Computational Modelling of Expressive Music Performance in Jazz Guitar:

A Machine Learning Approach

Sergio Iván Giraldo

TESI DOCTORAL UPF / 2016

Director de la tesi

Dr. Rafael Ramírez
Department of Information and Communication Technologies

Universitat
Pompeu Fabra
Barcelona

To Camilo...

Acknowledgements

Four years have passed since I joined the Music Technology Group to begin my PHD. During this period, many people have come across (inside and outside the MTG) giving important contributions to this work. For *contributions* I want to refer not only to the theoretical aspects of the investigation, but also to other types of helping support. Received contributions have taken several forms, from babysitting my kid for free in times of work overload, to informal discussions about the investigation, performing tests, doing recordings, or giving very specific theoretical advise. The list is large, and the *contributors* are many:

Firstly i would like to express my gratitude to my advisor, Rafael Ramírez, who has been the main *contributor* to this work. Not only for his excellent advise and guidance, but also for his patience, support, and friendship. Also for considering me to actively participate in the research projects that have been and are being developed right now. I would like to thank to Xavier Serra, Perfecto Herrera, Emilia Gómez and Enric Guaus for their advise and help on several concrete topics needed for this dissertation. I would like also to acknowledge the blind reviewers of the different publications done during this research for their useful and constructive comments. Additionally I would acknowledge the financial support received by the TIMUL and TELMI projects.

Secondly, I would like to express my gratitude to Camilo, for his patience, support, and love (and for always being so enthusiast about coming to the lab with dad!). To my family who has been supporting me from de distance with love and interest. To Yeliz for her unconditional support and love during these last months.

Thirdly I would like to thank people and colleagues inside the MTG, specially to Zacharias Vamvakuosis, Nadine Kroner, Sankalp Gulati, Ajay Srinivasamurthy, Helena Bantulà, Jose Zapata, and also outside the MTG to Vanessa Picone, Danni Pucha, Anna Herrero, Joan Viñals, Alejandro Gallón, Paula Betancur, Flavia Becerra, Vicktoria Triebner and all the friends and colleagues who have also been direct and indirect *contributors* to this dissertation by helping with recordings, giving very useful musical feedback, participating on the events, doing annotations, correcting/translating to catalan, reading and correcting English typos, playing music together, babysitting Camilo... etc. To all of them I want to deeply acknowledge their help and support.

Abstract

Computational modelling of expressive music performance deals with the analysis and characterization of performance deviations from the score that a musician may introduce when playing a piece in order to add expression. Most of the work in expressive performance analysis has focused on expressive duration and energy transformations, and has been mainly conducted in the context of classical piano music. However, relatively little work has been dedicated to study expression in popular music where expressive performance involves other kinds of transformations. For instance in jazz music, ornamentation is an important part of expressive performance but is seldom indicated in the score, i.e. it is up to the interpreter to decide how to ornament a piece based on the melodic, harmonic and rhythmic contexts, as well as on his/her musical background. In this dissertation we present an investigation in the computational modelling of expressive music performance in jazz music, using the as a case study. High-level features are extracted from the scores, and performance data is obtained from the corresponding audio recordings from which a set of performance actions are obtained semi automatically (including timing/energy deviations, and ornamentations). After each note is characterized by its musical context description, several machine learning techniques are explored to, on one hand, induce regression models for timing, onset and dynamics transformations, and classification models for ornamentation to render expressive performances of new pieces, and, on the other hand, learn expressive performance rules to analyse its musical meaning. Finally. we report on the relative importance of the considered features, quantitatively evaluate the accuracy of the induced models, and discuss some of the learnt expressive performance rules. Moreover, we present different approaches for semi-automatic data extraction-analysis, as well as, some applications in other research fields. The findings, methods, data extracted, and libraries developed for this work are a contribution to expressive music performance field, as well to other related fields.

Resumen

El modelado computacional de la expresividad en la interpretación musical trata sobre el análisis y la caracterización de las desviaciones que, con respecto a la partitura, los músicos introducen cuando interpretan una pieza musical para añadir expresividad. La mayoría del trabajo en análisis de la expresividad musical hace énfasis en la manipulación de la duración y el volumen de las notas, y ha sido principalmente estudiada en en el contexto de piano clásico. Sin embargo, muy poco esfuerzo ha sido dedicado al estudio de la expresividad en música popular. Concretamente, en música jazz acciones expresivas como los ornamentos son una parte importante de la expresividad musical ya que estos no están indicados en la partitura y es tarea del músico hacer uso de los mismos añadiendo o substituyendo notas en la partitura. Los músicos añaden ornamentos teniendo en cuenta el contexto melódico, armónico o rítmico del tema, o bien según su experiencia en el lenguaje jazzístico. En este trabajo, presentamos una investigación en el modelado computacional de la expresividad musical en música jazz, tomando la guitarra eléctrica como caso de estudio. En primer lugar, extraemos descriptores de alto nivel de las partituras y obtenemos datos de la ejecución a partir de las correspondientes grabaciones de audio, de donde obtenemos semiautomáticamente la desviaciones temporales y de energía de cada nota, así como la detección de ornamentos. Después de que cada nota ha sido caracterizada por su contexto musical, varios algoritmos de aprendizaje automático son explorados para, de un lado, inducir modelos de regresión para duración, comienzo de nota y volumen, y modelos de clasificación para ornamentos para, finalmente, renderizar ejecuciones musicales expresivas. Por otra parte, aplicamos técnicas de inducción automática de reglas al conjunto de descriptores obtenidos para obtener reglas de ejecución musical analizando su sentido musical. Por ultimo, analizamos la importancia relativa de los descriptores considerados, cuantitativamente evaluamos la exactitud de los modelos y discutimos acerca de las reglas obtenidas. Igualmente, reportamos métodos para la extracción-análisis semi-automático de datos, asi como aplicaciones en otros campos de investigación. Los resultados, los métodos presentados, así como los datos extraídos y las librerías de código generadas para llevar a cabo esta investigación constituyen un aporte relevante en el campo de estudio computacional de la expresividad musical, así como en otras áreas de investigación relacionadas.

Resum

El modelatge computacional de l'expressivitat en la interpretació musical, tracta sobre l'anàlisi i la caracterització de les desviacions que els músics introdueixen quan interpreten una peça musical, per afegir expressivitat, respecte la partitura. La major part del treball en anàlisi de l'expressivitat musical fa èmfasi en la manipulació de la durada i el volum de les notes. La majoria dels estudis s'han fet en el context de piano clàssic i molt poc esforç ha estat dedicat a la música popular. Concretament, en música jazz, accions expressives com els ornaments, són una part important de l'expressivitat musical; Tot i no estar indicats en la partitura, és tasca del músic fer ús dls ornaments, afegir o substituir notes en la partitura, tot tenint en compte el context melòdic, harmònic o rítmic del tema, o bé segons la seva experiència en el llenguatge jazzístic. En aquest treball, presentem una recerca en el modelatge computacional de l'expressivitat musical en música jazz, prenent la guitarra elèctrica com a cas d'estudi. En primer lloc, extraiem descriptors d'alt nivell de les partitures i obtenim dades de l'execució a partir dels corresponents enregistraments d'àudio, d'on també obtenim semiautomáticament les desviacions temporals i d'energia de cada nota així com la detecció d'ornaments. Després que cada nota hagi sigut caracteritzada pel seu context musical, diversos algoritmes d'aprenentatge automàtic són explorats per a diferents fins. D'un costat, induir models de regressió per a la durada, el començament de nota i el volum, i models de classificació per a ornaments per, finalment, renderitzar execucions musicals expressives. D'altra banda, apliquem tècniques d'inducció automàtica de regles al conjunt de descriptors obtinguts, per obtenir regles d'execució musical analitzant les seves implicacions musicals. Per últim, analitzem la importància relativa dels descriptors considerats, quantitativament avaluem l'exactitud dels models i discutim sobre les regles obtingudes. Igualment, reportem mètodes per a l'extracció-anàlisi semi-automàtic de dades, així com a aplicacions en altres camps de recerca. Els resultats, els mètodes presentats, així com les dades extretes i les llibreries de codi generades per dur a terme aquesta recerca, constitueixen una aportació rellevant en el camp d'estudi computacional de l'expressivitat musical i en altres àrees de recerca relacionades.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> If you have to ask what jazz is, you will never know . . .
>
> Louis Armstrong

1.1 Motivation

In this dissertation we present an investigation in *computational modeling* of *expressive music performance*, focusing on jazz guitar musicas a case study. Unarguably, music performance plays an important role in our culture. People clearly distinguish the manipulation of sound properties by different performers and create preferences based on these differences. As depicted in Figure 1.1, these manipulations can be understood as variations in timing, pitch, and energy, that performers introduce (consciously and/or unconsciously) when performing a musical piece to add expression. Without expressivity music would lack of its human aspect, which is an integral part of it.

A performance without expression would result in a mechanical rendering of the score. Unconscious variations might be produced by micro-variations in timing, energy and pitch, which are specific to the performer intuitive style of playing. Other expressive variations might be introduced consciously for specific purposes, e.g. to convey emotions, or to play in a particular music style (e.g. swinging quavers). Moreover, melodic variations of the written music (melody) are usually introduce as an expressive resource in the form

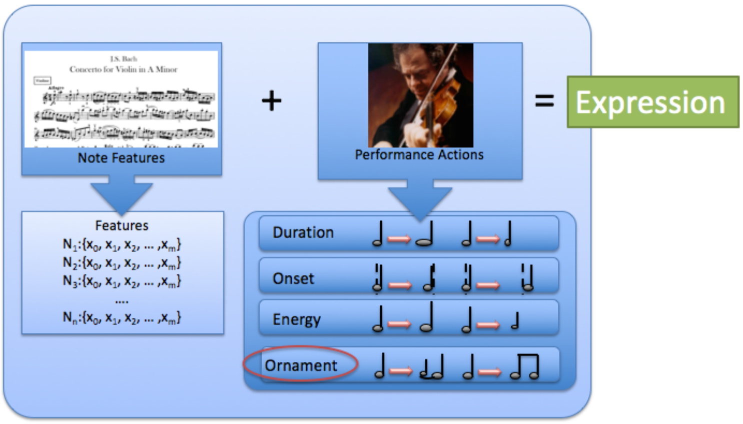


Figure 1.1: Performers introduce deviations in duration, onset, energy and note ornamentation to add expression

of ornaments. In classical music some of these expressive indications are present in the score. However, there is little quantitative information about how and in which contexts expressive performance occurs, as most of the performance information is acquired intuitively. This is particularly true in Jazz music where expressive deviations of the score are deliberately shaped by the musician. Performance annotations (e.g. ornaments and articulations) are seldom indicated in popular music (e.g. jazz music) scores, and it is up to the performer to include them by adding/substituting groups of notes based on melodic, harmonic and rhythmic contexts, as well as on his/her musical background. Therefore, in jazz music it may not always be possible to characterize ornaments with the archetypical classical conventions (e.g. trills and appoggiaturas). Furthermore, the performance of the melody in jazz context may lay in between an explicit performance of the notes of the score and free improvisation (also called *free ornamentation*). In Figure 1.2 a transcription of the performance of the first four bars of the jazz piece "*Yesterdays*" (by J. Kern) as performed by Wes Montgomery, illustrate this aspect of free ornamentation. Vertical arrows show the correspondence between the groups of notes used to ornament the melody and the actual score.

The problem of explaining the melodic expressive deviations in the performance of a jazz melody lay down in the domain of *expressive music performance* (EPM) research. EMP has become an important scientific domain of

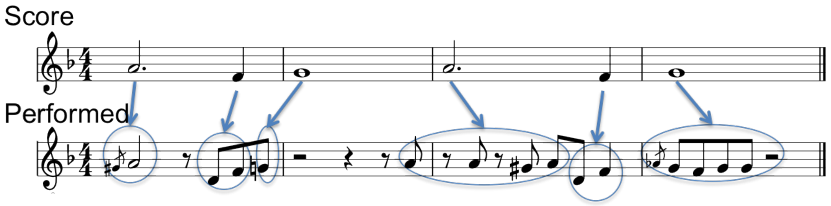


Figure 1.2: Free ornamentation in jazz music. A fragment of Yesterdays by J. Kerm as performed by jazz guitarist Wes Montgomery. Vertical arrows indicate correspondence between performed notes to *parent* score notes.

study over the last decades, aiming to quantify and analyze the expressive qualities imposed by the performer on to a otherwise "*dead*", *neutra*l or *inexpressive* music score (also called *nominal performance*). Several studies on EMP research have been conducted since early 20th century (Gabrielsson (1999, 2003); Palmer (1997)) from different perspectives (e.g. musicological, psycological, cognitive), and it has been mainly studied in classical music. The measurement of the different aspects of expression in a musical performance implies the need of using tools for large-scale data analysis. Therefore, computational approaches to study EPM have been proposed, in which the data extracted from real performances is used to formalize expressive models for different aspects of performance (for an overview see Goebl et al. (2008)). Thus, *computational systems for expressive music performance* (CEMPS) have been designed aiming to automatically generate human-like performances of *inexpressive* scores by introducing expressive variations in timing, energy, and pitch, based on both learning and non-learning computational models (for an overview see Kirke and Miranda (2013)). However, most of the proposed expressive music systems are targeted to classical piano music. Some exceptions include the work by (Arcos et al., 1998) and Ramírez and Hazan (2006). The former describe a system able to infer Jazz saxophone expressive performances from non-expressive monophonic descriptions using Case Based Reasoning. The later applies inductive logic programming to obtain models capable of generating and explaining expressive jazz saxophone performances.

Aforementioned approaches for jazz saxophone music (Arcos et al., 1998; Ramírez and Hazan, 2006) are able to predict ornamentation (among other performance actions). Approaches such as Grachten (2006) detects ornaments of multiple notes, to render expressive-aware tempo transformations. Other methods are able to recognize, characterize ornamentation in popular music, e.g. (Gómez et al., 2011; Perez et al., 2008). However, due to the complexity of free ornamentation, most of these approaches characterize ornamentation in constrained settings, for instance by restricting the study to one-note ornamentation or to notated trills in classical scores Puiggròs et al. (2006). The study of jazz guitar melody performance in the context of computational music expression modeling, considering complex ornamentation is an open research question that, to our knowledge, has not been studied in the past.

In this dissertation we present an investigation in the computational modelling of expressive music performance in jazz music, using the electric guitar as a case study. The aim of this project is two fold. Firstly, we present a machine learning approach to automatically generate expressive (ornamented) jazz performances from un-expressive music scores. Secondly, we present a data driven computational approach to induce expressive performance rule models for note duration, onset, energy, and ornamentation transformations in jazz guitar music. As a first step, high-level features are extracted from the scores and performance data is obtained from the corresponding audio recordings, by proposing methodologies to improve automatic performance transcription such as pitch profile extraction optimization and rule based and energy filters. A second step consists on comparing the similarity between the score and the performance by means of a Dynamic Time Warping approach, from which a set of performance actions are obtained semi automatically, which include timing/energy deviations, as well as, ornamentations. After each note is characterized by its musical context description, several machine learning techniques are explored to, on one hand, induce regression models for timing, onset and dynamics (i.e. note duration and energy) transformations, and classification models for ornamentation to later select the most suitable ornament for predicted ornamented notes based on note context similarity. On the other hand, we apply machine learning techniques to the resulting features to learn expressive performance rule models. We analyse the relative importance of the considered features, quantitatively evaluate the accuracy of the induced models, and discuss some of the learnt expressive performance rules. Experiments for semi-automatic data extraction and analysis are presented reporting on improvements obtained based on accuracy measures for each specific approach. We also report on the performance for the ornamentation, duration, onset, and energy models and rule learners . Similarities between the induced expressive rules and the rules reported in the literature are discussed.

The rest of this chapter is organized as follows. In Section 1.2, we formally define the research problem and main objectives. In Section 1.3 we comment about the scope of the investigation. Section 1.4 we define the scientific context and mention the areas of research linked to the main topic. In Section 1.5 we point out the the main contributions of this investigation. Finally, we give a brief overview of the structure of the dissertation in Section 1.6.

1.2 Problem Definition

The motivation of this investigation is to imitate (somehow) the way in which popular (concretely jazz) music is taught, by means of learning (copying) from the performances of expert musicians. Thus, the primary objective of this research is to develop a system for expressive music performance analysis and synthesis, in the context of jazz melodic performance, using the electric jazz guitar as a case study, to generate predictive models for expressive performance deviations in onset, duration, and energy, as well as complex note ornamentations. These expressive deviations will be calculated based on the measured differences between recorded performances of professional musicians and its respective scores. The expressive performance models will be trained using information (features) extracted from the score notes. We aim to apply and compare different learning algorithms to the task of performance deviation prediction and render human-like performances of new pieces. Throughout this dissertation we will use the term *Expressive Performance Actions* (EPAs) to refer to the mentioned deviations introduced by musicians to add expression. The general framework of the system is depicted in Figure 1.3.

Some of the research questions that arise within a methodology proposed in this investigation are: which learning schemes outperform at different particular performance deviations? The use of a predictive model is better than simply retrieving the most similar deviation performed on a particular case? is possible to obtain predicted rendered performances comparable to the ones made by a human? Furthermore, our goal is not only to obtain predictive models, but use the information that models provide in order to achieve a better understanding about how musicians make choices upon applying these performance deviations over a melody, based on the musical context of the piece. For instance, which of the score musical context attributes are the most relevant inputs to influence the decision of applying certain note deviation (e.g. ornament, enlarge/shorten, advance/delay)? Do the rules on how to apply these deviations obtained by a system have (or might have) any musical interpretation?

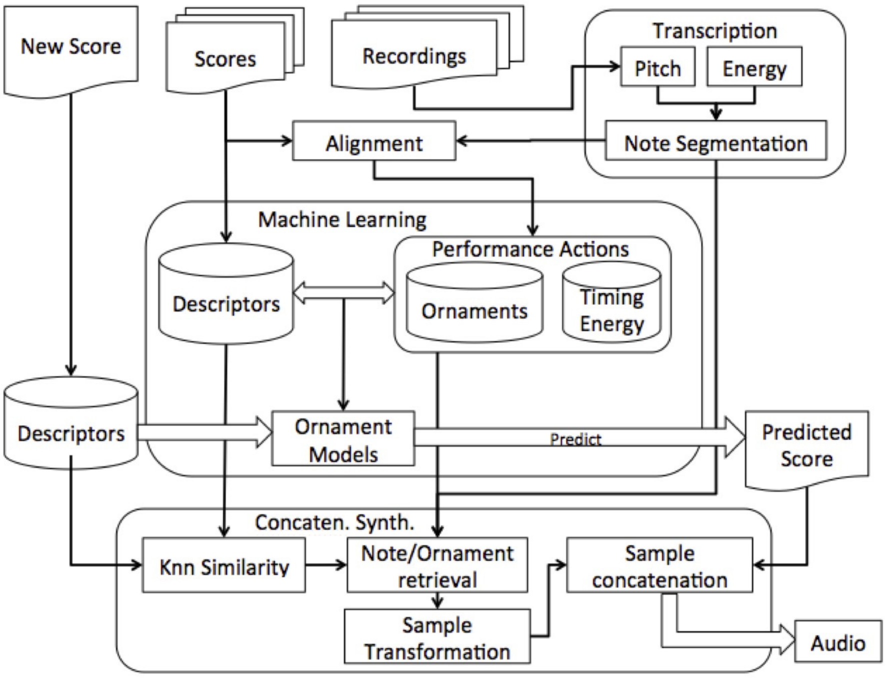


Figure 1.3: General framework for jazz guitar ornament modelling.

The proposed methodology imply several tasks that the system should achieve:

- *Musical context description*: investigate and develop tools for feature note extraction, in terms of the nominal properties of the notes, the musical context of the note within the score, as well as, melodic analysis based on musicological aspects or principles (e.g. expectation models).
- *Melodic transcription*: investigate on the methods for melodic extraction and automatic transcription for monophonic-monotimbral audio as well as monophonic-multitimbral audio, to obtain a machine representation of the recorded pieces in terms of note events (onset, duration, pitch, and energy)
- *Score to Performance Alignment*: investigate on methods for automatic score to audio alignment, to develop a methodology to obtain

a correspondence between performed notes and parent notes in the score.

- *Performance Actions Definition*: Define and quantify the performance deviations introduce by the performer, which include variations in timing energy and pitch as well as the detection and coding of ornaments.

1.3 Scope

To make the research feasible, some considerations about the scope of the investigation has been taken into account. The EPAs considered for this study are deviations in *duration*, *onset*, *energy* as well as the use of *ornamentation*, understood as the action of replacing a score note by subtracting or adding a note or a group of notes. The corpus of musical pieces consist of *jazz standards* taken from The real book. Melodies are performed monophonic, obtaining the audio signal directly from the pickups of an electric guitar, which facilitates the recording of the performance and the melodic performance transcription. A second type of performance data is obtained from commercial available audio recordings. In this investigation we constraint our analysis to the (so called) *natural* type of expressivity, in which the performer plays the melody based on his personal criteria on how he/she believes the piece should sound within jazz style, without any emotional expressive intention. However an application in mood emotional modeling is presented in Chapter 7. We mainly focus on musical content extraction/analysis from data, whereas for audio and signal processing tasks, we will rely on existing implementations for melodic extraction from polyphonic signals (Salamon and Gómez, 2012), pitch detection De Cheveigné and Kawahara (2002), beat tracking (Zapata et al., 2012), available in the *Essentia* library, as well as for pitch and time scaling (e.g. Serra (1997)).

1.4 Scientific context

As mentioned previously our investigation lays down in between the scientific domain of expressive music performance and computer science. The different research tasks involved in this investigation (see Section 1.2) may involve several areas of research as depicted in Figure 1.4. Data analysis from score and note description may involve the use of models for *melodic music analysis*, for which we make use of models of melodic expectation and melodic complexity from the domain of *music perception and cogni-*

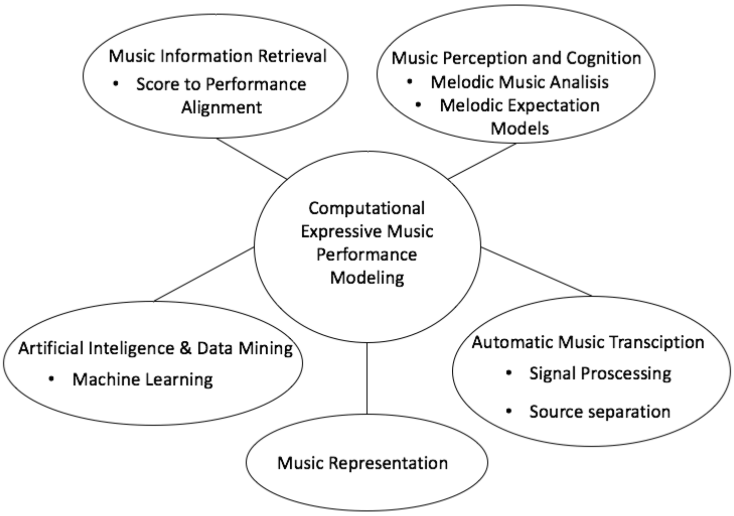

Music Information Retrieval
• Score to Performance Alignment
Music Perception and Cognition
• Melodic Music Analisis
• Melodic Expectation Models
Computational Expressive Music Performance Modeling
Artificial Inteligence & Data Mining
• Machine Learning
Automatic Music Transciption
• Signal Proscessing
• Source separation
Music Representation

Figure 1.4: Scientific context for computational modeling of expressive music performance.

tion, and *music representation* to transform it in a machine readable data. Data extraction from both monophonic (mono-timbral) and monophonic (polytimbral) audio signals may require techniques derived from the fields of *automatic music transcription*, *source separation* and *signal processing.* A key step is to find the correspondence between performed to *parent* score notes, which involve techniques for *score to performance alignment* used in the field of *music information retrieval.* Finally, the use of *machine learning* techniques from the domain of *artificial intelligence and data mining* are required to obtain predictive models for EPAs.

1.5 Contributions

The main contributions of this work can be summarized based on the different research topics involved in this investigations (explained in section 1.4) as follows.

1. **Music expressive performance**

    - A methodology for expressive performance modeling in jazz music, able to operate on complex or free ornamentation, using the jazz guitar as a case study.

- An statistical analysis of the performance of different algorithms used in the prediction of specific EPAs.
- Analysis of the relevance of the features involved in specific performance actions.

2. **Music representation and melodic description**

    - A methodology for note feature extraction and description.
    - The introduction of perceptual features for melodic description.

3. **Automatic music transcription**

    - A methodology for music representation of the performance based on the manipulation of pitch contours extracted from both Monophnic and polyphonic audio into note events.
    - A parameter optimization of the algorithm for melodic extraction from polyphonic signals, to the case study of the electric guitar in jazz misc context, by using genetic algorithms.

4. **System applications**

    - A system for neuro-feedback for real-time manipulation of expressive parameters based on the perceived emotional state.

1.6 Outline of the Dissertation

The outline of this dissertation is organized as follows. In Chapter 2 we present an overview of the state of the art in expressive music performance research, and the topics involved in this investigation. We focus on the review of the computational approaches for expressive music performance systems from the perspective of learning and nonlearning systems, and deepen in the ones targeted for jazz music. Other topics reviewed include automatic melodic transcription systems, melodic description and analyisis, and score to performance alignment.

In Chapter 3 we describe the data acquisition stage used for this work. We describe the musical corpus used in terms of the music scores and the recording material. We give an explanation on the process to obtain a machine representation of both the scores and the recorded performances for the monophonic- monotimbral case as well as for the monophonic multitimbral case.

In Chapter 4 we explain the methods used to extract note descriptors from the score data, as well as type of descriptors used. We also describe the methodology used to obtain the EPAs, which involve a methodology for score to performance alignment, to obtain the correspondence between performed and *parent* score notes. Finally, we comment on the process of the creation of a database of EPAs and note descriptors.

In Chapter 5 experiments for semi-automatic data extraction and analysis are presented reporting on improvements obtained based on accuracy measures for each specific approach.

In Chapter 6 we describe the modeling stage of the work. For monophonic monotimbral performances we obtain classification models for ornamentation, and regressive models for duration, energy and onset deviations. For monophonic monotimbral models, after discretizing the data we extract rules for classification for all the EPAs. In the first scenario note concatenation is performed to obtain a MIDI representation of the predicted pieces. In the second scenario an analysis of the obtained rules is performed in terms of its musical sense. In both scenarios we analyze which melodic descriptors are the most influential for each EPA by means of feature selection. We present an evaluation of the models in terms of its accuracy (for classification models) and correlation coefficient and explained variance (for regression models). An statistical improvement analysis against a base line is performed as well as learning curves on the variation of different relevant parameters (e.g. number of instance, attributes, and/or algorithm parameters). We perform an evaluation of the system by computing the alignment distance between the system and the target performance. Similarly, we discuss on the musical sense of the rules obtained and the feature selection process.

In Chapter 7 we present two applications of our the computational modeling approach. The first is a system for emotion modelling in real time with a direct application in neurofeedback. The second is a computational approach for music performance interaction analysis in jazz music.

Finally, In Chapter 8 we give a summary of the research presented, we comment on the main contributions of the work and explain the future directions of the current research.

Chapter 2

Background

In this chapter we provide a review of the state of the art in Expressive Music Performance Research. Music Expression has been empirically studied since the beginning of XIX century. This research area investigates the manipulation of sound properties that musicians (consciously or unconsciously) introduce when performing a musical piece to add expression. Research in Expressive Music Performance aims to understand and recreate expression in performances. We will start by pointing out some aspects of expressive performance and will define which are the expressive strategies a performer use to create expression. We will define what is a Computer System for Expressive Music Performance (CEMPS), and comment on the most relevant systems. We will categorize CEMPS based on how the performance knowledge is built on these systems. We will briefly review some previous work on ensemble jazz performance. Later, we will comment some previous work on the related tasks implied in this work, such as *Automatic Melodic Transcription*, and *Automatic Ornament Recognition*. Finally, we will briefly comment on the background concerning to applications developed in this project related to EEG emotion recognition, and musical neuro-feeback.

2.1 Expressive Music Performance Overview

There have been different attempts to define *expression* in music performance. Juslin (2001) defines it as the variations in timing, dynamics, timbre and pitch that makes possible to differentiate one performance from another. Goebl et al. (2008) defines it as an integral part of music: "*without expression, music would not be interesting for most part of listeners*, (...)

what listeners go to listen, when going to a concert, is the human expression that gives sense to music".

2.1.1 Why do performers play expressively?

This research question has been widely investigated in the past. Hypotheses suggests that performers often wish to express emotions (Juslin, 2001), and/or, that playing expressively helps to clarify the musical structure of a piece (Roger A. Kendall, 1990). Several studies published in music performance, specifically in baroque classical and romantic music (e.g. Seashore (1938), Palmer (1997) and Gabrielsson (2003)), aim to discover the musical features (aspects) that performers take into consideration to introduce expression in performances. One of these features is the hierarchical structure of music (Lerdahl and Jackendoff, 1983). Performers tend to express this hierarchy (i.e. notes → motifs → phrases → sections) by slowing down the tempo at the boundary of each structure in proportion to the level of the hierarchy. Another regular feature used is that higher pitched notes tend to be played louder, as well as notes what create melodic tension (relative to the key).

2.1.2 Expressive performance actions

Expressive performance actions can be defined as the strategies and variations introduced in a performance, which are not specified by the score. The most common expressive performance actions are performed as timing, duration, onset, and loudness deviations. Other common performance actions are ritardando, which is to slow down the tempo as the performer reaches the end of the piece or a segment or phrase. Articulation is the action of performing notes more legato (smoothly linked) or staccato (short and pronounced). Expressive intonation may be introduced in instruments with continuous pitch (e.g. string instruments), playing notes sharper or flatter or introducing vibrato. Expression may also be introduced by timber variation.

2.2 Research in Expressive Music Performance

Expressive music performance studies the micro variations a performer introduce (voluntary or involuntary) when performing a musical piece to add expression. Several studies investigating this phenomenon have been conducted from an empirical perspective,e.g. Gabrielsson (1999, 2003); Palmer

(1997). Computational approaches to study expressive music performance have been proposed, in which data is extracted from real performances and, later, used to formalize expressive models for different aspects of performance (for an overview see Goebl et al. (2008, 2014)). Computational systems for expressive music performance (CEMP) are targeted to automatically generate human-like performances by introducing variations in timing, energy, and articulation (Kirke and Miranda, 2013).

2.2.1 Computational Approaches: Non learning models

Two main approaches have been explored to computationally model expression. On one hand, expert-based systems obtain their rules manually from music experts. A relevant example is the work of the KTH group (Bresin and Friberg, 2000; Friberg et al., 2006; Friberg, 2006). Their *Director Musices* system incorporates rules for tempo, dynamic, and articulation transformations. Other examples of manually generated expressive systems are the Hierarchical Parabola Model (Todd, 1989, 1992, 1995), and the work by Johnson (1991a) who developed a rule-based expert system to determine expressive tempo and articulation for Bach's fugues from the Well-Tempered Clavier. The rules were obtained from two expert performers.

2.2.2 Computational Approaches: Learning models

Machine-learning-based systems obtain their expressive models from real music performance data by measuring the deviations of a human performance with respect to a neutral or *robotic* performance, using computational learning tools. For example, neural networks were used by Bresin (1998) to model piano performances, and by Camurri et al. (2000) to model emotional flute performances. Rule-based learning algorithms were used by Widmer (2003) to cluster piano performance rules. Other piano expressive performance systems worth mentioning are the ESP piano system by Grindlay (2005b) which utilize Hidden Markov Models, and the generative performance system of Miranda et al. (2010a) which uses genetic algorithms to construct tempo and dynamic curves.

Most of the proposed expressive music systems are targeted to classical piano music. More recently, there have been several approaches to computationally model expressive performance in popular music by applying machine learning techniques. Arcos et al. (1998) report on SaxEx, a performance system capable of generating expressive solo saxophone performances in Jazz, based on case-based reasoning. Ramírez and Hazan (2006) compare different machine learning techniques to obtain jazz saxophone performance models capable of both automatically synthesizing expressive performances and explaining expressive transformations. Grachten (2006) applies dynamic programming using an extended version of edit distance, and case-base reasoning to detect multiple note ornaments and render expressive-aware tempo transformations for jazz saxophone music.

In previous work (Giraldo, 2012; Giraldo and Ramírez, 2015a,d,c), ornament characterization in jazz guitar performances is accomplished using machine learning techniques to train models for note ornament prediction.

2.2.3 Computer systems for expressive performance (CEMPS)

Computer systems for expressive music performance (CSEMP) have been developed since early 1980 as a result of the growing use of sequencers and computers and the introduction of MIDI protocol, which made possible a standardized way of communication and synchronization between sequencers. These sequencers and computers were able to perform stored tunes in perfect metronome timing. However, these performances sounded robotic, as they lack the variations humans introduce to play expressively. Thus, a CSEMP is a computer system that aims to generate human-like expressive music performances. Kirke and Miranda (2013) classify CSEMPs in automated and semi-automated. An automated CSEMP has the ability (after set up and/or training) to generate a performance of a musical piece not seen before by the system, without manual intervention. A semi-automated system will require some manual input such as musicological analysis. This automated or manual analysis is done to obtain rules that control the transformations applied to a particular note (or group of notes) in order to achieve a desired expressive transformation. This set of rules will be referred as *Performance Knowledge*.

CEMPS can be grouped based on how its performance knowledge was built (rules that the system follows to apply expressive deviations from the score). The list of the systems, corresponding grouping, instrument model, and music style are shown in Table 2.1.

CEMP	Instrument	Style
Non learning systems		
Director of musices: Friberg et al. (2006)	All (Piano)	Classical
Hierarchical parabola model: Todd (1989, 1992, 1995)	Piano	Classical
Composer pulse and predictive amplitude shaping: Clynes (1995, 1986)	All	Classical
Bach fugue system: Johnson (1991b)	Keyboard	Classical
Trumpet synthesis: Dannenberg et al. (2007); Dannenberg and Derenyi (1998)	Trumpet	Classical
Rubato: Mazzola G. (1994); G. (2002)	All (Piano)	Classical
Pop E: Hashida et al. (2007)	Piano	Classical
Hermode tuning: W. (2004)	All	Baroque, Jazz/Pop
Computational music emotion rule system: Livingstone et al. (2010)	Piano	-
Linear regression		
Music interpretation system: Katayose et al. (1990)	Piano	Classical
CaRo: Canazza et al. (2000, 2001, 2003)	All	-
Artificial neural networks		
Artificial neural network piano system: Bresin (1998)	Piano	-
Emotional flute: Camurri et al. (2000)	Flute	Classical
Case and instance based systems		
SaxEx: Arcos et al. (1998)	Saxophone	Jazz
Kagurame: Suzuki et al. (1999)	Piano	Classical
Ha-Hi-Hun: Hirata and Hiraga (2002)	Piano	Classical
PLCG system: Widmer (2000, 2002, 2003)	Piano	Classical
Combined phrase decomposition PLCG: Widmer and Tobudic (2003)	Piano	Classical
Distall system: Tobudic and Widmer (2003)	Piano	Classical
Statistical graphical models		
Music plus one: Raphael (2001b,a, 2003)	All	Classical

ESP piano system: Grindlay (2005a)	Piano	Classical
Other regression models		
Drumming system: Carlson et al. (2003)	Drums	Electronic music
KCCA piano system: Dorard et al. (2007)	Piano	Classical
Evolutionary computation		
Genetic programming jazz sax: Ramirez et al. (2008)	Saxophone	Jazz
Multiagent system with imitation: Miranda et al. (2010b)	Piano	-
Ossia: Dahlstedt (2007)	Piano	Contemporary

Table 2.1: List of CEMPS grouped by performance knowledge

From the systems presented in Table 2.1, we deepen on some of the most commonly cited CEMPS in the literature, which have been influential for this study:

Director of Mucises (KTH)

The KTH rule system for music performance by Friberg et al. (2006) consists of a set of about 30 rules that control different aspects of expressive performance. These set of rules are the result of research initiated by Sundberg (1993); Sundberg et al. (1983) and Friberg (1991). The rules affect various parameters (timing, sound level, articulation) and may be used to generate expressive musical performances. The magnitude of each rule is controlled by a parameter k. Different combinations of k parameters levels model different performance styles, stylistic conventions or emotional intention. The result is a symbolic representation that may be used to control a synthesizer. A real-time based implementation of the KTH system is the pDM (Pure Data implementation of Director Musices Program) by Friberg (2006). Friberg implements an arousal/valence space control, defining a set of k values for the emotion at each quadrant of the space. Seven rules plus overall tempo and sound level are combined in such a way that they clearly convey the intended expression of each quadrant of the 2D emotional plane based on the research by Bresin and Friberg (2000) and Juslin (2001). Intermediate values are interpolated when moving across the plane.

Computational Music Emotion Rule System (CMERS)

The system proposed by Livingstone (2010) Livingstone et al. (2010) uses a similar approach to the KTH rule system. It is based on 19 rules obtained by analysis-by-synthesis applied to a phrase level hierarchy. The system uses micro-features and macro-features to perform deviations in the score, generating human-like performances, and also making possible to express emotions. It uses a 2D emotional plane in which one axis ranges from negative to positive level, and the other ranges from passive to active states, similar to the arousal-valence plane which will be explained in section 2.4. Authors label each quadrant in the 2D emotional plane representing emotions such as angry, bright, contented and despairing. Rules to convey emotions range from mayor to minor modes changes, tempo and dynamic variations, as well as, micro-variations introduced by humanization rules. Authors claim that CMERS is more successful than DM in conveying emotions, based on listening tests.

CaRo

Carranza et.al 2000 - 2001 Canazza et al. (2000), Canazza et al. (2001), Canazza et al. (2003) report on a system able to morph monophonic audio signals to convey different emotions in the emotional 2D plane. It represents performance actions at the local note level, which include inter-onset interval changes, brightness and loudness-envelope centroid. A linear model is used to characterize each performance action. The system learns how performances are perceived by listeners in terms of moods (hard, heavy, dark, bright and soft). This is done by analyzing the variability of results in listening experiments. A user can select any point in the 2D emotional space and generate a new expressive version of the piece. A trajectory line can be drawn in the 2D emotional space morphing different moods, in real-time.

Emotional Flute

Camurri et. al (2000) Camurri et al. (2000) presents a system, which uses explicit features and artificial neural networks (ANN). Features are similar to the ones used in the KTH rule system. Expressive actions include inter-onset intervals, loudness, and vibrato. Another ANN is used to segment the musical piece into phrases, and separate nets are used for timing, loudness (cressendo/decressendo), and duration. Two models were generated to handle vibrato. The system was trained with the performance of a flautist in nine different moods (cold, natural, gentle, bright, witty, serious, restless, passionate and dark). Performances were mapped into a 2D emotional space. Listening tests gave an accuracy of 77% when listeners attempt to label a particular emotion to rendered performances.

PLGC system

A long term multi-disciplinary project is reported by Widmer (2000) Widmer (2002), Widmer (2003), to study expressive music performance by the use of intelligent data analysis. The PCLG (Partition Cluster Learn Generalize) algorithm is an ensemble rule learning algorithm that aims to learn simple robust principles from complex data in the form of rules. The learning approach of the PLCG algorithm is divided into 4 stages. Firstly, a large dataset of training examples (which consisted of note descriptors along with the corresponding performance deviations), is obtained from the recordings of Mozart piano sonatas performed by a professional pianist. The data set is partitioned into smaller data subsets. Then, the FOIL algorithm is used to build rules for the considered performance actions, using each subset of training examples. Later, the most common rules are grouped by applying hierarchical clustering among the obtained rules using a syntactic-semantic rule similarity measure. Generalization is measured based on the coverage of the rules which is used as a stopping criteria for rule selection. Three performance actions are taken into consideration: tempo (*ritardando/accelerando*), dynamics (*crescendo/diminuendo*), and articulation (*staccato, legato, and portato*). Rules performance is measured based on the true/false positives coverage on the training set, as well over a test set. This test set consisted of the performance recordings of the same piano sonatas by another pianist, and 22 Chopin piano sonatas performed by 22 different pianists. Little degradation in rule coverage is interpreted as a good indicator of generality of the rule among performers. On the contrary high degradation is interpreted as unpredictability of the performance action.

2.2.4 Computer systems for expressive performance actions in jazz

From table 2.1 is clear that most of music expression research has been done in the context of classical music, concretely in piano classical music. These might be due to the fact that the piano keys can be easily treated as on/off switch devices. Moreover, most of the digital pianos have in-built MIDI interfaces. This facilitates the extraction on of the performance data in the form of musical events in a machine readable format, and avoids the complications derived from performing audio signal processing to obtain the data. In this section we outline the CEMPS presented in Table2.1 targeted to jazz performance.

Automatic rule induction in jazz performance

Ramírez and Hazan (2006) compares different machine learning techniques to obtain a jazz saxophone model capable of generating synthesized expressive performances and to explain the expressive transformations. They implement a system which uses inductive logic programming (ILP) which creates a logic set of rules that model expression from both intra-note and inter-note level. Intra-note level refers to a set of features that are relevant to the note itself, like pitch, attack, onset (inflections), while inter-note level refer to the musical context at which the note appears, like interval with the previous and the next note, duration and loudness. Thus, considering a set of inflections (intra-note level) and the musical context (inter note level), the system could predict the type of inflection to be used in a particular context. Using this information, they model various performers' styles of playing, so when a new performance is presented, the system is able to identify the performer by analysing the performance style. The intra-note set of features are represented by: attack level, sustain duration, legato (left-right), energy mean, spectral centroid and spectral tilt. For inter-note features set are Pitch, Duration, Previous note pitch, Next note Pitch, Next duration, and 3 set of Narmour structures. Depending on the type of instrument, the aspects of the performance to be taken in account for the performer identification task may vary. For example, in piano, dynamics and timing are relevant aspects for performer identification, but not timber, conversely to saxophone or singing voice, where timber is the most relevant attribute for this task. Two classifiers are used: one to map melodic fragments to the different possible performers, and a note classifier that maps each note of the performer to be identified into an Alphabet symbol, which is the set of clusters generated by all the notes performed by all performers. These classifiers are obtained by the use of different machine learning techniques: K-means clustering, Decision trees, Support Vector Machines, Artificial Neural Networks, Lazy Methods and Ensemble Methods.

The training recordings are segmented and the intra-note descriptors are computed for each note. Fuzzy k-means clustering is applied using the intra-note information to group similar notes. Then for each performer, the training recordings of that performer are collected, and inter-note descriptors are computed for each segmented note in the performer's recording. A classifier (Decision Tree) is build using inter-note features as attributes and the cluster previously calculated as a class. By clustering inter-note features they obtain sets of similar notes for all performers, and by building decision trees with intra-note features, they predict the notes a performer will play within a musical context.

Genetic programming

Ramirez et al. Ramirez et al. (2008) compares different machine learning techniques to obtain a jazz saxophone model capable of generating synthesized expressive performances and to explain the expressive transformations. Among the techniques they explore are genetic algorithms (GA) and inductive logic programming (ILP). They implement a system, which uses ILP to induce logic set of logic rules that model expression from both intra-note and inter-note level. Considering a set of inflections (intra-note level) and the musical context (inter note level), the system could predict the type of inflection to be used in a particular context. In addition, using this information they model various performers styles of playing, and train a classifier to identify different performer by their playing style.

Case Base Reasoning in jazz performance

Another system for jazz modeling is the one proposed by Arcos et al. (1998), the SaxEx. They report a system able to infer an expressive performance from a flat, non monophonic input, by using Case Based Reasoning. The musical context role of each note of the input is analyzed, and the system retrieves from a case memory of human performances, notes with similar roles, and using its properties transform the input notes. A first step of the note analysis is performed by spectral modelling techniques (Bonada et al., 2011), by comparing qualitative values (a.e. dynamics in dB) over a single note, and compare this value to the average value over the whole preformed piece. The second step of the note analysis uses Narmour's model (Narmour, 1992) and Generative Theory of Tonal Music (Lerdahl and Jackendoff, 1983) to determine the role of the note within the musical phrase: place on the melodic progression, metrical strength, duration, harmonic stability and relative importance in the bar. However this model can not explain the transformations realized to the inexpressive performance.

2.2.5 Jazz Guitar Expressive Performance Modeling

To our knowledge, the only published work towards Expressive Performance Modeling in Jazz Guitar was done by Giraldo (2012). The melody was extracted from a jazz guitar teaching series book Marshall (2000) in which the melody was recorded separately from the rhythmic section. This facilitated the note segmentation process which was done in a semi automated way (automatic onset recognition plus manual correction). A critical stage in the modelling process is to find which groups of notes of the performed score correspond to an embellished note in the original score. The correlation between score and performed notes was done manually from musicological knowledge by a professional jazz musician. After having the segmented audio melody, the inexpressive score, the performed score, and the score/performed note correspondence, we proceeded to obtain features for each note, and also a database of embellishments. We used machine-learning techniques to obtain models for duration, energy and embellishments variations. Feature selection was manually performed by choosing the features that would correlate better with the performance actions to be modeled, based on an expert musical knowledge. This initial selection was validated later using standard feature selection methods such as wrapper (ref) with forward and/or backward elimination. The best accuracy results were obtained with K star algorithm (REF). Tests were performed using the piece ”Yesterdays” performed by Wes Montgomery as training set, and first eight bars of the piece ”Autumn Leaves” for testing. Quantitative evaluation was performed based on listening tests.

2.2.6 Ensemble performance modeling

So far, we have review music expressive performance as the study of the transformations from the written score to actual performance of a musical piece. We have seen how CSEMPs make use of several techniques for analysis of recorded performances, extracting knowledge (in the form of rules or patterns) to generate both learning and non learning models of performance. Computer systems make use of the models to automatically generate human-like expressive performances. However, most of the studies have been limited to solo expressive music performance, and little work has been done in ensemble expressive music performance modelling, i.e. playing expressively in an ensemble, where complex interactions between musicians (playing several harmonies and melodic lines concurrently) may occur. Some studies have addressed the problem of synchronization. Repp (2005) studied synchronization on the task of tapping together, theorizing on two perceptual mechanisms that enable people to achieve *sensory-motor synchronization*: phase and period correction. Phase correction was applied by Wing et al. (2014) who propose a first-order linear phase correction model to predict synchronization in classical string quartets performances. Moore and Chen (2010) report on the modelling of the interactive behaviour of two members of a classical string quartet when performing musical notes in rapid succession, by recording the bow movements using angular velocity sensors. Goebl and Palmer (2009) study the effect of auditory feedback in ensemble performances of piano duets, in which *leader* and *follower* roles are assigned to each pianist. Expressive performance in string quartet ensembles is considered by Sundberg et al. (1989) following an *analysis by synthesis* approach. Raphael (2001b,a, 2003) report on a real-time accompaniment system able to play back following a soloist performance, using a Bayesian Belief Network to predict the soloist timing, and a Hidden Markov

Model (HMM) to relate the soloist and the accompaniment parts. Marchini (2014) studies ensemble expressive performance in classical quartets by building machine learning models based on audio and music content analysis, as well as motion caption systems, to predict the use of expressive parameters in order to obtain insights on the interactions among musicians.

2.3 Computational music processing related tasks

2.3.1 Melodic Transcription

Over the last decades there has been an increasing interest in the problem of automatic music transcription, which has proven to be a difficult task, as it requires both source separation and analysis of these sources (e.g. Ellis (1996)). According to Plumbley et al. (2002) automatic music transcription aims at identifying the instruments playing and the onset and duration of the notes played by each instrument to produce a written transcription of the piece (usually in the western musical notation). In this dissertation we will refer to music material based on its *monophonic/polyphonic* and *monotimbral/multitimbral* nature. Musical pieces are *monophonic/polyphonic* when only **one** instrument is playing *one/several* notes at a time. Similarly musical pieces are *monotimbral/multitimbral* when one/several instruments are playing at the same time.

Monophonic mono-timbral audio

Transcription from monophonic monotimbral audio has been usually performed in two consecutive steps: pitch tracking, and onset detection. Autocorrelation has been a widely used approach for fundamental pitch detection (*f0*) (De Cheveigné and Kawahara, 2002) and music transcription (Brown and Zhang, 1991), in which the cross correlation of a signal with itself as a function of the time lag between them is observed to find periodicities.

Monophonic multi-timbral audio

The most widely used methods for melodic extraction are the ones based on salience pitch calculation, e.g. Salamon and Gómez (2012). Usually these methods are performed in three steps. First, a spectral representation of the signal is computed using spectral analysis techniques (e.g. Fast Fourier Transform). Second, a salience function (time-frequency representation of pitch salience) is computed to obtain several *f0* candidates, usually using weighted harmonic summation methods. Finally, the melody peaks are selected based on tracking methods over frequency and time. Other methodologies use source separation based on timber models and grouping principles (Ozerov et al., 2007; Durrieu, 2010) , and some use stereo separation to estimate the panning of each source (Durrieu, 2010).

Some systems have been designed specifically for guitar transcription. Fiss and Kwasinski (2011) provide a real time polyphonic pitch detection method for transcribing the audio produced by a guitar into a tablature notation.

2.3.2 Ornament recognition

Automatic recognition and characterization of ornaments in music has been studied in the past, as part of the research in music expressive analysis. Perez et al. (2008) model mordents and triplets in Irish fiddle music with the aid of 3D motion sensors to capture bowing gestures and time-pitch curves analysis. Trills and appoggiaturas are modelled by Puiggròs et al. (2006) in bassoon recordings by automatically extracting timing and pitch information from the audio signal, and using machine learning techniques to induce an expressive performance model. Gómez et al. (2011) automatically detect ornaments in flamenco music (melismas) categorizing ornaments into six different types, and adapting the Smith-Waterman algorithm (Smith and Waterman, 1981) for sequence alignment. Casey and Crawford (2004) use the MPEG-7 standard audio descriptors to build a Hidden Markov Model classifier to automatically detect a subset of possible ornaments in 18th and 17th century lute music, based on the hypothesis that HMM state transitions occur at higher rates during ornaments than during non-ornamented segments of an audio signal.

2.4 Application fields overview

2.4.1 EEG based emotion detection

Emotion detection studies have explored methods using voice and facial expression information (Takahashi, 2004). Other approaches have used skin conductance, heart rate, and pupil dilation (Partala et.al, 2000)Partala et al. (2000). Different methods have been proposed to recognize emotions from EEG signals, (e.g. Chopin (2000); Takahashi (2004); Lin et al. (2010)), training classifiers and applying different machine learning techniques and methods. Ramirez and Vamvakuosis Ramirez and Vamvakousis (2012) propose a method based on mapping EEG activity into the bi-dimensional arousal/valence plane of emotions (Eerola and Vuoskoski, 2010). By measuring the alpha and beta activity on the prefrontal lobe, they obtain indicators for both arousal and valence. The computed values may be used to classify emotions such as happiness, anger, sadness and calm.

Other relevant approaches make use of fractal dimension for emotion recognition. Such is the case of Liu et al. (2010), which propose a real-time model based on Higuchi algorithm (Higuchi, 1988) to calculate Fractal dimension of EEG signal. Fractal dimension can be understood as a measure of the complexity of a signal. It has been previously used for the analysis of Electroencephalographic time series (Accardo et al., 1997). Liu model calculate fractal dimension from a band pass filtered EEG signal covering Alfa and Beta channels (2 Hz to 48 Hz) from FC6 electrode to measure Arousal. Valence calculation is performed from the difference of the fractal dimension values of electrodes AF3 and F4 based on the asymmetrical brain activation hypothesis.

2.4.2 Active music listening

Active music listening is a study field that aims to enable listeners to interactively control music. While most of the work in this area has focused on control music aspects such as playback, equalization, browsing and retrieval, there have been few attempts to controlling expressive aspects of music performance. Interactive performance systems have been developed in order to make possible for a listener to control music based on the conductor - orchestra paradigm. This is the case of the work of Fabiani (2011) who use gestures to control performance. Gesture parameters are mapped to performance parameters adapting the four levels of abstraction/complexity proposed by Camurri et al. (2000). This level of abstraction range from low-level parameters (physical level), such as audio signal, to high-level parameters (semantic descriptors), such as emotions. Thus, gesture analysis is done from low to high-level parameters, whereas synthesis is done from high to low level parameters. The control of mid and low level parameters of the performance is carried out using the KTH rule system by Friberg (2006).

Chapter 3

Data Acquisition

This chapter is devoted to present the music material used in this dissertation and the methodology used for data acquisition. The musical corpus used for this study consists of a selection of jazz standard pieces. Score data is obtained from the *lead sheets* available in the The real book. The corresponding performance data is obtained from both the recordings by professional guitarists and recordings extracted from commercial CDs. Score data is encoded in XMLmusic format, and performance data is extracted from recordings by proposing methodologies to improve automatic performance transcription. Firstly, we propose an approach to optimize pitch profile extraction from polyphonic signals (i.e. monophonic-multitimbral) using genetic algorithms. Secondly, we propose a pitch profile segmentation into note events by using adaptative threshold filters and two rule based filters, one based on minimum note duration and minimum gap duration, and another based on musical heuristics. Energy estimation from both monophonic-monotimbral and monophonic-multitimbral audio signals is mapped to note energy values.

3.1 Musical Corpus

The musical material consists of recordings of j*azz standards*, whose scores were available in the The real book. Two different strategies were used to obtain performance data. The first consisted of extracting musical data from monophonic audio recordings of an electric guitar, performed by 3 professional jazz guitarists. We will refer to this type of audio data as *monophonic-monotimbral* recordings. The other way was to obtain data from commercial audio recordings of a well known jazz guitarist (Grant Green). We will refer to this second type of audio data as *monophonic-multitimbral*. Each type of data source has its own advantages and disadvantages. The first source (monophonic-monotimbral) avoids the problem of voice separation, which is present in the second approach (monophonic-multitimbral). It also permits, not only to obtain a more accurate transcription of the musical events, but also to have a controlled setting for performance recording (e.g. indications on the type of expressivity, use of polyphony, or degree of deviation from the score). The second method validates the musical quality of the performance in terms of the expertise of the performer by obtaining the expressive musical data from recordings of a well known guitarists. Moreover, the scenario of extracting expressive information from commercial audio recordings, is a common practice used by jazz performers (and students) to learn and understand the expressive aspects within the music.

3.1.1 Music scores from Real book

The scores were extracted from the The real book, which is a collection of popular jazz tunes in the form of *lead sheets*, so called because they contain common aspects to most influential recorded performances, i.e. main melody, main chord progression, time signature and performance style (e.g. swing, beebop, ballad, etc.). The tunes were coded in MusicXML, an XML-based format for representing western music notation, which allows to store not only information about the notes (pitch, onset, and duration) but also other relevant information for note description such as chords, key, and tempo, among others. MusicXML format can be freely used under Public licence.

3.1.2 Monophonic-monotimbral recordings

The musical material corresponding to the monophonic-monotimbral data set, consisted of 27 jazz standard audio recordings (resulting in a total of 1383 notes) recorded by a professional jazz guitarist. The melodies selected belong to the most representative jazz repertoire, and they include different tempos, composers and jazz styles. As jazz genre comprises numerous substyles, to bound our study, some styles were not taken in consideration, such as Bee-bop tunes, because of its melodic complexity leave small room for ornamentation, as well as fusion jazz styles (e.g. bossanova, latin and funk grooves). Additional recordings of 16 selected pieces from the initial 27 jazz pieces were recorded by two more professional guitarists. This additional recording material was used for validation purposes of the models as will be explained later in Chapter 6.

The audio of the performed pieces was recorded from the raw signal of an electric guitar. The guitarist was instructed not to strum chords or play more than one note at a time. The guitarist recorded the pieces while playing along with pre-recorded commercial accompaniment backing tracks (Kennedy and Kernfeld, 2002). We opted to use audio backing tracks performed by professional musicians, as opposed to synthesized MIDI backing tracks, in order to provide a more natural and ecologically valid performance environment. However, using audio backing tracks required a preprocessing beat tracking task. Each piece's section was recorded once (i.e. no repetitions nor solos were recorded), For instance, a piece consisting in sections *AABB*, only sections *A* and *B* were considered. A list of the recorded pieces is presented in Table 3.1 along with the recorded tempo,

3.1.3 Monophonic-multimbral recordings

The music material considered as monophonic-multitimbral data consist of 16 commercial recordings of Grant Green, and their corresponding music scores (The real book). Table 3.2 shows the audio recordings considered. The instrumentation for most of the pieces consists of guitar (g), piano (p), double bass (b), and drums (d) (details can be found in the table). Green's particular style of playing uses a linear monophonic approach which facilitates voice separation of the main melody performed by the guitrar, from the acompaniment instruments. A total of 744 note events were extracted and manually corrected from the recordings. Details of melodic extraction will be explained in section 3.3)

Piece	Key	BPM	Aebersold Vol.	Form	Section analized
allOfMe	C	160	95	A_1, A_2	A_1
allTheThing.	Ab	132	43	A_1, B, A_2	A_1, B
aloneTog.	Dm	126	41	A_1, A_2, B, A_3	A_2, B
autumL.	Gm	174	20	$A_1, A_2, B,$	A_2, B
bodyAndS.	Db	56	93	A_1, A_2, B, A_3	A_2, B
byeByeBB.	F	138	39-65	A_1, B, A_2	A_1, B
daysOfWR.	F	88	40	A_1, A_2	A_1
equinox	Cm	100	57	A	A
footprints	Cm	176	33	A	A
four	Eb	132	65	A	A
haveYouM.	F	208	25	A_1, A_2, B, A_3	A_2, B
illRememb.	G	210	*	A_1, B_2, C_1, A_2, B_2	A_1, B_2, C_1
invitation	Cm	120	34	A_1, B, A_2	A_1, B
justFriends	F	120	59	A_1, B, A_2	A_1, B
ladyBird	C	152	70-99	A	A
likeSome.	C	195	23	A_1, A_2	A_1
lullabyOfB.	Eb	138	40	A_1, A_2, B, A_3	A_2, B
misty	Eb	66	41	A_1, A_2, B, A_3	A_2, B
myFunnyV.	Cm	78	25	A_1, B, A_2	A_1, B
outOfNo.	G	112	59	A_1, A_2	A_1
satinDoll	C	128	12	A_1, A_2, B, A_3	A_2, B
solar	Cm	160	88	A	A
stellaByS.	Bb	108	59	A_1, B, A_2	A_1, B
sweetGB.	Ab	234	39	A_1, A_2	A_1
takeTheA.	C	112	65	A_1, A_2, B, A_3	A_2, B
thereIsNoG.	Bb	196	94	A_1, B_1, A_2, B_2	A_1, B_1
thereWillN.	Eb	168	44	A_1, A_2	A_1

Table 3.1: Monophonic.monotimbral recordings list.

Album	Year	Instrumentation	Name	Author
Standars	1961	G. Green (g) W. Ware (b) A. Harewood (d)	All the things you are I'll remember April I Remember you Love walked in If I had you	J. Kern G. de Paul V. Schertzinger G. Gershwin Cambell & Connelly Connelly
Goodens Corner	1961	G. Green (g) S. Clark (p) S. Jones (b) L. Hayes (d)	On green dolphin street What is this thing called love	B. Kaper C. Porter
Nigeria	1962	G. Green (g) S. Clark (p) S. Jones (b) A. Blakey (d)	Airegin	S. Rollins
Green Street	1962	G. Green (g) B. Tucker (b) D. Bailey (d)	Alone together Moon river Round about midnight	A. Schwartz H. Mancini T. Monk
Born to be blue	1962	G. Green (g) S. Clark (p) S. Jones (b) L. Hayes (d)	If I should lose you My one and only love	R. Rainger G. Wood
Oleo	1962	G. Green (g) S. Clark (p) S. Jones (b) L. Hayes (d)	Tune up	M. Davies

Matador	1964	G. Green (g) McC. Tyner (p) B. Cranshaw (b) E. Jones (d)	My favorite things	Rogers & Hammerstein
I want to hold your hand	1965	G. Green (g) L. Young (o, b) E. Jones (d)	Speak low	K. Weill

Table 3.2: Monophonic.polytimbral recordings list.

3.2 Score Data Acquisition

In this section we explain how we obtain a machine readable representation from musical scores. Several computer music representation has been developed oriented to different applications (for an overview see Selfridge 1997). One of the most prevalent computer protocol for music representation is the MIDI (Musical Instrument Digital Interface) protocol established in the late 80's. However it is mainly based on protocols for hardware control, used for sound synthesis. In general, a computer music representation scheme might (at least) contain information about the pitch, onset, and duration each note. In this sense MIDI representation is limited in several situations. For example, MIDI is not able to handle information for note performance indications (e.g. dynamics, articulation, or ornamentation), nor chord information.

In general, a musical score is a representation of the essential information of notes: pitch, duration, and onset. For instance pitch is represented in the position of a musical figure over the lines or spaces of the staff, duration is represented by the shape of the note figure (e.g. Hole, half, quarter notes, quavers, semiquavers, etc.), and onset by the relation between the timing notation convention (*time signature* and *bars*) and the position of the musical figure within a *bar*. Other conventions are widely used in classical notation to indicate expressive performance parameter in a score. Indications refer to variations in dynamics (e.g. cressendo, piano, forte), as well to note articulation (e.g. legato and stacato), and ornamentation (e.g. grace note, trills, echape, etc.).

In classical music the score indicates how to interpret a musical composition, by means of the western musical notation system that include expressive performance indications. In contrast scores from the The real book (also called *fake books*) are transcriptions of the popular repertoire played by jazz musicians, intended to give a musician (or a group) the minimal information of a tune (melody and chords) so a band can fake an improvised performance/arrangement of the tune (if they do not already know it). Therefore, contrary to classical music, in jazz the performer is not intended to play an explicit performance of the piece as it is written on the score. There are two main differences between classical music scores and jazz music scores. Firstly, the aforementioned expressive performance indications, widely used in classical notation, are not usually present in the scores from the realbook, and is the performer who decided when and how introduce them based on his taste, background, knowledge and playing style. Secondly, in jazz (and popular music) the harmonic information is not explicitly written with its constituent notes on a staff, as is the case of classical music. Instead, several kinds of chord names and symbols are used above each melodic line. The performer decides the *texture* (e.g. inversions, tensions) for accompaniment. Moreover, ambiguous situations in the melody or chord information might be encountered between different *fake books* versions.

3.2.1 Scores to MusicXML Format

In previous section we emphasize on several aspects specific to jazz scores which serve as requirements to define de type of format needed for encoding. For this work we choose the musicXML format, an XML format designed for western musical notation, suitable for classic and popular music. Each score was re-written using an open source software for music notation (Froment et al., 2011), and then converted to MusicXML format containing onset, duration and tempo information, as well as contextual information (e.g. key, chords, mode). In each piece, *tempo* and *key* were adapted to match the recordings. Ambiguity in chord information in the scores was resolved as shown in Table 3.4.

Representation of notes

Throughout this dissertation pitch will be expressed rounded units of MIDI note numbers by

$$noteNumber_{MIDI} = round\left(69 + 12log_2\left(\frac{F_o}{440}\right)\right) \tag{3.1}$$

were F_0 is the fundamental frequency of the note. Midi note number and note names and octave is presented in Figure 3.3. Accordingly, *key*, *chroma* and *chord root* will be represented using the range of the first column of the Table 3.3 , as will be explained in the following sections.

Representation of chords

Chord information was represented by its root and its type (e.g. root: C, type: Maj7). Chord type was encoded using a description scheme based on *note degrees*. Note degrees refer to the relative position of a note with respect to its tonic (main note of the scale). In jazz music theory a similar approach is used for chord definition, in which the *chord note degrees* (notes conforming a particular chord) are expressed as the relative distance with

	Octave										
Note	-1	0	1	2	3	4	5	6	7	8	9
C	0	12	24	36	48	60	72	84	96	108	120
C#/Db	1	13	25	37	49	61	73	85	97	109	121
D	2	14	26	38	50	62	74	86	98	110	122
C#/Eb	3	15	27	39	51	63	75	87	99	111	123
E	4	16	28	40	52	64	76	88	100	112	124
F	5	17	29	41	53	65	77	89	101	113	125
F#/Gb	6	18	30	42	54	66	78	90	102	114	126
G	7	19	31	43	55	67	79	91	103	115	127
G#/Ab	8	20	32	44	56	68	80	92	104	116	-
A	9	21	33	45	57	69	81	93	105	117	-
A#/Bb	10	22	34	46	58	70	82	94	106	118	-
B	11	23	35	47	59	71	83	95	107	119	-

Table 3.3: MIDI note numbers

respect of the root of the chord. For example a *dominant seventh* chord is said to be conformed by the *root*, the *third*, the *fifth*, and the *flat seventh* (1,3,5,b7). In the Table 3.4 we define a chord topology, consistent with the representation of notes in terms of the MIDI number (Section 3.2.1)to facilitate interval operations (e.g. addition, substraction, distance) for descriptors calculation. In the table, *Chord note degrees* are represented using the range of the first column of Table 3.3, in which the root of the chord (fist degree) is indexed by zero, and the remaining are indexed accordingly. A total of 26 chord definitions are considered.

In some machine learning scenarios (e.g. regresion problems) this 26 labels for chord definitions would generate 26 features when data is converted from nominal to numerical (binary) which may generate overfitting problems. In this type of schemes we used instead a binary representation that reduces to 12 the number of numerical features needed to describe a chord. Thus, root is represented based on chroma information (in a range from 0 to 11), whereas chord degrees (within one octave) are represented by boolean variables. An example of this binary representation for a *G7b9b*13 is shown in Table 3.5

Representation of key

We opted for two representations of Key. First we used a linear representation in which the key center note is represented using a range from zero to eleven, according to the first column of Table 3.3. This representation is

Chord Type	Chord degrees
major	0 4 7
m (minor)	0 3 7
sus2	0 2 7
sus4	0 5 7
dim	0 3 6
aug	0 4 8
Maj7	0 4 7 11
6th	0 4 7 9
m7	0 3 7 10
m6	0 3 7 9
mMaj7	0 3 7 11
m7b5	0 3 6 10
dim7	0 3 6 9
7th	0 4 7 10
7sus4	0 5 7 10
7#5	0 4 8 10
7b5	0 4 6 10
7#11	0 4 6 7 10
Maj9	0 2 4 7 11
m9	0 2 3 7 11
6/9	0 2 4 7 9
m6/9	0 2 3 7 9
9th	0 2 4 7 10
7b9	0 1 4 7 10
7#9	0 3 4 7 10
13	0 2 4 7 9 10
7b9b13	0 1 4 7 8 10
7alt	0 1 3 4 6 8 10

Table 3.4: Chord description list.

	Chord degree											
	Root	*b*9	2	*b*3	3	4	#11	5	*b*13	6	*b*7	7
			9	#9		11	*b*5		#5	13		
*G*7*b*9*b*13	8	1	0	0	1	0	0	1	1	0	1	0

Table 3.5: Example of a binary representation of a *G*7*b*9*b*13 chord.

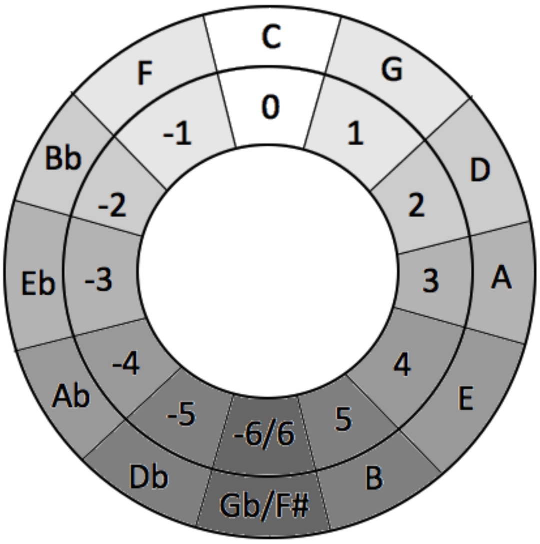


Figure 3.1: Numerical representation of the circle of fifths.

useful for interval operations for note description (e.g. distance between a note and the key). Second a representation using the *circle of fifths*. The circle of fifths is constructed by, starting at any pitch, ascending in intervals of fifths and descending in intervals of fourths. Distance between pitch classes in the circle of fifths (for notes, scales, key and chords) are associated to consonance or dissonance. High and low distance are correlated to high and low tension respectively (Lerdahl, 1996). Numerical representation of pitch clases in the circle of fifths is presented in Figure 3.1.

3.2.2 MusicXML Parser

From the score representation in musicXML format, we implemented a routine to parse the xml file and obtain a note data representation in the matlab environment. Note representation was obtained in two formats. First we followed the representation used in the Miditoolbox by Eerola and Toiviainen (2004), in which notes are represented in a note matrix. Table 3.6, shows an example of the notes events of the first four bars of the piece *All of me*. The motivation for using this representation was the possibility of using the tools provided music analysis in the *Midi Toolbox* as note descriptors

Onset (beat)	Duration (beat)	channel	Pitch (MIDI #)	Energy (MIDI #)	Onset (sec.)	Duration (sec.)
1,00	1,00	1,00	72,00	80,00	0,00	0,38
2,00	0,50	1,00	67,00	80,00	0,38	0,19
2,50	4,50	1,00	64,00	80,00	0,56	1,69
7,00	0,67	1,00	72,00	80,00	2,25	0,25
7,67	0,67	1,00	74,00	80,00	2,50	0,25
8,33	0,67	1,00	72,00	80,00	2,75	0,25

Table 3.6: Note events matrix obtained by the XML parser in the Miditoolbox format.

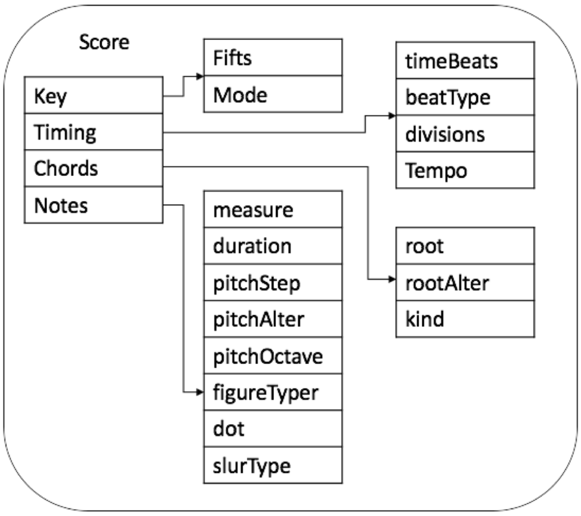


Figure 3.2: Score data structure.

(as will be explained in section 4.1). Second we obtain representation of each score on a data structure, which include note information (e.g. duration, pitch, etc.), as well as score context information (tempo, key, chords), as depicted in Figure 3.2. This type of encoding permits to handle data information for note description calculation in a more efficient way.

3.3 Performance Data Acquisition

In this section we will describe our approach to obtain a machine readable representation of the performance from the aforementioned two type

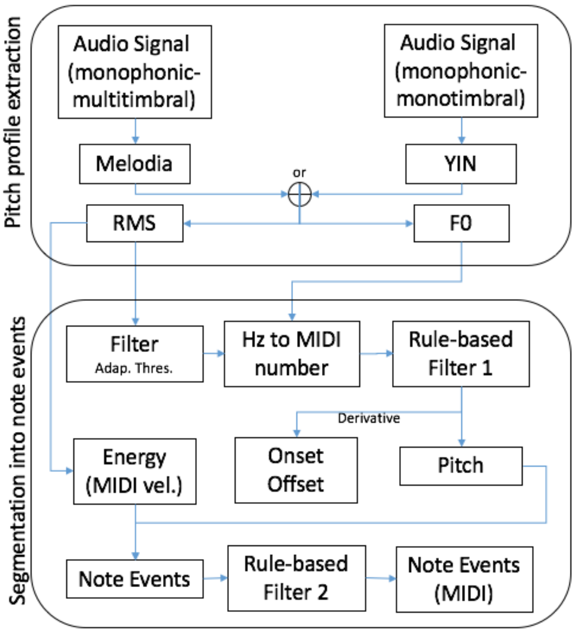


Figure 3.3: Automatic transcription of performance data.

of recordings: monophonic-monotimbral and monophonic-multitimbral. A similar approach was used for both types of recordings to obtain the data and is depicted in Figure 3.3, which differs only in the pitch profile extraction. A first stage consists on obtaining from recorded audio, a representation of the performed melody by its *pitch profile*. Pitch profile is defined by (Goto, 2004) as a sequence of fundamental frequency (F0) values corresponding to the perceived pitch of the main melody. In parallel we obtain a fame based energy estimation for the main melody. A second stage consists in the pitch profile *segmentation into note events*. As mentioned in Section 3.2.1, note events should include information about the onset, duration and energy of the note. Finally, this information is stored in MIDI format to later performance analysis 4.2.

3.3.1 Monophonic mono-timbral pitch profile extraction

Monophonic-monotimbral pitch profile extraction was performed using YIN algorithm (De Cheveigné and Kawahara, 2002), which performs a frame based estimation of the the *fundamental frequency* (F0) using autocorrelation methods. Yin algorithm was used with a hop size of $2.9ms.$, and a window size of $46, 44ms.$. Minimum and maximum frequencies were set accordingly to the guitar *tessitura*. We assume a comfortable (acceptable) range between notes A2 (110Hz) and D6 (1175 Hz) for playing a melody on a electric guitar in a jazz context.

Energy estimation

In the case of monophonic-monotimbral signal the energy estimation was done calculating the *root mean square* (RMS), using the same windowing scheme over the recorded melody signal as explained in Section 3.3.1.

3.3.2 Monophonic multi-timbral pitch profile extraction

For monophonic multi-timbral pitch contour extraction we used an optimized approach of the salience pitch method used by Salamon and Gómez (2012). However, the default parameters that the algorithm uses, were tuned using a grid search method over different data bases in which melody was performed mainly by singing voice, therefore, these default parameters fail to extract the melody for some specific instrument settings (jazz guitar recordings in our case). To overcome this issue, we proposed a method to optimize the algorithm's main parameters using genetic algorithms (Giraldo and Ramírez, 2014), to obtain optimal combination of parameter values swited for jazz guitar melodic extraction.

Melodia parameters for optimization

In *Melodia* approach melodic extraction is performed in several steps. First, in the *sinusoidal extraction* stage, the signal is filtered to enhance the most audible frequencies of human hearing range. Then a *Short Fourier Transform* is calculated using a *Hanning* window of 46.4ms with a hop size of 2.9ms and a 4 zero padding factor. Finally, the peaks are corrected using instantaneous frequency method (Paiva et al., 2006). Second, the *salience function* computation is performed based on the summation of the weighted energy of the harmonic peaks of a given frequency in order to obtain the f0 candidates. The number of harmonics considered and the weighting scheme is an important factor that affects the salience computation. Third, at the *pitch contour* computation stage, the peaks detected in the previous step are grouped into pitch contours, based on several thresholds defined per frame basis, as well as per time continuity basis. A fourth stage is *contour characterization* in which the melody contour is chosen among the contours created in the previous step. In this stage, a set of features that guide the system to select the main melody is implemented. These features include pitch deviation, pitch trajectory, presence of vibrato, as well as contour pitch, length and salience. The fifth and final stage consists on the *melody selection* which is performed in four main steps: voicing detection, octave error minimization, pitch outliers removal, and final melody selection.

The following algorithm's parameters, involved on previous calculations and subjected for optimization, were the ones found to be more sensitive in the medic extraction accuracy, based on initial testes:

- Peak Distribution Threshold: Allowed deviation below the peak salience mean over all frames (fraction of the standard deviation)
- Peak Frame Threshold: Per-frame salience threshold factor (fraction of the highest peak salience in a frame)
- Pitch Continuity: Pitch continuity cue(maximimal lower pitch change during 1ms time period)
- Time Continuity: Time continuity cue (the maximum allowed gap duration for a pitch contour)

Using genetic algorithms for parameter optimization

Genetic Algorithms (GA) are stochastic optimization algorithms which imitate the biological mechanisms of natural selection. GA are widely used in several optimization problems involving discontinuous, noisy, high- dimensional, and multi-modal objective functions. In contrast to other optimization algorithms, the search that GA's perform is done in a more global context, whereas others (e.g. gradient descent) perform search in a more local context. Genetic algorithms work as follows: First, an initial random population of individuals is created. This initial population serves as seed to generate future generations by means of combining different parents (crossover) and randomly modifying a single individual (mutation). The algorithm iterates, and in each iteration it selects the best subjects based on a fitness function. The next generation of individuals is generated by applying mutation and crossover. The individuals with best fitness values are preserved in the next generation, while the others are discarded. The process iterates until a maximum number of iterations is reached or the fitness relative value of the best individual does not change more than a tolerance value during several generations. The most popular applications of GA in a musical context have been done for music composition (Koga et al., 2013; Matić, 2013). Other approaches have used GA's for automatic music transcription (Reis et al., 2008), music segmentation (Rafael et al., 2013), and interactive music applications (Hung and Chang, 2011), among many others.

We used a ground truth generated with the monophonic monotimbral recordings to optimize the parameters. The monophonic-multitimbral recordings were used for testing. We manipulate the implementation of Melodia found in Essentia library (Bogdanov et al., 2013) to set the selected parameters as input variables. The initial population was setted using Melodia's default parameters, and maximum and minimum parameter values were set (according to the values reported) as,

$$\begin{aligned} Highbound &= [1, 1, 30, 150], \\ Initialpoint &= [0.9, 0.9, 27.56, 100], \\ Lowbound &= [0.5, 0.5, 10, 50], \end{aligned}$$

in which each value on each row vector corresponds to *peak distribution threshold*, *peak frame threshold*, *pitch continuity*, and *time continuity* respectively. Initial population was set randomly to 20 individuals (i.e. 20 vectors with different parameter values combinations). The stopping criteria used in this study was set to a maximum of 500 iterations, and a relative threshold change in fitness function of $1x10^{-6}$. Crossover factor was set to 0.8 and mutation factor was set to 0.02, which are typical settings for these values.

A fitness function was implemented in which the cost of the fitness function is calculated based on the overall accuracy measure, defined as the total proportion of frames correctly estimated by the algorithm compared to the ground truth (as reported by *melodia* authors). This proportion is measured based on the *true negatives* (TN) and *true positives* (TP) for which the pitch estimation is correct (between a range of +/- 1/4 of tone of the ground truth). TN and TP were calculated as follows,

$$TP = \sum_{n=1}^{N} 1200 * log_2 \left(\frac{F0_{mel}(n)}{F0_{gt}(n)} \right) > 50cents \tag{3.2}$$

$$TP + TN = \sum_{n=1}^{N} 1200 * log_2 \left(\frac{F0_{mel}(n)}{F0_{gt}(n)} \right) \tag{3.3}$$

where $F0_{mel}(n)$ and $F0_{gt}(n)$ corresponds to the frequency value obtained for the n^th window frame of the input signal (from a total of N frames in which the signal is windowed) by the *melodia* algorithm and the obtained as *ground truth* respectively. Thus the cost functions was calculated as,

$$j = 1 - \left(\frac{TP}{TP + TN}\right) \tag{3.4}$$

Different experiment configurations were set, for example, different audio mixes were created in which the melody (which was recorded on a separate channel) is at different sound levels with respect to the accompaniment track. Details on the experiments settings, parameter values obtained, and improvement measures can be found in Chapter 5

Energy estimation

The *Essentia* implementation of *Melodia* was modified to obtain an estimation of the energy of the main melody on a frame basis. *Melodia* calculates a confidence factor based on the average energy of the retrieved melodic segments (best candidates of main melody). This part of the implementation was modified for the algorithm to (additionally) output the energy value of each frame, rather than the average of the melody segment. This way we could obtain a frame by frame estimation of the main melody energy.

3.3.3 Segmentation of pitch profile

In this section we describe our approach to segment the obtained pitch profile into a set of discrete note events, i.e. grouping pitch frames in to notes defined by its pitch (in MIDI number), duration (seconds), onset (seconds), and energy (in MIDI velocity number). *Essentia* implementation for pitch contour segmentation uses the approach by Mcnab et al. (1996) which is based on energy and pitch segmentation. It uses an *island building* strategy to deal with *gross* errors and a fixed minimum and maximum energy threshold to find the notes' start and end boundaries. This approach is targeted for singing voice, in which examples were performed singing the syllables "*da*" or "*ta*", which permits to distinguish repeated notes with the same pitch based on amplitude. However, a fixed threshold might not be that successful in signals with significant loudness changes. In the case of guitar recordings, a fixed threshold would miss onsets (and offsets), specially when consecutive notes of the same pitch are played, in which the the energy drop is less accentuated. Therefore, we followed a similar approach (based on pitch and energy segmentation) but introducing several improvements. Firstly, for segmentation based on amplitude, we created an energy filter based on an adaptive threshold scheme. We explore two types of filter: a *median filter* to down-sample the envelope curve from the audio signal, and adaptative threshold filter using a *low pass filter* with *Hanning windowing* as a *detection function* (see Bello et al. (2005)), to remove peaks from the obtained pitch profile. This adaptative threshold might improve onset and offset detection as the threshold will increase in regions in which more note energy is present (e.g. consecutive notes). Secondly, for segmentation based in pitch, a rule based filter based on pitch transients was used. A set of rules based on maximum an minimum duration thresholds were defined and are explained in Section 3.3.3. Finally, a third filter was performed based on the note events obtained, in which duration and energy of the neighbour notes is used.

Adaptative energy threshold filtering

In Figure 3.4, a representation of the wave form (a) and the extracted pitch profile (b) is shown. In order to filter errors due to note transitions, or noise, a filter based on adapative energy threshold was implemented, following the approach described in Bello et al. (2005) for onset detection.

First an envelope curve (Figure 3.4c, blue) was obtained from the audio signal using the approach by Zölzer (2008) (implemented in *essentia*). The obtained envelop was down-sampled using a median filter, for which we employed the same hop-size (3ms) and frame-size (46.4ms) used for pitch detection. Thus, for each segment frame in the envelope we calculate the *first third median* value as follows:

$$envelope_{downSample}[i] = sort(envelope_{frame}[round(frameSize/3)]) \quad (3.5)$$

The down-sampled signal is depicted in Figure 3.4c (blue), which is then filtered using a *low pass filter*. For doing this, the signal is windowed (again) using a *Hanning* window of length 290ms (W = 100 Frames). Each value of the filtered signal is obtained as follows:

$$envelope_{filtered}[i] = ro + \lambda * \frac{1}{W} * \sum_{n=-W/2}^{W/2} hann(frame[n]) * frame[n] \quad (3.6)$$

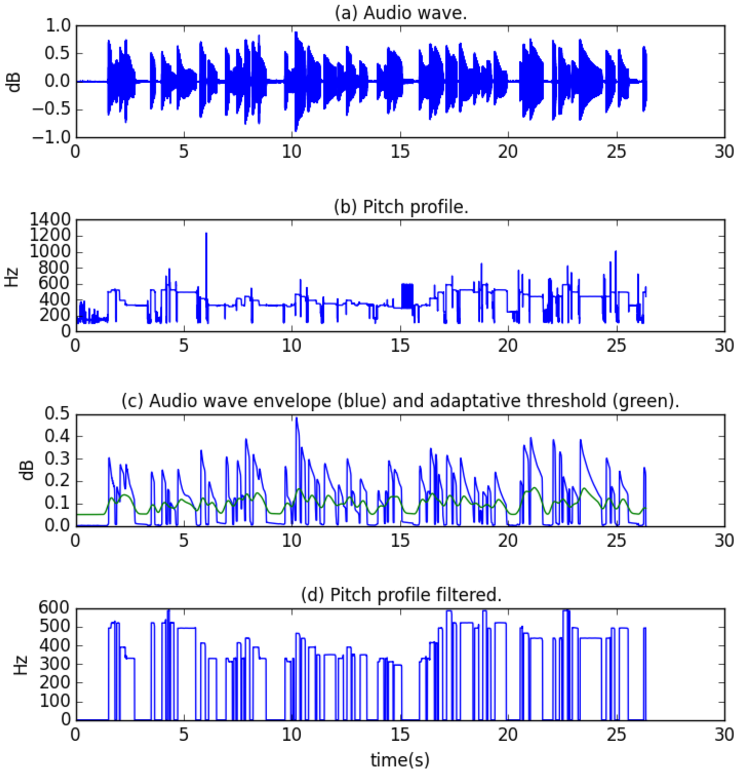


Figure 3.4: Pitch profile segmentation based on energy and pitch filters.

where W is the window length, and ro and λ are positive constants to increase/decrease *position* (over the energy axis) and *range* (max-min width) of the filter, respectively. Values for ro and λ were set as 0.05 and 0.8. The values used for ro, λ, and W were empirically found by experimentation over the data (an optimization approach to improve the selection of these parameters will be considered for future work). The obtained adaptative threshold curve is is shown in Figure 3.4 (green). Finally, the pitch profile values for which the down-sampled energy envelope is higher than the adaptative threshold obtained were kept, whereas the ones below were set to zero. The effect of applying the adaptative energy filter on the pitch profile is depicted in Figure 3.4d.

Rule Based Filtering

The first filter is aimed to remove short notes (20-40ms) and silent gaps (3-5ms). Short notes were defined to have an $f0 > 55Hz$ and a duration shorter than $w = 30ms$. Gaps were defined as notes with $f0 = 0$ and duration equal to five frames (3.6ms). Thus, short notes were filtered considering the following three cases (Gaps were always filtered using case 1):

- Case 1: The short note is a peak or a gap in the middle of a long note (previous interval equals the next interval in opposite direction). In this case the peak (or gap) is removed.
- Case 2: If the note is right in the middle of an ascending or descending interval. In this case the first half of the short note is assigned to the previous note and the second half of the note is assigned to the second note.
- Case 3: The short note is in an ascending or descending interval. In this case the short note is assigned to the closest long note (a note longer than 30ms).

Finding onsets and offsets

We designed a detection function based on the signal pitch changes. By differentiating the pitch profile, positive changes in pitch were set as onsets and negative were set as offsets.

Note energy mapping

The down-sampled signal envelope was also used to compute MIDI velocity, in which the mean energy of the frames composing a note was linearly mapped to a number between 60 to 100.

Rule based post filtering

We assumed that short notes (between 30ms and 100ms), and with low energy (compared to their respective neighbour notes) are prone to be errors. We calculated the product of duration and energy for each short note and the mean product of four neighbour notes (two consecutive previous and next notes). Notes with a product of less than 10 compared to the product of the neighbour notes were considered as errors.

Chapter **4**

Data analysis

In this chapter we explain the process for score data analysis and performance data analysis. Firstly, high-level features are extracted from the scores which include *nominal descriptors* that include information about the note about the note itself (e.g pitch, duration, onset, etc.), *local context descriptors* that include information about the neighbouring notes (e.g. previous/next duration, prev/netxt interval, etc.), *global context descriptors* which include information about the musical context in which the note occurs (e.g. key, tempo, current chord, etc) and finally, *perceptual descriptors* which include information calculated based on cognitive models. Secondly, performance data is analysed by comparing the similarity between the score and the performance by means of a Dynamic Time Warping approach, from which a set of performance actions are obtained semi automatically, which include timing/energy deviations, as well as, ornamentations. Finally, we explain how we construct a performance data base in which each note is characterized by the extracted descriptors along with the corresponding calculated performance action, which might include ornamentation.

4.1 Score analysis: feature extraction

In this section we explain the process to obtain note descriptors from the score data representation explained in previous sections. Feature extraction was performed following an approach similar to (Giraldo, 2012), in which each note was characterized by its *nominal* properties, its *Local* and *Global* context, and by a categorization based on *perceptual* models of music perception and cognition. We implemented our own feature extraction library for computing all the reported features, with the exception of the *perceptual features* for which we used the methods provided by the *miditoolbox* (Eerola and Toiviainen, 2004). We implemented the method by Grachten (2006) to parse the melodies and obtain for each note the label of the I-R *Narmour structure* (Narmour, 1992) to which it belongs. The concept of closure was based on metrical position and duration. The basic Narmour structures (P, D, R, and ID) and their derivatives (VR, IR, VP, and IP) are represented in Figure 4.1.The complete list of the 30 descriptors used for this study and its definition is summarized in Table 4.2.

Nominal descriptors

Nominal descriptors refer to the intrinsic properties of score notes (e.g. pitch, duration, and onset). Duration and onsets were described both in beats and seconds, as the duration in seconds depends on the tempo of the piece. For example the choice of ornamenting two different notes from different pieces with quarter note duration (beats) may differ if the pieces are played at slow and fast tempos. The energy descriptor refers to the loudness of the note, which in MIDI format is measured as velocity (how fast a piano key was pressed).

Local context descriptors

Given a particular note, its *Local context* descriptors refer to the properties of its neighboring notes, e.g. previous/next interval, previous/next duration ratio, previous/next inter-onset interval. In this work, only one previous and one following note were considered. Inter-onset distance (Giraldo and Ramírez, 2015a) refers to the onset difference between two consecutive notes.

Global context descriptors

Global context descriptors refer to the musical *context* in which the note occurs, e.g. tempo, chord, and key. The phrase descriptor (Giraldo and Ramírez, 2015a) refers to the note position within a phrase: initial, middle, or end. Phrase descriptors were obtained using the melodic segmentation approach by Cambouropoulos (1997), which indicates the probability of each note being at a phrase boundary. Probability values were used to decide if the note was a *boundary note*, annotated as either *initial (i)* or *ending (e)*. Non boundary notes were annotated as *middle (m)*. The phrase descriptor was introduced based on the hypothesis that boundary notes (i.e. initial or ending phrase notes) are more prone to be ornamented than middle

Time Signature	Very strong	Strong	Weak	Very Weak
4/4	Beat 1	Beat 3	Beats 2 and 4	other
3/4	Beat 1	none	Beats 2 and 3	other
6/8	Beat 1	Beat 2.5	Beats 1.5, 2, 3, and 3.5	other

Table 4.1: Metrical strength categorization.

notes. *Note to key* and *note to chord* descriptors are intended to capture harmonic analysis information, as they refer to the interval of a particular note with respect to the key and to the chord root, respectively. *Key* and *mode* refers to the key signature of the song (e.g. key: C, mode: Major). Mode is a binary descriptor (major or minor), whereas, key is represented numerically in the circle of fifths (e.g. $Bb = -1, C = 0, F = 1$ etc). However for some calculations (e.g note to key in Table 4.2) a linear representation of the notes (e.g. $C = 0$, $C\#/Db = 1$, $D = 2$, etc) is used for key. Also, it is worth noticing that the key descriptor may have 13 possible values as the extreme values (-6 and 6) correspond to enharmonic tonalities (Gb and $F\#$). The descriptor *Is chord note* was calculated using the chord type description of Table 3.4 in which each of the notes of the chord are shown using the aforementioned linear note representation. If a note corresponds to any of the notes included in the chord type description it is labelled as *yes*.

The *metrical strength* concept refers to the rhythmic position of the note inside the bar (Cooper and Meyer, 1963). Four levels of metrical strength were used to label notes in three common time signatures, depending on the beat at which the note occurs, as shown in Table 4.1.

Perceptual descriptors

Perceptual descriptors are inspired by music perception cognition models. Narmour's implication-realization model (Narmour, 1992) proposes eight basic melodic structures based intervallic expectation in melodies. The basic Narmour structures (P, D, R, and ID) and their derivatives (VR, IR, VP, and IP) are represented in Figure 4.1. Symbols refer to prospective or restrospective (shown in parenthesis in the Range column of Table 4.2) realization. Schellenberg (1997) simplified and quantified Narmour's model into 5 principles: registral direction, intervallic difference, registral return, proximity, and closure. *Tonal stability* Krumhansl and Kessler (1982) represents the degree of belonging to the (local) key context. *Melodic attraction* Lerdahl (1996) measures the *weight* (*anchoring strength*) of the pitches across the

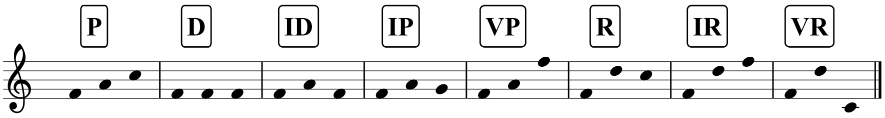


Figure 4.1: basic Narmour structures P, D, R, and ID, and their derivatives VR, IR, VP, and IP

pitch space. *Tessitura* and *mobility* are measures proposed by Von Hippel (2000). *Tessitura* is the standard deviation of the pitch height distribution and predicts the listener expectation of the tones being close to the median pitch. *Mobility* is based on the intuition that a melody is constrained to its tessitura and therefore melodies change direction after long intervals otherwise they will fall outside their comfortable range. This measure is calculated using one lag autocorrelation between consecutive pitches.

	Descriptor	Abbreviation	Units	Formula	Range
Nominal	Duration	ds_n	Seconds	ds_0	$[0, +\infty]$
	Duration	db_n	Beats	db_0	$[0, +\infty]$
	Onset	ons_n	Seconds	os_0	$[0, +\infty]$
	Onset	onb_n	Beats	ob_0	$[0, +\infty]$
	Onset in Bar	obm_n	Beats	$ob_0\%bpb$	$[0, +bpb]$
	Pitch	p_n	Semitones	p_0	$[1, 127]$
	Chroma	ch_n	Semitones	$p_0\%12$	$[0, 11]$
	Energy	v_n	MIDI vel	v_0	$[1, 127]$
Neighbour	Prev. duration	pds_n	Seconds	ds_{-1}	$[0, +\infty]$
	Prev. duration	pdb_n	Beats	db_{-1}	$[0, +\infty]$
	Next duration	nds_n	Seconds	ds_1	$[0, +\infty]$
	Next duration	ndb_n	Beats	db_1	$[0, +\infty]$
	Prev. interval	$pint_n$	Semitones	$p_{-1} - p_0$	$[-60, 60]$
	Next interval	$nint_n$	Semitones	$p_1 - p_0$	$[-60, 60]$
	Prev. io dist.	$piod_n$	Seconds	$os_0 - os_{-1}$	$[0, +\infty]$
	Next. io dist.	$piod_n$	Seconds	$os_1 - os_0$	$[0, +\infty]$
Context	Measure	m_n	Bars	m_0	$[0, +\infty]$
	Tempo	t_n	Bpm	t_0	$[30, 260]$
	Key	k_n	Semitones	k_0	$[-6, 6]$
	Mode	mod_n	Label	mod_0	$\{major, minor\}$
	Note to Key	$n2k_n$	Semitones	$ch_0 - k_0(linear)$	$[0, 11]$
	Chord root	chr_n	Semitones	chr_0	$[0, 11]$
	Chord type	cht_n	Label	cht_0	$\{$*+, 6, 7, 7#11, 7#5, 7#9, 7alt* $7b5.7b9, Maj7, dim, dim7,$ $m, m6, m7, m7b5, major\}$

	Note to chord	$n2ch_n$	Semitones	$ch_0 - chr_0$	$[0, 11]$
	Is chord note	$ichn_n$	Boolean	$isChNote(chr_0, cht_0, ch_0)$	$\{true, false\}$
	Met. Strength	mtr_i	Label	$metStr_0$	$\{Verystrong, Strong, Weak, Veryweak\}$
	Phrase	ph_n	Label	$phrase_0$	$\{initial, middle, final\}$
Perceptual	Narmour I-R	$nar1_n$	Label	$nar(p_{-1}, p_0, p_1)$	$\{P, D, R, ID, (P), (D), (R), (ID), VR, IR, VP, IP, (VR), (IR), (VP), (IP), dyadic, monadic\}$
		$nar2_n$		$nar(p_{-2}, p_{-1}, p_0)$	
		$nar3_n$		$nar(p_0, p_1, p_2)$	
	Regist. Dir.	$narRD$	int	$narRegDir()$	$\{0, 1\}$
	Inter. Diff.	$narID$	int	$narIntDiff()$	$\{0, 1\}$
	Regist. Ret.	$narRR$	int	$narRegRet()$	$\{0, 1, 2, 3\}$
	Proximity	$narP$	int	$narProx()$	$\{0, 1, 2, 3, 4, 5, 6\}$
	Closure	$narClos$	int	$narClos()$	$\{0, 1, 2\}$
	Consonance	$cons$	int	$consonance()$	$\{0, 10\}$
	Tonal stability	$tonal$	int	$tonality$	$\{0, 10\}$
	Mel. Attract.	$melAt$	%	$melattraction()$	$\{0, 1\}$
	Tessitura	$tessit$	semitones	$tessitura()$	$\{0, \inf\}$
	Mobility	mob	%	$mobility$	$\{0, 1\}$

Table 4.2: Features extracted from music scores. In the fifth row, in column Formula, *bpb* means *beats per bar*

4.2 Performance Analysis

Appoggiaturas, trills, mordents, turns, etc. are achetypical ornaments used in classical music to categorize ornaments. However this approach does not always apply in jazz music, as melodic embellishment in jazz lays in between this archetypical ornamentation and free improvisation. The context in which a musician may use ornaments is usually learnt by copying the playing style of other professional musicians. Furthermore, in popular music, ornaments depend widely on the musician background, taste and current intention, and they are used based on melodic, harmonic and rhythmic context. In the case of jazz music, the performance of a piece usually include the addition of different types of ornaments, such as passing notes, neighbor notes and chord scale notes. Typically, these ornaments may include short musical phrases (also called licks), often used as a preparation for a target note, or to replace long notes.

Thus, in jazz music the performance of a melody is not expected to be an explicit render of the written music, due to the improvisational nature of melodic ornamentation. Our aim is to automatically obtain for each performed note (or group of notes) its corresponding parent note in the score, as depicted in Figure 1.2.

In this section we will explain our approach for expressive performance analysis which consists on the categorization and measurement of the musical devices and deviations from the score that a musician introduce when performing a musical piece. Throughout this study we will refer to these deviations as *performance actions* (PAs).

4.2.1 Performance to score alignment

Score to performance alignment was performed to correlate each performed note with its respective *parent* note in the score as depicted in Figure 1.2. This procedure was carried out following the approach of (Giraldo and Ramírez, 2015b), in which *Dynamic Time Warping* (DTW) techniques were used to match performance and score note sequences. A similarity cost function was designed based on pitch, duration, onset, and phrase onset/offset deviations.

Distance Cost Function

Phrase onset and offset deviation were introduced to force the algorithm to map all the notes of particular short ornament phrase (*lick*) to one parent note in the score. We assumed that a group of notes conforming a *lick* are played *legato*. Therefore, the performed sequence is segmented in phrases, in which the time gap between consecutive notes is less than 50 ms. This threshold was chosen based on human time perception studies (Woodrow, 1951).

Each note from the score and the corresponding performed sequence is represented by a five position *cost vector* as

$$cs = (p(i), ds(i), ons(i), ons(i), ofs(i)) \tag{4.1}$$

and

$$cp = (p(j), ds(j), ons(j), ph_{ons}(j), ph_{ofs}(j)) \tag{4.2}$$

respectively, where cs is the score *cost vector* and cp us the performance *cost vector*. Index i refers to a note position in the score sequence, and j refers to a note position at the performed sequence. The onset of the first note of the lick phrase in which the j^{th} note of the performance sequence occurs is represented by $ph_{ons}(j)$. Similarly $ph_{ofs}(j)$ refers to the offset of the last note of the lick phrase in which the j^{th} note of the performance sequence occurs.

The total cost is calculated using the *Euclidean distance* as follows.

$$cost(i,j) = \sqrt{\sum_{n=1}^{5} (cs(n)_i - cp(n)_j)^2} \tag{4.3}$$

Notice that in equation (6.2) phrase onset and offset deviations are calculated when n equals four and five.

Dynamic Time Wrapping Approach

We apply dynamic time warping (DTW): a similarity matrix $H_{(m \times n)}$ is defined in which m is the length of the performed sequence of notes and n is the length of the sequence of score notes. Each cell of the matrix H is calculated as follows.

$$H_{i,j} = cost + min(H_{i-1,j}, H_{i,j-1}, H_{i-1,j-1}) \tag{4.4}$$

where min is a function that returns the minimum value of the preceding cells (up, left, and up-left diagonal). The matrix H is indexed by the note position of the score sequence and the note position of the performance sequence.

A backtrack path is obtained by finding the lowest cost calculated in the similarity matrix. Starting from the last score/performance note cell, the cell with the minimum cost at positions, $H_{(i-1)}$, $H_{(i,j-1)}$, and $H_{(i-1,j-1)}$ is stored in a backtrack path array. The process iterates until indexes arrive to the first position of the matrix, assigning each note in the performance to a parent note in the score.

Figure 4.2 presents an example of the resulting similarity matrix obtained for one of the recorded songs. The x-axis corresponds to the sequence of notes of the score and the y-axis corresponds to the sequence of performed notes. The cost of correspondence between all possible pair of notes is depicted darker for the highest cost (less similar) and lighter for the lowest cost (most similar). The dots on the graph show the backtrack path (or optimal path) found for alignment. Diagonal lines represent notes which were *not ornamented*, as the correspondence from the performance notes to the parent score notes is one to one. On the contrary vertical lines represent notes which were *ornamented*, as two or more performed notes correspond to one parent note in the score. Similarity horizontal lines represent *consolidation* (i.e. two or more notes are consolidated into one note). Blank cells on the horizontal context represent score notes being omitted (*deletion*) as well as blank cells on the vertical context represent a note *addition* (i.e a note added by the performer which does not have an specific correspondence to any note in the score).

Because there are not concrete rules to map performance notes to *parent* score notes, our alignment algorithm was evaluated by comparing its output with the level of agreement between five human experts who were asked to manually align performance and score note sequences. Accuracy of the system was estimated by quantifying how much each note pair produced by the algorithm agreed with the human experts, using penalty factors for high, medium, and low agreement. The results of the evaluation showed that the performance of our approach was comparable with that of the human annotators. Details of these evaluation will be explained in 5.

4.2.2 Expressive performance actions calculation

Previously we defined PAs in Section 4.2 as a set of musical resources used by musicians to add expression when performing a musical piece. In this section we explain how we categorize these PAs based on the type of align-

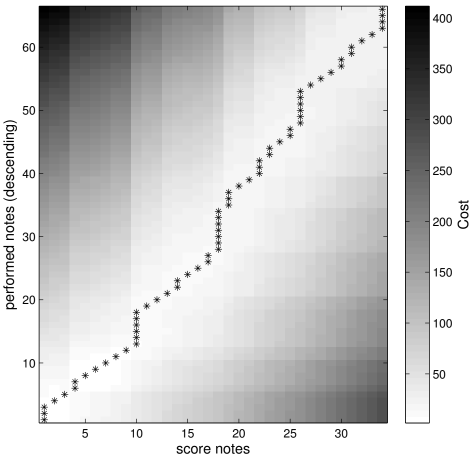


Figure 4.2: Similarity matrix of performed notes and score notes. Dots indicate alignment path between score and performance notes.

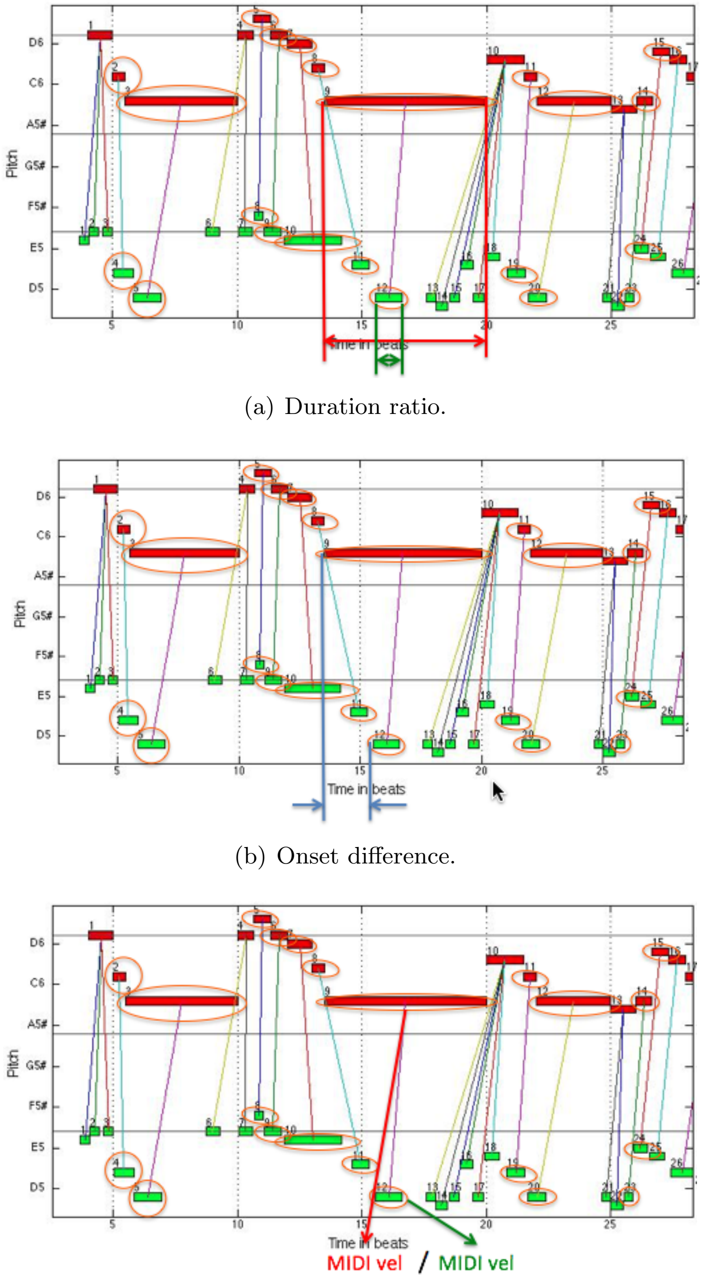
ment obtained in Section 4.2.1, following a similar approach as Grachten (2006). Concretely, we will consider PAs for two main classes out of the note alignment. The first class is the *non ornamented* (Figure 4.3), which represents the notes that have aforementioned *one to one* correspondence. For this non-ornamented notes, we will calculate three different PAs based on deviations in *onset*, *duration*, and *energy* with respect of the score.

The second class corresponds to *ornamented* notes (Figure 4.4). We will show in Chapter 6 that ornamented notes are a minority class (30% of the examples). Therefore, for this study we will classify as *ornamented* any PA consisting on the *addition*, *deletion*, *consolidation*, or fragmentation of notes.

Performance actions were calculated for each score note, as defined in Table 4.3, by measuring the deviations in onset, energy and duration. Again, indexes i and j refer to the note position at the score and the performance sequence, respectively.

4.2.3 Database Construction

The data collected was organized, storing each note descriptors along with its corresponding *performance action*. The *pitch*, *duration*, *onset*, and *en-*

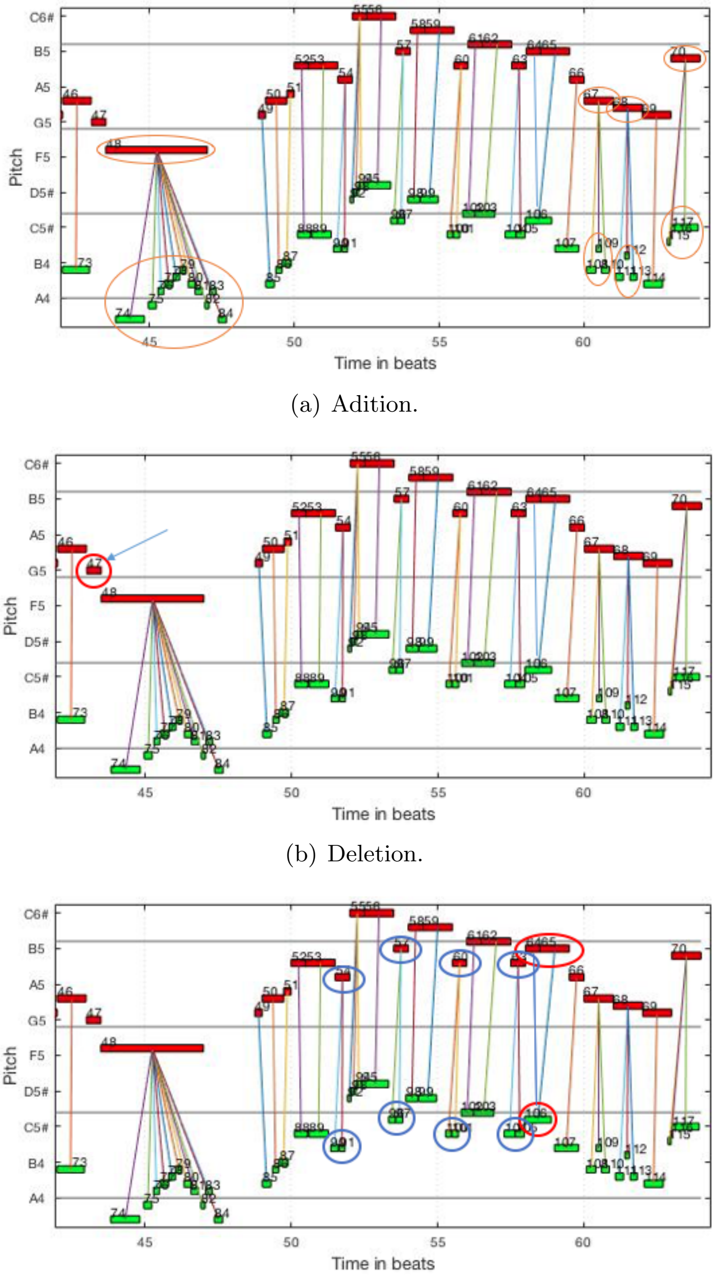


(a) Duration ratio.

(b) Onset difference.

(c) Energy ratio.

Figure 4.3: Expressive performance actions calculated for *not-ornamented* class. Red and green boxes corresponds to score and peformed notes respectively. Vertical lines indicate performance to score alingment. Notes linked with one line correspond to *not ornamented* class



(a) Adition.

(b) Deletion.

(c) Consolidation (red) and fragmentation (blue).

Figure 4.4: Expressive performance actions labeled as *ornamented* class. Red and green boxes corresponds to score and peformed notes respectively. Vertical lines indicate performance to score alingment. Notes linked with two of more lines (or not linked) correspond to *ornamented* class

PA	Abbreviation	Units	Formula	Range
Ornamentation	Orn_n	Boolean	$ornament(N_n)$	$\{yes, no\}$
Duration Ratio	Dr_n	Percentage	$\frac{db_j}{db_i} * 100$	$[0, +\infty]$
Onset Deviation	Od_n	Beats	$ob_j - ob_i$	$[0, +\infty]$
Energy Ratio	Er_n	Percentage	$\frac{v_j}{mean(v)} * 100$	$[0, +\infty]$

Table 4.3: Expressive performance actions calculation.

Score note index *(i)*	Perform. note index *(j)*	Pitch dev. *(semitones)*	Onset dev. *(beats)*	Durat. ratio *(beat frac.)*
1	1	−1	−1/2	1/16
1	2	0	0	2/3
2	3	−3	−1/2	1/2
2	4	0	0	1/2
4	6	0	−1/2	1/16
4	7	0	1/2	1/16
4	8	−1	3/2	1/16
4	9	0	2	1/16
5	10	−3	−1/2	1/2
5	11	0	0	1
6	12	1	−1/2	1/8
6	13	0	0	1/8
6	14	−2	1/2	1/8
6	15	0	1	1/8
6	16	0	3/2	1/8

Table 4.4: Example of ornament annotation for the music excerpt of Figure 1.2.

ergy deviations of each ornament note with respect to the score *parent* note were annotated as shown in Table 4.4.

Chapter 5

Preliminary Experiments

In Chapter 3 and 4 we explained how performance data was obtained from the corresponding audio recordings, by proposing methodologies to improve automatic performance transcription such as pitch profile extraction optimization and rule based and energy filters. In this chapter we present a set of experiments to validate the improvements proposed for performance data extraction and analysis. Firstly we present experiments on the task of obtaining a performance representation in the form of note events from the pitch profile segmentation. Secondly, we present experiments on the optimization on the extraction of the pitch profile from monophonic-multitimbral audio signals. Thirdly, we present some experiments performed to validate our score to performance alignment approach using Dynamic Time Warping. This validation is performed based on the agreement level between human annotators. Finally, we report on improvements obtained, which were quantified based on accuracy measures designed for each specific approach.

5.1 Pitch profile segmentation

Our approach for automatic pitch profile segmentation was tested over 27 monophonic-monotimbral recordings (Section 3.1.2). As explained in section 3.3.3 our aim is to segment the obtained pitch profile in to note events defined by onset, offset, pitch, and energy. An evaluation on the accuracy of the automatic segmentation approach was performed based on Clarisse et al. (2002), by comparing the similarity between a *ground truth* of transcribed melodies (hand corrected by a musician) and the ones extracted by the system. This comparison was performed using dynamic time warping (DTW), for which we used the cost function explained in Section 4.2.1. First, the alignment between the transcription obtained by the system and the human corrected transcription is done. Later, we determine the number of *insertions*, *deletions*, and *one to one note correspondence.* For the later case, we distinguished between *exact* correspondence and pitch recognition error higher than one semitone. The above criteria for accuracy is summarized as follows:

- Notes omitted: is the percentage of notes not detected by the system and present in the ground truth (percentage of *deletions*).
- Notes omitted + added: is the sum of above two measures.
- Right detected notes: is the percentage of notes detected by the system and present in the ground truth with "exact" *one to one correspondence.*
- Note recognition error > 1 semitone: is the percentage of notes detected by the system ans present in the ground truth with *one to one correspondence*, but with a pitch error higher than 1 semitone.

5.1.1 Experiment set up

We applied the aforementioned accuracy measures were tested on three different transcriptions obtained at three different stages of the pitch profile segmentation process: a first transcription obtained from the raw pitch profile (not filtered), a second transcription obtained using the adaptative energy threshold filter (only), and a third transcription obtained from using the adaptative energy filter and the rule based filter. We were also interested in testing our scheme and compare it against the approach proposed by Mcnab et al. (1996) (implemented in Essentia library (Bogdanov et al., 2013)) and the transcription approach by Mauch et al. (2015). Therefore, we obtained a midi transcription of the same melodies using our approach and the two aforementioned methods, to later calculate the same accuracy measures with respect the ground truth.

5.1.2 Results

Detailed results of the evaluation made on each transcription stage are provided in Table 5.1. In general, improvement on all the accuracy measures proposed is observed, specially at the note addition/omission indexes.

	Not filtered	AET filter	AET + RB Filter
notes omitted	5.54%	5.90%	9.58%
notes added	86.70%	26.49%	7.85%
notes omitted + added	92.24%	32.39%	17.43%
right detected notes	35.50%	74.99%	84.40%
note recognition error > 1 semitone	2.31 %	1.43%	1.10 %

Table 5.1: Results obtained by comparing extracted and corrected melodies using at different filtering stages. *AET* and RB are used as an acronyms for *adaptative energy threshold* and *rule based*, respectively.

	McNab	Tony	AET + RB Filter
notes omitted	5.54%	5.90%	9.58%
notes added	86.70%	26.49%	7.85%
notes omitted + added	92.24%	32.39%	17.43%
right detected notes	35.50%	74.99%	84.40%
note recognition error > 1 semitone	2.31 %	1.43%	1.10 %

Table 5.2: Accuracy measures comparison among transcription systems.

In Table 5.2 we compare the accuracy measures obtained using the approach by Mcnab et al. (1996) and our adaptative energy threshold approach with heuristic rules. From the table it can be seen an improvement in performance for all the accuracy measures. This might be due to the effect of the use of the addaptative threshold, which improves the onset and offset detection of the note events with the same pitch occurring consecutively at small time intervals.

These results are good indicators that pitch profile segmentation for melody transcription is improved by applying the two filtering strategies (*adaptative energy threshold filter and rule based filter based on musical heuristics*). However, it is important to notice that transcription accuracy is sensitive to the parameters set for energy filtering, as well as to the values defined for minimum (allowed) note/gap lengths.

5.2 Optimization of melodic extraction from polyphonic signals for jazz guitar

In this section we will deepen on the methodology and the experiments performed to optimize the parameters of the *melodia* algorithm Salamon and Gómez (2012). Two experiments were proposed to test our methodology for optimizing melody extraction. A first experiment was carried out using 22 simultaneous audio midi recordings. Ground truth was generated from the recorded MIDI data. A second experiment set-up was created in which the data was extracted from a jazz guitar teaching series book in which the melody is recorded separately from the accompaniment track. In this case ground truth was build from automatic extraction of the pitch contour of the melody track, using YIN algorithm (De Cheveigné and Kawahara, 2002) and performing manual correction afterwards.

5.2.1 First Experiment Set-up

The train data set consisted of 22 of the 27 jazz standard pieces, recorded by a professional guitarist, as mentioned in Section 3.1. The recording conditions were the ones mentioned in Section 3.1.2: The indications given to the musician were that melodies should be played monophonic, with no chord strums, or double notes. Only the main melody of the tune was recorded. Non repeated sections of the recorded melody were selected as shown in Table 3.1. The melodies were recorded over commercial backing tracks recordings (Kennedy and Kernfeld, 2002), in which bass and piano parts are recorded separately on each stereo channel. This made possible to create two different instrument settings:

- Trio setting: drums, bass and guitar.
- Quartet setting: piano, drums, bass and guitar (quartet).

Following a similar approach as Salamon and Gómez (2012), we created three audio mixes for each piece and for each setting (trio and quartet), in which the melody is at three different sound levels ($-6dB$, $0dB$ and $+6dB$) with respect to the overall sound level of the backing track. Eighteen of the tracks were used as train data to optimize parameters. The optimized parameters were tested in the four songs that were left aside for testing.

Building a ground truth

All the melodies were simultaneously recorded in both audio and MIDI, using a commercial guitar to MIDI converter (Limited, 2012). Because the accuracy of the device is not perfect, manual correction of the audio to MIDI converted melody was performed after each recording. Each MIDI event (note) has information of pitch (midi number), onset (in seconds) and duration (in seconds). Using this information and a hop size of 46.4 ms (as used by the melody extractor algorithm) it was possible to obtain the *ground truth* pitch contour of the melody frame by frame.

5.2.2 Second Experiment Set-up

In a second experiment, the data set consisted of four standard jazz tunes obtained from a jazz guitar teaching series book (Marshall, 2000), in which melody is recorded separately from the accompaniment track. This set was chosen to test our optimization strategy in a more real context in which common effects such as reverb can make melody extraction more difficult. We set up three different mixes with different sound levels of the melody, as performed in experiment number one (Section 5.2.1). For each mix we used a "leave one out" approach in which tree of the songs were used to optimize the parameters, and a forth was used for testing. This process was repeated iteratively four times.

Building a ground truth

In this case, the pitch contour of the melody was obtained from the melody track using Yin algorithm (De Cheveigné and Kawahara, 2002). We set the frame resolution for extraction of the algorithm to 46.4ms to be consistent with the frame resolution of *melodia* algorithm. Typical errors such as octave shifts and/or missing pitches, were corrected manually (by a trained musician) by comparing the obtained pitch contour against the melody signal spectrum.

5.2.3 Implementation: Genetic Algorithm

After defining an initial point, a range for each parameter and a fitness function, as explained in section 3.3.2, the genetic algorithm parameters were set with an initial population of a set of 20 individuals, i.e. vectors with different parameter values combinations. The stopping criteria was set to a maximum of 500 iterations, and a relative threshold change in fitness function of $1x10^{-6}$. Crossover factor was set to 0.8 and mutation factor was set to 0.02, which are typical settings for these values.

5.2.4 Results

For the first experiment (18 songs) the results for the training set is presented in Table 5.3. For each instrument setting (Trio and Quartet) we

	Trio		Quartet	
Optimization:	Before	After	Before	After
Mix level	Acc (%)	Acc (%)	Acc (%)	Acc (%)
0 dB	54.25	73.16	46.43	73.49
-6 dB	41.93	55.44	46.34	54.62
6 dB	79.75	82.80	81.15	82.54

Table 5.3: Accuracy after and before optimization for the **training set** for **first experiment**, at different mixing levels, and for each of instrument settings (Trio and Quartet).

	Trio		Quartet	
Optimization:	Before	After	Before	After
Mix level	Acc (%)	Acc (%)	Acc (%)	Acc (%)
0 dB	52.29	72.58	59.92	73.16
-6 dB	43.10	53.06	46.95	55.45
6 dB	81.52	82.80	74.08	81.37

Table 5.4: Accuracy after and before optimization for the **test set** for **first experiment**, at different mixing levels, and for each of instrument settings (Trio and Quartet).

apply the optimization methodology on the three different mixing levels. The accuracy was calculated summing all the frames of all the analysed tunes following the Equation 3.4, presented in section 3.3.2, for each instrument setting, before an after optimization.

In Table 5.5 we present the accuracy obtained in the test set. The melodies of the test songs were extracted using the default parameters, and the optimized ones. The results are shown for both of the instrument settings at the three different mixing levels, before and after optimization.

For the second experiment, a data-set of four songs extracted from a jazz guitar education series book (Marshall, 2000). We optimized the parameters using a leave one out approach. In Table 5.5 we present, the mean accuracies before and after optimization for the 4 folds.

From the tables it can be seen that the performance of the algorithm increases as the sound level of the mix increases, however the improvement on -6dB scenario is less than in +6dB scenario. This is expected since, as it is the case for human listeners, the melody is harder to be differentiated from the backing track if the sound level is too low. In Table 5.5, the results are similar as the ones described above, but the accuracy is lower in general.

	Trio		Quartet	
Optimization:	Before	After	Before	After
Mix level	Acc (%)	Acc (%)	Acc (%)	Acc (%)
0 dB	52.29	72.58	59.92	73.16
-6 dB	43.10	53.06	46.95	55.45
6 dB	81.52	82.80	74.08	81.37

Table 5.5: Mean accuracy after and before optimization for the **leave one out** approach for **second experiment**, at different mixing levels, and for each of instrument settings (Trio and Quartet).

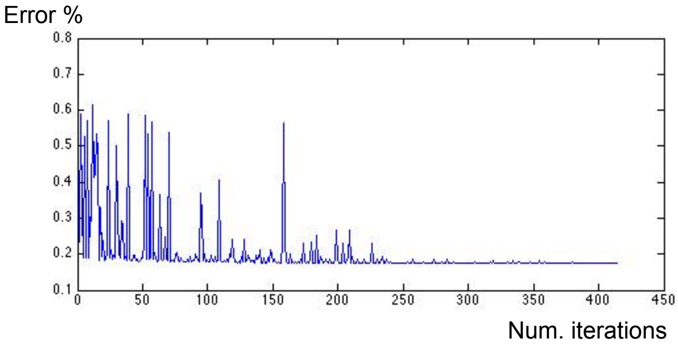


Figure 5.1: Learning curve of accuracy over number of iterations. Genaetic algorithm converges after 250 iterations.

This is due to the fact that we are now testing on unseen data. However in all the cases there is a considerable improvement of the performance using optimized parameters compared to the default ones.

In Figure 5.1 the error against the number of iterations is plotted for the quartet setting and an mix of +6dBs. In this figure it is possible to notice how after approximately 250 iterations the algorithm finds a minimum. Also, during the first iterations of the algorithm, it is possible to notice how sensitive is the algorithm to the chosen parameters.

5.2.5 Discussion

The work presented in this section refers to a methodology to solve an specific problem, rather than a system to be used to general problems (e.g as a melodic extraction plug-in). This same methodology could be applied to other performance settings for other instruments and other music styles.

However, the need of an annotated data set will be necessary in any case. For example if we aim to improve melody extraction for a certain performance settings (e.g. melody played by saxophone), an annotated data set with the ground truth would be necessary to optimize the parameters of the algorithm for that specific type of musical setting.

Choosing genetic algorithms against other search methods (e.g. Grid Search) was done mainly for efficiency reasons, as they provide an smarter strategy when searching over a six dimensional space. If we think of a method like grid search on a six dimensional space and we set 10 values per parameter, we will need to try 106 combinations to extract the melody. This is more computational expensive, compared to the method proposed: In Figure 3.1 the best fitness value is plotted against the number of iterations for one of the scenarios. A minimum is found after 250 iterations, with a population of 20 individuals per iteration, resulting in 5000 computations. In all scenarios the minimum was found in a similar number of iterations.

5.3 Human score to performance sequence matching for ornament detection.

Because there are not clear rules of how embellishments are performed, there is no ground truth for establishing a correspondence between performance and score notes. In this section we describe a methodology to obtain a ground truth for melodic correspondence matching between performances and scores, based on the agreement between the trained musicians' alignment. An experiment set-up was created in which we asked 5 musicians to align each score to its respective performance. Later an agreement analysis was performed between the alignments realized by the musicians. A measure to evaluate the automatic alignment against the alignment performed by the musicians was created based on the agreement analysis using penalization factors based on *high*, *medium*, and *low* agreement. Based on the accuracies obtained we estimate how close behaves the automatic alignment with respect to the human alignment.

5.3.1 Experiment set up.

In this experiment, we asked trained musicians to manually match performed notes with the corresponding parent score notes for each piece. Each of the pieces was annotated by 5 different musicians. Musicians were asked to associate performance notes to score notes by drawing lines in a piano

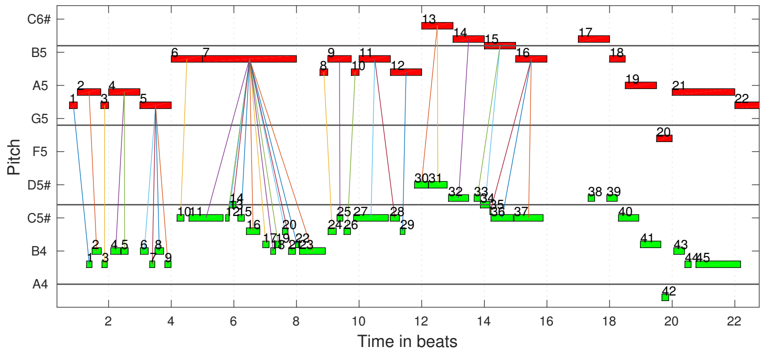


Figure 5.2: GUI for manual aligment of score and performance. Top sequence represents the score, botom represents the performance. Users were asked to link score and performance notes using vertical lines

roll representation in a GUI developed for this purpose. Figure 5.3 shows an example annotation of one user for a piece fragment. The upper note sequence corresponds to the score, and lower note sequence corresponds to the performance. Vertical/diagonal lines were marked by the musician to indicate performance to parent score note correspondence.

5.3.2 Agreement analysis

There are some cases in which the correspondence is ambiguous. Therefore, different musicians may choose to match different performance notes with one score note. In those cases we weighted each couple of linked notes based on how many musicians chose to link that particular pair of notes. Link occurrence count was stored in a matrix defined as follows:

$$H_{agree}[i,j] = \sum_{u=U}^{corresp_{i,j}(u)} \tag{5.1}$$

Where i, j corresponds to the note indexes of score and performance sequences, respectively, U is the total number of annotators (five), and $corresp$ is a binary function that returns 1 or 0 if the user u gave a positive or negative correspondence to a score-performance pair $[i, j]$ (i.e. if the pair was linked or not). This, the highest rating was given to a pair of notes which were matched by all five musicians (i.e. $highagreement = 5$), whereas the

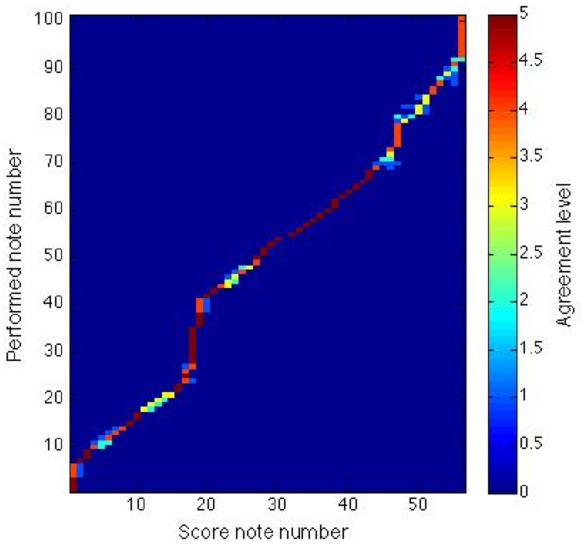


Figure 5.3: Agreement analysis example for *All of me*. Horizontal and vertical axes correspond to score and performance note sequences respectively. Red dots indicate regions of high agreement, whereas *clouds* of blue, yelow and green dots indicate low agreement

lowest rating was given to notes for which no musician match the pair (i.e. $lowagreement = 0$). In Figure 5.3, we present a graph of the link occurrence count matrix of one of the pieces in the dataset. In the figure is possible to identify sections in the piece with high agreement and sections with low agreement (ambiguity).

5.3.3 Results

We quantify the overall performance of our approach, based on the backtrack path output of the algorithm and the agreement analysis of human annotations. Accuracy was calculated by penalizing it when the algorithm output when it diverges from the human agreement. The evaluation criteria was defined as follows:

- High Agreement: if the algorithm matches a pair of notes with the highest agreement of the annotated dataset (i.e. all experts agree with the algorithm output), then the penalization is zero. On the contrary if the predicted matching pair of notes was not annotated by any of the experts then the penalty is 1.

- Medium Agreement: if the automatically matched pair of notes has medium agreement (i.e. 4/5 of the experts agree and 1/5 do not) the penalization is proportional to the agreement between a certain expert among the rest of the experts.

- Low Agreement: if the automatically matched pair of notes has medium agreement (i.e. 3/5 of the experts agree and 2/4 do not) the penalization is proportional to the agreement between a certain expert among the rest of the experts.

This means that the algorithm makes a mistake which is similar to the one a human expert would make. The overall accuracy is then calculated as follows:

$$accuracy = \left(\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} P_f[i,j]}{size(H_{DTW})} \right) \tag{5.2}$$

where H_{DWT} is the alignment matrix obtained using the *dynamic time warping* approach (as described in Section 4.2.1), and $P_f i, j$ is a penalization factor calculated by comparing (cell by cell) the alignment matrix obtained with the algorithm (H_{DTW}) and the one obtained with the agreement analysis (H_{agree}), as follows:

$$P_f(i,j) = \begin{cases} \text{High} & \begin{cases} \text{if } H[i,j]_{agree} = 5 \text{ and } H_{DTW}[i,j] = 1 & \rightarrow 1 \\ \text{if } H[i,j]_{agree} = 5 \text{ and } H_{DTW}[i,j] = 0 & \rightarrow 0 \\ \text{if } H[i,j]_{agree} = 0 \text{ and } H_{DTW}[i,j] = 1 & \rightarrow 0 \\ \text{if } H[i,j]_{agree} = 0 \text{ and } H_{DTW}[i,j] = 0 & \rightarrow 1 \end{cases} \\ \text{Medium} & \begin{cases} \text{if } H[i,j]_{agree} = 4 \text{ and } H_{DTW}[i,j] = 1 & \rightarrow 4/5 \\ \text{if } H[i,j]_{agree} = 4 \text{ and } H_{DTW}[i,j] = 0 & \rightarrow 1/5 \\ \text{if } H[i,j]_{agree} = 1 \text{ and } H_{DTW}[i,j] = 1 & \rightarrow 1/5 \\ \text{if } H[i,j]_{agree} = 1 \text{ and } H_{DTW}[i,j] = 0 & \rightarrow 4/5 \end{cases} \\ \text{Low} & \begin{cases} \text{if } H[i,j]_{agree} = 3 \text{ and } H_{DTW}[i,j] = 1 & \rightarrow 3/5 \\ \text{if } H[i,j]_{agree} = 3 \text{ and } H_{DTW}[i,j] = 0 & \rightarrow 2/5 \\ \text{if } H[i,j]_{agree} = 2 \text{ and } H_{DTW}[i,j] = 1 & \rightarrow 2/5 \\ \text{if } H[i,j]_{agree} = 2 \text{ and } H_{DTW}[i,j] = 0 & \rightarrow 3/5 \end{cases} \end{cases} \tag{5.3}$$

5.3.4 Discussion

The mean accuracy obtained for the 27 recordings set is of 80,76%, with a standard deviation of 0.10 (10%). We also quantify the agreement among experts. Each expert alignment was compared to the alignment made by the other 4 experts on each song, following the same evaluation criteria. In Figure 5.4 the calculated accuracy of the algorithm for each individual piece is depicted in the first bar of each song. The second bar represents the average agreement among experts. Results may indicate that the behaviour of the automatic alignment is comparable with the annotations made by humans.

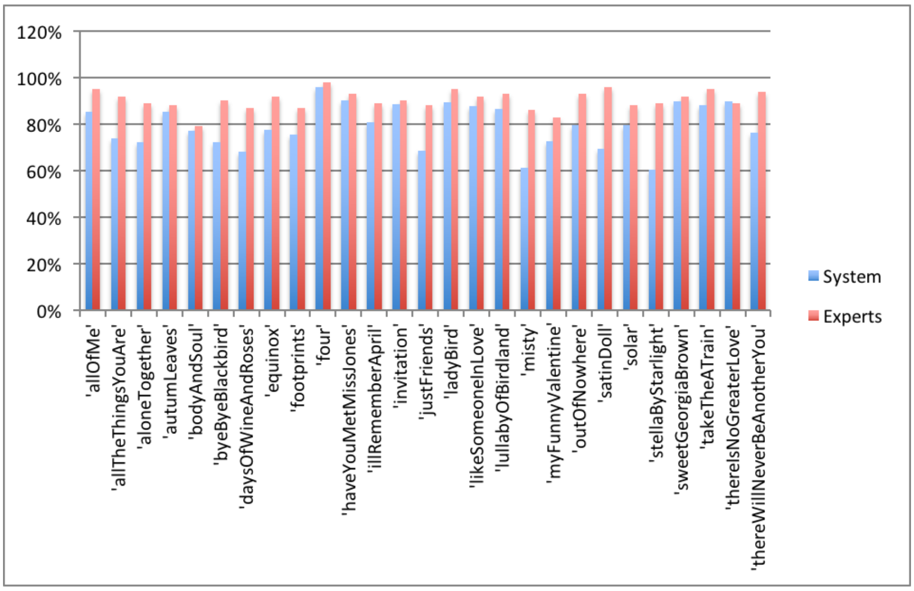


Figure 5.4: Accuracy based on agreement between automatic alignment systems and experts alignment

Chapter 6

Expressive Performance Modeling

In this chapter we present the experiments performed using our approach for expressive music performance modelling using machine learning. After each note was characterized by its musical context description (see Chapter 4), several machine learning techniques are explored to achieve two concrete aims. On one hand, from the recordings made by a professional guitarist (monophonic-monotimbral) we induce regression models for timing, onset and dynamics (i.e. note duration and energy) transformations, and classification models for ornamentation, to later select the most suitable ornament for predicted ornamented notes based on note context similarity, and finally render an expressive performance applying the performance actions predictions obtained with the induced models. On the other hand, from the commercial audio recordings of a well known guitarist (monophonic-multitimbral) we apply automatic knowledge discovery techniques to the resulting features to learn expressive performance rule models. We analyse the relative importance of the considered features, quantitatively evaluate the accuracy of the induced models, and discuss some of the learnt expressive performance rules. We report on the performance for the ornamentation, duration, onset, and energy models and rule learners . Similarities between the induced expressive rules and the rules reported in the literature are discussed. The rules' performer specificity/generality is assessed by applying the induced rules to performances of the same pieces performed by two other professional jazz guitar players.

6.1 Models from monophonic-monotimbral recordings

The aim of this section is twofold: 1) to train computational models of music expression using recordings of a professional guitar jazz player, and 2) synthesize expressive ornamented performances from inexpressive scores. In order to train a jazz guitar ornamentation model, we recorded a set of 27 jazz standards performed by a professional jazz guitarist (Section 3.1.2. We extracted symbolic features from the scores using information of each note , information of the neighbor notes, and information related to the musical context (Section 4.1. The performed pieces were automatically transcribed by applying note segmentation based on pitch and energy information (Section 3.3). After performing score to performance alignment, using *Dynamic Time Warping* (Section 4.2.1) , we calculated performance actions by measuring the deviations between performed notes and their respective parent notes in the score (Section 4.2.2. For model evaluation the data set was split using a leave-one-piece-out approach in which each piece was in turn used as test set, using the remaining pieces as training set. The expressive actions considered in this section were *duration ratio*, *onset deviation*, *energy deviation* and *ornamentation transformations.* Concatenative synthesis was used to synthesize new ornamented jazz melodies using adapted notes/ornaments samples from the segmented audio recordings.

6.1.1 Expressive performance modeling

Several machine learning algorithms (i.e. *Artificial Neural Networks* (ANN), *Decision Trees* (DT), *Support Vector Machines* (SVM), and *k-Nearest Neighbor* (k-NN)) were applied to predict the ornaments introduced by the musician when performing a musical piece. The accuracy of the resulting classifiers was compared with the *baseline classifier*, i.e. the classifier which always selects the most common class. Timing, onset and energy performance actions were modeled by applying several regression machine learning methods (i.e. *Artificial Neural Networks*, *Regression Trees* (RT), *Support Vector Machines*, and *k-Nearest Neighbor*).

Based on their accuracy we chose the best performance model and feature set to predict the different performance actions. Each piece (used as test set) was in turn predicted based on the models obtained with the remaining pieces (used as training set) and synthesized using a concatenative synthesis approach.

Algorithm comparison

In this study we compared four classification algorithms for ornament prediction, and four regression algorithms for duration ratio, onset deviation, and energy ratio. We used the implementation of the machine learning algorithms provided by the WEKA library (Hall et al., 2009). We applied, k-NN with $k = 1$, SVM with linear kernel, ANN consisting of a fully-connected multi-layer neural network with one hidden layer, and DT/RT with post pruning.

A paired T-test with a significance value of 0.05 was performed for each algorithm for the ornamentation classification task, over all the data set with 10 runs of 10-fold cross validation scheme. Experiments results are presented in Table 6.3 and will be commented in Section 7.2.4.

Feature selection

Both filter and wrapper feature selection methods were applied. Filter methods use a proxy measure (e.g. information gain) to score features, whereas wrappers make use of predictive models to score feature subsets. Features were filtered and ranked by information gain values, and a wrapper with greedy search and decision trees accuracy evaluation was used to select optimal feature subsets. We used the implementation of these methods provided by WEKA library (Hall et al., 2009). Selected features are shown in Table 6.1, and will be commented in Section 7.2.4.

Learning curves on the number of features, as well as on the number of instances were obtained to measure the learning rate of each of the algorithms. The selection of the model was based on the evaluation obtained with these performance measures.

Synthesis

Predicted pieces were created in both MIDI and audio formats. A concatenative synthesis approach was used to generate the audio pieces. This process consists of linking note audio samples from real performances to render a synthesis of a musical piece. The use of this approach was possible, as we have monophonic performance audio data in which onset and offset information was extracted based on energy and pitch information as described in Section 3.3.1. Therefore, it is possible to segment the audio signal into individual notes, and furthermore, obtain complete audio segments of ornaments.

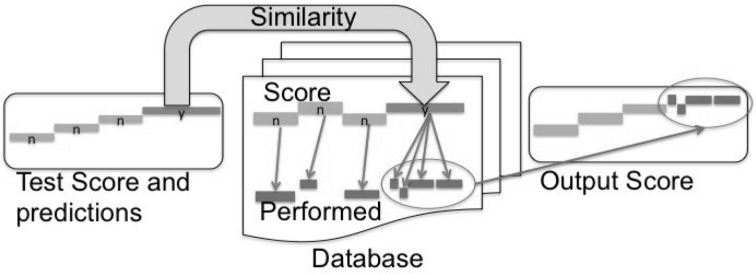


Figure 6.1: Concatenative Synthesis Approach.

Similarly to the evaluation of the different machine learning algorithms, for synthesis we have followed a leave-one-piece-out scheme in which on each fold, the notes of one piece were used as test set, whereas the notes of the remaining 26 songs were used as training set.

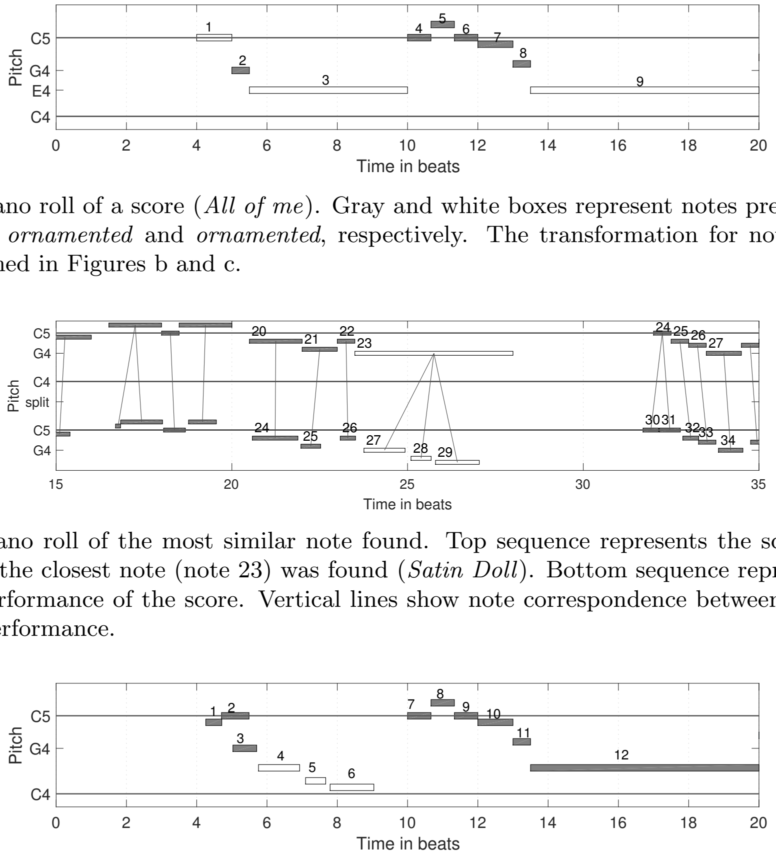
Note concatenation

The note concatenation process is divided into three different stages as depicted in Figure 6.1.

- Sample retrieval: For each note predicted as ornamented, the k-NN algorithm, using *Euclidean* distance similarity function based on note description, was applied to find the most suitable ornamentation in the database (see Section 4.2.3). This was done searching the most suitable ornament in the songs in the training set (Section 6.1.1).

- Sample transformation: For each note classified as ornamented, transformations in duration, onset, energy and pitch (in the case of ornaments) were performed based on the deviations stored in the database, as seen in Figures 6.2a and 6.2b. For audio sample transformation we used the time and pitch scaling approaches by Serra (1997). Notes classified as not ornamented were simply transformed as predicted by the duration, onset and energy models.

- Sample concatenation: Retrieved samples were concatenated based on final onset and duration information after transformation. The tempo of the score being predicted (in BPM), was imposed to all the retrieved notes.



(a) Piano roll of a score (*All of me*). Gray and white boxes represent notes predicted as *not ornamented* and *ornamented*, respectively. The transformation for note 3 is explained in Figures b and c.

(b) Piano roll of the most similar note found. Top sequence represents the score in which the closest note (note 23) was found (*Satin Doll*). Bottom sequence represents the performance of the score. Vertical lines show note correspondence between score and performance.

(c) Piano roll of a partial predicted score (*All of me*). Note number 3 of Figure (a) has been replaced by notes 27, 28, and 29 of Figure (b), obtaining notes 4, 5, and 6.

Figure 6.2: Sample retrieval and conatenation example.

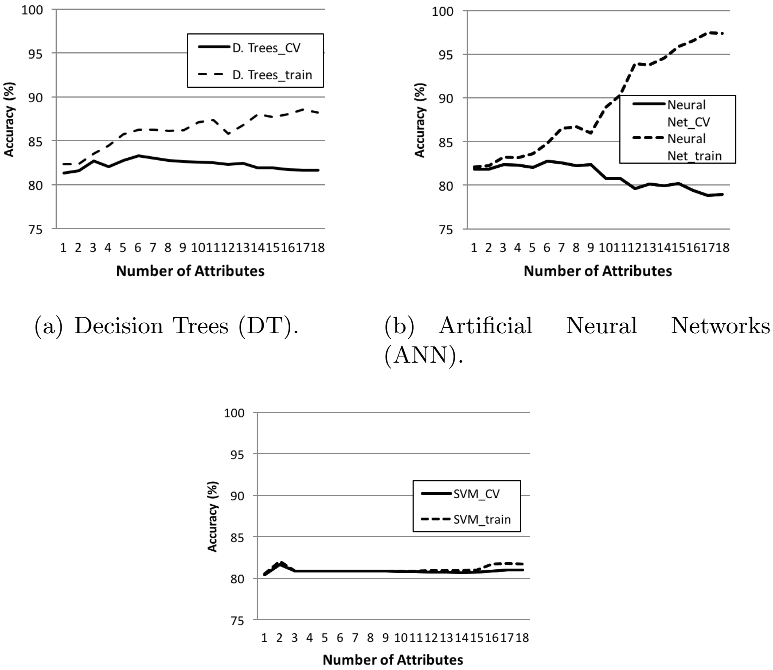
Info-Gain + Ranker	Wrapper + Greedy
Duration (sec)	Duration beat
Duration (beat)	Prev. duration (sec)
Phrase	Tempo
Prev. duration (sec)	Phrase
Onset in bar	Velocity
Metrical strength	Onset beat
Prev. duration (beat)	Duration (sec)
Next duration (beat)	Prev. duration (beat)
Next duration (sec)	Next duration (beat)
Narmour 1	Narmour 3
Tempo	Metrical strength
Chord type	Onset in Bar
Prev. interval	Narmour 1
Next interval	Prev. interval
Narmour 2	Narmour 2
Narmour 3	Is chord note
Is chord note	Onset sec
Mode	Key
keyMode	Chord root
	Next duration (sec)
	Pitch
	Note to key
	Note to chord
	Measure
	Next interval
	Chroma
	Chord type

Table 6.1: Features selected using Filter and the Wrapper (with J48 classifier) methods.

6.1.2 Results

Feature selection

The most relevant features found using the two selection methods described in Section 6.1.1 are shown in Table 6.1. The average correctly classified instances percentage (C.C.I.%) obtained using the features selected by the information gain filtering and the greedy search (decision trees) wrapper methods were 78.12% and 78.60%, respectively (F-measure of 0.857 and 0.866, respectively). Given that both measures are similar, i.e., not significantly different, the smallest subset was chosen.



(a) Decision Trees (DT).

(b) Artificial Neural Networks (ANN).

(c) Support Vector Machines (SVM).

Figure 6.3: Accuracies on increasing the number of attributes.

In Figure 6.3, accuracy on increasing the number of features based on the information gain ranking (explained in Section 6.1.1) are presented, for each of the algorithms used (SVM, ANN, DT). From the curves it can be seen that the subset with the first 3 features contains sufficient information, as additional features do not add significant accuracy improvement. SVM exhibits better accuracy on the cross validation scheme, and less over-fitting based on the difference between *cross validation* (CV) and *train set* (TS) accuracy curves.

Quantitative evaluation

Algorithm comparison results

For the (ornament) classification problem we compared each of the algorithms (SVM, DT, ANN and k-NN) with the baseline classifier (i.e. the

Dataset	Baseline	k-NN	DT	SVM	ANN
	(CCI%)	(CCI%)	(CCI%)	(CCI%)	(CCI%)
Ornament	72.74	70.58	78.68 ∘⋄	77.64 ∘⋄	76.60 ∘⋄

∘, • statistically significant improvement or degradation against *Baseline classifier.*

⋄, ⋆ statistically significant improvement or degradation against *Instance base learner.*

Table 6.2: Correctly Classified Instances (CCI%) comparison with Paired-T test for classification task.

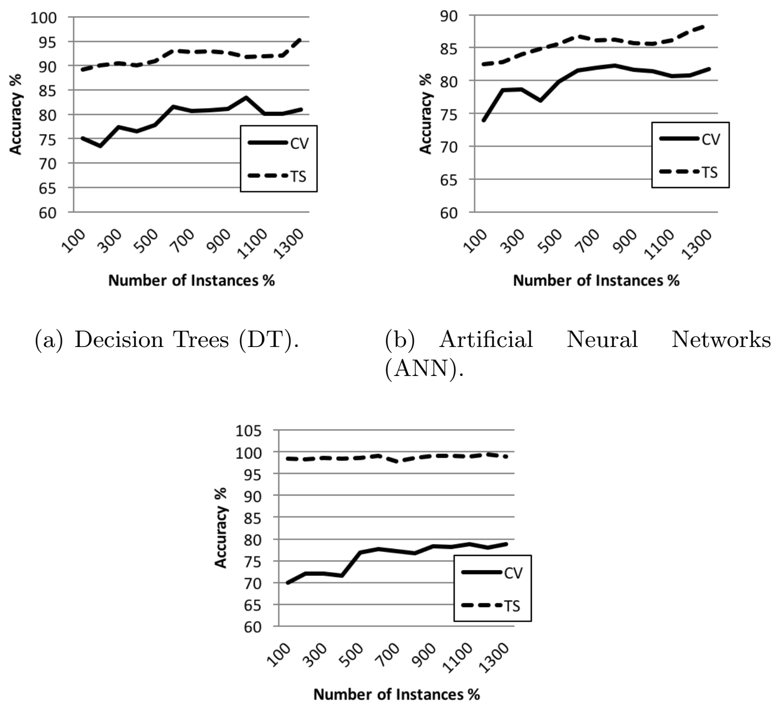
Dataset	k-NN		Reg. Trees		reg. SVM		ANN	
	PCC	R^2	PCC	R^2	PCC	R^2	PCC	R^2
Dur. rat.	0.20	0.04	0.25	0.06	0.17	0.03	0.19	0.04
Enr. rat.	0.19	0.04	0.37	0.14	0.38	0.14	0.37	0.14
Onset dev.	0.41	0.17	0.51	0.26	0.43	0.18	0.44	0.19
mean	0.25	0.08	0.38	0.15	0.33	0.12	0.33	0.12

Table 6.3: Pearson Correlation Coefficient (PCC) and Explained Variance (R^2) for the regression task.

majority class classifier) following the procedure explained in Section 6.1.1. From Table 6.2 it can be seen that all the algorithms present a statistically significant improvement, except k-NN. Given the accuracy results, we apply the ornamentation prediction model induced by the DT algorithm to determine whether a note is to be ornamented or not. We discarded the use of k-NN for this task due to its low accuracy, which led to larger mis-classifications of ornamented and not ornamented notes.

For the regression problems (duration, onset and energy prediction) we applied *regression trees*, *SVM*, *neural networks*, and *k-NN*, and obtained the correlation coefficient values shown in Table 6.3. *Onset deviation* has the highest correlation coefficient, close to 0.5.

For ornamentation classification using k-NN we explored several values for k ($1 \leq k \leq 10$). However, all of the explored values for k resulted in inferior classification accuracies when compared with decision trees and SVM. As in the case of $k = 1$, both the decision trees and SVM classifiers resulted in statistically significant higher accuraccies (based on T-test) when compared with the classifiers for $2 \leq k \leq 10$.



(a) Decision Trees (DT).

(b) Artificial Neural Networks (ANN).

(c) Support Vector Machines (SVM).

Figure 6.4: Accuracies on increasing the number of instances.

Learning Curves

The learning curves of accuracy improvement, for both cross validation and training sets, over the number of instances are shown in (Figure 6.4). The learning curves were used to measure the learning rate and estimate the level of overfitting. Data subsets of different sizes (in steps of 100 randomly selected instances) were considered and evaluated using 10-fold cross validation. In general, for the three models, it can be seen that the accuracy on CV tends to have no significant improvement above 600 instances.

Overfitting can be correlated to the difference between accuracy in CV and TS, in which a high difference means higher levels of overfitting. In this sense, in Figure 6.4 (c), SVM shows a high tendency for overfitting, but seems to slowly improve it over the number of instances. On the other hand, in Figure 6.4 (a) and (b), ANN and DT seem to improve overfiting between 700 and 1100 instances. This could mean that adding more instances may improve slightly the accuracy on both CV and TS for the three models, and may slightly improve overfitting for SVM, but this may not be the case for ANN and DT.

Obtained pieces

Figure 6.5 shows a MIDI piano roll of an example piece performed by a professional musician and the predicted performance obtained by the system, using a decision trees classifier. It can be noticed how the predicted piano roll follows a similar melodic structure as the one performed by the musician. For instance, for the score notes predicted correctly as ornamented (true positives), notes 1, 10, and 34 in Figure 6.5(a) (top sequence), the system finds ornaments of similar duration, offset and number of notes as the musician's performance. Also, score notes 3 and 9 of Figure 6.5(b) (false positives), are ornamented similarly as score notes 18 and 26 (Figure 6.5(a)) which are in a similar melodic context.

Duration and energy ratio curves

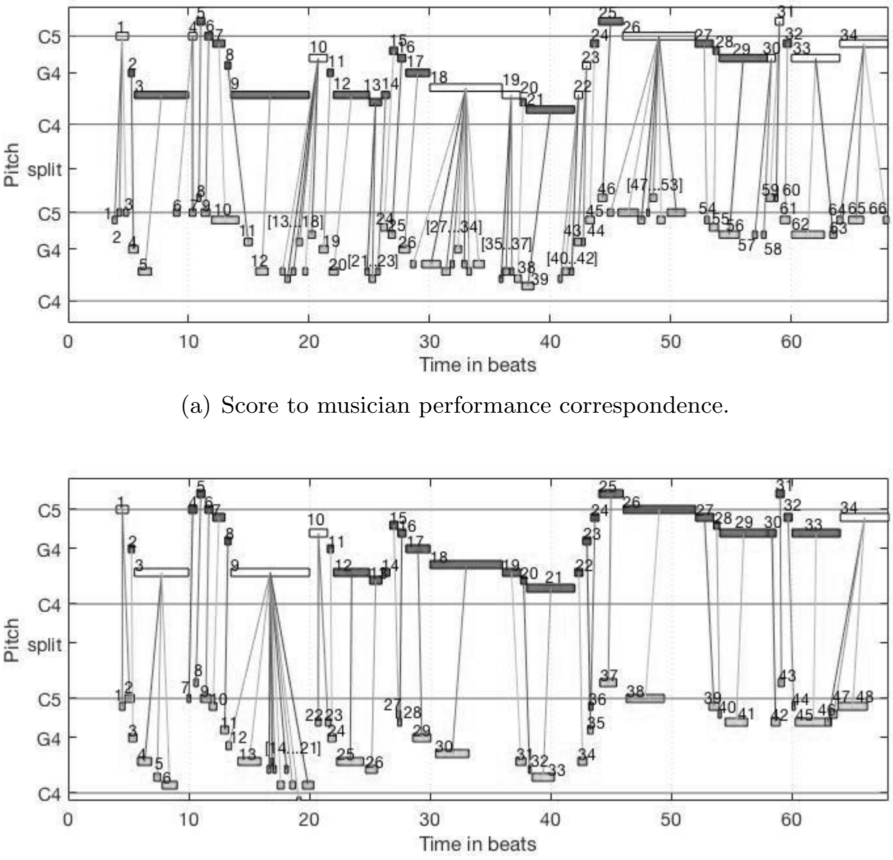
Duration and energy deviation ratio measured in the musician performance and predicted by the system for one example piece (*All of me*) are compared in Figure 6.6(a) and 6.6(b), respectively. We obtained similar results for the other pieces in the data set. Similarity between the contour of the curves indicate that the deviations predicted by the system are coherent with the ones performed by the musician.

Musical Samples

Musical examples of the automatically generated ornamented pieces can be found at the Online Supplement. The rendered audio of the *Yesterdays* music piece generated by the system (as test piece) has been included in this site.

6.1.3 Evaluation

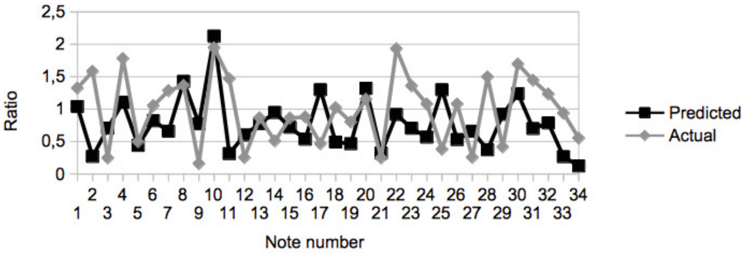
Perceptual tests to evaluate our approach on how the generated pieces are perceived by listeners presents a significant complication, because of variability in perception, as individual responses may be subjected to personal expectations (Poli et al., 2014). Moreover, ranking tests might be bias by



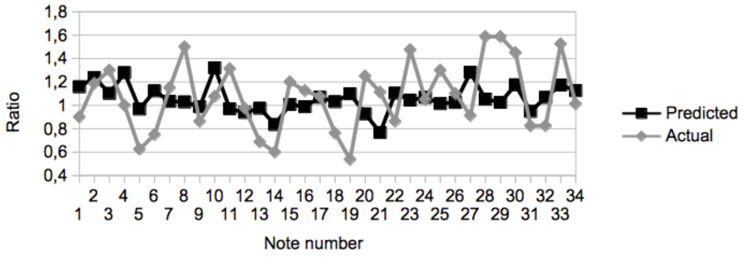
(a) Score to musician performance correspondence.

(b) Score to predicted performance correspondence.

Figure 6.5: Musician vs. predicted performance. Top and bottom sequences represents score and performance piano roll respectively. Vertical lines indicate score to performance note corresponcence. Gray and white boxes represent notes predicted as *not ornamented* and *ornamented*, respectively.



(a) Duration ratio: performed vs. predicted.



(b) Energy ratio: performed vs. predicted.

Figure 6.6: Performed vs. predicted duration and energy ratio example for *All of Me*. Gray and black lines represent performed and predicted ratios, respecively, for each note in the score.

the quality of the synthesis (which is out of the scope of this dissertation). Therefore, we followed a similar approach as Grachten (2006) to evaluate the performances generated by the system by computing the alignment distance between the system and the target performance. According to Grachten, the cost function must reflect the human perception of similarity between performances. Therefore, the cost function was redefined to include a weight vector for each *cost distance component* (i.e. *pitch*, *duration*, *onset*, *ornament onset*, and *ornament offset*) between score and performance. Given that a similarity ground truth was previously built based on agreement among human annotators (see Section 5.3), we optimized the weights of the new cost function to achieve the closest alignment as to the one obtained with the human annotations with highest agreement. Thus, the weights obtained might reflect the relevance of each distance component based on human perception.

Distance cost function redefinition

Alignment distance was previously computed using the cost function defined in Section 4.2.1 (see Equations 4.1 to 6.2). The weight vector for each distance component was defined as follows:

$$weigth(n) = [w_p, w_{ds}, w_{ons}, w_{ph-ons}, w_{ph-pfs}] \tag{6.1}$$

where $w_p, w_{ds}, w_{ons}, w_{ph-ons}$, and w_{ph-pfs} are weighting factors for each distance component (previously defined for *pitch* ($p(i)$), *duration* ($ds(i)$), *onset* ($ons(i)$), *ornament onset* ($ph_{ons}(i)$, and *ornament offset* ($ph_ofs(i)$)), respectively. Thus, Equation 6.2 for cost calculation using the *Euclidean distance* is redefined as follows:

$$cost(i,j) = \sqrt{\sum_{n=1}^{5}((cs(n)_i - cp(n)_j) * (weight(n)))^2} \tag{6.2}$$

Optimization of individuals weights for the cost function

For optimization we followed a similar approach as the one described in Section 3.3.2, using *genetic algorithms* for weights optimization. In this case the fitness function corresponds to the accuracy calculated for automatic alignment (se Section5.3.2) as measured in Equations 5.2 and 5.3. The initial population was setted as follows:

$$\begin{gathered} weight_{Highbound} = [1,1,1,1,1], \\ weight_{Initialpoint} = [0.5, 0.5, 0.5, 0.5], \\ weight_{Lowbound} = [0,0,0,0,0], \end{gathered}$$

according to the defined five dimensional vector of Equation 6.1. The optimization was run with an initial population set to 20 individuals, a stopping criteria of 500 iterations and relative threshold change in fitness function of $1x10^{-6}$. Again crossover and mutation factors were set to 0.8 and 0.02, respectively.

Distance evaluation results

Each piece was rendered using the predictions of each of the generated models. Later we calculate the similarity of each of the obtained pieces with

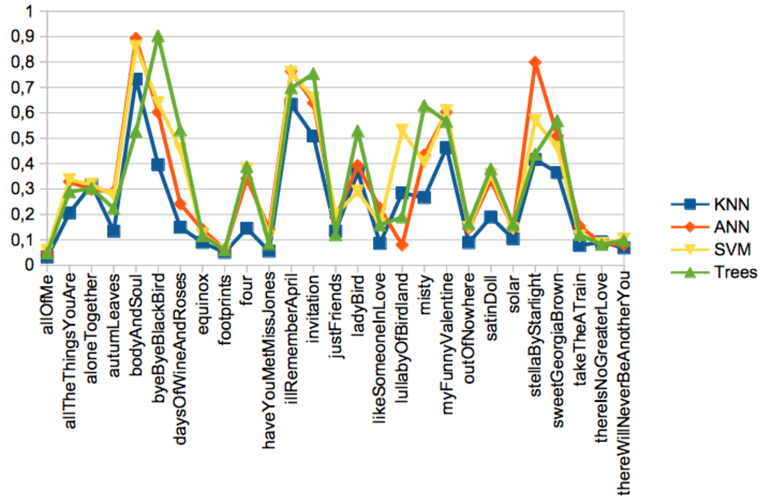


Figure 6.7:

respect a target performance, i.e. the performance recorded by the musician. Similarity was calculated by, firstly, calculating the alignment of each rendered piece and the corresponding target performance using the optimized cost function described in previous section. Normalized values for similarity are presented in Figure 6.7, where 1 accounts for maximum similarity and zero accounts for minimum similarity. From the figure it can be seen how *k-NN* algorithm similarity is lower than other algorithms. However, similarity obtained by all algorithms show a similar tendency over the pieces, which might indicate that performance of the models is dependent on the musical pieces it self.

6.1.4 Discussion

In this section we have presented a machine learning approach for expressive performance (ornament, duration, onset and energy) prediction and synthesis in jazz guitar music. We used a data set of 27 recordings performed by a professional jazz guitarist, and extracted a set of descriptors from the music scores and a symbolic representation from the audio recordings. In order to map performed notes to parent score notes we have automatically aligned performance to score data. Based on this alignment we obtained performance actions, calculated as deviations of the performance from the score. We created an ornaments database including the information of the ornamented notes performed by the musician. We have compared four learning algorithms to create models for ornamentation, based on performance measures, using a significance Paired T-test. Feature selection techniques were employed to select the best feature subset for ornament modeling. For synthesis purposes, instance based learning was used to retrieve the most suitable ornament from the ornamentation data base. A concatenative synthesis approach was used to automatically generate expressive performances of new pieces (i.e. pieces not in the training set).

6.2 Models from monophonic-multitimbral

In Chapter 7 we have reviewed several studies in computational modelling of expressive music performance. However, little attention was paid to the perspicuity of the extracted models in terms of its musical interpretation. Furthermore, the accuracy of the generated models is usually reported based mainly on accuracy measures. In this section we induce expressive performance rules by applying machine learning methods. In particular, we apply a propositional rule learner algorithm to obtain expressive performance rules from data extracted from commercial audio jazz recordings. We are interested in rules characterizing variations in timing (i.e. onset and duration deviation), energy (i.e. loudness) and ornamentation (i.e. insertion and deletion of an albitrary number of melody notes) in jazz guitar music. We align the scores to the corresponding audios and extract descriptors and EPA from the resulting alignment. We apply feature selection and machine learning algorithms to induce rule models for energy, onset, duration and ornamentation. Finally, we evaluate the accuracy of each of the models obtained, and discuss the obtained rules from a musicological perspective.

6.2.1 Materials

As mentioned in Section 3.1, the music material considered in this section consists of 16 commercial recordings of Grant Green (see Table 3.2), and their corresponding commercially available music scores (The real book). Music scores were obtained from The real book, a compilation of jazz pieces in the form of *lead sheets*. The collected music scores contain melody and harmony (i.e. chord progressions) information for the music material investigated.

6.2.2 Feature extraction: nominal descriptors

In Section 6.2.2 we explained how music scores were characterized by automatically extracting descriptors for each note. Because the aim of this Section is to obtain interpretable rules from a musical perspective, we select a set of features which were discretized as follows:

- Duration nominal. Duration in seconds was discretized into classes *very large*, *large*, *nominal*, *short*, and *very short*. We defined duration thresholds in seconds, as follows:

$$duration_{nom}(n) = \begin{cases} verylarge & \text{if } ds_n \geq 1.6s. \\ large & \text{if } 1.6s. \leq ds_n < 1s. \\ nominal & \text{if } 1s. \leq ds_n < 0.25s. \\ short & \text{if } 0.25s. \leq ds_n < 0.125s. \\ veryshort & \text{if } ds_n \leq 0.125s. \end{cases} \tag{6.3}$$

- Chroma nominal. The *chroma* value of each note was labelled as pitch classes (e.g. *C, D#, Eb*), according to the definition given in

Section3.2.1, as follows:

$$chroma_{nom}(n) = \begin{cases} C & \text{if } ch_n = 0 \\ C\#/Db & \text{if } ch_n = 1 \\ D & \text{if } ch_n = 2 \\ D\#/Eb & \text{if } ch_n = 3 \\ E & \text{if } ch_n = 4 \\ F & \text{if } ch_n = 5 \\ F\#/Gb & \text{if } ch_n = 6 \\ G & \text{if } ch_n = 7 \\ G\#/Ab & \text{if } ch_n = 8 \\ A & \text{if } ch_n = 9 \\ A\#/Bb & \text{if } ch_n = 10 \\ B & \text{if } ch_n = 11 \end{cases} \tag{6.4}$$

- Previous and next interval direction. Positive and negative intervals were labelled as descending and ascending intervals respectively, according with the calculation of *previous interval* ($pint_n$) and *next interval* ($nint_n$) given in Table 4.3, whereas intervals equal to zero were labelled as *unison*:

$$prevInt_{dir}(n) = \begin{cases} ascending & \text{if } pint_n > 0 \\ unison & \text{if } pint_n = 0 \\ descending & \text{if } 1s. \leq pint_n < 0 \end{cases} \tag{6.5}$$

$$nextInt_{dir}(n) = \begin{cases} ascending & \text{if } nint_n > 0 \\ unison & \text{if } nint_n = 0 \\ descending & \text{if } nint_n < 0 \end{cases} \tag{6.6}$$

- Previous and next interval size. Interval sizes were categorized into *small* and *large* based on the Implication-Realization model of Narmour (Narmour, 1992), which assumes that intervals smaller/larger than 6 semitones are perceived to be small/large. In this implementation we define intervals equal to 6 semitones to be large (as implemented by Grachten (2006)).

$$prevInt_{nom}(n) = \begin{cases} large & \text{if } pint_n \leq 6 \\ small & \text{if } pint_n < 6 \end{cases} \tag{6.7}$$

$$nextInt_{nom}(n) = \begin{cases} large & \text{if } nint_n \leq 6 \\ small & \text{if } nint_n < 0 \end{cases} \tag{6.8}$$

- Tempo nominal. Tempo indications in jazz often are refereed based on the performance style (e.g. Bebop, Swing) or on the sub-genre of the piece (e.g. medium, medium up swing, up tempo swing). However ambiguity on the BPM range for which this categorization corresponds exists among performers. In this section the discretization of the tempo of the piece was performed based on the performers' preferred tempo clusters found by Geoffrey L. Collier (1994). In the study, the tempo of several jazz recordings datasets are analysed and preferred tempo clusters of performers are found at 92, 117, 160, and 220 bpm. The study is based on the assumption that tempos in the range of 4 tempo cluster (attractor) may gravitate toward it. Based on this, we defined four different bpm ranges around each cluster and labelled it as follows.

$$tempo_{nom}(n) = \begin{cases} Up-tempo & \text{if } t_n \geq 180 \\ Medium & \text{if } 180 > t_n \geq 139 \\ Moderate & \text{if } 139 > t_n \geq 105 \\ Slow & \text{if } 105 > t_n \end{cases} \tag{6.9}$$

- Harmonic analysis (chord function). Chord definitions were discretized based on the chord simplification rules by Hedges et al. (2014), in which the notation of the chord type (*e.g.Ebmaj*7) is simplified according to the harmonic function of the chords. In this study we adapted the rules according to make them consistent according to the chord degree definitions given in Table 3.4, as follows:

$$chord_{func}(n) = \begin{cases} dom & \text{if } [4,10] \in chord\ degrees \\ maj & \text{if } [4] \in chord\ degrees \wedge [10] \notin chord\ degrees \\ min & \text{if } [3,7] \in chord\ degrees \\ dim & \text{if } ([0,3,6,] \vee [0,3,6,9]) = chord\ degrees \\ aug & \text{if } [\#5,+] \subset cht_n \\ hdim & \text{if } [0,3,6,10] = chord\ degrees \\ dom & \text{if } [10] \in chord\ degrees \wedge [sus] \subset cht_n \\ maj & \text{if } [10] \notin chord\ degrees \wedge [sus] \subset cht_n \\ NC & \text{if } no\ chord \end{cases} \tag{6.10}$$

6.2.3 Expressive performance actions characterizaton

Score notes aligned to exactly one performance note were labeled as *non ornamented*, whereas score notes aligned to non or several performance notes were labeled as *ornamented.* Performance actions deviations in duration, onset, and energy were discretized into classes, according to the Equations 6.11, 6.12, and deviation in energy was discretized according to Equation 6.13, as follows:

$$duration_{dev}(i,j) = \begin{cases} lengthen & \text{if } db_i - db_j \geq 1/16 \\ none & \text{if } -1/16 < db_i - db_j < 1/16 \\ shorten & \text{if } db_i - db_j \leq -1/16 \end{cases} \tag{6.11}$$

$$onset_{dev}(i,j) = \begin{cases} delay & \text{if } ob_i - ob_j \geq 1/16 \\ none & \text{if } -1/16 < ob_i - ob_j < 1/16 \\ advance & \text{if } ob_i - ob_j \leq -1/16 \end{cases} \tag{6.12}$$

$$energy_{dev}(i,j) = \begin{cases} piano & \text{if } v_n - mean(v) \leq -0.4 * stdev(v) \\ none & \text{if } -0.4 * stdev < v_n - mean(v) < 0.4 * stdev \\ forte & \text{if } v_n - mean(v) \geq 0.4 * stdev \end{cases} \tag{6.13}$$

where i and j refers to the index of the score and performance notes, respectively. In Equations 6.11 and 6.12

Duration was discretized into *lengthen*, *shorten*, and *none*; onset into *advance*, *delay*, and *none*; and energy into *piano*, *forte* and *none.* A note is considered to belong to class *lengthen/shorten*, if its performed duration is one *semiquaver* longer/shorter (or more/less) than its duration according to the score. Otherwise, it belongs to class *none.* Classes *advance*, *delay*, and *none* are defined analogously. A note is considered to be in class *forte/piano* if it is played louder/softer than the mean energy of the piece plus/minus 40% of its standard deviation and in class *none* otherwise.

6.2.4 Expressive performance modelling

Learning task.

We explored machine learning techniques to induce models for predicting the different expressive performance actions defined above. Concretely, our objective is to induce four classification models M1, M2, M3 and M4 for ornamentation, note duration, note onset, and note energy, respectively. The models are of the following form:

$$M1(FeatureSet) \rightarrow Ornamentation$$

$$M2(FeatureSet) \rightarrow Duration$$

$$M3(FeatureSet) \rightarrow Onset$$

$$M4(FeatureSet) \rightarrow Energy$$

Where $M1, M2, M3$ and $M4$ are functions which take as input the set of features (*FeatureSet*) shown in Table 2, and *Ornamentation*, *Duration*, *Onset* and *Energy* are the set of classes defined above for the corresponding performance actions.

Learning algorithm

We applied Ripper (Cohen, 1995), a rule learner algorithm. This algorithm is an optimized version of the sequential covering technique used to generate rules (e.g. PRISM algorithm by Cendrowska (1987)). The main motivation for applying the Ripper algorithm was that Ripper examines the classes in ascending order, starting with the minority class, which is very convenient in our problem set, as the classes for ornamentation are unbalanced. i.e. the percentage of ornamented notes is considerably lower than the percentage of non ornamented ones. Thus, the covering algorithm approach will try to isolate first the minority class (i.e. the class of *ornamented* notes)

Ripper evaluates the quality of rules using heuristic measures based on coverage (i.e. how much data they cover) and accuracy (i.e. how many mistakes they make). Once a rule is obtained the instances covered by the rule are removed from the data set, and the process iterates to generate a new rule, until no more data set is left. We used the WEKA library implementation of RIPPER (Hall et al., 2009).

Feature Selection

Automatic feature selection is a computational technique for identifying the most relevant features for a particular predictions task. We applied

EPA	Wrapper	Filter
Ornament	Duration (secs) Next duration (beats) Phrase Next Interval Next duration (secs)	duration (beat) Next duration (sec) Next interval Tessitura Narmour$_1$
Duration	Duration (secs) Narmour Duration (beats) Met. Strenght Phrase	Duration (beats) Duration (secs) Chroma Is chord note
Onset	Tempo Duration (secs) Next duration (secs) Prev. duration (secs) Chord Type	Pitch Narmour Tempo
Energy	Pitch Tempo Narmour key Metrical Strength	Pitch Tempo Phrase

Table 6.4: Most relevant features for each performance action obtained by both filter and wrapper feature selection

feature selection to identify the features which are most relevant for predicting the different expressive performance actions studied. We considered feature selection methods based both on the information gain provided by each individual feature (filter feature selection), and based on the accuracy of RIPPER applied to different feature subsets for predicting the EPA (wrapper feature selection). In both cases (i.e. filter and wrapper feature selection) we identified the best 5 features for each of the EPA.

6.2.5 Results

Feature selection

The most relevant feature subsets for each performance action and each feature selection method are shown in Table 6.4.

Dataset	Baseline	Ripper	
Ornament	66.67	69.38	○
Duration	45.03	57.10	○
Onset	36.69	60.53	○
Energy	39.11	52.48	

○ = statistically significant improvement (p<0.05) w.r.t baseline classifier

Table 6.5: Accuracy of models trained with all extracted features

Dataset	Baseline	Ripper
Ornament (filter)	66.67	71.03 ○
Ornament (wrapper)	66.67	70.47 ○
Duration (filter)	45.03	57.00 ○
Duration (wraper)	45.03	58.36 ○
Onset (filter)	36.69	51.75 ○
Onset (wraper)	36.69	53.34 ○
Energy (filter)	39.11	39.44
Energy (wrapper)	39.11	42.81 ○

○ = statistically significant improvement (p<0.05) w.r.t baseline classifier

Table 6.6: Accuracy of models trained with selected features

Model Evaluation

Table 6.5 and Table 6.6 show the accuracy of each performance action model trained with information of all features considered, and trained with selected features only. All results are obtaining using test data not used for training (i.e. using 10 runs of 10-fold cross validation), A statistical significance test (paired T-test with significance value of 0.05) against the majority class classifier was performed to validate the obtained results.

Expressive performance rules

The set of induced expressive performance rules for each performance action is shown bellow. A rule is expressed as

IF (condition) THEN (action)

where action computes a deviation of a parameter EPA.

- Ornamentation rules

  1. *IF duration of note is very long THEN ornament note*
  2. *IF duration of note is long AND note is the final note in a phrase THEN ornament note*
  3. *IF duration of note is long AND next note's duration is long THEN ornament note*
  4. *IF note is the 3rd note in an IP (Narmour) structure AND previous note's duration is not short AND next note's duration is short THEN ornament note*

- Duration rules

  1. *IF note is the final note of a phrase AND the note appears in an IP (Narmour) structure THEN shorten note*
  2. *IF note duration is longer than a dotted half note AND tempo is Medium (90-160 BPM) THEN shorten note*
  3. *IF note duration is less than an eighth note AND note is in a very strong metrical position THEN lengthen note.*

- Onset deviation rules

  1. *IF the note duration is short AND piece is up-tempo ($\geq$ 180 BPM) THEN advance note.*
  2. *IF the duration of the previous note is not short nor long AND the note's metrical strength is very strong THEN advance note.*
  3. *IF the duration of the previous note is short AND piece is up-tempo ($\geq$ 180 BPM) THEN advance note.*
  4. *IF the tempo is medium (90-160 BPM) AND the note is played within a tonic chord AND the next note's duration is not short nor long THEN delay note.*

- Energy deviation rules

  1. *IF the interval with next note is ascending AND the note pitch not high (lower than B3) THEN play piano.*
  2. *IF the interval with next note is descending AND the note pitch is very high (higher than C5) THEN play forte.*
  3. *IF the note is an eight note AND note is the initial note of a phrase THEN play forte*

Performance Action		# Rules	Grant Green TP	Grant Green FP	Musician 1 TP	Musician 1 FP	Musician 2 TP	Musician 2 FP
Orn.	Yes		52.4%	13.3%	60.6%	30.6%	52.5%	22.3%
	No	4	86.7%	47.6%	69.4%	39.4%	77.7%	47.5%
DR	Lengthen	3	50%	9.4%	32%	12.7%	27%	16.5%
	Shorten		51%	2.4%	45.2%	5.2%	31%	8.4%
ER	Forte	3	34.7%	11.7%	17.1%	18.5%	24.4%	18.9%
	Piano		21.3%	5.8%	13.6%	6.9%	15.9%	10.4%
OD	Advance	4	38.6%	6.6%	1.9%	10.8%	7.1%	4.8%
	Delay		49.8%	10.4%	28.8%	36.5%	29.9%	25.2%

Table 6.7: Model performance measured as true/false positives on train data (Grant Green) and test data (Musicians 1 and 2). Abbreviations Orn., DR, ER, and OD correspond to the studied performance actions *ornamentation*, *duration ratio*, *energy ratio*, and *onset deviation*. Similarly TP and FP accounts for *true/false positives*.

6.2.6 Evaluation

In order to assess the degree of performer-specificity of the rules induced from the Grant Green's recordings we have, similarly to Widmer (2003), applied the induced rules to performances of the same pieces performed by two other professional jazz guitar players. The two guitarists recorded the pieces while playing along with prerecorded accompaniment backing tracks, similarly to the Grant Green recording setting. We processed the recordings following the same methodology explained in Section 2.2. In Table 6.7 we summarize the coverage of the rules measured in terms of the *true positive* (TP) and *false positive* (TN) rate, which is the proportion of correctly and incorrectly identified positives, respectively. As seen in the first two rows of the table, no significant degradation on the rule coverage was found for ornamentation prediction, which might be a good indicator for generality the ornamentation rules. However, rules for duration, energy, and onset show a higher level of degradation, which may indicate that these performance actions vary among Grant Green and the other two musicians. Nevertheless, in order to fully validate this results a much larger number of performances should be taken into consideration.

6.2.7 Discussion

As can be seen from the feature selection analysis (Table 6.4), the most influential descriptors for predicting ornamentation in the investigated performance recordings are *duration in beats* and *Duration in seconds*. This may be explained by the fact that it is easier and more natural to ornament longer notes as opposed to shorter ones. In addition to allowing more time to plan the particular ornamentation when playing long notes, it is technically simpler to replace a long note with a sequence of notes than it is for shorter notes. *Duration in seconds* represents the absolute duration of a note, while *duration in beats* represents the relative duration of a note measured in beats. In general, notes with same *duration in beats* values may vary considerable depending on the tempo of the piece to which they belong. Intuitively, it is the duration of a note in seconds which is the most important feature according to what we have discussed above, so the fact that one feature selection method (e.g.filter feature selection) ranked first the duration in beats feature may indicate that the variation in tempo in the pieces in our data-set is not too important to show this fact. Similarly, *next duration in beats* and *next duration in seconds* have been found to be very informative features by the feature selection algorithms. This may be explained as in the case of the *duration in beats* and *duration in seconds* features: notes that are followed by long notes are more likely to be ornamented since it is possible to introduce extra notes by using part of the duration of the following note.

Next interval and *NarNext interval* are other informative features for ornamentation prediction as detected by the feature selection algorithms. The importance of *Next interval* may be interpreted by the fact that notes that are followed by notes forming an interval of more than 1 or 2 semitones may be ornamented by inserting one or more approximation notes. *Phrase* has been also identified as informative. This confirms our intuition that notes in phrase boundaries are more likely to be ornamented. *Nar* is related to the degree of expectation of a note's pitch, so the fact that this feature is among the five most informative features for predicting ornamentation may be due that musicians tend ornament highly expected notes in order to add variation and surprise to the performed a melody. This is interesting because according to Narmour's theory this expectations are innate in humans so it may be the case that the choice to ornament expected/unexpected notes can be the results of an intuitive and unconscious process.

As expected, the most informative features for predicting ornamentation include both temporal (e.g. *Duration in seconds* and *Duration in beats*) and melodic features (e.g. *Next interval* and *Nar*). They involve not only properties of the note considered, but also properties that refer to its musical context, i.e. its neighbouring notes (e.g. *Next duration*, *Next interval*, *Phrase* and *Nar*). Similar results were obtained for the other expressive performance actions (i.e. duration, onset and energy variations): Temporal features of the note considered and its context (e.g. *Duration in seconds*, *Duration in beats*, *Next duration* and *Prev duration*) are found to be informative, as well as melodic features (e.g. (e.g. *Pitch*, *Next interval* and *Nar*). Interestingly, *Pitch* was found to be the most informative feature for energy prediction. This may be explained by the tendency of the performer to play higher pitch notes softer than lower pitch ones. *Metrical strength* was found to be informative for duration variation prediction which seems intuitive since the note's duration is often used to emphasize the metrical strength or weakness of notes in a melody.

The difference between the results obtained (see Table 6.5) and the accuracy of a baseline classifier, i.e. a classifier guessing at random, indicates that the audio recordings contain sufficient information to distinguish among the different classes defined for the 4 performance actions studied, and that the machine learning method applied is capable of learning the performance patterns that distinguish these classes. It is worth noting that almost every model produced significantly better than random classification accuracies. This supports our statement about the feasibility of training classifiers for the data reported. However, note that this does not necessary imply that it is feasible to train classifiers for arbitrary recordings or performer.

The results also indicate that certain tasks proved to be more difficult to discriminate than others: The ornamentation model was found to be the most accurate, 78.8% trained with all features and 70.1% trained after feature selection (wrapper method). The onset model accuracy was found to be 60.5% when trained with all features and 63% when trained with selected features (wrapper method). The least accurate models and thus the most difficult to predict were the models for duration (51% and 56.1% for all features and selected features training, respectively) and energy variation (52.4% and 52.2% for all features and selected features training, respectively).

The accuracy of all models except the energy variation model improved after performing feature selection. In all cases wrapper feature selection resulted in better accuracies than filter feature selection. The improvement found with feature selection is marginal in most cases. However, this shows that it suffices to take into account a small subset of features (i.e. 5 or less features) in order to be able to predict with similar accuracy the performance actions investigated. The selected features contain indeed sufficient information to distinguish among the different classes defined for the 4 performance actions studied.

The expressive performance models induced consist of sets of conjunctive propositional rules which define a classifier for the performance actions, i.e. ornamentation, and duration, onset and energy deviation. These rules capture general patterns for classifying the musician's expressive decisions during performance.

The first ornamentation rule (i.e. (*IF duration of note is very long THEN ornament note*) specifies that if a note's duration is very long (i.e. longer than 1.6 seconds) then it is predicted as ornamented with a probability of 0.79. The precondition of this rule is fulfilled by 111 notes in the data set from which 88 are actually ornamented and 23 are not. This rule makes musical sense since long notes are likely to be ornamented. The second ornamentation rule (Rule O2) is similar in spirit, it specifies that if a note's duration is long (i.e. longer than 1 second) and this note is the ending note of a musical phrase, then it is predicted as ornamented with a probability of 0.74. Thus, this rule relaxes the constraint on the duration of the note but requires that the note appears at the end of a phrase in order to classify it as ornamented. The rule captures the intuition that phrase boundary notes (in this case notes at the ending of a phrase) are more likely to be ornamented. Rule O3 and Rule O4 add conditions about the duration of neighbouring notes (i.e. next and previous notes) in order to classify notes as ornamented. The intuition of these rules is that notes may be ornamented by using part of the duration of the neighbouring notes.

The rules about duration and onset transformations involve conditions that refer to note duration, metrical strength, and tempo. Long notes in medium tempo pieces are likely to be shortened (Rule D2), while short notes appearing in strong metrical positions are lengthened (Rule D3). The first onset rule (Rule T1) states that short notes in up-tempo pieces likely to be advanced, while Rule T2 constrains the first rule stating to advance notes that occur within a sequence of short notes. On the other hand, a note is delayed if it belongs to a medium tempo (i.e. 90-160 BPM) piece and it is played within a tonic chord and succeeded by a medium length note (Rule T4). Finally, energy deviation rules contain conditions that refers to the direction of the interval with respect to the next note. Rule E1 states that notes occurring in a low pitch register and in an ascending interval are played softer, whereas notes coming from higher pitch registers and in a descending intervals are played forte (Rule E2). Rule E3 states that a note occurring at the beginning of a phrase is accentuated by playing it forte.

The duration and energy rules induced in this section were compared with the rules obtained by Widmer (2003, 2002) (applying machine learning techniques to a data set of 13 performances of Mozart piano sonatas) as well as with the rules obtained by Friberg et al. (2006) (using an analysis by synthesis approach). Duration rule D3 is consistent with Widmer's TL2 rule "*Lengthen a note if it is followed by a substantially longer note*", which may imply that the note in consideration is short. However it contradicts its complementary condition TL2a ("*Lengthen a note if it is followed by a longer note and if it is in a metrically weak position*"). This might be due to the fact that note accentuation in jazz differ considerably from note accentuation in a classical music context, e.g. in case of swinging quavers, the first quaver (stronger metrical position) is usually lengthen. This however, is consistent with Friberg's *inégales* rule ("*Introduce long-short patterns for equal note values (swing)*"). Duration rule D2 can be compared with Widmer's rule TS2 ("*Shorten a note in fast pieces if the duration ratio between previous note and current note is larger than 2:1, the current note is at most a sixteen note, and it is followed by a longer note*). Similarly, duration rule D2 and D3 are consistent with Firdberg's *Duration-contrast* ("*Shorten relatively short notes and lengthen relatively long notes*"), as dotted half notes can be considered relatively long notes, and eight notes can be considered as relatively short notes. The rules take as preconditions the duration of the note and the tempo of the piece. Energy rules E1 and E2 are consistent with Friberg's *high-loud* ("*Increase sound level in proportion to pitch height*") and *phrase-arch* (*Create arch-like tempo and sound level changes over phrases*") rules, as notes in an ascending context might be played softer and vice-versa. However energy rule E3 contradicts *phrase-arch* rule. Energy rule E2 shares the interval condition of the next note of Widmer's DL2 rule ("*Stress a note by playing it louder if it forms the apex of an up-down melodic contour and is preceded by an upward leap larger than a minor third*"). In addition, Widmer's rules for attenuating dynamics of notes (play softer) and our energy rules share the fact that the rule preconditions include intervals with respect to neighbour notes.

All in all there are similarities between the rules induced in this section and the rules reported in the literature. However, at the same time, there are differences and even opposite findings, fact that is expected given the different data sets considered in the studies. While there seems to be similarities in expressive patterns in both classical and jazz music, clearly, both traditions have their own peculiarities and thus it is expected to find different/contradictory rules.

It has to be noted that the obtained expressive rules are specific to the studied guitarist and in particular to the considered recordings. Thus, the rules are by no means general rules of expressive performance in jazz guitar. Nevertheless, the induced rules are of interest since Grant Green is a musician recognized for his expressive performance style of jazz guitar.

CHAPTER **7**

Applications

In this chapter we present two main applications of our machine learnign approach for expressive music performance modelling. Firstly, and application consisting of a real-time systems for music neuro-feedback that allow users to control expressive parameters of a musical piece based on its detected emotional state. The system is later utilized in a pilot clinical study to treat depression in elderly people by combining music therapy, music neuro-feedbak, and emotional state recognition based on EEG. Secondly, an approach for ensemble expressive music performance analysis of a jazz quartet recording (guitar, bass, drums and piano), in which we make use of our modelling framework to extract models from the guitar and extend our methodology to extract and analyse performance data from the piano. We applied machine learning techniques to train models for each performance action, considering both solo and ensemble descriptors. The models' accuracy improvement when ensemble information was considered, might be explained by the interaction between musicians.

7.1 Emotional modelling for neuro-feedback

Active music listening has emerged as a study field that aims to enable listeners to interactively control music. Most of active music listening systems aim to control music aspects such as playback, equalization, browsing, and retrieval, but few of them aim to control expressive aspects of music to convey emotions. In this section our aim is to allow listeners to control expressive parameters in music performances using their perceived emotional state, as detected from their brain activity. We obtain electroencephalogram

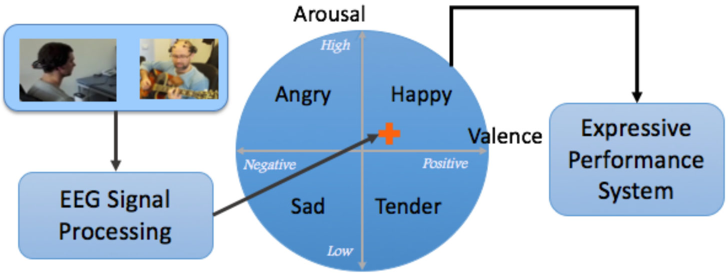


Figure 7.1: Theoretical frame work for expressive music control based on EEG arousal - valence detection.

(EEG) data using a low-cost EEG device and then map this information into a coordinate in the emotional arousal-valence plane. On the other hand, we train expressive performance models using linear regression for four different moods: happy, sad, angry and tender. We then interpolate these models in order to to obtain intermediate expressive models for other emotions. We apply the resulting models to generate performances with emotional content. Such implementation allows the generation of performances with mood transitions in real-time, by manipulating a cursor over the arousal valence plane. Thus the resulting coordinate obtained from the brain activity is used to apply expressive transformations to music performances in real time by mapping the output arousal valence coordinate obtained from the brain activity into the input coordinate of the expressive performance modelling system.

7.1.1 Brain activity to mood estimation

Our proposed approach to real-time EEG- based emotional expressive performance control is depicted in Figure 7.1. First, we detect EEG activity using the Emotiv Epoch headset (Researchers, 2012) . We base the emotion detection on the approach by Ramirez and Vamvakousis (2012). We measure the EEG signal using electrodes AF_3, AF_4, F_3, and F_4, which are located on the pre-frontal cortex. We use these electrodes because it has been found that the pre-frontal lobe regulates emotion and deals with conscious experience.

We model emotion using the arousal- valence plane, a two dimensional emotion model which proposes that affective states arise from two neurological systems: arousal related to activation and deactivation, and valence related to pleasure and displeasure. We are interested in characterizing four different emotions: happiness, anger, relaxation, and sadness. As depicted in Figure 7.1, each studied emotion belongs to a different quadrant in the arousal valence plane: happiness is characterized by high arousal and high valence, anger by high arousal and low valence, relaxation by low arousal and high valence, and sadness by low arousal and low valence.

Signal reprocessing

Alpha and Beta waves are the most often used frequency bands for emotion detection. Alpha waves are dominant in relaxed awake states of mind. Conversely Beta waves are used as an indicator of excited mind states. Thus, the first step in the signal preprocessing is to use a band pass filter in order to split up the signal in order to get the frequencies of interest, which are in the range of 8-12 Hz for alpha waves, and 12-30 Hz for beta waves. After filtering the signal we calculate the power of each alpha and beta bands using the logarithmic power representation proposed by Aspiras and Asari (2011). The power of each frequency band is computed by:

$$LP_f = 1 + log(\frac{1}{N}\sum_{n=1}^{N}(x_{nf})^2) \tag{7.1}$$

where is the magnitude of the frequency band f (alpha or beta), and N is the number of samples inside a certain window. Hence, we are computing the mean of the power of a group of N samples in a window and then compressing it by calculating the logarithm of the summation.

Arousal and valence calculation

After the band power calculation, the arousal value is computed from the beta/alpha ratio. Valence is calculated based on the asymmetric frontal activity hypothesis, where left frontal inactivation is linked to a negative emotion, whereas right frontal inactivation may be associated to positive emotions. Thus arousal and valence are calculated as follows:

$$arousal = \frac{b_{F3} + b_{F4}}{a_{F3} + a_{F4}} \tag{7.2}$$

$$valence = \frac{a_{F4}}{b_{F4}} - \frac{a_{F3}}{b_{F3}} \tag{7.3}$$

The values obtained for arousal and valence are calculated using sliding windows over the signal in order to smooth the signal. It is worth noting that there are not absolute levels for the maximum and the minimum values for both arousal and valence, as these values may differ from subject to subject, and also vary over time for the same subject. To overcome this problem we computed the mean of the last five seconds of a 20 second window and normalize the values by the maximum and minimum of these 20 second window. This way we obtain values that range between minus one and one. We consider a window size of 4 second with a hop size of 1 second.

Arrousal-Valence Accuracy Experiments

Two types of experiments were performed: a first one listening while sitting down and motionless and the other listening while playing (improvising) with a musical instrument. In both the aim was to evaluate whether the intended expression of the synthesized music corresponds to the emotional state of the user as characterized by his/her EEG signal. In both experiments subjects sat down in a comfortable chair facing two speakers. Subjects were asked to change their emotional state (from relaxed/sad to aroused/happy and vice versa). Each trial lasted 30 seconds with 10 seconds between trials. In experiment one the valence is set to a fixed value and the user tries to control the performance only by changing the arousal level. In experiment 2 the expression of the performance is dynamically changed between two extreme values (happy and sad), while the user is improvising playing a musical instrument. A 2-class classification task is performed for both experiments.

Two classifiers, Linear Discriminant Analysis and Support Vector Machines, are evaluated to classify the intended emotions, using 10 cross-fold validation. Initial results are obtained using the LDA and SVM implementations of the OpenVibe library (OpenVibe, 2010). Our aim was to quantify in which degree a classifier was able to separate the two intended emotions from the arousal/valence recorded data. For high-versus-low arousal classification we obtained a 77.23% for active listening without playing, and 65.86% for active listening when playing an instrument (improvising) along the synthesized expressive track, using SVM with radial basis kernel function. Results were obtained using 10-fold cross validation.

In Figure 7.2 is depicted one trial of the first experiment. The EEG signal of a subject and the calculated arousal are shown for the first, second, third

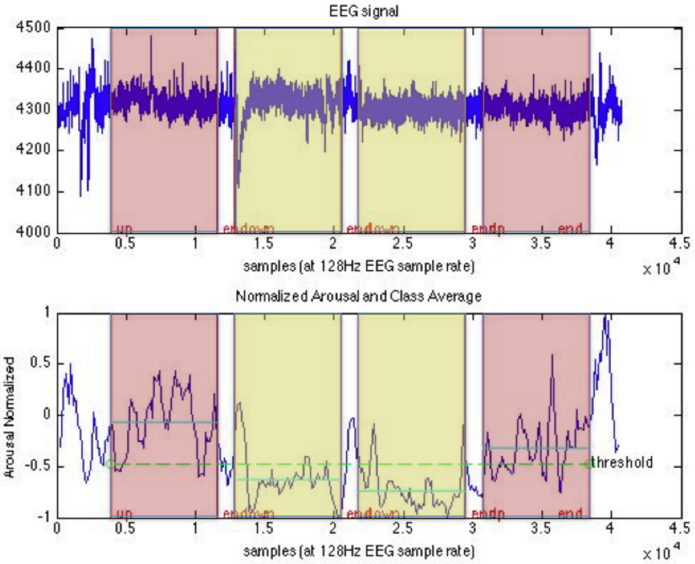


Figure 7.2: A subject's EEG signal (top) and calculated arousal (bottom). Vertical lines delimit each sub-trial for high arousal (1st and 4th sub-trials) and low arousal (2nd and 3rd sub-trials). Horizontal line represents the average of each class segment.

and fourth sub-trials. First and fourth sub-trials corresponds to time slots in which the user was asked to change his emotional state to high arousal. Second and third sub-trials correspond to low arousal, accordingly. In the figure, the average of the high arousal level sub-trials sections are noticeably higher than low-level ones.

7.1.2 Expressive performance system for real-time emotion modelling

In this section we explain the expressive performance system for real-time emotion modelling. The general framework of the modelling/synthesis system is depicted in Figure 7.3. Four moods were considered for this study (*happy, sad, angry, tender*) each corresponding to a quadrant in the arousal-valence plane. On the other hand three performance actions were taken in to consideration (*duration, energy and articulation*. A total of twelve Linear regression models were trained (three performance actions per four moods) using recordings of musical pieces performed by a professional guitarist in

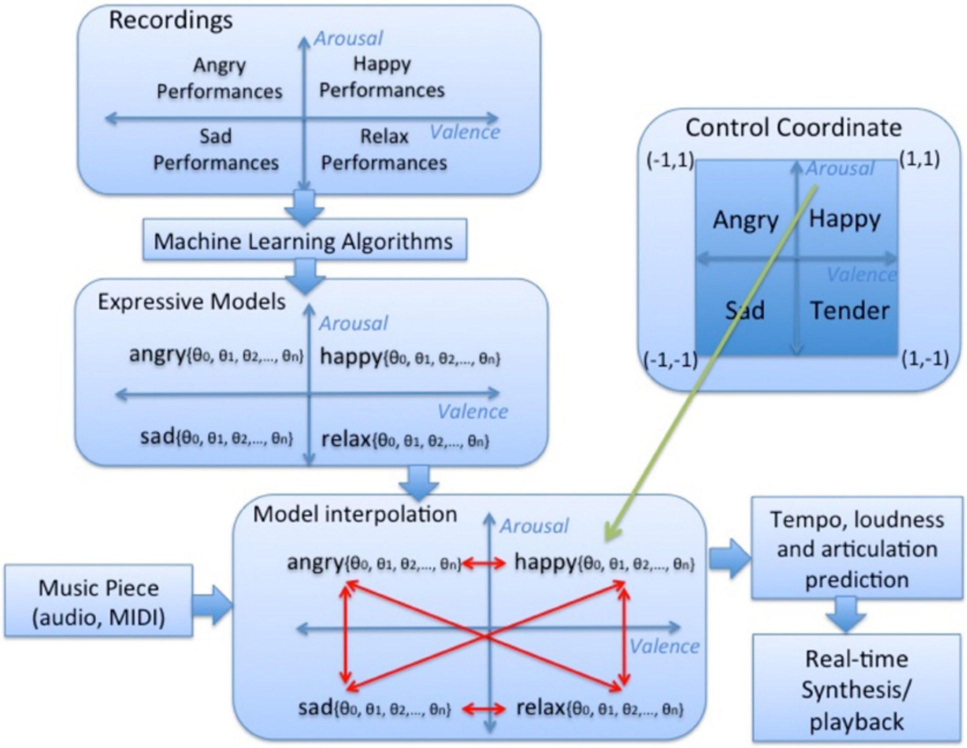


Figure 7.3: Theoretical frame work for expressive music control based on EEG arousal - valence detection.

each of the four moods. The input of the system is a control coordinate on the arousal valence plane. The coefficients of the linear models are interpolated along the arousal valence plane to obtain the prediction of the performance actions, which serve as input for mood modelling-synthesis system.

Data acquisition

The first step was to obtain a musicXML representation of the score of a set of pieces, which was performed as explained in Section 3.2.1.

On the other hand, we obtained recordings of performances of the pieces in four different emotions by 4 professional guitarists. The guitarists were instructed to play the same tune on each of the moods: happy, sad, angry and tender. Each performer could freely choose to play the tempo, energy, and articulation variations they considered better express each mood. Guitarists were also instructed not to ornament notes and not to perform pitch variations (i.e. play only the notes of the score). The recordings were obtained without metronome, to permit tempo variations in the performances. The audio signal of the recordings was processed as explained in Section 3.3, to obtain a machine readable representation of the performances.

Expressive performance actions

Expressive performance actions were defined in Section 4.2 as the strategies and changes introduced in a performance, which are not specified by the score. In this section we will focus on three specific performance actions:

- Duration ratio: Ratio between the score duration and the performed duration (see Table 4.3)
- Energy ratio: Ratio between loudness of a note and average loudness (see Table 4.3
- Articulation gap: Measures the level of staccato - legato, i.e. time interval (in seconds) between the offset of a note and the onset of the next note.

Performance actions were calculated for each note in the score-performance. Notice that for this application the note to performance alignment was one to one note correspondence, as performer were instructed not to include ornamentations.

Feature Extraction

Each note in the training data is annotated with a number of attributes representing both properties of the note itself (Intra-note features) and some aspects of the context in which the note appears (Inter-note features). Information about the note includes note duration, energy, pitch, while information about its context includes relative pitch and duration of neighbour notes, as well as, melodic and harmonic analysis. A complete list of the features and its description can be found on Table 4.2 .

Each note in the training data is annotated with a number of attributes as described in Section 4.1 representing nominal properties of the note as well as local and global contextual information of the note (see Table 4.2). Nominal descriptors were discretized so that each categorical value was encoded as a binary attribute.

Machine learning modelling

Linear regression models were trained to predict each performance action for each of the four moods considered. A total of twelve models were generated, expressed as a linear product between the set of features and trained *weights* with the following form:

- Models for happy mood

$$h\theta(\vec{x_n})_{hDR} = x_0\theta_{(hDR)0} + x_1\theta_{(hDR)1} + ... + x_n\theta_{(hDR)n} \tag{7.4}$$

$$h\theta(\vec{x_n})_{hER} = x_0\theta_{(hER)0} + x_1\theta_{(hER)1} + ... + x_n\theta_{(hER)n} \tag{7.5}$$

$$h\theta(\vec{x_n})_{hAG} = x_0\theta_{(hAG)0} + x_1\theta_{(hAG)1} + ... + x_n\theta_{(hAG)n} \tag{7.6}$$

- Models for sad mood

$$h\theta(\vec{x_n}))_{sDR} = x_0\theta_{(sDR)0} + x_1\theta_{(sDR)1} + ... + x_n\theta_{(sDR)n} \tag{7.7}$$

$$h\theta(\vec{x_n})_{sER} = x_0\theta_{(sER)0} + x_1\theta_{(sER)1} + ... + x_n\theta_{(sER)n} \tag{7.8}$$

$$h\theta(\vec{x_n})_{sAG} = x_0\theta_{(sAG)0} + x_1\theta_{(sAG)1} + ... + x_n\theta_{(sAG)n} \tag{7.9}$$

- Models for angry mood

$$h\theta(\vec{x_n})_{aDR} = x_0\theta_{(aDR)0} + x_1\theta_{(aDR)1} + ... + x_n\theta_{(aDR)n} \tag{7.10}$$

$$h\theta(\vec{x_n})_{aER} = x_0\theta_{(aER)0} + x_1\theta_{(aER)1} + ... + x_n\theta_{(aER)n} \tag{7.11}$$

$$h\theta(\vec{x_n})_{aAG} = x_0\theta_{(aAG)0} + x_1\theta_{(aAG)1} + ... + x_n\theta_{(aAG)n} \tag{7.12}$$

- Models for tender mood

$$h\theta(\vec{x_n})_{tDR} = x_0\theta_{(tDR)0} + x_1\theta_{(tDR)1} + ... + x_n\theta_{(tDR)n} \tag{7.13}$$

$$h\theta(\vec{x_n})_{tER} = x_0\theta_{(tER)0} + x_1\theta_{(tER)1} + ... + x_n\theta_{(tER)n} \tag{7.14}$$

$$h\theta(\vec{x_n})_{tAG} = x_0\theta_{(tAG)0} + x_1\theta_{(tAG)1} + ... + x_n\theta_{(tAG)n} \tag{7.15}$$

Where $h\theta$ are functions which take as input the set of features $\vec{x_n}$ (extracted from the scores) of the note n being played, and θ are the set of weights trained to predict each of the performance actions defined above. Each group of models were trained with the corresponding *happy, sad, angry,* and *tender* performances, respectively. Sub-indices conventions *h,s,a*, and *t* are abbreviations for *happy, sad, angry,* and *tender*, respectively. Similarly *DR, ER,* and *AR* are abreviations for *duration ratio*, *energy ratio*, and *articulation gap* (e.g. sub-index *hDR* refers to *Articulaton Ratio* at *happy* mood context).

Feature selection was performed using wrapper method with "best first" as a search method and with forward and backward elimination. Ten cross-fold validation was used for testing and evaluating the models. For each model we obtained a list of features that were most frequently selected by the feature selection method over the ten folds. For interpolation purposes we selected the same features for the four emotion models for each performance action, selecting the average of the most relevant features for the four models.

Interpolation

A coordinate $c(v_i, a_i)$ define 4 weighting areas on the arousal plane as depicted in Figure 7.4. The four emotions considered are defined at each quadrant of the arousal valence space. Given the coordinate, the amount of each of the four emotions present conforming the emotional state are defined by a weight proportional to the areas of the square which is formed in the opposite corner. Thus, the emotional state defined by the coordinate will consist of a proportion of each of the four emotion as follows:

$$h_{\%}(v_i, a_i) = \frac{A_3}{A_1 + A_2 + A_3 + A_4} \tag{7.16}$$

$$a_{\%}(v_i, a_i) = \frac{A_4}{A_1 + A_2 + A_3 + A_4} \tag{7.17}$$

$$s_{\%}(v_i, a_i) = \frac{A_1}{A_1 + A_2 + A_3 + A_4} \tag{7.18}$$

$$t_{\%}(v_i, a_i) = \frac{A_2}{A_1 + A_2 + A_3 + A_4} \tag{7.19}$$

The percentage of each emotion calculated is applied to each the pre-trained model (duration, energy, and articulation ratios) as follows:

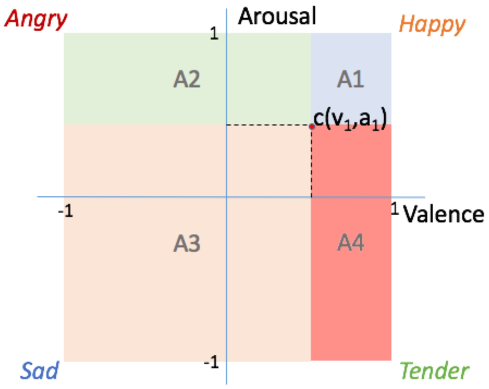


Figure 7.4: Emotion weighting areas defined by a coordinate $c(v_i, a_i)$ on the arousal-valence plane. Each emotion is proportional to the area formed on the opposite corner of the emotion defined at each quadrant.

$$\begin{aligned} M_{DR}(\vec{x_n}, v_i, a_i) = {} & h_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{hDR} \\ & + s_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{sDR} \\ & + a_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{aDR} \\ & + t_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{tDR} \end{aligned} \tag{7.20}$$

$$\begin{aligned} M_{ER}(\vec{x_n}, v_i, a_{)} = {} & h_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{hDR} \\ & + s_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{sDR} \\ & + a_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{aDR} \\ & + t_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{tDR} \end{aligned} \tag{7.21}$$

$$\begin{aligned} M_{AG}(\vec{x_n}, v_i, a_{)} = {} & h_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{hDR} \\ & + s_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{sDR} \\ & + a_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{aDR} \\ & + t_{\%}(v_i, a_i) * h\theta(\vec{x_n})_{tDR} \end{aligned} \tag{7.22}$$

Where M_{DR}, M_{ER}, and M_{AG} is are functions that take as input the set of features $\vec{x_n}$ of the n^{th} note being played, and a coordinate (v_i, a_i) on the arousal valence plane to apply the transformations in duration, energy and articulation to the note.

	Angry	Tender	Happy	Sad
DR	0.75	0.05	0.54	0.45
AG	0.34	0.37	0.42	0.43
ER	0.27	0.28	0.37	0.24

Table 7.1: Correlation Coeficients for Duration Ratio (DR), Articulation Gap (AG) and Energy Ratio (ER), with linear regression.

Synthesis

The linear regression modelling system was implemented in pure data. Note information and its descriptors were saved on a text file. Each time a note is read from the text file, the performance actions are calculated using the model and the interpolated coefficients based on a control position on a arousal valence plane. Thus, duration, energy (velocity midi control) and articulation (duration + time delay to the next note) are calculated.

Evaluation

In Table 7.1 the correlation coefficients for each model are shown. Variability in accuracy indicate that some combinations of performance actions-moods are more difficult to be captured by the models (e.g. duration ratio for tender mood and angry mood). Results are not surprising though, given the fact that human performers might have clearer performance strategies to convey certain moods better than others. For example, during the recording sessions, performers manifest to have a clearer idea towards an angry/sad performance, rather than tender/happy one.

7.1.3 Applications in neuro-feedback

The potential benefits of combining music therapy, neurofeedback and emotion detection for treating depression in elderly people was studied and reported in Ramirez et al. (2015), by introducing our music neurofeedback approach, allowing users to manipulate expressive performance parameters in music performances using their emotional state. A pilot clinical experiment was conducted at a residential home for the elderly in Barcelona, involving 10 participants (9 female, and 1 male, mean = 84, SD = 5.8), consisting of 10 sessions (2 per week) of 15 minutes each. On each session the participant was asked to sit comfortable in a chair facing two speakers, close their eyes and not to move (to avoid artefacts on the EEG signal). Wile listening to pre-selected music pieces, participants were encouraged to increase the loudness and the tempo of the pieces, which were mapped to the calculated levels of arousal and valence obtained from their brain activity. As explained in Section 7.1.1, arousal was computed as the ration between beta to alpha activity in the frontal cortex, whereas valence was computed as the relative frontal alpha activity of the right lobe compared to the left lobe. The coordinate in the arousal valence plane, obtained from the user's EEG activity is mapped to the pre-trained expressive performance system (Section 7.1.2) to apply the expressive transformations on the musical pieces being played on real-time. Pre and post evaluation was performed using the BDI (Beck Depression Inventory) test, which showed an average improvement on the depression condition of the participants. Moreover, the EEG data showed an statistically significant increase of the overall valence level at the end of the treatment compared to the starting level. This result may be interpreted as a decrease of the alpha activity in the frontal lobe which may indicate (as well) an improvement in the depression condition.

7.2 Performance interaction analysis in jazz

In previous sections, several examples of systems for expressive music performance have been presented. In Chapter 2 we have presented a review of CSEMPs in which there was a clear prevalence of systems targeted for classical piano music (see Table 2.1). Some exceptions included studies in jazz saxophone music (Ramírez and Hazan (2006), Arcos et al. (1998), Grachten (2006). Throughout this dissertation we have investigated in computational systems for music expressive performance in jazz guitar. In Section 2.2.6 we review some previous studies on ensemble performance which has been conducted in classical context (e.g. Marchini (2014); Sundberg et al. (1989). However, to our knowledge, few work has been done in the analysis of ensemble expressive performance in jazz context. In this Section we present an approach to study the interaction between performers on a jazz quartet (guitar and piano). We extend our modelling approach for guitar to extract *horizontal descriptors* for melody and chords, i.e. characterize each note from the melody as well as each chord from the scores. We will also calculate vertical descriptors which combine information from both melody and chords. Performance actions are calculated for guitar (melody) and piano (chords) respectively. After, we train computational models to predict the measured performance actions for guitar and piano, and compare the performance of the models between solo (considering only horizontal) and ensemble (considering horizontal and vertical) descriptors. Interaction between musicians is measured based on the improvement when ensemble information is considered.

This part of the research was done as part of the B. S. thesis by Bantula (2015) under the supervision of the author of this dissertation. This work was reported in Bantula et al. (2016).

7.2.1 Data acquisition

The music material used on this experiment consisted on the recording of seven jazz standard pieces performed by a jazz quartet (electric guitar, piano, electric bass, and drums) in a professional recording studio. The audio was captured separately for each instrument. Piano MIDI data was recorded simultaneously with audio from the standard MIDI out of the keyboard. Guitar performance was acquired in monophonic-monotimbral audio, as the guitarist was instructed to play the melodies of the pieces with out including chords or multiple notes at a time. On the other hand the pianist (as well as the other instruments) was instructed to accompany the performance of the guitarist, following the same score (which included chord information). For this study only guitar (melody) and piano (chords) data was taken in consideration. The scores were obtained from the The real book.

7.2.2 Data analysis

In this section we explain the data analysis for both guitar and piano, which consisted on the extraction of descriptors from the score of the pieces to obtain melodic information (performed by the guitar) and chord information (performed by the piano), as well as the calculation of the performance actions for both guitar and piano. Descriptors from the score were extracted to characterize every chord and every note from its horizontal and vertical contexts.

Score melodic data analysis

Melodic data analysis was performed following the methodology explained in Chapter 4, which consisted on the characterization of each note by a set of descriptors (see Section 4.1). In Figure 7.5 we illustrate an example of a note characterization: note description includes information of the note itself (*reference note*), information of its *neighbour* notes (*horizontal context of reference note*) and harmonic information calculated from the chord in

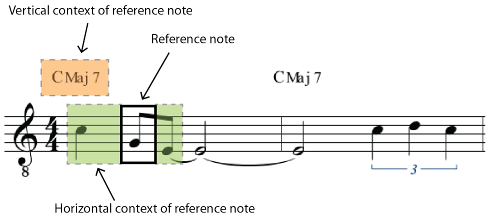

Vertical context of reference note
Reference note
CMaj 7
CMaj 7
3
Horizontal context of reference note

Figure 7.5: Excerpt of *All Of Me*: Horizontal and Vertical contexts for a reference note

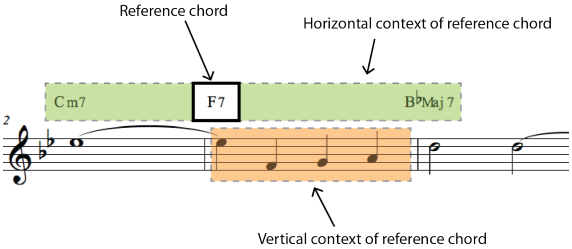

Reference chord
Horizontal context of reference chord
Cm7
F7
B♭Maj 7
2
Vertical context of reference chord

Figure 7.6: Excerpt of *Autumn Leaves*: horizontal and vertical contexts for the reference chord F7

which the note occurs (*vertical context of reference note*). Reference and horizontal descriptos were calculated according to the definition of *nominal* and *neighbour* descriptors of Table 4.2, whereas vertical descriptors were calculated as indicated in Table 7.6.

Score chord data analysis

A similar approach was used to perform the piano-chord data analysis. Chord descriptors (see Figure 7.6) included information of the chord itself (*refence chord*), information from the neighbour chords (*horizontal context of reference chord*), and averaged information of the notes occurring within the chord (*vertical context of reference chord*). Reference, horizontal, and vertical chord descriptors are shown in Tables 7.2, 7.4 and 7.3, respectively.

Guitar performance analysis

For guitar performance analysis, we followed the same approach as explained in Section 4.2, in which we first obtain a machine readable representation from the audio of the performance, by applying automatic transcription techniques. After, we aligned the performance to the score by applying

descriptor	units	computation	range
id	num	$root \rightarrow number$	[0,11]
type	label	*type*	{M, m, +,7, dim, half_dim}
tens	label	tension based on musical criteria	{++, +, -, - -}
C_dur_b	beats	*C_dur_b*	[1,∞)
on_b	beats	*on_b*	[1,∞)

Table 7.2: Individual descriptors for a reference chord (no context).

descriptor	units	computation	range
onset_b	beats	$min(onset_{notes})$	[1,∞)
dur_b	beats	$max(onset_{notes})$ $+max(dur_{notes})$ $-min(onset_{notes})$	[1,∞)
meanPitch (mP)	MIDI note	$mean(pitch_{notes})$	[36,96]
onset_s	seconds	$60 * \frac{onset\_b}{tempo}$	[1,∞)
dur_s	seconds	$60 * \frac{dur\_b}{tempo}$	[1,∞)
chroma	half tones	$mod_{12}(mP)$	[0,11]
measure	num	$measure$	[1,∞)
pre_dur_b	beats	pre_dur_b	[1,∞)
pre_dur_s	seconds	$60 * \frac{pre\_dur\_b}{tempo}$	[1,∞)
nxt_dur_b	beats	nxt_dur_b	[1,∞)
nxt_dur_s	seconds	$60 * \frac{nxt\_dur\_b}{tempo}$	[1,∞)
prev_int	half tones	$prev_{mP} - mP$	[1,∞)
next_int	half tones	$mP - next_{mP}$	[1,∞)
note2key	half tones	$chroma - key$	[0,11]
note2chord	half tones	$chroma - id$	[0,11]
isChordN*	label	-	{y,n}
mtr*	label	$mean(met_{pos}(notes))$	{strong, weak}
intHop*	num	$mean(intervals)$	[0,96]
melody*	num	$\frac{\#notes}{chord\_dur}$	-

Table 7.3: Chord melodic descriptors (vertical)

descriptor	**units**	**computation**	**range**
tempo	bpm	*tempo*	[1,300]
keyMode	label	*keyMode*	{major, minor}
numKey	num	key position in the Fifths Circle	[0,11]
keyDistance	half tones	$id - numKey$	[0,11]
metP*	label	metrical position	{strongest, strong, weak, weakest}
function	label	harmonic analysis from *keyDistance*	{tonic, subdom, dom, no_func}
next_root_int	half tones	$id - next_{id}$	[0,11]
prev_root_int	half tones	$prev_{id} - id$	[0,11]

Table 7.4: Chord harmonic descriptors (horizontal)

dynamic time warping techniques (see Section 4.2.1), and calculated performance actions as described in Table 4.3. Finally, we constructed a data base with the notes description along with the calculated performance actions, as explained in Section 4.2.3. Notice that for this study only *ornamentation* was considered, and other performance actions were not taken in consideration.

Piano performance analysis

We detected chords in the piano data by grouping notes played at the same time using heuristic rules based on the work done by Traube and Bernays (2012), who identify groups of consecutive notes which have *near-synchronous onsets*. Our approach consisted of three rules: the first one searched and grouped notes which were played at the same time. The second one, merged chords with an inter onset difference $< 100ms$. Finally, the third rule was designed to parse pedal notes (i.e. notes sustained through several chord changes).

Alignment was performed to link the detected chords with the score chords on a beat level basis: onsets/offsets of each performed chord were converted from seconds to beats (based on beat tracking information). Later, each score chord was aligned with the performed chord (or group of chords). Based on the alignment, performance actions for every score chord three performance actions were calculated for piano performance:

- **density**, defined as *low* or *high* depending on the number of chords used within a chord indicated in the score, as follows:

$$den(chords_S) = \begin{cases} low & \text{if } \dfrac{\sum chords_P}{dur(chords_S)} < 1/2 \\ \\ high & \text{if } \dfrac{\sum chords_P}{dur(chords_S)} \geq 1/2 \end{cases} \tag{7.23}$$

  where $chords_S$ is the corresponding chord on the score, $\sum chords_P$ is the amount of performed chords for a chord on the score, and $dur(chords_S)$ is the duration of the corresponding chord on the score

- **weight**, defined as *low* or *high* according to the total number of notes which were utilized to perform a score chord, as follows:

$$wei(chords_S) = \begin{cases} low & \text{if } \dfrac{\sum notes_P}{\sum chords_P} < 4 \\ \\ high & \text{if } \dfrac{\sum notes_P}{\sum chords_P} \geq 4 \end{cases} \tag{7.24}$$

  where, $chords_S$ is the corresponding chord on the score, $\sum notes_P$ is the total number of performed notes for a chord on the score, and $\sum chords_P$ is the amount of performed chords for a chord on the score

- **amplitude**, defined as *low* or *high* if the distance in semitones from the highest to the lowest performed note per chord in the score was larger than 18 (an octave and a half), as follows.

$$amp(chords_S) = \begin{cases} low & \text{if } max(pitch_{PN}) - min(pitch_{PN}) < 18 \\ \\ high & \text{if } max(pitch_{PN}) - min(pitch_{PN}) \geq 18 \end{cases} \tag{7.25}$$

where $chord_S$ is the corresponding chord on the score and $pitch_{PN}$ is the vector of pitch of the performed notes (PN) for a chord on the score.

Beat detection data

Recordings were performed without metronome, therefore, the tempo varied during the performance and beat positions were not equidistant. Beat tracking was performed over the audio mix using the approach by Zapata et al. (2012) to obtain a beat grid to set the onset and duration in beats. Later, the average tempo of each song was computed as,

$$tempo = round\left(\frac{60}{mean(diff(beats))}\right) \tag{7.26}$$

where $beats$ is the vector of tracked beats.

7.2.3 Machine learning

Datasets

Machine Learning models were trained to predict the aforementioned performance actions for both piano and guitar from the datasets of descriptors of chords and notes obtained from the scores. Hence we constructed three types of datasets, as shown in Figure 7.7. Training datasets combining horizontal and vertical descriptors were created as follows:

Simple Datasets (D1): Horizontal score context. Contains individual descriptors of the chords/notes.

Score Mixed Datasets (D2): D1 plus vertical **score** context. Contains merged descriptors of chords and notes.

Performance Mixed Datasets (D3): D1 plus vertical **performance** context (extracted from the manual transcriptions of the performances). Contained merged features of chords and notes. This data set aims to measure the interaction among musicians.

Piano models were trained as follows:

$$f(Chord) \rightarrow (Den, Wei, Amp) \tag{7.27}$$

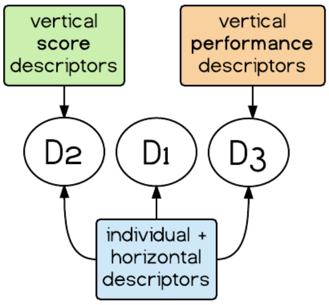

vertical score descriptors
vertical performance descriptors
D2
D1
D3
individual + horizontal descriptors

Figure 7.7: Three different datasets depending on the included descriptors

D1	D2	D3
metP	mtr	metP
C_dur_b	metP	C_dur_b
function	C_dur_b	intHop
type	isChordN	isChordN
tens	type	function
metP	tens	type
		tens

Table 7.5: Selected features for density

where $f(chord)$ is a function that takes as input the $Chord$ chord horizontal and/or vertical descriptors, and Den, Wei, Amp are the predicted *density*, *weight*, and *amplitude* performance actions, respectively.

Similarly, guitar models were trained as follows.

$$f(Note) \rightarrow (Emb) \tag{7.28}$$

where $f(note)$ is a function that takes as inputs a $Note$ characterized by the set of horizontal and/or vertical descriptors, and Emb is the predicted embellishment.

Feature Selection

For every dataset, we evaluated the descriptors by their information gain. Tables 7.2.3, 7.2.3, 7.2.3, and 7.2.3 show the best ranked descriptors for density, weight, amplitude and embellishments, respectively.

D1	D2	D3
tens	tens	tens
function	function	function
C_dur_b	type	type
metP	metP	metP
	isChordN	keyMode
	keyMode	isChordN
	mtr	tens

Table 7.6: Selected features for weight

D1	D2	D3
numKey	numKey	numKey
function	dur_s	pre_dur_s
type	duration_b	prev_int
tens	pre_dur_b	function
keyMode	nxt_dur_b	type
metP	isChordN	mtr
	function	isChordN
	mtr	tens
	type	keyMode
	tens	metP
	keyMode	
	metP	

Table 7.7: Selected features for amplitude

D1	D2	D3
phrase	phrase	phrase
dur_b	dur_b	dur_b
dur_s	dur_s	dur_s
pre_dur_b	pre_dur_b	pre_dur_b
pre_dur_s	pre_dur_s	pre_dur_s
onset	onset	onset
	tens	tens
	type	type
	function	function
	isChordN	isChordN
	keyMode	keyMode
		metP

Table 7.8: Selected features for embellishments, extracted from Table 4.2

Algorithms

We applied *decision trees*, *Support Vector Machine (SVM)* with a linear kernel and *Neural Networks (NN)* with one hidden layer. We used the implementation of these algorithms in the Weka Data Mining Software (Hall et al., 2009).

7.2.4 Results

Three models were generated for each performance action according to the three datasets (reference, horizontal, and vertical) datasets provided. We compared the accuracy among datasets and algorithms.

Piano data: density, weight and amplitude

We evaluated the accuracy (percentage of correct classifications) using 10-cross fold validation with 10 iterations. We performed statistical testing by using the t-test with a significance value of 0.05 to compare the methods with the baseline and decide if one produced measurably better results than the other. Table 7.9 shows the results for **density**. It can be seen that the accuracy increased when ensemble information was considered (datasets D1 and D2). The significant improvements were achieved by the algorithms NN and SVM, being 65.13 the highest accuracy reached with the dataset D2 which consisted in both harmonic and melodic score descriptors. For **weight** (Table 7.10), none of the results was statistically significant and the performance of the three models can be interpreted as random. The highest results were achieved when only piano information was considered (D1), showing no interaction between this performance action and the guitar melody. Table 7.11 presents the results for **amplitude**. In that case, the three algorithms reached their maximum accuracy when information of the ensemble performance (D3) was considered, which can be explained as a presence of correlation between the amplitude of the chords performed and the melody the piano player was hearing. Moreover, the results for the algorithms NN and SVM were statistically significant.

Guitar data: embellishments

Because guitar data presented a *skewed* classes distribution, we evaluated the performance of the models based on the sensitivity (true positive rate) rather than on the accuracy of the model. Table 7.12 presents the results obtained. It can be seen that, despite the low percentage of sensitivity,

Dataset	Baseline	NN	SVM	Decision Tree
D1	51.82	61.19 ○	62.13 ○	53.75
D2	51.82	61.72	65.13 ○	55.34
D3	51.82	55.75	61.75	57.65

○, ● statistically significant improvement or degradation

Table 7.9: Accuracy for the models of density in comparision to the baseline using NN, SVM and Decision Trees

Dataset	Baseline	NN	SVM	Decision Tree
D1	53.73	63.52	52.96	54.48
D2	53.73	50.62	49.64	51.85
D3	53.73	57.70	50.90	51.36

○, ● statistically significant improvement or degradation

Table 7.10: Accuracy for the models of weight in comparision to the baseline using NN, SVM and Decision Trees

Dataset	Baseline	NN	SVM	Decision Tree
D1	56.73	54.51	62.06	63.72
D2	56.73	57.11	60.90	60.93
D3	56.73	58.83	67.85 ○	67.98 ○

○, ● statistically significant improvement or degradation

Table 7.11: Accuracy for the models of amplitude in comparision to the baseline using NN, SVM and Decision Trees

the results for the three algorithms increased when considering ensemble information (D2, D3).

Dataset	NN	SVM	Decision Tree
D1	26	20	12
D2	30	38	26
D3	30	32	24

Table 7.12: Sensitivity percentage for embellishments

7.2.5 Discussion

We have generated models for different datasets consisting of information from individual performances and ensemble performances. Based on the accuracy and sensitivity of the models, we have obtained numerical results which have allowed us to estimate the level of interaction between musicians. The data analysis indicated that, in general terms, the performance actions of the accompaniment were influenced by the soloist and vice versa, since both written and performed descriptors contributed to a better performance of the models.

Chapter 8

Conclusions

In this chapter we present a summary of the main topics covered in this dissertation (Section 8.1). We also present the main contributions of our work in each of the topics investigated which were involved in the methodology (Section 1.5). Finally, we comment on some future work which we consider will complement the current investigation, exploring areas not fully covered as well as on future possible implementations and/or applications(Section 8.3).

8.1 summary

In this dissertation we have presented a computational modelling approach for expressive performance prediction and synthesis in jazz guitar music. Our primary objective has been to develop a system to generate predictive models for music expressive performance deviations in onset, duration, energy, as well as, complex ornamentation. Our motivation has been that, contrary to classical music, in jazz music expressive performance indications are seldom indicated on scores, and is the performer who introduce them based on his/her musical background/knowledge. The expressive devices used by musicians in popular music (concretely in jazz) is usually learnt by *copying* from the performance of expert musicians. Therefore, there is little quantitative information on how and in which context the manipulation of these expressive deviations occurs. The music material considered in this dissertation was obtained from both recordings from professional guitarists and commercial recordings obtained from audio CDs, and scores were obtained from The real book. We extracted a set of descriptors from the music scores and a symbolic representation from the audio recordings by implementing an automatic transcription scheme. In order to map performed notes to parent score notes we have automatically aligned performance to score data using a dynamic time warping approach. Based on this alignment we obtained performance actions, calculated as deviations of the performance from the score. We created an ornaments database including the information of the ornamented notes performed by the musician. Finally, we have applied machine learning to train predictive models for duration, onset, and energy deviations, as well as, for ornaments. Automatic induced rules were analysed from a musical perspective. In the next sections we summarize the main tasks performed in this work.

8.1.1 Semi-automatic data extraction

We have presented several approaches for semi-automatic data extraction from both monophonic-monotimbral/multitimbral audio signals. Firstly, we have presented an approach to automatically segment the extracted pitch profile form a monophonic-monotimbral/multitimbral audio signal, to obtain a MIDI-like representation of the note events. We have used *Yin* (De Cheveigné and Kawahara, 2002) and *Melodia* (Salamon and Gómez, 2012) algorithms respectively to extract a pith profile of the signal. Onset detection was performed applying adaptative energy filters, as well as, and pitch change detection. Heuristic rules were used to filter the resulting note events to remove noisy data. Results show good performance in the transcription of 30 audio samples. Secondly, we have presented a system to automatically recognize ornamentations in jazz music. We have used a data set of 27 audio recordings of jazz standards performed by a professional guitarist. We have applied Dynamic Time Warping to align the score with the performance of the musician, and match notes of the performance with the corresponding parent notes in the score. For evaluation purposes we have analysed the annotations of jazz musicians to generate an agreement level chart between the performance notes and parent score notes. Based on the experts' annotations, we have estimated the accuracy of the system by creating penalty factors based on how much the output of the algorithm differs from the human experts agreement. Results indicate that the accuracy of our approach is comparable with the accuracy of annotations of music experts. Thirdly, we have proposed a method to optimize the extraction parameters of *Melodia* algorithm for predominant melody extraction from polyphonic signals, using *Genetic Algorithms*. Our approach was oriented to improve the melodic extraction for specific instrument settings (e.g. electric guitar, bass, piano and drums). We have performed two experiments, a first one using 22 simultaneous midi and audio recordings, in which manual transcription was used to build the ground truth. Optimization was performed in 18 songs and 4 songs were left for testing. Two instrument settings were created: trio (guitar, bass, drums) and quartet (guitar, bass, piano, drums). Also different audio mixes were created in which the melody was at different sound levels with respect to the accompaniment track. The second experiment was performed on four commercial recordings in which the melody was recorded in a separated channel from the accompaniment track. In this experiment we used a *leave one out approach* for optimization and testing. Results show that our optimization methodology improves in all cases the overall accuracy of the detection for the first experiment. Lower improvement in overall accuracy in the second experiment was observed, which might be explained by the recordings context in which aspects like sound effects (e.g. reverb) in the melody track may induce more errors in the melodic detection.

8.1.2 Machine learning

He have reported on two main experiments applying machine learning techniques on data obtained from monophonic monotimbra and monophonic multitimbral data, respectively. In the first case our aim was to obtain models to rendering expressive performances and, on the second, was to obtain rules to be analysed from a musical perspective.

From monophonic monotimbral recordings, we have presented a machine learning approach for expressive performance (ornament, duration, onset and energy) prediction and synthesis in jazz guitar music. We used a data set of 27 recordings performed by a professional jazz guitarist, and extracted a set of descriptors from the music scores and a symbolic representation from the audio recordings. In order to map performed notes to parent score notes we have automatically aligned performance to score data. Based on this alignment we obtained performance actions, calculated as deviations of the performance from the score. We created an ornaments database including the information of the ornamented notes per- formed by the musician. We have compared four learning algorithms to create models for ornamentation, based on performance measures, using a significance Paired T-test. Feature selection techniques were employed to select the best feature subset for ornament modelling. For synthesis purposes, instance based learning was used to retrieve the most suitable ornament from the ornamentation data base. A concatenative synthesis approach was used to automatically generate expressive performances of new pieces (i.e. pieces not in the training set). Subjective perceptual evaluation based on listening tests might present difficulties based on the based on personal expectations of listeners . Therefore, we evaluated the performances generated by the system by computing the alignment distance between the system and the target performances. Evaluation results were consistent with the findings based on accuracy tests. Results may indicate that selected features contain sufficient information to capture the considered performance actions.

For the monophonic-multitimbral data we have presented a machine learning approach to obtain rule models for ornamentation, duration, onset and energy expressive performance actions. We considered 16 polyphonic recordings of American jazz guitarist Grant Green and the associated music scores. Note descriptors were extracted from the scores and audio recordings were processed in order to obtain a symbolic representation of the notes the main melody. Score to performance alignment was performed in order to obtain a correspondence between performed notes and score notes. From this alignment expressive performance actions were quantified. After discretizing the obtained performance actions we induced predictive models for each performance action prediction by applying a machine learning (sequential covering) rule learner algorithm. Extracted features were analysed by applying (both filter and wrapper) feature selection techniques. Models were evaluated using a 10-fold cross validation and statistical significance was established using paired t-test with respect to a baseline classifier. Concretely, the obtained accuracies for the ornamentation, duration, onset, and energy models are 71%, 58%, 60%, and 52%, respectively. Both the features selected and model rules showed musical significance. Similarities and differences among the obtained rules and the ones reported in the literature were discussed. Pattern similarities between classical and jazz music expressive rules were identified, as well as expected dissimilarities expected by the inherent particular musical aspects of each tradition. The induced rules' specificity/generality was assessed by applying them to performances of the same pieces performed by two other professional jazz guitar players. Results show a consistency in the ornamentation patterns between Grant Green and the other two musicians, which may be interpreted as a good indicator for generality of the ornamentation rules.

8.1.3 Applications

Finally, we have presented two applications of our the modelling strategies. A first application is a neuro-feedback approach which permits a user to manipulate some expressive parameters (duration, articulation and energy) of music performances using their emotional state in real-time. We have implemented a system for controlling in real-time the expressive aspects of a musical piece, by means of the emotional state detected from the EEG signal of a user. We have perform experiments in two different settings: a first one where the user tries to control the performance only by changing the arousal level, and a second one where the performance is dynamically changed between two extreme values (happy and sad), while the user is improvising playing a musical instrument. We applied machine learning techniques (LDA and SVM) to perform a two class classification task between two emotional states (happy and sad). Results suggest that EEG data contains sufficient information to distinguish between the two classes. This approach was utilized in a clinical pilot study for treating depression in elderly people, in which 10 participants were subjected to neuro-feedback sessions. Participants were instructed to listen to music pieces and were encouraged to increase the loudness and the and tempo of the pieces based on their arousal and valence levelsResults on the pre and post BDI depression test showed an average improvement on the BDI score. Also, EEG data analysis showed a relative decrease on the alpha activity. In a second application we have presented a study on the interaction between musicians by using our machine learning approach for jazz music. We have created a database consisting of recordings of 7 jazz standards performed by a quartet (piano, guitar, bass and drums) and their corresponding scores. After audio and score data extraction processing, we have computed vertical and horizontal descriptors for both notes and chords. Similarly, after aligning the score and the performance, we have measured some performance actions for guitar and piano. Finally, we have generated models for different datasets created with different combinations of horizontal (individual performances) and vertical (ensemble performances) descriptors. Based on the accuracy and sensitivity of the models, we have obtained numerical results which have allowed us to estimate the level of interaction between musicians. The data analysis may indicate that, the performance actions of the accompaniment are influenced by the soloist and vice versa, since both written and performed descriptors contributed to a better performance of the models.

8.2 Contributions

In this section we outline the main contributions of the dissertation

8.2.1 Contributions in expressive music performance modelling

- A methodology for expressive performance modeling in jazz music, able to operate on complex or free ornamentation, using the jazz guitar as a case study.
- A database of recordings with its respective transcription in machine readable format (MIDI) and score to performance alignment analysis, from 4 jazz guitar players, consisting of a total of 54 recordings that contain a total of 4537 notes.
- An statistical analysis of the performance of different algorithms used in the prediction of specific EPAs.
- Analysis of the relevance of the features involved in specific performance actions.
- Analysis of rules obtained from a musical perspective
- A comparison from the obtained rules with rules in literature
- An analysis on the gereability of the rules, based of rule coverage on jazz guitarrists performances.

8.2.2 Contributions on melodic representation and melodic description

- A library for note feature extraction and description implemented in Matlab code.
- A database of music scores and score descriptors with measured performance actions measured as deviations in timing, pitch, energy. Data base of descriptors include numerical and nominal descriptors. Similarly performance actions are obtain as numerical indexes as well as in the form of categories.
- The introduction of perceptual features for melodic description.

- Discretization of numerical features for the induction/study of expressive performance rules.

8.2.3 Contributions on automatic music transcription

- A methodology for music representation of the performance based on the manipulation of pitch contours extracted from both Monophnic and polyphonic audio into note events.
- A parameter optimization of the algorithm for melodic extraction from polyphonic signals, to the case study of the electric guitar in jazz misc context, by using genetic algorithms.

8.2.4 Contributions on score to performance alignment

- A methodology to automatically obtain score to performance correspondence for jazz guitar music, able to work on complex ornamentation cases.
- A GUI to facilitate the annotation of ground truth for score to performance alignment
- A framework to obtain a ground truth on jazz performance score alignment, based on agreement

8.2.5 Contributions of the applications developed

- A system for neuro-feedback for real-time manipulation of expressive parameters based on the perceived emotional state.

8.3 Future directions

8.3.1 Onset detection using ensemble methods

Currently we are extending our approach for automatic music transcription by applying the *staked generalization* ensemble method (*staking*), in which the output of three different pitch profile segmentation algorithms is used to train a metalearner to discover how best to combine the output of the three approaches for transient detection (i.e. detect the frames in which a note transition *onset* and/or *offset* occurs). The dataset is constructed based on the onset transition marking on a frame level (2.9 miliseconds) based on the pitch profile segmentation performed on 21 monophonic audio recordings of jazz standard pieces, performed by three different jazz guitarists, using two well known algorithms (Mcnab et al. (1996) and Mauch et al. (2015)) (which were reviewed previously in section 2.3.1), as well as our pitch profile segmentation approach based on energy and heuristic filters explained in Section 3.3.3. The manual transcribed pieces created by an human expert are marked as well on the same frame basis, and transition markings on the corresponding frames are used as classes. In all cases, each frame is marked as *transition* = *yes* when a note onset/offset (or both) is present in that particular frame, and *transition* = *no* otherwise. Thus, each frame will be treated as an *instance* for which we have 3 features which are the makings obtained based on the segmentation of the three algorithms considered. In real machine learning schemes, staking method does not simply attach the predictions of each classifier to build a training instance, as it will allow to learn simplistic rules by the meta learner. Usually 10 cross fold validation and leave one out methods are used to use the models generated on each fold to create new predictions on the data which to create the instances of the metalearner. In our specific case we used a sliding window of 100ms, based on the MIREX (Music Information Retrieval Evaluation eXchange) framework validation approach for onset detection. Thus, the features of the frames within a window are line up together to generate a 100 feature size instance that corresponds to the center frame of the window. We plan to test several machine learning schemes for the metalearner and compare them upon on the accuracy measured as correctly classified instances.

8.3.2 Polyphonic hexaphonic guitar expressive performance modelling

So far we have presented a monophonic approach for guitar expressive analysis. However, guitar is by nature a polyphonic instrument, therefore we plan to extend our approach for expressive guitar modelling to the hexaphonic scenario. We have explored the extraction and analysis of hexaphonic guitar signals in Angulo et al. (2016), where we presented an approach to visualize guitar performances, transcribing musical events into visual forms. Hexaphonic guitar processing was carried out (i.e. processing each of the six strings as an independent monophonic sound source) to obtain high level descriptors from the audio signal, and, after, different visualization mappings were tested to meaningfully/intuitively represent music. This work was part of the M.S. thesis by (Angulo et al., 2016) which was directed by the author of the present dissertation. Contributions of this work include several hexaphonic recordings, hardware for hexaphonic recording, as well as, libraries for independent string signal filtering and processing. Thus, we have a framework which makes feasible apply our machine learning approach to hexaphonic recordings.

8.3.3 Evaluation of performance based on machine learning modelling.

In the context of the TELMI project, we plan to implement our expressive approach for expressive music performance. On one hand we plan to obtain high level audio features and gesture actions from multimodal recordings of violin experts. From the estimation of music/sound features, we will propose models and techniques that map the sound domain to the control (gesture) domain (force, velocity, etc.). On the other hand we plan to propose generalized models from the analysis of a reference database of violin masters, to obtain metrics for the automatic evaluation and learning assessment.

%labelpart:appendix

Bibliography

Each reference indicates the pages where it appears.

A. Accardo, M. Affinito, M. Carrozzi, and F. Bouquet. Use of the fractal dimension for the analysis of electroencephalographic time series. *Biological cybernetics*, 77(5):339–50, November 1997. ISSN 0340-1200. URL http://www.ncbi.nlm.nih.gov/pubmed/9418215. 25

Iñigo Angulo, Sergio Giraldo, and Rafael Ramirez. Hexaphonic guitar transcription and visualization. In Richard Hoadley, Chris Nash, and Dominique Fober, editors, *Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2016*, pages 187–192, Cambridge, UK, 2016. Anglia Ruskin University. ISBN 978-0-9931461-1-4. 136

Josep Lluís Arcos, Ramon Lopez De Mantaras, and Xavier Serra. Saxex: A case-based reasoning system for generating expressive musical performances. *Journal of New Music Research*, 27(3):194–210, 1998. 3, 13, 15, 21, 116

T. H. Aspiras and V. K. Asari. Log power representation of eeg spectral bands for the recognition of emotional states of mind. In *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*, pages 1–5, Dec 2011. doi: 10.1109/ICICS.2011.6174212. 107

Helena Bantula. Multi player expressive performance modeling in jazz music. B.s. thesis, Pompeu Fabra University, Barcelona, Spain, 2015. 117

Helena Bantula, Sergio Giraldo, and Rafael Ramirez. Jazz ensemble expressive performance modeling. In *Proc. of the 17th Int. Conf. on Music*

Information Retrieval (ISMIR), pages 31–37, New York, New York, USA, 2016. 117

J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, Sept 2005. ISSN 1063-6676. doi: 10.1109/TSA.2005.851998. 44

Dmitry Bogdanov, Nicolas Wack, E. Gómez, Sankalp Gulati, Perfecto Herrera, O. Mayor, Gerard Roma, Justin Salamon, J. R. Zapata, and X. Serra. Essentia: an audio analysis library for music information retrieval. In *International Society for Music Information Retrieval Conference (ISMIR'13)*, pages 493–498, Curitiba, Brazil, 04/11/2013 2013. 42, 62

J. Bonada, X. Serra, X. Amatriain, and A. Loscos. Spectral processing. In Udo Zolzer, editor, *DAFX Digital Audio Effects*, pages 393–444. John Wiley and Sons Ltd, 2011. ISBN 978-0-470-66599-2. 21

Roberto Bresin. Artificial neural networks based models for automatic performance of musical scores. *Journal of New Music Research*, 27(3):239–270, 1998. 13, 15

Roberto Bresin and Anders Friberg. Emotional coloring of computer-controlled music performances. *Computer Music Journal*, 24(4):44–63, 2000. 13, 17

Judith C. Brown and Bin Zhang. Musical frequency tracking using the methods of conventional and narrowed autocorrelation. *The Journal of the Acoustical Society of America*, 89(5):2346–2354, 1991. doi: http://dx.doi.org/10.1121/1.400923. URL http://scitation.aip.org/content/asa/journal/jasa/89/5/10.1121/1.400923. 23

Emilios Cambouropoulos. *Music, Gestalt, and Computing: Studies in Cognitive and Systematic Musicology*, chapter Musical rhythm: A formal model for determining local boundaries, accents and metre in a melodic surface, pages 277–293. Springer, Berlin, Heidelberg, 1997. ISBN 978-3-540-69591-2. 48

Antonio Camurri, Roberto Dillon, and Alberto Saron. An experiment on analysis and synthesis of musical expressivity. In *Proceedings of 13th Colloquium on Musical Informatics*, 2000. 13, 15, 18, 25

S. Canazza, G. De Poli, C. Drioli, A. Rodà, and A. Vidolin. Audio morphing different expressive intentions for multimedia systems. *IEEE MultiMedia*, 7(3):79–83, July 2000. ISSN 1070-986X. URL http://dl.acm.org/citation.cfm?id=614667.614998. 15, 18

S. Canazza, G. De Poli, C. Drioli, A. Roda, and A. Vidolin. Expressive morphing for interactive performance of musical scores. In *Web Delivering of Music, 2001. Proceedings. First International Conference on*, pages 116–122, 2001. 15, 18

S. Canazza, G. De Poli, A. Rodà, and A. Vidolin. An abstract control space for communication of sensory expressive intentions in music performance. *Journal of New Music Research*, 32(3):281–294, 2003. doi: 10.1076/jnmr.32.3.281.16862. 15, 18

L. Carlson, A. Nordmark, and R. Wikilander. Reason version 2.5 propellerhead software, 2003. 16

Michael Casey and Tim Crawford. Automatic location and measurement of ornaments in audio recordings. In *Proc. of the 5th Int. Conf. on Music Information Retrieval (ISMIR)*, pages 311–317, 2004. 24

Jadzia Cendrowska. Prism: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 27(4):349–370, 1987. 95

A. Chopin. Eeg-based human interface for disabled individuals: Emotion expression with neural networks. Master's thesis, Tokyo Institute of Technology, Yokohama, Japan, 2000. 24

L. P. Clarisse, J. P. Martens, M. Lesaffre, B. De Baets, H. Demeyer, and M. Leman. An auditory model based transcriber of singing sequences. In *in ISMIR*, pages 116–123, 2002. 61

M. Clynes. Generative principles of musical thought: Integration of microstructure with structure. *Communication and Cognition AI, Journal for the Integrated Study of Artificial Intelligence, Cognitive Science and Applied Epistemology*, (3), 1986. 15

M. Clynes. Microstructural musical linguistics: composers pulses are liked most by the best musicians. *Cognition*, 55(3):269 – 310, 1995. ISSN 0010-0277. doi: 10.1016/0010-0277(94)00650-A. URL http://www.sciencedirect.com/science/article/pii/001002779400650A. 15

William W Cohen. Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning*, pages 115–123, 1995. 95

Grosvenor Cooper and Leonard B Meyer. *The rhythmic structure of music*, volume 118. University of Chicago Press, 1963. 49

P. Dahlstedt. Autonomous Evolution of Complete Piano Pieces and Performances. *In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007*, 4648, 2007. 16

RB. Dannenberg, H. Pellerin, and I. Derenyi. A study of trumpet envelopes. In *The 1998 International Computer Music Conference*, number April, pages 57–61, San Francisco, USA, 2007. International Computer Music Asociation. 15

Roger B Dannenberg and Istvan Derenyi. Combining instrument and performance models for high-quality music synthesis. *Journal of New Music Research*, 27(3):211–238, 1998. 15

Alain De Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002. 7, 23, 39, 64, 65, 130

L. Dorard, DR. Hardoon, and J. Shawe-Taylor. Can Style be Learned? A Machine Learning Approach Towards 'Performing' as Famous Pianists. *Music Brain and Cognition Workshop*, 2007. 16

Jean-Louis Durrieu. *Automatic transcription and separation of the main melody in polyphonic music signals.* PhD thesis, Ecole nationale supérieure des telecommunications-ENST, 2010. 24

T. Eerola and J. K. Vuoskoski. A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, 39(1):18–49, August 2010. ISSN 0305-7356. doi: 10.1177/0305735610362821. URL http://pom.sagepub.com/cgi/doi/10.1177/0305735610362821. 25

Tuomas Eerola and Petri Toiviainen. *MIDI Toolbox: MATLAB Tools for Music Research.* University of Jyväskylä, Jyväskylä, Finland, 2004. URL www.jyu.fi/musica/miditoolbox/. 37, 48

Daniel PW Ellis. *Prediction-driven computational auditory scene analysis.* PhD thesis, Massachusetts Institute of Technology, 1996. 23

M. Fabiani. *Interactive computer-aided expressive music performance.* PhD thesis, KTH School of Computer Science and Communication, Stockholm, SWEDEN, 2011. 25

X. Fiss and A. Kwasinski. Automatic real-time electric guitar audio transcription. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 373–376, May 2011. doi: 10.1109/ICASSP.2011.5946418. 24

Anders Friberg. Generative rules for music performance: A formal description of a rule system. *Computer Music Journal*, 15(2):56–71, 1991. 17

Anders Friberg. pdm: an expressive sequencer with real-time control of the kth music-performance rules. *Computer Music Journal*, 30(1):37–48, 2006. 13, 17, 25

Anders Friberg, Roberto Bresin, and Johan Sundberg. Overview of the kth rule system for musical performance. *Advances in Cognitive Psychology*, 2(2-3):145–161, 2006. 13, 15, 17, 103

Nicolas Froment, Werner Schweer, and Thomas Bonte. GTS: GNUmusescore. http://www.musescore.org/, 2011. 34

Mazzola G. *The Topos of Music: Geometric Logic of Concepts, Theory, and Performance*. Birkhäuser, Boston, 2002. 15

Alf Gabrielsson. The performance of music. In Diana Deutsch, editor, *The psychology of music*, volume 2, pages 501–602. Elsevier, 1999. 3, 12

Alf Gabrielsson. Music performance research at the millennium. *Psychology of music*, 31(3):221–272, 2003. 3, 12

James Lincoln Collier Geoffrey L. Collier. An exploration of the use of tempo in jazz. *Music Perception: An Interdisciplinary Journal*, 11(3): 219–242, 1994. ISSN 07307829, 15338312. 92

Sergio Giraldo. Modeling embellishment, duration and energy expressive transformations in jazz guitar. Master's thesis, Pompeu Fabra University, Barcelona, Spain, 2012. 14, 21, 47

Sergio Giraldo and Rafael Ramírez. Optimizing melodic extraction algorithm for jazz guitar recordings using genetic algorithms. In *Joint Conference ICMC-SMC 2014, Athens, Greece*, pages 25–27, 2014. 40

Sergio Giraldo and Rafael Ramírez. Computational modeling and synthesis of timing, dynamics and ornamentation in jazz guitar music. In *11th International Symposium on Computer Music Interdisciplinary Research CMMR 2015, Plymuth, UK*, pages 806–814, 2015a. 14, 48

Sergio Giraldo and Rafael Ramírez. Performance to score sequence matching for automatic ornament detection in jazz music. In *International Conference of New Music Concepts ICMNC 2015, Treviso, Italy*, page 8, 2015b. 53

Sergio Giraldo and Rafael Ramírez. Computational modelling of ornamentation in jazz guitar music. In *International Symposium in Performance Science*, pages 150–151, Kyoto, Japan, 02/09/2015 2015c. Ryukoku University, Ryukoku University. URL http://www.mtg.upf.edu/system/files/publications/ISPS2015_Giraldo.pdf. 14

Sergio Giraldo and Rafael Ramírez. Computational generation and synthesis of jazz guitar ornaments using machine learning modeling. In *Proceedings of the 11th International Conference on Machine Learning and Music(MML 2014) held in Vancouver, Canada, August, 2015*, pages 10–12, 2015d. 14

W. Goebl and C. Palmer. Synchronization of timing and motion among performing musicians. *Music Perception*, pages 427–438, 2009. URL http://www.jstor.org/stable/10.1525/mp.2009.26.5.427. 22

Werner Goebl, Simon Dixon, Giovanni De Poli, Anders Friberg, Roberto Bresin, and Gerhard Widmer. Sense in expressive music performance: Data acquisition, computational studies, and models. *Sound to sense-sense to sound: A state of the art in sound and music computing*, pages 195–242, 2008. 3, 11, 13

Werner Goebl, Simon Dixon, and Emery Schubert. Quantitative methods: Motion analysis, audio analysis, and continuous response techniques. *Expressiveness in music performance: Empirical approaches across styles and cultures*, page 221, 2014. 13

Francisco Gómez, Aggelos Pikrakis, Joaquín Mora, Juan Manuel Díaz-Bánez, Emilia Gómez, and Francisco Escobar. Automatic detection of ornamentation in flamenco. In *Fourth International Workshop on Machine Learning and Music MML*, pages 20–22, 2011. 4, 24

Masataka Goto. A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication*, 43(4):311 – 329, 2004. ISSN 0167-6393. doi: http://dx.doi.org/10.1016/j.specom.2004.07.001. URL `http://www.sciencedirect.com/science/article/pii/S0167639304000640`. Special Issue on the Recognition and Organization of Real-World Sound. 39

Maarten Grachten. *Expressivity-aware tempo transformations of music performances using case based reasoning*. PhD thesis, Universitat Pompeu Fabra, 2006. 3, 14, 48, 56, 86, 92, 116

G. Grindlay. *Modelling Expressive Musical Performance with Hidden Markov Models*. PhD thesis, UNIVERSITY OF CALIFORNIA SANTA CRUZ, 2005a. 16

Graham Charles Grindlay. *Modeling expressive musical performance with Hidden Markov Models*. 2005b. 13

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009. 77, 95, 125

M. Hashida, N. Nagata, and H. Katayose. jpop-e: an assistant system for performance rendering of ensemble music. In *Proceedings of the 7th international conference on New interfaces for musical expression*, NIME '07, pages 313–316, New York, NY, USA, 2007. ACM. doi: 10.1145/1279740.1279808. URL `http://doi.acm.org/10.1145/1279740.1279808`. 15

Thomas Hedges, Pierre Roy, and FranÃ§ois Pachet. Predicting the composer and style of jazz chord progressions. *Journal of New Music Research*, 43(3):276–290, 2014. doi: 10.1080/09298215.2014.925477. URL http://dx.doi.org/10.1080/09298215.2014.925477. 93

T. Higuchi. Approach to an irregular time series on the basis of the fractal theory. In *Physica D*, volume 31, page 277–283. 1988. 25

K. Hirata and R. Hiraga. Ha-hi-hun : Performance rendering system of high controllability, 2002. URL http://www.fun.ac.jp/~hirata/Papers/icad2002-rencon-ws.pdf. 15

Jui-Chung Hung and Ann-Chen Chang. Combining genetic algorithm and iterative music searching doa estimation for the cdma system. *Expert Systems with Applications*, 38(3):1895–1902, 2011. 42

Margaret L Johnson. Toward an expert system for expressive musical performance. *Computer*, 24(7):30–34, 1991a. 13

ML. Johnson. Toward an expert system for expressive musical performance. *Computer*, 24(7):30–34, 1991b. ISSN 0018-9162. doi: 10.1109/2.84832. 15

P. Juslin. Communicating emotion in music performance: A review and a theoretical framework. In Juslin and Sloboda, editors, *Music and emotion: Theory and research.Series in affective science.*, pages 309–337. Oxford University Press, New York, 2001. 11, 12, 17

H. Katayose, T. Fukuoka, K. Takami, and S. Inokuchi. Expression extraction in virtuoso music performances. In *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, volume i, pages 780–784 vol.1, 1990. doi: 10.1109/ICPR.1990.118216. 15

Gary Kennedy and Barry Kernfeld. Aebersold, jamey. In *The new Grove dictionary of jazz, vol. 1 (2nd ed.)*, pages 16–17. New York: Grove's Dictionaries Inc, 2002. ISBN 1561592846. 29, 64

Alexis Kirke and Eduardo R Miranda. An overview of computer systems for expressive music performance. In *Guide to computing for expressive music performance*, pages 1–47. Springer, 2013. 3, 13, 14

Shinpei Koga, Takafumi Inoue, and Makoto Fukumoto. A proposal for intervention by user in interactive genetic algorithm for creation of music melody. In *Biometrics and Kansei Engineering (ICBAKE), 2013 International Conference on*, pages 129–132. IEEE, 2013. 42

Carol L Krumhansl and Edward J Kessler. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological review*, 89(4):334, 1982. 49

F Lerdahl and R. Jackendoff. *A generative theory of tonal music / Fred Lerdahl, Ray Jackendoff.* MIT Press, Cambridge, Mass. :, 1983. ISBN 0262120941 0262620499 026262107. 12, 21

Fred Lerdahl. Calculating tonal tension. *Music Perception: An Interdisciplinary Journal*, 13(3):319–363, 1996. 37, 49

Sonnus Limited. G2m universal guitar to midi converter, 2012. URL `http://www.sonuus.com/products_g2m.html`. 64

Y. Lin, C. Wang, T. Jung, Senior Member, T. Wu, S. Jeng, J. Duann, and J. Chen. EEG-Based Emotion Recognition in Music Listening. *IEEE Transactions on Biomedical Engineering*, 57(7):1798–1806, 2010. 24

Y. Liu, O. Sourina, and MK. Nguyen. Real-Time EEG-Based Human Emotion Recognition and Visualization. In *2010 International Conference on Cyberworlds*, pages 262–269. Ieee, October 2010. ISBN 978-1-4244-8301-3. doi: 10.1109/CW.2010.37. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5656346`. 25

SR. Livingstone, RM., AR. Brown, and WF. Thompson. Changing musical emotion : a computational rule system for modifying score and performance. *Computer Music Journal*, 34(1):41–64, 2010. URL `http://eprints.qut.edu.au/31295/`. 15, 17

Marco Marchini. *Analysis of ensemble expressive performance in string quartets: a statistical and machine learning approach.* PhD thesis, Universitat Pompeu Fabra, 2014. 23, 116

W. Marshall. *Best of Jazz.* Hall Leonard, Milwaukee, W, USA, 2000. 21, 65, 66

Dragan Matić. A genetic algorithm for composing music. *Yugoslav Journal of Operations Research ISSN: 0354-0243 EISSN: 2334-6043*, 20(1), 2013. 42

M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon. Computer-aided melody note transcription using the tony software: Accuracy and efficiency. In *Proceedings of the First International Conference on Technologies for Music Notation and Representation*, May 2015. accepted. 62, 136

Zahorka O. Mazzola G. Tempo curves revisited: Hierarchies of performance fields. *Computer Music Journal*, 18(1):40–52, 1994. doi: 10.1080/09298219808570747. 15

Rodger J. Mcnab, Lloyd A. Smith, and Ian H. Witten. Signal processing for melody transcription. In *Proc. 19th Australasian Computer Science Conf., 301–307*, pages 301–307, 1996. 43, 62, 63, 136

Eduardo R Miranda, Alexis Kirke, and Qijun Zhang. Artificial evolution of expressive performance of music: an imitative multi-agent systems approach. *Computer Music Journal*, 34(1):80–96, 2010a. 13

ER Miranda, A. Kirke, and Q. Zhang. Artificial Evolution of Expressive Performance of Music: An Imitative Multi-Agent Systems Approach. *Computer Music Journal*, 34(1):80–96, March 2010b. ISSN 0148-9267. doi: 10.1162/comj.2010.34.1.80. URL `http://dx.doi.org/10.1162/comj.2010.34.1.80`. 16

G. P Moore and J. Chen. Timings and interactions of skilled musicians. *Biological cybernetics*, 103(5):401–14, November 2010. ISSN 1432-0770. doi: 10.1007/s00422-010-0407-5. URL `http://www.ncbi.nlm.nih.gov/pubmed/21046143`. 22

Eugene Narmour. *The analysis and cognition of melodic complexity: The implication-realization model.* University of Chicago Press, 1992. 21, 48, 49, 92

A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval. Adaptation of bayesian models for single-channel source separation and its application to voice/music separation in popular songs. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1564–1578, July 2007. ISSN 1558-7916. doi: 10.1109/TASL.2007.899291. 24

Rui Pedro Paiva, Teresa Mendes, and Amílcar Cardoso. Melody detection in polyphonic musical signals: Exploiting perceptual rules, note salience, and melodic smoothness. *Computer Music Journal*, 30(4):80–98, 2006. 40

Caroline Palmer. Music performance. *Annual review of psychology*, 48(1): 115–138, 1997. 3, 12

T. Partala, M. Jokinierni, and V. Surakka. Pupillary Responses To Emotionally Provocative Stimuli. In *ETRA 00: 2000 Symposium on Eye Tracking Research & Aplications*, pages 123–129, New York, New York, USA, 2000. ACM Press. 24

Alfonso Perez, Esteban Maestre, Stefan Kersten, and Rafael Ramírez. Expressive Irish fiddle performance model informed with bowing. In *Proceedings of the international computer music conference.* ICMC 2008, sarc, Belfast, N. Ireland, 2008. 4, 24

Mark D Plumbley, Samer A Abdallah, Juan Pablo Bello, Mike E Davies, Giuliano Monti, and Mark B Sandler. Automatic music transcription and audio source separation. *Cybernetics &Systems*, 33(6):603–627, 2002. 23

Giovanni De Poli, Sergio Canazza, Antonio Rodà, and Emery Schubert. The role of individual difference in judging expressiveness of computer-assisted music performances by experts. *ACM Trans. Appl. Percept.*, 11 (4):22:1–22:20, December 2014. ISSN 1544-3558. doi: 10.1145/2668124.

URL http://doi.acm.org/10.1145/2668124. 84

Montserrat Puiggròs, Emilia Gómez, Rafael Ramírez, Xavier Serra, and Roberto Bresin. Automatic characterization of ornamentation from bassoon recordings for expressive synthesis. In *Proceedings of 9th International Conference on Music Perception and Cognition*. University of Bologna (Italy), August 22-26, ICMP, 2006. 4, 24

Brigitte Rafael, Michael Affenzeller, and Stefan Wagner. Application of an island model genetic algorithm for a multi-track music segmentation problem. In *International Conference on Evolutionary and Biologically Inspired Music and Art*, pages 13–24. Springer, 2013. 42

Rafael Ramírez and Amaury Hazan. A tool for generating and explaining expressive music performances of monophonic jazz melodies. *International Journal on Artificial Intelligence Tools*, 15(04):673–691, 2006. 3, 13, 19, 116

Rafael Ramirez and Zacharias Vamvakousis. *Detecting Emotion from EEG Signals Using the Emotive Epoc Device*, pages 175–184. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35139-6. doi: 10.1007/978-3-642-35139-6_17. URL http://dx.doi.org/10.1007/978-3-642-35139-6_17. 25, 106

Rafael Ramirez, Amaury Hazan, Esteban Maestre, and Xavier Serra. A genetic rule-based model of expressive performance for jazz saxophone. *Computer Music Journal*, 32(1):38–50, 2008. 16, 20

Rafael Ramirez, Manel Palencia, Sergio Giraldo, and Zacharias Vamvakousis. Musical neurofeedback for treating depression in elderly people. *Frontiers in Neuroscience*, 9(354), 2015. ISSN 1662-453X. doi: 10.3389/fnins.2015.00354. URL http://www.frontiersin.org/auditory_cognitive_neuroscience/10.3389/fnins.2015.00354/abstract. 115

C. Raphael. Can the Computer Learn to Play Music Expressively? In *In Jaakkola T, Richardson T (eds) Proceedings of eighth international workshop on arificial inteligence and statistics*, pages 113–120. Morgan Kaufmann, 2001a. 15, 22

C. Raphael. A bayesian network for real-time musical accompaniment. In *Advances in Neural Information Processing Systems, NIPS 14*, page 14. MIT Press, 2001b. 15, 22

Christopher Raphael. Orchestra in a box: A system for real-time musical accompaniment. In *IJCAI workshop program APP-5*, pages 5–10, Acapulco, Mexico, 2003. Morgan Kaufmann. 15, 22

Gustavo Reis, Nuno Fonseca, Francisco Fernández De Vega, and Anibal

Ferreira. Hybrid genetic algorithm based on gene fragment competition for polyphonic music transcription. In *Workshops on Applications of Evolutionary Computation*, pages 305–314. Springer, 2008. 42

BH. Repp. Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review*, 12(6):996–992, 2005. URL http://link.springer.com/article/10.3758/BF03206433. 22

Emotiv Systems Inc. Researchers. Emotive epoc, 2012. URL http://emotiv.com/epoc/. 106

Edward C. Carterette Roger A. Kendall. The communication of musical expression. *Music Perception: An Interdisciplinary Journal*, 8(2):129–163, 1990. ISSN 07307829, 15338312. URL http://www.jstor.org/stable/40285493. 12

Justin Salamon and Emilia Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(6):1759–1770, 2012. 7, 23, 40, 63, 64, 130

E Glenn Schellenberg. Simplifying the implication-realization model of melodic expectancy. *Music Perception: An Interdisciplinary Journal*, 14 (3):295–318, 1997. 49

CE. Seashore. *Psicology of Music*. McGraw-Hill, New York, 1938. 12

Xavier Serra. Musical sound modeling with sinusoids plus noise. *Musical signal processing*, pages 91–122, 1997. 7, 78

Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981. 24

J. Sundberg, A. Friberg, and L. Frydén. Rules for automated performance of ensemble music. *Contemporary Music Review*, 1989. URL http://www.tandfonline.com/doi/full/10.1080/07494468900640071. 22, 116

Johan Sundberg. How can music be expressive? *Speech communication*, 13 (1):239–253, 1993. 17

Johan Sundberg, Anders Askenfelt, and Lars Frydén. Musical performance: A synthesis-by-rule approach. *Computer Music Journal*, 7(1):37–43, 1983. 17

T. Suzuki, T. Tokunaga, and H. Tanaka. A case based approach to the generation of musical expression. In *In Proc. of IJCAI*, pages 642–648, 1999. 15

K. Takahashi. Remarks on Emotion Recognition from Bio-Potential Signals. *2nd International Conference on Autonomous Robots and Agents*, pages

186–191, 2004. 24

The real book. *The real book*. Hall Leonard, Milwaukee, WI, USA, 2004. 7, 27, 28, 29, 33, 89, 117, 129

A. Tobudic and G. Widmer. Relational ibl in music with a new structural similarity measure. In *In Proceedings of the International Conference on Inductive Logic Programming*, pages 365–382. Springer-Verlag, 2003. 15

Neil Todd. A computational model of rubato. *Contemporary Music Review*, 3(1):69–88, 1989. 13, 15

Neil P McAngus Todd. The dynamics of dynamics: A model of musical expression. *The Journal of the Acoustical Society of America*, 91(6):3540–3550, 1992. 13, 15

Neil P McAngus Todd. The kinematics of musical expression. *The Journal of the Acoustical Society of America*, 97(3):1940–1949, 1995. 13, 15

Caroline Traube and Michel Bernays. Piano Touch Analysis: a Matlab Toolbox for Extracting Performance Descriptors from High Resolution Keyboard and Pedalling Data. *Journées d'Informatique Musicale (JIM)*, 2012. 120

Paul Von Hippel. Redefining pitch proximity: Tessitura and mobility as constraints on melodic intervals. *Music Perception: An Interdisciplinary Journal*, 17(3):315–327, 2000. 50

Sethares W. *Tuning timbre, spectrum, scale*. Springer, London, 2004. 15

G. Widmer. Large-scale induction of expressive performance rules: First quantitative results. In *In Proceedings of the International Computer Music Conference (ICMC'2000). San Francisco, CA: International Computer Music Association*, pages 344–347, 2000. 15, 18

Gerhard Widmer. Machine discoveries: A few simple, robust local expression principles. *Journal of New Music Research*, 31(1):37–50, 2002. 15, 18, 103

Gerhard Widmer. Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artificial Intelligence*, 146(2):129–148, 2003. 13, 15, 18, 99, 103

Gerhard Widmer and Asmir Tobudic. Playing mozart by analogy: Learning multi-level timing and dynamics strategies. *Journal of New Music Research*, 32(3):259–268, 2003. 15

Alan M. Wing, Satoshi Endo, Adrian Bradbury, and Dirk Vorberg. Optimal feedback correction in string quartet synchronization. *Journal of The Royal Society Interface*, 11(93), 2014. ISSN 1742-5689. doi:

10.1098/rsif.2013.1125. URL http://rsif.royalsocietypublishing.org/content/11/93/20131125. 22

Herbert Woodrow. *Time perception.* Wiley, 1951. 54

José R Zapata, André Holzapfel, Matthew EP Davies, João Lobato Oliveira, and Fabien Gouyon. Assigning a confidence threshold on automatic beat annotation in large datasets. In *13th International Society for Music Information Retrieval Conference, Porto, Portugal, October 8th-12th, ISMIR*, pages 157–162, 2012. 7, 122

Udo Zölzer. *Digital audio signal processing.* John Wiley & Sons, 2008. 44