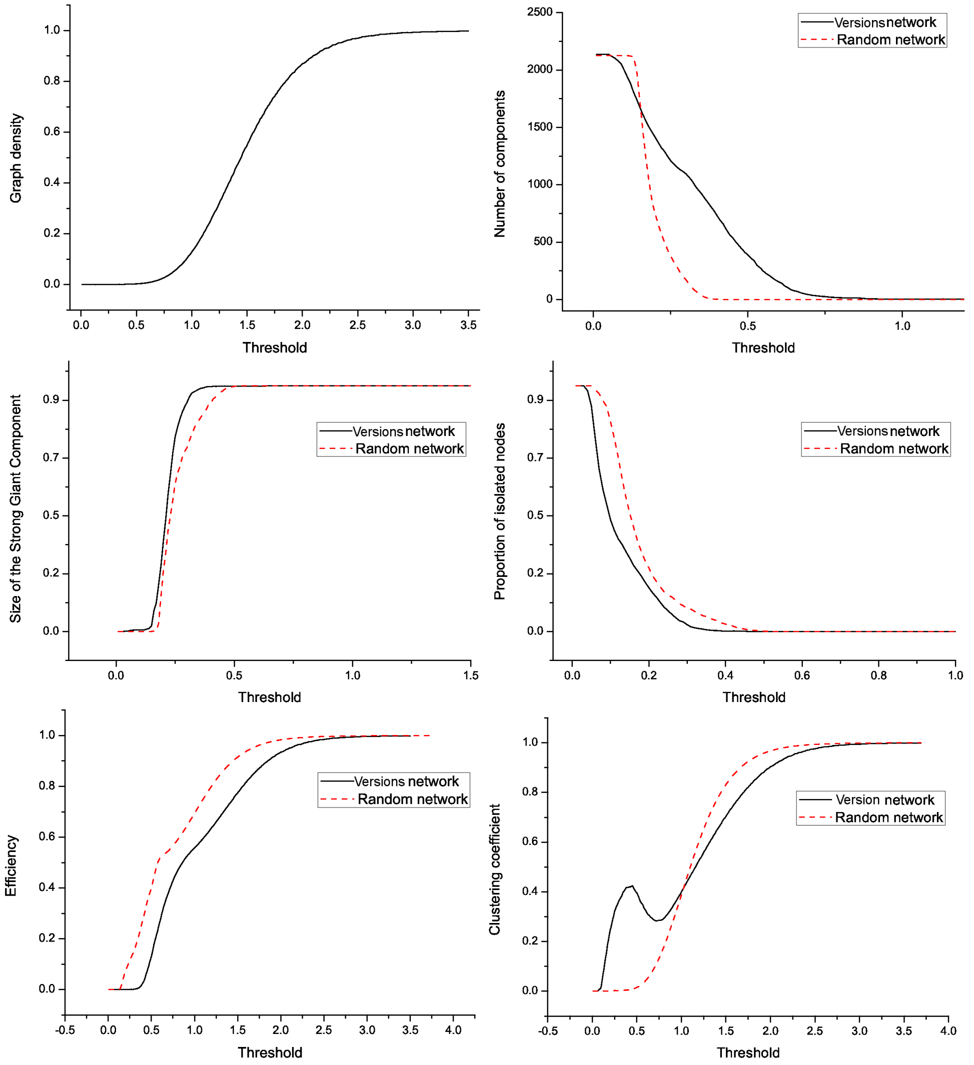


Figure 4.4: (Black solid lines) Evolution of six metrics of the network as a function of the threshold. These metrics are, from top left to bottom right: graph density, number of independent components, size of the strong giant component, number of isolated nodes, efficiency and clustering coefficient. (Red broken lines) Expected value in a random network with the same number of nodes and links.

We report the accuracies for setups 2.1 to 3 in Table 4.2. We see that accuracies for PM1 and PM3 are comparable to those achieved by the other algorithms and, in some setups, even better. The high values obtained (above 0.8 in the majority of cases, some of them nearly reaching 0.9) indicate that the considered approaches are able to effectively detect groups of versions. This allows the possibility of enhancing the answer of a query-based retrieval system by reporting these detected groups and thus reinforcing coherence within answers.

Algorithm	Setup				
	2.1	2.2	2.3	2.4	3
KM	0.657	0.662	0.681	0.692	*n.c.*
SL	0.786	0.808	0.876	0.889	0.777
CL	0.811	0.817	0.829	0.826	0.791
UPGMA	**0.823**	0.827	0.829	0.826	0.791
WPGMA	**0.825**	**0.842**	0.844	0.843	**0.815**
MO	0.802	0.829	**0.885**	**0.894**	**0.808**
PM1	0.807	**0.834**	**0.881**	**0.890**	0.807
PM2	0.773	0.771	*n.c.*	*n.c.*	*n.c.*
PM3	0.787	0.786	0.865	0.876	0.763

Table 4.2: Accuracy F for the considered algorithms and setups (see Table 4.1 for the details on the different setups). Due to algorithms complexity, some results were not computed (denoted as *n.c.*). The two highest F values for each setup are highlighted in bold.

4.4.3 Accuracy improvement

To assess accuracy improvements we independently optimized all distance thresholds d'_{Th} for each algorithm on setups 1.1 to 1.4. The relative accuracy increments Δ obtained for setups 2.1 to 3 are reported in Table 4.3. Overall, these relative increments are between 3 and 5% for UPGMA, WPGMA, MO and all PMs, with some of them reaching nearly 6 or 7%. We see that, in general, methods based on complex networks perform better, specially MO and PM1. We also see that the inclusion of 'noise' or 'control' songs ($U_{\mathrm{N}} = 400$, setups 2.3 and 2.4) affects the performance of nearly all algorithms, with the exception of poorly performing ones.

An additional out-of-sample test was done within the MIREX "audio cover song identification" task (Sec. 2.3.3). In the editions of 2008 and 2009 we submitted the same two versions of our system and obtained the two highest accuracies achieved up to the moment of writing this thesis[10]. The first version of the system (submitted solely to the 2008 edition) corresponded to the $Q_{\max}$ measure alone (explained in the previous chapter). The accuracy achieved with the $Q_{\max}$ approach was $\langle\overline{\psi}\rangle = 0.66$ (MIREX results have been shown in the previous chapter, Table 3.3). The second version of the system (submitted to both editions) comprised $Q_{\max}$ plus PM1[11] and the dissimilarity update of Eq. (4.6). This approach was called $Q^{*}_{\max}$, and achieved an accuracy of

[10] The results for 2008 and 2009 are available from http://music-ir.org/mirex/2008 and http://music-ir.org/mirex/2009, respectively.

[11] We only submitted PM1 because it was the only algorithm we had available at that time.

Algorithm	Setup				
	2.1	2.2	2.3	2.4	3
KM	2.26	2.40	2.06	2.29	*n.c.*
SL	2.26	2.40	1.16	2.29	2.05
CL	1.93	1.19	1.43	1.10	1.28
UPGMA	5.87	5.22	3.96	**3.49**	4.37
WPGMA	4.91	3.58	3.83	2.67	3.60
MO	**6.84**	**5.37**	**5.14**	2.94	**5.54**
PM1	**6.15**	**5.70**	**4.95**	**3.28**	**5.49**
PM2	5.98	4.85	*n.c.*	*n.c.*	*n.c.*
PM3	6.05	5.10	3.81	2.97	4.73

Table 4.3: Relative accuracy increase Δ for the considered setups (see Table 4.1 for the details on the different setups). Due to algorithms complexity, some results were not computed (denoted as *n.c.*). The two highest Δ values for each setup are highlighted in bold.

$\langle\overline{\psi}\rangle = 0.75$ (Table 3.3). This corresponds to a relative increment $\Delta = 13.64$. Such an increment is substantially higher than those achieved here with our data, most probably because the setup for the MIREX task is $U_{\mathrm{C}} = 30$, $C = 11$ and $U_{\mathrm{N}} = 770$. This specific setup might capitalize the effects that version set detection can have in improving the accuracy. In particular, when high cardinalities are considered, one can think of the techniques presented in this chapter to achieve more dramatic impacts in final accuracies.

As a further example, it may also be interesting to also see the results in absolute terms based on the collection subsets presented in Sec. 3.3.1 and PM1. With this setting we can compare the accuracies achieved by $Q_{\max}$ and $Q^*_{\max}$ in the same way as in the previous paragraph. We have that for MC-2125 (setup 3 here) we go from $\langle\overline{\psi}\rangle = 0.70$ to $\langle\overline{\psi}\rangle = 0.74$ ($\Delta = 5.71$), for MC-330 we go from $\langle\overline{\psi}\rangle = 0.75$ to $\langle\overline{\psi}\rangle = 0.82$ ($\Delta = 9.33$) and for MC-102 we go from $\langle\overline{\psi}\rangle = 0.82$ to $\langle\overline{\psi}\rangle = 0.91$ ($\Delta = 10.98$).

4.4.4 A note on the dissimilarity thresholds

In the parameter optimization stages reported for the two previous sections we have stated that the dissimilarity threshold d'_{Th} seems to be a critical parameter for all approaches. We should notice that alternative approaches for reducing the dependency to d'_{Th} were presented in Lagrange & Serrà (2010). In the same reference, we also provided evidence that d'_{Th} was more or less independent of the music collection (Fig. 4.5). For that we used MC-2125 and the "covers80" dataset[12] (Ellis & Cotton, 2007), a version collection commonly used in the

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

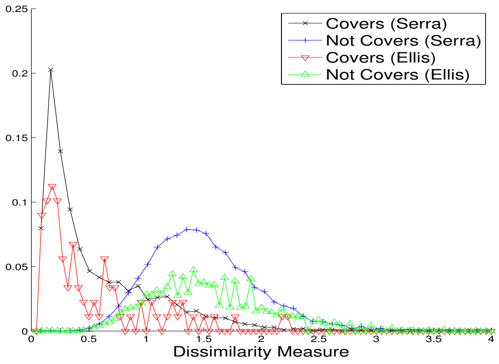


Figure 4.5: Normalized histograms for the dissimilarity measure $d'_{u,v}$ [plot obtained from Lagrange & Serrà (2010); the vertical axis represents the probability of $d'_{u,v}$]. The plot compares the $d'_{u,v}$ values obtained with the MC-2125 collection (solid lines with crosses, denoted in the legend as "Serra") and with the "covers80" collection (solid lines with triangles, denoted as "Ellis", see text). Values for versions and not versions are reported (denoted as "covers" and "not covers", respectively). A threshold estimate can be obtained visually.

MIR community[13]. In spite of this collection independence, we nevertheless hypothesize that d'_{Th} may still vary depending on the version identification approach, i.e. each approach might need its own d'_{Th}.

4.4.5 Computation time

In the application of these techniques to big real-world music collections, computational complexity is of great importance. To qualitatively evaluate this aspect, we report the average amount of time spent by the algorithms to achieve a solution for each setup (Fig. 4.6). We see that KM and PM2 are completely inadequate for processing collections with more than 2000 songs (e.g. setup 3). The steep rise in the time spent by hierarchical clustering algorithms to find a cluster solution for setup 3 also raises some doubts as to the usefulness of these algorithms for huge music collections [$O(U^2 \log U)$, Jain et al. (1999)]. Furthermore, hierarchical clustering algorithms, as well as the KM algorithm, take the full pairwise dissimilarity matrix as input. Therefore, with a music collection of, for instance, 10 million songs, this distance matrix might be difficult to handle.

In contrast, algorithms based on complex networks show a better performance (with the aforementioned exception of PM2). More specifically, MO, PM1 and PM3 use local information (i.e. at most the nearest r'_{Th} neighbors of the queries), while PM3 furthermore acts on a small subset of the links. It should also be noticed that the resulting network is very sparse, i.e. the number of

[13] Some remarks on this dataset have been made in Sec. 2.3.3.

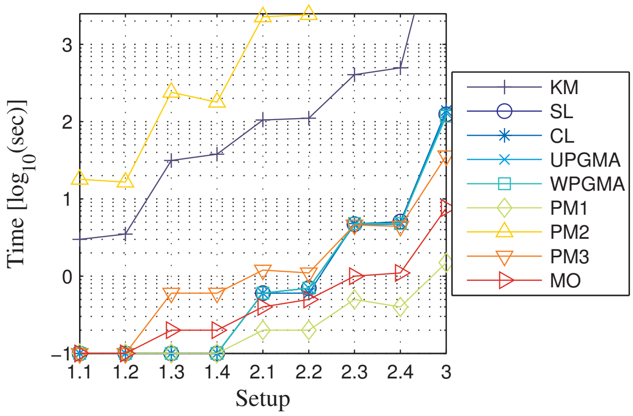


Figure 4.6: Average time performance for each considered setup. Algorithms were run with an Intel(R) Pentium(R) 4 CPU 2.40GHz with 512M RAM.

links is much lower than U^2 (Boccaletti et al., 2006) and, therefore, calculations on such graphs can be strongly optimized both in memory requirements and computational costs [as demonstrated, for instance, by Blondel et al. (2008), who have applied their method to networks of millions of nodes and links].

4.4.6 Error analysis

With the information about the identified version groups we can perform a further error analysis. In particular, it is interesting to look at the most outstanding 'confusions'. For instance, it could be interesting to look at groups of versions that are in fact composed of two or more *real* groups, i.e. two or more version groups that share a single detected cluster. Leaving behind a few cases which we are not able to explain in an intuitive manner, we find that the abovementioned 'cluster sharing' phenomenon usually has a musicological explanation. Indeed, the major source for this kind of 'confusions' seems to be the strong similarities between harmonic progressions of different songs (Table 4.4). Inside this category we can highlight some subgroups.

The first and primary source of confusion is the fact of sharing a chord progression. Indeed, there are many songs that can share their tonal or chord progression. However, by considering PCP descriptors instead of chords, and thus using a finer, more detailed characterization, one should presumably have less confusions of this kind. Nonetheless, the usage of tempo, transposition and structure invariance strategies again dramatically boost the number of possible confusions. That is, if there is a harmonically equivalent sequence of PCPs, the system sometimes detects it in spite of tempo and transposition changes, no matter its location within the piece.

A second source of confusions are the songs that have a chord progression involving just dominant and sub-dominant chords (I, IV and V, sometimes

Version sets	Original performer	Chord progression
“All along the watchtower”	Bob Dylan	C♯m, B, A
“Stairway to heaven”	Led Zeppelin	Am, G, F
“Boys don’t cry”	The Cure	A, Bm, C♯m, D
“Here there and everywhere”	The Beatles	G, Am, Bm, C
“Canon in D major”	Pachelbel	D, A, Bm, G
“Let it be”	The Beatles	C, G, Am, F
“No woman no cry”	Bob Marley	C, G, Am, F
“Go west”	Pet Shop Boys	C, G, Am, Em, F
“A whiter shade of pale”	Procol Harum	C, Em/G, Am, C, F
“Help me make it through the night”	Kris Kristofferson	D, D, G, G, D, D, E, E, A
“Oh darling”	The Beatles	A, D, A, B, E
“Imagine”	John Lennon	C, F, C, F, C, F, G, G
“Watching the weels”	John Lennon	C, F, C, F, C, F, Dm, G, G
“Take the A train”	Duke Ellington	C, C, D, D, Dm, G, C, Dm, G, C
“The lady is a tramp”	Mitzi Green	C, C, Dm, G, G, Cm, Dm, G, C
“O amor em paz”	Joao Gilberto	Bm, E, Am, D, G
“Mr. Sandman”	The Chordettes	B, E, A, D, G
“I’ll survive”	Gloria Gaynor	Am, Dm, G, C, F
“Over the rainbow”	Judy Garland	Csus4, Dm, G, C, F

Table 4.4: Some examples of version group confusions due to shared chord progressions.

substituting I by its minor relative VIm). One example employing a chord progression based on I, IV and V degrees is the common blues progression (I, IV, I, V, IV, I, V). Other examples are the song “Knocking on heaven’s door” (I, V, IV), originally performed by Bob Dylan, “Just like heaven” (V, IV, VIm, I), originally performed by The Cure or “No woman no cry”, originally performed by Bob Marley (I, V, VIm, IV, I). In fact, these songs, jointly with a few others that also combine the I, IV and V degrees, form a single compact cluster after our group detection stage.

A third example of confusions between version groups is found with songs that just have a one or two-chord progression. In this case, the tonal progression is barely definitive of the song and one should look at more detailed elements such as the melody and ornamentations.

Finally, we find some confusions with typical cadences or bass-lines. In particular when there is a dominant/tonic chain with the same root or predominant/fundamental notes. This is the case for example with the last group of songs in Table 4.4. All the confusions we have highlighted in this section were visible in the online demo of the system (see Appendix A).

4.5 The role of the original song within its versions

Following the “typical pattern clustering activity” we outlined in the introduction (Sec. 4.1), we now introduce the concepts of “cluster assessment” and “data abstraction” to clusters of versions. That is, assuming that we are able to correctly detect coherent groups of versions, we study the relationships between the songs inside these groups. In particular, we focus on the issues of prototype determination and compactness description.

It could be relevant to note that, in a data clustering context, many applications exploit compact cluster descriptions (Jain et al., 1999; Xu & Wunsch II, 2009). These compact descriptions are usually given in terms of representative patterns such as the centroid or, if we want to restrict ourselves to existing elements within the cluster, the medoid. In the context of version networks, one could also be interested in finding a compact representative description of a group of versions. Indeed, analogously to the clustering context, the centroids and medoids of version groups can be effectively estimated. This way, the centroid and the medoid of a group of versions would correspond to the 'average realization' and the 'best example' of the underlying musical piece, respectively (in other words, to the prototype).

From the point of view of music perception and cognition, a musical work or song can be considered as a category (Zbikowski 2002; see also Sec. 2.2.3). Categories are one of the basic devices to represent knowledge, either by humans or by machines (Rogers & McClelland, 2004). According to existing empirical evidence, some authors postulate that our brain builds categories around prototypes, which encapsulate the statistically most-prevalent category features, and against which potential category members are compared (Rosch & Mervis, 1975). With this view, after listening to several song versions, a prototype for the underlying musical piece would be abstracted by listeners. This prototype might encapsulate features such as the presence of certain motives, chord progressions or contrasts among different musical elements. In this scenario, new items will be then judged in relation to the prototype, forming gradients of category membership (Rosch & Mervis, 1975).

In the context of version communities, we hypothesize that the aforementioned gradients of category membership, in most cases, may point to the original song, i.e. the one which was released first[14]. In particular, we conjecture that, in one way or another, all song versions inherit some characteristics from this 'original prototype'. This feature, combined with the fact that new versions may also be inspired by other renditions, leads us to infer that the original song occupies a *central* position within a version community, being a referential or 'best example'.

To evaluate this hypothesis we manually check for original versions in setup 3 and discard the sets that do not have an original, i.e. the ones where the oldest song we have was not performed by the original artist. We find 426 originals out of 523 version sets. Throughout this section, we employ the directed weighted graph defined by the asymmetric matrix $\mathcal{D}$ (Sec. 4.2.2).

Initial supporting evidence that the original song is central within its commu-

[14] We want to avoid making subjective judgments about a song's popularity with regard to its versions.

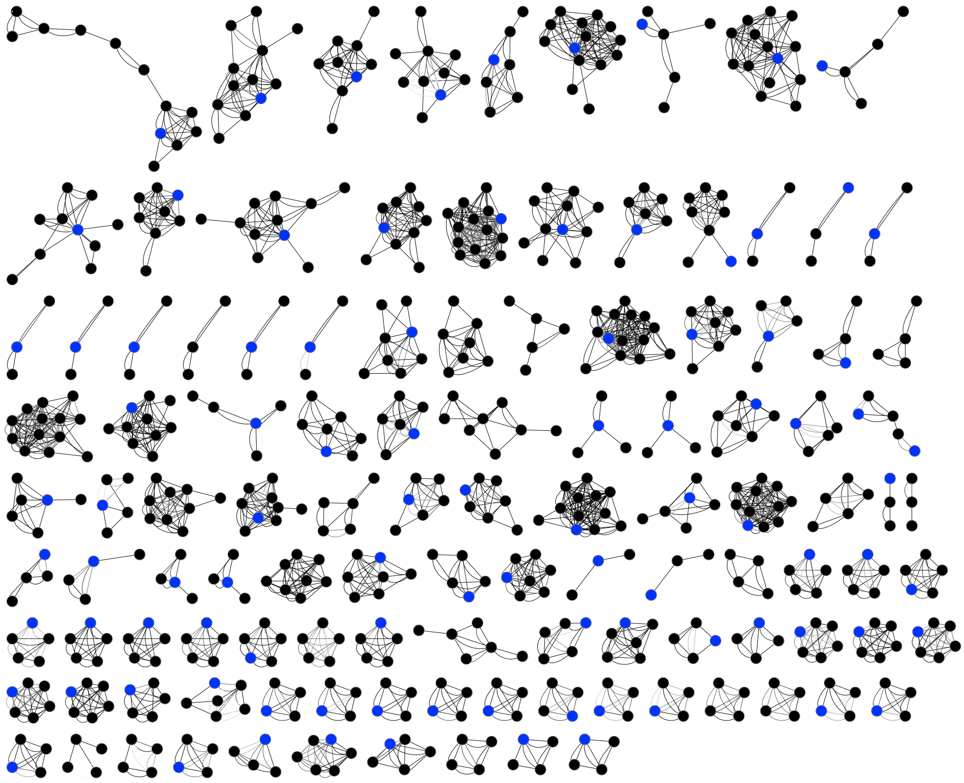


Figure 4.7: Graphical representation of the versions network with a strong threshold of 0.1. Original songs are drawn in blue, while other versions are in black.

nity is given by Figs. 4.7 and 4.8. In Fig. 4.7, we depict the resulting network after the application of a strong threshold (only using $d_{u,v} \leq 0.1$). We see that communities are well defined and also that many of the original songs are usually 'the center' of their communities. In Fig. 4.8, two cumulative distributions have been calculated: one for the weights of links exiting an original song (performed by the original artist, black solid line), and one for links exiting versions (performed by the original or another artist a posteriori from the original recording, blue broken line). The plotting of these cumulative distributions indicates that original songs tend to be connected to other nodes through links with smaller weights, that is, shorter distances (or higher similarities). The fact that the original song occupies a central position can be also observed qualitatively with the online demo of the system (Appendix A).

To evaluate the aforementioned hypothesis in a more formal way, we propose a study of the ability to automatically detect the original version within a community of versions. To this extent, we consider an ideal community detection algorithm (i.e. an algorithm detecting version communities with no false

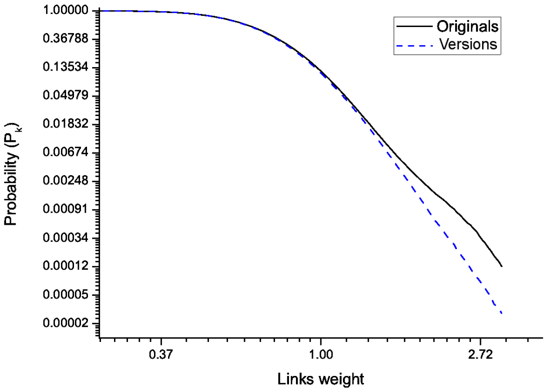


Figure 4.8: Cumulative weights distributions for links in the network, divided between links outgoing from an original song (black solid line) and from a song version (blue broken line) songs.

positives and no false negatives) and propose two different methods. These methods are based on the structure of weights of the obtained sub-network after the ideal community detection algorithm has been applied.

Closeness centrality This algorithm estimates the centrality of a node by calculating the mean path length between that node, and any other node in the sub-network (Barrat et al., 2004; Boccaletti et al., 2006). Note that the sub-network is fully connected, as no threshold has been applied in this phase. Therefore, the shortest path is usually the direct one. Mathematically, let $\mathcal{D}^{(k)}$ denote the sub-network containing the k-th community. Then the index i of the original (or prototype) song $v_i^{(k)}$ of the k-th community corresponds to

$$i = \underset{1 \leq u \leq C_k}{\arg\min} \left(\sum_{\substack{v=1 \\ v \neq i}}^{C_k} d_{u,v}^{(k)} \right), \tag{4.7}$$

where C_k is the cardinality of the k-th community. Notice that a similar methodology is employed in the clustering context to infer the medoid of a cluster (Jain et al., 1999; Xu & Wunsch II, 2009). Indeed, this was the initial strategy we followed in Serrà et al. (2009b).

MST centrality In this second algorithm we reinforce the role of central nodes. First, we calculate the minimum spanning tree (MST) for the sub-network under analysis (Costa et al., 2007). After that, we apply the previously described closeness centrality [Eq. (4.7)] to the resulting graph.

The results in Table 4.5 show the percentage of hits and misses for the detection of original songs in dependence of the cardinality C_k of the considered

Algorithm	C_k					
	2	3	4	5	6	7
Null hypothesis	50.0	33.3	25.0	20.0	16.7	14.3
Closeness centrality	59.4**	53.6**	43.1*	60.5**	48.0**	27.2
MST centrality	50.0	52.4**	60.7**	52.6**	48.0**	63.6**
U_S	*190*	*82*	*51*	*38*	*25*	*11*

Table 4.5: Percentage of hits and misses for the original song detection task depending on the cardinality C_k of the version communities. The * and ** symbols denote statistical significance at $p < 0.05$ and $p < 0.01$, respectively. The last line shows U_S, i.e. the number of communities for each cardinality.

community. We report results for C_k between 2 and 7 (the cardinalities for which our music collection has a representative number of communities U_C). The percentage of hits and misses can be compared to the null hypothesis of randomly selecting one song in the community.

We observe that, in general, accuracies are around 50% and, in some cases, they reach values of 60%. An accuracy of exactly 50% is obtained with $C_k = 2$ by both the null hypothesis and the MST centrality algorithm. This is because the MST is defined undirected, and there is no way to discriminate the original song in a sub-network of two nodes. As soon as $C_k > 2$, accuracies become substantially higher than the null hypothesis and statistical significance arises. Statistical significance is assessed with the binomial test (Kvam & Vidakovic, 2007).

4.6 Discussion and conclusion

In this chapter we build and analyze a musical network that reflects communities, where vertices correspond to different audio recordings and links between them represent the measure of resemblance between their musical (tonal) content. Moreover, we analyze the possibility of using such a network to apply different clustering and community detection algorithms to detect coherent groups of versions. Apart from considering a number of common approaches, three new alternatives for community detection are proposed. These alternatives achieve comparable accuracies to existing state-of-the-art methods, with similar or even faster computation times. In addition, we discuss a particular outcome from considering version communities, namely the analysis of the role of the original song within its versions. We show that the original song tends to occupy a central position within its group and, therefore, that a measure of centrality can be used to discriminate original songs from versions when the sub-network of these communities is considered. To the best of the authors' knowledge, the present work is the first attempt in this direction.

In the light of these results, complex networks stand as a promising research line within the specific task of version detection; but, at the same time, the proposed approach can be applied to any query-by-example system (Baeza-Yates & Ribeiro-Neto, 1999; Manning et al., 2008), and specially to other query-by-example MIR systems (Casey et al., 2008b; Downie, 2008).

In order to mitigate the confusions found in Sec. 4.4.6 we feel that the post-processing strategy we propose should be combined with some pre-processing ones. In particular, we hypothesize that, by considering descriptions of different musical facets, one could partition communities of more than one version set. We have seen that, for example, a common chord progression was the most remarkable musical facet between the elements of big communities joining two or more version sets. Therefore, the consideration of, for example, melodies or tempo-invariant rhythm descriptions could provide some informed ways of breaking down these communities of multiple sets. Such new descriptions could also be exploited in the case of incomplete communities, i.e. communities that do not contain the entire set of versions of the same piece. In general, more research is needed with regard to the combination of pre- and post-processing strategies. We have discussed individual pre- and post-processing strategies in Sec. 2.3.2. However, their combination still remains an open issue.

Finally, we should notice that some of the optimal thresholds for accuracy increments do not necessarily need to be the same as the ones used in version set detection. This therefore implies that the best performing methods for version set detection do not necessarily correspond with those achieving the highest accuracy increments (Secs. 4.4.2 and 4.4.3). In particular, the role of false positives becomes important due to the definition of $\hat{\mathcal{D}}$ [Eq. (4.2)]: false positives will be ranked higher than false negatives independently of their previous rank (see below). Furthermore, due to the use of different evaluation metrics, small changes in the optimal parameters could take place.

To illustrate the above reasoning, namely that the role of false positives determines different accuracies in the tasks of group detection and accuracy increment, consider the following example. Suppose the first items of the ranked answer to a concrete query $\mathring{u}_i$ are $\Lambda_u = \{\mathring{v}_j, v_k, \mathring{v}_l, v_m, \ldots\}$, where $\mathring{v}$ indicates effective membership to the same version group. Now suppose that clustering algorithm CA1 selects songs $\mathring{u}_i$, $\mathring{v}_j$, v_k and v_m as belonging to the same cluster, and that clustering algorithm CA2 selects $\mathring{u}_i$, $\mathring{v}_j$ and v_m. Both clustering algorithms would have the same recall $\bar{R}$ but CA2 will have a higher precision $\hat{P}$, and therefore higher accuracy value F [Eqs. (4.3-4.5)]. On the other hand, when evaluating Δ [Eq. (4.6)], CA2 will take a lower $\left\langle \overline{\psi}(\hat{\mathcal{D}}) \right\rangle$ value than CA1 (and thus a lower Δ) since v_m will be ranked higher than $\mathring{v}_l$ [Eq. (4.2)]. In summary: the clustering and community detection algorithms giving better community detection *and* more *suitable* false positives will achieve the highest increments.

CHAPTER 5

Towards model-based version detection

5.1 Introduction

A major characteristic that is largely shared among state-of-the-art approaches for version detection is the lack of specific modeling strategies for descriptor time series (Sec. 2.3). This is somehow surprising since, apart from benefits related to the generality and the compactness of the description, a modeling strategy could bring some light to the underlying dynamics of descriptor time series. In the present chapter we proceed in this direction by introducing a model-based system for version detection. In particular, we study a model-based forecasting approach, where we employ the concept of cross-prediction error. We now elaborate on this aspect based on Serrà et al. (2010b) and Serrà et al. (2010c).

Our approach essentially consists of first training a model to learn the characteristics of a query song's descriptor time series, and then assessing the predictions of the model when a target time series of a candidate song is considered. Intuitively, once a model has learned the patterns found in the time series of a given query song, one would expect the average prediction error to be relatively small when the time series of a candidate version is used as input. Otherwise, i.e. when an unrelated (non-version) candidate song is considered, the prediction error should be higher (provided that we use a suitable descriptor).

Although music descriptor time series are commonplace within the MIR community, little research has been done with regards to music modeling and forecasting using these time series as a starting point (bottom-up or data-driven approaches; Dubnov, 2006; Dubnov et al., 2007; Hazan et al., 2009). In fact, many strategies start from musical knowledge and test whether the observed data are consistent with the models (top-down or knowledge-driven approaches). In general, these top-down approaches are basically probabilistic (Abdallah & Plumbey, 2009; Eerola et al., 2002; Pachet, 2002; Paiement et al., 2009) and only consider melodic, simple, synthetic and/or few musical examples (Abdallah & Plumbey, 2009; Eerola et al., 2002; Paiement et al., 2009). Furthermore, they usually focus on scores or symbolic data (Abdallah & Plumbey, 2009; Eerola et al., 2002; Pachet, 2002), thus leaving aside many important aspects of the musical context and the specific rendition that can be captured by the music descriptor time series. For a quantitative characterization of descriptor time series, simple statistical moments, autoregressive modeling, or nonlinear time series analysis techniques have been used (Joder et al., 2009; Meng et al., 2007; Mierswa & Morik, 2005; Mörchen et al., 2006b; Serrà et al., 2009a).

In this chapter we take a bottom-up (data-driven) approach starting from music descriptor time series. Thereby we implicitly consider music recordings as the output of dynamical systems from which corresponding descriptor time series are recorded. We explore a number of popular modeling strategies from the linear and nonlinear time series analysis fields (Box & Jenkins, 1976; Kantz & Schreiber, 2004; Lütkepohl, 1993; Van Kampen, 2007; Weigend & Gershenfeld, 1993). We assess the out-of-sample cross-prediction capabilities of these strategies by training with one song's descriptor time series and testing against other songs with potentially similar musical content (a potential version).

We see that a model characterizing music descriptor time series allows for a simplified but still useful image of what is sequentially happening in a song's musical facet. In particular, we demonstrate that the concept of cross-prediction error can be effectively used for version detection. We show that the approach is very promising in the sense that it achieves competitive accuracies and furthermore provides additional advantages when compared to state-of-the-art approaches (such as lower computational complexities and potentially less storage requirements). Perhaps the most interesting aspect of the proposed approach is that no parameters need to be adjusted. More specifically, models' parameters and coefficients are automatically learned for each song and each descriptor time series individually (no intervention of the user is needed). Accordingly, the system can be readily applied to different music collections or descriptor time series.

5.2 Method

5.2.1 Overview

A brief overview of the model-based approach and the resulting structure of this chapter can be outlined as follows (Fig. 5.1). First, we extract tonal descriptor time series and perform transposition (Sec. 5.2.2). Then, a model is trained on the samples of a query song u. To do so we preliminarily perform state space embedding (Sec. 5.2.3). We study several time series models, both linear and nonlinear (Sec. 5.2.4). For each model, a number of parameter combinations are tested and the combination that achieves a lower in-sample (self-) prediction error is kept (Sec. 5.2.5). Indeed, by choosing the best parameter

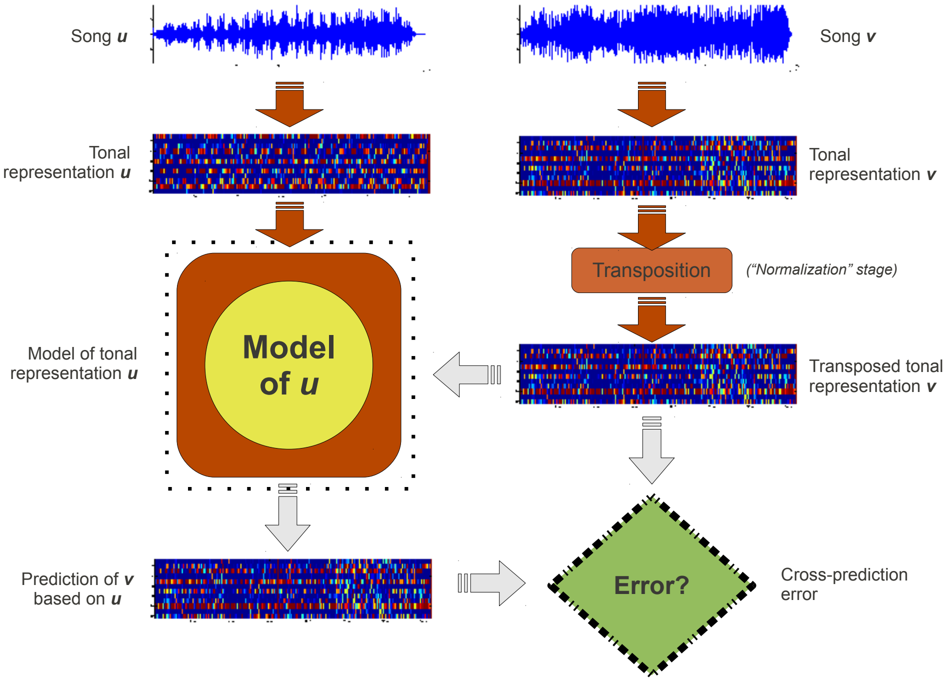


Figure 5.1: General block diagram of the model-based approach.

combination for each time series of each individual piece, we are already performing a partial modeling of the time series[1]. Next, we test the out-of-sample cross-prediction capabilities of the learned model (the model of query song u) on the samples of a candidate song v and compute the error done in this prediction (Sec. 5.2.6). This cross-prediction error is finally regarded as an indicator of version similarity.

We evaluate the approach following a similar methodology than the one explained in Chapter 3 (Sec. 5.3) and report the out-of-sample version retrieval accuracies for each model (Sec. 5.4). A brief discussion section is included in order to weigh the advantages of the proposed model-based approach (Sec. 5.5). In closing, we briefly summarize the achievements and propose some lines for further research (Sec. 5.6).

[1] We believe that the modeling of a time series is not only determined by the actual coefficients that we learn, but also by the parameters of the model themselves. We explain this aspect in detail in Sec. 5.2.5.

5.2.2 Descriptor extraction and transposition

The descriptor time series used in this chapter are the same as the ones explained in Chapter 3 (Sec. 3.2.2). We use PCP, TC and HC time series, which were denoted as $\bar{\mathcal{H}}$, $\bar{\mathcal{C}}$ and $\bar{\mathbf{g}}$, respectively. The only difference is that now the downsampling factor of these time series is lower, thus we have more samples to train our models. Specifically, we use a downsampling factor $\nu = 5$, which implies that the hop and frame sizes become 117 and 186 ms, respectively (see Sec. 3.2.2).

The way to achieve transposition invariance is again the same as explained in Chapter 3 (Sec. 3.2.3). We transpose PCP time series before obtaining TC and HC descriptors and test the $O = 2$ most probable transposition indices. From now on we also employ the notation $\mathcal{X} = [\mathbf{x}_1 \cdots \mathbf{x}_N]^{\mathrm{T}}$ introduced in Sec. 3.2.4 to refer to a time series of descriptors.

5.2.3 State space embedding

All the models described hereafter aim at predicting the future states of dynamical systems based on their present states (Kantz & Schreiber, 2004). Since an isolated sample $\mathbf{x}_i$ may not contain the necessary information for a reliable prediction at some future time step t, one could consider information from past samples. As a notational representation of the present and recent past of a time series we use the concept of delay coordinate state space embedding, a tool which is routinely employed in nonlinear time series analysis and which we already used in Sec. 3.2.4 [Eq. (3.16)]. Therefore, following the same steps of that section, we obtain a reconstructed time series $\hat{\mathcal{X}} = \left[\hat{\mathbf{x}}_{\lambda+1} \ldots \hat{\mathbf{x}}_{\hat{N}}\right]$. Recall that $\lambda = (m-1)\tau$ denotes the embedding window, with m and τ being the embedding dimension and the time delay, respectively. If each column vector $\mathbf{x}_i$ has X components, representing the X-dimensional sample of the i-th frame, the embedding operation produces column vectors $\hat{\mathbf{x}}_i$, representing (mX)-dimensional samples.

5.2.4 Time series models

To model and predict music descriptor time series we employ popular, simple, yet flexible time series models; both linear and nonlinear (Box & Jenkins, 1976; Kantz & Schreiber, 2004; Lütkepohl, 1993; Van Kampen, 2007; Weigend & Gershenfeld, 1993). Since we do not have a good and well-established model for music descriptor prediction, we try a number of standard tools in order to identify the most suitable one. All modeling approaches we employ have clearly different features. Therefore they are able to exploit, in a forecasting scenario, different structures that might be found in the data. In particular, in the case of music, we could expect them to exploit repetitions and transitions at multiple levels (notes, motifs, phrases, sections, etc.). As a linear approach we consider autoregressive models. Nonlinear approaches include locally constant, locally linear, globally nonlinear and probabilistic predictors.

Autoregressive models

A widespread way to model linear time series data is through an autoregressive (AR) process, where predictions are based on a linear combination of m previous measurements (Box & Jenkins, 1976). We here employ a multivariate AR model (Lütkepohl, 1993). In particular, we first construct delay coordinate state space vectors $\hat{\mathbf{x}}_i$ and then express the forecast $\tilde{\mathbf{x}}_{i+t}$ at t steps ahead from the i-th sample $\mathbf{x}_i$ as

$$\tilde{\mathbf{x}}_{i+t} = \mathcal{A}\,\hat{\mathbf{x}}_i, \tag{5.1}$$

where $\mathcal{A}$ is the $X \times mX$ coefficient matrix of the multivariate AR model. By considering samples $i = \lambda + 1, \ldots \hat{N} - t$, one obtains an overdetermined system

$$\tilde{\mathcal{X}}^{\mathrm{T}} = \mathcal{A}\,\hat{\mathcal{X}}^{\mathrm{T}} \tag{5.2}$$

which, by ordinary least squares fitting, allows the estimation of $\mathcal{A}$ (Press et al., 1992).

Threshold autoregressive models

Threshold autoregressive (TAR) models generalize AR models by introducing nonlinearity (Tong & Lim, 1980). A single TAR model consists of a collection of AR models where each single one is valid only in a certain domain of the reconstructed state space (separated by the "thresholds"). This way, points in state space are grouped into patches, and each of these patches is used to determine the coefficients of a single AR model (piecewise linearization).

For determining all TAR coefficients we partition the reconstructed space formed by $\hat{\mathcal{X}}$ into K non-overlapping clusters with a K-medoids algorithm (Parka & Jun, 2009) and determine, independently for each partition, AR coefficients as above [Eqs. (5.1) and (5.2)]. Importantly, each of the K AR models is associated to the corresponding cluster. When forecasting, we again construct delay coordinate state space vectors $\hat{\mathbf{x}}_i$ from each input sample $\mathbf{x}_i$, calculate their squared Euclidean distance to all $k = 1, \ldots K$ cluster medoids and forecast

$$\tilde{\mathbf{x}}_{i+t} = \mathcal{A}^{(k')}\,\hat{\mathbf{x}}_i, \tag{5.3}$$

where $\mathcal{A}^{(k')}$ is the $X \times mX$ coefficient matrix of the multivariate AR model associated to the cluster whose medoid is closest to $\hat{\mathbf{x}}_i$.

Radial basis functions modeling

A very flexible class of global nonlinear models are commonly called radial basis functions (RBF; Broomhead & Lowe, 1988). As with TAR models, one partitions the reconstructed state space into K clusters but, in contrast, a scalar RBF function ϕ is used for forecasting such that

$$\tilde{\mathbf{x}}_{i+t} = \mathbf{a}_0 + \sum_{k=1}^{K} \mathbf{a}_k \ \phi\left(\|\hat{\mathbf{x}}_i - \mathbf{b}_k\|\right), \tag{5.4}$$

where $\mathbf{a}_k$ are coefficient vectors, $\mathbf{b}_k$ are the cluster centers and $\| \ \|$ is a norm. In our case we use the cluster medoids for $\mathbf{b}_k$, the Euclidean norm for $\| \ \|$ and a Gaussian RBF function

$$\phi\left(\|\hat{\mathbf{x}}_i - \mathbf{b}_k\|\right) = e^{-\frac{\|\hat{\mathbf{x}}_i - \mathbf{b}_k\|^2}{2\theta\rho_k}}. \tag{5.5}$$

We partition the space formed by $\hat{\mathcal{X}}$ with the K-medoids algorithm as above, set ρ_k to the mean distance found between the elements inside the k-th cluster and leave θ as a parameter. Notice that for fixed centers $\mathbf{b}_k$ and parameters ρ_k and θ, determining the model coefficients becomes a linear problem that can be resolved again by ordinary least squares minimization. Indeed, a particularly interesting remark about RBF models is that they can be viewed as a (non-linear, layered, feed-forward) neural network where a globally optimal solution is found by linear fitting (Broomhead & Lowe, 1988; Weigend & Gershenfeld, 1993). In our case, for samples $i = \lambda + 1, \dots \hat{N} - t$, we are left with

$$\tilde{\mathcal{X}}^{\mathrm{T}} = \mathcal{A} \ \Phi, \tag{5.6}$$

where $\mathcal{A} = [\mathbf{a}_0 \mathbf{a}_1 \dots \mathbf{a}_K]$ is now an $X \times (K+1)$ coefficient matrix and $\Phi = \left[\mathit{\Phi}_{\lambda+1} \dots \mathit{\Phi}_{\hat{N}-t}\right]$ is a transformation matrix formed by column vectors

$$\mathit{\Phi}_i = \left(1, \phi\left(\|\hat{\mathbf{x}}_i - \mathbf{b}_1\|\right), \dots \phi\left(\|\hat{\mathbf{x}}_i - \mathbf{b}_K\|\right)\right)^{\mathrm{T}}. \tag{5.7}$$

Locally constant predictors

A zeroth-order approximation to the time series is given by a locally constant predictor (Farmer & Sidorowich, 1987). With this predictor, one first determines a neighborhood $\mathit{\Omega}_i$ of radius ϵ around each point $\hat{\mathbf{x}}_i$ of the reconstructed time series $\hat{\mathcal{X}}$. Then forecasts

$$\tilde{\mathbf{x}}_{i+t} = \frac{1}{|\mathit{\Omega}_i|} \sum_{\mathbf{x}_j \in \mathit{\Omega}_i} \mathbf{x}_{j+t}, \tag{5.8}$$

where $|\mathit{\Omega}_i|$ denotes the number of elements in $\mathit{\Omega}_i$. Notice that the unreconstructed versions $\mathbf{x}_j$ of the neighbors of $\hat{\mathbf{x}}_i$ are used.

In our experiments, ϵ is set to a percentage ϵ_κ of the mean distance between all state space points $\hat{\mathcal{X}}$ (we use the squared Euclidean norm). In addition, we require $|\mathit{\Omega}_i| \geq \eta$, i.e. a minimum of η neighbors is always included independently of their distance to $\hat{\mathbf{x}}_i$. Notice that this is almost a model-free approach with no coefficients to be learned: one just needs to set parameters m, τ, ϵ_κ and η.

Model	Parameter	Values
All	m	1, 2, 3, 5, 7, 9, 12, 15
All	τ	1, 2, 6, 9, 15
TAR & RBF	K	1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 15, 20, 30, 40, 50
RBF	θ	0.5, 0.75, 1, 1.25, 1.5, 2, 2.5, 3, 3.5, 4, 5, 7, 9
Locally constant	ϵ_κ	0.01, 0.025, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8
Locally constant	η	2, 5, 10, 15, 25, 50
Naïve Markov	K_i	8, 15, 30, 40, 50, 60, 70
Naïve Markov	K_o	5, 10, 20, 30, 40, 50

Table 5.1: Parameter values used for grid search.

Naïve Markov models

This approach is based on grouping inputs $\hat{\mathcal{X}}$ and outputs $\tilde{\mathcal{X}}$ into K_i and K_o clusters, respectively (Van Kampen, 2007). Given this partition, we fill in a $K_\mathrm{i} \times K_\mathrm{o}$ transition matrix $\mathcal{P}$, whose elements $p_{k_\mathrm{i},k_\mathrm{o}}$ correspond to the probability of going from cluster k_i of $\hat{\mathcal{X}}$ to cluster k_o of $\tilde{\mathcal{X}}$ (i.e. the rows of $\mathcal{P}$ sum up to 1). Then, when forecasting, a state space reconstruction $\hat{\mathbf{x}}_i$ of the input $\mathbf{x}_i$ is formed and the distance towards all K_i input cluster medoids is calculated.

In order to evaluate the performance of the Markov predictor in the same way as the other predictors, we use $\mathcal{P}$ to construct a deterministic output in the following way:

$$\tilde{\mathbf{x}}_{i+t} = \sum_{k_\mathrm{o}=1}^{K_\mathrm{o}} p_{k'_\mathrm{i},k_\mathrm{o}} \, \mathbf{b}_{k_\mathrm{o}}, \tag{5.9}$$

where $\mathbf{b}_{k_\mathrm{o}}$ denotes the medoid of (output) cluster k_o and k'_i is the index of the (input) cluster whose medoid is closest to $\hat{\mathbf{x}}_i$.

5.2.5 Training and testing

All previous models are completely described by a series of parameters (m, τ, K, θ, ϵ_κ, η, K_i, or K_o) and coefficients ($\mathcal{A}$, $\mathcal{A}^{(k)}$, $\mathcal{P}$, $\mathbf{b}_k$, or ρ_k). In our experiments, these values are learned independently for each song *and* descriptor using the entire time series as training set. This learning is done in an unsupervised way, with no prior information about parameters and coefficients. More specifically, for each song *and* descriptor time series we calculate the corresponding model coefficients for different parameter configurations and then select the solution that leads to the best in-sample approximation of the data. We perform a grid search for each possible combination that results from Table 5.1 on each model.

Since we aim at obtaining compact descriptions of our data and we want to avoid overfitting, we limit the total number of model parameters and coefficients to be less than 10% of the total number of values of the time series data.

This implies that parameter combinations leading to models with more than $(N \times X)/10$ values are automatically discarded at the training phase[2]. We also force an embedding window $\lambda < N/20$.

Intuitively, with such a search for the best parameter combination for a specific song's time series, part of the time series modeling is also done through the appropriate parameter setting, since m, τ and K are parameters that also define time series' characteristics (Kantz & Schreiber, 2004). Notice that the prediction horizon t cannot be optimized in-sample since best approximations would always correspond to $t = 1$ due to inherent sample correlations. The impact of t can only be assessed on the out-of-sample prediction, when the model is applied to the candidate song.

5.2.6 Prediction error

To evaluate prediction accuracy we use a normalized mean squared error measure (Weigend & Gershenfeld, 1993), both when training our models (to select the best parameter combination) and when retrieving versions based on cross-prediction. We define this error as

$$\xi = \frac{1}{N-t-\lambda} \sum_{i=\lambda+1}^{N-t} \frac{1}{X} \sum_{j=1}^{X} \frac{\left(\tilde{x}_{i+t,j} - x_{i+t,j}\right)^2}{{\sigma_j}^2}, \tag{5.10}$$

where ${\sigma_j}^2$ is the variance of the j-th descriptor component over all samples $i = \lambda + t + 1, \dots N$ of the target time series $\mathcal{X}$. We use the notation $\xi_{u,u}$ when a model trained on song u is used to forecast further frames of song u (self-prediction, in-sample error) and $\xi_{u,v}$ when a model trained on song u is used to forecast frames of song v (cross-prediction, out-of-sample error).

5.3 Evaluation methodology

5.3.1 Music collection and evaluation measure

The music collection we employ here is the same used in the other parts of the thesis (Sec. 3.3.1). In particular we use MC-102 and MC-2125. To evaluate the accuracy in identifying song versions we proceed exactly as in Chapter 3. Given a music collection with U songs, we calculate $\xi_{u,v}$ for all $U \times U$ possible pairwise combinations and then create a symmetric dissimilarity matrix $\mathcal{D}$, whose elements are $d_{u,v} = \xi_{u,v} + \xi_{v,u}$. Once $\mathcal{D}$ is computed, we use the mean of average precisions measure $\langle\overline{\psi}\rangle$ to evaluate version retrieval (Sec. 3.3.2).

[2] Of course this does not apply to the locally constant predictor, which, as already said, is an almost model-free approach.

5.3.2 Baseline predictors

Besides models in Sec. 5.2.4, we further assess our results with a set of baseline approaches that do not require parameter adjustments nor coefficient determination.

Mean

The prediction is simply the mean of the training data:

$$\tilde{\mathbf{x}}_{i+t} = \boldsymbol{\mu}, \tag{5.11}$$

$\boldsymbol{\mu}$ being a column vector. This predictor is optimal in the sense of Eq. (5.10) for i.i.d. time series data. Notice that, by definition, $\xi = 1$ when predicting with the mean of the time series data. In fact, ξ allows an estimation, in a variance percentage, of how our predictor compares to the baseline prediction given by Eq. (5.11).

Persistence

The prediction corresponds to the current value:

$$\tilde{\mathbf{x}}_{i+t} = \mathbf{x}_i. \tag{5.12}$$

This prediction yields low ξ values for processes that have strong correlations at t time steps.

Linear trend

The prediction is formed by a linear trend based on the current and the previous samples:

$$\tilde{\mathbf{x}}_{i+t} = 2\mathbf{x}_i - \mathbf{x}_{i-1}. \tag{5.13}$$

This is suitable for a smooth signal and a short prediction horizon t.

5.4 Results

In the work we reported in Serrà et al. (2010c) we saw that the prediction horizon t had an important impact on the system's performance, so we decided to study the accuracy $\langle\overline{\psi}\rangle$ for different t values with MC-102 (Fig. 5.2). We see that, except for the locally constant predictor, all models perform worse than the mean predictor for short horizons ($t \leq 3$). This performance increases with the horizon ($4 \leq t \leq 7$), but reaches a stable value for mid-term and relatively long horizons ($t > 7$), which is much higher than the mean predictor performance. In general, the maximal accuracy is obtained for $t = 19$, although it is not substantially different than accuracies reached for $t > 7$ (recall that $t = 1$ corresponds to 117 ms).

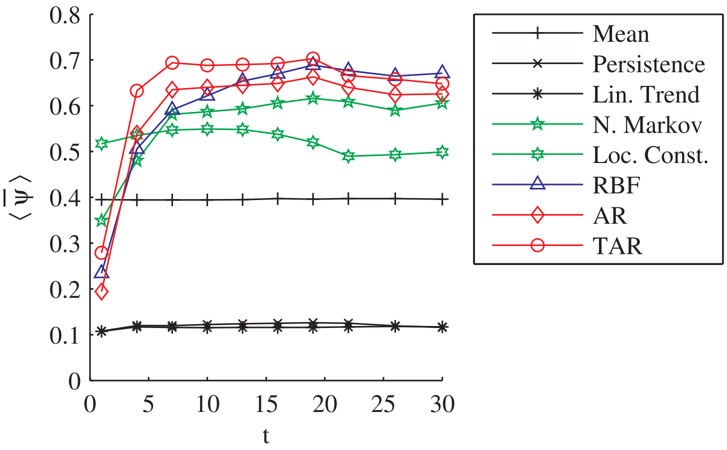


Figure 5.2: Mean of average precisions $\langle\overline{\psi}\rangle$ depending on the prediction horizon t. Results for the TC descriptor with all considered models (MC-102). PCP and HC time series yield qualitatively similar plots.

The ability to perform reliable cross-predictions at long horizons is, of course, related to the ability of the learned model to perform (self-) predictions at such a time span. To assess this latter ability we studied the self-prediction error $\xi_{u,u}$ as a function of the forecast horizon t (Serrà et al., 2010c). In general, we saw that $\xi_{u,u}$ increased rapidly for $t \leq 4$ but, surprisingly, it reached a stable plateau with all descriptors for $t > 10$, i.e. for prediction horizons of more than 1 s. Notably, in this plateau, $\xi_{u,u} < 1$. This indicated that, on average, there was a certain capability for the models to still perform predictions at relatively long horizons, and that these predictions were better than predicting with the mean. Overall, the previous fact reveals that descriptor time series are far from being i.i.d. data (even at relatively long t) and that models are capturing part of the long-term structures and repetitions found in our collection's recordings. We conjecture that these two facts play a crucial role in the cross-prediction scenario, allowing the correct detection of versions. For more details concerning the in-sample self-prediction capabilities of the considered models we refer the reader to Serrà et al. (2010c).

The fact that we detect versions better at mid-term and relatively long horizons could also have a musicological explanation. To see this we study matrices quantifying the transition probabilities between states separated by a time interval corresponding to the prediction horizon t. We first cluster a time series $\mathcal{X}$ into, for instance, 10 clusters and compute the medoids. We subsequently fill a transition matrix $\mathcal{P}$, with elements $p_{i,j}$. Here i and j correspond to the indices of the medoids to which respectively $\mathbf{x}_i$ and $\mathbf{x}_{i+t}$ are closest. This transition matrix is normalized so that each row adds up to 1. In Fig. 5.3 we show $\mathcal{P}$ for three different horizons ($t = 1$ in the first column, $t = 7$ in

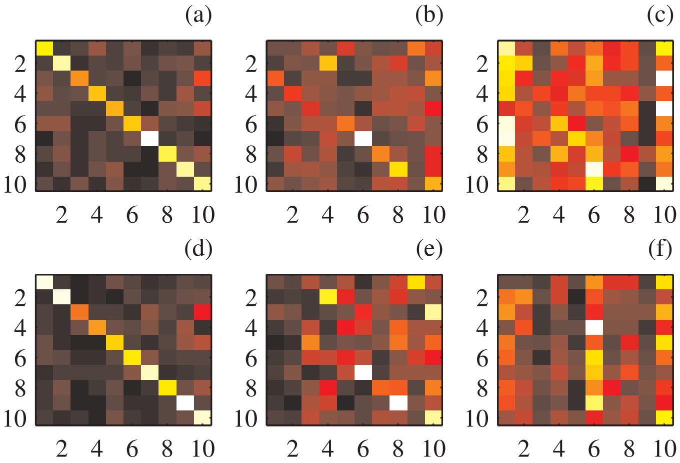


Figure 5.3: Transition matrices $\mathcal{P}$ for two versions (top) and two unrelated songs (bottom) using 10 input and 10 output clusters (see text). These transition matrices are computed for $t = 1$ (a,d), $t = 7$ (b,e) and $t = 15$ (c,f). Bright colors correspond to high transition probabilities (white and yellow patches).

the second column and $t = 15$ in the third column). Two unrelated songs are shown (one row each). The musical piece that provided the cluster medoids to generate $\mathcal{P}$ is a version of the first song (top row) but not of the second (bottom row).

We see that, for $t = 1$, $\mathcal{P}$ is highly dominated by persistence to the same cluster, both for the version (Fig. 5.3a) and the non-version (Fig. 5.3d) pair. This fact was also corroborated with the self-prediction results of the persistence-based predictor (Serrà et al., 2010c). Once t increases, characteristic transition patterns arise, but the similarity between matrices in Fig. 5.3b and 5.3e show that these patterns are not characteristic enough to define a musical piece. Compare for example the high values obtained for both matrices (b) and (e) at $p_{7,6}$, $p_{9,8}$, $p_{2,4}$, $p_{1,9}$, or $p_{3,10}$. We conjecture that these transitions define general musical features that are shared among a large number of subsets of recordings, not necessarily just the versions. For example, it is clear that there are general rules with regard to chord transitions, with some particular transitions being more likely than others (Krumhansl, 1990). Only when $t > 7$ transitions that discriminate between the dynamics of songs start to become apparent (see the distinct patterns in Figs. 5.3c and 5.3f). This distinctiveness can then be exploited to differentiate between versions and non-versions.

Results for version retrieval with MC-2125 indicate that the best model is the TAR model; although notable accuracies are achieved with the RBF method (Table 5.2). The AR and the naïve Markov models come next. Persistence and linear trend predictors perform at the level of the random baseline $\langle\overline{\psi}\rangle_{\text{null}}$. This is to be expected since no learning is performed for these predictors.

Methods	Descriptors		
	PCP	TC	HC
Linear trend	<0.01	<0.01	<0.01
Persistence	<0.01	<0.01	<0.01
Mean	0.15	0.09	0.01
Locally constant	0.25	0.28	0.05
Naïve Markov	0.37	0.38	0.05
AR	0.37	0.41	0.04
RBF	0.38	**0.44**	0.05
TAR	**0.39**	**0.44**	**0.06**

Table 5.2: Mean of average precisions $\langle\overline{\psi}\rangle$ for the version identification task (MC-2125). A prediction horizon of $t = 19$ was used. The maximum of the random baseline $\langle\overline{\psi}\rangle_{\text{null}}$ was found to be 0.008 within 99 runs.

In addition, we see that the HC descriptor is much less powerful than the other two. This is to be expected, since HC compresses tonal information to a univariate value. Furthermore, HC might be less informative than PCP or TC values themselves, which already contain the change information in their temporal evolution. Apart from this, we see that TC descriptors perform better than PCP descriptors. This does not necessarily imply that TC descriptors provide a better representation of the tonal information that is present in a recording, but that TAR models are better in capturing the essence of their temporal evolution.

5.5 Discussion

Even though the considered models yield a significant accuracy increase when compared to the baselines, it might still seem that a value of $\langle\overline{\psi}\rangle$ around 0.4 in an evaluation measure that ranges between 0 and 1 is not a big success for a version identification approach. To properly asses this accuracy one has to compare it against the accuracies of state-of-the-art approaches.

According to MIREX, the best accuracy achieved until the moment of writing this thesis for the version identification task was obtained with the previous model-free system of Chapter 3. This system, without any post-processing step, reaches $\langle\overline{\psi}\rangle = 0.66$ with the MIREX dataset and yields $\langle\overline{\psi}\rangle = 0.70$ with MC-2125 (Sec. 3.4.2). A former method by Serrà et al. (2008b) scored $\langle\overline{\psi}\rangle = 0.55$ with the MIREX data. Thus the cross-prediction approach does not outperform these methods. However, the cited methods were specifically designed for the task of identifying versions, while the cross-prediction approach is a general schema that does not incorporate all the specific modifications that could be beneficial for such a task (e.g. it does note take into account tempo or structural changes between versions, Sec. 2.3). To make further comparisons (at least qualitatively), one should note that $\langle\overline{\psi}\rangle$ values around 0.4 are in line with other state-of-the-art accuracies, or even better if we consider comparable music collections (see e.g. Table 2.2 in Sec. 2.3.3).

Beyond accuracy comparisons, some other aspects can be discussed. Indeed, another reason for appraising the solution obtained here comes from the consideration of storage capacities and computational complexities at the query retrieval stage. Since we limit our models to a size of 10% of the total number of training data (Sec. 5.2.5), they require 10% of the storage that would be needed for saving the entire time series (state-of-the-art systems usually store the full time series for each song). This fact could be exploited in a single-query retrieval scenario. In this setting, it would be sufficient to determine a dissimilarity measure ξ (Eq. 5.2.6) from the application of all candidates' models to the query song. Hence, only the models rather than the raw data would be required. Regarding computational complexity, many approaches for version identification are quadratic in the length of the time series, requiring at least an Euclidean distance calculation for every pair of sample points[3] (Sec. 2.3). Contrastingly, the approaches presented here are linear in the length of the time series. For example, with TAR models, we just need to do a pairwise distance calculation between the samples and the K medoids, plus a matrix multiplication and subtraction (notice that the former is not needed with AR models). If we compare the model-free approach of Chapter 3 with the TAR-based strategy by considering an average time series length $\bar{N}$, we have that the former is roughly $O\left(\bar{N}^2 mX\right)$, while the latter is $O\left(\bar{N}(K+X)mX\right)$, with $K+X \ll \bar{N}$. To illustrate this with specific numbers: with $\bar{N} = 2304$ (approximately 4 min of music), descriptor dimensionality $X = 12$ (the largest among PCP, TC and HC) and $K = 50$ (the maximum allowed, Table 5.1), we obtain a minimal relative speed improvement of $2304/(50+12) \approx 37$.

A further and very interesting advantage of using the approaches considered in this chapter is that no parameters need to be adjusted by the user. More specifically, models' parameters and coefficients are automatically learned for each song and descriptor time series individually by the minimization of the in-sample training error $\xi_{u,u}$. Usually, version identification algorithms have multiple parameters that can be dependent, for instance, on the music collection, the music descriptor time series, or the types of versions under consideration (Sec. 2.3). The model-free approach of Chapter 3 and our previous method of Serrà et al. (2008b) were not an exception: as there was no way to a priori set their specific parameters, these were set by trial and error with a representative (ideally out-of-sample) music collection. Since for the current approaches no such manual parameter optimization is required, their application to version identification is robust and straightforward.

[3] As examples we can mention the model-free approach of Chapter 3, or our previous method of Serrà et al. (2008b).

5.6 Conclusions and future work

In this chapter we explore a number of modeling strategies for version retrieval. In particular, we test a number of routinely employed time series models. These include linear and nonlinear predictors such as AR, TAR, RBF, locally constant and naïve Markov. These models are automatically trained for each song and descriptor time series individually. Training is done in an unsupervised way, performing a grid search over a set of parameter combinations and automatically determining the corresponding coefficients. We perform an in-sample self-prediction of the descriptor time series in order to assess which parameter combination gives the best approximation to the time series.

With the experiments above we demonstrate both the capacity of generalization of the considered models and the real-world applicability of out-of-sample cross-prediction errors. More specifically, we show that cross-predictions at mid-term and relatively long horizons permit the performance of version retrieval. In particular, AR, TAR and RBF methods achieve competitive accuracies.

In general, we see that considering cross-predictions of time series models leads to a parameter-free approach for version identification. Furthermore, the approach is fast, allows for reduced storage and still maintains a highly competitive accuracy when compared to state-of-the-art systems. Thus, time series modeling strategies stand as a really promising approach for version detection and, by extension, for music and multimedia retrieval in general.

Two important research lines stem from the work in the current chapter. First, it would be interesting to consider further time series models and to see how accurate they are in the version identification task. This partially points out the necessity of knowing more about the nature of a descriptor time series, an aspect which was initially assessed in Serrà et al. (2010c). In particular, our findings suggested that the temporal evolution of music descriptors might be explained by a concatenation of multiple autoregressive processes with superimposed noise. Interestingly, AR and TAR models yielded the lowest self-prediction errors (recall that they also reach the highest accuracies in the case of version retrieval presented in Sec. 5.4 above). In spite of the evidence found, we should be cautious since some contradictory evidence on the use of AR models for music descriptor time series exists. In particular, Meng et al. (2007) reported that AR modeling of descriptor time series was beneficial for genre classification, while Joder et al. (2009) reported that such a strategy was not useful for instrument classification.

The second research line to pursue is more practical and it is focused on the version retrieval task. Indeed, it would be important to see whether the accuracies achieved by a model-based approach can surpass the ones achieved by the best model-free approaches. In particular, there have been two important aspects missing in the formulation of our model-based approach: tempo and structure invariance. With regard to tempo invariance, we hypothesize that possibly working with tempo-insensitive representations of tonal information, such as the ones used e.g. by Ellis & Cotton (2007), could partially solve the problem. However, one should be careful in the beat-detection stage, since it could introduce additional errors to the system (Sec. 2.3). Notice that the introduction of a tempo invariant representation is only the most straightforward option and that further strategies can be devised, specially with the setting of the prediction horizon t and the time delay τ. With regard to structure invariance, the easiest way would be to cut the time series into short (maybe 30 s) overlapping segments and train different models on each segment. However, this solution would introduce additional computational costs since each error for each segment would need to be evaluated. Notice that both training and testing (error computation) phases should be appropriately 'tuned' in order to achieve structure invariance. Therefore, only modifying Eq. (5.2.6) to take the possibility of changes in the song structure into account would not be sufficient. Preliminary experiments with our data showed that the version retrieval accuracy was not increased when considering only the latter strategy. An additional issue with the overall structure invariance strategy is that of the number of samples that are needed to train a time series model. It could be the case that the time series samples found in 30 s may not be sufficient for a proper training.

CHAPTER 6

Summary and future perspectives

6.1 Introduction

Can a computer recognize the underlying musical composition behind a given interpretation? Can we automatically detect if two songs correspond to the same music piece despite many important musical variations? These were the kind of questions that motivated our research (Sec. 1.1). In the light of the results presented in this thesis we can now answer: *yes*. Certainly the solutions we propose are able to perform such tasks effectively in the majority of cases. Of course we should give a word of caution since our system is not 100% accurate. However, we have shown that part of the errors produced by the system are explainable from a perceptual and a musicological perspective (Secs. 3.4.5 and 4.4.6). Therefore, we could expect that humans would show similar mistakes than those observed in our systems.

We started this thesis with an introduction to automatic version identification, with a special focus on the context of music information retrieval, with some terminology remarks, and with the common musical variations between song versions (Chapter 1). The context of music information retrieval was further reviewed in our literature summary, emphasizing current approaches for version identification (Chapter 2). We then presented and evaluated our main approach for version identification, a model-free system (Chapter 3). We subsequently studied and assessed a post-processing strategy for the output of this system (Chapter 4). Finally, in contraposition to our model-free system, we presented and evaluated a model-based system which, although not outperforming the model-free one, had remarkable advantages (Chapter 5).

Towards the end of each chapter we have been providing the main conclusions regarding our work. These conclusions summarize in detail the work reported within each chapter, highlight relevant results and outcomes and comment on concrete aspects of each specific approach. Alternatively, in this chapter, more general or global statements are made. We end this dissertation with some brainstorming on future perspectives.

6.2 Summary of contributions

This thesis contributes to the processing, retrieval, organization and understanding of digital information, specifically multimedia information. More specifically, it strongly contributes to the field of audio content-based music information retrieval:

- It is, to the best of the author's knowledge, the first thesis entirely devoted to the topic of automatic audio-based version identification.
- It critically discusses version identification in the context of music information retrieval. This includes a critical assessment of current approaches and evaluation procedures.
- It provides a comprehensive overview of the scattered available literature on version identification based on the audio content. Specific emphasis is given to the main functional blocks that are needed to build a version identification system.
- It proposes a successful model-free approach for version identification. Noticeably, the quantification measures derived from this approach have many potential applications beyond music information retrieval (see below).
- It characterizes and exploits the output of a version identification system. In particular, it is shown that song versions of the same piece naturally cluster together, that these clusters can be effectively detected and that this information can be used to enhance the results of existing systems. Again, the application of the developed strategies goes beyond version retrieval.
- It explores the role that original songs play inside a group of versions, showing that there is a certain tendency for the original song to be central within the group.
- It explores model-based approaches for version identification. These approaches represent a very promising research line with regard to obtaining parameter-free systems that are fast, allow for reduced storage and are still competitive in accuracy.

In addition, it is worth noticing that the proposed model-free approach ($Q_{\max}$), together with its post-processed version ($Q^*_{\max}$), reached the two highest accuracies in the MIREX 2008 and 2009 editions of the "audio cover song identification task". At the moment of writing this thesis, the aforementioned accuracies remain the highest in all MIREX editions of said task (including the one in 2010). These accuracies clearly surpass those achieved by current state-of-the-art approaches, including a previous approach by the author which, at its time in 2007, achieved the highest MIREX score (Serrà et al., 2008b).

With regard to the general applicability of the proposed methods, it may be relevant to cite the words from the board member's report after reviewing our paper submitted to New Journal of Physics (Serrà et al., 2009a): "Both referees agree that the study is interesting. However the first referee does not think that the studying music retrieval problem is of interest to the readership of NJP. I do not agree with this view since much of our physics studies in recent years are applications of physics methods to multidisciplinary fields. I think that developing novel physical methods for automatic classification of digital information and in particular automatic identification of cover songs is of much interest. I therefore recommend accepting the paper in NJP". Time has shown that the board member was right: the paper was among the 10% most downloaded papers across all Institute of Physics[1] journals within the first month of publication[2].

The outcomes of the research carried out in this thesis have been published in the form of several papers in international conferences, journals and a book chapter. Some of these publications have been featured in a number of public and private communication media[3]. An online demo of the system was also presented at an international conference (Appendix A). Moreover, part of this research has been deployed into a commercial media broadcast monitoring service[4] by the company Barcelona Music and Audio Technologies and the author is inventor of two patents applied by the same company. The full list of the author's publications and patents is provided in an annex to this thesis (Appendix B).

6.3 Some future perspectives

Some new avenues for research have already been advanced in the last chapters of the thesis. For example, it is clear that further pre- and post-processing techniques can provide a valuable accuracy increase in current systems. An additional issue is that of the simultaneous combination of pre- and post-processing strategies.

With regard to pre-processing techniques, we are particularly optimistic about the combination of different sources of information. Indeed, there are different musical facets that can be shared within versions. Therefore different methods for extracting these 'essential' characteristics would be necessary. Although

[1] http://www.iop.org

[2] Tim Smith, publisher of New Journal of Physics, personal communication, October 2009.

[3] A selection of these appearances can be found in the author's web page: http://joanserra.weebly.com

[4] http://www.bmat.com/vericast

many of these methods already exist, there still is much room for improvement. We are particularly thinking about methods for melody and polyphonic pitch estimation, chord recognition and descriptor extraction in general (Casey et al., 2008b). Source separation would also be an important tool in version identification (Foucard et al., 2010).

An important issue arises regarding the combination of these multiple sources of information. In particular, one needs to decide where to combine the information and to devise the corresponding strategy for doing so (early and late-fusion schemes). This is a general problem shared across information science disciplines (e.g. Ross & Jain, 2003; Tahani & Keller, 1990; Temko et al., 2007). Many strategies exist, however there does not seem to be a clear winner. With regard to specific post-processing techniques, we advocate general clustering and classification techniques. Although these techniques are already incorporated in the state-of-the-art, we think they deserve further exploration. Perhaps a good starting point would be the incorporation of time series *inside* the clustering or classification algorithm (e.g. the alignment kernels of Joder et al., 2009).

The discussion above leads us to the use of different strategies to model the information extracted from audio. Related to this aspect, we believe that model-based approaches for version identification are a promising research direction to pursue. We find the reasons presented in the corresponding section to be a good indicator of what further research on this aspect can offer. It would be particularly interesting to see how hidden Markov models (HMM; Cappé et al., 2005; Rabiner, 1989) can be adapted to version identification. Such models have been very successful within the speech processing community (Rabiner & Juang, 1993) and thus are well-researched and established (see e.g. Pujol et al., 2005; Wilpon et al., 1990; Woodland & Povey, 2002). However, we conjecture that specific adaptations would be needed, in particular adaptations dealing with tempo and structure invariance (see e.g. Batlle et al. (2002) for some structure invariance adaptations of HMMs in the context of audio fingerprinting). In addition, one should use a continuous version of such models, since quantization of observations is not trivial in the case of version identification systems. Finally, an important point would be the incorporation of musical knowledge to the model. One way to incorporate such knowledge is by a case-based reasoning approach [Kolodner (1993); c.f. Arcos et al. (1997)].

In general, model-based approaches have an important industrial advantage over model-free ones: computational complexity. Indeed, more effort is needed in order to achieve scalable solutions that are able to effectively deal with music collections of millions of items. This is not a straightforward task, and current low-complexity methods fail to detect many songs when they are submitted to a pure/genuine version identification task (Sec. 2.3). Scalable methods need to achieve comparable (or better) accuracies than current non-scalable ones.

Another avenue for research is that of detecting musical quotations (Sec. 1.2.2). In classical music, there is a long tradition of composers citing phrases or motives from other composers (e.g. Alban Berg quoting Bach's chorale "Es ist genug" in his "Violin Concerto" or Richard Strauss quoting Beethoven's "Eroica symphony" in his "Metamorphosen for 23 solo strings"). In popular music there are also plenty of quotations (e.g. The Beatles' ending section of "All you need is love" quotes the French anthem "La Marseillaise" and Glen Miller's "In the mood" or Madonna's "Hung up" quoting ABBA's "Gimme, gimme, gimme"), and even modern electronic genres massively borrow loops and excerpts from any existing recording. As the quoted sections are usually of short duration, special adaptations of the current version identification algorithms would be required to detect them. In addition to facilitating legal procedures, linking diverse musical works this way opens new interesting ways for navigating across huge music collections. A related but different approach is to find, on a large-scale, music audio segments "that are similar not only in feature statistics, but in the relative positioning of those features" in time (Ellis et al., 2008).

The role of original songs within a group of versions is a research issue that deserves further exploration. In particular, it remains to be seen if some way to quantify the 'originality' of recordings exists or, at least, if some trends can be observed. Not only experiments with groups of versions should be performed, but also with pairwise comparisons. In the latter scenario perhaps one could maybe employ some measures of causality (e.g. Granger, 1969) or information transfer (e.g. Schreiber, 2000). However, we hypothesize that more informed and precise descriptions of the recordings should be used.

In order to identify song versions, the usual approach pays attention solely to the musical facets that are shared among them. This makes sense if we consider the task as a pure identification task. However, if we want to go beyond identification, we cannot suppose that musical changes do not affect the *similarity* between versions. With current systems, if two songs are versions and have the same timbre characteristics and a third song is also a version but does not exhibit the same timbre, they will score the same similarity. Future works approaching version similarity in a stricter sense (not just identification) might benefit from also considering also differences between music recordings so that, in the previous example, the third version is less similar than the first two (c.f. Tversky, 1977).

Determining version similarity in a stricter sense would have some practical consequences and would be a useful feature for music retrieval systems. Therefore, depending on the goals of the listeners, different degrees of similarity could be required. Here we have a new scenario where the ill-defined but typical music similarity problem needs to be addressed (Berenzweig et al., 2004).

Finally, on a more general side, automatic version identification calls for a human-motivated approach (Sec. 2.2.3). Current methods are constituted by a number of algorithms that do not resemble the ways humans process music information at all. It would be very interesting to devise a version identification system that performs the task as humans would. Indeed, a perceptually-inspired model for the processing of music signals plus a cognitively-motivated way to select and store relevant information items and a psychologically-sound comparison of such items would be a remarkable outcome.

Joan Serrà, Barcelona, February 9, 2011.

Bibliography

Aach, J. & Church, G. (2001). Aligning gene expression time series with time warping algorithms. *Bioinformatics*, *17*, 495–508.

Abdallah, S. & Plumbey, M. D. (2009). Information dynamics: patterns of expectation and surprise in the perception of music. *Connection Science*, *21*(2), 89–117.

Adams, N. H., Bartsch, N. A., Shifrin, J. B., & Wakefield, G. H. (2004). Time series alignment for music information retrieval. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 303–310.

Agger, G. (1999). Intertextuality revisited: dialogues and negotiations in media studies. *Canadian Journal of Aesthetics*, *4*. Available online: http://www.uqtr.ca/AE/vol_4/gunhild(frame).htm.

Agus, T. R., Thorpe, S. J., & Pressnitzer, D. (2010). Rapid formation of robust auditory memories: insights from noise. *Neuron*, *66*, 610–618.

Ahonen, T. E. (2010). Combining chroma features for cover version identification. In *Proc. of the Int. Soc. for Music Information Retrieval (ISMIR) Conf.*, pp. 165–170.

Ahonen, T. E. & Lemstrom, K. (2008). Identifying cover songs using normalized compression distance. In *Proc. of the Int. Workshop on Machine Learning and Music (MML)*, 5.

Allen, G. (2000). *Intertextuality: the new critical idiom.* New York, USA: Routledge, Taylor and Francis.

Andoni, A. & Indyk, P. (2008). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, *51*(1), 117–122.

Andrzejak, R. G. (2010). Nonlinear time series analysis in a nutshell. Tech. rep., Universitat Pompeu Fabra, Barcelona, Spain. Available online: http://www.cns.upf.edu/ralph/teachingM/Kansas3.pdf.

Antani, S., Kasturi, R., & Jain, R. (2002). A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, *35*(4), 945–965.

Arcos, J. L., López-de Mantaras, R., & Serra, X. (1997). SaxEx: a case-based reasoning system for generating expressive musical performances. In *Proc. of the Int. Computer Music Conf. (ICMC)*, pp. 329–336.

Aucouturier, J. J. & Pachet, F. (2004). Improving timbre similarity. How high is the sky? *Journal of Negative Results on Speech and Audio Sciences*, *1*(1).

Baeza-Yates, R., Calderón-Benavides, L., & González-Caro, C. (2006). The intention behind web queries. In F. Crestani, P. Ferragina, & M. Sanderson (Eds.) *Lecture Notes in Computer Science*, SPIRE 2006, pp. 98–109. Berlin, Germany: Springer.

Baeza-Yates, R., Castillo, C., & Efthimiadis, E. N. (2007). Characterization of national web domains. *ACM Trans. on Internet Technology*, *7*(2), 9.

Baeza-Yates, R. & Perleberg, C. S. (1996). Fast and practical approximate string matching. *Information Processing Letters*, *59*, 21–27.

Baeza-Yates, R. & Ribeiro-Neto, B. (1999). *Modern information retrieval.* New York, USA: ACM Press.

Bailes, F. (2010). Dynamic melody recognition: distinctiveness and the role of musical expertise. *Music and Cognition*, *38*(5), 641–650.

Barrat, A., Barthélemy, M., Pastor-Satorras, R., & Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the USA*, *101*, 3747.

Bartsch, N. A. & Wakefield, G. H. (2005). Audio thumbnailing of popular music using chroma-based representations. *IEEE Trans. on Multimedia*, *7*(1), 96–104.

Batlle, E., Masip, J., & Guaus, E. (2002). Automatic song identification in noisy broadcast audio. In *Proc. of the Signal and Image Processing Conf. (SIP)*, pp. 101–111.

Bello, J. P. (2007). Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 239–244.

Bello, J. P. & Pickens, J. (2005). A robust mid-level representation for harmonic content in music signals. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 304–311.

Berenzweig, A., Logan, B., Ellis, D. P. W., & Whitman, B. (2004). A large scale evaluation of acoustic and subjective music similarity measures. *Computer Music Journal*, *28*(2), 63–76.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics*, *10*, 10008.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., & Hwang, D.-U. (2006). Complex networks: structure and dynamics. *Physics Reports*, *424*(4), 175–308.

Bor, J. (2002). *The raga guide.* Monmouth, UK: Nimbus Communications.

Box, G. & Jenkins, G. (1976). *Time series analysis: forecasting and control.* Oakland, USA: Holden-Day, rev. edn.

Bregman, A. S. (1990). *Auditory scene analysis: the perceptual organization of sound.* Cambridge, USA: MIT Press.

Broomhead, D. S. & Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, *2*, 321–355.

Buldú, J. M., Cano, P., Koppenberger, M., Almendral, J., & Boccaletti, S. (2007). The complex network of musical tastes. *New Journal of Physics*, *9*, 172.

Caldwell, J. & Boyd, M. (2010). Dies irae. *Grove Music Online. Oxford Music Online.* Available online: http://www.oxfordmusiconline.com/subscriber/article/grove/music/40040.

Cano, P., Batlle, E., Kalker, T., & Haitsma, J. (2005). A review of audio fingerprinting. *Journal of VLSI Signal Processing Systems*, *41*(3), 271–284.

Cano, P., Celma, O., Koppenberger, M., & Buldú, J. M. (2006). Topology of music recommendation networks. *Chaos: an Interdisciplinary Journal of Nonlinear Science*, *16*(1), 013107.

Cappé, O., Moulines, E., & Rydén, T. (2005). *Inference in hidden Markov models.* New York, USA: Springer Science.

Casey, M., Rhodes, C., & Slaney, M. (2008a). Analysis of minimum distances in high-dimensional musical spaces. *IEEE Trans. on Audio, Speech and Language Processing*, *16*(5), 1015–1028.

Casey, M. & Slaney, M. (2006). The importance of sequences in musical similarity. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5, pp. V–V.

Casey, M., Veltkamp, R. C., Goto, M., Leman, M., Rhodes, C., & Slaney, M. (2008b). Content-based music information retrieval: current directions and future challenges. *Proceedings of the IEEE*, *96*(4), 668–696.

Casey, M. & Westner, W. (2000). Separation of mixed audio sources by independent subspace analysis. In *Proc. of the Int. Computer Music Conf. (ICMC)*, pp. 154–161.

Chai, W. (2005). *Automated analysis of musical structure*. Ph.D. thesis, Massachussets Institute of Technology, USA.

Chandran, V., Carswell, B., Boashash, B., & Elgar, S. L. (1997). Pattern recognition using invariants defined from higher order spectra: 2-D image inputs. *IEEE Trans. on Image Processing*, *6*(5), 703–712.

Chávez, E., Navarro, G., Baeza-Yates, R., & Marroquín, J. L. (2001). Searching metric spaces. *ACM Computing Surveys*, *33*(3), 273–321.

Chew, E. (2000). *Towards a mathematical model of tonality*. Ph.D. thesis, Massachussets Institute of Technology, USA. Available online: http://dspace.mit.edu/handle/1721.1/9139.

Chew, G., Mathiesen, T. J., Payne, T. B., & Fallows, D. (2010). Song. *Grove Music Online. Oxford Music Online*. Available online: http://www.oxfordmusiconline.com/subscriber/article/grove/music/50647.

Cho, T., Weiss, R. J., & Bello, J. P. (2010). Exploring common variations in state of the art chord recognition systems. In *Proc. of the Sound and Music Computing Conf. (SMC)*, 1.

Clauset, A., Newman, M. E. J., & Moore, C. (2003). Finding community structure in very large networks. *Physical Review E*, *70*(6), 066111.

Cohn, R. (1997). Neo-Riemannian operations, parsimonious trichords and their tonnetz representations. *Journal of Music Theory*, *1*(41), 1–66.

Comins, J. A. & Genter, T. Q. (2010). Working memory for patterned sequences of auditory objects in a songbird. *Cognition*, *117*(1), 38–53.

Conrad, R. (1965). Order error in immediate recall of sequences. *Journal of Verbal Learning and Verbal Behaviour*, *4*(3), 161–169.

Costa, L. d. F., Oliveira, O. N., Travieso, G., Rodrigues, F. A., Villas Boas, P. R., Antiqueira, L., Viana, M. P., & Correa da Rocha, L. E. (2008). Analyzing and modeling real-world phenomena with complex networks: a survey of applications. Working manuscript, arXiv:0711.3199v2. Available online: http://arxiv.org/abs/0711.3199.

Costa, L. d. F., Rodrigues, F. A., Travieso, G., & Villas Boas, P. R. (2007). Characterization of complex networks: a survey of measurements. *Advances in Physics*, *56*, 167–242.

Coyle, M. (2002). Hijacked hits and antic authenticity: cover songs, race and postwar marketing. In R. Beebe, D. Fulbrook, & B. Saunders (Eds.) *Rock over the edge: transformations in popular music culture*, pp. 133–157. Durham, UK: Duke University Press.

Cronin, C. (2002). The music plagiarism digital archive at Columbia law library: an effort to demystify music copyright infringement. In *Proc. of the IEEE Int. Conf. on Web Delivering of Music (WEDELMUSIC)*, pp. 1–8.

Dalla Bella, S., Peretz, I., & Aronoff, N. (2003). Time course of melody recognition: a gating paradigm study. *Perception and Psychophysics*, *7*(65), 1019–1028.

Daniélou, A. (1968). *Northern Indian Music.* London, UK: Barrie & Rockliff.

Dannenberg, R. B., Birmingham, W. P., Pardo, B., Hu, N., Meek, C., & Tzanetakis, G. (2007). A comparative evaluation of search techniques for query-by-humming using the MUSART testbed. *Journal of the American Society for Information Science and Technology*, *58*(5), 687–701.

Danon, L., Díaz-Aguilera, A., Duch, J., & Arenas, A. (2005). Comparing community structure identification. *Journal of Statistical Mechanics*, *9*, 09008.

De Cheveigne, A. (2005). Pitch perception models. In C. J. Plack, A. J. Oxenham, R. R. Fray, & A. N. Popper (Eds.) *Pitch: neural coding and perception*, chap. 6, pp. 169–233. New York, USA: Springer Science.

De Cheveigne, A. & Kawahara, H. (2001). Comparative evaluation of F0 estimation algorithms. In *Proc. of the Eurospeech Conf.*, pp. 2451–2454.

Deliege, I. (1996). Cue abstraction as a component of categorisation processes in music listening. *Psychology of Music*, *24*(2), 131–156.

Di Buccio, E., Montecchio, N., & Orio, N. (2010). A scalable cover identification engine. In *Proc. of the ACM Multimedia Conf. (ACM-MM)*, pp. 1143–1146.

Dixon, S. & Widmer, G. (2005). MATCH: a music alignment tool chest. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 492–497.

Dowling, W. J. (1978). Scale and contour: two components of a theory of memory for melodies. *Psychological Review*, *85*(4), 341–354.

Dowling, W. J. & Harwood, D. L. (1985). *Music cognition.* San Diego, USA: Academic Press.

Downie, J. S. (2008). The music information retrieval evaluation exchange (2005–2007): a window into music information retrieval research. *Acoustical Science and Technology*, *29*(4), 247–255.

Downie, J. S., Bay, M., Ehmann, A. F., & Jones, M. C. (2008). Audio cover song identification: MIREX 2006-2007 results and analyses. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 468–473.

Dubnov, S. (2006). Spectral anticipations. *Computer Music Journal*, *30*(2), 63–83.

Dubnov, S., Assayag, G., & Cont, A. (2007). Audio oracle: a new algorithm for fast learning of audio structures. *Int. Computer Music Conference (ICMC)*, pp. 224–228.

Eckmann, J. P., Kamphorst, S. O., & Ruelle, D. (1987). Recurrence plots of dynamical systems. *Europhysics Letters*, *5*, 973–977.

Eerola, T., Järvinen, T., Louhivuori, J., & Toiviainen, P. (2001). Statistical features and perceived similarity of folk melodies. *Music Perception*, *18*(3), 275–296.

Eerola, T., Toiviainen, P., & Krumhansl, C. L. (2002). Real-time prediction of melodies: continuous predictability judgements and dynamic models. *Int. Conf. on Music Perception and Cognition (ICMPC)*, pp. 473–476.

Egorov, A. & Linetsky, G. (2008). Cover song identification with IF-F0 pitch class profiles. *Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.*

Ellis, D. P. W. & Cotton, C. (2007). The 2007 LabROSA cover song detection system. *Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.*

Ellis, D. P. W., Cotton, C., & Mandel, M. (2008). Cross-correlation of beat-synchronous representations for music similarity. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 57–60.

Ellis, D. P. W. & Poliner, G. E. (2007). Identifying cover songs with chroma features and dynamic programming beat tracking. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, pp. 1429–1432.

Facchini, A., Kantz, H., & Tiezzi, E. (2005). Recurrence plot analysis of nonstationary data: the understanding of curved patterns. *Physical Review E*, *72*, 021915.

Farmer, J. D. & Sidorowich, J. J. (1987). Predicting chaotic time series. *Physical Review Letters*, *59*(8), 845–848.

Foote, J. (1999). Visualizing music and audio using self-similarity. In *Proc. of the ACM Int. Conf. on Multimedia*, pp. 77–80.

Foote, J. (2000a). ARTHUR: Retrieving orchestral music by long-term structure. In *Proc. of the Int. Symp. on Music Information Retrieval (ISMIR)*, 130.

Foote, J. (2000b). Automatic audio segmentation using a measure of audio novelty. In *Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME)*, vol. 1, pp. 452–455.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, *486*(3), 75–174.

Fortunato, S. & Castellano, C. (2009). Community structure in graphs. In R. A. Meyers (Ed.) *Encyclopedia of complexity and system science*, pp. 1141–1163. Berlin, Germany: Springer.

Foucard, R., Durrieu, J.-L., Lagrange, M., & Richard, G. (2010). Multimodal similarity between musical streams for cover version detection. In *Proc. of the IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, pp. 5514–5517.

Frankel, A. S. (1998). Sound production. In W. F. Perrin, B. Wursig, & J. G. M. Thewissen (Eds.) *Encyclopedia of Marine Mammals*, pp. 1126–1137. San Diego, USA: Academic Press.

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

Gainza, M. (2009). Automatic musical meter detection. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 329–332.

Gerhard, D. (1999). Audio visualization in phase space. In *Proc. of Bridges 99: Mathematical Connections in Art, Music and Science*, pp. 137–144.

Gómez, E. (2006). *Tonal description of music audio signals.* Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, Spain. Available online: http://mtg.upf.edu/node/472.

Gómez, E. & Herrera, P. (2004). Estimating the tonality of polyphonic audio files: cognitive versus machine learning modelling strategies. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 92–95.

Gómez, E. & Herrera, P. (2006). The song remains the same: identifying versions of the same song using tonal descriptors. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 180–185.

Gómez, E., Herrera, P., Cano, P., Janer, J., Serrà, J., Bonada, J., El-Hajj, S., Aussenac, T., & Holmberg, G. (2008). Music similarity systems and methods using descriptors. *Patent WO 2009/001202.*

Gómez, E., Ong, B. S., & Herrera, P. (2006a). Automatic tonal analysis from music summaries for version identification. In *Proc. of the Conv. of the Audio Engineering Society (AES)*, 6902.

Gómez, E., Streich, S., Ong, B. S., Paiva, R. P., Tappert, S., Batke, J. M., Poliner, G. E., Ellis, D. P. W., & Bello, J. P. (2006b). A quantitative comparison of different approaches for melody extraction from polyphonic audio recordings. Tech. rep., Universitat Pompeu Fabra, Barcelona, Spain. Available online: http://mtg.upf.edu/node/460.

Goto, M. (2006). A chorus-section detection method for musical audio signals and its application to a music listening station. *IEEE Trans. on Audio, Speech and Language Processing*, *14*(5), 1783–1794.

Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., & Cano, P. (2006). An experimental comparison of audio tempo induction algorithms. *IEEE Trans. on Speech and Audio Processing*, *14*(5), 1832–1844.

Grachten, M., Arcos, J. L., & López-de Mantaras, R. (2004). Melodic similarity: looking for a good abstraction level. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 210–215.

Grachten, M., Arcos, J. L., & López-de Mantaras, R. (2005). Melody retrieval using the implication/realization model. *Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.*

Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, *37*(3), 424–438.

Groth, A. (2005). Visualization of coupling in time series by order recurrence plots. *Physical Review E*, *72*(4), 046220.

Gusfield, D. (1997). *Algorithms on strings, trees and sequences: computer sciences and computational biology.* Cambridge, UK: Cambridge University Press.

Hal Leonard Corp. (2004). *The real book.* Milwaukee, USA: Hal Leonard Corp., 6th edn.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The Weka data mining software: an update. *ACM SIGKDD Explorations*, *1*(1), 10–18.

Harte, C., Sandler, M. B., & Gasser, M. (2006). Detecting harmonic change in musical audio. In *Proc. of the ACM Workshop on Audio and Music Computing Multimedia*, pp. 21–26.

Harte, C. A. & Sandler, M. B. (2005). Automatic chord identification using a quantized chromagram. In *Proc. of the Conv. of the Audio Engineering Society (AES)*, pp. 28–31.

Harwood, D. L. (1976). Universals in music: a perspective from cognitive psychology. *Ethomusicology*, *20*(3), 521–533.

Hazan, A., Marxer, R., Brossier, P., Purwins, H., Herrera, P., & Serra, X. (2009). What/when causal expectation modelling applied to audio signals. *Connection Science*, *21*(2), 119–143.

Hegger, R., Kantz, H., & Matassini, L. (2000). Denoising human speech signals using chaoslike features. *Physical Review Letters*, *84*(14), 3197–3200.

Heikkila, J. (2004). A new class of shift-invariant operators. *IEEE Signal Processing Magazine*, *11*(6), 545–548.

Henson, R. (2001). Serial order in short term memory. *The Psychologist*, *14*(2), 70–73.

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

Hughes, J. M., Graham, D. J., & Rockmore, D. N. (2010). Quantification of artistic style through sparse coding analysis in the drawings of Pieter Bruegel the Elder. *Proceedings of the National Academy of Sciences of the USA*, *107*(4), 1279–1283.

Hughes, J. P. (2006). Embedding nonlinear dynamical systems: a guide to Takens' theorem. Tech. rep., The University of Manchester, Manchester, UK. Available online: http://eprints.ma.man.ac.uk/175.

Huron, D. (2006). *Sweet anticipation: music and the psychology of expectation.* Cambridge, USA: MIT Press.

Hyer, B. (2010). Tonality. *Grove Music Online. Oxford Music Online.* Available online: http://www.oxfordmusiconline.com/subscriber/article/grove/music/28102.

Izmirli, Ö. (2005). Tonal similarity from audio using a template based attractor model. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 540–545.

Jain, A. K. & Dubes, R. C. (1988). *Algorithms for clustering data.* Advanced reference series. Upper Saddle River, USA: Prentice Hall.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, *31*(3), 264–323.

Jensen, J. H., Christensen, M. G., Ellis, D. P. W., & Jensen, S. H. (2008a). A tempo-insensitive distance measure for cover song identification based on chroma features. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2209–2212.

Jensen, J. H., Christensen, M. G., Ellis, D. P. W., & Jensen, S. H. (2009). Quantitative analysis of a common audio similarity measure. *IEEE Trans. on Audio, Speech and Language Processing*, *17*(4), 693–703.

Jensen, J. H., Christensen, M. G., & Jensen, S. H. (2008b). A chroma-based tempo-insensitive distance measure for cover song identification using the 2D autocorrelation. *Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.*

Jeong, H., Mason, S. P., Barábasi, A. L., & Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature*, *411*, 41–42.

Joder, C., Essid, S., & Richard, G. (2009). Temporal integration for audio classification with application to musical instrument classification. *IEEE Trans. on Audio, Speech and Language Processing*, *17*(1), 174–186.

Juola, P. (2008). Authorship attribution. *Foundations and Trends in Information Retrieval*, *1*(3), 233–334.

Juslin, P. N., Friberg, A., & Bresin, R. (2002). Toward a computational model of expression in performance: the GERM model. *Musicae Scientiae, Special Issue 2001-2002*, pp. 63–122.

Kantz, H. & Schreiber, T. (2004). *Nonlinear time series analysis.* Cambridge, UK: Cambridge University Press, 2nd edn.

Kernfeld, B. (2006). *The story of fake books: bootlegging songs to musicians.* Lanham, USA: The Scarecrow Press.

Kim, S. & Narayanan, S. (2008). Dynamic chroma feature vectors with applications to cover song identification. In *Proc. of the IEEE Workshop on Multimedia Signal Processing (MMSP)*, pp. 984–987.

Kim, S., Unal, E., & Narayanan, S. (2008). Fingerprint extraction for classical music cover song identification. In *Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME)*, pp. 1261–1264.

Kim, Y. E. & Perelstein, D. (2007). MIREX 2007: audio cover song detection using chroma features and hidden Markov model. *Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.*

Klette, R. & Zamperoni, P. (1996). *Handbook of image processing operators.* New York, USA: John Wiley and Sons.

Kolodner, J. L. (1993). *Case-based reasoning.* Burlington, USA: Morgan Kauffmann.

Kotska, S. (2005). *Materials and techniques of the 20th century music.* Upper Saddle River, USA: Prentice Hall, 3rd edn.

Krumhansl, C. L. (1990). *Cognitive foundations of musical pitch.* Oxford, UK: Oxford University Press.

Kurth, F. & Müller, M. (2008). Efficient index-based audio matching. *IEEE Trans. on Audio, Speech and Language Processing*, *16*(2), 382–395.

Kuusi, T. (2009). Tune recognition from melody, rhythm and harmony. In *Proc. of the Conf. of the European Soc. for the Cognitive Sciences of Music (ESCOM)*, pp. 610–614.

Kvam, P. H. & Vidakovic, B. (2007). *Nonparametric statistics with applications to science and engineering.* Hoboken, USA: John Wiley and Sons.

Lagrange, M. & Serrà, J. (2010). Unsupervised accuracy improvement for cover song detection using spectral connectivity network. In *Proc. of the Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 595–600.

Latora, V. & Marchiori, M. (2001). Efficient behavior of small-world networks. *Physical Review Letters*, *87*, 198701.

Lee, K. (2006). Identifying cover songs from audio using harmonic representation. *Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.*

Lee, K. (2008). *A system for acoustic chord transcription and key extraction from audio using hidden Markov models trained on synthesized audio.* Ph.D. thesis, Stanford University, USA.

Leman, M. (1995). *Music and schema theory: cognitive foundations of systematic musicology.* Berlin, Germany: Springer.

Lemstrom, K. (2000). *String matching techinques for music retrieval.* Ph.D. thesis, University of Helsinki, Finland.

Lerdahl, F. (2001). *Tonal pitch space.* Oxford, UK: Oxford University Press.

Lerdahl, F. & Jackendorff, R. (1983). *A generative theory of tonal music.* Cambridge, USA: MIT Press.

Lesaffre, M. (2005). *Music Information Retrieval: conceptual framework, annotation and user behavior.* Ph.D. thesis, Ghent University, Belgium.

Levitin, D. J. (2007). *This is your brain on music: the science of a human obsession.* London, UK: Atlantic books.

Liem, C. C. S. & Hanjalic, A. (2009). Cover song retrieval: a comparative study of system component choices. In *Proc. of the Int. Soc. for Music Information Retrieval (ISMIR) Conf.*, pp. 573–578.

Lütkepohl, H. (1993). *Introduction to multiple time series analysis.* Berlin, Germany: Springer, 2nd edn.

Lynch, M. P., Eilers, R. E., Oller, D. K., & Urbano, R. C. (1990). Innateness, experience and music perception. *Psychological Science*, *1*(4), 272–276.

Mäkinen, V., Navarro, G., & Ukkonen, E. (2005). Transposition invariant string matching. *Journal of Algorithms*, *56*(2), 124–153.

Manning, C. D., Prabhakar, R., & Schutze, H. (2008). *An introduction to information retrieval.* Cambridge, UK: Cambridge University Press.

Mardirossian, A. & Chew, E. (2006). Music summarization via key distributions: analyses of similarity assessment across variations. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, 282.

Marler, P. & Slabbekoorn, H. W. (2004). *Nature's music: the science of birdsong.* San Diego, USA: Academic Press.

Marolt, M. (2006). A mid-level melody-based representation for calculating audio similarity. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 280–285.

Marolt, M. (2008). A mid-level representation for melody-based retrieval in audio collections. *IEEE Trans. on Multimedia*, *10*(8), 1617–1625.

Marwan, N. & Kurths, J. (2002). Nonlinear analysis of bivariate data with cross recurrence plots. *Physics Letters A*, *302*(5), 299–307.

Marwan, N., Romano, M. C., Thiel, M., & Kurths, J. (2007). Recurrence plots for the analysis of complex systems. *Physics Reports*, *438*(5), 237–329.

Marwan, N., Thiel, M., & Nowaczyk, N. R. (2002a). Cross recurrence plot based synchronization of time series. *Nonlinear Processes in Geophysics*, *9*, 325–331.

Marwan, N., Wessel, N., Meyerfeldt, U., Schirdewan, A., & Kurths, J. (2002b). Recurrence-plot-based measures of complexity and its application to heart rate variability data. *Physical Review E*, *66*(2), 026702.

Matassini, L., Kantz, H., Holyst, J., & Hegger, R. (2002). Optimizing of recurrence plots for noise reduction. *Physical Review E*, *65*, 021102.

Meng, A., Ahrendt, P., Larsen, J., & Hansen, L. K. (2007). Temporal feature integration for music genre classification. *IEEE Trans. on Audio, Speech and Language Processing*, *15*(5), 1654–1664.

Mierswa, I. & Morik, K. (2005). Automatic feature extraction for classifying audio data. *Machine Learning Journal*, *58*, 127–149.

Miotto, R. & Orio, N. (2008). A music identification system based on chroma indexing and statistical modeling. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 301–306.

Mithen, S. (2007). *The singing Neanderthals: the origins of music, language, mind and body.* Cambridge, USA: Harvard University Press.

Molina-Solana, M., Arcos, J. L., & Gómez, E. (2010). Identifying violin performers by their expressive trends. *Intelligent Data Analysis*, *14*, 555–571.

Mörchen, F., Mierswa, I., & Ultsch, A. (2006a). Understandable models of music collections based on exhaustive feature generation with temporal statistics. In *Proc. of the ACM Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 882–891.

Mörchen, F., Ultsch, A., Thies, M., & Löhken, I. (2006b). Modelling timbre distance with temporal statistics from polyphonic music. *IEEE Trans. on Speech and Audio Processing*, *14*(1), 81–90.

Mosser, K. (2010). Cover songs: ambiguity, multivalence, polysemy. *Popular Musicology Online.* Available online: http://www.popular-musicology-online.com/issues/02/mosser.html.

Müllensiefen, D. & Pendzich, M. (2009). Court decisions on music plagiarism and the predictive value of similarity algorithms. *Musicae Scientiae, Discussion Forum 4B*, pp. 207–238.

Müller, M. (2007). *Information Retrieval for Music and Motion.* Berlin, Germany: Springer.

Müller, M. & Appelt, D. (2008). Path-constrained partial music synchronization. In *Proc. of the IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, pp. 65–68.

Müller, M. & Ewert, S. (2008). Joint structure analysis with applications to music annotation and synchronization. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 389–394.

Müller, M. & Ewert, S. (2010). Towards timbre-invariant audio features for harmony-based music. *IEEE Trans. on Audio, Speech and Language Processing*, *18*(3), 649–662.

Müller, M. & Kurth, F. (2006a). Enhancing similarity matrices for music audio analysis. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. V–V.

Müller, M. & Kurth, F. (2006b). Towards structural analysis of audio recordings in the presence of musical variations. *EURASIP Journal on Advances in Signal Processing*, *2007*(89686), 1–18.

Müller, M., Kurth, F., & Clausen, M. (2005). Audio matching via chroma-based statistical features. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 288–295.

Myers, C. (1980). *A comparative study of several dynamic time warping algorithms for speech recognition*. Master's thesis, Massachussets Institute of Technology, USA.

Myers, C., Rabiner, L. R., & Rosenberg, A. E. (1980). Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Trans. on Audio, Speech and Language Processing*, *28*(6), 623– 635.

Nadeu, C., Macho, D., & Hernando, J. (2001). Time and frequency filtering of filter-bank energies for robust HMM speech recognition. *Speech Comunication*, *34*, 93–114.

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

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, *45*, 167–256.

Ong, B. S. (2007). *Structural analysis and segmentation of music signals*. Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, Spain. Available online: http://mtg.upf.edu/node/508.

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

Oppenheim, A. V., Schafer, R. W., & Buck, J. B. (1999). *Discrete-Time Signal Processing*. Upper Saddle River, USA: Prentice Hall, 2 edn.

Orio, N. (2006). Music retrieval: a tutorial and review. *Foundations and Trends in Information Retrieval*, *1*(1), 1–90.

Oswald, J. (1985). Plunderphonics, or audio piracy as a compositional prerogative. *Wired Society Electro-Acoustic Conference*. Available online: http://www.plunderphonics.com/xhtml/xplunder.html.

Pachet, F. (2002). The continuator: musical interaction with style. *Journal of New Music Research*, *31*(1), 1–9.

Pachet, F. (2005). Knowledge management and musical metadata. In D. Schwartz (Ed.) *Encyclopedia of Knowledge Management*. Harpenden, UK: Idea Group.

Paiement, J. F., Grandvalet, Y., & Benigo, S. (2009). Predictive models for music. *Connection Science*, *21*(2), 253–272.

Pampalk, E. (2006). *Computational models of music similarity and their application to music information retrieval*. Ph.D. thesis, Vienna University of Technology, Vienna, Austria. Available online: http://www.ub.tuwien.ac.at/diss/AC05031828.pdf.

Papadopoulos, H. & Peeters, G. (2007). Large-scale study of chord estimation algorithms based on chroma representation and HMM. In *Proc. of the Int. Conf. on Content-Based Multimedia Information (CBMI)*, pp. 53–60.

Parka, H. S. & Jun, C. S. (2009). A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, *36*(2), 3336–3341.

Patel, A. (2008). *Music, language and the brain*. Oxford, UK: Oxford University Press.

Peeters, G. (2007). Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 35–40.

Peeters, G., La Burthe, A., & Rodet, X. (2002). Toward automatic music audio summary generation from signal analysis. In *Proc. of the Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 94–100.

Pickens, J. (2004). *Harmonic modeling for polyphonic music retrieval*. Ph.D. thesis, University of Massachussetts Amherst, USA.

Pickens, J., Bello, J. P., Monti, G., Sandler, M. B., Crawford, T., Dovey, M., & Byrd, D. (2003). Polyphonic score retrieval using polyphonic audio queries: a harmonic modeling approach. *Journal of New Music Research*, *32*(2), 223–236.

Plasketes, G. (2010). Re-flections on the cover age: a collage of continuous coverage in popular music. *Popular Music and Society*, *28*(2), 137–161.

Pohle, T., Schnitzer, D., Schedl, M., Knees, P., & Widmer, G. (2009). On rhythm and general music similarity. In *Proc. of the Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 525–530.

Poliner, G. E., Ellis, D. P. W., Ehmann, A., Gómez, E., Streich, S., & Ong, B. S. (2007). Melody transcription from music audio: approaches and evaluation. *IEEE Trans. on Audio, Speech and Language Processing*, *15*(4), 1247–1256.

Polotti, P. & Rocchesso, D. (2008). *Sound to sense - sense to sound: a state of the art in sound and music computing.* Berlin, Germany: Logos.

Porter, S. E. (1997). The use of the old testament in the new testament: a brief comment on method and terminology. In C. A. Evans & J. A. Sanders (Eds.) *Early christian interpretations of the scriptures of Israel: investigations and proposals*, pp. 79–96. Sheffield, UK: Sheffield Academic Press.

Posner, R. A. (2007). *The little book of plagiarism.* New York, USA: Pantheon Books.

Press, W. H., Flannery, B. P., Tenkolsky, S. A., & Vetterling, W. T. (1992). *Numerical recipes.* Cambridge, UK: Cambridge University Press, 2nd edn.

Pujol, P., Pol, S., Nadeu, C., Hagen, A., & Bourlard, A. (2005). Comparison and combination of features in a hybrid HMM/MLP and a HMM/GMM speech recognition system. *IEEE Trans. on Audio, Speech and Language Processing*, *13*(1), 14–22.

Purwins, H. (2005). *Profiles of pitch classes - circularity of relative pitch and key: experiments, models, momputational music analysis and perspectives.* Ph.D. thesis, Technischen Universität, Berlin, Germany.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Rabiner, L. R. & Juang, B. H. (1993). *Fundamentals of speech recognition.* Upper Saddle River, USA: Prentice Hall.

Ratzan, L. (2004). *Understanding information systems: what they do and why we need them.* Chicago, USA: American Library Association.

Ravuri, S. & Ellis, D. P. W. (2010). Cover song detection: from high scores to general classification. In *Proc. of the IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, pp. 55–58.

Reiss, J. D. & Sandler, M. B. (2003). Nonlinear time series analysis of musical signals. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, pp. 1–5.

Resnick, P. & Varian, H. L. (1997). Recommender systems. *Communications of the ACM*, *40*(3), 56–58.

Riley, M., Heinen, E., & Ghosh, J. (2008). A text retrieval approach to content-based audio retrieval. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 295–300.

Rizo, D., Lemstrom, K., & Iñesta, J. M. (2009). Ensemble of state-of-the-art methods for polyphonic music comparison. In *Proc. of the European Conference on Digital Libraries (ECDL), Workshop on Exploring Musical Information Spaces*, pp. 46–51.

Röbel, A. & Rodet, X. (2005). Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, pp. 30–35.

Robert, C. P. & Casella, G. (2004). *Monte Carlo statistical methods*. Berlin, Germany: Springer, 2nd edn.

Robine, M., Hanna, P., Ferraro, P., & Allali, J. (2007). Adaptation of string matching algorithms for identification of near-duplicate music documents. In *Proc. of the ACM SIGIR Workshop on Plagiarism Analysis, Authorship Identification and Near-Duplicate Detection (PAN)*, pp. 37–43.

Rogers, T. T. & McClelland, J. L. (2004). *Semantic cognition: a parallel distributed processing approach*. Cambridge, USA: MIT Press.

Rosch, E. & Mervis, C. (1975). Family resemblances: studies in the internal structure of categories. *Cognitive Psychology*, *7*, 573–605.

Ross, A. & Jain, A. (2003). Information fusion in biometrics. *Pattern Recognition Letters*, *24*(13), 2115–2125.

Roth, P. M. & Winter, M. (2008). Survey of appearance-based methods for object recognition. Tech. rep. Available online: http://web.mit.edu/~wingated/www/introductions/appearance_based_methods.pdf.

Russell, S. J. & Norvig, P. (2003). *Artificial intelligence: a modern approach*. Upper Saddle River, USA: Prentice Hall.

Sahoo, N., Callan, J., Krishnan, R., Duncan, G., & Padman, R. (2006). Incremental hierarchical clustering of text documents. In *Proc. of the ACM Int. Conf. on Information and Knowledge Management*, pp. 357–366.

Sailer, C. & Dressler, K. (2006). Finding cover songs by melodic similarity. *Music Information Retrieval Evaluation eXchange (MIREX) extended abstract*. Available online: http://www.music-ir.org/mirex/abstracts/2006/CS_sailer.pdf.

Sankoff, D. & Kruskal, J. (1983). *Time warps, string edits and macromolecules*. Reading, USA: Addison-Wesley.

Sauer, T. D. (2006). Attractor reconstruction. *Scholarpedia*, *1*(10), 1727.

Saygin, A. P., Cicekli, I., & Akman, V. (2000). Turing test: 50 years later. *Minds and Machines*, *10*(4), 463–518.

Scaringella, N., Zoia, G., & Mlynek, D. (2006). Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, *23*(2), 133–141.

Schaefer, R. S., Farquhar, J., Blokland, Y., Sadakata, M., & Desain, P. (2010). Name that tune: decoding music from the listening brain. *NeuroImage*. In press.

Schellenberg, E. G., Iverson, P., & McKinnon, M. C. (1999). Name that tune: identifying familiar recordings from brief excerpts. *Psychonomic Bulletin and Review*, *6*(4), 641–646.

Schreiber, T. (2000). Measuring information transfer. *Physical Review E*, *85*, 461–464.

Schulkind, M. D., Posner, R. J., & Rubin, D. C. (2003). Musical features that facilitate melody identification: how do you know it's your song when they finally play it? *Music Perception*, *21*(2), 217–249.

Selfridge-Field, E. (1998). Conceptual and representational issues in melodic comparison. In W. B. Hewlett & E. Selfridge-Field (Eds.) *Melodic similarity: concepts, procedures and applications*, *Computing in Musicology*, vol. 11, pp. 3–64. Cambridge, USA: MIT Press.

Serrà, J. (2007a). *Music similarity based on sequences of descriptors: tonal features applied to audio cover song identification*. Master's thesis, Universitat Pompeu Fabra, Barcelona, Spain. Available online: http://mtg.upf.edu/node/536.

Serrà, J. (2007b). A qualitative assessment of measures for the evaluation of a cover song identification system. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 319–322.

Serrà, J., Gómez, E., & Herrera, P. (2008a). Transposing chroma representations to a common key. In *Proc. of the IEEE CS Conf. on The Use of Symbols to Represent Music and Multimedia Objects*, pp. 45–48.

Serrà, J., Gómez, E., & Herrera, P. (2010a). Audio cover song identification and similarity: background, approaches, evaluation and beyond. In Z. W. Ras & A. A. Wieczorkowska (Eds.) *Adv. in Music Information Retrieval*, *Studies in Computational Intelligence*, vol. 16, chap. 14, pp. 307–332. Berlin, Germany: Springer.

Serrà, J., Gómez, E., Herrera, P., & Serra, X. (2008b). Chroma binary similarity and local alignment applied to cover song identification. *IEEE Trans. on Audio, Speech and Language Processing*, *16*(6), 1138–1152.

Serrà, J., Gómez, E., Herrera, P., & Serra, X. (2008c). Statistical analysis of chroma features in Western music predicts human judgments of tonality. *Journal of New Music Research*, *37*(4), 299–309.

Serrà, J., Kantz, H., & Andrzejak, R. G. (2010b). Model-based cover song detection via threshold autoregressive forecasts. In *Proc. of the ACM Multimedia, Workshop on Music and Machine Learning (MML)*, pp. 13–16.

Serrà, J., Kantz, H., Serra, X., & Andrzejak, R. G. (2010c). Predictability of music descriptor time series and its application to cover song detection. *IEEE Trans. on Audio, Speech and Language Processing.* Submitted.

Serrà, J., Serra, X., & Andrzejak, R. G. (2009a). Cross recurrence quantification for cover song identification. *New Journal of Physics*, *11*, 093017.

Serrà, J., Zanin, M., Herrera, P., & Serra, X. (2010d). Characterization and exploitation of community structure in cover song networks. *Pattern Recognition Letters.* Submitted.

Serrà, J., Zanin, M., Laurier, C., & Sordo, M. (2009b). Unsupervised detection of cover song sets: accuracy increase and original detection. In *Proc. of the Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 225–230.

Serra, X. (1997). Musical sound modeling with sinusoids plus noise. In C. Roads, S. T. Pope, A. Picialli, & G. De Poli (Eds.) *Musical Signal Processing*, Studies on New Music Research, pp. 91–122. London, UK: Swets and Zeitlinger.

Sheh, A. & Ellis, D. P. W. (2003). Chord segmentation and recognition using EM-trained Hidden Markov Models. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 183–189.

Sisman, E. (2010). Variations. *Grove Music Online. Oxford Music Online.* Available online: http://www.oxfordmusiconline.com/subscriber/article/grove/music/29050pg10.

Smith III, J. O. (2010a). *Mathematics of the discrete Fourier transform with audio applications.* Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, USA, 2nd edn. Online resource: https://ccrma.stanford.edu/~jos/mdft.

Smith III, J. O. (2010b). *Spectral audio signal processing.* Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, USA. Online resource: https://ccrma.stanford.edu/~jos/sasp.

Solis, G. (2010). I did it my way: rock and the logic of covers. *Popular Music and Society*, *33*(3), 297–318.

Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, *60*(3), 538–556.

Strogatz, S. H. (2001). Exploring complex networks. *Nature*, *410*, 268–276.

Tahani, H. & Keller, J. M. (1990). Information fusion in computer vision using the fuzzy integral. *IEEE Trans. on Systems, Man and Cybernetics*, *20*(3), 733–741.

Takens, F. (1981). Detecting strange attractors in turbulence. *Lecture Notes in Mathematics*, *898*, 366–381.

Taylor, R. P., Guzman, R., Martin, T. P., Hall, G. D. R., Micolich, A. P., Jonas, D., Scannell, B. C., Fairbanks, M. S., & Marlow, C. A. (2007). Authenticating Pollock paintings using fractal geometry. *Pattern Recognition Letters*, *28*(6), 695–702.

Teitelbaum, T., Balenzuela, P., Cano, P., & Buldú, J. M. (2008). Community structures and role detection in music networks. *Chaos: an Interdisciplinary Journal of Nonlinear Science*, *18*(4), 043105.

Temko, A., Macho, D., & Nadeu, C. (2007). Fuzzy integral based information fusion for classification of highly confusable non-speech sounds. *Pattern Recognition*, *41*(5), 1814–1823.

Theodoridis, S. & Koutroumbas, K. (2006). *Pattern recognition.* San Diego, USA: Academic Press, 3rd edn.

Todd, N. P. (1992). The dynamics of dynamics: a model of musical expression. *Journal of the Acoustical Society of America*, *91*(6), 3540–3550.

Tong, H. & Lim, K. S. (1980). Threshold autoregression, limit cycles and cyclical data. *Journal of the Royal Statistical Society*, *42*(3), 245–292.

Tsai, W.-H., Yu, H.-M., & Wang, H.-M. (2005). A query-by-example technique for retrieving cover versions of popular songs with similar melodies. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 183–190.

Tsai, W.-H., Yu, H.-M., & Wang, H.-M. (2008). 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.

Turing, A. (1950). Computing machinery and intelligence. *Mind*, *59*(254), 433–460.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*, 327–352.

Typke, R. (2007). *Music retrieval based on melodic similarity.* Ph.D. thesis, Utrecht University, Netherlands.

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

Ukkonen, E., Lemstrom, K., & Mäkinen, V. (2003). Sweepline the music! *Computer Science in Perspective*, pp. 330–342.

Unal, E. & Chew, E. (2007). Statistical modeling and retrieval of polyphonic music. In *Proc. of the IEEE Workshop on Multimedia Signal Processing (MMSP)*, pp. 405–409.

Urbano, J., Lloréns, J., Morato, J., & Sánchez-Cuadrado, S. (2010). Using the shape of music to compute the similarity between symbolic musical pieces. In *Proc. of the Int. Symp. on Computer Music Modeling and Retrieval (CMMR)*, pp. 385–396.

Van Kampen, N. G. (2007). *Stochastic processes in physics and chemistry.* Amsterdam, The Netherlands: Elsevier, 3rd edn.

Van Kranenburg, P. (2010). *A computational approach to content-based retrieval of folk-song melodies.* Ph.D. thesis, Utrecht University, Utrecht, The Netherlands. Available online: http://www.cs.uu.nl/groups/MG/multimedia/publications/art/petervankranenburg-dissertation.pdf.

Vlachos, I. & Kugiumtzis, D. (2008). State space reconstruction for multivariate time series prediction. *Nonlinear Phenomena in Complex Systems*, *11*(2), 241–249.

Von Luxburg, A. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416.

Voorhees, E. M. & Harman, D. K. (2005). *TREC: experiment and evaluation in information retrieval.* Cambridge, USA: MIT Press.

Webber Jr., C. L. & Zbilut, J. P. (1994). Dynamical assessment of physiological systems and states using recurrence plot strategies. *Journal of Applied Physiology*, *76*(2), 965–973.

Weigend, A. S. & Gershenfeld, N. A. (1993). *Time series prediction: forecasting the future and understanding the past.* Boulder, USA: Westwiew Press.

Weinstein, D. (1998). The history of rock's pasts through rock covers. In T. Swiss, J. Sloop, & A. Herman (Eds.) *Mapping the beat: popular music and contemporary theory*, pp. 137–151. Oxford, UK: Blackwell Publishing Ltd.

West, K. & Lamere, P. (2007). A model-based approach to constructing music similarity functions. *EURASIP Journal on Applied Signal Processing*, *2007*(1), 24602.

Wilpon, J. G., Rabiner, L. R., Lee, C. H., & Goldman, E. (1990). Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Trans. on Acoustics, Speech and Signal Processing*, *38*(11), 1870–1878.

Witmer, R. & Marks, A. (2010). Cover version. *Grove Music Online. Oxford Music Online.* Available online: http://www.oxfordmusiconline.com/subscriber/article/grove/music/49254.

Witten, I. H. & Frank, E. (2005). *Data mining: practical machine learning tools and techniques.* Amsterdam, The Netherlands: Elsevier, 2nd edn.

Wittgenstein, L. (1953). *Philosophical investigations.* Oxford, UK: Blackwell Publishing Ltd.

Woodland, P. C. & Povey, D. (2002). Large scale discriminative training of hidden Markov models for speech recognition. *Computer, Speech and Language*, *16*(1), 25–47.

Xu, R. & Wunsch II, D. C. (2009). *Clustering.* Piscataway, USA: IEEE Press.

Yang, C. (2001). Music database retrieval based on spectral similarity. In *Proc. of the Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 37–38.

Yano, C. R. (2005). Covering disclosures: practices of intimacy, hierarchy and authenticity in a Japanese popular music genre. *Popular Music and Society*, *28*(2), 193–205.

Yu, Y., Downie, J. S., Chen, L., Oria, V., & Joe, K. (2008). Searching musical audio datasets by a batch of multi-variant tracks. In *Proc. of the ACM Multimedia Conf.*, pp. 121–127.

Zbikowski, L. M. (2002). *Conceptualizing music: cognitive structure, theory and analysis.* Oxford, UK: Oxford University Press.

Zbilut, J. P., Giuliani, A., & Webber Jr., C. L. (1998). Detecting deterministic signals in exceptionally noisy environments using cross-recurrence quantification. *Physics Letters A*, *246*(1), 122–128.

Zbilut, J. P. & Webber Jr., C. L. (1992). Embeddings and delays as derived from quantification of recurrence plots. *Physics Letters A*, *171*(3), 199–203.

Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: a literature survey. *ACM Computing Surveys*, *35*(4), 399–458.

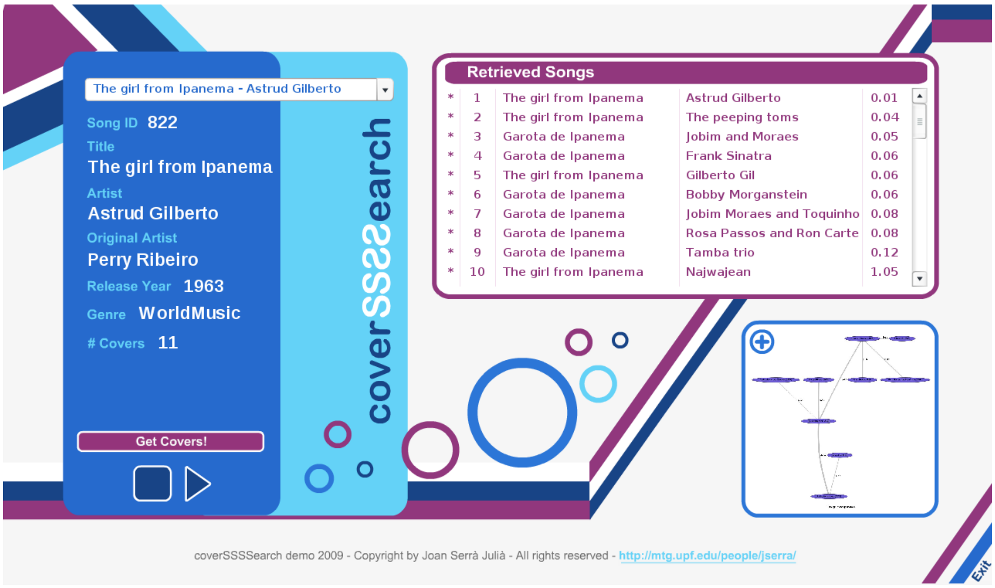
Zhao, Y. & Karypis, G. (2002). Evaluation of hierarchical clustering algorithms for document datasets. In *Proc. of the Conf. on Knowledge Discovery in Data (KDD)*, pp. 515–524.

Appendix A: the system's demo

We presented an online demo of our version identification system in 2009 at the International Society for Music Information Retrieval Conference (ISMIR) which was held in Kobe, Japan. With it, we assessed the output of a version similarity system through a graphical user interface (Fig. 1). The demo is still running at the moment of writing this thesis, although it is not publicly available (for more details please contact the author).

The system is based on the $Q^{*}_{\max}$ measure, i.e. it is based on the $Q_{\max}$ measure as explained in Chapter 3 and furthermore incorporates a version group detection layer such as the ones we have exposed in Chapter 4 (Serrà et al., 2009a, 2010d). For group detection we used the firstly proposed method in that chapter (PM1, Sec. 4.2.3). The recordings shown in this demo correspond to our music collection MC-2125 (Sec. 3.3.1). It has to be noted that, for favoring speed and due to some technical issues, all computations have been made off-line.

With this demo the user can browse a version collection via query-by-example. The results of the search are shown in a ranked list, together with the obtained



The girl from Ipanema - Astrud Gilberto

Song ID 822
Title The girl from Ipanema
Artist Astrud Gilberto
Original Artist Perry Ribeiro
Release Year 1963
Genre WorldMusic
Covers 11

Retrieved Songs

*	1	The girl from Ipanema	Astrud Gilberto	0.01
*	2	The girl from Ipanema	The peeping toms	0.04
*	3	Garota de Ipanema	Jobim and Moraes	0.05
*	4	Garota de Ipanema	Frank Sinatra	0.06
*	5	The girl from Ipanema	Gilberto Gil	0.06
*	6	Garota de Ipanema	Bobby Morganstein	0.06
*	7	Garota de Ipanema	Jobim Moraes and Toquinho	0.08
*	8	Garota de Ipanema	Rosa Passos and Ron Carte	0.08
*	9	Garota de Ipanema	Tamba trio	0.12
*	10	The girl from Ipanema	Najwajean	1.05

coverSSSearch demo 2009 - Copyright by Joan Serrà Julià - All rights reserved - http://mtg.upf.edu/people/jserra/

Figure 1: Snapshot of the online demo.

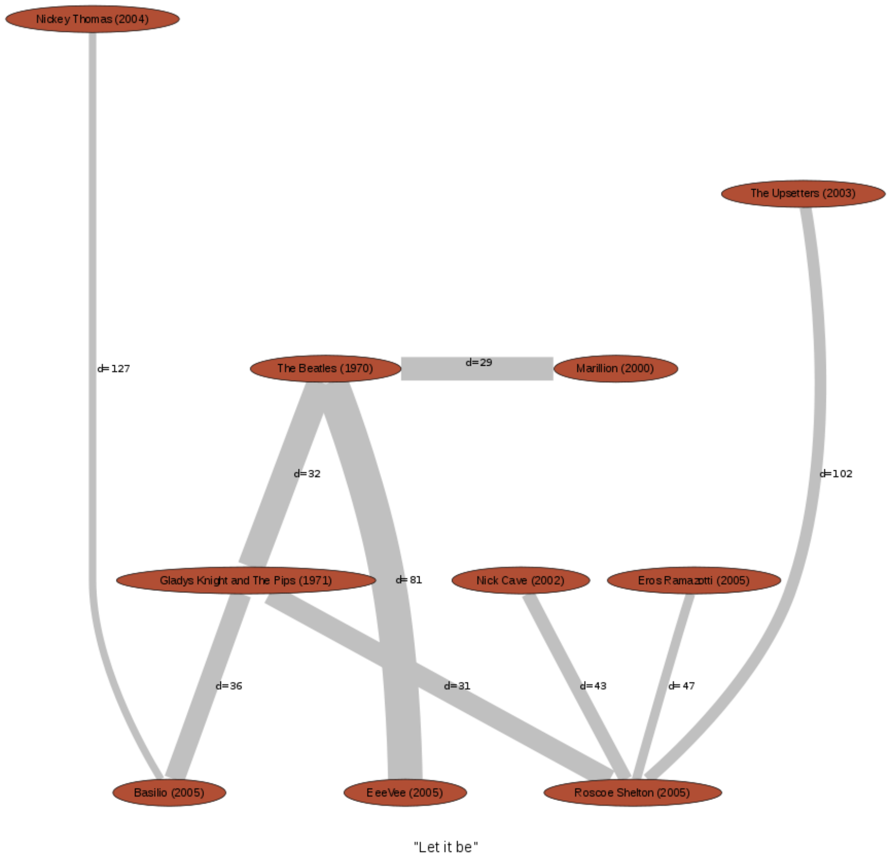


Figure 2: Detail of a version network.

distances to the query (Fig. 1, top right). For comparison purposes, metadata and ground truth information are also shown (Fig. 1, left). Furthermore, for exploring and visualizing the results of the system, a graph renderization for each automatically detected version set is depicted (Fig. 1, bottom right). A zoom on this part can be seen in Fig. 2. In the graph, nodes correspond to music recordings and edges reflect the similarity between these recordings.
To build such a graph we exploit $Q^*_{\max}$, which is reflected in the thickness of the edges (the thicker the edge, the more similar in terms of a tonal progression). Nevertheless, we also incorporate timbral similarity, which is reflected in the length of the edges (the shorter the edge, the more similar in terms of timbre). This timbral similarity is computed via the common Kullback-Leibler divergence between one-Gaussian mixture models of Mel-frequency cepstral coefficients extracted on a frame-by-frame basis [see e.g. Jensen et al. (2009) and references therein].

Appendix B: publications by the author

Submitted

Serrà, J., Zanin, M., Herrera, P., & Serra, X. (2010). Characterization and exploitation of community structure in cover song networks. Pattern Recognition Letters.

Serrà, J., Kantz, H., Serra, X., & Andrzejak, R. G. (2010). Predictability of music descriptor time series and its application to cover song detection. IEEE Trans. on Audio, Speech, and Language Processing.

ISI-indexed peer-reviewed journals

Bogdanov, D., Serrà, J., Wack, N., Herrera, P., & Serra, X. (2011). Unifying low-level and high-level music similarity measures. IEEE Trans. on Multimedia. In press.

Laurier, C., Meyers, O., Serrà, J., Blech, M., Herrera, P., & Serra, X. (2010). Indexing music by mood: design and integration of an automatic content-based annotator. Multimedia Tools and Applications, 48 (1), 161–184.

Serrà, J., Serra, X., & Andrzejak, R. G. (2009). Cross recurrence quantification for cover song identification. New Journal of Physics, 11, 093017.

Serrà, J., Gómez, E., Herrera, P., & Serra, X. (2008). Statistical analysis of chroma features in Western music predicts human judgments of tonality. Journal of New Music Research, 37 (4), 299–309.

Serrà, J., Gómez, E., Herrera, P., & Serra, X. (2008). Chroma binary similarity and local alignment applied to cover song identification. IEEE Trans. on Audio, Speech, and Language Processing, 16 (6), 1138–1152.

Other journals

Serrà, J., Zanin, M., & Herrera, P. (2011). Cover song networks: analysis and accuracy increase. Int. Journal of Complex Systems in Science, 1, 55-59.

Invited book chapters

Serrà, J., Gómez, E., & Herrera, P. (2010a). Audio cover song identification and similarity: background, approaches, evaluation, and beyond. In Z. W. Ras & A. A. Wieczorkowska (Eds.) Advances in Music Information Retrieval, Studies in Computational Intelligence, vol. 16, chap. 14, pp. 307–332. Berlin, Germany: Springer.

Full-article contributions to peer-reviewed conferences

Serrà, J., de los Santos, C. A., & Andrzejak, R. G. (2011). Nonlinear audio recurrence analysis with application to genre classification. In Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP). In press.

Serrà, J., Kantz, H., & Andrzejak, R. G. (2010). Model-based cover song detection via threshold autoregressive forecasts. In Proc. of the ACM Multimedia, Workshop on Music and Machine Learning (MML), pp. 13–16.

Lagrange, M. & Serrà, J. (2010). Unsupervised accuracy improvement for cover song detection using spectral connectivity network. In Proc. of the Int. Soc. for Music Information Retrieval Conf. (ISMIR), pp. 595–600.

Bogdanov, D., Serrà, J., Wack, N., & Herrera, P. (2009). From low-level to high-level: comparative study of music similarity measures. In Proc. of the IEEE Int. Symp. on Multimedia. Workshop on Advances in Music Information Research (AdMIRe), pp. 453–458.

Serrà, J., Zanin, M., Laurier, C., & Sordo, M. (2009). Unsupervised detection of cover song sets: accuracy increase and original detection. In Proc. of the Int. Soc. for Music Information Retrieval Conf. (ISMIR), pp. 225–230.

Laurier, C., Sordo, M., Serrà, J., & Herrera, P. (2009). Music mood representations from social tags. In Proc. of the Int. Society for Music Information Retrieval Conf. (ISMIR), pp. 381–386.

Akkermans, V., Serrà, J., & Herrera, P. (2009). Shape-based spectral contrast descriptor. In Proc. of the Sound and Music Computing Conf. (SMC), pp. 143–148.

Laurier, C., Meyers, O., Serrà, J., Blech, M., & Herrera, P. (2009). Music mood annotator design and integration. In Proc. of the Int. Workshop on Content-Based Multimedia Indexing (CBMI), pp. 156–161.

Serrà, J., Gómez, E., & Herrera, P. (2008). Transposing chroma representations to a common key. In Proc. of the IEEE CS Conf. on The Use of Symbols to Represent Music and Multimedia Objects, pp. 45–48.

Serrà, J. & Gómez, E. (2008). Audio cover song identification based on sequences of tonal descriptors. In Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pp. 61–64.

Serrà, J. (2007). A qualitative assessment of measures for the evaluation of a cover song identification system. In Proc. of the Int. Conf. on Music Information Retrieval (ISMIR), pp. 319–322.

Other contributions to conferences

Wack, N., Laurier, C., Meyers, O., Marxer, R., Bogdanov, D., Serrà, J., Gómez, E., & Herrera, P. (2010). Music classification using high-level models. Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.

Serrà, J., Zanin, M., & Herrera, P. (2010). Cover song networks: analysis and accuracy increase. Net-Works International Conf.

Herrera, P., Serrà, J., Laurier, C., Guaus, E., Gómez, E., & Serra, X. (2009). The discipline formerly known as MIR. Int. Soc. for Music Information Retrieval Conf. (ISMIR), special session on The Future of MIR.

Serrà, J., Zanin, M., & Andrzejak, R. G. (2009). Cover song retrieval by cross recurrence quantification and unsupervised set detection. Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.

Wack, N., Guaus, E., Laurier, C., Meyers, O., Marxer, R., Bogdanov, D., Serrà, J., & Herrera, P. (2009). Music type groupers (MTG): generic music classification algorithms. Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.

Bogdanov, D., Serrà, J., Wack, N., & Herrera, P. (2009). Hybrid similarity measures for music recommendation. Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.

Serrà, J. (2009). Assessing the results of a cover song identification system with coverSSSSearch. Int. Soc. for Music Information Retrieval Conf. (ISMIR), Demo Session.

Serrà, J., Gómez, E., & Herrera, P. (2008). Improving binary similarity and local alignment for cover song detection. Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.

Serrà, J. & Gómez, E. (2007). A cover song identification system based on sequences of tonal descriptors. Music Information Retrieval Evaluation eXchange (MIREX) extended abstract.

Theses

Serrà, J. (2007). Music similarity based on sequences of descriptors: tonal features applied to audio cover song identification. Master's thesis, Universitat Pompeu Fabra, Barcelona, Spain.

Patents

Serrà, J. (2009). Method for calculating measures of similarity between time signals. Patent application number US 12/764424.

Serrà, J. (2009). Método para calcular medidas de similitud entre señales temporales. Patent application number P 2009/01057.

Gómez, E., Herrera, P., Cano, P., Janer, J., Serrà, J., Bonada, J., El-Hajj, S., Aussenac, T., & Holmberg, G. (2008). Music similarity systems and methods using descriptors. Patent WO 2009/001202 .

Gómez, E., Herrera, P., Cano, P., Janer, J., Serrà, J., Bonada, J., El-Hajj, S., Aussenac, T., & Holmberg, G. (2008). Music similarity systems and methods using descriptors. Patent US 2008/0300702.

Additional and up-to-date information about the author may be found at the author's web page[5].

[5] http://joanserra.weebly.com