Software-Based Extraction of Objective Parameters from Music Performances

vorgelegt von
Diplom-Ingenieur
Alexander Lerch
aus Erlangen

Von der Fakultät I – Geisteswissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades eines
Dr. phil.

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Eberhard Knobloch
Berichter: Prof. Dr. Stefan Weinzierl
Berichter: Prof. Dr. Roger B. Dannenberg

Tag der wissenschaftlichen Aussprache: 27.10.2008

Berlin 2008
D83

Abstract

Different music performances of the same score may significantly differ from each other. It is obvious that not only the composer's work, the score, defines the listener's music experience, but that the music performance itself is an integral part of this experience. Music performers use the information contained in the score, but interpret, transform or add to this information.
Four parameter classes can be used to describe a performance objectively: tempo and timing, loudness, timbre and pitch. Each class contains a multitude of individual parameters that are at the performers' disposal to generate a unique physical rendition of musical ideas.
The extraction of such objective parameters is one of the difficulties in music performance research. This work presents an approach to the software-based extraction of tempo and timing, loudness and timbre parameters from audio files to provide a tool for the automatic parameter extraction from music performances.
The system is applied to extract data from 21 string quartet performances and a detailed analysis of the extracted data is presented.
The main contributions of this thesis are the adaptation and development of signal processing approaches to performance parameter extraction and the presentation and discussion of string quartet performances of a movement of Beethoven's late String Quartet op. 130.

music performance, music performance analysis, automatic tempo extraction, loudness analysis, timbre analysis, string quartet performance, audio content analysis, audio-to-score-matching

Zusammenfassung

Verschiedene Aufführungen des gleichen musikalischen Werkes unterscheiden sich deutlich voneinander. Es ist offensichtlich, daß das Musikerlebnis des Hörers nicht nur durch die zugrundeliegende Partitur bestimmt wird, sondern auch maßgeblich von der Interpretation dieser Partitur durch die aufführenden Musiker. Diese deuten, modifizieren oder erweitern die im Notenbild enthaltenen Informationen im Zuge ihrer Darbietung.
Eine solche Musikaufführung läßt sich mit Parametern der Parameterkategorien Tempo, Lautheit, Klangfarbe und Tonhöhe objektiv beschreiben. Jede der vier Kategorien stellt eine Vielzahl von Parametern bereit, die es den Musikern ermöglicht, musikalische Ideen auf eine einmalige physikalische Art umzusetzen. Die Extraktion solcher Parameter ist eine der typischen Problemstellungen der Aufführungsanalyse. Diese Arbeit präsentiert ein Softwaresystem, das als Werkzeug zur automatischen Extraktion von Tempo-, Lautheits- und Timbremerkmalen angewendet werden kann.
Dieses System wurde für eine systematische Analyse von 21 Streichquartettaufnahmen eingesetzt.
Die Arbeit widmet sich hauptsächlich zwei Thematiken, der Entwicklung und Optimierung von Algorithmen der Audiosignalverarbeitung zur Parameterextraktion aus Audioaufnahmen musikalischer Aufführungen sowie der Analyse und Diskussion von Streichquartrettaufführungen eines Satzes aus Beethovens spätem Streichquartett op. 130.

musikalische Interpretation, Aufführungsanalyse, Tempoerkennung, Lautheitsanalyse, Klangfarbenanalyse, Streichquartettanalyse, Musikanalyse

Acknowledgments

First and foremost, I would like to express my gratitude to Stefan Weinzierl without whom this work never would have started. He piqued my interest in music performance analysis and made this work possible. My sincere thanks go to Roger B. Dannenberg for his commitment, his insightful comments and the interesting discussions.
I have been very lucky to be able to count on Tim Flohrer and Martin Schwerdtfeger, my friends and colleagues at *zplane.development*, who not only encouraged my plans to write a doctorate thesis but actively supported me during all this time.
I am very grateful to all the people who have reviewed the thesis or parts of it and helped me to improve it. Particularly, I would like to express my gratitude to Alexander Vorwerk and Odoch Hawkins for their detailed and invaluable feedback.
I am obliged to my family for their continuous encouragement and support.
This research has been partially funded by the City of Berlin by means of a NaFöG PhD scholarship.

Contents

List of Symbols and Abbreviations

ACA	Audio Content Analysis
ADSR	technical description of the volume envelope phases of single sounds, frequently used in synthesizers: attack, decay, sustain and release
ANOVA	Analysis of Variance
AOT	Acoustic Onset Time
API	Application Programmer's Interface
AU	Audio Unit (Apple)
$\mathfrak{B}$	similarity matrix
$\mathfrak{b}(m, n)$	similarity matrix entry
$\mathcal{C}$	number of audio channels
CD	Compact Disc
CPU	Central Processing Unit (processor)
CQT	Constant Q Transform
δ	evaluation criterion
DFT	Discrete Fourier Transformation
DIN	Deutsches Institut für Normung
DMA	Deutsches Musikarchiv
DTW	Dynamic Time Warping
f	frequency in Hz
f_S	sample rate in Hz

FEAPI	Feature Extraction Application Programmer's Interface
FFT	Fast Fourier Transformation
FLTK	Fast Light Toolkit
GPL	Gnu General Public License
GUI	Graphical User Interface
$\mathcal{H}$	hop-size between STFT blocks
HMM	Hidden Markov Model
IBI	Inter-Bar-Interval
IEC	International Electrotechnical Commission
IOI	Inter-Onset-Interval
ISO	International Organization for Standardization
ITU	International Telecommunication Union
$\mathcal{K}$	block-size of STFT blocks
LADSPA	Linux Audio Developer's Simple Plugin API
$\mathcal{M}$	number of similarity matrix rows
MDS	Multidimensional Scaling
MFCC	Mel Frequency Cepstral Coefficient
MIDI	Musical Instrument Digital Interface
MIREX	Music Information Retrieval Evaluation eXchange
MPA	Music Performance Analysis
MPEG	Motion Picture Experts Group
$\mathcal{N}$	number of similarity matrix columns
$N'(z)$	specific loudness on the bark scale
NOT	Note Onset Time
$O(m)$	observation at block index m
$\omega(k, m)$	instantaneous frequency at bin k for block m

$OPCn$	n^{th} PCA component (variables: overall features, observations: performances)
$\mathfrak{p}$	MIDI pitch
P	alignment path
PAT	Perceptual Attack Time
PCA	Principal Component Analysis
PCn_F	n^{th} PCA component (variables: features, observations: performances)
POT	Perceptual Onset Time
PPM	Peak Programme Meter
RMS	Root Mean Square
$RPCn$	n^{th} rotated component from the selected $OPCn$ components
$\mathfrak{S}(n)$	score or audio event with list index n
SDK	Software Development Kit
SD	Semantic Differential
SIMD	Single Instruction Multiple Data
SSE	Streaming SIMD Extension (Intel)
STFT	Short Time Fourier Transformation
SVN	Subversion Version Control System
t	time in s
τ	MIDI tick in beats
$UPCn$	n^{th} PCA component (variables: performances, observations: features)
$v(n)$	low level feature at block n
VST	Virtual Studio Technology (Steinberg)
$x(i)$	audio signal amplitude at index i
$X(k,m)$	STFT result for signal x at bin k for block m

Chapter 1

Introduction

Music is a performing art. In most of its genres, it requires a performer or a group of performers who "self-consciously enacts music for an audience" [Slo85]. In classical or traditional western music, the performer renders the composer's work, a score containing musical ideas and performance instructions, into a physical realization.

Different performances of the same score may significantly differ from each other, indicating that not only the score defines the listener's music experience, but also the performance itself. Performers can be identified by listeners with regard to certain characteristics of their performances, and certain performers can be as famous as composers. A performance is a unique physical rendition or realization of musical ideas that is never just a reproduction but always a (new) interpretation. The performer is expected to "animate the music, to go *beyond* what is explicitly provided by the notation or aurally transmitted standard - to be 'expressive' " [Cla02b]. Bach explains [Bac94]

> Worinn aber besteht der gute Vortrag? in nichts anderm als der Fertigkeit, musikalische Gedancken nach ihrem wahren Inhalte und Affeckt singend oder spielend dem Gehöre empfindlich zu machen.

If different performances of the same piece of music are expected to represent the underlying musical ideas, why do they differ so clearly from each other, and what are the differences and commonalities between them?

For a better understanding of the role of music performances, it is helpful to consider the performance as embedded into a chain of musical communication starting at the composer and his score and ending with the listener, as shown in Fig. 1.1. The model is loosely based on Kendall's three-stage model featuring *Composer*, *Performer* and *Listener* [KC90]. The feedback paths indicate possible interrelations with the performance.

Obviously, no direct communication takes place between composer and listener. Instead, the composer translates his musical ideas into a score which is analyzed

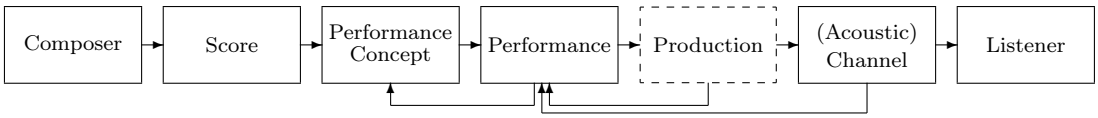

FIGURE 1.1: Chain of Musical Communication

by the performer to derive a performance concept or plan and finally to render the acoustic realization — the performance — that is subsequently perceived by the listener. Each of the communication stages allows or even enforces interpretation, modification, addition and dismissal of information. Chapter 2 provides more in-depth analysis of several of the communication stages within the context of music performance.

Music Performance Analysis (MPA) aims at obtaining a basic understanding of music performances. A good example for applied MPA, although highly subjective, are reviews of concerts and recordings that do not focus on the score information but rather on a specific performance or rendition of this score.

First systematic studies of music performance date to the beginning of the 20^{th} century, when mechanical and technical tools became available to record, reproduce and eventually to analyze music performances that previously had been unique, non-repeatable experiences. It was not only the reproducibility but also the objectivity of the newly available data that motivated researchers to discover music performances as a topic of scientific interest. Piano rolls for example — used to record and reproduce performances on mechanic pianos — proved to be excellent sources of detailed tempo and timing data for the recorded performances. Mechanical sensors and cameras allowed to track performance data such as hammer movements in pianos, and oscillographs and similar devices allowed the frequency analysis of recorded performances. The evolution of measurement devices, the introduction of MIDI (Musical Instrument Digital Interface) as a standard for control and recording of electronic musical instruments as well as the rise of digital approaches in signal recording, storage and analysis contributed to the development of the research field Music Performance Analysis. Especially during the last decade, new possibilities of data extraction and data mining were introduced and helped to simplify and speed up the process of analysis significantly. Despite all technical improvements, the main difficulties in performance research appear to remain the same as before:

- How to extract data that fulfills high demands on reliability, validity and external validity, i.e. the significance of the gathered data set, to allow general conclusions to be drawn?

- How to structure and interpret the extracted information in a musically, musicologically or psychologically meaningful way?

The first difficulty is actually a combination of problems; although for many pieces a nearly limitless number of recordings can be found, only the audio recording of these performances is available instead of detailed and accurate performance data provided by sensors frequently used in performance research. Since the "manual" extraction of performance data from audio is time-consuming, automated data extraction by a software system can be used for previously impracticable large-scale analyses while providing objective and reproducible results. Recently, modern digital audio signal approaches have led to encouraging results in the context of audio content analysis. For example, the accuracy and reliability of high-level data extracted from audio signals increased significantly.

The aim of this work is to adapt and develop such approaches for the use in a software system for the automatic acquisition of performance data from audio recordings in a sufficiently robust and accurate way, and to make the extracted data easily accessible to the analyst. This will be referred to as a *descriptive* approach which presents characteristics and properties of musical performances, as opposed to an *interpretative* approach that would attempt to explain the results in their musical, psychological or other context. For example, it is neither the goal of a descriptive approach to reveal a concept of interpretation or a performance plan nor to assess performance quality or to develop models of performance reception by listeners.

For this purpose, we restrict ourselves to the analysis of recordings from professional music performances of pre-existent compositions, available in classical score notation and do not aim at the analysis of improvisation, sight-reading and rehearsals or music that does not stand in the western concert tradition. There are no restrictions on instrumentation or genre, but the focus lies on polyphonic or multi-voiced ensemble music performed by more than one musician.

To demonstrate the suitability of the presented system, an analysis of string quartet performances is undertaken. The analysis of chamber ensemble performances is a rather neglected object of study, and the current understanding of music performance is mainly gained from piano performances. The presented results can be used to verify if and how these insights can be transfered to ensemble music with non-keyboard instruments.

In summary, the main contributions of this work are the design and implementation of a software system dedicated to music performance analysis, the presentation of optimized methods for audio content analysis, and the performance analysis of string quartet recordings.

Chapter 2 is an introduction to music performance and its characteristics. Furthermore, it summarizes past and present approaches to systematic performance

research with a focus on the extraction, the properties and the interpretation of the investigated performance data.

Chapter 3 describes the algorithmic design of the software library for automatic tempo and timing extraction from an audio file utilizing a score representation of the piece of music. The algorithm is based on a Dynamic Time Warping approach that finds the optimal global match between discrete times of a performed audio recording and the note events of a quantized MIDI file, given a fitting similarity measure between audio and MIDI data.

Chapters 4 and 5 describe the selection, interpretation and implementation of various low-level audio features for the analysis of both musical dynamics and timbre variation in music performances.

Chapter 6 presents the implementation of the complete software system for music performance analysis which is split into two parts, the performance data extraction and sonification and visualization of the data.

A systematic study of 21 performances of a movement of Beethoven's string quartet No. 13 op. 130 can be found in Chap. 7. It investigates tempo, loudness and timbre characteristics extracted from commercial recordings with recording dates between 1911 and 1999. The final Chap. 8 summarizes and concludes this thesis.

Chapter 2

Music Performance & Performance Analysis

2.1 Music Performance

The chain of musical communication, depicted in Fig. 1.1, shows that the composer communicates musical ideas or information via the score to the performer. It should be clearly distinguished between the terms *musical score* and *music*. According to Hill, the score is not the music itself, but sets down musical information, together with indications on how this information may be interpreted [Hil02]. Other authors describe the score as one of a number of possible representations such as a Compact Disc (CD), recordings and written descriptions or see the score as a "blueprint for a performance" [Cla02b].

A score that stands in the tradition of western music history always contains information on pitch and (relative) duration of each note; almost always instructions on musical dynamics appear in the score as well. Other instructions for example on character, quality or specific ways to perform may also be found in the score. Some of the contained information is available only implicitly (e.g. information on the musical structure) or might be ambiguous or hidden, complicating its description and quantification (compare [Dor42], [Mey56], [Pal97], [BM99]).

All this information is subject to the performers' interpretation — they detect and evaluate implicit information, try to understand and explain performance instructions, identify ways to convey their understanding of musical ideas to the listener and transform the discrete score representation of pitch, duration and dynamics to continuous scales.

It can be observed that later scores tend to be more explicit in terms of performance instructions than earlier scores, indicating that composers tried to eliminate the unspecified or ambiguous information in the score [Dor42]. This

may be due to the increasing awareness of the fact that scores often take into account performance rules that may seem "natural" at the time of composition but may change over decades and centuries, possibly leading to unintended performances.

Although the literature on musical performance frequently conveys the impression that imprecision and restriction of the score representation is undesirable, the fact is that there can be no true or absolute interpretation. Music is a living art and constant re-interpretation of music representations is the artistic breath that gives music life.

Seashore introduced the idea of defining the expressive parts of a performance as deviations from a "neutral", mechanical score rendition [Sea38]. However, the assumption that all information on such a "neutral" performance is already contained explicitly in the score seems unlikely on second thought, as the understanding and interpretation of a score might require cultural, historical and musicological considerations as well.

Other authors defined a neutral performance as a performance that is *perceived* as mechanic (which may not be necessarily a mechanical performance [Par03]). A different suggestion had been that the required neutral reference performance should be a performance with "perfectly normative rubato (and the equivalent on all other relevant expressive parameters)" [Cla91], that is a performance that matches all standard expectations of the listener.

Although controlled deviations from such a (normative or subjective) reference are most definitely directly connected with the perception of musical expression, they should not be confused with the expression or expressive deviations, as these terms "usually refer to physical phenomena, that is, deviation in timing, articulation, intonation, and so on in relation to a literal interpretation of the score. This use should be distinguished from a more general meaning of expression in music" as the expression's domain is the mind of the listener or the performer [Gab99].

Every performance requires a concept or plan which can be created by either a rigorous or a rather intuitive and unsystematic analysis of the score (for instance for sight-reading). This analysis should probably not be seen as an independent process applied to the act of interpretation but as "an integral part of the performing process" [Rin02].

The performance plan is a mental representation of the music [Gab99] that is an abstract list of actions that may be realized in an indefinite number of ways and is specified only relative to the context [Slo82]. Both authors stress the importance of structural and other "musical" information for this performance plan, but it also has to contain all intentions of the performer on what to express or convey to the listener. Of course the performance plan is so closely related to the performance itself that in many cases it does not make sense to

treat them separately, and the following paragraphs will not always differentiate between the plan and the performance itself.

Every music performance is highly individual in both its production and its perception. Still, a list of parameters that the performance may depend on can be compiled. The number of influencing parameters on the performance (and the performance plan) itself is probably infinite; nevertheless, the following list attempts to describe the main influences that may explicitly or implicitly influence a musical performance (also compare [Dor42], [Slo82], [Slo85], [Pal97], [TH02], [Wal02], [Cla02b], [Cla02a], [Par03], [Jus03a], [Jus03b]).

- **general interpretative rules**:
  These are rules, conventions, or norms that every performance follows because it would be perceived as uncommon or even unnatural otherwise.
- **performance plan and expressive strategy**:
  A concept of interpretation as a list of actions that may be influenced by
  - *interpretation of musical structure* or shape, e.g. the question of how to successfully convey melody, phrases, etc. to the listener.
  - *addition of unexpectedness* or deviation from expected conventions or rules.
  - *stylistic and cultural context and rules* that may vary over time or between countries or follow "performance fashions" [Cla02b], including instruments or instrument characteristics (such as timbre), used tuning frequencies and temperaments, and typical performance styles with respect to articulation, ornamentation, vibrato styles, tempo, *rubato* styles, etc.
    This may apply for both the historic context (the time the piece of music was composed or premiered) as well as for the context at the time of the performance.
  - *musical mood and emotional expression* that the performer plans to convey to the listener.
  - *performance context* such as the expected audience, the style and performance plan of other performances and works in the concert program.
- **the performers' personal, social and cultural background**:
  A very broad category that includes e.g. previous performing and general experiences, teachers and mentors, attitude, manners and mannerisms, etc.
- **physical influences**:
  The auditory and motorical or — more generally — physical abilities of the performer, general human limitations (e.g. in timing precision,

breathing) as well as attributes of the musical instrument that can impose limitations on e.g. fingering, changing of hand positions etc. may lead to forced or unintended deviations from the performance plan.

- **rehearsal**:
  The rehearsal phase allows direct feedback on the performance plan and may also train some specific motorical abilities of the performer. It should be noted that a rehearsal can also be seen as a performance itself.

- **immediate influences**:
  Influences that may change the performance at the time of performance and may lead to a deviation from the performance concept such as

  - *runtime feedback control*, i.e. the feedback that the performer directly receives that may consist of auditory, visual, tactile, and other cues [Tod93]. This includes various parameters such as the instrument's sound and reaction, the performance of co-performers, the acoustics of the environment, the reaction of the audience etc.
  - *external influences* not directly related to the performance such as humidity, temperature, distractions, etc.
  - *"internal" influences* such as the emotional and physical state of the performers (stress, stage fright, fatigue, illness, etc.)

Expressive movements are sometimes also considered to be part of a performance, since performers may move in ways that are not directly related to the generation of sound but to the character of music. In the context of this dissertation, only the acoustical properties of a performance will be taken into account.

Four classes of acoustical parameters that can be used for the description or characterization of music performances have already been identified in the 1930s by Seashore [Sea38]:

- **tempo and timing**: global or local tempo and its variation, *rubato*, or expressive timing, subtle variation of note lengths in phrases, articulation of tones, etc.

- **velocity, loudness or intensity**: musical dynamics, *crescendo* and *diminuendo*, accents, tremolo, etc.

- **pitch**: temperament, tuning frequency, expressive intonation, vibrato, glissando, etc.

- **timbre**: sound quality and its variation resulting from instrumentation and instrument-specific properties such as bow positioning (string instruments).

Recorded performances can differ significantly from the live performance, even in the case of so-called live recordings ([Cla02a], [Joh02]). The reason is that more persons than the performers themselves, e.g. the producer and sound engineer, may influence the final result during this production stage. Furthermore, mechanical and technological restrictions enforce differences between an original and reproduced performance, but also open up new possibilities to improve a recorded performance in the post-production process. For example, it is established recording practice (at least in the context of classical music) to not only record complete performances and finally choose the "best", but instead to record several or many so-called *takes* of passages of the musical piece. The recording process can also involve repeated listening to the recorded takes and discussions on the performance with influence on the following performances. Afterward, it is decided which parts of these takes will finally be used on the published CD and these will be edited in a way that the cuts are inaudible. Having analyzed seven productions of Beethoven's 9^{th} Symphony, Weinzierl und Franke found between 50 and 250 cuts between different recording takes in each production; the average number of edits increased with the technical evolution [WF02]. Nowadays, Digital Audio Workstations allow to edit music signals at nearly any score position.

Microphones and their positioning as well as signal processing done by the sound and mastering engineers may impact the loudness, the timbre, the reverberation and other parameters of the recording. These "interventions" can also vary over time to artificially increase or decrease acoustical or performance-based effects (e.g. increase the loudness of a specific instrument for its solo part etc.). Maempel et al. give an overview on processing options and typical objectives in the post production context [MWK08].

The musician's and the producer team's influences are not distinguishable on the final product, for example the CD. Therefore, the resulting recording including the (post) production stage will be referred to as performance in the remainder of this text; this seems to be a valid approach as the artist usually states his final agreement with the recording.

It should be kept in mind that it might not only be the editing and processing that differentiate a recorded performance from a live performance, but also the possible adaptation of the performer to a different reception and expectation in the recording context [Cla02a]. However, these recordings represent one of the principal forms in which music has been available in the last and the current century.

The listener, as the receiving end point of the communication chain, subjectively interprets the music. He listens to a performance and conceives musical ideas and other information that is conveyed by the performance. Since the recipient is affected by the incoming information, at this point in the communication chain the subjective effects of a performance can be analyzed. As Lundin points out, the kinds of possible affective reactions are practically limitless [Lun53].

2.2 Music Performance Analysis

Music Performance Analysis (MPA) aims at studying the performance of a musical score rather than the score itself. It deals with the observation, extraction, description, interpretation and modeling of music performance parameters as well as the analysis of attributes and characteristics of the generation and perception of music performance. Three basic directions can be roughly distinguished in the field of systematic performance analysis:

- to study the performance itself: to identify common and individual characteristics in the performance data, general performance rules, or differences between individual performances
- to study the generation or production of a performance: to understand the underlying principles of performance plans, the relation of the performers' intention to objective performance parameters (see below), and to investigate the performers' motoric and memory skills
- to study the reception of a performance: to comprehend how performances or the variation of specific parameters are perceived by a listener, and to study how he is affected

MPA could on the one hand lead to more explicit formulations of the different (objective) performance characteristics in the practice of music-teaching or enable the development of teaching assisting systems that give the student direct and objective feedback on the performance parameters. On the other hand, it could assist the implementation of performance models that generate computer renditions of human-like music performances. MPA also gains insights that can be valuable for the investigation of music esthetics and music history.

One of the problems of MPA is to define a suitable reference that the extracted performance data may be compared to. While a *mechanical* rendition seems to be an obvious choice as reference, other reference renditions such as a (human) performance that attempts a mechanical rendition, a rendition that is perceived to be mechanical, a rendition that is perceived to be standard or common, or an average rendition calculated from many performances could be considered to be more meaningful reference renditions. However, in the latter cases the reference renditions can only be valid in a specific context. This will usually not be desirable from the analyst's point of view.

As Clarke points out, "musical analysis is not an exact science and cannot be relied upon to provide an unequivocal basis for distinguishing between errors and intentions" [Cla04], emphasizing the challenge of meaningful interpretation of extracted performance data. A related difficulty that music performance analysis has to deal with is to distinguish between inherent performance attributes and individual performance attributes. In the context of musical

accents, Parncutt [Par03] distinguishes between *immanent accents* that are assumed to be apparent from the score (structural, harmonic, melodic, metrical, dynamic, instrumental) and *performed accents* that are "added" to the score by the performer. This approach may be applied to nearly all extracted parameters, and in the general case it might not be possible to distinguish score-inherent and performer-induced characteristics.

The interpretation of importance and meaning of characteristics derived from performance data is a difficult task. In the end, final conclusions can only be drawn by taking into account subjective judgments. The methodology and questionnaire or rating scale for such subjective tests and how they can be taken into account, however, has only begun to evolve to systematic approaches during the last centuries. The problem of extracting relevant characteristics is apparent in the design of systems intended to automatically generate music performances from a score. Clarke notes (in the context of parameters possibly influencing performances): "Whatever the attitude and strategy of different performers to this wealth of influence, it is clear that a theory of performance which is presented as a set of rules relating structure to expression is too abstract and cerebral, and that the reality is far more practical, tangible and indeed messy" [Cla02b].

Different areas of research contribute to the field of MPA, including musicology, (music) psychology and engineering. An introduction to the research field is given by Clarke [Cla04]. Articles providing extensive overviews have been compiled for example by Gabrielsson [Gab99], Palmer [Pal97] and Goebl et al. [GDP+05]. The following sections do not reiterate these but intend to give an impression on the variety of different approaches to the analysis of music performance.

There are several possibilities to structure the available literature on musical performance analysis. The publications have been grouped depending on different characteristics of method and methodology, although this may lead to multiple citations of the same publications.

2.3 Analysis Data

2.3.1 Data Acquisition

The acquisition of empirical data is one of the crucial points in systematic music performance analysis. Among the various methods that have been proposed and used to acquire data, two general approaches can be identified: monitoring performances (or performance parameters) by mechanical or technical devices, or extracting the parameters from an audio recording of the performance. Both concepts have inherent advantages and disadvantages.

The monitoring approach usually provides accurate and detailed results since the measurement devices can track the performance parameters more or less directly, but the analysis is exclusively restricted to specific performances that were produced under special conditions and with the specific performers that were available.

The direct extraction of performance parameters from the audio — as opposed to from the instrument with sensors — is difficult and most definitely results in less accurate data. This is true for both the manual annotation of audio (such as marking note onset times) and the fully automated extraction of data. Additionally, some parameters of interest may be even impossible to extract from the audio, such as information on piano pedaling or note-off times. Other parameters of interest such as the performers' movements are obviously not extractable from the audio at all.

The advantage of extracting parameters directly from the audio signal is the possibility to analyze an enormous and continuously growing heritage of recordings, including outstanding and legendary performances recorded throughout the last century and until now. Hence, audio-based approaches allow to widen the empirical basis considerably with respect to the amount of available sources and their significance.

Audio Content Analysis, an increasingly important branch of Music Information Retrieval, deals with the automatic extraction and analysis of (musical) information from digital audio signals. The majority of the published algorithms work "blind", meaning that they only have audio data available as input information while any additional input such as the score representation of the analyzed music is not available. Thus, most of these systems aim at the extraction of *score* information from the audio rather than the extraction of *performance* information (such as so-called transcription systems). This is however not a technical necessity, so similar approaches can be utilized to extract performance information directly from the audio signal. The increasing accuracy and robustness of these systems will make such approaches more and more important for MPA.

2.3.1.1 Piano or Keyboard Performance

The introduction of mechanical pianos at the end of the 19^{th} century made the acquisition of objective performance data possible through piano rolls. For example, Hartmann [Har32] presented an early analysis of tempo and timing of two piano performances based on their piano rolls. There are also later approaches to the analysis of performance data from piano rolls [Dov95].

Other historic approaches used proprietary sensors that were built to extract performance data. The most prominent example is the *Iowa Piano Camera* that was used by Seashore [Sea38] and his team at the University of Iowa in the

1930's. For each piano key, this "camera" recorded onset and note-off times and hammer velocity by optical means. Another example of a proprietary system is Shaffer's Bechstein grand piano [Sha84], using photo cells to detect hammer movements.

The introduction of the MIDI (Musical Instrument Digital Interface) specification (latest revision see [MID01]) in the 1980's resulted in an increasing number of electronic instruments and MIDI sequencers as well as computer hardware and software solutions that supported this specification and opened up new possibilities to measure, store and analyze pianists' performance data. Partly, music performance research has been done with the help of electronic instruments such as synthesizer keyboards and electronic pianos ([Pal89], [DH94], [Rep96b]), but the majority concentrated on using acoustic instruments with built-in sensors that automatically output MIDI data such as the Yamaha Disklavier product series or Bösendorfer grand pianos with the so-called *SE-System* ([Rep96a], [Rep96d], [Rep96c], [Rep97c], [Rep97a], [Bre00], [SL01], [Goe01], [WAD+01], [Sta01], [Wid02], [WT03], [Wöl04], [WDPB06], [TMCV06]).

As already pointed out, the analysis of performances that have not or cannot be recorded on specifically equipped instruments has to be based on the audio data itself. This is the case for the vast majority of available recordings.

To extract the tempo curve from an audio recording, the usual approach is to either tap along with the performance ([DG02], [Hon06]) or to manually annotate the onset times in a wave editor/display or a similar application ([Pov77], [Rep90], [Rep92], [Rep97b], [Rep98], [Rep99a], [Rep99b], [Wid95a], [Wid95b], [Wid98a]). Both approaches have also been automated or partly automated by the use of automatic beat tracking systems — followed by manual correction of beat times — ([Wid98b], [ZW03]/ [WZ04], [Tim05], [DGC06]) or more recently by alignment algorithms using score or MIDI data as additional input ([Ari02], [MKR04], [DW05]). The main difference between tap-along and beat-tracking approaches as compared to manual onset time annotation and alignment systems is that in the former case the resulting tempo curve resolution is on beat level, meaning that between-beat timing variations cannot be analyzed, while the latter usually takes into account each single note onset time, whether this note lies on the beat or not.

A focus on piano performances can be observed in the literature. One of the obvious reasons is that the piano is a very common instrument with a large (solo) repertoire, but there are more reasons that make the piano an appealing choice. The tones produced by a piano have a percussive character that makes this instrument far more suitable for accurate timing analysis than for instance string instruments. Its mechanics make it possible to measure data with sensors less intrusive and probably easier than on other instruments that offer a more direct interaction between performer and sound production. Furthermore, the pianist is in some ways more restricted than other instrumentalists; he is limited to fixed (and equal-tempered) pitch frequencies, which rules out key or

harmony dependent intonation and other performance specifics such as vibrato. He also has little influence on the timbre of a played note, and after hitting a key, he is not able to control any of the typical note parameters such as pitch, loudness or timbre except its duration. From a technical point of view, these restrictions seem to make the piano a rather unattractive instrument with limited degrees of freedom, but even with these limitations, piano performances are an integral part of western cultural life, meaning that the mentioned restrictions do not really impede the communication of musical expression between pianist and audience. The reduction of possible parameter dimensions is however beneficial in performance research because it simply keeps the measurement dataset smaller. Last but not least, the (commercial) availability of electronic and acoustic instruments using MIDI as a universal communication protocol simplified the performance data acquisition significantly since custom-built solutions were no longer necessary. While the recording of MIDI data from other non-keyboard instruments is at least partly possible, the fact that MIDI is a keyboard-focused protocol results in limited usefulness in many cases. Despite the good reasons for the usage of piano as the main instrument for performance analysis, it has not yet been conclusively shown that the insights gained from piano performance analysis can be applied to performances with other instruments and ensembles (although the not-so-many studies on other instruments indicate that this might at least partly be the case).

2.3.1.2 Other Instruments or Instrumentations

Most non-piano instruments represented in the musical performance literature are monophonic, meaning that never two or more notes can occur simultaneously. In this case, common approaches to frequency analysis can be assumed to be robust enough to extract the pitch variation over time. Proprietary as well as commercially available systems have been applied to the task of pitch extraction form the audio signal ([Sea38], [Sch40], [Dil01], [FJP03], [Wal04], [Bow06], [Orn07], [Rap07], [MAG08], [RPK08]). Seashore invented the "Tonoscope" for the pitch analysis of monophonic signals [Sea02]. It consists of a rotating drum covered with a paper containing small dots, each representing a certain frequency. The input signal is — by the means of a light-emitting gas tube — projected on the rotating paper. If the input frequency matches one of the frequencies a dot represents, this line of dots will stand still for the observer and gives a clear indication of the frequency. The "Melograph" used in [Orn07] appears to be basically of a similar design. Other studies work with spectrogram visualizations, use commercially available software solutions for the detection of monophonic pitches, or implemented their own software algorithms for the pitch detection.

The majority of these systems are not able to extract note onset times, so tempo and timing information is either not analyzed or is extracted by manual annotation. However, to name two counter-examples, Kendall compared timing

and dynamics of monophonic melodies performed on piano, clarinet, oboe, violin, and trumpet [KC90] and Ramirez et al. used automatically extracted timing data for the identification of performers of violin recordings [RPK08].

The tempo and timing data for other, non-monophonic signals has usually been extracted by tapping along (e.g. [Hon06]) or by manually setting onset time labels (e.g. [Ras79], [Jer03a], [Jer04]). Clynes [CW86] did not analyze the tempo on a beat or onset level but measured the overall duration of single movements.

2.3.2 Instrumentation & Genre

The majority of musical performance research focuses on the piano as the instrument of main interest.

Other individual instruments include the singing voice ([Sea38], [Sch40], [FJP03], [Rap07]), string instruments such as violin, viola, and violoncello ([Sea38], [KC90], [Dil01], [Bow06], [Orn07], [MAG08], [RPK08]), wind instruments such as flute, clarinet, oboe and trumpet ([KC90], [Wal04], [Orn07]), organ ([Jer03a], [Jer04]) and percussion instruments ([Dah00]).

Publications researching chamber music performances show up less frequently (e.g. [Ras79], [CW86], [Hon06]).

A large variety can be found in the style of musical pieces chosen for performance research. The date of composition of the analyzed musical pieces ranges from the 16^{th} to the 20^{th} century and a general focus on well-known and popular composers such as Bach, Mozart, Beethoven, Schumann, and Chopin can be observed.

2.3.3 Variety & Significance of Input Data

With respect to the question if and how reliably conclusions can be drawn from the extracted data, it is important to verify how and from whose performance this data has been generated.

For example, it could be argued that performance data gathered under "laboratory conditions" is insignificant per se due to the unnatural recording environment; however, these special conditions are also given for many (studio) recording sessions that resulted in recordings that are in fact perceived as convincing performances by the listeners, so we may disregard this point of view.

Still, when the data is acquired under such laboratory conditions, it implies that the number and possibly the skill of the available performers might be restricted. For example, research had partly been done on student performances (e.g. [Rep96a], [Rep96d], [Rep96c], [Rep97c], [Rep97b], [Bre00], [SL01], [Goe01],

[Wöl04], [Bow06]). This fact by itself is not too remarkable, but it nevertheless emphasizes the question if and how research methods and conclusions take into account the possible discrepancies between the performances of student pianists (or just *available* pianists) and the performances of professional and *famous* pianists. Under the assumption that fame is related to higher professional skills of the performer this could be a noteworthy criterion.

Due to the difficulties of acquiring large sets of performance data described above, the number of performers per study is usually small. The majority of research in the presented paper database has been done with a number of five or less performers per publication ([Har32], [Pov77], [Ras79], [Sha84], [KC90], [DH94], [HHF96], [Rep96b], [Bre00], [Dah00], [LKSW00], [Dil01], [GD01], [Shi01], [Sta01], [WAD+01], [Wid02], [Wid98b], [FJP03], [WT03], [Dil04], [Jer04], [WDPB06], [DGC06], [Hon06], [Rap07]) or six to ten performers ([Rep96a], [Rep96d], [Rep96c], [Rep97c], [Rep97a], [Rep99d], [ZW03]/ [WZ04], [Wöl04]). Examples of publications evaluating more performers are [Orn07] with 15 performers, [Rep90] and [Rep97b] with 19 and 20 performers, respectively, [Goe01] with 22 performers, [Rep92] (and using the same data set [Wid95a], [Wid95b], [Wid98a]) with 24 performers and finally [Rep98]/ [Rep99a]/ [Rep99b] with an outstanding number of 108 performers (115 performances). This raises the question if and how insights gained from a small group of performers can be extrapolated to allow general assumptions on performances.

Table 2.1 summarizes the characteristics of the analyzed data set for many of the cited publications. Although the usefulness of such a summary is obviously limited, it gives a quick overview of the data set properties.

Publication	*Title*	*Instrument*	*Composer*	*#Performers*	*#Pieces*	*#Performances*
[Bre00]	Articulation Strategies in Expressive Piano Performance	Piano	Mozart	5	1 (excerpt)	45
[CW86]	Music As Times Measure	String Quartet	Beethoven, Haydn, Ravel, Bartok, Janacek	1	5	39
[DH94]	Does expressive timing in music performance scale proportionally with tempo?	Keyboard (harpsichord sound)	Beethoven	2	1	6
[Dil01]	Extracting audio cues in real time to understand musical expressiveness	Recorder	Corelli	1	1 (excerpt)	5
[Dil04]	On the Recognition of Expressive Intention in Music Playing: A Computational Approach with Experiments and Applications	Piano	Mozart	2	1 (excerpt)	2
[FJP03]	The Influence of the Practice of Basso Continuo on the intonation of a Professional Singer in the Time of Monteverdi	Voice	Monteverdi	1	1 (excerpt)	$8 \cdot 3$
[GD01]	Analysis of tempo classes in performances of Mozart sonatas	Piano	Mozart	1/4	12/12 (excerpts)	12/48
[Goe01]	Melody lead in piano performance: Expressive device or artifact?	Piano	Chopin	22	2 (excerpts)	44+44
[Har32]	Untersuchungen über metrisches Verhalten in musikalischen Interpretationsvarianten	Piano	Beethoven	2	1	2
[Hon06]	Motor Action in Performance - Rostropovich and Richters Repeated Renditions of Prokofievs Cello Sonata Op.119	Violoncello & Piano	Prokoviev	1	1	2
[HTS02]	Effects of Musical Tempo and Mode on Arousal, Mood, and Spatial Abilities	Keyboard (Piano sound)	Mozart	1	1	4
[Jer03a]	Measurements and Models of Musical Articulation	Organ	Bach	5	4 (excerpts)	14
[Jus00]	Cue Utilization of Emotion in Music Performance: Relating Performance to Perception	Guitar	-	3	3 (short melodies)	36
[KC90]	The Communication of Musical Expression	Piano, Clarinet, Oboe, Violin, Trumpet	Purcell, Haydn, Schubert, Webern	5	4 (melody excerpts)	60

Continued on next page

Publication	*Title*	*Instrument*	*Composer*	*#Performers*	*#Pieces*	*#Performances*
[LKSW00]	Realtime Analysis of Dynamic Shaping	Piano	Satie	2	1	2
[Orn07]	An empirical study of intonation in performances of J.S. Bachs Sarabandes: temperament, ´melodic charge´ and ´melodic intonation´	Flute, Violoncello	Bach	15	2	15
[Pal89]	Mapping Musical Thoughts to Musical Performance	Keyboard (Piano Sound)	Mozart/Brahms	6/8	1/1 (excerpts)	2/8
[Pov77]	Temporal Structure of Performed Music. Some Preliminary Observations	Keyboard (Piano Sound)	Bach	3	1	3
[Rap07]	The Marvels of the Human Voice: Poem-Melody-Vocal Performance	Voice	Berlioz, Schubert, Puccini, Offenbach	7	4 (excerpts)	8
[Rep90]	Patterns of Expressive Timing in Performances of a Beethoven Minuet by Nineteen Famous Pianists	Piano	Beethoven	19	1	19
[Rep92] (also used in [Wid95a], [Wid95b], [Wid98a])	Diversity and commonality in music performance: An analysis of timing microstructure in Schumanns ´Träumerei´	Piano	Schumann	24	1	28
[Rep96a]	Patterns of note onset asynchronies in expressive piano performance	Piano	Schumann, Debussy, Chopin	10	3	$30 \cdot 3$
[Rep96b]	Pedal Timing and Tempo in Expressive Piano Performance: A Preliminary Investigation	Keyboard (Piano sound)	Schumann	2	1	18
[Rep96d]	The dynamics of expressive piano performance: Schumanns ´Träumerei´ revisited	Piano	Schumann	10	1	$10 \cdot 3$
[Rep96c]	The Art of Inaccuracy: Why Pianists Errors are Difficult to Hear	Piano	Schumann, Debussy, Chopin, Grieg	10	4	$40 \cdot 3$
[Rep97c]	The effect of tempo on pedal timing in piano performance	Piano	Beethoven, Brahms	10	2 (excerpts)	$20 \cdot 3$
[Rep98], [Rep99a]	A microcosm of musical expression. I. & II.	Piano	Chopin	108	1 (excerpt)	115
[Sch40]	The Psychology of Music	Voice	Bach-Gounod	5	1	5

Continued on next page

Publication	*Title*	*Instrument*	*Composer*	*#Performers*	*#Pieces*	*#Performances*
[Sea38]	Psychology of Music	Violin/ Piano/ Voice	Bach, Schubert-Wilhelmj/ various/ various	9/ various/ 9	5/ various/ 7	12/ various/ 10
[Sha84]	Timing in Solo and Duet Piano Performances	Piano/Piano Duet	Chopin	1/1	1/1	3/2
[Shi01]	Three expressive strategies of expert performance of an excerpt by J.S. Bach	Violoncello	Bach	3	1	3
[SL01]	Tracking Performance Correlates of Changes in Perceived Intensity of Emotion During Different Interpretation of a Chopin Piano Prelude	Piano	Chopin	10	1	20
[Sta01]	A Computational Model for Discriminating Music Performers	Piano	Mozart	2	7 (excerpts)	14
[Tim05]	Predicting the similarity between expressive performances of music from measurements of tempo and dynamics	Piano	Chopin, Mozart	11	2 (excerpts)	11
[TMCV06]	Listeners emotional engagement with performances of a Scriabin étude: an explorative case study	Piano	Scriabin	1	1	3
[Wid02], [WT03]	Machine Discoveries: A Few Simple, Robust Local Expression Principles/Playing Mozart by Analogy: Learning Multi-level Timing and Dynamics Strategies	Piano	Mozart	1	13	13
[WAD+01]	The timing of grace notes in skilled musical performance at different tempi: a preliminary case study	Piano	Beethoven	1	1 (excerpt)	$9 \cdot 5$
[ZW03]	Expressive Timing and intensity profiles in mental performances	Piano	Beethoven	6	12	48

TABLE 2.1: Overview over the analyzed data set in selected MPA publications

2.3.4 Extracted Parameters

The basic classes of objective performance parameters have been identified by Seashore in the 1930s as tempo and timing, pitch, dynamics, and timbre [Sea38].

The variation of tempo and timing over time is one of the most thoroughly researched aspects in MPA. The extracted onset times are usually converted into relative inter-onset-intervals (IOI) by calculating the discrete derivative. Then, each data point is normed by the corresponding note duration from the score in beat. The resulting curve of normed IOIs is an inverted representation of the tempo with the unit $^{s}/_{Beat}$ (as opposed to the usual musical tempo definition in $^{Beat}/_{s}$, compare Chap. 3). The analysis of the *articulation* is in most cases restricted to keyboard performances that have been captured in MIDI format. Articulation is then simply interpreted as a measure of performed note overlap or note duration with respect to the score note duration.

In order to analyze the musical dynamics in a performance, the level or loudness over time is extracted using sound intensity or psycho-acoustically motivated loudness measurements. Strictly speaking, such measurements do not correspond directly to musical dynamics as these would depend on the musical context, on the instrument or instrumentation, on the timbre, etc. Nevertheless, intensity and loudness measurements seem to be a good approximation to dynamics (see e.g. [Nak87], [Ger95], Chap. 4).

Pitch-related performance parameters such as vibrato and intonation can be directly analyzed by extracting a fundamental frequency or pitch curve from the audio signal. Due to technological restrictions of current analysis systems for polyphonic music, this usually has been limited to monophonic input signals.

The analysis of timbre deviations in performances is probably one of the least-researched parameters in MPA. This may be on the one hand due to the multidimensional nature of timbre (compare Chap. 5), on the other hand because it is assumed to be of least importance and partly of high correlation with dynamics.

2.4 Research Results

2.4.1 Performance

Many studies focus on a rather descriptive approach to performance analysis by just analyzing extracted data such as the tempo curve ([Har32], [Sea38], [Pov77], [Sha84], [Pal89], [Rep90], [Rep92], [Rep98]) or the loudness/intensity curve ([Sea38], [Rep96d], [Rep99a], [Shi01]) to identify attributes of the extracted parameters between different performances and performers.

The relation of musical structure (melodic, metric, rhythmic, harmonic, etc.) or the musical gestalt to tempo and loudness deviations has been intensely researched ([Har32], [Sha84], [Slo85], [DH93], [Rep96d], [Kru96], [Pal97], [Rep99a], [LKSW00], [TADH00], [Hon06], [WDPB06]). Most authors agree on the close relationship between musical structure such as musical phrases or accents and performance deviations mainly in tempo and timing. In particular, larger tempo changes seem to be most common at phrase boundaries. There is a general tendency to apply *ritardandi* or note lengthening at the end of a phrase and moments of musical tension ([Pal89], [Rep90], [Rep92], [Rep98]). Shifres found indications that the loudness patterns are used to outline more global structural levels while *rubato* patterns have been mostly used for local structural information in his test set [Shi01]. Some of these systematic deviations, both in timing and dynamics, are apparently even applied — although less prominent — if the performer is asked to deliver a "mechanical" rendition (that is, with constant tempo and dynamics) of the musical piece (see [Sea38], [Pal89], [KC90]).

Repp found a coupling of timing and dynamic patterns [Rep96d], but in a later study, he only found weak relationships between timing and dynamics [Rep99a].

Desain et al. and Repp report on the influence of overall tempo on expressive timing strategies ([DH94], [Rep95]). They find that the concept of relational invariance cannot be simply applied to expressive timing at different tempi, a result similar to Windsor's [WAD+01], who analyzed tempo-dependent grace note timing. The overall tempo might also influence overall loudness [DP04], an effect that they link to the increasing amplitude of pianists' vertical finger movements toward higher tempi.

Goebl [GD01] investigated the relationship of the composer's tempo indications (*andante*, *allegro*, etc.) with the "real" tempo and was not able to separate different tempo classes sufficiently with the tempo extracted from the performance. The number of note events per minute, however, seemed to be easier to map to the tempo indications.

Studies on the timing of pedaling in piano performance can be found in [Rep96b], [Rep97c]. The observations seem to be hard to generalize, but a relationship between pedal timing and overall tempo can be identified.

The articulation, or the amount of key (non-)overlap has been studied (in the context of keyboard instruments) in [Har32], [Pal89], [Rep97a], [Bre00] and [Jer03a]/ [Jer03b]/ [Jer04]. In summary, key overlap times for legato articulation seem to decrease with increasing Inter-Onset-Intervals (IOIs).

The accuracy of timing synchronization of two and more performers has been studied in [Ras79] and [Sha84], with the result that performers are highly capable of synchronizing onset times even when modulating the tempo over time. Other publications deal with the timing synchronicity between both hands or between the melody and the accompaniment in piano music [Har32],

[Sha84]. In many cases of piano performance, a lead of the melody before accompanying voices can be observed [Pal89], but whether this represents a performance concept or a consequence of the higher velocity of the melody tones is subject of discussion ([Rep96a], [Goe01]).

The evaluation of the consistency of repeated performances of the same performers has shown their ability to reproduce a rendition quite exactly in terms of timing ([Sea38], [Sha84]), dynamics ([Rep96d]), and pedal timing ([Rep96b]). This seems to be the case for performances spaced by several years as well ([Rep99a], [Hon06]). Only measuring the overall movement durations of several performances of the same ensemble over several years, Clynes found very stable overall tempi [CW86].

Performance data from student and professional performances has been compared in [Pal89] and [Rep97b]. While individual differences tended to be more pronounced among the professionals, both groups seemed to share the same general performance concepts.

Statistical and machine learning approaches have been tested to use the extracted tempo and loudness information for the purpose of classification, structuring the data or extracting general rules from the data. Dovey tried to extract general as well as individual rules from two of Rachmaninov's piano roll recordings by using Inductive Logic Programming [Dov95]. Supervised learners can be used to assign representations of the extracted performance data to the corresponding artists with promising results ([Sta01], [ZW03]/ [WZ04], [Dil04]). Other machine learning methods have been used to identify general performance rules ([Wid95a], [Wid95b], [Wid98a], [Wid98b], [Wid02], [WT03]) and to determine individual differences between artists [Wid98b].

Repp [Rep98], [Rep99a] investigated the (statistical) relationships between the extracted performance data and sociocultural variables such as the artists' gender, nationality, year of birth and recording date but, although some significant correlations could be found, pointed out that these results should be regarded with caution and that individual differences are likely to outweigh any sociocultural correlations.

Walker showed that instrumental timbre may influence several performance parameters such as timing, articulation, and dynamics [Wal04].

The analysis of vocal performances focuses frequently on the evaluation of vibrato rates and depth and the change or stability of pitch over time ([Sea38], [Sch40], [Rap07], [Bow06]) or other intonation characteristics of the performance ([FJP03], [Orn07]). Fletcher analyzed the vibrato (and other acoustical features) of flute players [Fle75].

2.4.2 Performer

While the publications listed above deal mainly with the analysis of the performance itself, the second area of musical performance analysis tries to determine the capabilities, goals, and characteristics of performers.

For example, Repp analyzed the kind of errors (i.e. pitch deviations from score) pianists make during a performance [Rep96c] and checked if and how severe they were perceived by listeners, coming to the conclusion that the errors concentrated in less important parts of the score in which they were harder to recognize.

The relationship between the performers' intentions and the parameters extracted from performances has been studied in various ways. Palmer found good correspondence between notated intentions with respect to melody and phrasing and the extracted timing parameters [Pal89]. Also, systematic relationships between intended emotionality of the performance and the performance data (that is, representations of loudness and timing) can be detected ([Jus00], [Dil01]/ [Dil03]/ [Dil04]).

Other studies investigate the importance of the feedback of the music instrument to the performer (see e.g. [Slo82]); there have been studies that report on the effect of deprivation of auditory feedback ([Rep99d], [Wöl04]), investigated the performers' reaction to delayed or changed auditory feedback ([PP02], [FP03], [Pfo05]) or evaluated the role of tactile feedback in a piano performance [GP08].

Publications on the nature of memorization and learning of a musical piece (or its performance) tried to identify differences between novice and expert performers [DP00], to learn more on the nature of performance memory itself ([Pal00], [MP03], [Pal06]), and to find out more on the relation between a real and a virtual, imagined performance [Wöl04].

2.4.3 Recipient

It is the listener of a music performance who ultimately consumes, interprets and probably judges a music performance. Overall judgment ratings of performance data have been evaluated in various studies. In an early publication, Repp reported some significant relations of ratings to measured timing patterns [Rep90], while in a later study he had to conclude that "the aesthetic impression of the original recordings rested primarily on aspects other than those measured (such as texture, tone, or aspects of timing and dynamics (...))" [Rep99b]. Timmers did a similarity rating experiment and concluded that performances are judged in other ways than generally used to represent performance data [Tim05]. In [Tim01], she let listeners rate the goodness of fit of two succeeding parts of different performance pairs. Kendall investigated the communication of three levels of expressiveness: without expression, with appropriate expression, and with exaggerated expression [KC90]. Listeners were in many cases able

to identify these three levels. Thompson et al. investigated the variation of listener ratings for a performance over time and found that the listening time to reach a decision was typically in the short range of $15 - 20\,\mathrm{s}$ [TWV07].

The difficulties of studying emotional affection of the listener of a music performance are discussed by Scherer [Sch03a], who criticizes "the tendency to assume that music evokes 'basic' or 'fundamental' emotions" such as anger, fear, etc. Despite such difficulties in approach and methodology, many attempts have been made to investigate the relationship between emotional affections and objective performance data. For example, Juslin detected relationships between moods and tempo and loudness cues [Jus00], and Kantor reported indications of associations of such cues and emotional reactivity [Kan06]. Similar conclusions have been drawn in [SL01] and [Sch04] from studying the time-varying emotional valence or the arousal and its relationship with performance data. Timmers found strong correlations between the dynamics and listener's judgments of emotionality [TMCV06] and very good communication of emotional activity between performer and listener [Tim07a]. In another study, she examined the influence of recording age and reproduction quality, observing that judgments of age and quality changed strongly with the recording date, in contrast to the perceived emotion that were mostly independent of the recording date; the communication of emotional valence tended to be more restrained for old recordings [Tim07b]. Husain varied the tempo and the mode (major, minor) of a performance and found indications that tempo modifications had an effect on arousal and mode modifications on mood [HTS02]. Krumhansl evaluated the influences on timing and loudness variations on judgments of musical tension and found a close relationship of musical structure with both the listeners' musical tension rating and the performance data [Kru96].

The tempo perception of a music performance has been studied by Dixon, who found listeners to prefer smoothed beat sequences over the performed ones [DGC06]. Lapidaki investigated the dependency of the initial tempo of a performance on the preferred tempo of a musical piece [Lap00]; he found a general dependency, but he also identified a group of listeners that were able to come to very consistent tempo preferences. Repp found systematic deviations between the tapping of listeners and metronomical time of music events, a result that seems to correspond well with the performers' inability to render a performance mechanically [Rep99c]. Aarden reported dependencies between tempo and "melodic expectancy" [Aar06].

Of course, there are many more research angles from which music performance can be studied. For example, the impact of visual performance cues on judgments of tension and musical phrasing can be found in [TCV03], [DP06], and the brain activation at listeners of music performances is measured in [NLSK02]. Furthermore, the design of computational models of music performances is a closely related topic of research. Most prominent is the *KTH Model* or the

KTH rule system developed at KTH[1] over the last 30 years (compare [SAF83], [FBS06]). Other models have been published by Todd (e.g. [Tod92], [Tod95]) and Mazzola et al. (e.g. [MZ94]). More recently, Widmer et al. proposed an automatically trained model for music performance [WT03].

2.5 Software Systems for Performance Analysis

The number of complete software systems dedicated to music performance analysis is limited. In most cases, research focuses on the extraction of single performance parameters.

POCO [Hon90] is a software for the analysis and automatic generation of music performances. It is a comparably old system that still is frequently used by a group of researchers at the Music Cognition Group of Amsterdam. It seems to have rather comprehensive analysis functions but is restricted to MIDI (or other symbolic) input data.

An early approach to extract performance data from audio signals while utilizing a MIDI representation of the score was proposed by Scheirer [Sch95]. Scheirer, targeting the analysis of piano performances, used filter bank outputs combined with an onset detection algorithm to extract timing and velocity data. The system has apparently not been used for performance analysis in later publications.

The work at the ÖFAI (Austrian Research Institute for Artificial Intelligence) by Widmer, Goebl, Dixon et al. (see selected publications above) has introduced a variety of tools to extract performance data from audio signals that in combination probably comes closest to a complete state of the art system for music performance analysis. Some of their individual tools are available online, but they remain individual components for performance analysis rather than an integrated system.

Dillon [Dil04] presented a software system for music performance analysis that does work on audio input, but targets mainly at subjective aspects of performance analysis such as the recognition of "expressive intentions" and the "detection of arousal". The audio processing itself is — as it is considered to be only one small part of a bigger system — relatively simple; it aims at monophonic input sources and is therefore probably not too suitable for the analysis of polyphonic audio input.

[1] Royal Institute of Technology, Sweden

Chapter 3

Tempo Extraction

Tempo and Timing are among the most important performance parameters. Musical tempo is usually given in the unit *beats per minute* (BPM), and can be defined as the rate at which beats, i.e. perceived pulses with equal duration units, occur [DH93]. From a more score-based point of view, two definitions of the beat duration are common, either as the denominator of the time signature of the musical score or simply as the length of a quarter note. Different representations of tempo are of interest in performance analysis, e.g. the overall tempo, the tempo variation over time, and its micro structure.

The measure of overall tempo is not in every case as simple as one would imagine at first glance: by dividing the overall number of beats by the length in minutes one receives a proper estimate of the *mean tempo*, but the result does not necessarily match the *perceived tempo* a listener would indicate; there is a difference between the mean tempo and the perceived tempo. Gabrielsson [Gab99] distinguishes between the *mean tempo* and the *main tempo*, the latter being a measure with slow beginnings or final *ritardandi* removed. Repp [Rep94] found good correlation of the mean value of a logarithmic Inter-Onset-Interval distribution with the perceived tempo. Goebl [GD01] proposes a *mode tempo* that is computed by sweeping a window over the histogram displaying occurrence of inter-beat intervals and selecting the maximum position as mode tempo. In most cases, the result should be similar to the position of the histogram maximum.

The tempo variation over time or the local tempo can be extracted by identifying the event time of every beat t_b and calculating the local tempo between beats i and $i+1$ by

$$BPM_{local}(i) = \frac{60\,\mathrm{s}}{t_b(i+1) - t_b(i)} \tag{3.1}$$

Alternatively, the time of every event t_o may be extracted, disregarding whether it is a beat or not, to calculate the local tempo between two events. In this case,

the distance in beats $\Delta\tau_{i,i+1}$ between two events has to be known to calculate the correct micro tempo.

$$BPM_{micro}(i) = \frac{60\,\mathrm{s}}{t_o(i+1) - t_o(i)} \cdot \Delta\tau_{i,i+1} \tag{3.2}$$

The latter has the advantage of not being restricted to the beat resolution and thus revealing the tempo micro-structure.

There have been many publications dealing with the automatic extraction of tempo from a digital audio signal. One group of common approaches can be summarized under the term "tempo tracking systems" or "beat tracking systems".

Scheirer [Sch98] presented a tempo extraction system using a bank of resonance filters that process envelope differences. This has the advantage of not requiring a dedicated onset detection, but leads to a quantized tempo histogram. A similar approach is used by Klapuri [Kla03].

In contrast to these systems, a frequently attempted approach is to extract the onset times of all musical events in a first processing stage, followed by an adaptive beat tracking engine to extract the tempo and the beat locations with the information provided by the series of onsets. Examples can be found in publications of Goto ([GM95], [Got01]), Dixon [Dix99] and Meudic [Meu02]. In more recent publications, Laroche [Lar03] as well as Peeters [Pee05] use dynamic programming techniques to determine the tempo curve from the onset locations.

All of these approaches have in common that they are "blind" in the sense that they do not have and do not require information on the analyzed audio material such as the overall number of beats. In the context of this work, this is not optimal because:

- these systems usually do not react very well to sudden tempo changes
- these systems usually try to find the best match between the resulting beats (or beat grid) and the extracted onset times, which may not be a correct assumption for frequent syncopations or rests
- additional information in form of MIDI (score) files is available and could easily be utilized to improve the accuracy of the results.

Hence, we are interested in an algorithm for the automatic synchronization of audio data with MIDI (or, more general, the score) data that associates each symbolic (score) event with its actual time of occurrence in the audio signal. In general, such approaches are usually called *Performance-to-Score-Matching* systems.

3.1 Performance to Score Matching

Performance-to-Score-Matching systems can be differentiated by their capabilities of real-time matching. Real-time systems are usually called *Score Following* systems, and non-real-time (or offline) implementations are referred to as *Audio-to-Score Alignment* or *Audio-Score Synchronization* systems.

Possible applications of such alignment systems could be (compare e.g. [SRS03]):

- linking notation and performance in applications for musicologists to enable to work on a symbolic notation while listening to a real performance
- using the alignment score as a distance measure for finding the best matching document from a database
- musicological comparison of different performances
- construction of a new score describing a selected performance by adding information as dynamics, mix information, or lyrics
- performance segmentation into note samples automatically labeled and indexed in order to build a unit database
- musical tutoring or coaching where the timing of a recorded performance is compared to a reference performance

3.1.1 Score Following

Historically, the research on matching a pre-defined score automatically with a performance goes back to the year 1984. At that time, Dannenberg [Dan84] and Vercoe [Ver84] independently presented systems for the automatic computer-based accompaniment of a monophonic input source in real-time.

In the following years, Dannenberg and Bloch ([BD85], [DM88]) enhanced Dannenberg's system by allowing polyphonic input sources and increasing its robustness against musical ornaments and by using multiple agent systems. Vercoe [VP85] focused on the implementation of learning from the real performance to improve the score follower's accuracy.

Baird et al. ([BBZ90], [BBZ93]) proposed a score following system with MIDI input (for the performance) that is based on the concept of musical segments as opposed to single musical events; the tracking algorithm itself is not described in detail.

Heijink [Hei96] and Desain et al. [DHH97] presented a score following system that takes into account structural information as well. It uses a combination of strict pitch matching between performance and score and dynamic programming.

While many of previously presented publications focus on the score following part rather than audio processing itself, Puckette and Lippe ([PL92], [Puc95]) worked on systems with audio-only input with monophonic input signals such as clarinet, flute, or vocals.

Vantomme [Van95] presented a monophonic score following system that uses temporal patterns from the performer as its primary information. From a local tempo estimate he predicts the next event's onset time and detects if the expected onset time matches the measured onset time within a tolerance. In the case of an 'emergency', he falls back to the use of pitch information.

Grubb and Dannenberg ([GD97], [GD98]) proposed, in the context of a monophonic vocal performance, a system that uses fundamental frequency, spectral features and amplitude changes as extracted features for the tracking process to enhance the system's robustness. The estimated score position is calculated based on a probability density function conditioned on the distance computed from the previous score event, from the current observation, and from a local tempo estimate.

Raphael published several approaches that make use of probabilistic modeling and machine learning approaches incorporating Markov Models ([Rap99], [Rap01], [Rap04]).

Cano et al. [CLB99] presented a real-time score following system for monophonic signals based on a Hidden Markov Model (HMM). They used the features zero crossings, energy and its derivative, and three features based on fundamental frequency.

Orio et al. ([OD01], [OLS03]) introduced a score-following system for polyphonic music that utilizes a two-level HMM that models each event as a state in one level, and models a signal model with attack sustain and rest phase in a lower level. They use a so-called *Peak Structure Distance* (PSD) that represents the energy sum of band pass filter outputs with the filters centered around the harmonic series of the pitch of the score event under consideration.

Cont [Con06] presented a polyphonic score following system using hierarchical HMMs that uses learned pitch templates for multiple fundamental frequency matching.

3.1.2 Audio to Score Alignment

The publications presented above deal with score-following as a real-time application. The following publications deal with the related topic of non-real-time audio to score alignment.

The importance of reliable pattern matching methods has already been recognized in early publications on score following and alignment; in most cases dynamic programming approaches have been used, see for example Dannenberg's publications on score-following mentioned above, and Large [Lar93].

Orio and Schwarz [OS01] presented an alignment algorithm for polyphonic music based on dynamic time warping that uses a combination of local distances (similarity measures). It uses the PSD [OD01], a *Delta of PSD* (ΔPSD) that models a kind of onset probability, and a *Silence Model* for low energy frames.

Meron and Hirose [MH01] proposed a similar approach with audio features that are relatively simple to compute and added a post-processing step after the dynamic time warping to refine the alignment.

Arifi et al. [Ari02], [ACKM04] proposed a system that attempts to extract multiple pitches segmented into onsets and performs a dynamic programming to align MIDI data to the extracted data. The algorithm has been tuned for polyphonic piano music.

Turetsky and Ellis [TE03] avoided the problems of calculating a spectral similarity measure between symbolic and audio representation by generating an audio file from the (reference) MIDI data and aligning the two audio sequences. For the alignment, a dynamic programming approach is being used as well.

Similarly, Dannenberg and Hu ([DH03], [HDT03]) generated an audio file from the MIDI file to align two audio sequences. They calculate the distance measure based on 12-dimensional pitch chromagrams (each element representing a octave independent pitch class). The alignment path is then calculated by a dynamic programming approach.

Shalev-Shwartz et al. [SSKS04] presented a non-real-time system for audio to score alignment that uses dynamic programming but additionally provides a training stage. Here, they derived a confidence measure from audio and MIDI similarity data and trained a weight vector for these features to optimize the alignment accuracy over the training set. The audio feature set contains simple pitch-style features extracted by band-pass filtering, derivatives in spectral bands to measure onset probability, and a time deviation from the local tempo estimate.

The alignment system of Müller et al. [MKR04] is also based on dynamic programming. It is targeted at piano music, but they claim genre-independence. For the pitch feature extraction, they used a (zero-phase) filter-bank based approach, with each band pass' center frequency located at a pitch of the equal-tempered scale; the filter outputs are used to extract onset times per pitch.

Dixon and Widmer [DW05] presented an audio-to-audio alignment tool for polyphonic music that works in pseudo-real-time with a modified dynamic programming algorithm. As similarity measure, a Spectral Flux grouped into semi-tone bands is used.

In summary, the standard approach to audio to score alignment consists of three major processing steps: the audio feature extraction that in most cases approximates a pitch-like representation, a similarity or distance measure that computes the vector distance between audio and symbolic (score) features,

and the actual alignment or path finding algorithm that is either based on a dynamic programming/time-warping approach or on HMMs.

3.2 Proposed Algorithm

The implemented algorithm makes use of this three-stage processing model, as there do not seem to be promising alternatives available. In the opinion of the author, the emphasis of previous research on audio-score alignment was on the evaluation and improvement of statistical models for path finding, while less energy has been put into both the extraction of significant features from the audio input and the computation of the similarity measure. Therefore, the development of the presented system focused on the first two steps. As path finding algorithm, a dynamic time warping approach has been chosen; a HMM would also have been an option, but is — at least in the context of a non-real-time system — a very similar approach that is not expected to be superior for this application (compare e.g. [OS01], [DH03]).

The following main conditions had to be considered during the algorithm's design stage:

- Polyphony and multi-timbre: the input data contains multiple voices that may play several different notes at the same time. A voice can be either one instrument or a group of instruments. Each voice's velocity and timbre cannot be assumed constant over time.
- Intonation: Neither tuning frequency nor temperament or time varying changes in intonation are known a priori.
- Robustness: The algorithm has to be robust against errors that are likely to occur due to the complexity of the analyzed data and possible performance errors.
- Target performances: emphasis should be on polyphonic chamber music with the option to analyze larger ensemble and orchestra performances.

In multi-voiced or polyphonic music, one has to deal not only with varying tempo, but possibly also with timing differences between voices, i.e. notes that theoretically start simultaneously according to the score, but have different onset times in reality. In this case, the question arises if one onset is more important for tempo extraction or if one should average over all onsets. Desain and Honing [DH94] argue that the onset that belongs to the melody note is the most important (leading to the underlying problem of melody identification from the score), while Repp [Rep90] simply takes the earliest onset as the most important one. Both approaches can be challenged with good arguments (compare e.g. [GP01], [Goe02]). In the case of music with several instrumentalists such as

chamber or symphonic music — the target input data for this algorithm — the theoretical number of pseudo-simultaneous onsets may theoretically be as large as the number of musicians in the ensemble, making it impossible to discriminate the onsets between notes or instrument groups. This is, however, probably not necessary, as intended deviations will only happen infrequently (e.g. with soloists). Hence, it is assumed that simultaneous score onsets appear approximately simultaneously in the analyzed audio as well, and that unintended deviations simply lead to a less precisely extractable onset time.

Figure 3.1 shows a flow chart of the algorithm for the extraction of the tempo curve. A complete audio file and the corresponding MIDI file are processed. Since the tempo of the audio file is extracted with respect to the MIDI data reference, the MIDI file should be quantized to a constant tempo and should not include possible (performance-based) deviations from this constant tempo.

The presented algorithm is also able to extract the alignment information between two audio signals or two signals, however, this functionality has not been implemented.

As depicted in the flow chart, the algorithm can be structured into the three processing blocks mentioned above, and an additional pre-processing block. The *Pre-Processing* stage extracts information from the audio file that is required at later processing stages, the *Processing* transforms audio and MIDI data into meaningful internal representations for the computation of the similarity matrix that represents a *Similarity Measure* between every pair of observations of MIDI signal and audio signal. Finally, the most probable global alignment path is computed through this similarity matrix in the *Tempo Curve Extraction* stage to receive pairs of onset times, which can then be directly converted to a tempo curve.

Each individual processing block is explained in detail in the following sections.

3.2.1 Definitions

The audio input signal is a series of samples $x_c(i)$ with $\mathcal{C}$ channels.The down-mixed signal is computed by averaging over the channels:

$$x(i) = \frac{1}{\mathcal{C}} \sum_{c=0}^{\mathcal{C}-1} x_c(i) \tag{3.3}$$

In the case of block-based audio processing with the block length $\mathcal{K}$, the block boundaries $b_{start,stop}$ are shifted by the hop size $\mathcal{H}$ (in samples), resulting in overlapping blocks. This means

$$b_{start}(m+1) = b_{start}(m) + \mathcal{H} \tag{3.4}$$

$$b_{stop}(m+1) = b_{stop}(m) + \mathcal{H} \tag{3.5}$$

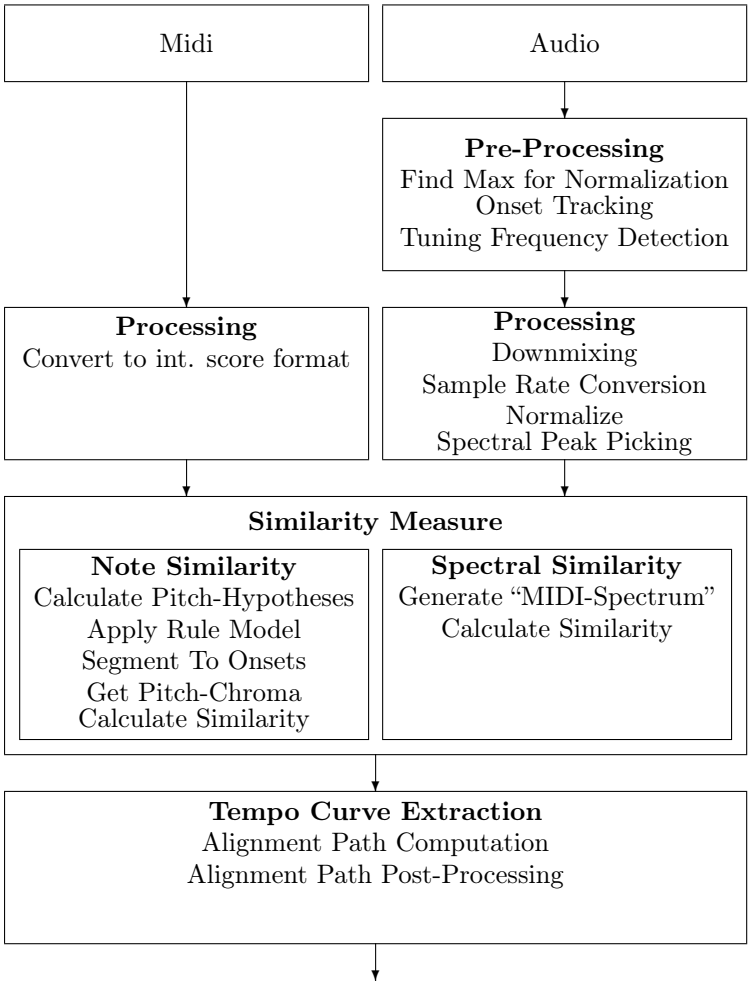

FIGURE 3.1: Flow chart of processing steps for the tempo extraction stage

The k^{th} bin of the *Discrete Fourier Transformation* (DFT, compare Chap. A.1) of the m^{th} block of signal x is then denoted $X(k, m)$. The DFT has an overall length of $\mathcal{K}$ bins. Since the DFT is calculated over a short block of data, it will also be referred to as *Short Time Fourier Transformation* (STFT).

3.2.2 Pre-Processing

This stage extracts information that is required in later processing stages; the overall maximum is required for audio file normalization (see Sect. 3.2.3.2), the extracted onsets for the onset-based pitch segmentation (see Sect. 3.2.4.2) and the overall tuning frequency for the correct frequency to pitch mapping (see Sect. 3.2.4). The whole audio file has to be processed before the next processing stage can be entered.

3.2.2.1 Overall Maximum

In order to be able to normalize all input audio files to the same maximum amplitude, the overall absolute amplitude maximum of the downmixed (and resampled, see Sect. 3.2.3.2) audio file is extracted.

$$x_{max} = \max_{\forall i} |x(i)| \tag{3.6}$$

3.2.2.2 Onset Extraction

The audio input is assumed to represent a series of musical sound events such as tones or the stroke on a percussive instrument. This assumption of a clear segmentation of the audio stream into distinct events is simplifying in general. Wright points out that musical meaning and even rhythm can be conveyed by audio streams with no clear division into distinct events [Wri08]. In our application where we only deal with western music we simply accept this simplified point of view because it is assumed to be valid for the majority of possible input signals, performances that are renditions of an event-based score format.

The start of a (musical) sound event is its onset. The term onset is frequently used as a synonym to onset time, but it should be more correct to state that its time position (i.e. the onset time) is one (most likely the main) property of the onset, while an onset can have other properties, e.g. its strength.

In most cases, the start of a musical sound is not an exact point in time, but a time span, the *rise time* or *initial transient time*. This is basically the time from the first instrument-induced measurable oscillation until either the quasi-periodic state or a maximum amplitude is reached. Other definitions such as the maximum envelope slope have also been used. The rise time can vary significantly between different musical instruments or groups of instruments, e.g. from about 5 ms for some percussive instruments up to 200 ms for woodwind instruments (flute) under certain circumstances [Reu95].

Three different definitions of onset times can generally be distinguished as pointed out by Repp [Rep96a]:

1. *Note Onset Time* (NOT): the time when the instrument is triggered to make a sound. In the MIDI domain, the NOT is exactly the time of the Note-On command. Depending on the instrument or sample used for sound generation, this is not necessarily the time when the signal becomes audible or detectable.

2. *Acoustic Onset Time* (AOT): the first time when a signal or an acoustic event is theoretically measurable. Sometimes the AOT is called *Physical Onset Time*.

3. *Perceptual Onset Time* (POT): the first time when the event can be perceived by the listener. The POT might also be distinguished from the *Perceptual Attack Time* (PAT), the time that is relevant for the rhythmic perception of the sound [Gor84]. While the PAT might occur later than the POT, they will be equal in many cases. For the sake of simplicity, there will be no distinction made between POT and PAT in the following.

The POT can never occur before the AOT, which never occurs before the NOT. Due to the "perceptual" definition of the POT, the exact location cannot be determined easily but has to be measured in a listening test. Gordon [Gor84] and Zwicker found strong location drifts of the PAT (here: POT) depending on the waveform properties during the rise time [ZF99].

Given the three definitions above, the question arises which of the three onset times should be assumed to be the reference onset time for the evaluation of the onset detection. Due to the symbolic nature of the NOT, it simply cannot be detected from the audio signal. The choice between AOT and POT might be application-dependent; assuming that musicians adapt their timing to their sound perception and that most ACA-Systems are trying to analyze the *perceptible* audio content, the POT is most likely the time that is wanted.

In order to estimate the required time accuracy of an onset detection system, the human ability to exactly locate onset times and to distinguish succeeding onsets is of great interest, since most algorithms are targeting to be at least as accurate as the human perception.

Hirsh found that temporal discrimination of two succeeding onsets is possible if the onset time difference is as little as 2 ms [Hir59]. However, in order to determine the order of the stimuli, their distance had to be about 20 ms. The measurements were done with synthetic signals with short rise times.

Gordon reported a standard deviation of 12 ms for the accuracy of onset times specified by test listeners, using 16 real-world monophonic sounds of different instruments played in an infinitely long loop pattern with Inter-Onset-Intervals (IOIs) of 600 ms [Gor84]. Friberg and Sundberg undertook a similar experiment using tone stimuli [FS92]. For IOIs smaller than 240 ms, they reported a just noticeable difference of about 10 ms, and increasing values for larger IOIs.

Repp reported for the manual annotation of onset times by one listener in the context of piano recordings a mean absolute measurement error of about 4.3 ms and a maximum error of about 35 ms [Rep92]. In a recent investigation, Leveau et al had three test subjects annotating the onset times in audio files of various genres and instrumentations. [LDR04]. The results showed a mean absolute measurement error over all test data of about 10 ms; for one piece of classical music, the mean absolute measurement error nearly reached 30 ms.

Rasch evaluated the onset time differences between instruments in three ensemble performances [Ras79]. He found synchronization deviations in a range

between 30 ms and 50 ms between the (string and woodwind) instruments, while the mean onset time differences were in the range of ±6 ms. However, it is complicated to distinguish between the accuracy of measurement and performance in this case.

For piano duet performance, Shaffer reported standard deviations within the voices between 14 and 38 ms [Sha84].

It may be concluded that the measurement accuracy highly depends on the used input data. Several publications imply that a reasonable demand for the detection accuracy of an automatic onset detection system cannot be smaller than in a range of $5 - 10$ ms and has to be as high as 50 ms or even higher for instruments with long rise times. Multiple quasi-simultaneous onsets may increase this range further.

Automatically extracted onset data can for example be used in applications for the detection of tempo, beat locations, time signature, automatic transcription as well as the segmentation of audio signals. Therefore, active research takes place in this field.

A typical onset tracking system initially extracts the derivative of an envelope representation of the audio signal. The resulting curve is smoothed over time, and the negative values will be discarded (half-wave rectification), as an "amplitude" increase is expected at the onset time. Finally, the locations of significant maxima, picked by a peak picking algorithm, are regarded as detected onset times. The following selection of publications gives an incomplete overview on different approaches to onset tracking.

Schloss presented an onset detection algorithm that makes direct use of the audio signal's envelope slope, extracted in the time domain [Sch85]. Newer publications usually make use of STFT-based techniques for the extraction of onset curves, based on the differences between succeeding (overlapping) STFT-blocks, sometimes grouped into frequency bands. While they can be partly distinguished by the number of frequency bands they analyze, their main difference is the used distance measure $d(m)$ between succeeding STFTs.

Most of the DFT-based distances measures are fairly similar. For example, Laroche [Lar03] used a *Flux*-based distance (with an additional square root function to increase lower signal amplitudes)

$$d_{lar}(m) = \sum_{k=k(f_{min})}^{k(f_{max})} \sqrt{|X(k,m)|} - \sqrt{|X(k,m-1)|}, \tag{3.7}$$

Duxbury et al. [DBDS03] proposed the distance between complex STFT bins

$$d_{dux}(m) = \sum_{k=0}^{\mathcal{K}/2-1} |X(k,m) - X(k,m-1)|, \tag{3.8}$$

while Hainsworth and Macleod [HM03] calculate a logarithmic distance

$$d_{hai}(m) = \sum_{k=0}^{\mathcal{K}/2-1} \log_2 \left(\frac{|X(k,m)|}{|X(k,m-1)|} \right). \tag{3.9}$$

Goto and Muraoka [GM95] proposed a distance that takes into account frequency variation over time. To do so, they identify all indices k with

a) higher power than the maximum of the four closest preceding bins

$$\begin{aligned} A &= X(k,m-1)^2 \\ B &= X(k-1,m-1)^2 \\ C &= X(k+1,m-1)^2 \\ D &= X(k,m-2)^2 \\ E_{max}(k,m) &= \max(A,B,C,D) \end{aligned} \tag{3.10}$$

and

b) the same condition fulfilled for the maximum power of the three closest succeeding bins

$$E_{k,m+1} = \max(X(k,m+1)^2, X(k-1,m+1)^2, X(k+1,m+1)^2) \tag{3.11}$$

The distance is then computed from the maximum of the current and succeeding power value $E_t(k,n) = \max(X(k,n)^2, X(k,n+1)^2)$ by

$$d_{got}(k,m) = \begin{cases} E_t(k,m) - E_{max}(k,m), & \text{if } (X(k,m)^2 > E_{max}(k,m)) \wedge \\ & (E_{k,m+1} > E_{max}(k,m)) \\ 0, & \text{otherwise} \end{cases} \tag{3.12}$$

$$d_{got}(m) = \sum_{k=0}^{\mathcal{K}/2-1} d(k,m) \tag{3.13}$$

The Goto-distance appears to be somewhat heuristically tuned, and will vary with changing ratio of DFT size and sample rate. However, pre-tests showed promising results for this distance, and a DFT-based onset tracking system has been implemented [Kli04] with a distance measure based on this Goto-distance [GM95]. Not only one single distance measure is calculated, but several measures per frequency band. The number of frequency bands is adaptively determined with a *Spectral Centroid* measure (see below) from the input signal. The peak picking algorithm, using a sliding threshold on the half-wave-rectified signal, is applied per frequency band. The detected onsets for each band are finally combined to one resulting series of onset times $\hat{t}_o(i)$.

	Klich/Lerch	**Goto et al.**	**Duxbury et al.**	**Hainsworth et al.**
DR	73.1%	70.1%	62.1%	46.4%

TABLE 3.1: Detection Rate DR of the used onset tracking system compared to other systems

Without going into detail on the evaluation methodology, database and metrics (which can be found in the corresponding M.A. thesis [Kli04]), Table 3.1 shows the summarized evaluation results of this onset tracking system compared to three others.

The presented detection rate DR is calculated as

$$DR = \frac{\#\text{ correct detections}}{\#\text{ correct detections} + \#\text{ missing detections} + \#\text{ wrong detections}}. \tag{3.14}$$

Additionally, a simpler onset measure is computed for later usage. The distance of this second onset measure is based on a Spectral Flux

$$d_{fl}(m) = \sum_{k=k(f_{min})}^{k(f_{max})} |X(k,m)| - |X(k,m-1)|. \tag{3.15}$$

The result is computed by the difference to a smoothed version $d_s(m)$

$$\zeta(m) = d_{fl}(m) - d_s(m) \tag{3.16}$$

and is normalized to its absolute maximum over all blocks. No half-wave rectification has been applied in order to determine not only "onset probability" but also "note-off probability".

3.2.2.3 Tuning Frequency Detection

The concert pitch or standard (musical) pitch is used for tuning one or more musical instruments and is defined to be the pitch $A4$. Its frequency, the tuning frequency, is standardized internationally to 440 Hz [ISO75], but the exact frequency used by musicians can vary due to various reasons such as the use of historic instruments or timbre preferences, etc.

For pitch based applications in the field of musical content analysis, various approaches to fundamental frequency detection have been proposed, but the mapping of frequencies to pitches is frequently regarded to be trivial, assuming the mid frequencies of the pitches to be tuned with reference to a standardized tuning frequency of 440 Hz for the pitch $A4$. The author argues in [Ler06] that for many applications of pitch extraction, a tuning frequency estimation — preferably automatic — is recommended to ensure robust pitch results.

The following systems have been proposed to find the best tuning frequency match automatically:

Scheirer used a set of narrow bandpass filters with their mid frequencies at particular bands that have been handpicked to match pitches from the analyzed score [Sch95]. These filters are swept over a small frequency range. The estimated tuning frequency is then determined by the frequency of the maximum filter output sum.

Dixon proposed a peak detection algorithm in the DFT domain, calculating the instantaneous frequency of the detected peaks, and adapting the equal-tempered reference frequencies iteratively until the distance between detected and reference frequencies is minimal [Dix96]. The adaptation amount is calculated by the low pass-filtered geometric mean of previous and current reference frequency estimates.

Zhu et al. computed a constant Q transform (CQT) with the frequency spacing of 10 cent over a range of 7 octaves [ZKG05]. The detected peaks in the CQT spectrum are grouped based on the modulus distance against the concert pitch. If the maximum energy of the resulting 10-dimensional vector is above a certain energy threshold, it is used for later processing. For the results of all processing blocks (if not discarded), a 10-dimensional so-called tuning pitch histogram is computed, and the tuning frequency is chosen corresponding to the bin with the maximum count.

Using a CQT with 33 cent frequency spacing, Harte and Sandler estimate the exact peak positions by interpolation [HS05]. A histogram of the peak positions based on the modulus distance against the concert pitch is computed over the length of the audio file, and the tuning frequency is set according to its maximum.

In the context of single-voiced input signals, Ryynänen added the modulus distance of detected base frequencies to a 10-dimensional histogram that is low pass-filtered over time [Ryy04]. Then, a 'histogram mass centre' is computed and the tuning frequency is adjusted according to this mass center.

A previous version of the following method for automatic tuning frequency detection that is described below has been published in [Ler04]. The input audio samples are processed by a filter bank of steep resonance filters. In the range of 2 octaves around the pitch $A4$, there are 24 groups of filters in (equal-tempered) semi-tone distance, with each group consisting of 3 filters. The mid frequencies of each group are spaced with 12 cent and the mid frequency of the centered filter is selected based on the current tuning frequency assumption. All filters have the same quality factor Q. The filter output energy per processing block of length 20 ms is then grouped based on the modulus distance against the concert pitch, resulting in a 3-dimensional vector E for each block i.

The symmetry of the distribution of the three accumulated energies gives an estimate on the deviation from the current tuning frequency compared to the assumption. If the distribution is symmetric, e.g. $E(0, i)$ equals $E(2, i)$, the assumption was correct. In the other case, all filter mid frequencies are

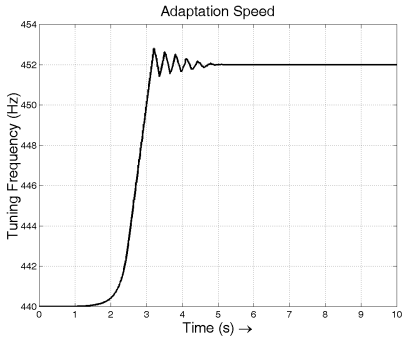

FIGURE 3.2: Adaptation of tuning frequency from initial setting of 440 Hz to target 452 Hz

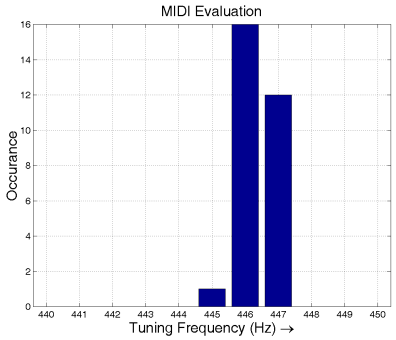

FIGURE 3.3: Distribution of results for the MIDI-generated test set tuned at 446 Hz

adjusted with the objective to symmetrize the energy distribution in the following processing blocks. The RPROP-algorithm is used as adaptation rule because it allows fast and robust adaptation without the requirement of specifically controlling the adaption step size [RB93]. The adaption rule for the adjustment of the assumed tuning frequency f_{A4} of the following processing block $i+1$ is:

$$f_{A4}(i+1) = \Big(1 + \eta \cdot \text{sign}\Big(E(2,i) - E(0,n)\Big)\Big) \cdot f_{A4}(i) \tag{3.17}$$

with η being scaled up if sign returns the same result as for the previous block, and scaled down otherwise. To ensure high accuracy, η is initialized with a small value. Figure 3.2 shows the adaptation from the initial tuning frequency 440 Hz to the real frequency 452 Hz. Adaptation is parameterized for accuracy rather than speed in this case, so it takes the algorithm more than 3 s to converge to the target frequency.

While this approach allows real-time processing and permanent adaptation to possibly varying tuning frequencies, in the current context the overall tuning frequency is computed by finding the maximum count in a histogram containing the estimates of all processing blocks. The histogram classes are spaced by one Hertz; while this is not completely consistent since, on the pitch scale, the width of these classes decreases slightly with increasing tuning frequency, it nevertheless was chosen considering that on the one hand, the deviations are small compared to the expected accuracy, on the other hand these class labels are the most transparent for the user when interpreting the result.

To verify the algorithm's accuracy, a test with a small database of 29 input files generated from MIDI content was performed. The files were generated with equal temperament and pitched to a tuning frequency of 446 Hz and were significantly longer than 10 s.

Figure 3.3 shows the result for this test set. The result is correct in a range of ± 1 Hz around the reference. Coincidently, this range roughly corresponds to the

just noticeable frequency difference humans are able to recognize ($2 - 4$ cent) [ZF99]. A possible explanation for the asymmetry of the results might be the deviations of the harmonics from the equal-tempered scale because all harmonics are interpreted as fundamental frequencies during pitch chroma computation.

The algorithm is expected to give slightly less accurate results when alternative temperaments are used.

3.2.3 Processing

The purpose of the processing stage is to convert MIDI and audio data into similar representations that can be utilized to calculate the similarity matrix between all time positions of the audio and MIDI file.

3.2.3.1 Internal Score Format

The choice of MIDI as input format [MID01] is more or less obligatory not because of superiority over other score formats (actually MIDI is relatively limited compared to 'real' score formats) but because of its popularity and spread. The MIDI reference file is parsed and the MIDI input data is converted to the internal score format. This score format contains, similar to the MIDI format, a stream of events. The stored information per note is mainly the MIDI pitch $\mathfrak{p}$ and its velocity $\mathfrak{v}$. A MIDI pitch can be computed from its fundamental frequency f by

$$\mathfrak{p}(f) = 69 + 12 \cdot \log_2 \left(\frac{f}{f_{A4}} \right) \tag{3.18}$$

Every increase of 1 then corresponds to a increase in pitch of one semi-tone on the equal-tempered scale. $\mathfrak{p}$ is usually given as integer value but is represented here in floating point format to allow an accurate pitch representation. The velocity $\mathfrak{v}$ is standardized for MIDI as integer value between $0 \ldots 127$; for the internal score format, it is mapped to a floating point value between $0 \ldots 1$.

While the MIDI-format features each single note independently, the internal score format combines simultaneous sounding notes into one entry, which will be referenced as *score event*. Every note onset or note-off time leads to a new entry, resulting in a sequence of score events that mark every change of polyphony and pitch over time. This format has the advantage of allowing simple access for the current use-case, e.g. for retrieving all sounding notes at a specific time. Still, it is also limited by some disadvantages of the MIDI representation that might have been useful in the context of performance analysis. These are mainly the lack of performance instructions in the score (dynamics, *ritardandi* and *accelerandi*, explanations, etc.) and the problem that ornaments and *arpeggii* are either not present in the MIDI data, or conducted only in one specific way of a nearly unlimited number of alternatives. More information on general

requirements of a score format for performance to score alignment systems can be found in [Sch03b].

The result of this processing step is a series of score events $\mathfrak{S}_S(n)$ extracted from the MIDI input data; the overall number of score events is $\mathcal{N}$. Each event contains the start time $t_S(j)$ and end time as well as a list of notes. Each note is defined by its pitch and its velocity.

3.2.3.2 Downmixing, Sample Rate Conversion and Normalization

Since all the required information for the tempo extraction process has to be available in a mono audio channel as well, the input channels are downmixed to a single audio channel, compare Eq. (3.3).

To ensure identical analysis time resolution for all kinds of audio input file formats, the sample rate of the incoming audio is converted to an internal processing sample rate. Currently, this internal processing sample rate is defined to be $f_S = 48\,\text{kHz}$. A so-called windowed sinc-based sample rate conversion is used that was implemented based on a publication by Smith and Gossett [SG84].

Level differences between different audio signals can be immense. These differences do not change the musical content that is relevant for the extraction of tempo information, but may influence the results of some of the later processing stages. This is unwanted as the same audio input at different levels should still lead to the same results. Therefore, the audio data is normalized:

$$x_{norm}(i) = \frac{x(i)}{x_{max}} \tag{3.19}$$

Dynamic differences between different sections of the audio file will remain, but this normalization is a simple and efficient way to reduce inter-file loudness differences.

3.2.3.3 Spectral Peak Picking

The objective of this processing step is to extract the tonal content in form of peaks from the audio data. This is done in the frequency domain by calculating a Short Time Fourier Transformation (STFT) of overlapping blocks of input data. To deal with the inherent problem of insufficient frequency resolution at low frequencies at a sufficient time resolution as defined by the DFT block-size, the STFT is calculated with the two different lengths $\mathcal{K}_s = 2048$ samples and $\mathcal{K}_l = 8192$ samples to obtain usable time resolution for higher frequencies with the shorter STFT, and better frequency resolution for lower frequencies with the longer STFT.

Each overlapping block of audio input data is processed as shown in Fig. 3.4.

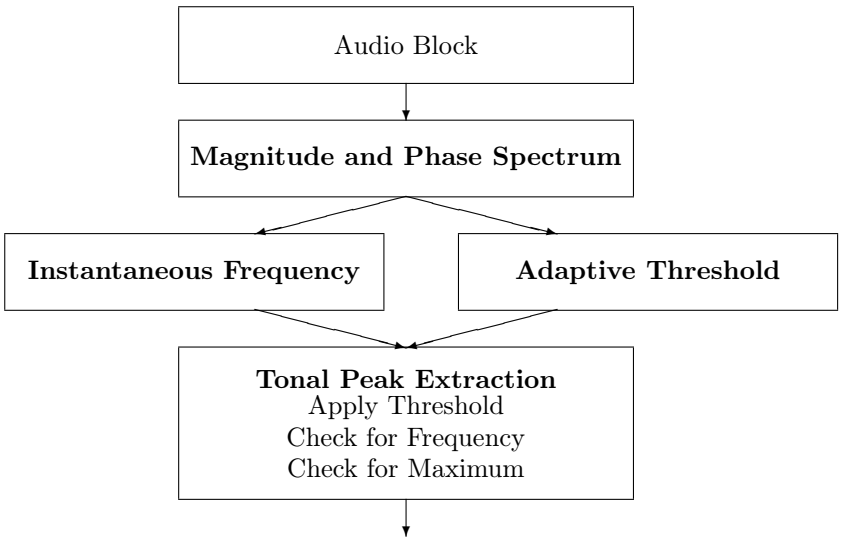

FIGURE 3.4: Flow chart of the Spectral Peak Picking Process

To qualify as peak, a spectral value has to have the following properties:

1. a bin frequency between pre-defined frequency thresholds
2. a prominent amplitude compared to the surrounding values to ensure it is of relative importance
3. good correspondence of instantaneous frequency with bin frequency to ensure tonality
4. a zero-crossing of the derivative to ensure it is a maximum

The first claim is addressed by restricting the search range, i.e. the start and stop bin.

The second property is assured by applying a threshold to the magnitude spectrum $|X(k)|$, with only values above this threshold are admitted as peak candidates. The threshold $T(k)$ is computed with

$$\begin{aligned} T'(k) &= \frac{1}{\mathcal{L}+1} \sum_{l=k-\mathcal{L}/2}^{k+\mathcal{L}/2} |X(l)| \\ T(k) &= \begin{cases} T'(k), & T'(k) > \mathfrak{c} \cdot \max\limits_{0 \le k \le \mathcal{K}/2-1} |X(k)| \\ \mathfrak{c} \cdot \max\limits_{0 \le k \le \mathcal{K}/2-1} |X(k)|, & T'(k) \le \mathfrak{c} \cdot \max\limits_{0 \le k \le \mathcal{K}/2-1} |X(k)| \end{cases} \end{aligned} \tag{3.20}$$

The threshold is calculated by a moving average filter of length $\mathcal{L}$. The constant $\mathfrak{c}$ guarantees that the minimum threshold value depends on the maximum value

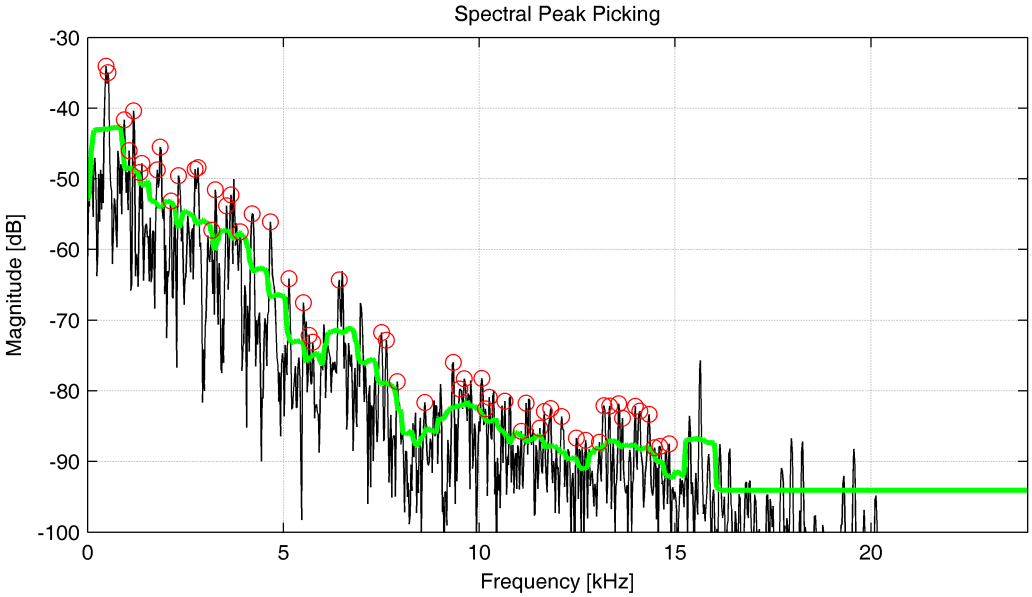

FIGURE 3.5: Spectral Peak Picking for one block of audio data; the magnitude spectrum is drawn in black, the adaptive threshold in green, and the detected peaks are highlighted in red

of spectrum $|X(k)|$ (currently, the threshold is not allowed to drop more than 60 dB below the maximum). The implementation features also an absolute threshold minimum to avoid unnecessary peak detection e.g. during pauses. The resulting threshold can also be understood as a rudimentary approximation to an psycho-acoustic masking threshold as used in perceptual audio coding (compare [Ler08]).

The third claim is addressed by using a phase-based estimator for the frequency of sinusoidal components (see [LM07]) and comparing the results with their corresponding bin frequencies. Phase-based estimators make use of the fact that the (instantaneous) frequency can be computed by the derivative of the phase. This means that an instantaneous frequency ω at bin k can be calculated by utilizing the unwrapped phase difference of two succeeding STFT blocks at this bin.

$$\omega(k,m) = \left[\frac{\mathcal{K}}{2\pi\mathcal{H}} \cdot \Big(\phi(k,m) - \phi(k,m-1)\Big)\right]_{unwrap} \tag{3.21}$$

The more this calculated instantaneous frequency deviates from the corresponding bin's frequency, the lower the likelihood will be that this is a tonal or sinusoidal component, but either a noisy component or a side lobe component due to DFT-windowing. Therefore, a bin will only considered to be a peak candidate for small deviations between instantaneous frequency and bin frequency.

The fourth claim is addressed by selecting only those values that are true local maxima, meaning that both neighboring bins have lower amplitudes than the value under consideration.

Figure 3.5 shows an example of a magnitude spectrum, the corresponding threshold, and the selected peaks.

The extracted peak frequencies are converted in a MIDI-pitch based format with Eq. (3.18).

The result of this processing step is thus a series of peak observations $O_A(m)$ of length $\mathcal{M}$. Each observation contains a list of tonal peaks extracted from one STFT. The number of peaks may vary between observations.

3.2.4 Similarity Measure

The calculation of the similarity $\mathfrak{b}(m,n)$ between each audio observation $O_A(m)$ and each score event $\mathfrak{S}_S(n)$ is an important processing step of the tempo extraction stage with significant influence on the resulting alignment path. Since the similarity is computed between each observation and each event, the results can be plot as a matrix $\mathfrak{B}$ with dimension $\mathcal{M} \times \mathcal{N}$ ($\mathcal{M}$: number of observations, $\mathcal{N}$: number of score events).

As Orio and Schwarz point out [OS01], it is a challenge to find a similarity measure that matches feature vectors extracted from audio to feature vectors extracted or generated from score. To make these formats comparable, basically two different approaches can be chosen, either to transform the MIDI information into a format similar to the extracted audio information, or to convert the data extracted from the audio file into a more MIDI-like representation.

Here, two similarity measures have been implemented that attempt to represent both approaches, the *Spectral Similarity* method that transforms the MIDI data, and the *Note Similarity* method that transforms the extracted audio data. Both of them are based on the previously extracted spectral peaks per block. The resulting similarity matrix $\mathfrak{B}$ is calculated by the weighted sum of the spectral similarity matrix $\mathfrak{B}_{SS}$ and the note similarity matrix $\mathfrak{B}_{NS}$:

$$\mathfrak{B} = \lambda \cdot \mathfrak{B}_{SS} + (1-\lambda) \cdot \mathfrak{B}_{NS} \tag{3.22}$$

$$\mathfrak{b}(m,n) = \lambda \cdot \mathfrak{b}_{SS}(m,n) + (1-\lambda) \cdot \mathfrak{b}_{NS}(m,n) \tag{3.23}$$

with λ representing the relative weight (range $0 \ldots 1$) of the spectral similarity in the overall similarity calculation that can be adjusted. This similarity can also be interpreted as an estimation of the probability of observation $O_A(m)$ when the current score event is $\mathfrak{S}_S(n)$:

$$\mathfrak{b}(m,n) = P(O_A(m)|\mathfrak{S}_S(n)) \tag{3.24}$$

Figure 3.6 shows an example of an overall similarity matrix. A path with high similarity leads from the upper left corner to the lower right corner. A few

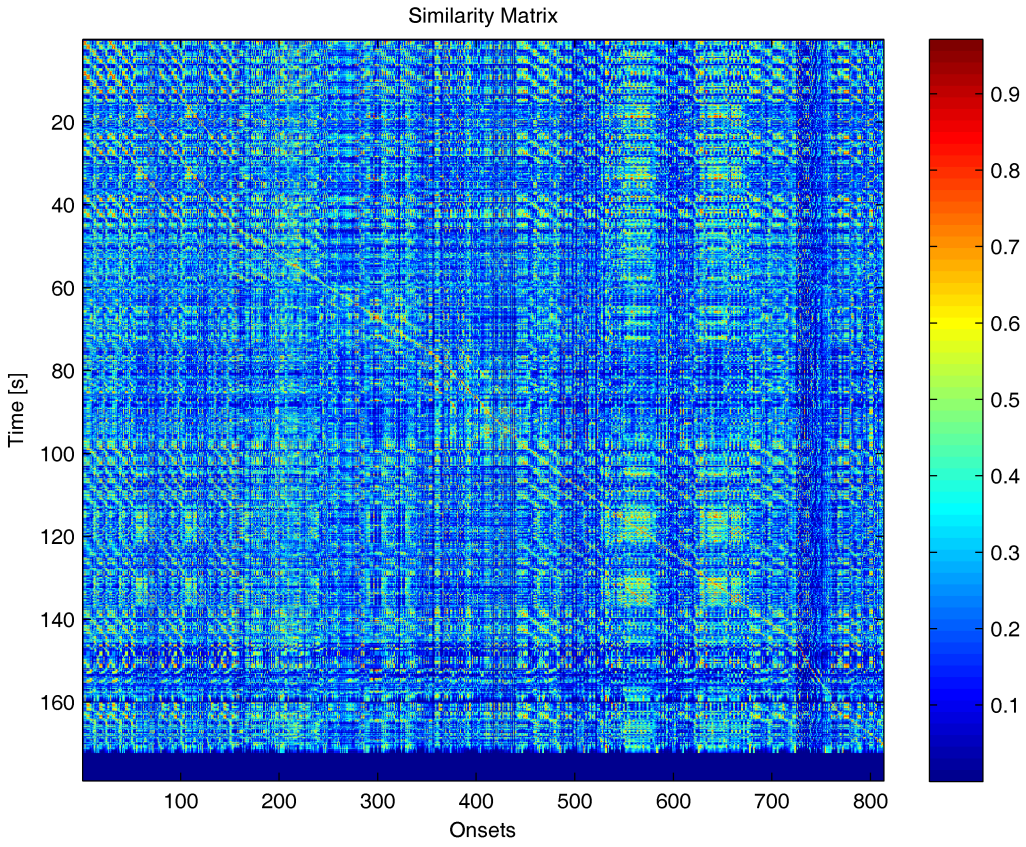

FIGURE 3.6: Example of a similarity matrix. Blue pixels indicate low similarity, red pixels high similarity. The alignment path to be detected runs from the upper left corner to the lower right corner

seconds silence can be easily identified at the end of the performance (lower edge). The matrix also shows some structural information of the piece of music: for example, diagonal lines running parallel indicate repetitions.

3.2.4.1 Spectral Similarity

This method works directly on the peaks extracted from the audio signal that are stored in the observation series $O_A(m)$ and uses them without additional processing. The disadvantage is that a model has to be applied to $\mathfrak{S}_S(n)$ to transform the "symbolic" score event data into a representation similar to the peaks extracted from the audio file. Since $\mathfrak{S}_S(n)$ only contains the fundamental frequency information (more accurately: the pitchof the fundamental frequency) and no information on number and level of the harmonics, a simple model is used to "generate" these harmonics: eight harmonics are synthesized with their

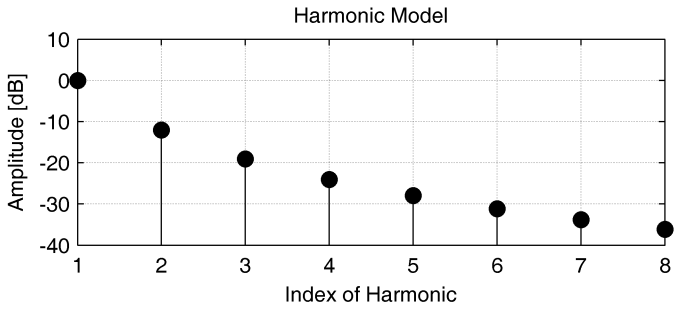

FIGURE 3.7: Amplitude of synthesized Harmonics from MIDI input

amplitude A_i decreasing:

$$A_i = \frac{A_0}{i^2} \quad \text{with } i = 1 \ldots 7 \tag{3.25}$$

Figure 3.7 plots the resulting amplitude of the generated harmonics.

Such simple models usually do not fit the reality well in the general case since they are built only with rough assumptions on timbre or instrumentation of the modeled data. Naturally, the harmonic template can be exchanged with different templates to provide a better match with the real harmonic distribution of the audio input. There have also been attempts of automatic adaption of the harmonic structure to the analyzed audio material; for example, Goto defines, in a different scope, several so called "tone models" that reflect the harmonic structure and try to match a weighted combination of these different models to the real harmonic structure [Got04]. Cont used previously learned pitch templates as basis for learning an appropriate decomposition in a real-time score following system [Con06].

But in the case of symphonic music or, more general, music played by several instrumentalists, we face a combination of different instruments played at different levels and with mixtures that can change rapidly, so it seems unlikely that there is a general model we could adapt toward, even if the model is time-dependent. Whatever model is chosen for the harmonic structure, the majority of observed harmonics probably will not be accurately modeled, and it seems unlikely that a more sophisticated model would prove to be superior in this case.

$\mathfrak{S}_S(n)$ is transformed with the presented harmonic model into a series of observations $O_S(n)$ that includes — just as $O_A(m)$ — a list of peaks per observation. This similar representation allows direct comparison of $O_S(n)$ and $O_A(m)$ for the similarity measure.

This approach is by some means related to Orio and Schwarz' *Peak Structure Distance* (PSD), for which they apply rectangular band pass filters to the spectrum at the locations of all harmonics of the expected MIDI pitch to evaluate if the harmonic content of the audio block matches this harmonic model well [OS01].

Peak Vectors

In order to be able to compute a similarity measure between the list of peaks in each audio observation $O_A(m)$ and each score observation $O_S(m)$, they are both converted into vectors with linear pitch spacing. Both resulting vectors, $\underline{\vartheta}_A(m)$ and $\underline{\vartheta}_S(n)$, now contain all peak amplitudes arranged on a linear pitch scale with a pitch resolution of 2 cent. The single peaks have been smoothed out by convolution with a Gaussian window with a standard deviation of 50 cent.

Similarity Measure

The spectral similarity $\mathfrak{b}_{SS}(m,n)$ of the two vectors $\underline{\vartheta}_A(m)$ and $\underline{\vartheta}_S(n)$ is calculated by a normalized correlation

$$\mathfrak{b}_{SS}(m,n) = \frac{\underline{\vartheta}_A(m)^T \cdot \underline{\vartheta}_S(n)}{\|\underline{\vartheta}_A(m)\| \cdot \|\underline{\vartheta}_S(n)\|} \tag{3.26}$$

The dimension of $\mathfrak{B}_{SS}$ is $\mathcal{M} \times \mathcal{N}$.

3.2.4.2 Note Similarity

For the calculation of the Note Similarity, the extracted audio peaks are transformed into a score-like representation. With this approach, there is on the one hand the danger to discard information from the audio that might be useful and on the other hand to use information that might be wrong since decisions have to be made during the conversion process which information will be used and which will not. As this approach is related to the transcription from audio to MIDI, a problem that remains unsolved in the general case, conversion errors can be expected.

Calculate Pitch Hypotheses

Pitch hypotheses, or candidates for possible fundamental frequencies, are selected per audio block m from the previously extracted peaks $O_A(m)$. The list of a priori pitch hypotheses can be large, since only the following requirements apply for a peak to be added to the list hypotheses:

1. the fundamental frequency $f_f(\mathfrak{p})$ has to lie within pre-defined pitch boundaries
2. the fundamental frequency $f_f(\mathfrak{p})$ has to be found in the list of peaks
3. there has to be at least one other peak at frequency $f_p(\mathfrak{p})$ in the list with

$$|f_p(\mathfrak{p}) - i \cdot f_f(\mathfrak{p})| < \epsilon \quad \text{, with } i = 1 \ldots 10,\ \epsilon = \text{max. deviation} \tag{3.27}$$

meaning that each pitch hypothesis p has to have at least two harmonics, the fundamental frequency and a multiple of the fundamental. The maximum deviation is limited to 12.5 cent (the range of an 1/8 tone) in the current implementation.

Pitch Detection Rules

The list of pitch hypotheses is then processed to discard "improbable" candidates. This is done with a set of rules that each increase or decrease the likelihood of a single hypothesis. For each rule, the weighting is done with a sigmoid function:

$$\psi_r(w) = \frac{1}{1 + e^{-s_r \cdot (w - \kappa_r)}} \tag{3.28}$$

with s as steepness and κ as neutral threshold being set rule dependent. The overall likelihood per hypothesis is then computed by the geometric mean over all rules

$$\psi = \sqrt[r]{\prod_{\forall r} \psi_r} \tag{3.29}$$

The following rules are applied to each hypothesis per block:

1. **relative number of harmonics**: the relation of the number of detected harmonics h_d and the number of harmonics that theoretically could have been found in the search range h_e

$$w = \frac{h_d}{h_e} \tag{3.30}$$

2. **absolute number of harmonics**: the number of harmonics that have been detected

$$w = h_d \tag{3.31}$$

3. **absolute volume**: salience of the mean RMS of all harmonic amplitudes A of the current hypothesis

$$w = \sqrt{\frac{1}{h_d} \sum_{\forall i} A_i^2} \tag{3.32}$$

4. **relative volume**: relationship of mean RMS of all harmonic amplitudes of the current hypothesis to the hypothesis with the highest RMS in this block

$$w = \frac{\sqrt{\frac{1}{h_d} \sum_{\forall i} A_i^2}}{RMS_{max}} \tag{3.33}$$

5. **energy distribution**: relative distance from the centroid of the harmonic energy to the fundamental frequency

$$w = \frac{\sum_{\forall i} f_i \cdot A_i}{f_f \cdot \sum_{\forall i} A_i} \tag{3.34}$$

6. **frequency deviation**: distance from the fundamental frequency to the equal-tempered mid frequency.

$$w = |\mathfrak{p}(f_f) - round\,(\mathfrak{p}(f_f))| \tag{3.35}$$

Then, two more rules are applied to take into account inter-block dependencies

7. **remove spurious entries**: remove all results that are isolated and too short to be considered as a note
8. **remove discontinuities**: remove pauses that are too short to be considered as a pause and are between to segments with similar pitch contents

Onset-based Segmentation

The accepted pitch hypotheses are still in a per-block representation. In this segmentation step, their pitch $\mathfrak{p}(f_f)$ and velocity $\mathfrak{v}$ are averaged over all blocks between two succeeding onset times. The series of onset marks $\hat{t}_o(l)$ that has been extracted in the pre-processing step (see Sect. 3.2.2.2) is used to segment the block data, but occasionally additional onsets are added if the pitch content between two onset marks changes significantly. These additional onsets are added if a new pitch occurs at a time that significantly deviates from the nearest (already detected) onset time. This procedure, combined with the onset detection form the pre-processing, has the tendency to insert an additional onset mark instead of discarding it when in doubt. This behavior is desired in the current context.

The result is a series of audio events $\mathfrak{S}_A(l)$ in the internal score format. In the optimal case, that is without any detection errors, this series of audio events $\mathfrak{S}_A(l)$ would equal the series of score events $\mathfrak{S}_S(n)$.

Pitch Chromagram

The Pitch Chroma(-gram) (also called Pitch Class Profile) is an octave independent representation of the pitch content in a predefined time interval. It is a vector with twelve entries spaced in equal-tempered semi-tones, each entry referring to a pitch class such as C, $C^{\#}$, $D, \ldots$ In general, its advantage is the robustness against octave errors, with the self-evident disadvantage that the octave information is lost, so a note repetition is not distinguishable from jumping an octave.

While such a pitch chroma representation has been frequently used in the past, see e.g. Krumhansl's tonal distributions [Kru90], Bartsch and Wakefield were probably the first to propose its use in the context of audio signal processing [BW01]. Nowadays, this representation can be frequently found in audio signal processing publications e.g. on automatic key finding [Izm05] or as a feature in a more general context of music information retrieval [TEC02]. The exact

computation of the pitch chroma varies from publication to publication; in all cases the spectrum is grouped into semi-tone bands, a measure of salience is computed in each band, and the sum of all semi-tones per pitch class is calculated. In the simplest case, the amplitude or energy per un-windowed semi-tone band is integrated and added to the corresponding pitch class entry. Here, we exclusively take into account the extracted fundamental frequencies with their velocities to build up the pitch chroma, and construct a comparable pitch chroma from the score events.

The pitch chroma $\underline{\nu}$ contains for each of the 12 pitch classes the sum of the velocity of all occurrences of this pitch class of the octaves. It is computed for each audio block m and for each score event n with all notes or their pitches $\mathfrak{p}$ and velocities $\mathfrak{v}$:

$$\begin{aligned} \nu_0 &= \sum_{\forall \mathfrak{p} \text{ with } mod(\mathfrak{p},12)=0} \mathfrak{v} \\ \nu_1 &= \sum_{\forall \mathfrak{p} \text{ with } mod(\mathfrak{p},12)=1} \mathfrak{v} \\ &\vdots \\ \nu_{11} &= \sum_{\forall \mathfrak{p} \text{ with } mod(\mathfrak{p},11)=11} \mathfrak{v} \end{aligned} \tag{3.36}$$

The pitch chromagrams for the audio events are replicated in a manner that they are spaced equidistantly with the audio hop size $\mathcal{H}$ in order to match the number of audio pitch chromagrams with the number of observations $O_A(m)$: $\mathcal{M}$; the number of score pitch chromagrams remains $\mathcal{N}$, as the number of score events $\mathfrak{S}_S(n)$.

Similarity Measure

The note similarity $\mathfrak{b}_{NS}(m,n)$ is then calculated in a similar way as the Spectral Similarity $\mathfrak{b}_{SS}(m,n)$

$$\mathfrak{b}_{NS}(m,n) = \frac{\underline{\nu}_A(m)^T \cdot \underline{\nu}_S(n)}{\|\underline{\nu}_A(m)\| \cdot \|\underline{\nu}_S(n)\|} \tag{3.37}$$

The dimension of $\mathfrak{B}_{NS}$ is $\mathcal{M} \times \mathcal{N}$.

3.2.5 Tempo Curve Extraction

3.2.5.1 Alignment Path Finding

The problem of the optimal alignment of audio and MIDI signal can be stated as finding and following the path with the highest similarity score through our list of score events $\mathfrak{S}_S(n)$ given the series of observations $O_A(m)$, which means through the similarity matrix $\mathfrak{B}$. The resulting path $P(q)$ is a series of matrix indices (m, n) with length Q. In each row, it contains the matrix indices (m, n) that define the path.

The path $P(q)$ has the following properties for the current system:

1. **path start**: $P(0) = (0, 0)$, i.e. $m = 0 \ \wedge \ n = 0$
2. **path end**: $P(Q-1) = (\mathcal{M}-1, \mathcal{N}-1)$, i.e. $m = \mathcal{M}-1 \ \wedge \ n = \mathcal{N}-1$
3. **path length**: $Q = \mathcal{M}$
4. **causality**:

$$\begin{aligned} m|_{P(q+1)} &> m|_{P(q)} \\ n|_{P(q+1)} &\geq n|_{P(q)} \end{aligned}$$

5. **monotonicity**:

$$\begin{aligned} m|_{P(q+1)} &= (m+1)|_{P(q)} \\ n|_{P(q+1)} &\leq (n+1)|_{P(q)} \end{aligned}$$

The latter two properties mean that it is not allowed to go back in time and that every observation has to be assigned to one score event; while several observations (row indices) can be assigned to one score event (column indices), it is not allowed to assign one observation to several score events, meaning that the row index is incremented for the path in any case. Therefore, there are only two possible path transitions from $P(q)$ to index $P(q+1)$: stay at the same score event, or step to the next score event. This restriction is allowed since — as the hop size $\mathcal{H}$ is much smaller than the minimum note length — it is obvious that $\mathcal{M} \gg \mathcal{N}$.

A dynamic time warping algorithm is applied to compute the path with the maximum overall score[1] through the matrix $\mathfrak{B}$. This algorithm is similar to the Viterbi Algorithm [Vit67], but works additive and has a special term, the onset weight extracted in the pre-processing. A score matrix $\mathfrak{D}$ is defined containing the accumulated score with the dimension $\mathcal{M} \times \mathcal{N}$. The stages of the algorithm are:

[1]Note that in this context, the term *score* does not refer to the musical score but to a measure of fit. The inverse score would be the so-called alignment *cost*.

1. **Initialization**

$$\begin{aligned} \mathfrak{D}(0,0) &= \mathfrak{b}(0,0) \\ \mathfrak{D}(m,-1) &= -\infty \quad 0 \leq m \leq \mathcal{M}-1 \\ \mathfrak{D}(-1,n) &= -\infty \quad 0 \leq n \leq \mathcal{N}-1 \\ P(0) &= (0,0) \end{aligned} \tag{3.38}$$

2. **Recursion**

$$\begin{aligned} \mathfrak{D}(m,n) &= \max\Big(\mathfrak{D}(m-1,n)+\zeta(m), \mathfrak{D}(m-1,n-1)-\zeta(m)\Big) + \\ &\quad +\mathfrak{b}(m,n) \end{aligned} \tag{3.39}$$

$$\Psi(m,n) = \begin{cases} (-1,0) & \text{if } \mathfrak{D}(m-1,n) > \mathfrak{D}(m-1,n-1) \\ (-1,-1) & \text{if } \mathfrak{D}(m-1,n) \leq \mathfrak{D}(m-1,n-1) \end{cases} \tag{3.40}$$

3. **Termination**

$$m = \mathcal{M}-1 \tag{3.41}$$

4. **Path Backtracking**

$$P(m) = P(m+1) + \Psi(P(m+1)),\ m = \mathcal{M}-2, \mathcal{M}-3, \ldots, 0 \tag{3.42}$$

The onset probability term $\zeta(m)$ is added to encourage a state transition if the onset probability is high and to discourage it otherwise. Ψ is a provisional result that contains the backtracking directions for each matrix element.

3.2.5.2 Alignment Path Post-Process

The length of the computed alignment path P is now $\mathcal{M}$, but since one can only align "real" musical events, and since only the alignment of the onset times between audio and score is of interest, the path dimension can be reduced to length $\mathcal{N}$. This is done by discarding all path entries that do not contain a state transition, resulting in exact one alignment time for each score event. The series of resulting alignment times will be referred to as $t_A(n)$.

The presented path finding algorithm finds the best global path solution. This is not necessarily the best local path at all possible locations. Furthermore, detection inaccuracies can lead to deviations from the detected path. To minimize these effects, the extracted alignment times have to be post-processed.

At this stage, two problems have to be faced:

1. How to identify regions that have to be corrected?

2. How to correct the alignment times?

For the identification of regions to correct, a normalized tempo curve is computed with

$$Tempo^*_{ld} = \log_2 \left(\frac{\mu_{t_S,BPM}}{\mu_{t_A,BPM}} \cdot \frac{t_S(n+1) - t_S(n)}{t_A(n+1) - t_A(n)} \right) \tag{3.43}$$

to obtain a logarithmic tempo representation with the 0 dB-point referring to a reference point that normalizes the extracted mean tempo $\mu_{t_A,BPM}$ to the reference mean tempo $\mu_{t_S,BPM}$. This representation is symmetric in terms of positive and negative tempo changes with each increase to the double tempo resulting in an increase of 1.

Path indices are assumed to be incorrect if the absolute difference of the logarithmic tempo difference between the current value and its preceding value is larger than 1, as sudden tempo changes of factor two and higher are considered unlikely to appear.

The onset times are adjusted within a time window (currently 0.75 s) around the erroneous tempo entry. The maximum amount of correction per onset time decreases with its distance from the error location.

Additionally, there is the option to smooth out the tempo curve with a low pass filter afterward. This filter cannot be applied to the tempo curve directly, as smoothing the discrete derivative of the alignment times would most definitely lead to a time offset with increasing time, so the filter has to be directly applied to the alignment times. For the same reason, an infinite impulse response filter cannot be used. The filtering process is further complicated by the fact that the signal is not sampled at equidistant times; the samples are spaced as the onsets occur in the score file. For these reasons, a moving average filter was selected that is windowed by a Hanning window, taking into account only alignment times within the window (finite impulse response) and allowing to compute the window weight depending on the specific sample position within the window.

3.2.6 Evaluation

3.2.6.1 Goals and Constraints

In the evaluation of audio-to-score alignment systems, measuring the accuracy and the robustness of the alignment is of interest. The accuracy as well as the robustness are criteria for the fitting between a reference alignment path and the extracted path. The average or overall distance is a measure of accuracy and the maximum local path distance a measure of robustness.

The granularity of the evaluation measure varies in the literature from comparison on onset level over beat level up to the bar level and is dependent on the

available reference data set and the intended application. For the visualization of the current score position, an accuracy on the bar level might be sufficient, but here, measurement at the onset level is required. Lower granularities can be thought of (such as ADSR[2] envelope granularity) but are technically not feasible for polyphonic audio-to-score alignment systems within the foreseeable future.

In the optimal case, a system would have similar or (slightly) better characteristics than that of human listeners. Higher resolution would hardly be necessary for MPA. The human onset detection accuracy has already been detailed in Sect. 3.2.2.2. It can probably be assumed that human listeners are not able to notice deviations of less than 50 ms in the case of several performers and instruments without sharp attacks such as string or woodwind instruments. The deviations can be assumed to be higher for highly polyphonic music and large ensemble sizes.

Ornamentations like trills, grace notes and *arpeggii* are either not present in the input MIDI file, or performed in one of many possible ways. As the MIDI format offers no option to include score-like ornamentation instructions, alignment of these passages is likely to fail. In the best case, these parts should not be included in the evaluation, but this would require a preceding selection of parts that are musically important for the alignment task and parts that are not important. To avoid any arbitrary decisions, all parts of the test set have been labeled as valid for evaluation, accepting possible inaccuracies.

Assembling a test set for the evaluation of audio-to-score alignment systems is a difficult task; the ground truth has to be generated either manually by annotating all onset times, a time-consuming task for longer pieces of audio, or generated from MIDI data, potentially provoking criticism on how representative the test set is for "real" input data. Only recently, there have been first attempts to construct a database with performance data that could be used as a ground truth for evaluation purposes [HMK08].

3.2.6.2 Criteria

While the evaluation task is basically similar to the evaluation of onset tracking systems (compare Sect. 3.2.2.2), one major difference is that the evaluated system detects every onset, and does not detect additional onsets; in other words, there are no false negatives or false positives. This simplifies the evaluation as only the deviation between matching pairs of onset times has to be taken into account. The pairwise difference or deviation between two sequences of length $\mathcal{N}$, the reference onset times $t_R(n)$ and the extracted onset times $t_A(n)$, is examined.

[2] ADSR envelope: used in synthesizers to describe the volume envelope of a sound in four phases: attack, decay, sustain and release

Mean

The mean deviation calculates the arithmetic average over all differences. It is zero if the density distribution of the differences is symmetric around zero. Otherwise, it shows that the algorithm has a tendency to detect onsets earlier or later than they really are.

$$\delta_{mean} = \frac{1}{\mathcal{N}} \sum_{\forall n} t_R(n) - t_A(n) \tag{3.44}$$

Median

The median of the differences is the value that separates the lower half of the density distribution of differences from the higher half. The relation of mean and median can help identify asymmetric density distributions. Furthermore, the median provides a more reliable result if outliers obfuscate the mean calculation.

Manhattan Distance

The Manhattan or normalized 1-Norm Distance calculates the overall magnitude difference of the extracted values from the reference values. This is a fitting measure of overall accuracy.

$$\delta_{L_1} = \frac{1}{\mathcal{N}} \sum_{\forall n} |t_R(n) - t_A(n)| \tag{3.45}$$

Standard Deviation

The Standard Deviation (or the normalized 2-Norm) illustrates how the density distribution is centered around its mean. The result is close to zero if all differences are similar to the mean value and takes high values if the differences are more scattered.

$$\delta_{\sigma} = \sqrt{\frac{1}{\mathcal{N}} \sum_{\forall n} \left(t_R(n) - t_A(n) - \delta_{mean}\right)^2} \tag{3.46}$$

Maximum Difference

The Maximum Difference is an indication of the algorithm's robustness as it measures how far the result differs from the reference in the worst case. It is also known as the Chebyshev Distance.

$$\delta_{max} = \max_{\forall N} \left(|t_R(n) - t_A(n)|\right) \tag{3.47}$$

Mean Number of Matches

The mean number of matches is an intuitive measure of the algorithm's accuracy. It is the number of differences with a magnitude smaller than a reference interval τ in relation to the overall number of onsets.

$$\delta_\tau = \frac{1}{\mathcal{N}} \sum_{\forall n} x(n) \quad \text{with } x(n) = \begin{cases} 0 & \text{if } |t_R(n) - t_A(n)| \leq \tau \\ 1 & \text{otherwise} \end{cases} \tag{3.48}$$

3.2.6.3 Test Set

As pointed out above, the assembly of a representative test set is difficult. Therefore, the used test set consists only of audio files that were generated from MIDI files. However, some effort has been put into making the test set as representative as possible. All audio sequences except the Mozart excerpts (see below) were generated with a well-known sample library that comes with the software sampler *Kontakt 2* from the company Native Instruments. The synthesis of orchestra instruments by sample library players is usually supposed to produce more realistic sounds than "pure" synthesis. Furthermore, reverberation was added to increase naturalness. The test set consists of five sequences with the basic properties summarized in Table 3.2. All sequences are polyphonic and contain orchestral instrumentation.

ID	*Composer*	*Piece*	*Length (s/Ons/Beats)*	*Contrib. %*
brahms	Brahms	Symphony No.3 F Major I.Movement (1 – 109)	278.3s/3592/678	49.63
bach	Bach	Violin Concerto a minor I.Movement	268.5s/1295/341	26.09
haydn	Haydn	Symphony No.90 C Major III.Movement (1 – 116)	149.1s/600/345	12.09
moz_1	Mozart	Piano Concerto No.20 d minor I.Movement (33 – 72)	72.0s/319/156	6.43
moz_2	Mozart	Piano Concerto No.20 d minor I.Movement (448 – 480)	66.7s/286/124	5.76

TABLE 3.2: Properties of the test set for the evaluation of the automatic tempo extraction and their relative contribution to the overall test set

The *bach* sequence is the first movement of the Violin Concerto BWV1041. A tempo curve has been applied to the sequence to model increasing and decreasing tempo as well as sudden tempo changes. In this case, the tempo curve is not musically motivated but was rather applied in order to ensure that the test set contains such tempo variations. The *haydn* sequence is the beginning of the third movement of the 90^{th} symphony. It does not, as the only sequence in the test set, contain any tempo changes. Theoretically, the tempo extraction algorithm should not perform better for sequences with constant tempo as opposed to varying tempo, and this sequence has been included in the

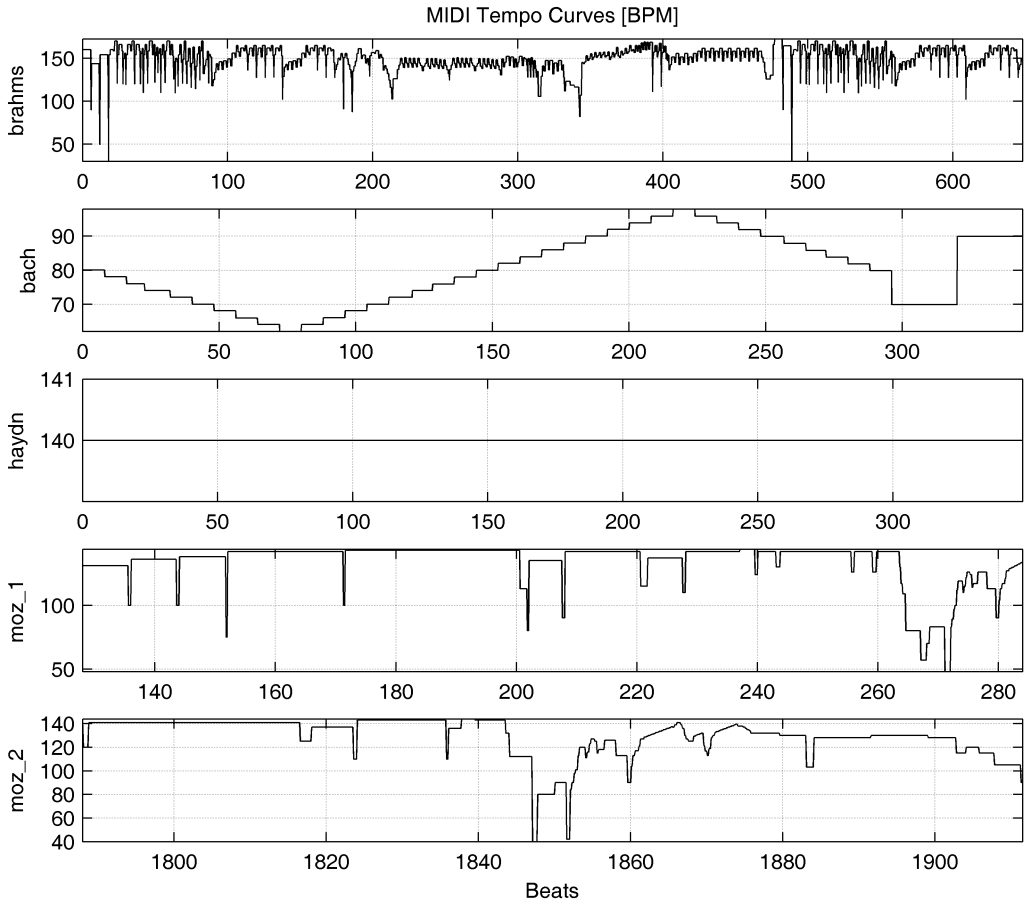

FIGURE 3.8: Tempo curves in BPM for every file of the test set

test set to verify this assumption. The two Mozart excerpts (*moz_1*, *moz_2*) from the first movement of the piano concerto KV466 should be the most representative samples in the test set. The audio signal was synthesized in the context of a diploma thesis that evaluated how realistic a classical orchestral score can be rendered by means of software instruments [Gsc04]. Unfortunately for our evaluation, the resulting audio file was not solely generated from MIDI but was mixed together with a separately recorded piano track; therefore, only excerpts without piano from this sequence can be used. Two excerpts were chosen from beginning and end of the movement. The *brahms* excerpt is the beginning of the first movement of his third symphony and features the most extensive instrumentation in the test set. The tempo curve was already part of the MIDI file.[3]

Figure 3.8 plots the tempo curves for all files in the test set.

[3]available at `www.kunstderfuge.com/brahms.htm`

3.2.6.4 Results

Table 3.3 and Fig. 3.9 summarize the evaluation results.

	brahms	***bach***	***haydn***	***moz_1***	***moz_2***	***Overall***
δ_{mean}	-0.0259	-0.0487	-0.0744	-0.0228	-0.0332	-0.0379
δ_{med}	-0.0950	-0.0227	-0.0580	-0.0047	-0.0083	-0.0533
δ_{L_1}	0.2864	0.0642	0.0993	0.0580	0.0669	0.1785
δ_{σ}	0.4902	0.1285	0.1409	0.0852	0.1324	0.3573
δ_{max}	2.4965	1.4309	0.8555	0.3667	1.4190	2.4965
δ_{25}	0.0824	0.4695	0.2150	0.4326	0.3741	0.2388
δ_{50}	0.1616	0.6927	0.3833	0.6176	0.6014	0.3816
δ_{75}	0.2728	0.7876	0.5933	0.7304	0.7448	0.5025
δ_{100}	0.3841	0.8386	0.7100	0.7868	0.8671	0.5958
δ_{125}	0.4641	0.8826	0.7900	0.8339	0.8986	0.6615
δ_{150}	0.5173	0.9104	0.8333	0.9091	0.9161	0.7062
δ_{175}	0.5566	0.9259	0.8617	0.9373	0.9336	0.7360
δ_{200}	0.5981	0.9367	0.8950	0.9498	0.9441	0.7649
δ_{250}	0.6654	0.9560	0.9233	0.9843	0.9510	0.8094
δ_{300}	0.7276	0.9683	0.9283	0.9906	0.9650	0.8453

TABLE 3.3: Evaluation results in seconds

The overall results are calculated over all onset deviations from all files. Thus, files in the test set with a higher number of onsets influence the overall result more than shorter excerpts. The contribution of each piece in percent is shown in the last column of Table 3.2; the *brahms* excerpt nearly constitutes half of the test set. In general, the accuracy tends to decrease with the complexity of the tracked score and instrumentation, a result that was to be expected. Thus, the *brahms* excerpt shows the worst accuracy which in turn influences the overall result significantly. In the following, the *brahms* results will partly be discussed separately from the results for the rest of the test set. The mean and median of the detected difference are between 5 ms and 100 ms after the MIDI onset, as is the average absolute difference between reference and extracted onset times for all pieces except *brahms*. The standard deviation is between 80 ms and 150 ms, but as high as 500 ms for the *brahms* example.

Except for very complex scores, approximately 90% of the onsets are correctly detected within a tolerance window of $150 - 200$ ms, but single outliers with differences with more than a second do occur occasionally.

3.2.6.5 Discussion

The system offset is to be expected as the perceptual onset time cannot appear before and is not likely to appear at the MIDI onset time that constituted the ground truth. This is because the used instrument samples will include the initial transient phase. Standard deviations smaller than 200 ms are in a range that can probably be considered as acceptable for ensemble or orchestra music, however, the standard deviation results for the large orchestra score indicate that the system does not provide sufficient accuracy for the analysis of such large instrumentations. Another possible, although less likely interpretation

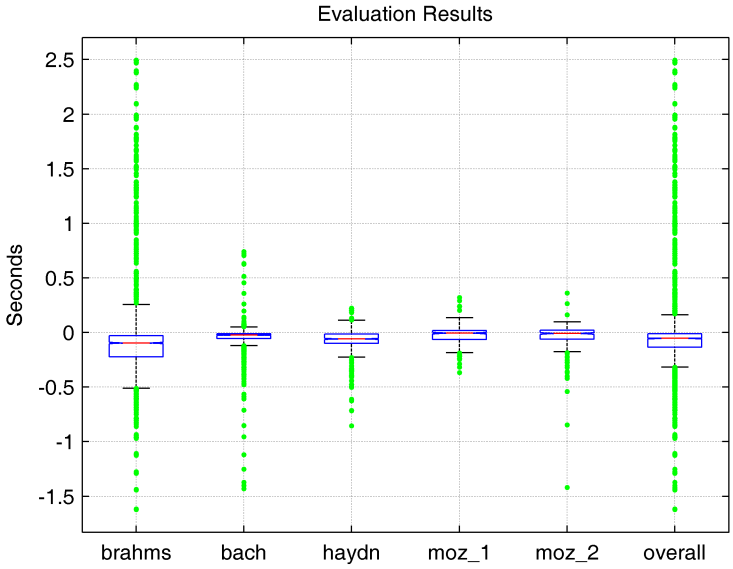

FIGURE 3.9: Box plot of the tempo extraction results presented in Table 3.3

of the inferior results for the *brahms* example would be that the deviations depend on the length of the test files, as larger deviations are more likely to occur in longer test sequences. As expected, the system does not appear to be susceptible to sudden tempo changes.

Typical positions of larger deviations are repetitions of single notes and chords as the system has difficulties to detect which event should be assigned to the corresponding score positions, ornamentations like timpani rolls that produce a large number of onsets in a short time and might mask the tonal content for short time frames, and passages with very low relative amplitude.

Due to the multi-voiced test set with orchestra instrumentation used in this evaluation we have to be cautious in comparing the presented results with other evaluation results. A good example for a related evaluation is the score-following evaluation that has been done in the context of MIREX (Music Information Retrieval Evaluation eXchange).[4] The MIREX results are not directly comparable because on the one hand, real-time score followers were evaluated that may also report false negatives and positives and have to work within considerable constraints compared to a non-real time algorithms, on the other hand only monophonic audio excerpts were aligned. They used a large test set of short excerpts, and reported standard deviations between reference and extracted data in the range between 10 ms and 4 s.

Overall, it can be concluded from the presented results that automatic tempo extraction is possible within reasonable tolerances, but it still requires human interaction for the correction of certain outliers as these will limit the usefulness of the results.

[4] `www.music-ir.org/mirex/2006/index.php/Score_Following_Result`

Chapter 4

Dynamics Feature Extraction

Performance instructions for the variation of dynamics are relatively vague in the score; usually only $5-8$ dynamic steps are used in the instructions, complemented by indications for smooth loudness transitions and dynamic accents. Additionally, properties such as instrumentation, number of voices or musical tension will influence the performance and perception of musical dynamics.

Musical dynamics are closely related to loudness, a fact that is not surprising since the performance instructions are loudness-related. But absolute loudness cannot be the only cue to understand musical dynamics; for example, while listening to a piano-passage on a hi-fi system, the reproduction volume can be manipulated without loosing the *piano* or *fortissimo* character of the performance. Therefore, the communication of musical dynamics has to use additional properties besides the absolute loudness. This has also been shown by Nakamura, who found that listeners' abilities to comprehend musical dynamics as intended by the performer outperform predictions of these dynamics based on measurements of acoustic intensity [Nak87]. The additional cues that enable listeners to do so are most likely related to timbre and (musical) context. However, the intensity or loudness is probably the attribute with the strongest correlation to dynamics and remains the most representative way of measuring musical dynamics objectively.

In the analysis of musical dynamics, the performance researcher has to deal with a number of possible representations:

- musical or score-like indications of dynamics: e.g. ***pp*** (*pianissimo*), ***p*** (*piano*), ***mf*** (*mezzoforte*), ***f*** (*forte*), ***ff*** (*fortissimo*), *crescendo* or *decrescendo*, ***sf*** (*sforzando*)
- mechanical measurements on instruments such as the hammer velocity of a piano key: unit $^{\mathrm{m}}/_{\mathrm{s}}$

- representations of mechanical measurements such as MIDI velocity: unitless scale $0, 1, 2 \ldots 126, 127$
- acoustical measurements such as sound pressure level: unit usually Decibels, ranging from $-\infty \ldots 0\,\mathrm{dB_{FS}}$
- algorithmic approximations to perceived loudness: measured either in weighted decibels, for example $\mathrm{dB_A}$, or other units such as sone
- subjective measurements of perceived loudness: on any appropriate scale
- subjective measurements of perceived music dynamics indications: see score-based indications above

All of these different representations can be found in the performance analysis literature. The problem behind this multitude of different representations is that, although they are similar in a way that their values can be ordered on a monotonically increasing scale from low to high or quiet to loud, a mapping between those scales is either not fixed or can be found only approximately. The relationship between hammer velocity and MIDI velocity is not standardized and instrument dependent. For the Yamaha Disklavier, Goebl and Bresin were able to fit a logarithmic curve into the relationship of hammer and MIDI velocity [GB01]. They also found that for the Yamaha Disklavier and the Bösendorfer SE System, the relationship between MIDI velocity and sound pressure level is nearly linear when disregarding very low and high values [GB03]. Dannenberg investigated the RMS peak level of various synthesizers and software instruments and found great differences among different synthesizers [Dan06]. He identified a general trend for the velocity to be related to the square root of the RMS peak instead of its logarithm. Using one electronic instrument, Taguti measured the A-weighted sound pressure level [Tag03] dependent on velocity and key. The results, displayed over various keys for different input velocities, showed non-systematic deviations of up to 10 dB from a constant level among keys. Palmer and Brown found linear relationships between hammer velocity and peak amplitude for a Bösendorfer SE-System [PB91].

Modeling perceived loudness with objective measurement methods is an old and lively field of psycho-acoustic research. Proposed models range from the measurement of intensity or sound pressure with a previously applied weighting filter (such as A, B, or C-weighting to approximate the human level sensitivity in different frequency regions at different loudness ranges) to relatively complex psycho-acoustic models for loudness calculation (compare [MGB97], [ZF99]). In a recording studio environment, more devices for the monitoring of level and loudness can be found (Peak Meter, VU Meter).

4.1 Implemented Features

The decision of what might be the appropriate algorithm to apply to the extraction of dynamics features from music performances is hard to make, facing the diversity of different approaches. The vast majority of performance researchers use simple intensity measurements, but several studies make also use of an implementation of Zwicker's loudness model (e.g. [DGW02]). Nakamura's study has shown that intensity is of relevance in the analysis of musical dynamics [Nak87] and Geringer found a high correlation between intensity values and the corresponding (continuously recorded) loudness judgments of listeners to classical music [Ger95]. Benjamin found comparably small differences between the Zwicker loudness and an A-weighted intensity measure for broadcast streams, suggesting that the latter might be used as a reasonable approximation [Ben02]. Although these are indications that (frequency weighted) intensity measurements might be sufficient or appropriate to analyze musical dynamics, a concluding selection of the most suitable loudness measure is difficult. Therefore, a group of various algorithms representing different approaches to the problem has been selected for implementation: measures used during the recording and mastering sessions, (weighted) intensity or RMS measures, and two variations of a comparably complex psycho-acoustic model for loudness measurement.

All implemented loudness measurements that will be referred to as loudness features in the following and are converted to Decibel (dB_{FS}) after their computation. Since the signal amplitude is normalized to ± 1, this conversion can simply be done by applying the following operation to the extracted feature value v_{in}.

$$v = 20 \cdot \log_{10}(v_{in} + \epsilon) \tag{4.1}$$

with the small constant ϵ to avoid the calculation of $\log(0)$.

4.1.1 Peak Meter

A Peak Meter (also PPM for Peak Programme Meter) is one of the standard ways to control the level of audio signals in recording studio environments. It is not really a measure of loudness, but rather an envelope measure that reacts fast on rising amplitudes (short attack time AT) and slow on decreasing amplitudes (long release time RT). This feature is somewhat similar to the peak amplitude measure for which Palmer found a quasi-linear relationship with hammer velocity on a Bösendorfer piano [PB91].

The implementation is based on the peak meter described in Zölzer [Zöl97] and is depicted in the block diagram shown in Fig. 4.1. The output is computed sample per sample.

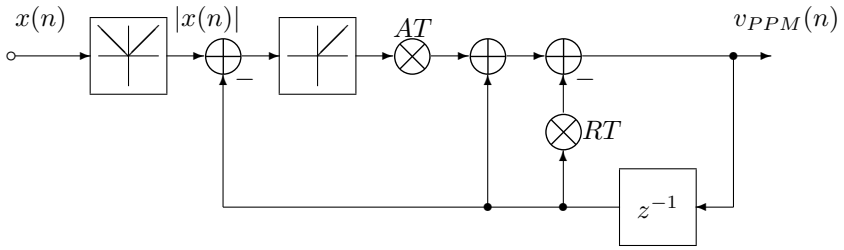

FIGURE 4.1: Block diagram of the implemented peak meter

The time constants have been adjusted to standard recording studio equipment times (compare [DIN94]):

- attack time (compare coefficient AT): 10 ms
- release time (compare coefficient RT): 1500 ms

4.1.2 VU Meter

The VU Meter or volume indicator is a frequently used equipment in recording studios to display the signal level in *Volume Units*. In contrast to the Peak Meter, it averages out short level variations in order to approximate perceived loudness. The VU is calculated by smoothing the absolute value of the input sample with a second order IIR low pass filter.

4.1.3 Root Mean Square Based Features

4.1.3.1 RMS

The RMS (Root Mean Square) is the intensity of a signal. It is the equivalent of the sound pressure level in the digital domain and is the most frequently used measure in music performance analysis. It can be calculated for a time frame of length $\mathcal{K}$ samples by

$$v_{RMS}(n) = \sqrt{\frac{1}{\mathcal{K}} \sum_{k=n-\mathcal{K}/2}^{n+\mathcal{K}/2-1} x(k)^2} \tag{4.2}$$

or can be approximated with a single-pole low pass filter

$$v'_{RMS}(n) = (1-\alpha) \cdot v'_{RMS}(n-1) + \alpha \cdot x(n)^2 \tag{4.3}$$

with α being computed from the integration time $T = {}^{\mathcal{K}}/_{f_s}$ with

$$\alpha = e^{\frac{-2.2}{f_s \cdot T}}. \tag{4.4}$$

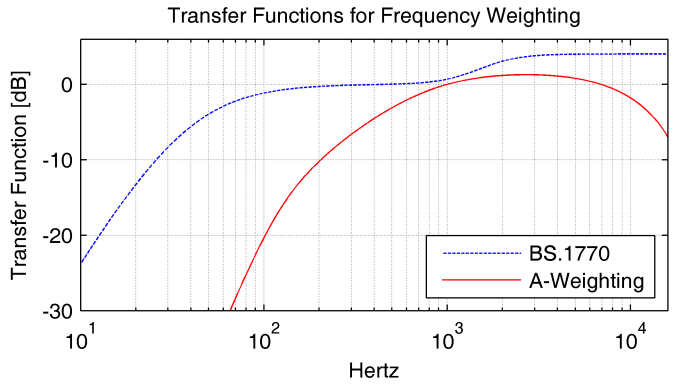

FIGURE 4.2: Frequency Weighting Transfer Functions: A-Weighting and frequency weighting for ITU-R BS.1770

4.1.3.2 A-weighted RMS

The A-weighted RMS measure is a frequently used loudness measurement. The A-weighting filter is applied before the RMS calculation and models the frequency-dependent sensitivity of the human ear at low levels. Its transfer function is depicted in Fig. 4.2.

4.1.3.3 ITU-R BS.1770

This loudness measurement has been recently standardized in ITU recommendation BS.1770 [ITU06], and should offer high correlation with subjective loudness ratings [SN03]. It is also an RMS measurement, but two filters are applied in a pre-processing stage: a high pass and a high shelving filter (the latter only being used for multichannel processing). The combined pre-filter curve is displayed in Fig. 4.2.

4.1.4 Zwicker Loudness Features

Based on the results of psycho-acoustic experiments, Zwicker proposed a functional model for the computation of loudness (compare [ZF99]).

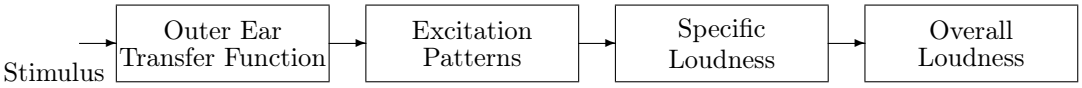

FIGURE 4.3: Flow Chart of Zwicker's model for loudness computation

In contrast to other models, masking phenomena are taken into account in order to calculate the excitation patterns. Two approaches have been selected from the literature to represent Zwicker's loudness model here. The loudness is extracted in overlapping blocks at equidistant time steps.

4.1.4.1 DIN 45631

This norm standardizes the calculation of loudness as proposed by Zwicker for an input signal that is already available in third band levels [DIN91]. These third band levels are partly combined to approximate a Bark scale that models the frequency grouping in the human ear. The outer ear transfer function, defined by a table, is applied to these bands that are afterward subjected to a simple masking model. This model only takes into account the masking of higher frequencies. The resulting specific loudness per band is finally summed up in order to compute the overall loudness.

The used implementation of this norm does not in any way normalize the audio data as this is neither required in the current context nor possible without a predefined listening level. Therefore, the results are incorrect when attempted to be used as a standardized *sone* measure, but remain correct in their interrelations.

4.1.4.2 ITU-R BS.1387

This ITU document is not a recommendation for loudness calculation, but for the objective measurement of quality impairment of coded audio signals [ITU01]. However, it includes an STFT-based loudness measure based on Zwicker's model which has been implemented here with minor modifications.

The magnitude spectrum, calculated from the STFT, is weighted by an outer ear transfer function $W(k)$ derived from a model of the absolute threshold of hearing introduced by Terhardt [Ter79]:

$$W(k) = -0.6 \cdot 3.64 \cdot \left(\frac{f(k)}{1000\,\mathrm{Hz}}\right)^{-0.8} + 6.5 \cdot e^{-0.6\left(\frac{f(k)}{1000\,\mathrm{Hz}}-3.3\right)^2} - 10^{-3} \cdot \left(\frac{f(k)}{1000\,\mathrm{Hz}}\right)^{3.6} \quad [dB] \tag{4.5}$$

Then, the frequency bins are grouped into critical bands (Bark scale) with a resulting resolution of approximately four bands per Bark. The frequency-to-bark transformation is based on a model by Schroeder [SAH79]:

$$z = 7 \cdot \operatorname{arcsinh}\left(\frac{f}{650\,\mathrm{Hz}}\right) \tag{4.6}$$

In order to compute the excitation patterns in the bark domain, a level dependent triangular spreading function is applied to the bark spectrum to model masking effects. The spreading function, as depicted in Fig. 4.4, computes masking toward both lower and high frequencies, but only the higher frequency part is level dependent.

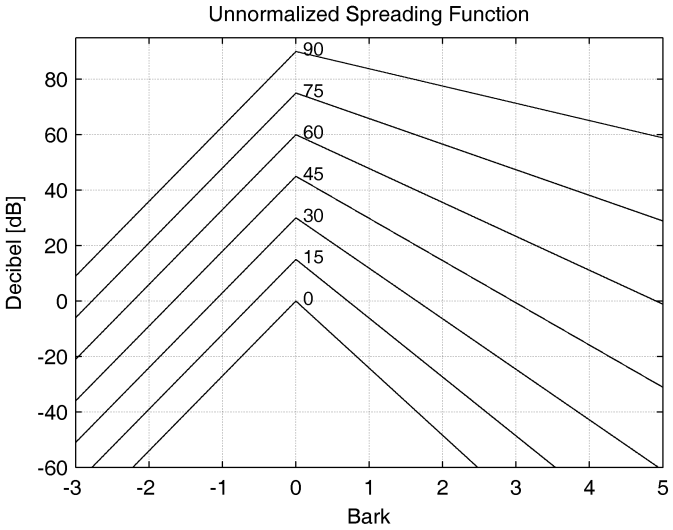

FIGURE 4.4: Unnormalized level-dependent spreading function for the computation of the masking threshold in ITU-R recommendation BS.1387

The resulting excitation patterns $E(z)$ are then used to compute the specific loudness $N'(z)$ with a formula introduced by Zwicker that additionally applies nonlinear weighting.

$$N'(z) = \left(\frac{S_N(z)}{s(z)}\right)^{0.23} \cdot \left(\left(1 - s(z) + s(z) \cdot \frac{E(z)}{S_N(z)}\right)^{0.23} - 1\right) \tag{4.7}$$

$S_N(z)$ models the excitation at the absolute threshold of hearing, and $s(z)$ the so-called masking index for the band with bark index z.

The sum over all specific loudness bands is then the resulting loudness.

Corresponding to the implementation of the DIN norm, the implementation of this recommendation does not normalize the audio data as this is neither required nor possible (no listening level) in the current context.

4.2 Example Results

Figure 4.5 displays the extracted feature values for an example file that consists of a string quartet movement.

For all extracted features for this example file, a string quartet movement composed by Beethoven, the pairwise correlation has been calculated (see Table 4.1). The correlation between features v_1, v_2 is calculated by

$$r_{1,2} = \frac{\sum_{n=0}^{N-1} (v_1(n) - \mu_{v1}) \cdot (v_2(n) - \mu_{v2})}{N \cdot \sigma_{v1} \cdot \sigma_{v2}} \tag{4.8}$$

with μ being the arithmetic mean, σ the standard deviation and N the number of values per features.

Each feature vector consists of more than 18000 observations. High correlations between several pairs of extracted features can be observed.

	$v_{Zw(DIN)}$	$v_{Zw(1387)}$	v_{RMS}	v_{1770}	v_{dBA}	v_{PPM}	v_{VU}
$v_{Zw(DIN)}$	1.000	0.784	0.935	0.964	0.938	0.879	0.921
$v_{Zw(1387)}$	0.784	1.000	0.836	0.821	0.803	0.834	0.838
v_{RMS}	0.935	0.836	1.000	0.960	0.996	0.969	0.987
v_{1770}	0.964	0.821	0.960	1.000	0.961	0.911	0.939
v_{dBA}	0.938	0.803	0.996	0.961	1.000	0.963	0.980
v_{PPM}	0.879	0.834	0.969	0.911	0.963	1.000	0.942
v_{VU}	0.921	0.838	0.987	0.939	0.980	0.942	1.000

TABLE 4.1: Correlation between all pairs of loudness features for the whole string quartet movement displayed in Fig. 4.5

All correlations are significant with a p-value $p \leq 10^{-12}$ (two-tailed test). The p-value is computed with

$$p = \text{tcdf}\left(\frac{r \cdot \sqrt{N-2}}{\sqrt{1-r^2}}\right) \tag{4.9}$$

with tcdf being Student's t cumulative distribution function.

The mean and standard deviation of a feature set — and thus the result of the computed correlation — may be influenced by extreme feature results at signal pauses at the beginning and end of the audio file. In order to evaluate these dependencies, the same calculation was carried out on the same data set but omitting features from start and end of the file, resulting in an overall number of 14000 observations. Table 4.2 shows the resulting correlation between pairs of features where the same tendencies as in the table above can be observed.

	$v_{Zw(DIN)}$	$v_{Zw(1387)}$	v_{RMS}	v_{1770}	v_{dBA}	v_{PPM}	v_{VU}
$v_{Zw(DIN)}$	1.000	0.969	0.898	0.915	0.871	0.878	0.895
$v_{Zw(1387)}$	0.969	1.000	0.877	0.899	0.847	0.865	0.866
v_{RMS}	0.898	0.877	1.000	0.941	0.993	0.983	0.978
v_{1770}	0.915	0.899	0.941	1.000	0.930	0.927	0.905
v_{dBA}	0.871	0.847	0.993	0.930	1.000	0.976	0.971
v_{PPM}	0.878	0.865	0.983	0.927	0.976	1.000	0.939
v_{VU}	0.895	0.866	0.978	0.905	0.971	0.939	1.000

TABLE 4.2: Correlation between all pairs of loudness features for an excerpt of the string quartet movement displayed in Fig. 4.5

The high values for the correlation indicate that many of the extracted features could probably be omitted without losing information. The ultimate goal should be to select the most relevant features, but this selection has to be psycho-acoustically motivated and is unknown for the current context. In order to reduce the number of features, a principal component analysis (PCA) has been applied to the loudness and timbre features for the example analysis (see Chap. 7).

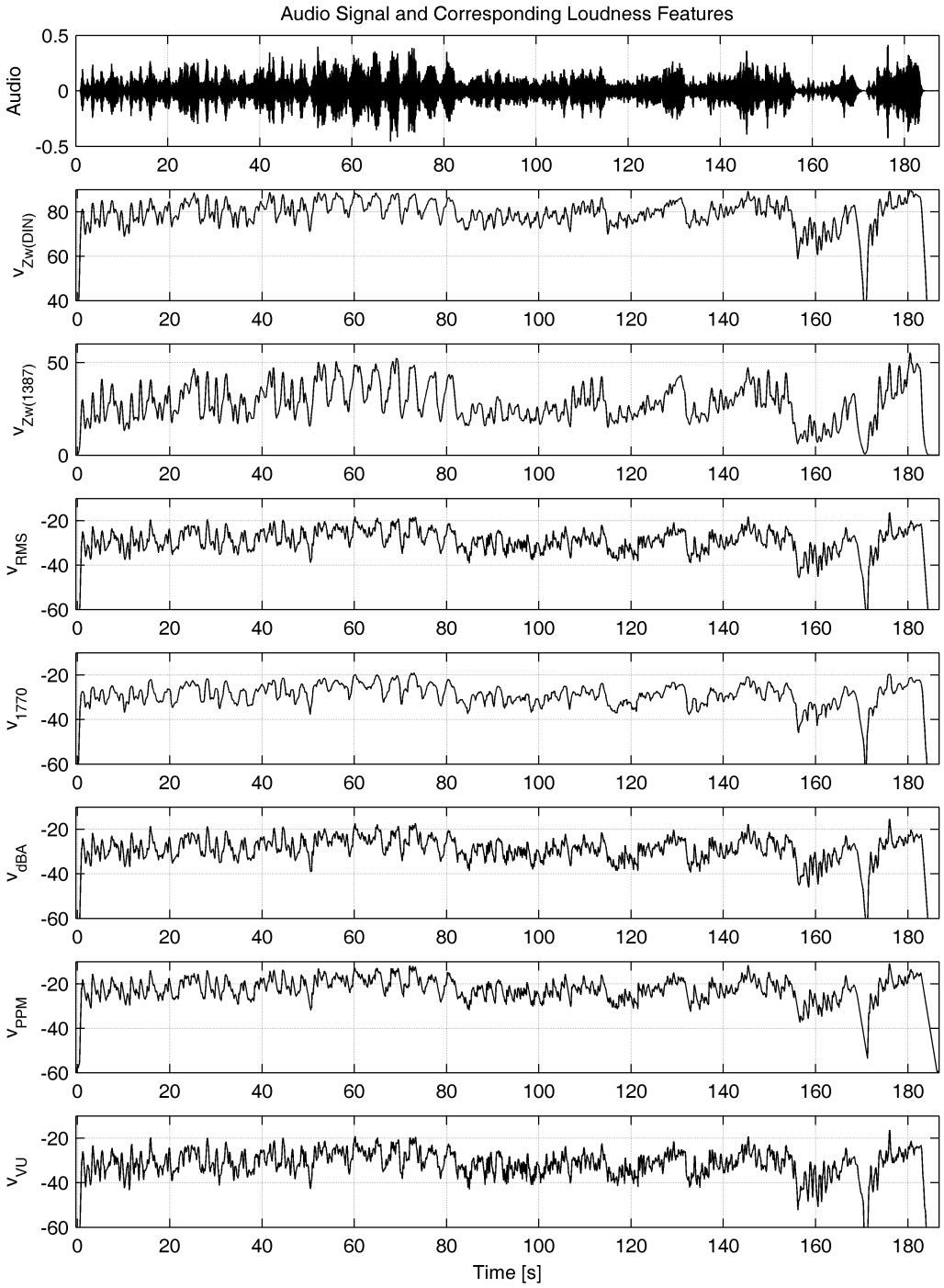

FIGURE 4.5: Input audio and loudness features for one example audio file (string quartet)

Chapter 5

Timbre Feature Extraction

The timbre of a sound is its *sound color*, its *quality* or its *texture*. Besides pitch and loudness, timbre is considered as "the third attribute of the subjective experience of musical tones" [RP82]. Timbre can be explained by two closely related phenomena, which will be referred to as *timbre quality* and *timbre identity*.

The timbre quality allows humans to group together different sounds originating from the same source such as two recordings made with the same instrument. Timbre identity enables the differentiation of two sounds with the same tone characteristics (loudness, pitch if available) played on two instruments. Thus, the quality represents general timbre properties of a sound ("sounds like a violin") while the timbre identity refers to instrument specifics ("one violin sounds better than the other").

In performance analysis, timbre is one of the least researched parameters. One of the reasons is that specific performance instructions do not exist concerning the specification of timbre, while for example for the specification of loudness various symbols and terms have been established. Instructions on timbre variations are usually given only implicitly either by describing the mood or by requiring specific playing techniques.

A study by Stolla is one of the rare examples of studying timbre characteristics in the context of music performances [Sto04]. He approached this topic by computing an overall octave band spectrum per recording and found only small differences between the analyzed recordings. Walker investigated the influence of instrumental timbre on several performance parameters such as timing, articulation, and dynamics [Wal04].

Loudness and pitch are unidimensional properties, as sounds with different loudness or pitch can be ordered on a single scale from quiet to loud and low to high, respectively. Timbre is a multidimensional property ([ZF67], [Moo97]); this complicates its definition. A good summary over the various attempts of the definition of the term timbre can be found in [San08]. The most prominent

example is probably the definition of the American Standards Association from 1960 that defined timbre as "that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar" [ASA60]. This definition has been criticized repeatedly by researchers mainly because it [Bre94]:

- does not attempt to explain what timbre is, but only what timbre is *not*, i.e. loudness and pitch,
- implies that timbre only exists for sounds with a pitch, implicating that for example percussive instruments do not have a timbre.

Helmholtz was probably the first to detect the dependency between the timbre of a sound and the relative amplitudes of the harmonics during the second half of the 19^{th} century [Hel70]. Although he noted other influences that play a role in defining the quality of a tone such as the "beginning" and "ending" of a sound, he restricted his definition of timbre ("Klangfarbe") to the harmonic amplitude distribution only.

Stumpf extended the definition of timbre by two more attributes [Stu90]. He named the relative amplitude of harmonics, the form and length of the initial transient time and note endings, and added sounds and noise as the third timbre-determining component.

Seashore restricted the term timbre to the harmonic structure that "is expressed in terms of the number, distribution, and relative intensity of its partials", but he additionally introduced the term *sonance*, referring to "the successive changes and fusions which take place within a tone from moment to moment" [Sea38]. This distinction did, however, not find broad acceptance by researchers. Instead, timbre is nowadays understood as phenomenon that takes into account both spectral patterns and temporal patterns ([Reu95], [Moo97]).

To investigate the number and characteristics of timbre dimensions, usually semantic differentials (SD) or multidimensional scaling (MDS) techniques are used. For the SD method, sounds are rated on scales that are characterized by pairs of opposite verbal attributes. For MDS, a group of listeners adjusts the timbre distance (or similarity) between pairs of stimuli and the result is mapped or fitted into a geometric space that has the number of dimensions specified by the investigator.

In the search of verbal attributes for the timbre of steady sounds, von Bismarck found that the timbre of his test set could be almost completely described by a rating on four scales: *dull—sharp*, *compact—scattered*, *full—empty*, *colorful—colorless* [Bis74b]. In a more detailed study on sharpness, he comes to the conclusion that the sharpness is determined by the position of energy concentration in the sound's spectrum [Bis74a].

Grey found three salient dimensions that correlated well to the spectral energy distribution, the low amplitude high frequency energy in the initial attack segment, and the synchronicity of the attack of higher harmonics combined with the level of spectral fluctuation [Gre77].

Wessel found two dimensions that could be represented by the spectral energy distribution and the nature of the onset transient [Wes78].

Iverson and Krumhansl found two dimensions as well with high correlation to the results of the Spectral Centroid (see below) and the amplitude envelope [IK93].

McAdams et al. identified three dimensions that correlated well with Spectral Centroid, logarithmic attack time, and Spectral Flux (see below) [MWD$^+$95].

Lakatos also found three dimensions and was able to find high correlation with the Spectral Centroid and the logarithmic attack time, but he failed to find a fitting acoustic correlate to the third dimension [Lak00].

Marozeau et al. found four dimensions, three of them with high correlation to Spectral Centroid, attack time and Spectral Spread (see below), and the fourth, less salient dimension, correlated with fundamental frequency in some cases [MCMW03].

Caclin et al. confirmed the high correlation of two dimensions of the timbre space with Spectral Centroid and logarithmic attack time; furthermore, they sometimes found — context dependent — the Spectral Flux to be correlated with a third dimension [CMSW05].

Apparently, there is an agreement on two dimensions of timbre that can be referred to as *brightness* and *impulsiveness*. The brightness (also: *sharpness*) represents spectral, the impulsiveness temporal aspects of timbre. Timbre is most definitely higher-dimensional, but the identification of more dimensions is difficult; the number and the nature of these higher dimensions can probably be assumed to be context-dependent.

A high correlation of the brightness dimension with the Spectral Centroid has been reported (compare also [SW06]) and the attack time or similar measures seem to correlate well with the impulsiveness. There is no agreement on the usability of the Spectral Flux as an acoustical correlate for one of the higher dimensions, however, there are indications that it may be useful in a specific context.

The research on timbre and its dimensions has been done almost exclusively on isolated (synthesized or instrumental) tones. The identification of the dimensions of the timbre space is even more complex for music recordings; timbre dimensions and possible acoustic correlates for complex input signals have yet to be identified. Furthermore, a parameter such as the attack time of a single tone ("impulsiveness") cannot be extracted from a complex mixture of

instruments and tones in recordings of polyphonic music — at least not in the general case.

Therefore, the choice of the implemented timbre features has been based on the following considerations:

- The feature can be extracted from a polyphonic music signal and its calculation does not require the availability of individual time signals for each single tone or the amplitudes of each harmonic of this single tone,
- the feature is related to the acoustic features that show good correlation with the identified timbre dimensions for single tones, and/or
- the feature has shown to be of relevance in a related context. For example, many features that describe the spectral shape of a signal block (Spectral Centroid, Spectral Rolloff, Spectral Spread, MFCCs, etc.) have been successfully used for audio genre classification [BL04] or music similarity calculations [PFW05]. If these timbre-related features are reliable enough to be used for such tasks, they can be assumed to be of some significance in our application.

5.1 Implemented Features

All timbre features are extracted from overlapping blocks of the audio signal and are calculated via the STFT $X(k, n)$ of each block.

5.1.1 Spectral Rolloff

The *Spectral Rolloff* is a measure of the bandwidth of the audio signal. It is defined as the frequency bin below which the accumulated magnitudes of the STFT reaches 85 % of the overall sum:

$$v_{SR}(n) = i \left| \sum_{k=0}^{i} |X(k,n)| = 0.85 \cdot \sum_{k=0}^{\mathcal{K}/2-1} |X(k,n)| \right. \tag{5.1}$$

Optionally, both sums could also start from a higher frequency bin than 0 to avoid the clustering of the results at low values, for example due to DC offset problems.

Low values indicate a low audio bandwidth.

5.1.2 Spectral Flux

The *Spectral Flux* measures the rate of change of the spectral shape:

$$v_{SF}(n) = \sqrt{\sum_{k=0}^{\mathcal{K}/2-1} \left(|X(k,n)| - |X(k,n-1)|\right)^2}. \tag{5.2}$$

It can be interpreted as a simplified approximation to the sensation *roughness* that is modeled by Zwicker and Fastl as quasi-periodic changes in the excitation pattern levels [ZF99].

Low values indicate steady-state input signals and low roughness.

5.1.3 Spectral Centroid

The *Spectral Centroid* describes the gravity center of spectral energy. As has been shown above, it is closely related to the brightness (or: sharpness) of a single tone. Zwicker and Fastl presented a model of sharpness that uses the excitation patterns to compute the sharpness: it is the integral of the specific loudness values in each critical band weighted with the bark frequency of this band and divided by the overall loudness, i.e. the unweighted integral of all specific loudness results [ZF99]. They also apply an additional weighting function to increase the influence of high critical bands.

For the Spectral Centroid, no excitation patterns but only the spectral power spectrum is used. The critical band scale is approximated by applying a logarithm to the frequencies with a reference point of 1 kHz:

$$v_{SC}(n) = \frac{\sum_{k=k(f_{min})}^{\mathcal{K}/2-1} \log_2\left(\frac{f(k)}{1000\,\mathrm{Hz}}\right) \cdot X(k,n)^2}{\sum_{k=k(f_{min})}^{N/2-1} X(k,n)^2} \tag{5.3}$$

In this specific implementation, all bins corresponding to frequencies below 62.5 Hz are combined to one band with a mid frequency of 31.25 Hz. Various definitions of the Spectral Centroid can be found in the literature: the magnitude spectrum or the excitation patterns are alternatives to the power spectrum; also a linear frequency scale or a bark scale can be used. The usage of the spectral power instead of the spectral magnitude results in less contribution from spectral components with low volume to the Spectral Centroid. The presented definition is similar to the *AudioSpectrumCentroidType* as defined in the MPEG-7 standard [ISO02].

Low values correspond to low brightness.

5.1.4 Spectral Spread

The *Spectral Spread*, sometimes also referred to as *instantaneous bandwidth*, describes how the spectrum is concentrated around the Spectral Centroid and represents a more technical description of spectral shape. The following definition is also based on the MPEG-7 standard [ISO02].

$$v_{SS}(n) = \sqrt{\frac{\sum_{k=k(f_{min})}^{\mathcal{K}/2-1} \left(\log_2\left(\frac{f(k)}{1000\,\mathrm{Hz}}\right) - v_{SC}(n)\right)^2 \cdot X(k,n)^2}{\sum_{k=k(f_{min})}^{\mathcal{K}/2-1} X(k,n)^2}} \tag{5.4}$$

If the spectral power is concentrated closely around the Spectral Centroid, the Spectral Spread computation will yield low values.

5.1.5 Mel Frequency Cepstral Coefficients

The *Mel Frequency Cepstral Coefficients* ($MFCC$s) are a compact representation of the shape of the spectral envelope of an audio signal. Their calculation takes into account the nonlinear human perception of pitch by utilizing the Mel-Scale. The MFCCs have been introduced in the field of speech signal processing [RJ93] and have been found to be useful in music signal processing as well ([TC00], [PDW03], [BL03], [JCMJ06]).

The implementation of MFCC calculation is based on Slaney's Matlab Auditory Toolbox and consists of the following basic steps [Sla98]:

- map the magnitude spectrum $X(k,n)$ to the Mel scale with a set of overlapping triangular windows
- compute the logarithm of each resulting Mel band
- compute a Discrete Cosine Transform (DCT) to the Mel bands

It has been shown that a small subset of the resulting MFCCs already contains the principal information in the context of audio signal classification ([TC02], [MB03]), so for our purpose we restrict ourselves to the first four coefficients to keep the number of features small.

	v_{SF}	v_{SR}	v_{SC}	v_{SS}	v_{MFC0}	v_{MFC1}	v_{MFC2}	v_{MFC3}
v_{SF}	1.000*	-0.042	0.168	0.014	0.660*	0.071	0.057	-0.071
v_{SR}	-0.042	1.000*	-0.062	0.428	-0.282	-0.327	0.117	-0.102
v_{SC}	0.168	-0.062	1.000*	-0.043	0.428	-0.303	-0.165	-0.006
v_{SS}	0.014	0.428	-0.043	1.000*	0.038	-0.084	0.167	0.028
v_{MFC0}	0.660*	-0.282	0.428	0.038	1.000*	0.345	0.109	-0.181
v_{MFC1}	0.071	-0.327	-0.303	-0.084	0.345	1.000*	0.427	-0.183
v_{MFC2}	0.057	0.117	-0.165	0.167	0.109	0.427	1.000*	0.550*
v_{MFC3}	-0.071	-0.102	-0.006	0.028	-0.181	-0.183	0.550*	1.000*

TABLE 5.1: Correlation between all pairs of spectral features for the whole string quartet movement displayed in Fig. 5.1

	v_{SF}	v_{SR}	v_{SC}	v_{SS}	v_{MFC0}	v_{MFC1}	v_{MFC2}	v_{MFC3}
v_{SF}	1.000*	0.081	-0.040	-0.020	0.660*	0.134	0.082	-0.078
v_{SR}	0.081	1.000*	0.633*	0.398	0.157	-0.405	0.053	-0.072
v_{SC}	-0.040	0.633*	1.000*	0.023	0.021	-0.382	-0.124	-0.051
v_{SS}	-0.020	0.398	0.023	1.000*	0.138	-0.013	0.170	0.067
v_{MFC0}	0.660*	0.157	0.021	0.138	1.000*	0.508*	0.206	-0.273
v_{MFC1}	0.134	-0.405	-0.382	-0.013	0.508*	1.000*	0.431	-0.185
v_{MFC2}	0.082	0.053	-0.124	0.170	0.206	0.431	1.000*	0.596*
v_{MFC3}	-0.078	-0.072	-0.051	0.067	-0.273	-0.185	0.596*	1.000*

TABLE 5.2: Correlation between all pairs of spectral features for an excerpt of the string quartet movement displayed in Fig. 5.1

5.2 Example Results

Fig. 5.1 displays the extracted feature values for an example file that consists of string quartet movement.

As for the loudness features, the normalized covariance between all pairs of features has been computed with Eq. (4.8), first with all observations and then omitting observations from start and end of the audio file (which was the same as used to display the loudness features). No pair of features is highly correlated, neither for the complete set nor for the truncated set of observations (see Tables 5.1 and 5.2). Results marked with an asterisk have an absolute correlation $|r| \geq 0.5$ and are at the same time significantly correlated with a p-value $p \leq 0.01$ (two-tailed test).

As already mentioned in the previous Chap. 4, a PCA over the whole feature set including the dynamics features can be found in Chap. 7 to analyze the dimensionality of the complete feature space.

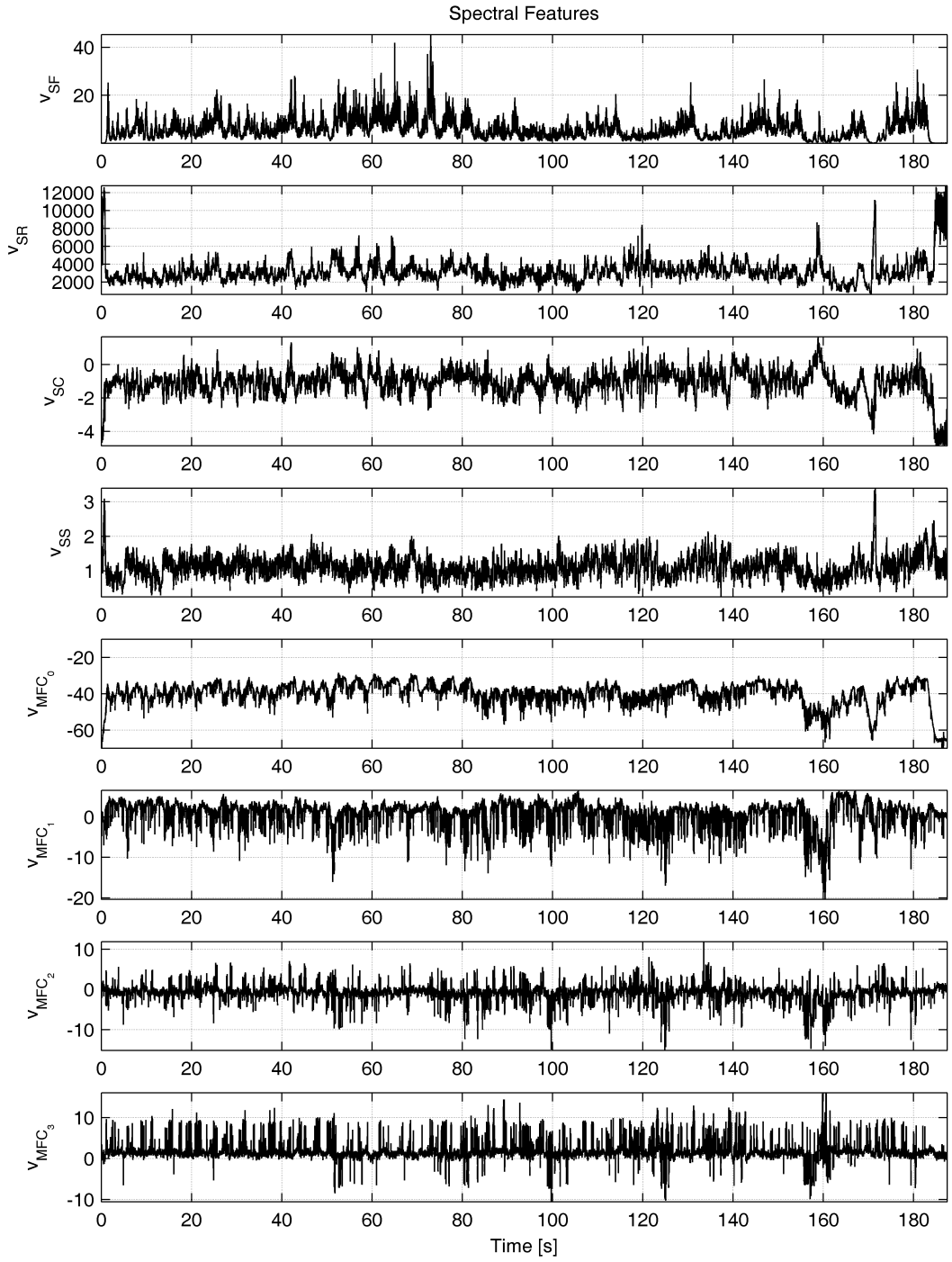

FIGURE 5.1: Spectral features for one example audio file (string quartet)

Chapter 6

Software Implementation

The presented algorithms have been implemented in the programming language *C++* [ISO03]. The resulting source code is able to compile cross-platform (Windows, MacOS X, Linux) and was thoroughly tested on 32-bit Microsoft Windows operating systems. To decrease development time, three open source libraries have been made use of: FLTK [S+08] as GUI (Graphical User Interface) toolkit library, PortAudio [B+08] for sound IO, and SndLib [Sta08] for wave file parsing. An additional library has been used for performance optimization (see Sect. 6.1.2).

The software consists of two binary executables as depicted in Fig. 6.1; one to deal with the performance data extraction and the second to provide means for the qualitative analysis of the extracted performance data.

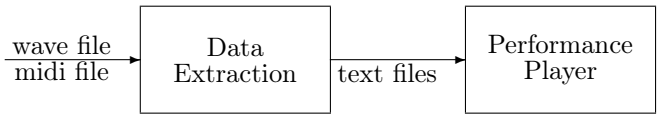

FIGURE 6.1: Two processing stages of the implemented software

The main advantage of having two software solutions instead of one is the openness of the resulting system. One the one hand, the extracted data can be easily visualized and analyzed with any other software solution, on the other hand separate programs can be used to extract data that can be played and visualized with the *Performance Player*. The text file format used for the extracted data allows easy import, export, and parsing with other applications. While the implementation of such options in a monolithic software application would of course be possible, the handling of the application might become more complicated for the user and requires more in-depth thought of user interface design.

While this chapter is focused on implementation details, Chap. B describes the usage of the software.

6.1 Data Extraction

The data extraction stage consists of two independent parts - the tempo extraction and the low level feature extraction. The first part deals with the calculation of the audio-score synchronization while the second part extracts all timbre and loudness features. Both processes are implemented as non-real-time processes — as this is required by the audio-to-score alignment algorithm — and are combined in a single command line application.

The tempo extraction requires two input files, the audio and the MIDI file. The audio file, representing the performance, should contain audio data in any file format supported by the used library SndLib (e.g. `.wav`, `.aiff`, etc.); it should provide good technical quality (reasonable audio bandwidth at a sample rate higher than 32 kHz and a relatively low background noise level) and a performance being reasonably free of errors. The MIDI file should be of MIDI formats 0, 1, or 2 and should contain a quantized form of the score, as opposed to a MIDI performance of the score. Furthermore, it is assumed that the MIDI file contains the same structure as the performance, meaning that e.g. repetitions appear equally in both score (MIDI) and performance (audio) representation.

An optional text file input allows to set reference synchronization points (audio time at a specific MIDI tick) to force the alignment path to go through these points. This gives the user the possibility to set predefined points either to fix the alignment at points where a previous calculation yielded inaccuracies or to set obvious synchronization points a priori. Defining one or more synchronization points results also in a significant performance increase (for example, setting a synchronization point at the middle of the performance will practically halve the runtime of the tempo extraction).

All output data is written to text files in a user-defined directory. The alignment result text file contains a list of the computed synchronization points between audio and MIDI, i.e. seconds and MIDI beats.

The timbre and loudness feature extraction requires only the input audio file and outputs a text file with an ordered list of time stamps and feature values. To enable easy and fast addition of new low level features, a plugin interface for low level feature extraction has been designed and implemented: FEAPI (Feature Extraction Application Programmer's Interface).

6.1.1 FEAPI

Many of the features to be extracted for music performance analysis, especially the ones for dynamics and timbre analysis, are similar or even identical to features that are used in Audio Content Analysis (ACA) as so-called low level features. These low level features can serve as building blocks for constructing higher level, more semantically meaningful properties of the music derived from the audio signal. A low level feature can thus be defined as one or more values extracted from the audio signal that can be used to describe a property of the signal but is not necessarily musically or musicologically meaningful all by itself.

Many applications in the ACA field require a large number of these features to be extracted from the audio signal. Different applications or research projects need similar or identical low level features, utilizing their own implementation of well-known and often algorithmically not too complex features. Redundant work is being done to "reinvent the wheel" each time. Furthermore, the integration of already implemented features in a new application is usually time-consuming even if the source code is available, and almost impossible if this is not the case.

A commonly accepted plugin Application Programmer's Interface (API) for low level features would enable the reuse of already implemented features without additional effort. Furthermore, feature extraction plugins could be exchanged between projects, researchers and companies, if required also in binary format to protect the intellectual property of the development party. A well-defined API can also speed up the development process, since the implementation can focus more on the algorithmic aspects and less on implementation issues such as API design. Therefore, a joint effort has been initiated with participants from four institutions (*Ghent University*, Belgium, *IRCAM*, Paris, France, *Technical University of Berlin*, Germany and *zplane.development*, Berlin, Germany) to define and implement such an API with the name *FEAPI — Feature Extraction API* [LET05].

A plugin is considered to be a library that can be linked dynamically at runtime utilizing a previously defined interface. For the host application, using a plugin means to load the library and to use the exported functions during runtime.

6.1.1.1 Related Work

At the time of development and publication of this API, there was no widely accepted audio feature extraction plugin API in use by the ACA community.

Marsyas is an audio analysis and synthesis framework with emphasis on ACA which allows a user to extend the framework by deriving from a base class tightly integrated in the framework [TC00]. As such, it does not define a true feature extraction plugin API. *Maaate!* is an audio analysis toolkit that

provides a plugin interface, but has a strong focus on processing sound files in MPEG format [PP01]. The source code of both is published under the GNU General Public License (GPL, [Fre08]), thus enforcing the publication of source code for all plugin or host implementations, which may prevent them from being used in a commercial context where source code distribution is not appropriate.

More recently and after the initial publication of FEAPI, another plugin API dedicated to feature extraction was published that provides similar functionality and properties as FEAPI: Vamp [CLSB06].

Besides these ACA-related APIs, several plugin APIs are commonly used in the world of audio signal processing and virtual instruments. These APIs are mainly designed for transforming an audio stream into a new audio stream (effect processing) or for generating an audio stream in reaction to incoming MIDI events (virtual instruments). They are not easily adaptable to the demands of audio feature extraction. Examples of such APIs are *LADSPA* (Linux Audio Developer's Simple Plugin API, [Fur08]), *VST* (Virtual Studio Technology by Steinberg, [Ste08]) and *AU* (Audio Units by Apple, [App08]). Some influences from these API designs can be found in FEAPI.

The *VST*-SDK additionally provides an offline extension allowing audio data analysis. This extension could basically be used for feature extraction. While this had the advantage of compatibility with some already available hosts, the VST-offline interface has the following disadvantages that make the definition of a dedicated feature extraction plugin API reasonable:

- *restriction of capabilities*: the offline interface is per definition not able to handle audio streams, only audio files. Furthermore, the handling and storage of large and complex features sets is laborious.
- *complexity of plugin implementation*: the API requires working with audio file handles, and there is a bidirectional communication between plugin and host.
- *non-open license*: the definition of the future capabilities and extensions is under control of a company and therefore cannot easily be influenced by the requirements of researchers. Open source projects are not allowed to distribute the source files of the SDK with their code, so in the case of source code distribution every possible user theoretically has to sign an individual license agreement with the license holder.

6.1.1.2 Requirements and Considerations

Every attempt to specify an application programming interface demands careful consideration of the required functionality and capabilities as well as of usability and simplicity. Usually, a compromise between capabilities and ease of use has

to be found since they may contradict each other. The (non-trivial) technical requirements for the capabilities of the feature extraction plugin API were defined as:

- support for different and possibly varying sample rates of the extracted features,
- support for multiple independent instances of each plugin,
- support for multidimensional features,
- high probability of unique plugin identification by the host without a registration process,
- support for the calculation of multiple features in one plugin if required by the developer,
- support for sufficient timing information to allow synchronization of features with different sample rates and
- push-style processing of audio buffers (data source can be anything: files, live streams, etc.).

The following restrictions were agreed upon to allow for simple usage and implementation of the API. They may have both technical and usability reasons:

- memory allocated internally by the plugin is never used outside the plugin, and shared memory has to be allocated by the host,
- the plugin cannot call host functions meaning that the host has to poll for status requests etc.,
- no file handles are used in the API,
- no developer-specified graphical user interface (GUI) is required to run the plugin,
- only one data type (namely `float`) is allowed for inputs, outputs and parameters,
- no thread safety of the API: the host has to ensure that e.g. the request for results does not interfere with a running process call.

To allow cross-platform compatibility and integration in as many programming languages as possible, the plugin interface was chosen to be defined in the programming language *C*. *C* and *C++* are commonly used by researchers and companies in the audio signal processing context and compilers are available for practically all possible target platforms. Besides the API itself, a software development kit (SDK) providing *C++* wrapper classes is available. These classes allow easy access to a plugin from the host side as well as easy implementation of plugins by inheriting from a base class plugin on the plugin side.

6.1.1.3 Design and Architecture

Basically, the API provides two types of data inputs/outputs, called *signal* and *parameter*. Both have to be of the data type `float`. A signal can be both an input or a result. Input signals, which are usually thought to be audio signals, are restricted to a constant sample rate. Parameters are used to change the plugin properties. Signal and parameter properties are defined by means of structures that are displayed in Fig. 6.2. Both structures provide extensive plain text information, information about range and quantization as well as other useful data.

```
typedef struct
{
    char    acName[1024];
    char    acUnit[1024];
    char    acDescription[4096];
    float   fRangeMin;
    float   fRangeMax;
    float   fQuantizedTo;
    float   fSampleRate;
} FEAPI_SignalDescription_t;
```

```
typedef struct
{
    char    acName[1024];
    char    acUnit[1024];
    char    acDescription[4096];
    float   fRangeMin,
            fRangeMax,
            fDefaultValue;
    float   fQuantizedTo;
    int
bIsChangeableInRealTime;
} FEAPI_ParameterDescription_t;
```

FIGURE 6.2: Structures for the description of FEAPI signal and parameter properties

The input signal is passed to the plugin by simply pushing new buffers of data to the process function. The host can poll for new results at any time, taking into account the thread safety issues mentioned above.

Plugins can be distinguished by a quintuple of information: the library name itself, the plugin name string, the plugin vendor (implementor, manufacturer) string, the vendor-specific plugin ID and the vendor-specific plugin version info.

6.1.1.4 Usage

The plugin function `FEAPI_CreatePluginInstance` has to be called to create a new instance of the plugin and let it do some basic internal initialization.

The functions `FEAPI_GetPluginAPIVersion` and `FEAPI_GetPluginCanDo` allow the host to retrieve some plugin-specific information including API version, supported number of channels or supported sample rates. This information can be used by the host to decide whether it supports this type of plugin or not, and if it does, to address the plugin in the correct way.

The call of `FEAPI_InitializePlugin` is required to initialize the plugin with the non-varying parameters, the input sample rate and the number of audio channels. CPU-intensive calculations necessary for internal initialization of

buffers, filters, etc. should be done here too. Furthermore, some vendor-specific user data can be handed over to the plugin. If initialization fails, the plugin can not be used and the host has to destroy the plugin instance.

To retrieve information about the available plugin parameters and the calculated features, the functions `FEAPI_GetPluginResultDescription` and `FEAPI_GetPluginParameterDescription` can be used. Figure 6.2 provides some insight into the available information.

`FEAPI_ProcessPlugin` is the function that performs the actual processing. The host simply needs to maintain a continuous stream of audio data blocks that are handed over to the plugin by calling this function. Additionally, a time stamp for the input data is passed adjoins the data.

The host can then check whether a result is available or not. If so, it can check its size and the function `FEAPI_GetPluginResult` can be called to obtain the result. A time stamp for the result is returned as well.

`FEAPI_ProcessPluginDone` has to be called when no more audio data is available to signal to the plugin that all processing is done and allow it to do some final processing based on what it has left in its internal buffers if needed.

Finally, `FEAPI_DestroyPluginInstance` destroys the plugin instance.

Besides these, a few additional interface functions are available; these are explained on the FEAPI web site together with additional documentation [LET08].

6.1.1.5 Software Development Kit (SDK)

The API itself consists of a *C* header file containing type definitions of all data types that are passed through the interface and all functions which a plugin must provide. Together with the actual API, a *C++* SDK has been developed which wraps all the functions of the plugins and the API in *C++* classes. With these wrappers, a FEAPI plugin instance can be handled as a *C++* object. The specific plugin class is derived from a plugin base class (`FEAPI_CPluginBaseClass`). The methods provided by the base class, directly representing the functions specified by the API, are re-implemented by the specific plugin class. This structure is similar to VST and allows an easy and fast plugin development process. The methods of the plugin object are called by *C*-style stub functions which are also provided by the SDK. These stub functions do nothing more than translating the *C* function calls into *C++* method calls.

To ensure that the API and the related source code are easily available, a project has been started on SourceForge.net[1] [LET08]. SourceForge provides several useful software management services such as the revision control SVN,

[1] `www.sourceforge.net`

bug tracking, mailing lists etc. Source code and in-depth information on the presented API is publicly available for download. The source code is licensed under a BSD-style license [Ope08], which is a simple, permissive and widely spread license. At the same time, the BSD license is compatible with the GNU GPL [Fre08] and the sources can be used — although under some minor restrictions — in commercial applications as well.

In summary, the presented API offers a solution for the technical requirements of low level feature extraction in an ACA context, as well as platform independence, a simple interface and an open license. The API provides a push-style interface allowing live streams as well as file streams. It supports multidimensional features to be extracted with constant as well as varying sample rates and provides sufficient information for the time synchronization of audio and features. The provided SDK, including example plugins and a simple command-line-based host, should allow a steep learning curve.

6.1.2 Performance Optimizations

In order to achieve reasonable application runtime for the rather complex and workload-consuming algorithms implemented, the signal processing parts of the application have been optimized for performance, i.e. for low workload. This was done in a fast and simple way by using a commercially available cross-platform vector optimization library by the company zplane.development[2]. This library contains Altivec and SSE, SSE2, and SSE3 optimizations for vector operations with complex and real (floating point) data, an optimized Fast Fourier Transformation (FFT) as well as several trigonometric and logarithmic functions that can be applied to data vectors. It ensures optimal performance on various systems due to CPU runtime detection that allows to dispatch each function call to its appropriate CPU-optimized implementation. Furthermore, the library also offers a generic C^{++} implementation of all functions to ensure compatibility with platforms without SIMD instruction set.

6.2 Performance Player

The *Performance Player* software loads the previously extracted data of multiple performances of the same piece from the text files and provides the means for qualitative performance analysis. In the literature, different approaches to the analysis and the comparison of several performances can be found. One of the simplest solutions is to plot the data (tempo, loudness, or timbre) over time, with the disadvantage of not being able to visually align the data of different performances. Hence, the "standard" visualization that is used in nearly all performance studies dating from the 1930s ([Har32], [Sea38]) until

[2] `www.zplane.de`

today is to plot the data over beats or bars, or in general over a score-inherent pulse. This allows to compare different performance data values with respect to their score position instead of their point in time and was also chosen as data visualization for the Performance Player, where the performance data is plotted with reference to the beats extracted from the MIDI file to ensure identical length and easy comparison of all performances.

An alternative display method would have been to plot the data as deviations from a mechanic rendition ([KC90], [Pal89]), however, this poses the problem of identifying an appropriate "mean" value for tempo, loudness, or specific timbre features of the mechanic rendition. It also assumes that the analysis of timing can be independent of the overall performance tempo, an assumption with questionable validity (compare Chap. 2.2).

Other approaches attempt to visualize both the short- and the long-term structure of loudness and timbre variations. Langner et al. presented a pseudo-three-dimensional loudness plot with the abscissa representing the time, the ordinate representing the integration time of a smoothing filter and the color of each plot point the corresponding loudness level [LKSW00]. The smoothing filter integration time dimension visualizes the loudness rendition on different levels ranging from short term to long term variation.

Segnini proposed the "Timbrescape" for the visualization of timbre variations [Seg06]. He defined the timbre to consist of three dimensions, the fundamental amplitude, the harmonics amplitude, and the sum of the remaining spectral components and assigns a base color (RGB: red, green and blue) to each of these dimensions. The value of a timbre parameter is then mapped to the intensity of the corresponding color. Every signal excerpt can thus be represented by one color composed of its components red, green and blue. The first row of the Timbrescape contains only one single colored dot that represents the average (timbre) intensities for the whole file. The next rows contain increasingly more color dots with each representing a shorter signal excerpt. The final shape of the Timbrescape is triangular and provides the highest time resolution at the lowest row.

Dixon et al. introduced the "Performance Worm" for the real-time performance visualization in the tempo-loudness plane ([DGW02], [LG02]). For each observation point in time, the corresponding tempo and loudness values are plotted on this plane with the abscissa representing the tempo axis and the ordinate the loudness axis. The transparency of the plotted dots increases with time, resulting in a worm shape with recent dots representing more recent events being dark and preceding dots slowly fading away.

The tempo-loudness plane has also been used in other publications of the same Austrian research group. Widmer used performance-worm-like representations of phrases with the length of four beats to generate a so-called performance alphabet that consists of prototype gestures or shapes to represent typical phrasing of individual artists [WZ04].

Other representations are more suited for specific research questions. For example, histograms can be used for a statistical analysis of the tempo data (Goebl et al. [GD01]).

Sapp proposed a "Scape Plot" to visualize correlations between performances [Sap07]. Similar to the Timbrescape, it has a triangular shape and increasing time resolution from top to bottom. However, the Scape Plot is used to analyze the correlations between a single performance and a large data set of performances; here, one color represents one specific performance. The single dot in the first row is then colored in the specific color of the performance from the data set that yields the highest correlation to the analyzed performance. The dots of the following rows are also colored according the performance with the highest correlation, but the correlation length decreases as the number of dots per line increase, i.e. only parts of the pairs of performances are correlated. The more a Scape Plot consists of a single color, the more similar are the analyzed performance and the performance corresponding to this dominant color. In a later publication, Sapp argues for the suitability of this Scape Plot and the calculated correlation results, respectively, for more detailed inter-performance similarity measurements [Sap08].

The *Performance Player* presented here offers the following features:

- display of one selected performance parameter (tempo, loudness or timbre features, ...) for multiple performances plotted on a MIDI beat axis,
- possibility to zoom in and out of the displayed graphs,
- optional and adjustable smoothing of the feature results for the better visualization of various levels of long term and short term variations of each feature (see Sect. 6.2.1),
- calculation of overall features to support the identification of interesting overall properties or differences between different performances (see Sect. 6.2.2),
- play-back engine that allows to play the audio performances to permit subjective perceptual comparisons. The engine plays the files on a linear time-scale but keeps track of the corresponding beat position, allowing to switch instantly between different performances while remaining at the same corresponding score position.

6.2.1 Smoothing Filter

A smoothing filter is (optionally) applied to the feature data. The smoothing filter is a zero-phase filter that is based on a single-pole IIR filter but implemented in an anti-causal way by processing the input data in both forward and

reverse direction. Since the single-pole filter is therefore applied to the feature curve two times, it provides the steepness of a two-pole filter. The single pole filter equation is:

$$v'(m) = \alpha \cdot v'(m-1) + (1-\alpha) \cdot v(m) \tag{6.1}$$

with α being a value between $0 \ldots 1$ adjustable by the user.
The filter implementation requires an input signal that is sampled at equidistant intervals. Although the feature values are displayed in a non-equidistant fashion, i.e. over beats instead of over time, the computation of the filter output is carried out assuming a linear *time* scale. Alternatively, it would have been possible to estimate the features values $v(n)$ at beat positions with appropriate interpolation approaches; this had the advantage of computing the filter on sampling intervals more meaningful from a musical point of view. A short informal evaluation resulted in only minor differences between the results calculated by these two approaches (the second one implemented with a simple linear interpolation), so for the sake of simplicity, the first approach without interpolation has been selected.

6.2.2 Overall Results for each Feature

For each extracted parameter or feature, overall values are computed to describe properties of the extracted feature curve. Many of these features are known to be useful for statistical signal description. The length of the feature curve is $\mathcal{M}$, its discrete derivative is $v_D(m)$. As for the smoothing filter, the overall feature calculation in the *Performance Player* is carried out on the linear time scale rather than on the beat scale.

6.2.2.1 Overall Maximum

The overall maximum is the highest feature value over the whole performance.

$$Max = \max_{0 \le m \le \mathcal{M}-1} v(m) \tag{6.2}$$

$$Max_D = \max_{0 \le m \le \mathcal{M}-2} v_D(m) \tag{6.3}$$

6.2.2.2 Overall Minimum

The overall minimum is the lowest feature value over the whole performance.

$$Min = \min_{0 \le m \le \mathcal{M}-1} v(m) \tag{6.4}$$

$$Min_D = \min_{0 \le m \le \mathcal{M}-2} v_D(m) \tag{6.5}$$

6.2.2.3 Overall Range

The overall range shows the amplitude range that is covered by the feature.

$$Range = Max - Min \tag{6.6}$$
$$Range_D = Max_D - Min_D \tag{6.7}$$

6.2.2.4 Overall Histogram Maximum

The overall histogram maximum represents the most frequent feature value, or more exactly the mean of the range of feature values that occurs most frequently. This could for example be used as a simple approximation of the "modal tempo", as proposed by Goebl et al. [GD01].

From the extracted features over time $v(m)$ (and their derivative $v_D(m)$, respectively), a histogram h — similar to a discrete amplitude density function — is calculated. The number of histogram classes $\mathcal{G}$ is

$$\mathcal{G} = \min\left[50, \frac{\mathcal{M}}{4}\right] \tag{6.8}$$

to ensure that the number of histogram classes is always significantly smaller than the number of feature values. Each class represents the number of occurrences of feature values in a specified range, and the range is computed by $\kappa = {}^{(Max-Min)}/_{\mathcal{G}}$. The sum of all histogram classes equals the overall number of feature values $\mathcal{M}$. The result is then

$$Max_{Hist} = Min + \frac{\kappa}{2} + \kappa \cdot \underset{0 \le g \le \mathcal{G}-1}{\operatorname{argmax}}\left(h(g)\right), \tag{6.9}$$

which is the feature value representing the class with the maximum number of occurrences.

6.2.2.5 Overall Arithmetic Mean

This is the overall arithmetic mean value.

$$AMean = \frac{1}{\mathcal{M}} \sum_{m=0}^{\mathcal{M}-1} v(m) \tag{6.10}$$
$$AMean_D = \frac{1}{\mathcal{M}-1} \sum_{m=0}^{\mathcal{M}-2} v_D(m) \tag{6.11}$$

6.2.2.6 Overall Geometric Mean

This is the overall geometric mean value.

$$GMean = \sqrt[\mathcal{M}]{\prod_{m=0}^{\mathcal{M}-1} v(m)} \tag{6.12}$$

$$GMean_D = \sqrt[\mathcal{M}-1]{\prod_{m=0}^{\mathcal{M}-2} v_D(m)} \tag{6.13}$$

6.2.2.7 Overall Harmonic Mean

This is the overall harmonic mean value.

$$HMean = \frac{\mathcal{M}}{\sum_{m=0}^{\mathcal{M}-1} \frac{1}{v(m)}} \tag{6.14}$$

$$HMean_D = \frac{\mathcal{M}-1}{\sum_{m=0}^{\mathcal{M}-2} \frac{1}{v_D(m)}} \tag{6.15}$$

6.2.2.8 Overall RMS

The root mean square value is related to the power of the feature curve.

$$RMS = \sqrt{\frac{1}{\mathcal{M}} \sum_{m=0}^{\mathcal{M}-1} v(m)^2} \tag{6.16}$$

$$RMS_D = \sqrt{\frac{1}{\mathcal{M}-1} \sum_{m=0}^{\mathcal{M}-2} v_D(m)^2} \tag{6.17}$$

6.2.2.9 Overall Standard Deviation

The standard deviation is a measure for the variation of the feature signal around its (arithmetic) mean value.

$$STD = \sqrt{\frac{1}{\mathcal{M}} \sum_{m=0}^{\mathcal{M}-1} \Big(v(m) - AMean \Big)^2} \tag{6.18}$$

$$STD_D = \sqrt{\frac{1}{\mathcal{M}-1} \sum_{m=0}^{\mathcal{M}-2} \Big(v_D(m) - AMean_D \Big)^2} \tag{6.19}$$

6.2.2.10 Overall Skewness

The skewness, also referred to as third central moment of a variable divided by the cube of its standard deviation, is a measure of asymmetry of the probability density function. If the feature values are distributed symmetrically around their mean, the skewness will be zero. If the data is more spread out to the left it will be negative while the results will positive for data that is spread out to values above the mean value.

$$Skew = \frac{1}{STD^3 \cdot \mathcal{M}} \sum_{m=0}^{\mathcal{M}-1} \Big(v(m) - AMean \Big)^3 \tag{6.20}$$

$$Skew_D = \frac{1}{STD^3 \cdot (\mathcal{M}-1)} \sum_{m=0}^{\mathcal{M}-2} \Big(v_D(m) - AMean_D \Big)^3 \tag{6.21}$$

6.2.2.11 Overall Kurtosis

The Kurtosis is a measure of 'non-gaussianity' of a random variable and is also referred to as its fourth central moment divided by the fourth power of the standard deviation.

$$Kurt = \frac{1}{STD^4 \cdot \mathcal{M}} \sum_{m=0}^{\mathcal{M}-1} \Big(v(m) - AMean \Big)^4 \tag{6.22}$$

$$Kurt_D = \frac{1}{STD^4 \cdot (\mathcal{M}-1)} \sum_{m=0}^{\mathcal{M}-2} \Big(v_D(m) - AMean_D \Big)^4 \tag{6.23}$$

A Kurtosis result higher than 3 indicates a density distribution with a more acute peak around the mean than a Gaussian distribution, a result lower than 3 indicates a smaller peak.

6.2.3 Graphical User Interface

Figures 6.3, 6.4, and 6.5 present the Graphical User Interface of the *Performance Player* while displaying the tempo curve, the smoothed Zwicker-Loudness and the smoothed Spectral Flux, respectively. The calculated overall values for one selected file are being displayed in the bar below the graph. This bar can be hidden or collapsed to provide a larger interface area for the curve display; in all three examples the file open section on the left hand side of the interface has already been collapsed.

FIGURE 6.3: Performance Player Graphical User Interface with the tempo curve of three string quartet performances

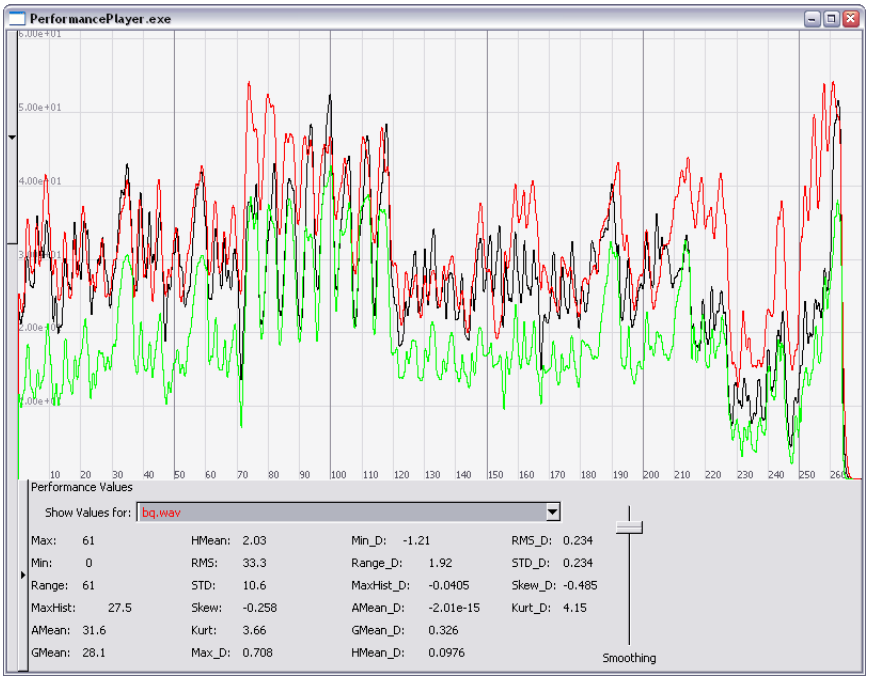

FIGURE 6.4: Performance Player Graphical User Interface with the smoothed Zwicker-Loudness curve of three string quartet performances

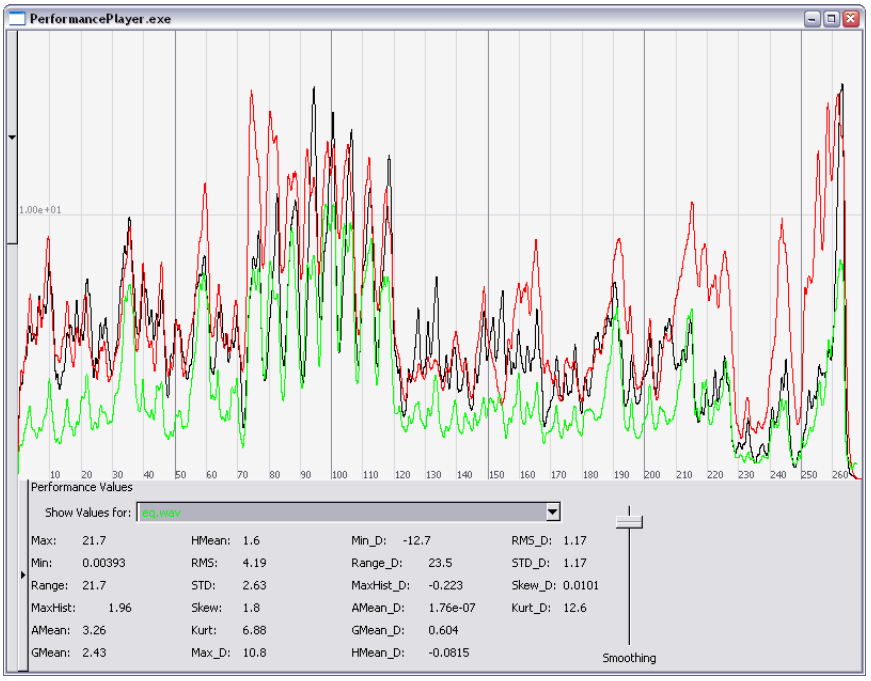

FIGURE 6.5: Performance Player Graphical User Interface with the smoothed Spectral Flux curve of three string quartet performances

Chapter 7

String Quartet Performance Analysis

Analyses of music performances have for several reasons been focused on either piano performances or performances with monophonic instruments. These reasons are mostly of a technical nature, e.g. the availability of the MIDI protocol for keyboard instruments and robust procedures to extract pitch information from single-voiced recordings (compare Sect. 2.2).

Ensemble performance is a rather neglected object of study. Therefore, it has been decided to apply the presented software system to string quartet performances. String quartets take a prominent position in the chamber music genre. Furthermore, a string quartet performance has some characteristics that differentiate it clearly from a keyboard performance. In contrast to solo keyboard performances, ensemble performances require interaction between performers. Also, each string instrument can vary its timbre and loudness even after the note begins. Finally, different playing techniques offer a wide range of tone articulation and tone quality.

Although a thorough study of all these various aspects will not be possible in the following sections, various properties of string quartet performance are investigated in the hope that other studies will follow and contribute to this fascinating field.

7.1 Musical Score

Ludwig van Beethoven's string quartet No. 13 op. 130 in B-flat major is one of his late string quartets, compositions that are regarded as masterpieces of this genre. The five late string quartets have been composed during the years $1822 - 26$, overlapping with the composition of the Missa Solemnis ($1819 - 23$) and the 9^{th} Symphony ($1822 - 24$). The string quartet op. 130 was premiered on March 21^{st}, 1826. Later that year the last movement (the Great Fugue) was replaced by a new finale that was Beethoven's very last completed composition.

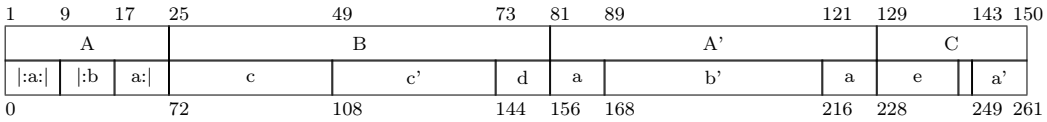

FIGURE 7.1: Structural Analysis of the fourth movement of Beethoven's string quartet No. 13 op. 130 B-Flat major; the numbers above the plot indicate the bars, the numbers below the plot the beats

Two of the work's six movements, the second and the fourth, had to be repeated at their premiere because they were favorably received by the audience. This study focuses on performances of the fourth movement which is labeled *Allegro Assai. Alla danza tedesca*, the latter meaning "German Dance". It is in the key G major, its time signature is $^3/_8$ and it contains 150 bars of which the Bars $1-8$ and $9-24$ are repeated. Its musical character is generally described as "lighthearted" and "charming". As with many dance forms, it has a very clear musical structure as shown in Fig. 7.1. This formal structure is built on phrases with a length of eight bars that usually consist of two four-bar phrases. The only deviation from this underlying eight-bar grid can be found at the end (part C) with the section e being of $12+2$ bars length.

The overall count of onsets (note events) in the piece is 812, and the shortest note values are $^1/_{16}$ notes.

A pulse of $^1/_8$ notes carries throughout the whole piece except in section e, where the momentum seems to stop as the instruments have a quiet conversation, completing each others' phrases.

The piece starts and stops with the following eight-bar main theme (section a, different continuation at the end):

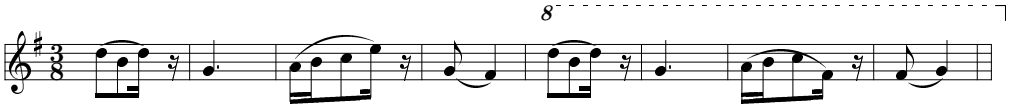

FIGURE 7.2: Main theme of the fourth movement of string quartet No. 13: bars $1-8$

For the remainder of this chapter, it is recommended to the interested reader to have the score of the piece available. Although not mandatory, it will help to follow the structural aspects of the analysis.

ID	*Artist*	*Active*	*Country*	*Rec Year*	*Label & No.*	*Pub Year*
ABQ	Alban Berg Qu.	1971-08	AT	1989	EMI 5736062	1999
AQ	Amadeus Qu.	1947-87	UK	1962	DGG 4631432	1963
BQ	Busch Qu.	1913-52	AT/USA	1941	Sony MPK47687	1991
BSQ	Budapest Streichqu.	1917-67	HU	1926-30	Electrola DB1549	1926-30
EQ	Emerson Qu.	1976-now	USA	1994	DGG 4470752	1997
GQ	Guarneri Qu.	1964-now	USA	1987	Decca 4429402	2007
HSQ	Hollywood String Qu.	1939-61	USA	1957	Testament SBT3082	1996
JQ	Juilliard Qu.	1947-now	USA	1970	Sony S8K87889	2002
KQ	Karl Klingler Streichqu.	1905-40?	DE	1911	Odeon Record Rxx76282	1911
LEQ	Léner String Qu.	1918-30?	HU	1918-20	Columbia L1931	1918-20
LQ	Lindsay Qu.	1965-05	UK	1990?	ASV 602	1991
LSQ	Lasalle String Qu.	1946-88	USA	1972	DGG 4537682	1997
MQ	Melos Qu.	1965-05	DE	1985	DGG 4156762	1986
PQ	Petersen Qu.	1979-now	DE	1999	Capriccio 10851	1999
QI	Quartetto Italiano	1945-85	IT	1969	Philips 4540622	1996
RQ	Rosé Qu.	1882-38	AT	1923-24	Homocord B8216	1923-24
SQ	Smetana Qu.	1945-89	CZ	1982	Denon COCO-79681	2004
TQ	Tokyo String Qu.	1969-now	J	1990-91	RCA RD609753	1992
QV1	Quatuor Vegh	1940-80	HU	1952	Music & Arts CD-10847	2001
QV2	Quatuor Vegh	1940-80	HU	1973	Valois Auvidis V4400	1986
YQ	Yale Qu.	1965?-78?	USA	1971	Brilliant Classics 99127	1998?

TABLE 7.1: Summary of the analyzed string quartet performances

7.2 Recordings

Table 7.1 summarizes the set of analyzed performances. It contains 21 recordings made during the 20$^{\text{th}}$ century between the years 1911 and 1999. The four oldest performances were recorded from shellac discs at the "Deutsches Musikarchiv"[1] (DMA), the remaining performances were copied from CDs.

The Quatuor Vegh ensemble is present with two recordings spaced approximately 20 years apart. The remainder of the recordings has been performed by different string quartet ensembles.

The majority of the ensembles are (or have been) well-known and several of them have received awards for their performances of Beethoven string quartets.

[1] `http://www.d-nb.de/wir/ueber_dnb/dma.htm`

7.3 Procedure

7.3.1 Audio Treatment

The recordings that were copied from CDs remained unprocessed, but the sonic quality of the four historic recordings copied from shellac was very poor with lots of crackles and noise. Since this is a problem for the alignment algorithm, these four recordings were processed several times by de-crackling and de-noising software. This processing has been done at the DMA with the software package *Magix Samplitude*. Since such software not only removes the unwanted audio components but also degrades the quality of the musical signal, a compromise had to be made between the level of crackles and noise and the remaining sonic quality of the recordings. The resulting recordings contained passages where fast notes could not be discriminated by ear and where the sound quality makes it difficult to recognize instruments, e.g. the violin truly as a violin. Therefore, the individual results for these recordings should be regarded with caution. This applies especially to the "Rosé Quartett"-recording (RQ), but to the other three historic recordings (BSQ, KQ and LEQ) as well. These problems are present for nearly all extracted parameters:

- *timing*: due to difficulties to identify specific note onsets,
- *timbre*: due to shellac reproduction quality and processing,
- *loudness*: due to remaining crackles and noise and due to some resonances at specific frequencies,
- *tuning frequency*: due to possible differences between recording and reproduction speed.

7.3.2 Analysis Data

The onset times, the loudness and timbre features have been extracted from the audio files as discussed in Chaps. 3, 4 and 5. The validity of the onset time extraction has been verified by sonification: a MIDI file of the piece has been modified to match the extracted onset times. This file has then been rendered to audio (using a piano sound) and played along with the performance. Where necessary, onset times have been corrected manually.[2]

The extracted onset times $t_o(i)$ have been converted to a series of normalized Inter-Onset-Intervals (IOIs) by calculating the discrete derivative and normal-

[2]The identification of note onsets by ear and eye was not in all cases simple: many notes tend to be "faded-in" by the instrumentalists making it hard to identify discrete onset times, and shorter note values sometimes get masked by their surrounding context.

izing them to their corresponding length in beats $\Delta\tau$, which is the length in quarter note values (see also Eq. (3.2)):

$$IOI(i) = \frac{t_o(i+1) - t_o(i)}{\Delta\tau_{i+1,i}}. \tag{7.1}$$

The resulting series of IOI-observations is of length 811 per performance.

For the analysis of tempo, Inter-Bar-Intervals (IBIs) have been computed by calculating the discrete derivative of the onset times at the downbeats, resulting in 173 observations[3] per performance. The bar level has been selected because it is assumed to be the tactus that would usually be perceived (or tapped along) by listeners. An onset is present at the beginning of every bar so that the resulting IBI series has equidistant sampling intervals. The series has been normalized by the factor 1.5, the distance in beats between two succeeding bar lines.

The loudness and timbre features have been extracted on a linear time scale. To be able to compare them across performances, they have to be mapped to a linear beat scale utilizing the extracted onset times. This mapping, an interpolation that could also be referred to as sample rate conversion, has been done by cubic spline interpolation. The (equidistant) beat resolution has been chosen to be the length of a $1/32$ note, resulting in four observations per feature per $1/8$ note. Overall, this results in 2084 observations per feature per performance when discarding the very last $1/8$ note because of the missing tempo information.

Wherever the feature series had to be compared directly to the series of IOIs, the (unconverted) features have been mapped to the non-equidistant sampling intervals of the onset times. This mapping has been done by cubic spline interpolation as well.

The beat count starts with index 0 (as are practically all indices throughout this thesis), but the bar count starts with index 1 as this is the usual practice in bar numbering. Each beat has the length of two $1/8$ notes, each bar the length of three $1/8$ notes.

For the remainder of this chapter, significant correlations will be marked with an asterisk in tables. The applied threshold for significance is $p \leq 0.01$ if not stated otherwise.

7.3.3 Feature Space Dimensionality Reduction

The number of loudness and timbre features is too large for detailed analysis. Without further knowledge on the psycho-acoustic and musical relevance of the extracted features — a topic that requires further research — any selection seems

[3] 150 bars plus two repetitions á 8 and 16 bars

	$PC1_F$	$PC2_F$	$PC3_F$	$PC4_F$
PC Var (%)	50.59	16.06	9.56	7.01
$v_{Zw(DIN)}$	−0.341	+0.096	+0.035	−0.002
$v_{Zw(1387)}$	−0.335	+0.119	+0.022	−0.012
v_{RMS}	−0.354	−0.032	−0.019	+0.131
v_{1770}	−0.349	−0.034	−0.022	+0.103
v_{dBA}	−0.349	−0.069	−0.037	+0.145
v_{PPM}	−0.353	−0.020	−0.019	+0.125
v_{VU}	−0.344	−0.028	−0.006	+0.134
v_{SF}	−0.270	−0.011	−0.059	−0.172
v_{SR}	−0.017	+0.535	+0.252	−0.147
v_{SC}	−0.029	+0.501	+0.035	+0.039
v_{SS}	−0.018	+0.322	+0.339	−0.436
v_{MFC0}	−0.281	+0.027	+0.007	−0.421
v_{MFC1}	−0.071	−0.472	+0.004	−0.533
v_{MFC2}	−0.034	−0.291	+0.626	−0.148
v_{MFC3}	+0.004	−0.137	+0.648	+0.438

TABLE 7.2: Variance (first row) and loadings (second to last row) of the first four principal components of the feature set

to be arbitrary. To reduce the number of feature dimensions without directly discarding features, the complete feature set (consisting of seven loudness features and eight timbre features for all files of the test set) has been subjected to Principal Component Analysis (PCA, compare Chap. A.2). PCA transforms the feature data to a new orthogonal coordinate system and orders the resulting components according to their amount of variance. This allows selection of the components that contribute most to the variance of the feature set (low order components) and to discard higher order components.

Each feature is an input variable of the PCA and the corresponding feature results extracted from all performances are the observations of this variable. The number of remaining principal components has been chosen as the number of components with an eigenvalue higher than 1. The more similar different input variables or features are, the smaller will be the number of the resulting components.

Prior to this operation, each feature has been normalized in order to have a mean of zero and a standard deviation of 1. Table 7.2 shows the relative amount of variance of the selected first four principal components as well as the PCA loadings per component and feature.

Together, these four components explain more than 80% of the variance of the feature set as shown in the first row of Table 7.2. Each column represents the PCA loadings for one component. The first component $PC1_F$ is obviously related to loudness. All loudness features — the Zwicker loudness $v_{Zw(DIN)}$, $v_{Zw(1387)}$, the RMS based measurements v_{RMS}, v_{1770}, v_{dBA} and the studio monitoring device measurments v_{PPM}, v_{VU} — contribute nearly equally to this component, as does the first Mel Frequency Cepstral Coefficient (v_{MFC0}) which can be argued to be loudness related as well. The other features have only low loadings except for the Spectral Flux v_{SF} that also shows a higher loading. The three remaining components are clearly timbre dimensions. The

	$r(IOI, PC1_F)$	$r(IOI, PC2_F)$	$r(IOI, PC3_F)$	$r(IOI, PC4_F)$
ABQ	+0.14*	+0.02	−0.03	−0.05
AQ	+0.09*	−0.04	+0.02	−0.10*
BQ	+0.05	−0.08	−0.06	−0.05
BSQ	+0.07	−0.09*	−0.03	−0.04
EQ	+0.12*	+0.01	+0.03	−0.10*
GQ	+0.08	−0.00	+0.04	−0.14*
HSQ	−0.01	−0.05	+0.07	−0.06
JQ	+0.05	−0.07	+0.07	−0.08*
KQ	+0.03	−0.05	+0.02	−0.06
LEQ	−0.05	−0.06	−0.05	+0.04
LQ	+0.04	−0.03	−0.05	−0.02
LSQ	+0.09*	−0.07	−0.01	−0.05
MQ	−0.01	−0.01	−0.04	−0.01
PQ	+0.02	−0.03	+0.07	−0.08*
QI	+0.04	+0.05	+0.01	−0.06
QV1	+0.06	+0.02	+0.06	−0.11*
QV2	+0.09*	−0.05	+0.04	−0.05
RQ	−0.09*	−0.03	+0.04	−0.03
SQ	−0.01	−0.07	−0.00	−0.02
TQ	+0.06	−0.01	+0.04	−0.10*
YQ	+0.06	+0.01	+0.03	−0.09*
Overall	+0.04*	−0.03*	+0.01	−0.06*

TABLE 7.3: Correlation between the series of IOIs and the four components $PC1_F - PC4_F$

features contributing to the second component are mainly the Spectral Rolloff v_{SR} and the Spectral Centroid v_{SC}, although the MFCCs contribute as well. This component can probably be interpreted as brightness or sharpness. The third and fourth components show a mixture of different MFCCs that cannot easily be interpreted psycho-acoustically but as a measure of shape of the spectral envelope. The Spectral Spread contributes to all three timbre components equally high.

In the following, the selected four principal components $PC1_F - PC4_F$ will be referred to as features.

The features $PC1_F - PC4_F$ are — thanks to the properties of the PCA — not correlated, but in order to check for a relationship between them and the tempo (or more specifically the series of IOIs), a correlation is computed for each performance and over the whole data set, respectively. The more the tempo variation is related to the variation of other features, the higher will be the absolute correlation result. Table 7.3 displays the results.

Several of the resulting correlations are significant, but their values are so small that it is probably safe to conclude that there are practically no loudness and timbre features to tempo interrelations, meaning that these are independent performance dimensions. The few significant results hint at the possibility of the following tendencies: loudness and brightness tend to increase with tempo increases or to decrease with tempo decreases[4], and there is a tendency of $PC4_F$ to decrease with increasing tempo or vice versa.

[4]Note that the series of IOI is an *inverse* tempo representation and the loadings of $PC1_F$ are *negative*.

7.4 Overall Performance Profiles

In order to investigate common attributes, the performances were subjected to PCA for each individual feature (IOI, IBI and $PC1_F - PC4_F$). In this case, the performances are the variables, and the extracted feature values per file are the observations. If there are high correlations between IOI, IBI or $PC1_F - PC4_F$ components among the performances, there will only be few principal components (to be called *Profiles*) that will account for most of the variance. If many performances have high loadings on a profile, this profile can be interpreted as a prototypical performance. Otherwise, that is if all performances are highly individual, all components will approximately account for the same amount of variance.

The resulting PCA components will be referred to as $UPCn(Feature)$. PCAs were computed for three different excerpts of the piece: all bars, Bars $1-128$ (excluding the final part C) and Bars $1-24$ including the two repetitions. The following sections provide a more detailed analysis per feature. The detailed overview on the selected principal components and their relative amount of variance can be found in the Appendix in Tables C.1, C.2 and C.3.

As the feature input had to be normalized to a mean of 0 and a standard deviation of 1, the following analysis concerns only the relative changes of the extracted features. It cannot be interpreted in terms of absolute quantities.

7.4.1 Tempo

For the tempo analysis, the Inter-Bar-Intervals IBI have been investigated. High IBI values correspond to slow tempi in relation to the mean tempo, while low values refer to tempi faster than the mean tempo.

Figure 7.3 depicts the first component of the overall tempo profile for the whole movement.

It is the only selected main component accounting for 87% of the variance to which all performances are correlated significantly ($r > 0.7$, $p < 0.01$, see

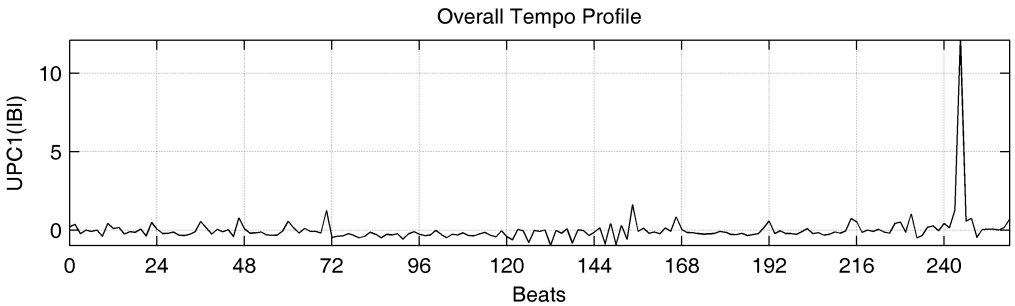

FIGURE 7.3: Grand Overall Tempo Profile $UPC1(IBI)$: the main component of the overall tempo profile, computed over the whole piece

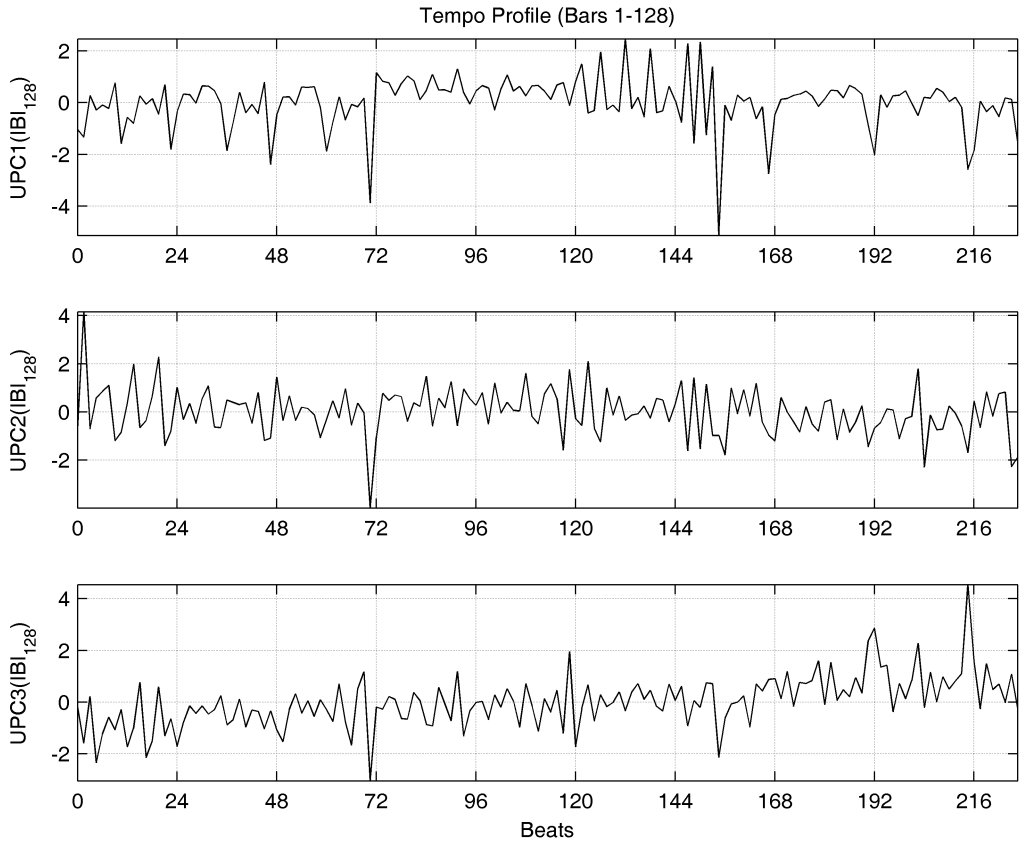

FIGURE 7.4: Tempo Components $UPC1-3(IBI_{128})$: the main tempo profile components for the first 128 bars

Table C.4). The conclusion that all performances use a very similar underlying tempo strategy can, however, be misleading since the fermata that ends section e (Beat 245.5) dominates the profile. Therefore, we focus on the first 128 bars (parts A, B and A', Beats $0-228$) for the tempo analysis. Here, three main components can be identified as depicted in Fig. 7.4. These three components account for 56%, 6% and 5% of the variance, respectively.

The first component $UPC1(IBI_{128})$ is significantly correlated to all performances with correlations $-0.88 \leq r \leq -0.37$ (compare Table C.5). Because of the negative correlation, low values correspond to low tempi and vice versa for this first component. The component shows the general tendency to play part B (Bars $25-80$, Beats $72-156$) faster than the surrounding parts. The following part A' (Bars $81-128$, Beats $156-228$) is played in a relatively constant tempo. Individual parts and sections are clearly separated by *ritardandi*, in many cases preceded by small *accelerandi* (see begin and end of part B but also the local minima in part A in a distance of eight bars (12 beats) as well as the local minima in part A' that clearly separate section b' from the surrounding sections a).

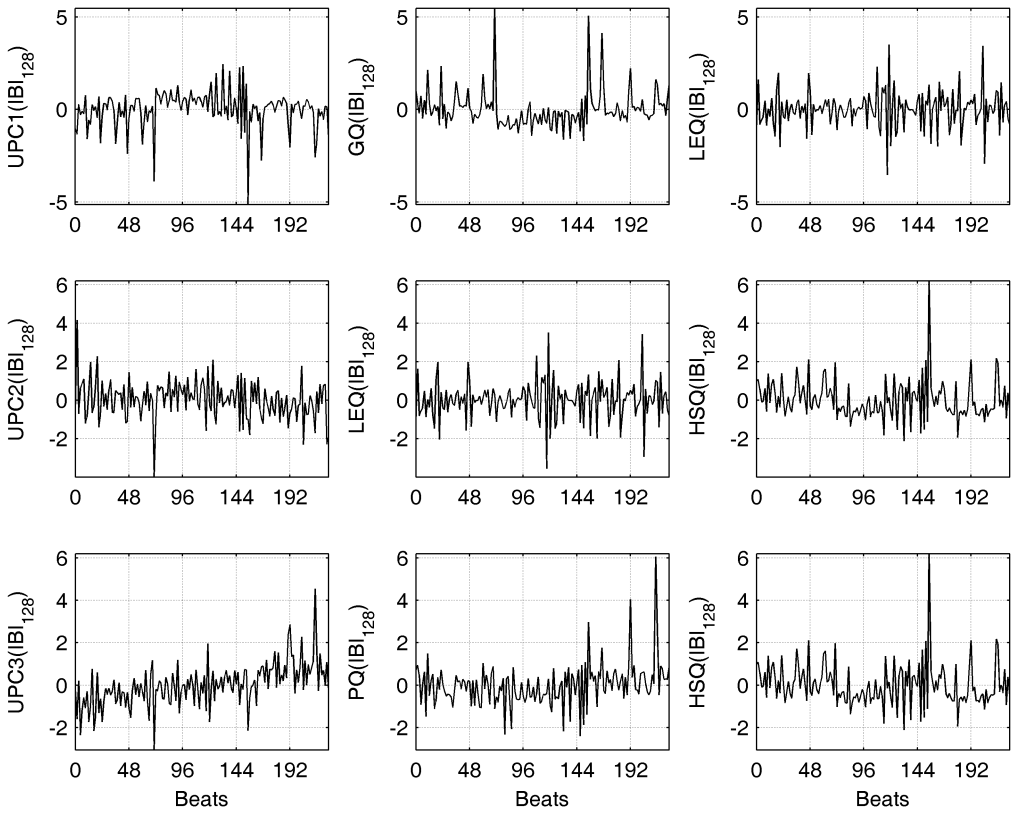

FIGURE 7.5: Tempo components $UPC1-3(IBI_{128})$ (column 1) and the individual tempo profiles with the highest correlation to them (column 2) and the lowest correlation to them (column 3)

The second component $UPC2(IBI_{128})$ is significantly correlated to only approximately half the performances with correlations between $-0.3 \leq r \leq 0.4$ (compare Table C.5). The change in the sign may indicate that some performances follow contradictive tempo strategies.

The third component $UPC3(IBI_{128})$ is significantly correlated to eight performances of the test set with correlations $-0.44 \leq r \leq 0.5$, four of them correlated positively. The component shows the tendency to change the tempo slowly over time. Those performances that correlate negatively with the component tend to increase the tempo, while the others tend to end at a slower tempo.

Figure 7.5 shows the discussed three main components together with the performances that show the highest and lowest correlation to them.

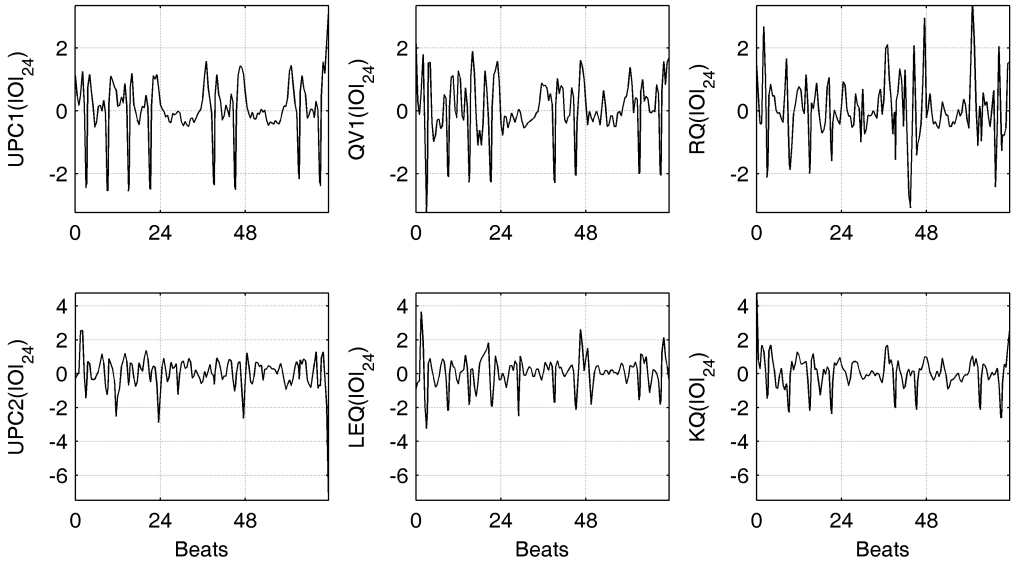

FIGURE 7.6: Timing components $UPC1-2(IOI_{24})$ (column 1) and the individual timing profiles with the highest correlation to them (column 2) and the lowest correlation to them (column 3)

7.4.2 Timing

A more detailed analysis of timing or microtempo has been approached by investigating the normalized IOIs for the first 24 bars including both repetitions. Figure 7.6 displays the two selected components and the normalized timing profiles of the performances that correlate highest and lowest to the components.

The first component accounts for approximately 60% of the variance, the second component for about 6%. All performances are significantly correlated to the first component with correlations $0.5 \leq r \leq 0.9$ (compare Table C.6). This means that for this component, higher values correspond to lower tempi. Only 11 performances are significantly correlated with the second component, some of them negatively and some of them positively.

The first component $UPC1(IOI_{24})$ shows a prominent *ritardando* at the end. This was to be expected according to the tempo profile analysis, as was the slightly lower tempo at the very beginning. Very prominent in the first timing component are the eight local minima (meaning a sudden tempo increase) which appear at all places where two $1/16$ notes appear: at Bars 3 and 7 (Beats 3, 9, 15 and 21) and at Bars 19 and 23 (Beats 39, 45, 63 and 69), meaning that all $1/16$ notes are played faster than the average tempo. Section b (Bars $9-16$ or Beats $24-36/48-60$) shows a timing profile without sudden changes; there seems to be a tendency to play with a higher tempo at Bars $13-15$ (Beats $30-36/54-60$) before the *ritardando* that announces the main theme reappearing in Bar 16.

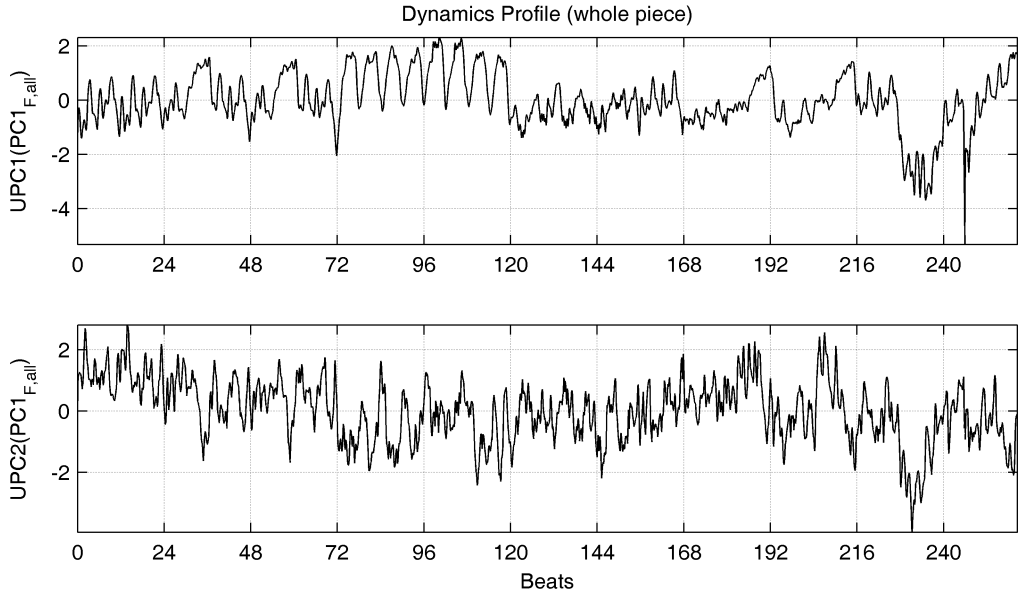

FIGURE 7.7: Loudness Components $UPC1-2(PC1_{F,all})$: the main loudness profile components computed over the whole piece

The second component $UPC2(IOI_{24})$ is hard to interpret; prominent are four local minima appearing at the last eighth note of each phrase (Beats 11.5, 23.5, 47.5 and 71.5).

The first timing component accounts for 60% of the variance and all investigated performances seem to follow the same basic timing strategy. This might be due to the character of the piece of music that possibly offers no alternative approaches to timing strategies.

7.4.3 Loudness

Applying a PCA to the (inverse) loudness feature $PC1_F$, two main components can be identified accounting for approximately 76% and 5% of the overall variance. They are depicted in Fig. 7.7. The first component seems to reflect the general loudness strategy very well as the correlation to all performances is significant and its magnitude is higher than 0.7 (compare Table C.7), except for one historic recording (RQ). Although the correlations are negative, increasing profile values correspond to increasing loudness values because $PC1_F$ had negative loadings for all initial loudness features.

The first component $UPC1(PC1_{F,all})$ mirrors the musical structure at least as clearly as did the main tempo profile. It begins with the repeated Bars $1-8$ (section a, Beats $0-12/12-24$), followed by the repetition of Bars $9-24$ (sections b and a, Beats $24-48/48-72$). Section c in Bars $25-56$ (Beats $72-120$) can be easily identified by the periodic *crescendo*/*decrescendo* pattern. The following Bars $57-80$ (Beats $120-156$) are marked as ***p*** in

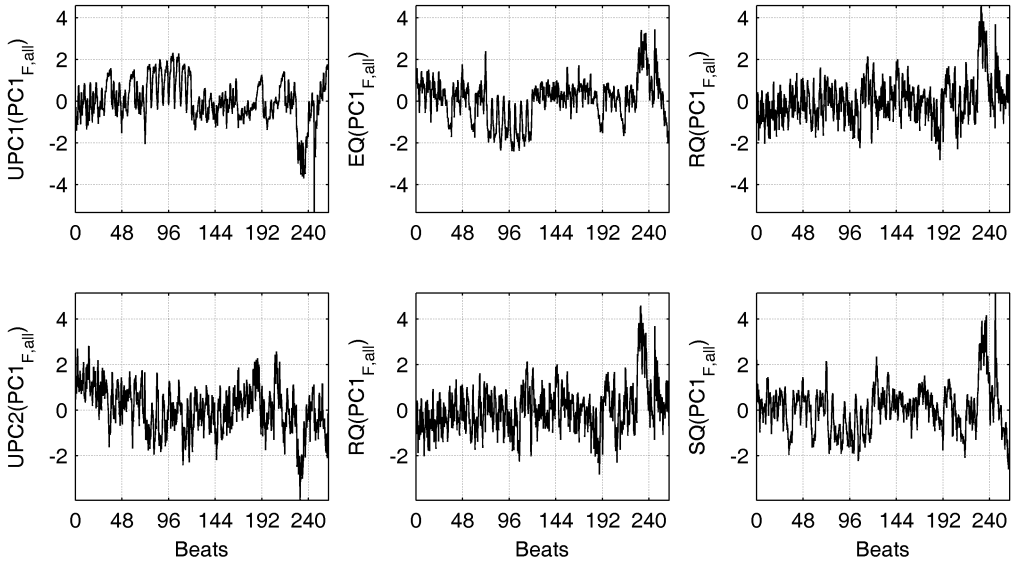

FIGURE 7.8: Loudness components $UPC1-2(PC1_{F,all})$ (column 1) and the loudness profiles with the highest correlation to them (column 2) and the lowest correlation to them (column 3); note that the $PC1_F$-profiles show the inverse loudness due to the negative loadings for this component

the score, reflected by the lower volume in $UPC1(PC1_{F,all})$. The single-voiced section *e* starting from Bar 129 (Beat 246) can also be easily identified as can the following *crescendo* toward the end.

Every part seems to have a very characteristic shape that can be distinguished from others. For example, the eight bars of section *a* (Beats $0-12$) that are repeated in the beginning show a characteristic pattern of four peaks (every second bar) that appears in a similar fashion every time when this main theme is played: Bars $17-24$ (Beats $36-48/60-72$), Bars $81-88$ (Beats $156-168$), Bars $121-128$ (Beats $216-228$), and less obvious the finishing eight bars (12 beats). Section *b* is dominated by the *crescendo* from Bars $13-16$ before the main theme reappears and section *c* periodically increases and decreases the volume with a periodicity of four bars. It can be observed that section *b'* (Bars $89-120$, Beats $168-216$) consists of two similar parts with increasing volume. The absolute minimum of the loudness component can be found before Beat 246 where a tacit with fermata can be found in the score.

In contrast to the first component, the second component $UPC2(PC1_{F,all})$ is not clearly interpretable. Figure 7.8 shows the most similar and dissimilar performances with respect to the loudness profile.

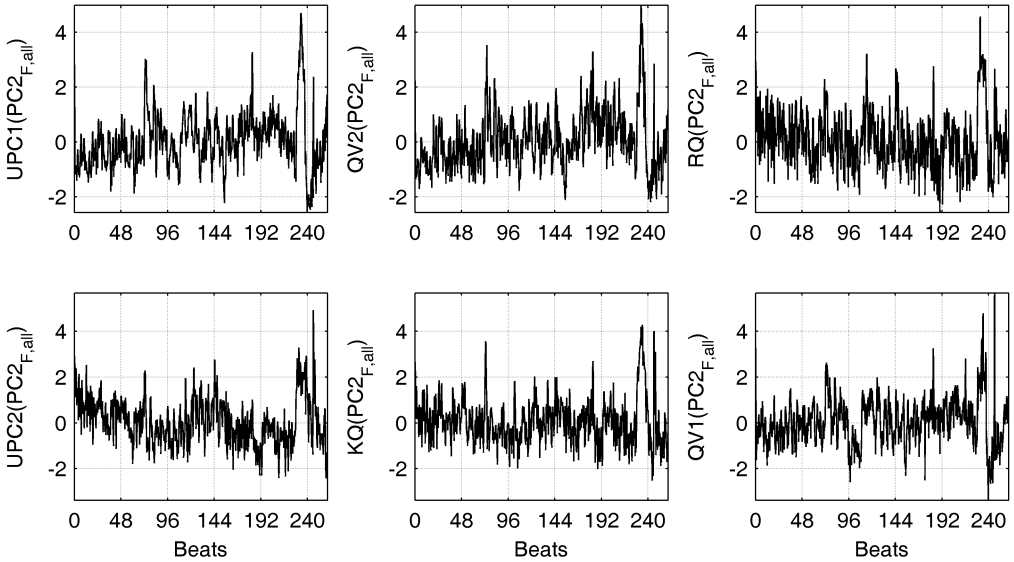

FIGURE 7.9: Brightness components $UPC1-2(PC2_{F,all})$ (column 1) and the brightness profiles with the highest correlation to them (column 2) and the lowest correlation to them (column 3)

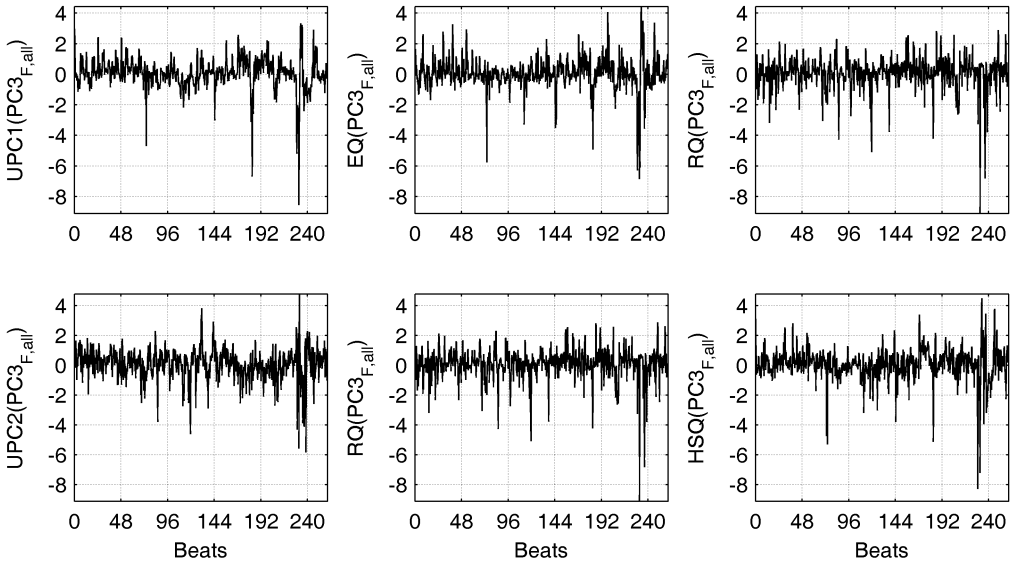

FIGURE 7.10: $PC3_F$ components $UPC1-2(PC3_{F,all})$ (column 1) and the profiles with the highest correlation to them (column 2) and the lowest correlation to them (column 3)

7.4.4 Timbre

Figures 7.9, 7.10 and 7.11 show the main components for the timbre-related features. Although some structural properties can be identified with the knowledge that was gained from interpreting the loudness and tempo profiles,

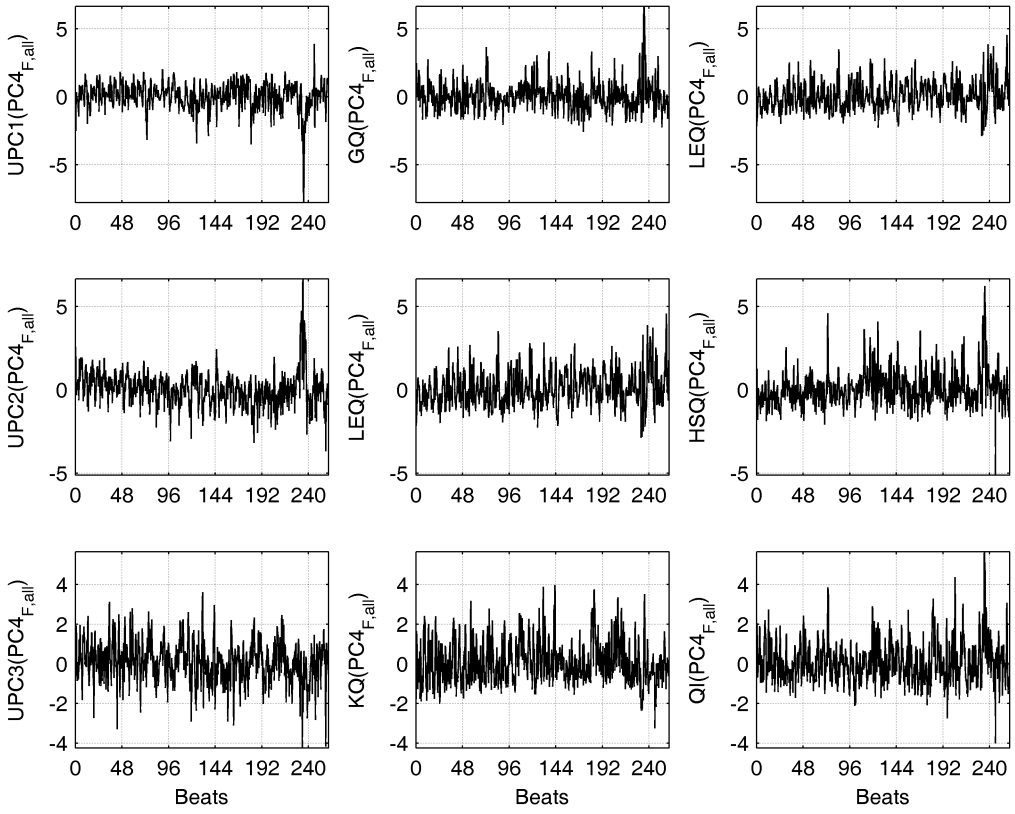

FIGURE 7.11: $PC4_F$ components $UPC1-3(PC4_{F,all})$ (column 1) and the profiles with the highest correlation to them (column 2) and the lowest correlation to them (column 3)

it is very difficult to extract general characteristics or to find explanations for the specific form of the profile such as the relation to musical structure.

The PCA tended to separate those recordings copied from shellac from the others: for all three $UPC1$ timbre profiles, the four historic recordings (BSQ, LEQ, KQ, RQ) have the lowest correlation, but are highly correlated to the corresponding $UPC2$ profiles. The audio quality may not only impact the overall (or average) timbre quality but its variation over time as well.

7.5 Performance Similarity

7.5.1 Repetition Similarity

For the two repeated passages (Repetition 1: Bars $1-8$, Repetition 2: Bars $9-24$), the similarity is investigated by calculating the correlation between the first and the second rendition for each feature. The results are displayed

	Rep. 1					***Rep. 2***				
	Bars $1-8$					***Bars*** $9-24$				
	IOI	$PC1_F$	$PC2_F$	$PC3_F$	$PC4_F$	IOI	$PC1_F$	$PC2_F$	$PC3_F$	$PC4_F$
ABQ	+0.76*	+0.83*	+0.92*	+0.72*	+0.51*	+0.68*	+0.91*	+0.83*	+0.80*	+0.73*
AQ	+0.71*	+0.94*	+0.13	+0.48*	+0.56*	+0.90*	+0.91*	+0.84*	+0.81*	+0.83*
BQ	+0.75*	+0.78*	+0.54*	+0.82*	+0.83*	+0.92*	+0.90*	+0.66*	+0.61*	+0.75*
BSQ	+0.58*	+0.66*	+0.67*	+0.73*	+0.38*	+0.72*	+0.91*	+0.80*	+0.66*	+0.63*
EQ	+0.70*	+0.92*	+0.70*	+0.68*	+0.71*	+0.87*	+0.97*	+0.81*	+0.72*	+0.72*
GQ	+0.84*	+0.93*	+0.57*	+0.84*	+0.82*	+0.70*	+0.93*	+0.87*	+0.72*	+0.79*
HSQ	+0.88*	+0.94*	+0.56*	+0.72*	+0.69*	+0.93*	+0.95*	+0.91*	+0.38*	+0.77*
JQ	+0.84*	+0.76*	+0.65*	+0.63*	+0.49*	+0.70*	+0.95*	+0.79*	+0.75*	+0.76*
KQ	+0.80*	+0.84*	+0.72*	+0.84*	+0.77*	+0.86*	+0.78*	+0.72*	+0.33*	+0.53*
LEQ	+0.53*	+0.39*	+0.47*	+0.67*	+0.60*	+0.56*	+0.59*	+0.39*	+0.54*	+0.46*
LQ	+0.73*	+0.82*	+0.40*	+0.60*	+0.57*	+0.67*	+0.90*	+0.60*	+0.62*	+0.74*
LSQ	+0.84*	+0.95*	+0.39*	+0.80*	+0.85*	+0.85*	+0.96*	+0.85*	+0.82*	+0.78*
MQ	+0.42	+0.94*	+0.71*	+0.51*	+0.57*	+0.67*	+0.92*	+0.75*	+0.83*	+0.85*
PQ	+0.44	+0.91*	+0.15	+0.73*	+0.65*	+0.57*	+0.92*	+0.80*	+0.70*	+0.72*
QI	+0.84*	+0.86*	+0.20	+0.77*	+0.67*	+0.54*	+0.91*	+0.76*	+0.78*	+0.72*
QV1	+0.81*	+0.93*	+0.35*	+0.69*	+0.69*	+0.92*	+0.84*	+0.65*	+0.75*	+0.64*
QV2	+0.62*	+0.86*	+0.46*	+0.66*	+0.65*	+0.81*	+0.96*	+0.74*	+0.41*	+0.63*
RQ	+0.82*	+0.73*	+0.68*	+0.69*	+0.77*	+0.69*	+0.66*	+0.74*	+0.77*	+0.74*
SQ	+0.80*	+0.66*	+0.30*	+0.82*	+0.52*	+0.60*	+0.93*	+0.84*	+0.77*	+0.58*
TQ	+0.70*	+0.92*	+0.41*	+0.70*	+0.68*	+0.66*	+0.92*	+0.83*	+0.69*	+0.74*
YQ	+0.73*	+0.86*	+0.59*	+0.65*	+0.69*	+0.75*	+0.92*	+0.79*	+0.44*	+0.70*

TABLE 7.4: Correlation results between repeated parts (Repetition 1: Bars $1-8$, Repetition 2: Bars $9-24$) per feature per performance performance

in Table 7.4. Because of the relatively short series of observations to compare, these results should only be interpreted as tendencies; some of the extracted features may show more detection inaccuracies in this passage than in others. This could possibly impact the results as these inaccuracies most likely will not be averaged out in such a short series of observations. This problem may be indicated by the generally higher correlation results for the second repetition that is twice as long as repetition 1.

The HSQ performance shows the highest similarity between the two renditions for the IOI series for both repeated passages with correlations of $r = 0.88$ and $r = 0.93$, respectively. The lowest IOI correlations can be observed for MQ (repetition 1, $r = 0.42$) and QI (repetition 2, $r = 0.54$); PQ shows comparably low correlations for both the first and the second repetition ($r = 0.44, r = 0.57$).

The loudness variation is very similar between the repetitions as conveyed by the results of the correlation of $PC1_F$. Most of the results are significant and in the range of $r = 0.9$, with the main exception of some historic recordings that show lower correlation results: LEQ with values of $r = 0.4$ and $r = 0.6$ and for the second repetition RQ and KQ with correlations of $r = 0.66$ and $r = 0.78$, respectively.

The timbre dimensions $PC2-4_F$ results show larger variations of the correlation results, for example in the range of $0.1 \leq r \leq 0.9$ for $PC2_F$ in repetition 1.

Figures 7.12, 7.13, 7.14 and 7.15 show examples of similar and dissimilar performances for both repetitions and the features IOI and $PC1_F$.

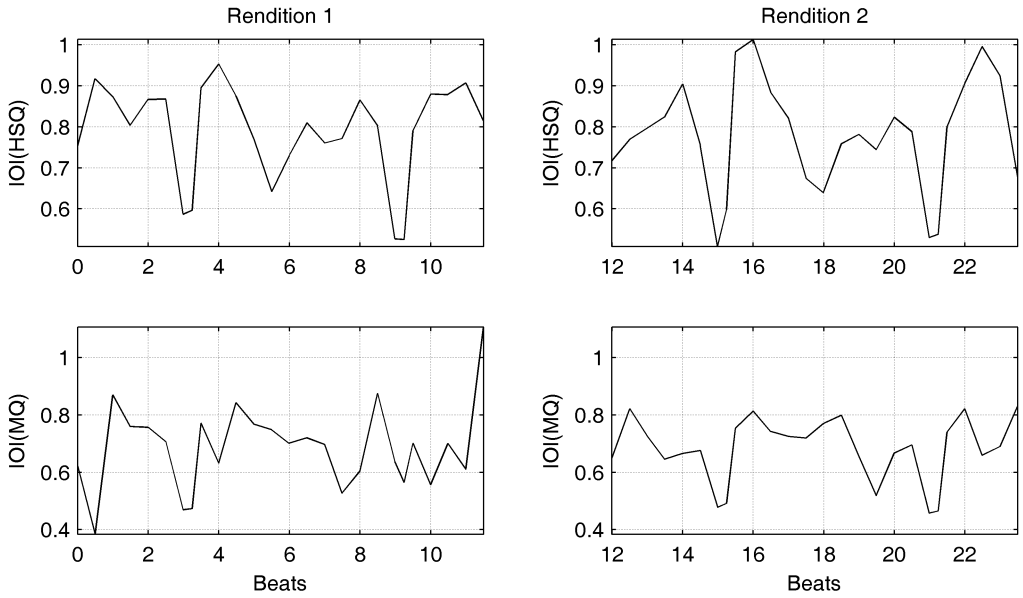

FIGURE 7.12: Repetition 1 in the timing (IOI) domain: Examples for a very similar (above) and dissimilar (below) repetition

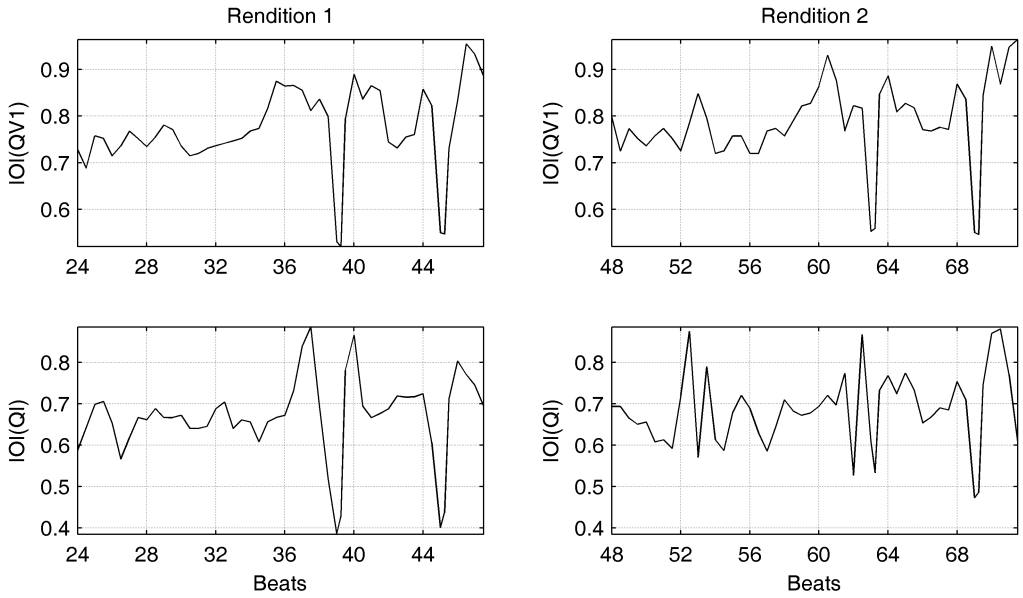

FIGURE 7.13: Repetition 2 in the timing (IOI) domain: Examples for a very similar (above) and dissimilar (below) repetition

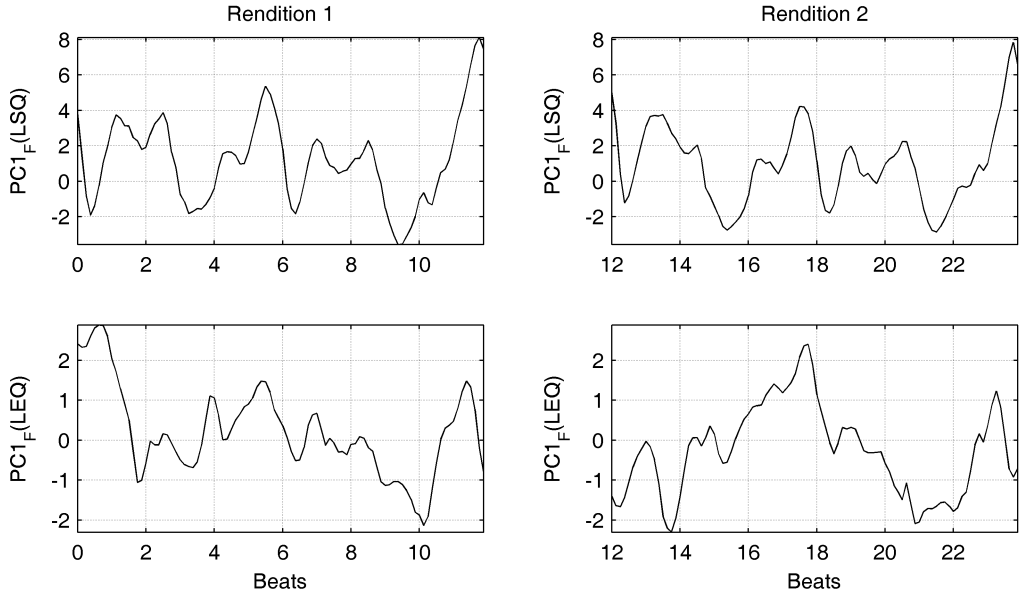

FIGURE 7.14: Repetition 1 in the Loudness ($PC1_F$) domain: Examples for a very similar (above) and dissimilar (below) repetition

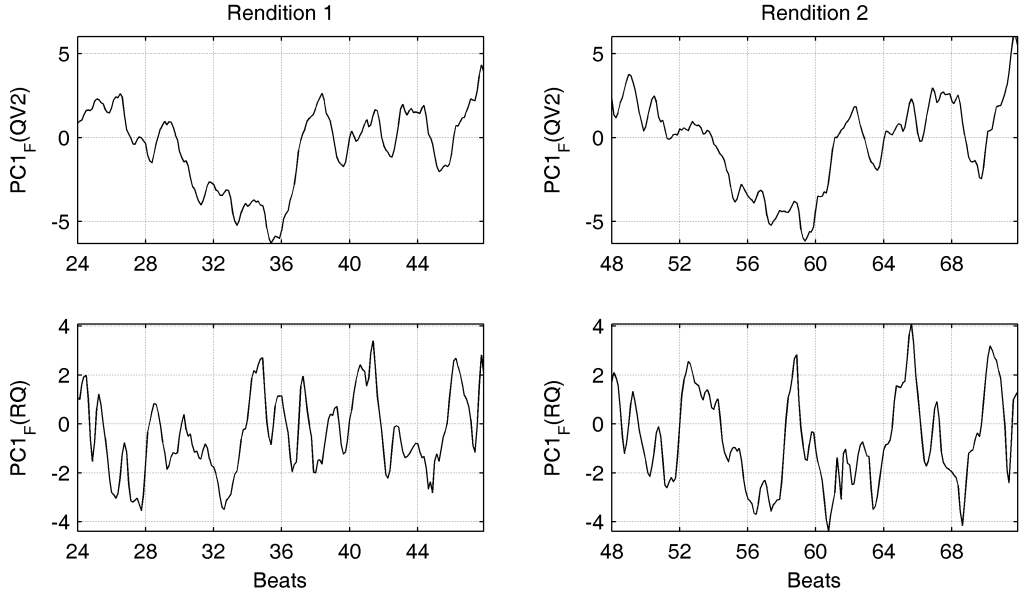

FIGURE 7.15: Repetition 2 in the Loudness ($PC1_F$) domain: Examples for a very similar (above) and dissimilar (below) repetition

7.5.2 Overall Similarity

Similar to the computation of the repetition similarity, the correlation between all performances is calculated per feature for the whole file (only Bars $1-128$ for IBI and IOI features). The detailed results can be inspected in tables C.14, C.15, C.16, C.17, C.18 and C.19.

As expected, the same general tendencies can be identified as could be for the correlation between the UPC-components and the individual performances. The loudness profiles show high similarity to each other, the tempo and timing profiles are relatively highly correlated as well, and the correlation seems to decrease with features $PC2-4_F$.

The two recordings performed by the same ensemble $QV1$ and $QV2$ show high correlation results for nearly all features, but more similar pairs of performances can be found within the other performances.

7.6 Overall Observations

For the analysis and visualization of absolute results, the components $PC1-4_F$ are meaningless due to their normalization. Hence, a set of four features has been selected to represent the feature set: the BS.1770 loudness v_{1770}, the Spectral Rolloff v_{SR}, the Spectral Centroid v_{SC}, and the MFCC3 v_{MFC3}. These features have been selected by trying to find a reasonable compromise between the following conditions:

- good representation of one of the first three components in Table 7.2,
- comparably high absolute loadings in Table 7.2 and
- good psycho-acoustic relevance and interpretability.

Table 7.5 summarizes the overall results for each performance. Overlined values stand for arithmetic averages, σ for the standard deviation of a feature and the indices $p25, p75$ for the boundaries of the lower and upper quartiles which separate the lowest and highest 25% of the data from the middle 50% around the median. The tempo computation has been done in the IBI domain but has been converted to BPM values where appropriate to allow a more intuitive understanding. Both the Spectral Centroid and the Spectral Rolloff have been converted to the unit kHz for easier interpretation.

The mean tempo $\overline{BPM}$, representing the first row of the table, is consistently lower than the maximum of the tempo histogram (second row). This verifies other observations (compare [Rep98], [GD01]) indicating that deviations from the most frequently used tempo to slower tempi occur more often or are more prominent than deviations to higher tempi. For example, the fermata at

section e will have large impact on the arithmetic mean value of the tempo. The maximum of the tempo histogram BPM_{Hist} (which will be referred to as overall tempo) varies over performances from below 80 BPM (QV1) up to more than 110 BPM (BQ). Most performances have tempi between 85 BPM and 95 BPM. The quartile boundaries BPM_{p25}, BPM_{p75} show the tendency to vary the tempo more toward lower tempi than toward higher tempi. For some performances, the upper boundary is *below* the overall tempo, indicating that the histogram maximum is not always at the same position as the median. Repp proposed a measure of relative tempo modulation depth where the standard deviation is divided by the mean IOI, as he observed a dependency of tempo and standard deviations. More specifically, he noted that slow performances tended to vary the tempo more than fast performances [Rep98]. This relative modulation depth can be found in the table as $\sigma(IBI)/\overline{IBI}$. The results vary between 0.13 (RQ) and 0.29 (GQ, QI) and are difficult to interpret.

The tuning frequencies f_{A4} vary between 436 Hz (AQ, LEQ, RQ) and 452 Hz (KQ), corresponding to deviations of approximately −16 cent and +46 cent from 440 Hz. The large deviations occurring in the historic recordings may originate from a difference between the recording and reproduction speed.

The average volume level $\overline{v_{1770}}$ of the recordings is — ignoring the four historic recordings (BSQ, KQ, LEQ, RQ) that have been leveled manually — in a range of $-35\,\mathrm{dB_{FS}}$ (EQ) to $-26\,\mathrm{dB_{FS}}$ (HSQ) with standard deviations $\sigma(v_{1770})$ around 4 dB to 5 dB (EQ).

The average bandwidth (roll-off) v_{SR} and the brightness (centroid) v_{SC} vary between 2.1 kHz/0.3 kHz (LEQ) and 4.1 kHz/0.9 kHz (BQ). A quick qualitative verification by ear can confirm this fact; although both recordings are very old, the LEQ recording suffers from both the shellac sound quality and the de-noising/de-crackling processing resulting in a rather muffled sound. The BQ recording has been re-mastered for CD publication and sounds bright and sharp. The modulation depth of the bandwidth $\sigma(v_{SR})$ lies between 0.5 kHz (ABQ) and 1 kHz (HSQ) and the brightness modulation $\sigma(v_{SC})$ between 0.1 kHz (KQ) and 0.3 kHz (ABQ).

	$\overline{BPM}$	BPM_{Hist}	BPM_{p25}	BPM_{p75}	$\sigma(IBI)/\overline{IBI}$	f_{A4}	$\overline{v_{1770}}$	$\sigma(v_{1770})$	$\overline{v_{SR}}$	$\sigma(v_{SR})$	$\overline{v_{SC}}$	$\sigma(v_{SC})$	$\overline{v_{MFC3}}$	$\sigma(v_{MFC3})$
ABQ	92.23	92.72	89.06	99.28	0.17	443	-29.72	4.45	3.22	0.50	0.73	0.31	2.25	1.43
AQ	100.26	105.28	98.25	105.56	0.19	436	-29.84	4.73	3.18	0.76	0.50	0.16	1.48	1.56
BQ	111.48	113.77	107.49	118.76	0.16	447	-27.89	3.65	4.09	0.62	0.85	0.29	1.47	1.21
BSQ	78.11	82.48	77.01	83.43	0.22	450	-30.25	4.11	3.00	0.63	0.37	0.16	2.00	1.17
EQ	76.82	78.73	74.66	85.47	0.26	442	-34.96	5.24	2.78	0.51	0.55	0.19	1.60	1.28
GQ	72.63	79.22	71.31	80.43	0.29	439	-28.24	4.60	2.92	0.69	0.43	0.22	2.19	1.35
HSQ	76.03	80.90	73.81	82.30	0.24	444	-25.64	4.38	2.61	1.01	0.40	0.16	1.67	1.22
JQ	88.00	95.75	86.66	96.19	0.26	443	-28.64	3.98	2.81	0.57	0.53	0.21	2.04	1.30
KQ	94.77	100.03	91.92	103.26	0.26	452	-26.46	3.51	3.54	0.73	0.49	0.11	1.66	1.22
LEQ	97.78	100.56	93.97	105.43	0.24	436	-32.60	6.28	2.08	0.82	0.29	0.10	1.86	1.12
LQ	78.96	88.01	77.10	86.64	0.24	440	-28.77	4.05	2.83	0.61	0.56	0.17	2.13	1.34
LSQ	78.39	85.83	76.86	85.44	0.21	440	-28.10	4.32	2.77	0.66	0.54	0.17	1.51	1.46
MQ	86.42	89.25	84.49	93.01	0.28	446	-28.46	4.32	3.06	0.75	0.52	0.20	1.70	1.29
PQ	95.88	95.04	93.07	102.89	0.21	443	-31.80	4.70	3.13	0.72	0.54	0.23	1.89	1.45
QI	84.76	92.72	83.97	91.71	0.29	444	-28.48	4.06	2.89	0.68	0.56	0.18	1.67	1.33
QV1	76.57	77.14	74.57	82.06	0.24	441	-28.60	4.18	3.56	0.91	0.61	0.26	1.85	1.48
QV2	79.16	84.68	77.50	87.48	0.28	447	-32.12	5.17	3.08	1.00	0.43	0.21	2.08	1.15
RQ	80.71	86.22	76.77	86.69	0.13	436	-35.86	4.80	2.82	0.85	0.38	0.12	1.79	1.19
SQ	80.96	84.87	79.14	86.15	0.17	444	-30.97	4.01	3.07	0.72	0.45	0.20	2.07	1.21
TQ	81.61	87.20	78.48	87.93	0.23	443	-32.87	4.64	3.40	0.95	0.51	0.19	1.55	1.34
YQ	90.41	95.75	87.83	96.24	0.18	440	-27.23	4.69	2.83	0.56	0.48	0.18	2.05	1.63

TABLE 7.5: Overall Results for all Performances; 4 classes from left to right: tempo observations, tuning frequency, loudness observations, timbre observations

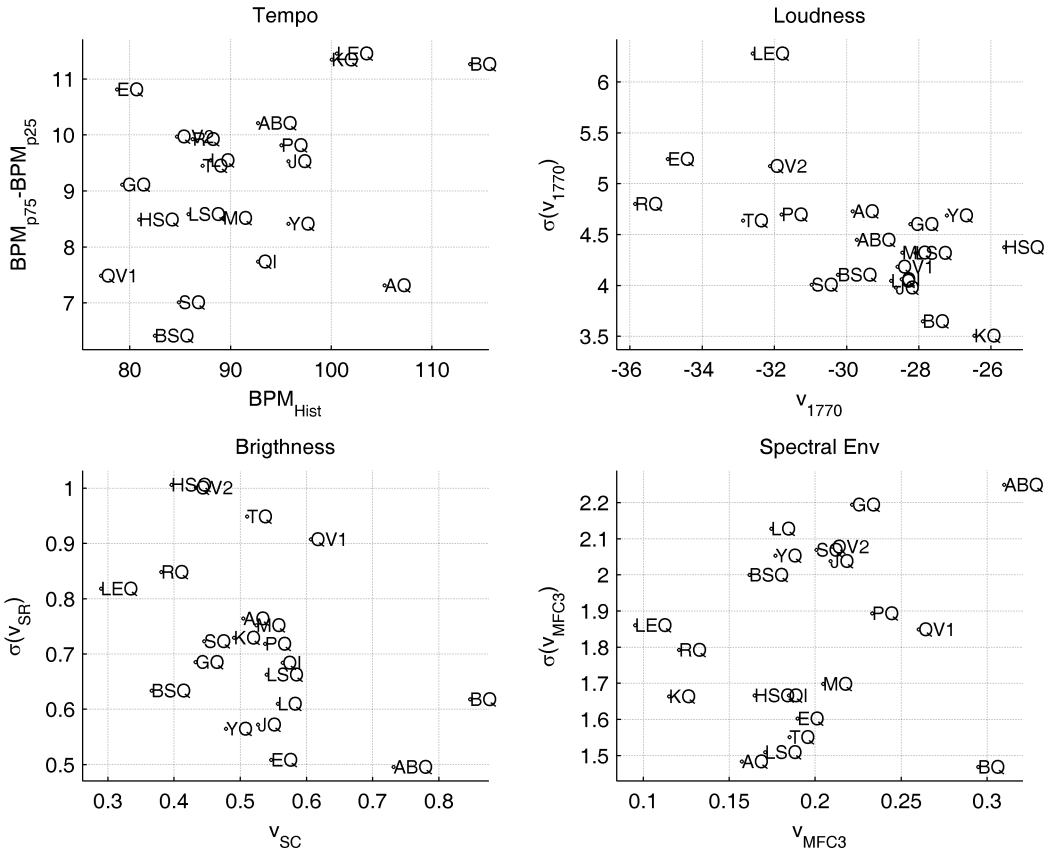

FIGURE 7.16: Location of every analyzed performance in the tempo, loudness, brightness and "spectral envelope" space (from upper left to lower right); for each component, the abscissa represents the overall value and the ordinate the modulation depth

Figure 7.16 visualizes the locations of the performances in scatter plots of selected features from the tempo, loudness and timbre domains. Performances with a close distance can considered to be similar in the corresponding domain and performances clearly separated from the others can be considered to be dissimilar.

Most notable in the tempo domain is the BQ performance that is both very fast and has strong tempo modulations, which is however not a general relationship as there are other performances that are fast with weak modulations (AQ), slow with strong modulations (EQ) and slow with weak modulations (QV1, BSQ).

When ignoring the historical recordings, the loudness domain is marked by EQ (low loudness, high modulation), BQ (lowest modulation) and HSQ (highest loudness).

The BQ performance has the highest brightness, followed by the ABQ performance that additionally features the lowest bandwidth modulation. EQ has

also low bandwidth modulation, as opposed to QV2 and HSQ, the latter also being the performance with the lowest brightness when ignoring the historic recordings.

In the domain of the spectral envelope, BQ represents again an outlier, as do ABQ and AQ.

7.6.1 Dimensionality of Overall Observations

The presented set of 13 overall features has been subjected to PCA in order to investigate which features account for the highest variance between performances. In this case, the overall features represent the PCA variables and the performances their observations (i.e. 21 observations per variable). By some means, this PCA is similar to the PCA done for the feature space dimensionality reduction (see Sect. 7.3.3), but here, only overall observations per performance (such as the mean tempo) are transformed instead of their variation over the whole piece (such as the series of IOIs.

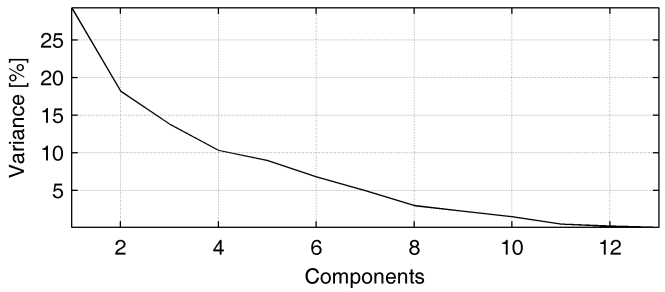

FIGURE 7.17: Relative variance of each component (OPC) resulting from a PCA applied to the overall observations

Figure 7.17 visualizes the relative amount of variance of the components $OPCn$. There is no obvious discontinuity detectable, and the first five components comply with the previously used relevance criterion of an eigenvalue higher than 1 and have been selected. Combined, they account for approximately 75% of the variance.

No obvious relation between the component loadings and individual features can be identified (see Table C.11 for the loadings of each component). Therefore, varimax rotation has been applied to the five components to check whether more clear relationships can be identified between features and the components. The resulting loadings of the rotated components $RPCn$ can be seen in Table 7.6.

While the loadings matrix of the rotated principal components still shows no obvious relationships, some tendencies can be identified. The first rotated component contains relatively high loadings for the three timbre features Spectral Rolloff, Spectral Centroid and the modulation of the Spectral Centroid

	RPC1	***RPC2***	***RPC3***	***RPC4***	***RPC5***
$\overline{BPM}$	−0.05	−0.58	+0.01	+0.02	+0.02
BPM_{Hist}	+0.04	−0.62	+0.02	+0.01	−0.06
$\Delta BPM_{p25,p75}$	−0.05	−0.33	−0.43	−0.20	+0.30
$\sigma(IBI)/\overline{IBI}$	+0.25	+0.11	−0.34	−0.18	−0.32
f_{A4}	−0.15	+0.04	−0.51	+0.05	−0.34
$\overline{v_{1770}}$	+0.14	−0.13	+0.09	−0.10	−0.65
$\sigma(v_{1770})$	+0.22	−0.04	+0.03	−0.08	+0.48
$\overline{v_{SR}}$	−0.51	−0.01	−0.08	+0.30	−0.09
$\sigma(v_{SR})$	+0.07	+0.20	−0.10	+0.61	+0.09
$\overline{v_{SC}}$	−0.50	−0.12	+0.04	−0.08	−0.03
$\sigma(v_{SC})$	−0.57	+0.16	+0.01	−0.21	+0.06
$\overline{v_{MFC3}}$	−0.03	+0.23	−0.07	−0.62	+0.06
$\sigma(v_{MFC3})$	−0.06	−0.03	+0.64	−0.10	−0.14

TABLE 7.6: Loadings of the five rotated PCA components for each overall observations

and thus seems to be brightness related, while the second component has the highest loadings for the overall tempo. The third rotated component represents a combination of tuning frequency, deviation of the spectral envelope, and tempo modulation, and the fourth component shows the highest loadings for v_{MFC2} and bandwidth modulation. The highest loadings of the fifth component are both loudness related components (mean and standard deviation). Note that in the rotated domain, the order of the components is not related anymore to their importance or salience.

Although the components are difficult to interpret, it can be concluded that practically all extracted overall features contribute to the variance between the analyzed performances. Especially the features brightness, tempo, tempo modulation, loudness and tuning frequency contribute to the salient five dimensions.

7.6.2 Relationships between Overall Observations

Correlations between the overall observations, computed over all performances, showed only a few relationships (compare Table C.20, with the indices explained in the following Table C.21). The positive correlation of quartet foundation year and timbre features can most likely be explained by the evolving recording quality over years; alternative explanations are the impact of the four shellac recordings on the correlation results or the change of recording esthetics over time. The negative correlation between the mean loudness and the loudness modulation indicates that loudness variation decreases for louder recordings. The loudness modulation has also a significant negative correlation to the mean Spectral Centroid and the mean v_{MFC2}, a relationship that is hard to explain. Surprising and without explanation is the significant negative correlation of loudness modulation and tuning frequency, as these two values had been expected to be independent from each other.

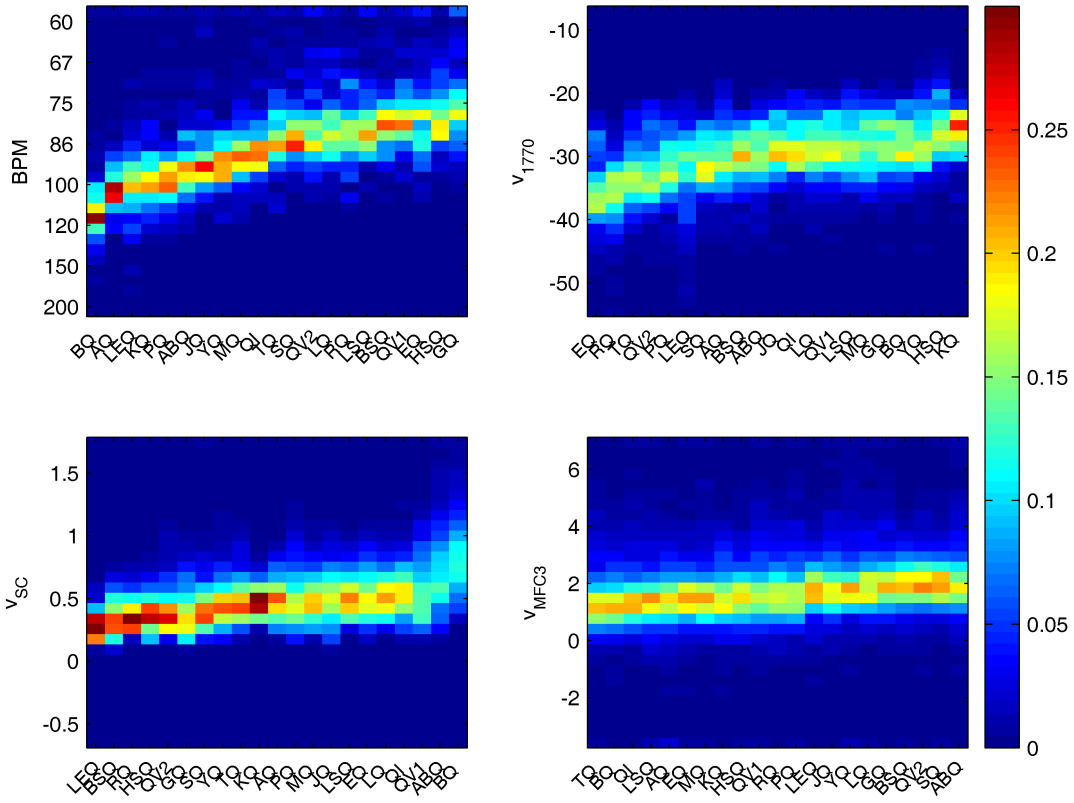

FIGURE 7.18: Distribution (relative occurrence) of feature results per performance, the performances are sorted with respect to their median values

Figure 7.18 displays the relative number of occurrences (density function) of four features for all performances with the performances being sorted according to their feature median to exemplify that there are no obvious relationships between the median value and the deviation from it.

Table 7.7 shows that there are no significant relationships between the continent of origin and the overall observations or between recording date and the variables (the data set has been split only into two groups for this analysis, before 1960 and after 1960). The only significant dependencies that could be detected by this ANOVA analysis are the trivial relationships between the recording date group and the exact recording year as well as the year of foundation.

Feature	*Rel. to Cont. of origin*	*Rel. to rec. year*
Rec.Y.	F=0.66, p=0.43	F=51.72, p=0.00*
Fo.Y.	F=0.69, p=0.42	F=27.67, p=0.00*
$\overline{BPM}$	F=0.14, p=0.71	F=1.83, p=0.19
BPM_{Hist}	F=0.02, p=0.88	F=1.35, p=0.26
Δ_{BPM}	F=0.45, p=0.51	F=0.01, p=0.91
$\sigma_{IOI}/\overline{IOI}$	F=0.05, p=0.82	F=1.71, p=0.21
f_{A4}	F=0.14, p=0.72	F=0.00, p=0.95
$\overline{v_{1770}}$	F=1.80, p=0.20	F=0.10, p=0.76
$\sigma_{v_{1770}}$	F=0.07, p=0.79	F=0.01, p=0.93
$\overline{v_{SC}}$	F=0.10, p=0.76	F=0.46, p=0.50
$\sigma_{v_{SC}}$	F=1.63, p=0.22	F=2.56, p=0.13
$\overline{v_{MFC2}}$	F=0.54, p=0.47	F=0.52, p=0.48
$\sigma_{v_{MFC2}}$	F=0.44, p=0.52	F=2.16, p=0.16
$\overline{v_{MFC3}}$	F=0.59, p=0.45	F=2.84, p=0.11
$\sigma_{v_{MFC3}}$	F=0.52, p=0.48	F=1.76, p=0.20

TABLE 7.7: ANOVA results for dependencies between overall observations and two groups (first column: continent of origin, second column: recorded before or after the year 1960)

7.7 Summary

The variation of the tempo, timing and loudness profiles is closely related to the musical structure, indicated by sudden changes at phrase boundaries and typical shapes corresponding to specific parts. This is to be expected as it verifies the results of several previous studies, e.g. [Pal89], [Rep90], [Rep92], [Rep98].

In the strategy of applying dynamic variations, but also for the variation of tempo and timing, all ensembles seemed to follow similar approaches, as the analysis revealed high similarity between loudness and timing profiles, respectively. In general, the study revealed high correlations between performances. Although the correlation measure cannot be assumed to be of high perceptual relevance, it may indicate that the analyzed performances show more commonalities than individualities. One possible explanation could be the piece of music that has been analyzed, as it might offer only limited degrees of freedom for different approaches to performance because of its dance-form character.

The timing, loudness and timbre profiles have only weak correlations to each other and only for some of the analyzed recordings. This indicates that they are unrelated, independent performance dimensions, a result that supports Repp's findings for tempo and loudness features [Rep99a].

The presented analysis of timbre features yielded inconsistent results. Although the author could verify by ear that the mean Spectral Centroid is indeed a fitting measure for the brightness or sharpness of a complete recording, the general problem of timbre feature interpretation in a musical context still remains unsolved. The computed timbre profiles showed no obvious tendencies that could be used as an impetus for a more detailed timbre analysis. The results of the Principal Component Analysis applied to both the features (PC_F)

and the overall results (OPC, RPC) showed that timbre features cannot be neglected and account for variance between performances. There can be hope that other approaches to timbre analysis of performances might prove to be more successful.

The missing clear relationship between timbre features and musical structure that can be observed for timing and loudness features is an indication that the variation of timbre features is — in contrast to tempo and loudness features — not well suited to represent formal musical (macro-)structure. This result could be helpful in the context of feature selection for algorithms that segment music files into their structural components.

No interrelations between overall performance parameters and other parameters such as recording year and country of origin could be detected; it seems that the cultural background of the musicians has no real impact on tempo, loudness or timbre characteristics of the recording. This confirms findings by Repp ([Rep98], [Rep99a]) and Stolla [Sto04].

The PCA analysis of the overall features showed that all overall performance parameters contributed significantly to the variance between performances.

Chapter 8

Conclusion

8.1 Summary

Music performance is the substantial link in the chain of musical communication between composer and listener. It is the performance that renders the musical ideas contained within the score into a physical realization that can be perceived by the listener, making the performance an appealing object of study for musicologists, psychologists and other scientists. Four objective parameter classes that define a performance can be summarized to be: tempo and timing, loudness, timbre and pitch. Each of these classes contains a multitude of single parameters which can be varied by the performers. For example, they choose their main performance tempo, may perform different parts of a piece in different tempi, can apply *rubati* to specific passages, are able to extend or reduce the length of tacits and notes, and control the articulation of notes.

Three facets of music performance research can be identified in the literature; namely, to study the performance, the performer or the listener. All of them have to deal with some typical difficulties that complicate the analysis process. These difficulties include the identification and interpretation of performance characteristics. However, the process of performance data acquisition probably poses one of the most serious immediate problems. This seems also to be the main reason for performance studies to focus on the analysis of performances with piano or monophonic instruments, because the technical possibilities to extract analysis data from such performances can considered to be mature and exact, as opposed to the data acquisition from performances with multiple instruments.

In order to provide the means for the analysis of ensemble performances, a software system has been designed to enable the automatic extraction of performance parameters from audio recordings by means of audio content analysis algorithms extracting parameters of three classes: tempo, loudness and timbre.

Extraction of Tempo & Timing

The timing is the most important parameter to extract, not only because it can be considered to be the most salient performance attribute. The timing data is also required to convert loudness and timbre features from the time scale to a score-representing beat scale. This is necessary to compare a repertory of several performances at specific score positions.

The timing information is extracted by synchronization (or alignment) of the audio file with a score representation in MIDI format. Dynamic Time Warping (DTW) was used for the path extraction as a standard approach to this problem. DTW is a method that finds an optimal match between two sequences, in this case the audio and the MIDI file. The final result of the warping is the time of each note onset in the score.

The development of the matching algorithm was focused on the design of an appropriate similarity measure between audio and MIDI sequences as this is crucial for the success of the synchronization procedure. The first step, the identification of tonal content in the audio signal, has been accomplished by combining several known approaches: the usage of two STFTs of different lengths for low and higher frequency analysis, the computation of the magnitude spectrum's derivative to pick only components with peak magnitudes, the calculation of the instantaneous frequency to pick only components with matching phase derivatives and the use of a simplified masking threshold to discard irrelevant components.

Since the initial representations of audio and MIDI are quite different, two similarity measures have been developed: one to transform the MIDI data into a representation similar to the internal audio format and the other to convert the audio data in a more score-like representation. The overall similarity measure is computed by superposition of both measures. The latter similarity measure implements methods for polyphonic audio transcription. A rule-based fuzzy model has been developed that computes the probability of a tonal component to be a fundamental frequency, given a set of rules that assign a probability to each base frequency candidate depending on the detected harmonic distribution.

To further improve the accuracy of fundamental frequencies, a new approach has been presented to estimate the tuning frequency of a music recording using a filter bank with constantly adapting mid frequencies. The algorithm is able to detect tuning frequencies in a range of ± 0.5 semi-tones with a comparably high resolution of 1 Hz. The correct determination of the underlying tuning frequency allows pitch-based analysis systems to work more accurately on recordings of instruments tuned to other tuning frequencies than 440 Hz. The algorithm has been published in [Ler06].

The score (or cost) computation of the dynamic time warping algorithm has been extended by the usage of a signal-adaptive onset probability measure to encourage path transitions to a new score event for high onset probabilities and vice versa.

Extraction of Dynamics & Timbre

A set of 15 accepted loudness and timbre features that have shown psychoacoustic and practical relevance in different contexts has been selected and implemented to investigate their relevance in the context of music performance analysis. In order to reduce the dimensionality of this feature space, the features were subjected to principal component analysis (PCA) which revealed one principal loudness-related component and three principal timbre-related components.

To enable fast and easy extensibility toward new and additional features, a plugin interface for the extraction of low level features from audio signals has been introduced. FEAPI (Feature Extraction Plugin API) offers platform independence and is available as full source code [LET08]. The main technical features are a push style interface that allows the processing of live streams as well as file streams, support for the extraction of multidimensional features, variable output sample rates and time stamp handling for the proper synchronization of audio and features. The plugin approach enables the reuse of already implemented features without additional effort. Furthermore, feature extraction plugins could easily be exchanged between projects, researchers and companies, if required also in binary format to protect the intellectual property of the development party. The FEAPI concept has been published in [LET05].

Performance Player

For the visualization of the extracted features (tempo, loudness and timbre) as well as to allow the qualitative comparison of the analyzed performances, the "Performance Player" has been presented. It is a software application that displays the extracted features on a beat scale and allows switching between the performances during play-back to enable comparison by ear at specific score positions.

String Quartet Performance Analysis

A detailed and systematic performance analysis of 21 string quartet performances of a movement of Beethoven's string quartet No. 13 op. 130 has been carried out. To the best knowledge of the author, such a study has not been previously approached. Although the performances had quite different overall tempi, the timing and especially loudness variation yielded high correlation results between performances. This has been shown by calculating both inter-performance correlations and a PCA of the extracted parameters. But as long as the perceptual relevance of such measures remains unclear, it is difficult to draw any conclusions with respect to performance similarity. The formal musical structure is closely matched in the variation of timing and loudness patterns and appears to be the most salient determinant of the performance strategy, a result that corresponds well to the research literature.

The ability of performers to reproduce the performance of a score accurately in the case of repetitions has also been confirmed by this study.

This study is one of the first attempts to the systematic analysis of timbre variation in performances. However, the timbre variations showed no close resemblance to the musical structure like the variations of the other features and were particularly difficult to interpret. Nevertheless, it has been shown that timbre dimensions contribute to the variance between performances. Other approaches to analysis of the timbre parameters might prove more successful in interpreting the result.

No significant relationships between the performance data and socio-cultural parameters such as the country of origin or the recording date could be found.

8.2 Potential Algorithmic Improvements

The implemented algorithms offer several possibilities for improvements as does any machine recognition system. The most promising points for improvements can be found at the tonal peak picking processing stage, the fundamental frequency detection and the onset tracking.

The picking of tonal components could possibly be improved by using a more sophisticated model for the computed masking threshold such as used in perceptual audio coding. The tonality estimation might be improved by utilizing time and frequency domain prediction for the identification of non-tonal (non-predictable) components.

The fundamental frequency detection stage offers many possible ways of improvements. For example, the amplitude of a specific harmonic that at the moment might be fully assigned to two or even more base frequency hypotheses should only be partially assigned to each base frequency, taking into account its relative harmonic position. The pitch detection could be enhanced by additional rules that take into account more information of the voicing of the pitch hypotheses over time. Ultimately, the implementation of a model of established voice leading rules could be used to adjust the likelihood of the extracted fundamental frequency hypotheses. Furthermore, information from the input score file could be used to discard less probable hypotheses (compare [WPD06]).

One idea to improve the onset detection accuracy in the context of audio-to-score-matching systems could be to iteratively adjust the onset tracker's detection sensitivity until the number of detected onsets matches the number of events from the score closely.

An extension of the system toward intonation analysis would add the missing fourth class (pitch) of objective performance parameters that has been — except for the tuning frequency — neglected in the current implementation. The best

way to integrate such a component would be after the dynamic time warping stage when the assignment of each score event to a certain passage of the audio file has already been done. Then, each note of the score could be matched with the extracted base frequencies for this time frame, resulting in detailed intonation information per STFT block for that note.

8.3 Future Directions

The field of music performance research has produced a wealth of data in terms of tempo and loudness information from piano performances, in many cases gathered via MIDI. The presented system allows extension of this data set toward data extracted from audio recordings of both piano and ensemble performances. This will hopefully enable a better understanding of music performance, for example by studying large data sets with respect to general interpretative rules, expressive strategies, historic influences etc. and by analyzing ensemble performances and genres that hitherto have been neglected in performance research.

In the foreseeable future, the rise of optimized and new signal processing approaches will increase robustness and accuracy of the data acquired from audio further, allowing the performance analysis of practically every recording with any ensemble size.

As mentioned previously, this work puts emphasis on the data extraction stage. However, data extraction is only the first requirement necessary in clearing the way for extensive performance analysis. The final instance for the description, interpretation and judgment of a performance is always the listener, but the (subjective) criteria used during this process are only partially known. The identification of perceptually relevant performance dimensions is required to be able to distinguish the substantial data characteristics from negligible characteristics.

Only after determining these perceptual dimensions of music performance is it possible to find valid representations of the extracted data that allow us to characterize specific performances, to identify the perceptually important differences and commonalities between performances and to interpret the results in a meaningful way. However, to come to valid conclusions, this research has to be based on reliable data sets, a fact that emphasizes the importance of robust and accurate parameter extraction. By approaching this topic, the presented work can hopefully serve as a basis for further contributions to the intriguing field of music performance research.

List of Figures

List of Tables

Appendix A

Standard Transformations

This appendix provides a short introduction to some transformations that are used in the thesis but could not be introduced properly. It is not the aim of the following section to explain the algorithmic details but mainly to summarize some of their properties and provide definitions where appropriate.

A.1 Discrete Fourier Transformation

The Discrete Fourier Transformation (DFT) of the input audio signal $x(n)$ is defined as

$$X(k) = \frac{1}{\mathcal{N}} \sum_{n=0}^{\mathcal{N}-1} x(n) \cdot e^{-j2\pi \frac{nk}{\mathcal{N}}} \tag{A.1}$$

Here, the Short Time Fourier Transformation (STFT) over an excerpt of $\mathcal{K}$ samples is calculated for each block index m with a hop size of length $\mathcal{H}$.

$$X(k,m) = \frac{1}{\mathcal{K}} \sum_{n=0}^{\mathcal{K}-1,} w(n) \cdot x(n + m \cdot \mathcal{H}) \cdot e^{-j2\pi \frac{nk}{\mathcal{K}}} \tag{A.2}$$

The window function $w(n)$ is applied to decrease the amount of windowing artifacts. The most frequently used window function in this thesis is the Hanning window:

$$w(n) = \frac{1}{2} \cdot \left(1 - \cos\left(\frac{2\pi \cdot n}{\mathcal{K}+1}\right)\right) \tag{A.3}$$

The result $X(k,m)$ is a vector of complex numbers with the length $\mathcal{K}/2$ for block m. Each bin k represents a specific frequency $f = k \cdot f_S/\mathcal{K}$.

A.2 Principal Component Analysis

Principal Component Analysis (PCA) is an orthogonal transformation that is frequently used for dimensionality reduction. It computes a transformation matrix that maps the input variables into an orthogonal space. The axes of the resulting coordinate system point into the directions of the highest variance. PCA tries to concentrate the main variance in as few transformed variables, the components, as possible. Components with low variance are often considered to be of no importance and are omitted, resulting in a lower dimensional space.

Figure A.1 shows the original and rotated axes for two variables x_1 and x_2.

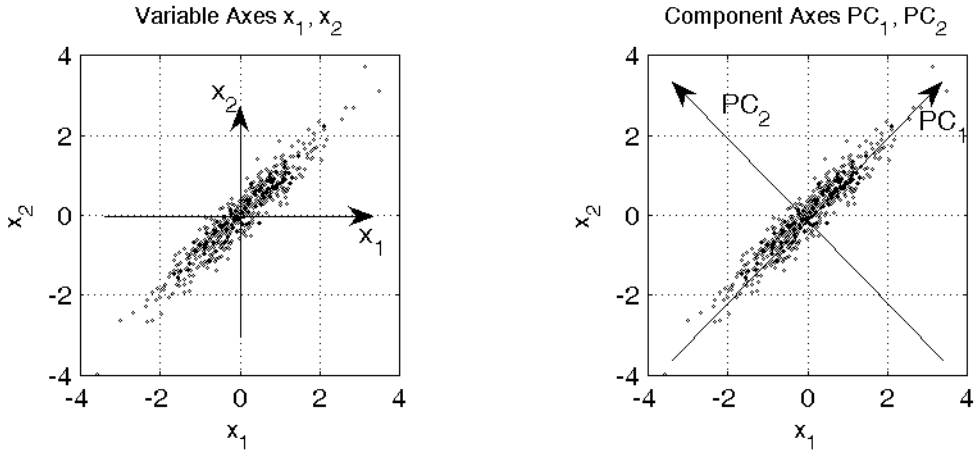

FIGURE A.1: Scatter Plot of a two-dimensional data set with variables x_1, x_2, and the rotated coordinate system after PCA with the component axes PC_1, PC_2

The component matrix **PC** can be computed by multiplication of the variable matrix **V** with the transformation matrix **C**:

$$\mathbf{PC} = \mathbf{C}^T \cdot \mathbf{V} \tag{A.4}$$

Each row of the input variable matrix **V** contains the observations of one variable; with N variables and M observations per variable, the dimensions of matrix **V** is $N \times M$. The coefficient matrix **C** has the dimensions $N \times N$ with each column representing the transformation coefficients for one component. The resulting component matrix **PC** has the same dimensions as the input variable matrix. When the number of dimensions is reduced to N', the coefficient matrix will have the dimensions $N' \times N$.

In the course of the process, the eigenvalues of each component is being computed. This value can be used to select the *principal* components. A typical threshold for omission is an eigenvalue lower than one. This equals the criterion of a threshold of $1/n$ for the relative variance a component accounts for.

Appendix B

Software Documentation

B.1 Parameter Extraction

The software for the performance data extraction is a command line application. That means that on Windows operating systems, a command prompt has to be opened to start the software. The executable is at the moment still named with its development name: `AudioMidiSyncTestCL.exe`.

The following processing steps are being displayed in the course of a program run:

- Audio Feature Extraction: loudness and timbre feature extraction
- Audio Pre-Processing: pre-processing stage of the tempo extraction (onset and tuning frequency detection)
- Audio Processing: computation of the internal series of audio observations
- MIDI Processing: conversion of the MIDI data to the internal score format
- Similarity Matrix Calculation: computation of the similarity measure between score and audio observations
- Alignment Path Extraction: dynamic time warping computation to find the path with the highest overall similarity
- Result File Writing: storing the results in text files
- Memory Deallocation: internal cleanup before the program quits

B.1.1 Command Line

Since the application does not integrate a "normal" command line parser, the order of input arguments must not be changed. The synopsis is

```
AudioMidiSyncTestCL.exe audiofile midifile outdir [synctxtfile]
```

where `audiofile` is the input audio file containing the performance, `midifile` is the quantized midi file containing the score information, `outdir` is the directory where the extracted information is stored into (it is strongly recommended to use `"outdir.analyviz"` as directory name to ensure compatibility with the Performance Player) and `synctxtfile` is a text file containing manual sync point annotations.

B.1.2 Input and Output Files

B.1.2.1 Audio File Format

The widely used `.wav` and `.aif` file formats are supported. The input files are expected to be of reasonable quality in terms of both recording and performance quality.

B.1.2.2 MIDI File Format

MIDI files in formats 0, 1, and 2 are being supported. The input MIDI file should contain a quantized version of the score because it is important that MIDI beats correspond to score events. The input file may contain tempo curve information as this is ignored by the file parser.

The MIDI file has to match the score the performance is based on exactly; e.g. the number of repetitions has to be identical in audio and MIDI file.

B.1.2.3 Sync Text File Format

The optional input text file allows the user to specify a list synchronization points. This allows on the one hand to correct wrong alignment path calculations, on the other hand to speed up the similarity matrix calculation significantly since the number of similarity measures to calculate decreases.

The synchronization points can be specified in a text file as two columns that are separated by a tabulator. The first column contains the time in seconds and the second column the corresponding MIDI tick. Note that the first MIDI tick starts with 0.

An example of a text file with two synchronization points would be

```
52.907  88.5
85.304  145
```

Both columns have to be monotonically increasing with increasing row number. The text file has to be concluded with exactly one empty line.

B.1.2.4 Output Files

Low Level Features

The output files can be found in the directory specified as `outdir` in the command line. The files that contain the low level features follow the following naming scheme: `audiofile.`*PluginName.FeatureIndex*`.feapi.res`, e.g. `abq.wav.Loudness.0.feapi.res`. Each file consists of two tabulator-separated columns; the first column contains the time stamp, and the second column contains the corresponding feature value.

Tuning Frequency

The tuning frequency of the performance is stored in a file named `audiofile.tf.res`, e.g. `abq.wav.tf.res`. This text file contains nothing but a single value, the tuning frequency in Hz.

Timing

The timing information can be found in a file named `audiofile.timeproc.res`, e.g. `abq.wav.timeproc.res`. It contains four columns separated by tabulator. Of these columns, only the first and the fourth are of real interest to the analyst: the first column contains the time in s and the fourth column the corresponding MIDI tick. The second column contains some interpolated tick information, and the third column the onset times extracted from the MIDI file.

B.2 Performance Player

The Performance Player is a software that allows you to play back the analyzed performances while displaying the extracted data. Note that the data extraction stage has to be successfully finished for all performances before loading them.

B.2.1 Loading Performances

To be able to load a performance, the Performance Player requires the extracted data in a subdirectory relative to the input files directory. The subdirectory has to be named as the audio file and has to have an additional extension `.analyviz`. For example the analyzed audio file can be found in the directory `c:\perf\` under the name `abq.wav` then the Performance Player expects to find a directory `c:\perf\abq.wav.analyviz\` that contains all output files (compare Sect. B.1.2.4).

To load the performance data, simply choose one or more audio files (of the same piece of music).

B.2.2 Visualize Parameters

The tempo profile for each file is automatically being displayed upon loading the corresponding audio files. Other features can be selected for display with the Drop-Down box labeled *Show:* on the left-hand side.

Several descriptive values (sub-features like different mean values etc.) are being computed from the selected feature for one file. This file can be selected via a Drop-Down Box in the *Performance Values* section below the feature display.

B.2.3 Play Performances

Playback of the performances is possible by hitting the *Play*-button on the left hand side of the interface. The performance that is currently selected for sub-feature calculation is played back. To switch between performances choose a different performance from the Drop-Down menu. This can be also done during playback to switch instantly to a different performance. The switching is done at the same score position (not at the same time position) to allow easy qualitative comparisons between various performances.

Appendix C

Result Tables - String Quartet Analysis

	IBI_{all}	IOI_{all}	$PC1_{F,all}$	$PC2_{F,all}$	$PC3_{F,all}$	$PC4_{F,all}$
UPC1	+87.62*	+75.94*	+75.87*	+60.64*	+50.16*	+45.39*
UPC2	+2.70	+3.63	+5.03*	+8.39*	+5.60*	+7.26*
UPC3	+1.56	+2.52	+2.51	+3.18	+4.37	+5.23*
UPC4	+1.17	+2.30	+2.15	+2.88	+3.70	+3.87
UPC5	+1.06	+1.98	+1.63	+2.66	+3.34	+3.52
UPC6	+0.75	+1.63	+1.57	+2.40	+3.15	+3.40
UPC7	+0.72	+1.57	+1.27	+2.21	+2.96	+3.27
UPC8	+0.66	+1.39	+1.25	+1.82	+2.68	+3.03
UPC9	+0.52	+1.24	+1.08	+1.65	+2.67	+2.96
UPC10	+0.46	+1.12	+0.97	+1.55	+2.49	+2.56
UPC11	+0.42	+0.99	+0.84	+1.49	+2.25	+2.30
UPC12	+0.39	+0.93	+0.78	+1.35	+2.20	+2.21
UPC13	+0.34	+0.87	+0.76	+1.30	+2.10	+2.10
UPC14	+0.31	+0.67	+0.65	+1.27	+1.85	+2.01
UPC15	+0.26	+0.63	+0.62	+1.21	+1.79	+1.98
UPC16	+0.23	+0.59	+0.61	+1.16	+1.69	+1.72
UPC17	+0.21	+0.54	+0.55	+1.14	+1.63	+1.65
UPC18	+0.20	+0.45	+0.53	+1.07	+1.56	+1.57
UPC19	+0.15	+0.40	+0.51	+0.93	+1.38	+1.37
UPC20	+0.15	+0.31	+0.47	+0.88	+1.23	+1.34
UPC21	+0.13	+0.30	+0.33	+0.83	+1.20	+1.24

TABLE C.1: Amount of variance accounted for by each PCA component UPC for the analysis over the whole piece (rows: principal components, columns: observation labels); the components marked with an asterisk have been selected to be the relevant main components

	IBI_{128}	IOI_{128}	$PC1_{F,128}$	$PC2_{F,128}$	$PC3_{F,128}$	$PC4_{F,128}$
UPC1	+56.06*	+43.20*	+69.88*	+51.03*	+45.24*	+40.26*
UPC2	+5.92*	+5.14*	+5.88*	+8.99*	+6.03*	+6.15*
UPC3	+5.24*	+5.07*	+3.21	+4.22	+4.59	+5.51*
UPC4	+3.94	+4.25	+2.77	+3.97	+4.10	+4.26
UPC5	+3.50	+4.13	+2.38	+3.61	+3.71	+4.06
UPC6	+3.21	+4.01	+2.15	+3.09	+3.46	+3.76
UPC7	+2.72	+3.59	+1.67	+2.56	+3.13	+3.59
UPC8	+2.58	+3.33	+1.48	+2.33	+3.02	+3.42
UPC9	+2.20	+2.96	+1.37	+2.17	+2.84	+3.04
UPC10	+2.06	+2.92	+1.26	+2.13	+2.74	+2.97
UPC11	+1.96	+2.69	+1.10	+1.95	+2.52	+2.66
UPC12	+1.67	+2.56	+0.97	+1.72	+2.47	+2.54
UPC13	+1.54	+2.47	+0.91	+1.66	+2.32	+2.32
UPC14	+1.44	+2.35	+0.84	+1.61	+2.15	+2.23
UPC15	+1.20	+2.24	+0.75	+1.56	+2.02	+2.21
UPC16	+1.14	+1.95	+0.73	+1.50	+1.90	+2.10
UPC17	+0.95	+1.74	+0.65	+1.39	+1.77	+1.91
UPC18	+0.90	+1.57	+0.61	+1.31	+1.72	+1.86
UPC19	+0.66	+1.47	+0.51	+1.17	+1.50	+1.84
UPC20	+0.61	+1.25	+0.49	+1.04	+1.45	+1.74
UPC21	+0.49	+1.10	+0.38	+1.01	+1.31	+1.57

TABLE C.2: Amount of variance accounted for by each PCA component UPC for the analysis over bars $1-128$ (rows: principal components, columns: observation labels); the components marked with * have been selected to be the relevant main components

	IBI_{24}	IOI_{24}	$PC1_{F,24}$	$PC2_{F,24}$	$PC3_{F,24}$	$PC4_{F,24}$
UPC1	+51.51*	+59.73*	+67.66*	+42.69*	+36.91*	+35.10*
UPC2	+9.80*	+6.35*	+6.11*	+10.52*	+7.88*	+8.91*
UPC3	+6.00*	+4.55	+5.18*	+5.99*	+6.75*	+7.57*
UPC4	+5.14*	+4.04	+3.52	+5.04*	+5.98*	+6.04*
UPC5	+4.45	+3.49	+2.46	+4.50	+5.20*	+5.00*
UPC6	+3.67	+2.67	+2.24	+3.70	+4.51	+4.32
UPC7	+3.12	+2.51	+1.97	+3.25	+3.90	+3.78
UPC8	+2.98	+2.18	+1.62	+3.05	+3.65	+3.73
UPC9	+2.53	+2.12	+1.35	+2.84	+3.32	+3.28
UPC10	+2.01	+1.88	+1.30	+2.49	+2.83	+2.84
UPC11	+1.76	+1.69	+1.08	+2.29	+2.76	+2.67
UPC12	+1.41	+1.44	+0.91	+1.89	+2.44	+2.58
UPC13	+1.24	+1.30	+0.80	+1.86	+2.20	+2.25
UPC14	+0.99	+1.12	+0.74	+1.73	+1.92	+1.92
UPC15	+0.88	+0.92	+0.66	+1.59	+1.79	+1.76
UPC16	+0.74	+0.83	+0.55	+1.50	+1.55	+1.71
UPC17	+0.50	+0.82	+0.47	+1.41	+1.53	+1.54
UPC18	+0.38	+0.74	+0.44	+1.16	+1.43	+1.41
UPC19	+0.36	+0.60	+0.36	+0.97	+1.31	+1.31
UPC20	+0.31	+0.53	+0.30	+0.92	+1.16	+1.20
UPC21	+0.23	+0.50	+0.29	+0.62	+0.98	+1.08

TABLE C.3: Amount of variance accounted for by each PCA component UPC for the analysis over bars $1-24$ (rows: principal components, columns: observation labels); the components marked with * have been selected to be the relevant main components

	$UPC1(IBI_{all})$, IBI_{all}	$UPC1(IOI_{all})$, IOI_{all}	$UPC1(PC1_{F,all})$, $PC1_{F,all}$	$UPC1(PC2_{F,all})$, $PC2_{F,all}$	$UPC1(PC3_{F,all})$, $PC3_{F,all}$	$UPC1(PC4_{F,all})$, $PC4_{F,all}$
ABQ	+0.92*	+0.86*	−0.86*	+0.76*	+0.72*	−0.72*
AQ	+0.95*	+0.91*	−0.91*	+0.76*	+0.79*	−0.62*
BQ	+0.90*	+0.77*	−0.83*	+0.81*	+0.72*	−0.66*
BSQ	+0.95*	+0.85*	−0.84*	+0.75*	+0.60*	−0.51*
EQ	+0.95*	+0.93*	−0.95*	+0.82*	+0.81*	−0.76*
GQ	+0.96*	+0.95*	−0.91*	+0.84*	+0.78*	−0.82*
HSQ	+0.97*	+0.95*	−0.92*	+0.85*	+0.72*	−0.74*
JQ	+0.93*	+0.88*	−0.92*	+0.82*	+0.71*	−0.69*
KQ	+0.95*	+0.89*	−0.78*	+0.49*	+0.52*	−0.40*
LEQ	+0.86*	+0.72*	−0.74*	+0.58*	+0.54*	−0.15*
LQ	+0.96*	+0.93*	−0.89*	+0.79*	+0.74*	−0.73*
LSQ	+0.90*	+0.87*	−0.88*	+0.80*	+0.76*	−0.79*
MQ	+0.95*	+0.90*	−0.91*	+0.80*	+0.72*	−0.78*
PQ	+0.93*	+0.83*	−0.90*	+0.86*	+0.72*	−0.67*
QI	+0.96*	+0.93*	−0.93*	+0.82*	+0.73*	−0.74*
QV1	+0.97*	+0.96*	−0.87*	+0.82*	+0.78*	−0.73*
QV2	+0.96*	+0.94*	−0.91*	+0.88*	+0.75*	−0.72*
RQ	+0.73*	+0.53*	−0.50*	+0.45*	+0.44*	−0.29*
SQ	+0.94*	+0.76*	−0.91*	+0.85*	+0.71*	−0.75*
TQ	+0.95*	+0.92*	−0.91*	+0.84*	+0.76*	−0.80*
YQ	+0.94*	+0.88*	−0.90*	+0.79*	+0.73*	−0.63*

TABLE C.4: Correlation between the first principal component $UPC1$ and the corresponding feature values for each performance over the whole piece

	$r(UPC1(IBI_{128}), IBI_{128})$	$r(UPC2(IBI_{128}), IBI_{128})$	$r(UPC3(IBI_{128}), IBI_{128})$
ABQ	−0.69*	−0.18	+0.33*
AQ	−0.73*	+0.35*	+0.02
BQ	−0.67*	+0.37*	−0.35*
BSQ	−0.71*	−0.10	−0.02
EQ	−0.86*	−0.29*	+0.04
GQ	−0.88*	−0.25*	−0.04
HSQ	−0.83*	−0.00	+0.01
JQ	−0.73*	−0.27*	−0.08
KQ	−0.76*	−0.03	+0.08
LEQ	−0.37*	+0.57*	+0.39*
LQ	−0.82*	−0.06	+0.12
LSQ	−0.73*	−0.13	−0.44*
MQ	−0.69*	+0.04	−0.17
PQ	−0.67*	+0.07	+0.51*
QI	−0.78*	+0.22*	+0.09
QV1	−0.85*	−0.02	+0.20*
QV2	−0.86*	−0.25*	+0.07
RQ	−0.64*	+0.26*	−0.23*
SQ	−0.76*	−0.10	−0.14
TQ	−0.77*	−0.06	−0.06
YQ	−0.66*	+0.40*	−0.23*

TABLE C.5: Correlation between the selected principal components $UPCn(IBI_{128})$ and the corresponding feature values IBI_{128} for each performance over the first 128 bars

	$r(UPC1(IOI_{24}), IOI_{24})$	$r(UPC2(IOI_{24}), IOI_{24})$
ABQ	+0.81*	−0.17
AQ	+0.80*	+0.15
BQ	+0.74*	+0.27*
BSQ	+0.79*	+0.09
EQ	+0.83*	−0.22*
GQ	+0.79*	−0.41*
HSQ	+0.89*	+0.11
JQ	+0.82*	−0.07
KQ	+0.84*	−0.00
LEQ	+0.64*	+0.47*
LQ	+0.70*	−0.21*
LSQ	+0.76*	−0.46*
MQ	+0.64*	−0.09
PQ	+0.53*	+0.28*
QI	+0.79*	+0.34*
QV1	+0.89*	+0.08
QV2	+0.83*	−0.26*
RQ	+0.52*	−0.15
SQ	+0.81*	−0.17
TQ	+0.75*	+0.29*
YQ	+0.83*	+0.22*

TABLE C.6: Correlation between the selected principal components $UPCn(IOI_{24})$ and the corresponding feature values IOI_{24} for each performance over the first 24 bars

	$r(UPC1(PC1_{F,all}), PC1_{F,all})$	$r(UPC2(PC1_{F,all}), PC1_{F,all})$
ABQ	−0.86*	−0.07*
AQ	−0.91*	+0.12*
BQ	−0.83*	+0.11*
BSQ	−0.84*	−0.17*
EQ	−0.95*	+0.12*
GQ	−0.91*	−0.07*
HSQ	−0.92*	+0.12*
JQ	−0.92*	+0.04
KQ	−0.78*	−0.39*
LEQ	−0.74*	−0.30*
LQ	−0.89*	+0.08*
LSQ	−0.88*	+0.19*
MQ	−0.91*	+0.05*
PQ	−0.90*	+0.08*
QI	−0.93*	+0.09*
QV1	−0.87*	+0.01
QV2	−0.91*	+0.10*
RQ	−0.50*	−0.79*
SQ	−0.91*	−0.00
TQ	−0.91*	+0.14*
YQ	−0.90*	+0.06*

TABLE C.7: Correlation between the selected principal components $UPCn(PC1_{F,all})$ and the corresponding feature values $PC1_{F,all}$ for each performance over the whole piece

	$r(UPC1(PC2_{F,all}), PC2_{F,all})$	$r(UPC2(PC2_{F,all}), PC2_{F,all})$
ABQ	+0.76*	−0.24*
AQ	+0.76*	+0.04
BQ	+0.81*	−0.16*
BSQ	+0.75*	+0.39*
EQ	+0.82*	−0.25*
GQ	+0.84*	−0.04
HSQ	+0.85*	+0.15*
JQ	+0.82*	−0.18*
KQ	+0.49*	+0.72*
LEQ	+0.58*	+0.57*
LQ	+0.79*	−0.13*
LSQ	+0.80*	+0.01
MQ	+0.80*	−0.20*
PQ	+0.86*	−0.07*
QI	+0.82*	−0.12*
QV1	+0.82*	−0.00
QV2	+0.88*	−0.05*
RQ	+0.45*	+0.66*
SQ	+0.85*	−0.11*
TQ	+0.84*	−0.14*
YQ	+0.79*	−0.08*

TABLE C.8: Correlation between the selected principal components $UPCn(PC2_{F,all})$ and the corresponding feature values $PC2_{F,all}$ for each performance over the whole piece

	$r(UPC1(PC3_{F,all}), PC3_{F,all})$	$r(UPC2(PC3_{F,all}), PC3_{F,all})$
ABQ	+0.72*	−0.09*
AQ	+0.79*	+0.10*
BQ	+0.72*	+0.05
BSQ	+0.60*	+0.32*
EQ	+0.81*	−0.09*
GQ	+0.78*	−0.16*
HSQ	+0.72*	−0.03
JQ	+0.71*	−0.12*
KQ	+0.52*	+0.52*
LEQ	+0.54*	+0.39*
LQ	+0.74*	+0.04
LSQ	+0.76*	−0.17*
MQ	+0.72*	−0.13*
PQ	+0.72*	−0.17*
QI	+0.73*	−0.03
QV1	+0.78*	−0.06*
QV2	+0.75*	+0.03
RQ	+0.44*	+0.64*
SQ	+0.71*	−0.23*
TQ	+0.76*	−0.04
YQ	+0.73*	−0.19*

TABLE C.9: Correlation between the selected principal components $UPCn(PC3_{F,all})$ and the corresponding feature values $PC3_{F,all}$ for each performance over the whole piece

	$r(UPC1(PC4_{F,all}), PC4_{F,all})$	$r(UPC2(PC4_{F,all}), PC4_{F,all})$	$r(UPC3(PC4_{F,all}), PC4_{F,all})$
ABQ	−0.72*	+0.28*	−0.21*
AQ	−0.62*	−0.15*	−0.15*
BQ	−0.66*	+0.05	+0.14*
BSQ	−0.51*	−0.43*	+0.36*
EQ	−0.76*	−0.08*	+0.06*
GQ	−0.82*	+0.04	+0.06*
HSQ	−0.74*	−0.00	+0.02
JQ	−0.69*	−0.18*	−0.03
KQ	−0.40*	−0.44*	+0.57*
LEQ	−0.15*	−0.65*	−0.37*
LQ	−0.73*	+0.11*	−0.22*
LSQ	−0.79*	+0.15*	−0.24*
MQ	−0.78*	+0.22*	+0.01
PQ	−0.67*	+0.11*	+0.13*
QI	−0.74*	+0.01	+0.01
QV1	−0.73*	−0.03	−0.28*
QV2	−0.72*	−0.00	+0.09*
RQ	−0.29*	−0.62*	−0.38*
SQ	−0.75*	+0.24*	−0.01
TQ	−0.80*	+0.15*	+0.09*
YQ	−0.63*	−0.10*	+0.22*

TABLE C.10: Correlation between the selected principal components $UPCn(PC4_{F,all})$ and the corresponding feature values $PC4_{F,all}$ for each performance over the whole piece

	OPC1	***OPC2***	***OPC3***	***OPC4***	***OPC5***
$\overline{BPM}$	−0.32	−0.44	−0.09	+0.09	−0.17
BPM_{Hist}	−0.30	−0.43	−0.11	+0.16	−0.27
$\Delta BPM_{p25,p75}$	−0.06	−0.39	−0.25	−0.45	−0.11
$\sigma(IBI)/\overline{IBI}$	+0.13	+0.29	−0.29	−0.10	−0.37
f_{A4}	−0.21	+0.29	−0.49	−0.16	−0.06
$\overline{v_{1770}}$	−0.24	+0.30	−0.06	+0.33	−0.47
$\sigma(v_{1770})$	+0.34	−0.35	+0.12	−0.16	+0.04
$\overline{v_{SR}}$	−0.42	+0.10	−0.14	+0.00	+0.40
$\sigma(v_{SR})$	+0.21	+0.03	−0.32	+0.28	+0.46
$\overline{v_{SC}}$	−0.46	+0.01	+0.12	−0.17	+0.15
$\sigma(v_{SC})$	−0.33	+0.18	+0.23	−0.37	+0.24
$\overline{v_{MFC3}}$	+0.10	+0.20	+0.28	−0.50	−0.27
$\sigma(v_{MFC3})$	−0.15	+0.05	+0.55	+0.33	−0.07

TABLE C.11: Loadings for the five selected components resulting from the PCA over the overall features (OPC)

	ABQ	AQ	BQ	BSQ	EQ	GQ	HSQ	JQ	KQ	LEQ	LQ	LSQ	MQ	PQ	QI	QV1	QV2	RQ	SQ	TQ	YQ
ABQ	1.00*	0.86*	0.80*	0.89*	0.91*	0.90*	0.89*	0.86*	0.89*	0.80*	0.89*	0.82*	0.87*	0.86*	0.87*	0.90*	0.89*	0.63*	0.89*	0.87*	0.85*
AQ	0.86*	1.00*	0.85*	0.89*	0.87*	0.89*	0.94*	0.89*	0.90*	0.82*	0.90*	0.83*	0.92*	0.89*	0.94*	0.94*	0.91*	0.67*	0.88*	0.92*	0.93*
BQ	0.80*	0.85*	1.00*	0.84*	0.86*	0.85*	0.87*	0.82*	0.85*	0.80*	0.86*	0.85*	0.87*	0.82*	0.84*	0.86*	0.83*	0.72*	0.86*	0.86*	0.88*
BSQ	0.89*	0.89*	0.84*	1.00*	0.89*	0.92*	0.93*	0.88*	0.91*	0.82*	0.92*	0.86*	0.93*	0.88*	0.91*	0.92*	0.90*	0.66*	0.88*	0.91*	0.90*
EQ	0.91*	0.87*	0.86*	0.89*	1.00*	0.96*	0.93*	0.90*	0.90*	0.77*	0.92*	0.90*	0.90*	0.88*	0.89*	0.92*	0.93*	0.71*	0.91*	0.91*	0.86*
GQ	0.90*	0.89*	0.85*	0.92*	0.96*	1.00*	0.93*	0.93*	0.91*	0.79*	0.93*	0.90*	0.91*	0.88*	0.91*	0.93*	0.93*	0.71*	0.91*	0.93*	0.87*
HSQ	0.89*	0.94*	0.87*	0.93*	0.93*	0.93*	1.00*	0.91*	0.93*	0.83*	0.94*	0.89*	0.95*	0.92*	0.94*	0.96*	0.94*	0.70*	0.91*	0.94*	0.92*
JQ	0.86*	0.89*	0.82*	0.88*	0.90*	0.93*	0.91*	1.00*	0.91*	0.77*	0.89*	0.84*	0.89*	0.85*	0.91*	0.91*	0.91*	0.66*	0.87*	0.91*	0.88*
KQ	0.89*	0.90*	0.85*	0.91*	0.90*	0.91*	0.93*	0.91*	1.00*	0.84*	0.92*	0.83*	0.91*	0.90*	0.94*	0.94*	0.93*	0.68*	0.89*	0.91*	0.90*
LEQ	0.80*	0.82*	0.80*	0.82*	0.77*	0.79*	0.83*	0.77*	0.84*	1.00*	0.84*	0.73*	0.83*	0.83*	0.86*	0.84*	0.83*	0.57*	0.80*	0.83*	0.82*
LQ	0.89*	0.90*	0.86*	0.92*	0.92*	0.93*	0.94*	0.89*	0.92*	0.84*	1.00*	0.87*	0.92*	0.92*	0.92*	0.94*	0.93*	0.69*	0.89*	0.91*	0.89*
LSQ	0.82*	0.83*	0.85*	0.86*	0.90*	0.90*	0.89*	0.84*	0.83*	0.73*	0.87*	1.00*	0.88*	0.78*	0.83*	0.86*	0.87*	0.71*	0.86*	0.86*	0.85*
MQ	0.87*	0.92*	0.87*	0.93*	0.90*	0.91*	0.95*	0.89*	0.91*	0.83*	0.92*	0.88*	1.00*	0.90*	0.94*	0.93*	0.91*	0.65*	0.90*	0.91*	0.93*
PQ	0.86*	0.89*	0.82*	0.88*	0.88*	0.88*	0.92*	0.85*	0.90*	0.83*	0.92*	0.78*	0.90*	1.00*	0.91*	0.93*	0.91*	0.64*	0.86*	0.89*	0.88*
QI	0.87*	0.94*	0.84*	0.91*	0.89*	0.91*	0.94*	0.91*	0.94*	0.86*	0.92*	0.83*	0.94*	0.91*	1.00*	0.94*	0.94*	0.64*	0.89*	0.92*	0.91*
QV1	0.90*	0.94*	0.86*	0.92*	0.92*	0.93*	0.96*	0.91*	0.94*	0.84*	0.94*	0.86*	0.93*	0.93*	0.94*	1.00*	0.95*	0.67*	0.90*	0.94*	0.92*
QV2	0.89*	0.91*	0.83*	0.90*	0.93*	0.93*	0.94*	0.91*	0.93*	0.83*	0.93*	0.87*	0.91*	0.91*	0.94*	0.95*	1.00*	0.65*	0.89*	0.92*	0.89*
RQ	0.63*	0.67*	0.72*	0.66*	0.71*	0.71*	0.70*	0.66*	0.68*	0.57*	0.69*	0.71*	0.65*	0.64*	0.64*	0.67*	0.65*	1.00*	0.70*	0.69*	0.68*
SQ	0.89*	0.88*	0.86*	0.88*	0.91*	0.91*	0.91*	0.87*	0.89*	0.80*	0.89*	0.86*	0.90*	0.86*	0.89*	0.90*	0.89*	0.70*	1.00*	0.91*	0.88*
TQ	0.87*	0.92*	0.86*	0.91*	0.91*	0.93*	0.94*	0.91*	0.91*	0.83*	0.91*	0.86*	0.91*	0.89*	0.92*	0.94*	0.92*	0.69*	0.91*	1.00*	0.91*
YQ	0.85*	0.93*	0.88*	0.90*	0.86*	0.87*	0.92*	0.88*	0.90*	0.82*	0.89*	0.85*	0.93*	0.88*	0.91*	0.92*	0.89*	0.68*	0.88*	0.91*	1.00*

TABLE C.12: *IBI* Profile correlation between all performances (whole piece)

	ABQ	AQ	BQ	BSQ	EQ	GQ	HSQ	JQ	KQ	LEQ	LQ	LSQ	MQ	PQ	QI	QV1	QV2	RQ	SQ	TQ	YQ
ABQ	1.00*	0.79*	0.66*	0.73*	0.81*	0.81*	0.79*	0.76*	0.74*	0.58*	0.79*	0.69*	0.75*	0.71*	0.78*	0.80*	0.81*	0.45*	0.68*	0.77*	0.75*
AQ	0.79*	1.00*	0.70*	0.76*	0.84*	0.85*	0.88*	0.80*	0.80*	0.65*	0.84*	0.78*	0.82*	0.75*	0.86*	0.89*	0.83*	0.47*	0.67*	0.84*	0.82*
BQ	0.66*	0.70*	1.00*	0.65*	0.72*	0.70*	0.71*	0.68*	0.67*	0.59*	0.71*	0.66*	0.70*	0.60*	0.71*	0.72*	0.68*	0.40*	0.54*	0.68*	0.69*
BSQ	0.73*	0.76*	0.65*	1.00*	0.79*	0.82*	0.80*	0.73*	0.74*	0.60*	0.79*	0.73*	0.77*	0.69*	0.78*	0.82*	0.78*	0.41*	0.64*	0.79*	0.70*
EQ	0.81*	0.84*	0.72*	0.79*	1.00*	0.92*	0.88*	0.83*	0.80*	0.65*	0.88*	0.82*	0.84*	0.79*	0.86*	0.89*	0.89*	0.47*	0.69*	0.84*	0.79*
GQ	0.81*	0.85*	0.70*	0.82*	0.92*	1.00*	0.89*	0.85*	0.80*	0.65*	0.90*	0.83*	0.85*	0.78*	0.88*	0.91*	0.90*	0.48*	0.70*	0.88*	0.80*
HSQ	0.79*	0.88*	0.71*	0.80*	0.88*	0.89*	1.00*	0.83*	0.85*	0.69*	0.87*	0.83*	0.86*	0.79*	0.89*	0.93*	0.89*	0.50*	0.69*	0.87*	0.83*
JQ	0.76*	0.80*	0.68*	0.73*	0.83*	0.85*	0.83*	1.00*	0.75*	0.58*	0.81*	0.72*	0.80*	0.71*	0.83*	0.83*	0.82*	0.44*	0.61*	0.81*	0.82*
KQ	0.74*	0.80*	0.67*	0.74*	0.80*	0.80*	0.85*	0.75*	1.00*	0.61*	0.80*	0.76*	0.78*	0.72*	0.81*	0.86*	0.84*	0.53*	0.69*	0.79*	0.80*
LEQ	0.58*	0.65*	0.59*	0.60*	0.65*	0.65*	0.69*	0.58*	0.61*	1.00*	0.66*	0.63*	0.62*	0.59*	0.69*	0.69*	0.64*	0.36*	0.53*	0.66*	0.61*
LQ	0.79*	0.84*	0.71*	0.79*	0.88*	0.90*	0.87*	0.81*	0.80*	0.66*	1.00*	0.79*	0.85*	0.78*	0.86*	0.88*	0.87*	0.45*	0.68*	0.84*	0.78*
LSQ	0.69*	0.78*	0.66*	0.73*	0.82*	0.83*	0.83*	0.72*	0.76*	0.63*	0.79*	1.00*	0.77*	0.68*	0.77*	0.83*	0.81*	0.49*	0.63*	0.77*	0.77*
MQ	0.75*	0.82*	0.70*	0.77*	0.84*	0.85*	0.86*	0.80*	0.78*	0.62*	0.85*	0.77*	1.00*	0.74*	0.85*	0.86*	0.85*	0.42*	0.64*	0.81*	0.79*
PQ	0.71*	0.75*	0.60*	0.69*	0.79*	0.78*	0.79*	0.71*	0.72*	0.59*	0.78*	0.68*	0.74*	1.00*	0.79*	0.79*	0.80*	0.41*	0.60*	0.74*	0.71*
QI	0.78*	0.86*	0.71*	0.78*	0.86*	0.88*	0.89*	0.83*	0.81*	0.69*	0.86*	0.77*	0.85*	0.79*	1.00*	0.89*	0.88*	0.44*	0.69*	0.85*	0.81*
QV1	0.80*	0.89*	0.72*	0.82*	0.89*	0.91*	0.93*	0.83*	0.86*	0.69*	0.88*	0.83*	0.86*	0.79*	0.89*	1.00*	0.91*	0.48*	0.72*	0.89*	0.85*
QV2	0.81*	0.83*	0.68*	0.78*	0.89*	0.90*	0.89*	0.82*	0.84*	0.64*	0.87*	0.81*	0.85*	0.80*	0.88*	0.91*	1.00*	0.47*	0.70*	0.85*	0.82*
RQ	0.45*	0.47*	0.40*	0.41*	0.47*	0.48*	0.50*	0.44*	0.53*	0.36*	0.45*	0.49*	0.42*	0.41*	0.44*	0.48*	0.47*	1.00*	0.41*	0.43*	0.45*
SQ	0.68*	0.67*	0.54*	0.64*	0.69*	0.70*	0.69*	0.61*	0.69*	0.53*	0.68*	0.63*	0.64*	0.60*	0.69*	0.72*	0.70*	0.41*	1.00*	0.72*	0.67*
TQ	0.77*	0.84*	0.68*	0.79*	0.84*	0.88*	0.87*	0.81*	0.79*	0.66*	0.84*	0.77*	0.81*	0.74*	0.85*	0.89*	0.85*	0.43*	0.72*	1.00*	0.80*
YQ	0.75*	0.82*	0.69*	0.70*	0.79*	0.80*	0.83*	0.82*	0.80*	0.61*	0.78*	0.77*	0.79*	0.71*	0.81*	0.85*	0.82*	0.45*	0.67*	0.80*	1.00*

TABLE C.13: *IOI* Profile correlation between all performances (whole piece)

	ABQ	AQ	BQ	BSQ	EQ	GQ	HSQ	JQ	KQ	LEQ	LQ	LSQ	MQ	PQ	QI	QV1	QV2	RQ	SQ	TQ	YQ
ABQ	1.00*	0.44*	0.35*	0.63*	0.66*	0.64*	0.49*	0.47*	0.55*	0.32*	0.59*	0.40*	0.39*	0.49*	0.50*	0.60*	0.60*	0.32*	0.58*	0.47*	0.37*
AQ	0.44*	1.00*	0.63*	0.42*	0.56*	0.57*	0.62*	0.48*	0.50*	0.33*	0.55*	0.42*	0.46*	0.53*	0.65*	0.66*	0.58*	0.51*	0.50*	0.56*	0.61*
BQ	0.35*	0.63*	1.00*	0.41*	0.49*	0.54*	0.51*	0.39*	0.44*	0.27*	0.53*	0.58*	0.51*	0.34*	0.54*	0.55*	0.47*	0.52*	0.51*	0.48*	0.58*
BSQ	0.63*	0.42*	0.41*	1.00*	0.55*	0.65*	0.60*	0.49*	0.55*	0.24*	0.55*	0.52*	0.56*	0.41*	0.55*	0.51*	0.60*	0.41*	0.55*	0.53*	0.49*
EQ	0.66*	0.56*	0.49*	0.55*	1.00*	0.85*	0.70*	0.65*	0.62*	0.20*	0.73*	0.67*	0.55*	0.57*	0.58*	0.78*	0.83*	0.47*	0.68*	0.66*	0.44*
GQ	0.64*	0.57*	0.54*	0.65*	0.85*	1.00*	0.71*	0.73*	0.65*	0.23*	0.75*	0.71*	0.59*	0.53*	0.62*	0.73*	0.82*	0.52*	0.65*	0.74*	0.46*
HSQ	0.49*	0.62*	0.51*	0.60*	0.70*	0.71*	1.00*	0.54*	0.59*	0.30*	0.72*	0.65*	0.65*	0.60*	0.64*	0.73*	0.72*	0.52*	0.58*	0.62*	0.49*
JQ	0.47*	0.48*	0.39*	0.49*	0.65*	0.73*	0.54*	1.00*	0.57*	0.12	0.60*	0.52*	0.44*	0.43*	0.52*	0.58*	0.67*	0.46*	0.57*	0.61*	0.45*
KQ	0.55*	0.50*	0.44*	0.55*	0.62*	0.65*	0.59*	0.57*	1.00*	0.23*	0.60*	0.47*	0.48*	0.53*	0.66*	0.64*	0.63*	0.54*	0.58*	0.55*	0.49*
LEQ	0.32*	0.33*	0.27*	0.24*	0.20*	0.23*	0.30*	0.12	0.23*	1.00*	0.33*	0.19	0.23*	0.31*	0.39*	0.31*	0.20*	0.25*	0.22*	0.26*	0.29*
LQ	0.59*	0.55*	0.53*	0.55*	0.73*	0.75*	0.72*	0.60*	0.60*	0.33*	1.00*	0.59*	0.54*	0.60*	0.61*	0.72*	0.70*	0.51*	0.57*	0.58*	0.47*
LSQ	0.40*	0.42*	0.58*	0.52*	0.67*	0.71*	0.65*	0.52*	0.47*	0.19	0.59*	1.00*	0.55*	0.22*	0.50*	0.54*	0.65*	0.48*	0.59*	0.57*	0.46*
MQ	0.39*	0.46*	0.51*	0.56*	0.55*	0.59*	0.65*	0.44*	0.48*	0.23*	0.54*	0.55*	1.00*	0.44*	0.50*	0.54*	0.56*	0.36*	0.52*	0.44*	0.50*
PQ	0.49*	0.53*	0.34*	0.41*	0.57*	0.53*	0.60*	0.43*	0.53*	0.31*	0.60*	0.22*	0.44*	1.00*	0.52*	0.69*	0.59*	0.37*	0.41*	0.48*	0.40*
QI	0.50*	0.65*	0.54*	0.55*	0.58*	0.62*	0.64*	0.52*	0.66*	0.39*	0.61*	0.50*	0.50*	0.52*	1.00*	0.67*	0.63*	0.51*	0.53*	0.59*	0.53*
QV1	0.60*	0.66*	0.55*	0.51*	0.78*	0.73*	0.73*	0.58*	0.64*	0.31*	0.72*	0.54*	0.54*	0.69*	0.67*	1.00*	0.80*	0.52*	0.61*	0.64*	0.47*
QV2	0.60*	0.58*	0.47*	0.60*	0.83*	0.82*	0.72*	0.67*	0.63*	0.20*	0.70*	0.65*	0.56*	0.59*	0.63*	0.80*	1.00*	0.47*	0.59*	0.69*	0.46*
RQ	0.32*	0.51*	0.52*	0.41*	0.47*	0.52*	0.52*	0.46*	0.54*	0.25*	0.51*	0.48*	0.36*	0.37*	0.51*	0.52*	0.47*	1.00*	0.50*	0.49*	0.50*
SQ	0.58*	0.50*	0.51*	0.55*	0.68*	0.65*	0.58*	0.57*	0.58*	0.22*	0.57*	0.59*	0.52*	0.41*	0.53*	0.61*	0.59*	0.50*	1.00*	0.60*	0.49*
TQ	0.47*	0.56*	0.48*	0.53*	0.66*	0.74*	0.62*	0.61*	0.55*	0.26*	0.58*	0.57*	0.44*	0.48*	0.59*	0.64*	0.69*	0.49*	0.60*	1.00*	0.52*
YQ	0.37*	0.61*	0.58*	0.49*	0.44*	0.46*	0.49*	0.45*	0.49*	0.29*	0.47*	0.46*	0.50*	0.40*	0.53*	0.47*	0.46*	0.50*	0.49*	0.52*	1.00*

TABLE C.14: *IBI* Profile correlation between all performances (128 bars)

	ABQ	AQ	BQ	BSQ	EQ	GQ	HSQ	JQ	KQ	LEQ	LQ	LSQ	MQ	PQ	QI	QV1	QV2	RQ	SQ	TQ	YQ
ABQ	1.00*	0.47*	0.32*	0.53*	0.52*	0.52*	0.42*	0.37*	0.41*	0.21*	0.47*	0.30*	0.31*	0.37*	0.41*	0.50*	0.55*	0.21*	0.42*	0.45*	0.40*
AQ	0.47*	1.00*	0.42*	0.44*	0.48*	0.48*	0.51*	0.36*	0.38*	0.21*	0.46*	0.34*	0.33*	0.37*	0.47*	0.54*	0.46*	0.26*	0.29*	0.47*	0.42*
BQ	0.32*	0.42*	1.00*	0.40*	0.41*	0.41*	0.39*	0.36*	0.33*	0.27*	0.39*	0.34*	0.35*	0.20*	0.42*	0.46*	0.32*	0.16*	0.22*	0.36*	0.38*
BSQ	0.53*	0.44*	0.40*	1.00*	0.52*	0.56*	0.49*	0.39*	0.43*	0.21*	0.45*	0.46*	0.35*	0.32*	0.48*	0.55*	0.52*	0.22*	0.39*	0.51*	0.39*
EQ	0.52*	0.48*	0.41*	0.52*	1.00*	0.74*	0.54*	0.49*	0.42*	0.22*	0.57*	0.48*	0.39*	0.45*	0.48*	0.67*	0.67*	0.22*	0.40*	0.53*	0.36*
GQ	0.52*	0.48*	0.41*	0.56*	0.74*	1.00*	0.55*	0.54*	0.44*	0.19*	0.61*	0.57*	0.41*	0.43*	0.47*	0.66*	0.68*	0.27*	0.41*	0.60*	0.40*
HSQ	0.42*	0.51*	0.39*	0.49*	0.54*	0.55*	1.00*	0.40*	0.51*	0.28*	0.51*	0.46*	0.40*	0.43*	0.47*	0.63*	0.58*	0.29*	0.35*	0.47*	0.39*
JQ	0.37*	0.36*	0.36*	0.39*	0.49*	0.54*	0.40*	1.00*	0.33*	0.14*	0.46*	0.33*	0.35*	0.31*	0.38*	0.47*	0.49*	0.22*	0.23*	0.42*	0.52*
KQ	0.41*	0.38*	0.33*	0.43*	0.42*	0.44*	0.51*	0.33*	1.00*	0.22*	0.41*	0.31*	0.29*	0.34*	0.39*	0.44*	0.45*	0.40*	0.33*	0.37*	0.30*
LEQ	0.21*	0.21*	0.27*	0.21*	0.22*	0.19*	0.28*	0.14*	0.22*	1.00*	0.21*	0.19*	0.16*	0.14*	0.27*	0.28*	0.17*	0.14*	0.15*	0.24*	0.24*
LQ	0.47*	0.46*	0.39*	0.45*	0.57*	0.61*	0.51*	0.46*	0.41*	0.21*	1.00*	0.45*	0.44*	0.40*	0.46*	0.59*	0.54*	0.22*	0.34*	0.46*	0.36*
LSQ	0.30*	0.34*	0.34*	0.46*	0.48*	0.57*	0.46*	0.33*	0.31*	0.19*	0.45*	1.00*	0.31*	0.23*	0.32*	0.41*	0.49*	0.23*	0.32*	0.34*	0.34*
MQ	0.31*	0.33*	0.35*	0.35*	0.39*	0.41*	0.40*	0.35*	0.29*	0.16*	0.44*	0.31*	1.00*	0.28*	0.40*	0.42*	0.41*	0.13*	0.26*	0.30*	0.31*
PQ	0.37*	0.37*	0.20*	0.32*	0.45*	0.43*	0.43*	0.31*	0.34*	0.14*	0.40*	0.23*	0.28*	1.00*	0.35*	0.48*	0.44*	0.17*	0.24*	0.37*	0.30*
QI	0.41*	0.47*	0.42*	0.48*	0.48*	0.47*	0.47*	0.38*	0.39*	0.27*	0.46*	0.32*	0.40*	0.35*	1.00*	0.56*	0.45*	0.20*	0.35*	0.46*	0.44*
QV1	0.50*	0.54*	0.46*	0.55*	0.67*	0.66*	0.63*	0.47*	0.44*	0.28*	0.59*	0.41*	0.42*	0.48*	0.56*	1.00*	0.66*	0.26*	0.40*	0.56*	0.41*
QV2	0.55*	0.46*	0.32*	0.52*	0.67*	0.68*	0.58*	0.49*	0.45*	0.17*	0.54*	0.49*	0.41*	0.44*	0.45*	0.66*	1.00*	0.24*	0.45*	0.51*	0.39*
RQ	0.21*	0.26*	0.16*	0.22*	0.22*	0.27*	0.29*	0.22*	0.40*	0.14*	0.22*	0.23*	0.13*	0.17*	0.20*	0.26*	0.24*	1.00*	0.23*	0.21*	0.17*
SQ	0.42*	0.29*	0.22*	0.39*	0.40*	0.41*	0.35*	0.23*	0.33*	0.15*	0.34*	0.32*	0.26*	0.24*	0.35*	0.40*	0.45*	0.23*	1.00*	0.47*	0.31*
TQ	0.45*	0.47*	0.36*	0.51*	0.53*	0.60*	0.47*	0.42*	0.37*	0.24*	0.46*	0.34*	0.30*	0.37*	0.46*	0.56*	0.51*	0.21*	0.47*	1.00*	0.40*
YQ	0.40*	0.42*	0.38*	0.39*	0.36*	0.40*	0.39*	0.52*	0.30*	0.24*	0.36*	0.34*	0.31*	0.30*	0.44*	0.41*	0.39*	0.17*	0.31*	0.40*	1.00*

TABLE C.15: *IOI* Profile correlation between all performances (128 bars)

	ABQ	AQ	BQ	BSQ	EQ	GQ	HSQ	JQ	KQ	LEQ	LQ	LSQ	MQ	PQ	QI	QV1	QV2	RQ	SQ	TQ	YQ
ABQ	1.00*	0.76*	0.62*	0.72*	0.80*	0.79*	0.77*	0.76*	0.70*	0.62*	0.78*	0.71*	0.76*	0.81*	0.77*	0.78*	0.77*	0.46*	0.76*	0.76*	0.74*
AQ	0.76*	1.00*	0.78*	0.71*	0.90*	0.81*	0.84*	0.86*	0.66*	0.63*	0.80*	0.80*	0.81*	0.84*	0.85*	0.79*	0.82*	0.39*	0.81*	0.86*	0.83*
BQ	0.62*	0.78*	1.00*	0.67*	0.78*	0.69*	0.73*	0.75*	0.64*	0.61*	0.75*	0.76*	0.80*	0.70*	0.79*	0.64*	0.72*	0.34*	0.74*	0.79*	0.75*
BSQ	0.72*	0.71*	0.67*	1.00*	0.74*	0.80*	0.76*	0.75*	0.70*	0.64*	0.75*	0.72*	0.77*	0.76*	0.76*	0.69*	0.75*	0.51*	0.77*	0.72*	0.74*
EQ	0.80*	0.90*	0.78*	0.74*	1.00*	0.84*	0.87*	0.89*	0.70*	0.67*	0.84*	0.86*	0.83*	0.86*	0.88*	0.82*	0.87*	0.40*	0.85*	0.90*	0.85*
GQ	0.79*	0.81*	0.69*	0.80*	0.84*	1.00*	0.85*	0.84*	0.71*	0.67*	0.80*	0.81*	0.81*	0.83*	0.82*	0.81*	0.82*	0.51*	0.81*	0.81*	0.79*
HSQ	0.77*	0.84*	0.73*	0.76*	0.87*	0.85*	1.00*	0.87*	0.62*	0.65*	0.81*	0.81*	0.83*	0.86*	0.84*	0.81*	0.84*	0.39*	0.84*	0.84*	0.84*
JQ	0.76*	0.86*	0.75*	0.75*	0.89*	0.84*	0.87*	1.00*	0.66*	0.66*	0.82*	0.80*	0.82*	0.85*	0.85*	0.80*	0.82*	0.46*	0.84*	0.84*	0.83*
KQ	0.70*	0.66*	0.64*	0.70*	0.70*	0.71*	0.62*	0.66*	1.00*	0.65*	0.67*	0.62*	0.68*	0.64*	0.70*	0.65*	0.67*	0.61*	0.70*	0.68*	0.68*
LEQ	0.62*	0.63*	0.61*	0.64*	0.67*	0.67*	0.65*	0.66*	0.65*	1.00*	0.56*	0.60*	0.63*	0.59*	0.61*	0.60*	0.65*	0.50*	0.67*	0.64*	0.69*
LQ	0.78*	0.80*	0.75*	0.75*	0.84*	0.80*	0.81*	0.82*	0.67*	0.56*	1.00*	0.77*	0.80*	0.80*	0.84*	0.77*	0.78*	0.41*	0.78*	0.82*	0.76*
LSQ	0.71*	0.80*	0.76*	0.72*	0.86*	0.81*	0.81*	0.80*	0.62*	0.60*	0.77*	1.00*	0.83*	0.77*	0.83*	0.74*	0.81*	0.32*	0.78*	0.86*	0.77*
MQ	0.76*	0.81*	0.80*	0.77*	0.83*	0.81*	0.83*	0.82*	0.68*	0.63*	0.80*	0.83*	1.00*	0.81*	0.85*	0.78*	0.82*	0.43*	0.83*	0.81*	0.80*
PQ	0.81*	0.84*	0.70*	0.76*	0.86*	0.83*	0.86*	0.85*	0.64*	0.59*	0.80*	0.77*	0.81*	1.00*	0.83*	0.80*	0.82*	0.40*	0.83*	0.80*	0.80*
QI	0.77*	0.85*	0.79*	0.76*	0.88*	0.82*	0.84*	0.85*	0.70*	0.61*	0.84*	0.83*	0.85*	0.83*	1.00*	0.78*	0.84*	0.41*	0.82*	0.86*	0.82*
QV1	0.78*	0.79*	0.64*	0.69*	0.82*	0.81*	0.81*	0.80*	0.65*	0.60*	0.77*	0.74*	0.78*	0.80*	0.78*	1.00*	0.83*	0.44*	0.78*	0.77*	0.77*
QV2	0.77*	0.82*	0.72*	0.75*	0.87*	0.82*	0.84*	0.82*	0.67*	0.65*	0.78*	0.81*	0.82*	0.82*	0.84*	0.83*	1.00*	0.37*	0.84*	0.83*	0.83*
RQ	0.46*	0.39*	0.34*	0.51*	0.40*	0.51*	0.39*	0.46*	0.61*	0.50*	0.41*	0.32*	0.43*	0.40*	0.41*	0.44*	0.37*	1.00*	0.45*	0.37*	0.38*
SQ	0.76*	0.81*	0.74*	0.77*	0.85*	0.81*	0.84*	0.84*	0.70*	0.67*	0.78*	0.78*	0.83*	0.83*	0.82*	0.78*	0.84*	0.45*	1.00*	0.80*	0.83*
TQ	0.76*	0.86*	0.79*	0.72*	0.90*	0.81*	0.84*	0.84*	0.68*	0.64*	0.82*	0.86*	0.81*	0.80*	0.86*	0.77*	0.83*	0.37*	0.80*	1.00*	0.81*
YQ	0.74*	0.83*	0.75*	0.74*	0.85*	0.79*	0.84*	0.83*	0.68*	0.69*	0.76*	0.77*	0.80*	0.80*	0.82*	0.77*	0.83*	0.38*	0.83*	0.81*	1.00*

TABLE C.16: $PC1_F$ Profile correlation between all performances (whole piece)

	ABQ	AQ	BQ	BSQ	EQ	GQ	HSQ	JQ	KQ	LEQ	LQ	LSQ	MQ	PQ	QI	QV1	QV2	RQ	SQ	TQ	YQ
ABQ	1.00*	0.58*	0.68*	0.50*	0.67*	0.57*	0.55*	0.61*	0.24*	0.33*	0.61*	0.59*	0.60*	0.68*	0.65*	0.55*	0.63*	0.20*	0.68*	0.64*	0.57*
AQ	0.58*	1.00*	0.65*	0.52*	0.62*	0.63*	0.59*	0.57*	0.38*	0.46*	0.57*	0.63*	0.54*	0.65*	0.63*	0.62*	0.65*	0.37*	0.57*	0.63*	0.53*
BQ	0.68*	0.65*	1.00*	0.54*	0.66*	0.62*	0.62*	0.69*	0.32*	0.37*	0.71*	0.66*	0.66*	0.67*	0.65*	0.65*	0.70*	0.26*	0.68*	0.65*	0.57*
BSQ	0.50*	0.52*	0.54*	1.00*	0.53*	0.61*	0.67*	0.52*	0.62*	0.61*	0.52*	0.64*	0.50*	0.61*	0.52*	0.55*	0.66*	0.48*	0.64*	0.53*	0.56*
EQ	0.67*	0.62*	0.66*	0.53*	1.00*	0.69*	0.64*	0.68*	0.21*	0.34*	0.60*	0.63*	0.70*	0.71*	0.69*	0.64*	0.71*	0.26*	0.68*	0.73*	0.67*
GQ	0.57*	0.63*	0.62*	0.61*	0.69*	1.00*	0.70*	0.75*	0.34*	0.46*	0.63*	0.66*	0.65*	0.69*	0.66*	0.71*	0.72*	0.34*	0.67*	0.71*	0.66*
HSQ	0.55*	0.59*	0.62*	0.67*	0.64*	0.70*	1.00*	0.67*	0.49*	0.56*	0.62*	0.64*	0.65*	0.72*	0.68*	0.71*	0.73*	0.44*	0.68*	0.71*	0.69*
JQ	0.61*	0.57*	0.69*	0.52*	0.68*	0.75*	0.67*	1.00*	0.27*	0.39*	0.64*	0.60*	0.72*	0.65*	0.67*	0.71*	0.70*	0.28*	0.69*	0.71*	0.69*
KQ	0.24*	0.38*	0.32*	0.62*	0.21*	0.34*	0.49*	0.27*	1.00*	0.58*	0.35*	0.38*	0.27*	0.36*	0.32*	0.37*	0.40*	0.59*	0.37*	0.27*	0.31*
LEQ	0.33*	0.46*	0.37*	0.61*	0.34*	0.46*	0.56*	0.39*	0.58*	1.00*	0.34*	0.42*	0.32*	0.46*	0.45*	0.48*	0.47*	0.52*	0.42*	0.44*	0.40*
LQ	0.61*	0.57*	0.71*	0.52*	0.60*	0.63*	0.62*	0.64*	0.35*	0.34*	1.00*	0.65*	0.67*	0.66*	0.65*	0.64*	0.68*	0.28*	0.68*	0.64*	0.59*
LSQ	0.59*	0.63*	0.66*	0.64*	0.63*	0.66*	0.64*	0.60*	0.38*	0.42*	0.65*	1.00*	0.61*	0.68*	0.60*	0.61*	0.71*	0.34*	0.66*	0.66*	0.58*
MQ	0.60*	0.54*	0.66*	0.50*	0.70*	0.65*	0.65*	0.72*	0.27*	0.32*	0.67*	0.61*	1.00*	0.66*	0.62*	0.62*	0.71*	0.30*	0.71*	0.66*	0.69*
PQ	0.68*	0.65*	0.67*	0.61*	0.71*	0.69*	0.72*	0.65*	0.36*	0.46*	0.66*	0.68*	0.66*	1.00*	0.71*	0.68*	0.78*	0.33*	0.71*	0.72*	0.66*
QI	0.65*	0.63*	0.65*	0.52*	0.69*	0.66*	0.68*	0.67*	0.32*	0.45*	0.65*	0.60*	0.62*	0.71*	1.00*	0.68*	0.70*	0.28*	0.69*	0.72*	0.65*
QV1	0.55*	0.62*	0.65*	0.55*	0.64*	0.71*	0.71*	0.71*	0.37*	0.48*	0.64*	0.61*	0.62*	0.68*	0.68*	1.00*	0.70*	0.36*	0.62*	0.70*	0.59*
QV2	0.63*	0.65*	0.70*	0.66*	0.71*	0.72*	0.73*	0.70*	0.40*	0.47*	0.68*	0.71*	0.71*	0.78*	0.70*	0.70*	1.00*	0.30*	0.78*	0.72*	0.68*
RQ	0.20*	0.37*	0.26*	0.48*	0.26*	0.34*	0.44*	0.28*	0.59*	0.52*	0.28*	0.34*	0.30*	0.33*	0.28*	0.36*	0.30*	1.00*	0.27*	0.32*	0.34*
SQ	0.68*	0.57*	0.68*	0.64*	0.68*	0.67*	0.68*	0.69*	0.37*	0.42*	0.68*	0.66*	0.71*	0.71*	0.69*	0.62*	0.78*	0.27*	1.00*	0.71*	0.69*
TQ	0.64*	0.63*	0.65*	0.53*	0.73*	0.71*	0.71*	0.71*	0.27*	0.44*	0.64*	0.66*	0.66*	0.72*	0.72*	0.70*	0.72*	0.32*	0.71*	1.00*	0.67*
YQ	0.57*	0.53*	0.57*	0.56*	0.67*	0.66*	0.69*	0.69*	0.31*	0.40*	0.59*	0.58*	0.69*	0.66*	0.65*	0.59*	0.68*	0.34*	0.69*	0.67*	1.00*

TABLE C.17: $PC2_F$ Profile correlation between all performances (whole piece)

	ABQ	AQ	BQ	BSQ	EQ	GQ	HSQ	JQ	KQ	LEQ	LQ	LSQ	MQ	PQ	QI	QV1	QV2	RQ	SQ	TQ	YQ
ABQ	1.00*	0.54*	0.48*	0.36*	0.57*	0.51*	0.45*	0.50*	0.32*	0.35*	0.58*	0.55*	0.50*	0.48*	0.51*	0.58*	0.53*	0.27*	0.48*	0.53*	0.46*
AQ	0.54*	1.00*	0.51*	0.42*	0.65*	0.61*	0.51*	0.50*	0.47*	0.50*	0.59*	0.65*	0.50*	0.53*	0.54*	0.60*	0.55*	0.36*	0.51*	0.52*	0.57*
BQ	0.48*	0.51*	1.00*	0.49*	0.52*	0.53*	0.46*	0.52*	0.41*	0.34*	0.48*	0.53*	0.51*	0.51*	0.54*	0.51*	0.55*	0.26*	0.42*	0.56*	0.45*
BSQ	0.36*	0.42*	0.49*	1.00*	0.41*	0.45*	0.43*	0.42*	0.43*	0.36*	0.39*	0.34*	0.39*	0.42*	0.36*	0.46*	0.40*	0.35*	0.41*	0.40*	0.42*
EQ	0.57*	0.65*	0.52*	0.41*	1.00*	0.64*	0.54*	0.52*	0.38*	0.39*	0.56*	0.60*	0.56*	0.60*	0.53*	0.64*	0.59*	0.31*	0.59*	0.63*	0.55*
GQ	0.51*	0.61*	0.53*	0.45*	0.64*	1.00*	0.52*	0.54*	0.27*	0.37*	0.51*	0.59*	0.56*	0.57*	0.53*	0.56*	0.52*	0.31*	0.58*	0.57*	0.64*
HSQ	0.45*	0.51*	0.46*	0.43*	0.54*	0.52*	1.00*	0.50*	0.36*	0.32*	0.52*	0.54*	0.54*	0.46*	0.46*	0.58*	0.57*	0.29*	0.50*	0.52*	0.47*
JQ	0.50*	0.50*	0.52*	0.42*	0.52*	0.54*	0.50*	1.00*	0.30*	0.35*	0.48*	0.51*	0.47*	0.44*	0.55*	0.54*	0.54*	0.22*	0.50*	0.47*	0.56*
KQ	0.32*	0.47*	0.41*	0.43*	0.38*	0.27*	0.36*	0.30*	1.00*	0.31*	0.37*	0.37*	0.39*	0.34*	0.35*	0.34*	0.42*	0.34*	0.19*	0.39*	0.28*
LEQ	0.35*	0.50*	0.34*	0.36*	0.39*	0.37*	0.32*	0.35*	0.31*	1.00*	0.39*	0.34*	0.31*	0.35*	0.34*	0.36*	0.37*	0.37*	0.36*	0.32*	0.37*
LQ	0.58*	0.59*	0.48*	0.39*	0.56*	0.51*	0.52*	0.48*	0.37*	0.39*	1.00*	0.54*	0.51*	0.45*	0.59*	0.55*	0.50*	0.33*	0.49*	0.53*	0.51*
LSQ	0.55*	0.65*	0.53*	0.34*	0.60*	0.59*	0.54*	0.51*	0.37*	0.34*	0.54*	1.00*	0.55*	0.55*	0.53*	0.54*	0.56*	0.23*	0.52*	0.58*	0.54*
MQ	0.50*	0.50*	0.51*	0.39*	0.56*	0.56*	0.54*	0.47*	0.39*	0.31*	0.51*	0.55*	1.00*	0.45*	0.47*	0.55*	0.52*	0.19*	0.49*	0.59*	0.48*
PQ	0.48*	0.53*	0.51*	0.42*	0.60*	0.57*	0.46*	0.44*	0.34*	0.35*	0.45*	0.55*	0.45*	1.00*	0.52*	0.55*	0.48*	0.19*	0.53*	0.53*	0.54*
QI	0.51*	0.54*	0.54*	0.36*	0.53*	0.53*	0.46*	0.55*	0.35*	0.34*	0.59*	0.53*	0.47*	0.52*	1.00*	0.54*	0.53*	0.33*	0.47*	0.52*	0.54*
QV1	0.58*	0.60*	0.51*	0.46*	0.64*	0.56*	0.58*	0.54*	0.34*	0.36*	0.55*	0.54*	0.55*	0.55*	0.54*	1.00*	0.54*	0.33*	0.54*	0.56*	0.52*
QV2	0.53*	0.55*	0.55*	0.40*	0.59*	0.52*	0.57*	0.54*	0.42*	0.37*	0.50*	0.56*	0.52*	0.48*	0.53*	0.54*	1.00*	0.31*	0.49*	0.61*	0.45*
RQ	0.27*	0.36*	0.26*	0.35*	0.31*	0.31*	0.29*	0.22*	0.34*	0.37*	0.33*	0.23*	0.19*	0.19*	0.33*	0.33*	0.31*	1.00*	0.28*	0.34*	0.26*
SQ	0.48*	0.51*	0.42*	0.41*	0.59*	0.58*	0.50*	0.50*	0.19*	0.36*	0.49*	0.52*	0.49*	0.53*	0.47*	0.54*	0.49*	0.28*	1.00*	0.50*	0.63*
TQ	0.53*	0.52*	0.56*	0.40*	0.63*	0.57*	0.52*	0.47*	0.39*	0.32*	0.53*	0.58*	0.59*	0.53*	0.52*	0.56*	0.61*	0.34*	0.50*	1.00*	0.51*
YQ	0.46*	0.57*	0.45*	0.42*	0.55*	0.64*	0.47*	0.56*	0.28*	0.37*	0.51*	0.54*	0.48*	0.54*	0.54*	0.52*	0.45*	0.26*	0.63*	0.51*	1.00*

TABLE C.18: $PC3_F$ Profile correlation between all performances (whole piece)

	ABQ	AQ	BQ	BSQ	EQ	GQ	HSQ	JQ	KQ	LEQ	LQ	LSQ	MQ	PQ	QI	QV1	QV2	RQ	SQ	TQ	YQ
ABQ	1.00*	0.38*	0.47*	0.20*	0.51*	0.57*	0.48*	0.46*	0.09*	0.01	0.51*	0.62*	0.57*	0.46*	0.50*	0.55*	0.47*	0.13*	0.56*	0.57*	0.43*
AQ	0.38*	1.00*	0.36*	0.32*	0.49*	0.46*	0.41*	0.40*	0.23*	0.18*	0.45*	0.51*	0.41*	0.42*	0.45*	0.45*	0.42*	0.21*	0.37*	0.40*	0.36*
BQ	0.47*	0.36*	1.00*	0.33*	0.46*	0.51*	0.44*	0.42*	0.28*	0.04	0.44*	0.50*	0.51*	0.42*	0.49*	0.41*	0.49*	0.15*	0.44*	0.54*	0.38*
BSQ	0.20*	0.32*	0.33*	1.00*	0.35*	0.39*	0.40*	0.36*	0.40*	0.17*	0.33*	0.27*	0.31*	0.29*	0.34*	0.33*	0.40*	0.23*	0.31*	0.33*	0.37*
EQ	0.51*	0.49*	0.46*	0.35*	1.00*	0.64*	0.51*	0.52*	0.36*	0.10*	0.49*	0.59*	0.52*	0.45*	0.50*	0.52*	0.52*	0.24*	0.51*	0.59*	0.51*
GQ	0.57*	0.46*	0.51*	0.39*	0.64*	1.00*	0.59*	0.57*	0.28*	0.09*	0.53*	0.61*	0.61*	0.56*	0.60*	0.55*	0.57*	0.18*	0.59*	0.66*	0.54*
HSQ	0.48*	0.41*	0.44*	0.40*	0.51*	0.59*	1.00*	0.45*	0.29*	0.10*	0.54*	0.52*	0.57*	0.47*	0.48*	0.57*	0.51*	0.20*	0.55*	0.56*	0.41*
JQ	0.46*	0.40*	0.42*	0.36*	0.52*	0.57*	0.45*	1.00*	0.26*	0.19*	0.42*	0.47*	0.46*	0.39*	0.51*	0.47*	0.48*	0.27*	0.47*	0.47*	0.50*
KQ	0.09*	0.23*	0.28*	0.40*	0.36*	0.28*	0.29*	0.26*	1.00*	0.11*	0.16*	0.19*	0.27*	0.27*	0.27*	0.21*	0.29*	0.22*	0.20*	0.33*	0.27*
LEQ	0.01	0.18*	0.04	0.17*	0.10*	0.09*	0.10*	0.19*	0.11*	1.00*	0.07*	0.10*	0.03	0.07*	0.08*	0.12*	0.09*	0.31*	0.01	0.05	0.10*
LQ	0.51*	0.45*	0.44*	0.33*	0.49*	0.53*	0.54*	0.42*	0.16*	0.07*	1.00*	0.63*	0.58*	0.45*	0.54*	0.58*	0.50*	0.20*	0.53*	0.55*	0.36*
LSQ	0.62*	0.51*	0.50*	0.27*	0.59*	0.61*	0.52*	0.47*	0.19*	0.10*	0.63*	1.00*	0.65*	0.45*	0.51*	0.59*	0.56*	0.21*	0.61*	0.66*	0.41*
MQ	0.57*	0.41*	0.51*	0.31*	0.52*	0.61*	0.57*	0.46*	0.27*	0.03	0.58*	0.65*	1.00*	0.54*	0.55*	0.52*	0.49*	0.12*	0.67*	0.68*	0.41*
PQ	0.46*	0.42*	0.42*	0.29*	0.45*	0.56*	0.47*	0.39*	0.27*	0.07*	0.45*	0.45*	0.54*	1.00*	0.57*	0.43*	0.40*	0.08*	0.47*	0.57*	0.39*
QI	0.50*	0.45*	0.49*	0.34*	0.50*	0.60*	0.48*	0.51*	0.27*	0.08*	0.54*	0.51*	0.55*	0.57*	1.00*	0.51*	0.51*	0.22*	0.51*	0.57*	0.43*
QV1	0.55*	0.45*	0.41*	0.33*	0.52*	0.55*	0.57*	0.47*	0.21*	0.12*	0.58*	0.59*	0.52*	0.43*	0.51*	1.00*	0.51*	0.30*	0.51*	0.51*	0.33*
QV2	0.47*	0.42*	0.49*	0.40*	0.52*	0.57*	0.51*	0.48*	0.29*	0.09*	0.50*	0.56*	0.49*	0.40*	0.51*	0.51*	1.00*	0.12*	0.49*	0.53*	0.42*
RQ	0.13*	0.21*	0.15*	0.23*	0.24*	0.18*	0.20*	0.27*	0.22*	0.31*	0.20*	0.21*	0.12*	0.08*	0.22*	0.30*	0.12*	1.00*	0.11*	0.14*	0.14*
SQ	0.56*	0.37*	0.44*	0.31*	0.51*	0.59*	0.55*	0.47*	0.20*	0.01	0.53*	0.61*	0.67*	0.47*	0.51*	0.51*	0.49*	0.11*	1.00*	0.62*	0.45*
TQ	0.57*	0.40*	0.54*	0.33*	0.59*	0.66*	0.56*	0.47*	0.33*	0.05	0.55*	0.66*	0.68*	0.57*	0.57*	0.51*	0.53*	0.14*	0.62*	1.00*	0.46*
YQ	0.43*	0.36*	0.38*	0.37*	0.51*	0.54*	0.41*	0.50*	0.27*	0.10*	0.36*	0.41*	0.41*	0.39*	0.43*	0.33*	0.42*	0.14*	0.45*	0.46*	1.00*

TABLE C.19: $PC4_F$ Profile correlation between all performances (whole piece)

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
1	+1.00*	+0.93*	−0.27	−0.30	−0.14	+0.18	−0.14	−0.05	+0.02	−0.01	−0.23	+0.30	+0.50*	+0.23	+0.45
2	+0.93*	+1.00*	−0.19	−0.24	−0.13	+0.29	−0.12	+0.06	+0.11	−0.07	−0.29	+0.25	+0.43	+0.22	+0.56*
3	−0.27	−0.19	+1.00*	+0.96*	+0.40	−0.36	+0.06	+0.12	−0.04	+0.36	−0.23	+0.42	+0.14	−0.29	+0.08
4	−0.30	−0.24	+0.96*	+1.00*	+0.38	−0.31	+0.03	+0.19	−0.11	+0.30	−0.24	+0.37	+0.02	−0.30	+0.07
5	−0.14	−0.13	+0.40	+0.38	+1.00*	+0.03	+0.02	−0.23	+0.28	+0.04	−0.10	+0.20	−0.01	−0.08	−0.32
6	+0.18	+0.29	−0.36	−0.31	+0.03	+1.00*	+0.29	+0.25	+0.03	−0.19	+0.13	−0.19	−0.10	+0.05	−0.16
7	−0.14	−0.12	+0.06	+0.03	+0.02	+0.29	+1.00*	+0.33	−0.57*	+0.49	−0.03	+0.21	+0.20	−0.03	−0.38
8	−0.05	+0.06	+0.12	+0.19	−0.23	+0.25	+0.33	+1.00*	−0.61*	+0.22	−0.11	+0.26	+0.18	+0.00	+0.28
9	+0.02	+0.11	−0.04	−0.11	+0.28	+0.03	−0.57*	−0.61*	+1.00*	−0.63*	+0.24	−0.50	−0.32	+0.07	−0.09
10	−0.01	−0.07	+0.36	+0.30	+0.04	−0.19	+0.49	+0.22	−0.63*	+1.00*	−0.01	+0.71*	+0.58*	−0.25	+0.14
11	−0.23	−0.29	−0.23	−0.24	−0.10	+0.13	−0.03	−0.11	+0.24	−0.01	+1.00*	−0.43	−0.27	−0.23	−0.28
12	+0.30	+0.25	+0.42	+0.37	+0.20	−0.19	+0.21	+0.26	−0.50	+0.71*	−0.43	+1.00*	+0.79*	−0.18	+0.33
13	+0.50*	+0.43	+0.14	+0.02	−0.01	−0.10	+0.20	+0.18	−0.32	+0.58*	−0.27	+0.79*	+1.00*	+0.23	+0.28
14	+0.23	+0.22	−0.29	−0.30	−0.08	+0.05	−0.03	+0.00	+0.07	−0.25	−0.23	−0.18	+0.23	+1.00*	−0.01
15	+0.45	+0.56*	+0.08	+0.07	−0.32	−0.16	−0.38	+0.28	−0.09	+0.14	−0.28	+0.33	+0.28	−0.01	+1.00*

TABLE C.20: Correlations between overall observations

Index	***Feature***
1	Rec.Y.
2	Fo.Y.
3	$\overline{BPM}$
4	BPM_{Hist}
5	Δ_{BPM}
6	$\sigma_{IOI}/\overline{IOI}$
7	f_{A4}
8	$\overline{v_{1770}}$
9	$\sigma_{v_{1770}}$
10	$\overline{v_{SC}}$
11	$\sigma_{v_{SC}}$
12	$\overline{v_{MFC2}}$
13	$\sigma_{v_{MFC2}}$
14	$\overline{v_{MFC3}}$
15	$\sigma_{v_{MFC3}}$

TABLE C.21: Labels for Indices of Table C.20

Bibliography

[Aar06] Aarden, Bret: How the Timing Between Notes Can Impact Musical Meaning. In: *Proc. of the 9th International Conference on Music Perception and Cognition (ICMPC)*. Bologna, August 2006

[ACKM04] Arifi, Vlora ; Clausen, Michael ; Kurth, Frank ; Müller, Meinard: Score-PCM Music Synchronization based on Extracted Score Parameters. In: *Proc. of the 2nd International Symposium on Computer Music Modeling and Retrieval (CMMR)*. Trondheim, June 2004

[App08] Apple Computer, Inc.: *Audio Units*. Version: 2008. `http://developer.apple.com/audio/audiounits.html`, Retrieved on: 2008 July 8th

[Ari02] Arifi, Vlora: *Algorithmen zur Synchronisation von Musikdateien im Partitur-, MIDI- und PCM-Format*. Bonn, Rheinische Friedrich-Wilhelms-Universität, Dissertation, 2002

[ASA60] ASA S1.1-1960: Acoustical Terminology / American Standards Association (ASA). 1960. – Standard

[B+08] Bencina, Ross et al.: *PortAudio – portable cross-platform Audio API*. Version: 2008. `http://www.portaudio.com`, Retrieved on: 2008 April 30th

[Bac94] Bach, Carl Philipp Emanuel: *Versuch über die wahre Art das Clavier zu spielen*. Kassel : Bärenreiter, 1994. – Facsimile Reprint of the publications in 1753 and 1762, edited by Wolfgang Horn

[BBZ90] Baird, Bridget ; Blevins, Donald ; Zahler, Noel: The Artificially Intelligent Computer Performer: The Second Generation. In: *Interface – Journal of New Music Research* 19 (1990), pp. 197–204

[BBZ93] Baird, Bridget ; Blevins, Donald ; Zahler, Noel: Artificial Intelligence and Music: Implementing an Interactive Computer Performer. In: *Computer Music Journal* 17 (1993), No. 2, pp. 73–79

[BD85] Bloch, Joshua J. ; Dannenberg, Roger B.: Real-Time Computer Accompaniment of Keyboard Performances. In: *Proc. of the International Computer Music Conference (ICMC)*. Vancouver, October 1985

[Ben02] Benjamin, Eric: Comparison of Objective Measures of Loudness Using Audio Program Material. In: *Proc. of the 113rd AES Convention, Preprint 5703*. Los Angeles : Audio Engineering Society, October 2002

[Bis74a] Bismarck, Gottfried von: Sharpness as an Attribute of the Timbre of Steady Sounds. In: *Acustica* 30 (1974), pp. 159–172

[Bis74b] Bismarck, Gottfried von: Timbre of Steady Sounds: A Factorial Investigation of its Verbal Attributes. In: *Acustica* 30 (1974), pp. 146–159

[BL03] Burred, Juan José ; Lerch, Alexander: A hierarchical approach to automatic musical genre classification. In: *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFx)*. London, September 2003

[BL04] Burred, Juan José ; Lerch, Alexander: Hierarchical Automatic Audio Signal Classification. In: *Journal of the Audio Engineering Society (JAES)* 52 (2004), No. 7/8, pp. 724–739

[BM99] Beran, Jan ; Mazzola, Guerino: Analyzing Musical Structure and Performance – A Statistical Approach. In: *Statistical Science* 14 (1999), No. 1, pp. 47–79

[Bow06] Bowman Macleod, Rebecca: *Influences of Dynamic Level and Pitch Height on the Vibrato Rates and Widths of Violin and Viola Players*. Tallahassee, Florida State University, College of Music, Dissertation, 2006

[Bre94] Bregman, Albert S.: *Auditory Scene Analysis*. MIT Press, 1994. – 1st Paperback Edition

[Bre00] Bresin, Roberto: *Virtual Virtuosity – Studies in Automatic Music Performance*. Stockholm, Royal Institute of Technology (KTH), Dissertation, 2000

[BW01] Bartsch, Mark A. ; Wakefield, Gregory H.: To Catch a Chorus: Using Chroma-Based Representations for Audio Thumbnailing. In: *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. New Paltz, October 2001

[Cla91] Clarke, Eric F.: Expression and Communication in Musical Performance. In: Sundberg, Johan (Ed.) ; L., Nord (Ed.) ; Carlson, R. (Ed.): *Music, Language, Speech and Brain*. London : Macmillan Press, 1991

[Cla02a] Clarke, Eric: Listening to performance. In: Rink, John (Ed.): *Musical Performance – A Guide to Understanding*. Cambridge : Cambridge University Press, 2002. – Fifth printing 2006

[Cla02b] Clarke, Eric: Understanding the psychology of performance. In: Rink, John (Ed.): *Musical Performance – A Guide to Understanding*. Cambridge : Cambridge University Press, 2002. – Fifth printing 2006

[Cla04] Clarke, Eric: Empirical Methods in the Study of Performance. In: Clarke, Eric (Ed.) ; Cook, Nicholas (Ed.): *Empirical Musicology*. Oxford : Oxford University Press, 2004

[CLB99] Cano, Pedro ; Loscos, Alex ; Bonada, Jordi: Score-Performance Matching using HMMs. In: *Proc. of the International Computer Music Conference (ICMC)*. Beijing, October 1999

[CLSB06] Cannam, Chris ; Landone, Christian ; Sandler, Mark ; Bello, Juan Pablo: The Sonic Visualiser: A Visualisation Platform for Semantic Descriptors from Musical Signals. In: *Proc. of the 7th International Conference on Music Information Retrieval (ISMIR)*. Victoria, October 2006

[CMSW05] Caclin, Anne ; McAdams, Stephen ; Smith, Bennet K. ; Winsberg, Suzanne: Acoustic correlates of timbre space dimensions: A confirmatory study using synthetic tones. In: *Journal of the Acoustical Society of America (JASA)* 118 (2005), No. 1, pp. 471–482

[Con06] Cont, Arshia: Realtime Audio to Score Alignment for Polyphonic Music Instruments using Sparse Non-Negative Constraints and Hierarchical HMMs. In: *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* Vol. 5. Toulouse, May 2006

[CW86] Clynes, Manfred ; Walker, Janice: Music As Time's Measure. In: *Music Perception* 4 (1986), No. 1, pp. 85–120

[Dah00] Dahl, Sofia: The Playing of an Accent – Preliminary Observations from Temporal and Kinematic Analysis of Percussionists. In: *Journal of New Music Research* 29 (2000), No. 3, pp. 225–233

[Dan84] Dannenberg, Roger B.: An On-Line Algorithm for Real-Time Accompaniment. In: *Proc. of the International Computer Music Conference (ICMC)*. Paris, 1984

[Dan06] Dannenberg, Roger B.: The Interpretation of MIDI Velocity. In: *Proc. of the International Computer Music Conference (ICMC)*. New Orleans, November 2006

[DBDS03] Duxbury, Chris ; Bello, Juan Pablo ; Davies, Mike ; Sandler, Mark: Complex Domain Onset Detection for Musical Signals. In: *Proc. of the 6th International Conference on Digital Audio Effects (DAFx)*. London, October 2003

[DG02] Dixon, Simon ; Goebl, Werner: Pinpointing the Beat: Tapping to Expressive Performances. In: *Proc. of the 7th International Conference on Music Perception and Cognition (ICMPC)*. Sydney, July 2002

[DGC06] Dixon, Simon ; Goebl, Werner ; Cambouropoulos, Emilios: Perceptual Smoothness of Tempo in Expressively Performed Music. In: *Music Perception* 23 (2006), No. 3, pp. 195–214

[DGW02] Dixon, Simon ; Goebl, Werner ; Widmer, Gerhard: The Performance Worm: Real Time Visualisation of Expression based on Langner's Tempo Loudness Animation. In: *Proc. of the International Computer Music Conference (ICMC)*. Göteborg, September 2002

[DH93] Desain, Peter ; Honing, Henkjan: Tempo Curves Considered Harmful. In: *Time in Contemporary Musical Thought, Contemporary Music Review* 7 (1993), No. 2, pp. 123–138

[DH94] Desain, Peter ; Honing, Henkjan: Does expressive timing in music performance scale proportionally with tempo? In: *Psychological Research* 56 (1994), pp. 285–292

[DH03] Dannenberg, Roger B. ; Hu, Ning: Polyphonic Audio Matching for Score Following and Intelligent Audio Editors. In: *Proc. of the International Computer Music Conference (ICMC)*. Singapore, September/October 2003

[DHH97] Desain, Peter ; Honing, Henkjan ; Heijink, Hank: Robust Score-Performance Matching: Taking Advantage of Structural Information. In: *Proc. of the International Computer Music*

Conference (ICMC). Thessaloniki : International Computer Music Association, September 1997

[Dil01] Dillon, Roberto: Extracting audio cues in real time to understand musical expressiveness. In: *Proc. of the MOSART workshop*. Barcelona, November 2001

[Dil03] Dillon, Roberto: A Statistical Approach to Expressive Intention Recognition in Violin Performances. In: *Proc. of the Stockholm Music Acoustics Conference (SMAC)*. Stockholm, August 2003

[Dil04] Dillon, Roberto: *On the Recognition of Expressive Intention in Music Playing: A Computational Approach with Experiments and Applications*. Genoa, University of Genoa, Faculty of Engineering, Dissertation, 2004

[DIN91] DIN 45631:1991: Berechnung des Lautstärkepegels und der Lautheit aus dem Geräuschspektrum / DIN. 1991. – Standard

[DIN94] DIN IEC 60248-10:1994: Elektroakustische Geräte – Teil 10: Spitzenspannungs-Aussteuerungsmeßgerät / DIN. 1994. – Standard

[Dix96] Dixon, Simon: A Dynamic Modelling Approach to Music Recognition. In: *Proc. of the International Computer Music Conference (ICMC)*. Hong Kong, August 1996

[Dix99] Dixon, Simon: A Beat Tracking System for Audio Signals. In: *Proc. of the Conference on Mathematical and Computational Methods in Music*. Vienna, December 1999

[DM88] Dannenberg, Roger B. ; Mukaino, Hirofumi: New Techniques for Enhanced Quality of Computer Accompaniment. In: *Proc. of the International Computer Music Conference (ICMC)*. Cologne, September 1988

[Dor42] Dorian, Frederick: *The History of Music in Performance – The Art of Musical Interpretation from the Rennaisance to Our Day*. New York : W. W. Norton & Company Inc, 1942

[Dov95] Dovey, Matthew J.: Analysis of Rachmaninoff's Piano Performances Using Inductive Logic Programming (Extended Abstract). In: *Proc. of the 8th European Conference on Machine Learning (ECML)*. Heraclion, September 1995, pp. 279–282

[DP00] Drake, Carolyn ; Palmer, Caroline: Skill acquisition in music performance: relations between planning and temporal control. In: *Cognition* 74 (2000), pp. 1–32

[DP04] Dalla Bella, Simone ; Palmer, Caroline: Tempo and Dynamics in Piano Performance: The role of movement amplitude. In: *Proc. of the 8th International Conference on Music Perception & Cognition (ICMPC)*. Evanston, August 2004

[DP06] Drake, Carolyn ; Palmer, Caroline: Cross-modal interaction in the perception of musical performance. In: *Cognition* 101 (2006), pp. 80–113

[DW05] Dixon, Simon ; Widmer, Gerhard: MATCH: A Music Alignment Tool Chest. In: *Proc. of the 6th International Conference on Music Information Retrieval (ISMIR)*. London, September 2005

[FBS06] Friberg, Anders ; Bresin, Roberto ; Sundberg, Johan: Overview of the KTH rule system for musical performance. In: *Advances in Cognitive Psychology, Special Issue on Music Performance* 2 (2006), No. 2-3, pp. 145–161

[FJP03] Faraone, G. ; Johansson, S. ; Polotti, Pietro: The Influence of the Practice of Basso Continuo on the intonation of a Professional Singer in the Time of Monteverdi. In: *Proc. of the Stockholm Music Acoustics Conference (SMAC)*. Stockholm, August 2003

[Fle75] Fletcher, Neville H.: Acoustical correlates of flute performance technique. In: *Journal of the Acoustical Society of America (JASA)* 57 (1975), No. 1, pp. 233–237

[FP03] Finney, Steven A. ; Palmer, Caroline: Auditory feedback and memory for music performance: Some evidence for an encoding effect. In: *Memory & Cognition* 31 (2003), No. 1, pp. 51–64

[Fre08] Free Software Foundation: *GNU General Public License*. Version: 2008. `http://www.gnu.org/licenses/gpl.html`, Retrieved on: 2008 July 8th

[FS92] Friberg, Anders ; Sundberg, Johan: Perception of just noticeable time displacement of a tone presented in a Metrical Sequence at Different Tempos. In: *STL-QPSR* 33 (1992), No. 4, pp. 97–108

[Fur08] Furse, Richard: *LADSPA*. Version: 2008. `http://www.ladpsa.org`, Retrieved on: 2008 July 8th

[Gab99] Gabrielsson, Alf: The Performance of Music. In: Deutsch, Diana (Ed.): *The Psychology of Music*. 2nd Edition. San Diego : Academic Press, 1999

[GB01] Goebl, Werner ; Bresin, Roberto: Are computer-controlled pianos a reliable tool in music performance research? Recording

and reproduction precision of a Yamaha Disklavier grand piano. In: *Proc. of the MOSART workshop*. Barcelona, November 2001

[GB03] Goebl, Werner ; Bresin, Roberto: Measurement and reproduction accuracy of computer-controlled grand pianos. In: *Journal of the Acoustical Society of America (JASA)* 114 (2003), No. 4, pp. 2273–2283

[GD97] Grubb, Lorin ; Dannenberg, Roger B.: A Stochastic Method of Tracking a Vocal Performer. In: *Proc. of the International Computer Music Conference (ICMC)*. Thessaloniki, September 1997

[GD98] Grubb, Lorin ; Dannenberg, Roger B.: Enhanced Vocal Performance Tracking Using Multiple Information Sources. In: *Proc. of the International Computer Music Conference (ICMC)*. Ann Arbor, October 1998

[GD01] Goebl, Werner ; Dixon, Simon: Analysis of tempo classes in performances of Mozart sonatas. In: Lappalainen, H. (Ed.): *Proc. of the 7th International Symposium on Systematic and Comparative Musicology (ISSCM), 3rd International Conference on Cognitive Musicology (ICCM)*. Jyväskyla, June 2001

[GDP+05] Goebl, Werner ; Dixon, Simon ; Poli, Giovanni De ; Friberg, Anders ; Bresin, Roberto ; Widmer, Gerhard: Sense in Expressive Music Performance: Data Acquisition, Computational Studies, and Models. In: Leman, Marc (Ed.) ; Cirotteau, Damien (Ed.): *Sound to Sense, Sense to Sound: A State-of-the-Art*. Version 0.1. Logos Berlin, November 2005

[Ger95] Geringer, John M.: Continuous Loudness Judgments of Dynamics in Recorded Music Excerpts. In: *Journal of Research in Music Education (JRME)* 43 (1995), Spring, No. 1, pp. 22–35

[GM95] Goto, Masataka ; Muraoka, Yoichi: Music Understanding At The Beat Level – Real-time Beat Tracking For Audio Signals. In: *Proc. of the Workshop on Computational Auditory Scene Analysis (IJCAI)*, 1995

[Goe01] Goebl, Werner: Melody lead in piano performance: Expressive device or artifact? In: *Journal of the Acoustical Society of America (JASA)* 110 (2001), No. 1, pp. 563–572

[Goe02] Goebl, Werner: The influence of relative intensity on the perception of onset asynchronies. In: Stevens, C. (Ed.) ; Burnham, D. (Ed.) ; McPherson, G. (Ed.) ; Schubert, E. (Ed.) ; Renwick,

J. (Ed.): *Proc. of the 7th International Conference on Music Perception and Cognition (ICMPC)*. Sydney, July 2002

[Gor84] Gordon, John W.: *Perception of Attack Transients in Musical Tones*. Stanford, Stanford University, Center for Computer Research in Music and Acoustics (CCRMA), Dissertation, 1984

[Got01] Goto, Masataka: An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds. In: *Journal of New Music Research* 30 (2001), No. 2, pp. 159–171

[Got04] Goto, Masataka: A Real-time Music-scene-description System: Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-world Audio Signals. In: *Speech Communication* 43 (2004), No. 4, pp. 311–329

[GP01] Goebl, Werner ; Parncutt, Richard: Perception of onset asynchronies: Acoustic Piano versus Synchronized complex versus pure tones. In: *Meeting of the Society for Music Perception and Cognition (SMPC)*. Kingston, August 2001

[GP08] Goebl, Werner ; Palmer, Caroline: Tactile feedback and timing accuracy in piano performance. In: *Experimental Brain Research* 186 (2008), No. 3, pp. 471–479

[Gre77] Grey, John M.: Multidimensional scaling of musical timbres. In: *Journal of the Acoustical Society of America (JASA)* 61 (1977), No. 5, pp. 1270–1277

[Gsc04] Gscheidle, Ralf: Virtuelle Simulation Sinfonischer Musik. In: *Proc. of the VDT International Audio Convention (23. Tonmeistertagung)*. Leipzig, November 2004

[Har32] Hartmann, Artur: Untersuchungen über das metrische Verhalten in musikalischen Interpretationsvarianten. In: *Archiv für die gesamte Psychologie* 84 (1932), pp. 103–192

[HDT03] Hu, Ning ; Dannenberg, Roger B. ; Tzanetakis, George: Polyphonic Audio Matching and Alignment for Music Retrieval. In: *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. New Paltz, October 2003

[Hei96] Heijink, Hank: *Matching Scores and Performances*, Nijmegen University, M.A. Thesis, June 1996

[Hel70] Helmholtz, Hermann von: *Die Lehre von den Tonempfindungen als physiologische Grundlage für die Theorie der Musik*. 3rd Edition. Braunschweig : Vieweg, 1870

[HHF96] Hoshishiba, Takayuki ; Horiguchi, Susumu ; Fujinaga, Ichiro: Study of Expression and Individuality in Music Performance Using Normative Data Derived from MIDI Recordings of Piano Music. In: *Proc. of the International Conference on Music Perception and Cognition (ICMPC)*. Montreal, August 1996

[Hil02] Hill, Peter: From score to sound. In: Rink, John (Ed.): *Musical Performance – A Guide to Understanding*. Cambridge : Cambridge University Press, 2002. – Fifth printing 2006

[Hir59] Hirsh, Ira J.: Auditory Perception of Temporal Order. In: *Journal of the Acoustical Society of America (JASA)* 31 (1959), No. 6, pp. 759–767

[HM03] Hainsworth, Stephen ; Macleod, Malcolm: Onset Detection in Musical Audio Signals. In: *Proc. of the International Computer Music Conference (ICMC)*. Singapore, September/October 2003

[HMK08] Hashida, Mitsuyo ; Matsui, Toshie ; Katayose, Haruhiro: A New Music Database Describing Deviation Information of Performance Expressions. In: *Proc. of the 9th International Conference on Music Information Retrieval (ISMIR)*. Philadelphia, September 2008

[Hon90] Honing, Henkjan: POCO: An Environment for Analysing, Modifying, and Generating Expression in Music. In: *Proc. of the International Computer Music Conference (ICMC)*. Glasgow, September 1990

[Hon06] Hong, Ju-Lee: Motor Action in Performance – Rostropovich and Richter's Repeated Renditions of Prokofiev's Cello Sonata Op.119. In: *Proc. of the Digital Music Research Network Conference (DMRN)*. London, 2006

[HS05] Harte, Christopher A. ; Sandler, Mark B.: Automatic Chord Identification Using a Quantised Chromagram. In: *Proc. of the 118th AES Convention, Preprint 6412*. Barcelona : Audio Engineering Society, May 2005

[HTS02] Husain, Gabriela ; Thompson, William F. ; Schellenberg, E. Glenn: Effects of Musical Tempo and Mode on Arousal, Mood, and Spatial Abilities. In: *Music Perception* 20 (2002), No. 2, pp. 151–171

[IK93] Iverson, Paul ; Krumhansl, Carol L.: Isolating the dynamic attributes of musical timbre. In: *Journal of the Acoustical Society of America (JASA)* 94 (1993), No. 5, pp. 2595–2603

[ISO75] ISO 16:1975: Acoustics – Standard tuning frequency (Standard musical pitch) / ISO. 1975. – Standard

[ISO02] ISO/IEC JTC1/SC29 15938-4:2002: Information technology – Multimedia content description interface – Part 4: Audio / ISO/IEC. 2002. – Standard

[ISO03] ISO/IEC JTC1/SC22 14882:2003: Programming languages – C++ / ISO/IEC. 2003. – Standard

[ITU01] ITU-R BS.1387:2006: Method for objective measurements of perceived audio quality / ITU. 2001. – Recommendation

[ITU06] ITU-R BS.1770:2006: Algorithms to measure audio programme loudness and true-peak audio level / ITU. 2006. – Recommendation

[Izm05] Izmirli, Özgür: Template based key finding from audio. In: *Proc. of the International Computer Music Conference (ICMC)*. Barcelona, September 2005

[JCMJ06] Jensen, Jesper H. ; Christensen, Mads G. ; Murthi, Manohar N. ; Jensen, Søren H.: Evaluation of MFCC estimation techniques for music similarity. In: *Proc. of the XIV. European Signal Processing Conference (EUSIPCO)*. Florence, September 2006

[Jer03a] Jerkert, Jesper: *Measurements and Models of Musical Articulation*, DTH, Department of Speech, Music and Hearing, Master's Thesis, October 2003

[Jer03b] Jerkert, Jesper: Measurements and Models of Musical Articulation. In: *Proc. of the Stockholm Music Acoustics Conference (SMAC)*. Stockholm, August 2003

[Jer04] Jerkert, Jesper: Music Articulation in the Organ. In: *Proc. of Joint Baltic-Nordic Acoustics Meeting*. Mariehamn, June 2004

[Joh02] Johnson, Peter: The legacy of recordings. In: Rink, John (Ed.): *Musical Performance – A Guide to Understanding*. Cambridge : Cambridge University Press, 2002. – Fifth printing 2006

[Jus00] Juslin, Patrick N.: Cue Utilization of Emotion in Music Performance: Relating Performance to Perception. In: *Journal of Experimental Psychology* 26 (2000), No. 6, pp. 1797–1813

[Jus03a] Juslin, Patrik N.: Five myths about expressivity in music performance and what to do about them. In: *Proc. of the International Conference on Arts and Humanities*. Honululu, Hawaii, January 2003

[Jus03b] Juslin, Patrik N.: Studies of Music Performance: A Theoretical Analysis of Empirical Findings. In: *Proc. of the Stockholm Music Acoustics Conference (SMAC)*. Stockholm, August 2003

[Kan06] Kantor-Martynuska, Joanna: Emotion-relevant characteristics of temperament and the perceived magnitude of tempo and loudness of music. In: *Proc. of the 9th International Conference on Music Perception and Cognition (ICMPC)*. Bologna, August 2006

[KC90] Kendall, Roger A. ; Carterette, Edward C.: The Communication of Musical Expression. In: *Music Perception* 8 (1990), No. 2, pp. 129–164

[Kla03] Klapuri, Anssi P.: Musical Meter Estimation and Music Transcription. In: *Proc. of the Cambridge Music Processing Colloquium*. Cambridge, 2003

[Kli04] Klich, Ingmar: *Automatische Erkennung von Onsets in Musiksignalen zur Steuerung von Beattracking-Systemen*, Technische Universität Berlin, M.A. Thesis, August 2004

[Kru90] Krumhansl, Carol L.: *Cognitive Foundations of Musical Pitch*. New York : Oxford University Press, 1990

[Kru96] Krumhansl, Carol L.: A Perceptual Analysis Of Mozart's Piano Sonata K. 282: Segmentation, Tension and Musical Ideas. In: *Music Perception* 13 (1996), No. 3, pp. 401–432

[Lak00] Lakatos, Stephen: A common perceptual space for harmonic and percussive timbres. In: *Perception & Psychophysics* 62 (2000), No. 7, pp. 1426–1439

[Lap00] Lapidaki, Eleni: Stability of Tempo Perception in Music Listening. In: *Music Education Research* 2 (2000), No. 1, pp. 25–44

[Lar93] Large, Edward W.: Dynamic programming for the analysis of serial behaviors. In: *Behavior Research Methods, Instruments, and Computers* 25 (1993), No. 2, pp. 238–241

[Lar03] Laroche, Jean: Efficient Tempo and Beat Tracking in Audio Recordings. In: *Journal of the Audio Engineering Society (JAES)* 51 (2003), No. 4, pp. 226–233

[LDR04] Levau, Pierre ; Daudet, Laurent ; Richard, Gaël: Methodology and Tools for the Evaluation of Automatic Onset Detection Algorithms in Music. In: *Proc. of the 5th International Conference on Music Information Retrieval (ISMIR)*. Barcelona, October 2004

[Ler04] Lerch, Alexander: Ein Ansatz zur automatischen Erkennung der Tonart in Musikdateien. In: *Proc. of the VDT International Audio Convention (23. Tonmeistertagung)*. Leipzig, November 2004

[Ler06] Lerch, Alexander: On the Requirement of Automatic Tuning Frequency Estimation. In: *Proc. of the 7th International Conference on Music Information Retrieval (ISMIR)*. Victoria, October 2006

[Ler08] Lerch, Alexander: Bitratenreduktion. In: Weinzierl, Stefan (Ed.): *Handbuch der Audiotechnik*. Berlin : Springer, 2008, pp. 849–884

[LET05] Lerch, Alexander ; Eisenberg, Gunnar ; Tanghe, Koen: FEAPI: A Low Level Feature Extraction Plugin API. In: *Proc. of 8th Int Conference on Digital Audio Effects (DAFx'05)*. Madrid, September 2005

[LET08] Lerch, Alexander ; Eisenberg, Gunnar ; Tanghe, Koen: *FEAPI*. Version: 2008. `http://www.sf.net/projects/feapi`, Retrieved on: 2008 July 8th

[LG02] Langner, Jörg ; Goebl, Werner: Representing expressive performance in tempo-loudness space. In: *Proc. of the 10th Anniversary Conference European Society for the cognitive sciences of Music (ESCOM)*. Liege, April 2002

[LKSW00] Langner, Jörg ; Kopiez, Reinhard ; Stoffel, Christian ; Wilz, Martin: Realtime Analysis of Dynamik Shaping. In: *Proc. of the 6th International Conference on Music Perception and Cognition (ICMPC)*. Keele, August 2000

[LM07] Lagrange, Matthieu ; Marchand, Sylvain: Estimating the Instantaneous Frequency of Sinusoidal Components Using Phase-Based Methods. In: *Journal of the Audio Engineering Society (JAES)* 55 (2007), No. 5, pp. 385–399

[Lun53] Lundin, Robert W.: *An Objective Psychology of Music*. New York : The Ronald Press Company, 1953

[MAG08] Molina-Solana, Miguel ; Arcos, Josep Lluís ; Gomez, Emilia: Using Expressive Trends for Identifying Violin Recordings. In: *Proc. of the 9th International Conference on Music Information Retrieval (ISMIR)*. Philadelphia, September 2008

[MB03] McKinney, Martin F. ; Breebart, Jeroen: Features for Audio and Music Classification. In: *Proc. of the 4th International Conference on Music Information Retrieval (ISMIR)*. Baltimore, October 2003

[MCMW03] Marozeau, Jeremy ; Cheveigné, Alain de ; McAdams, Stephen ; Winsberg, Suzanne: The dependency of timbre on fundamental frequency. In: *Journal of the Acoustical Society of America (JASA)* 114 (2003), No. 5, pp. 2946–2957

[Meu02] Meudic, Benoit: A Causal Algorithm for Beat Tracking. In: *Proc. of the 2nd International Conference on Understanding and Creating Music*. Caserta, November 2002

[Mey56] Meyer, Leonard B.: *Emotion and Meaning in Music*. Chicago : University of Chicago Press, 1956

[MGB97] Moore, Brian C.J. ; Glasberg, Brian R. ; Baer, Thomas: A Model for the Prediction of Thresholds, Loudness and Partial Loudness. In: *Journal of the Audio Engineering Society (JAES)* 45 (1997)

[MH01] Meron, Yoram ; Hirose, Keikichi: Automatic alignment of a musical score to performed music. In: *Acoustical Science & Technology* 22 (2001), No. 3, pp. 189–198

[MID01] MIDI Manufacturers Association: Complete MIDI 1.0 Detailed Specification V96.1, 2nd edition / MMA. 2001. – Standard

[MKR04] Müller, Meinard ; Kurth, Frank ; Röder, Tido: Towards an Efficient Algorithm for Automatic Score-to-Audio Synchronization. In: *Proc. of the 5th International Conference on Music Information Retrieval (ISMIR)*. Barcelona, October 2004

[Moo97] Moore, Brian C.J.: *An Introduction to the Psychology of Hearing*. 4th Edition. London : Academic Press, 1997

[MP03] Meyer, Rosalee K. ; Palmer, Caroline: Temporal and Motor Transfer in Music Performance. In: *Music Perception* 21 (2003), No. 1, pp. 81–104

[MWD+95] McAdams, Stephen ; Winsberg, Suzanne ; Donnadieu, Sophie ; Soete, Geert De ; Krimphoff, Jochen: Perceptual scaling of synthesized musical timbres: common dimensions, specifities, and latent subject classes. In: *Psychological Research* 58 (1995), pp. 177–192

[MWK08] Maempel, Hans-Joachim ; Weinzierl, Stefan ; Kaminski, Peter: Audiobearbeitung. In: Weinzierl, Stefan (Ed.): *Handbuch der Audiotechnik*. Berlin : Springer, 2008, pp. 719–784

[MZ94] Mazzola, Guerino ; Zahorka, Oliver: The RUBATO Workstation for Musical Analysis and Performance. In: *Proc. of the 3rd International Conference on Music Perception and Cognition (ICMPC)*. Liege, July 1994

[Nak87] Nakamura, Toshie: The communication of dynamics between musicians and listeners through musical performance. In: *Perception & Psychophysics* 41 (1987), No. 6, pp. 525–533

[NLSK02] Nair, Dinesh G. ; Large, Edward W. ; Steinberg, Fred ; Kelso, J.A. S.: Perceiving Emotion in Expressive Performance: A Functional MRI Study. In: Stevens, C. (Ed.) ; Burnham, D. (Ed.) ; McPherson, G. (Ed.) ; Schubert, E. (Ed.) ; Renwick, J. (Ed.): *Proc. of the 7th International Conference on Music Perception and Cognition (ICMPC)*. Sydney, July 2002

[OD01] Orio, Nicola ; Déchelle, Francois: Score Following Using Spectral Analysis and Hidden Markov Models. In: *Proc. of the International Computer Music Conference (ICMC)*. Habana, September 2001

[OLS03] Orio, Nicola ; Lemouton, Serge ; Schwarz, Diemo: Score Following: State of the Art and New Developments. In: *Proc. of the Conference of New Interfaces for Musical Expression (NIME)*. Montreal, May 2003

[Ope08] Open Source Initiative: *BSD License*. Version: 2008. `http://www.opensource.org/licenses/bsd-license.php`, Retrieved on: 2008 July 8th

[Orn07] Ornoy, Eitan: An empirical study of intonation in performances of J.S. Bach's Sarabandes: temperament, 'melodic charge' and 'melodic intonation'. In: *Orbis MusicÆ* 14 (2007), pp. 37–76

[OS01] Orio, Nicola ; Schwarz, Diemo: Alignment of Monophonic and Polyphonic Music to a Score. In: *Proc. of the International Computer Music Conference (ICMC)*. Habana, September 2001

[Pal89] Palmer, Caroline: Mapping Musical Thoughts to Musical Performance. In: *Journal of Experimental Psychology: Human Perception and Performance* 15 (1989), No. 2, pp. 331–346

[Pal97] Palmer, Caroline: Music Performance. In: *Annual Review of Psychology* 48 (1997)

[Pal00] PALMER, Caroline: Conceptual and Motor Learning in Music Performance. In: *Psychological Science* 11 (2000), pp. 63–68

[Pal06] PALMER, Caroline: Nature of memory for music performance skills. In: ALTENMÜLLER, Eckart (Ed.) ; WIESENDANGER, Mario (Ed.) ; KESSELRING, Jurg (Ed.): *Music, Motor Control and the Brain*. Oxford : Oxford University Press, 2006, pp. 39–53

[Par03] PARNCUTT, Richard: Accents and expression in piano performance. In: NIEMÖLLER, Klaus W. (Ed.): *Perspektiven und Methoden einer Systemischen Musikwissenschaft*. Frankfurt/Main : Peter Lang, 2003

[PB91] PALMER, Caroline ; BROWN, Judith C.: Investigations in the amplitude of piano tones. In: *Journal of the Acoustical Society of America (JASA)* 90 (1991), No. 1, pp. 60–66

[PDW03] PAMPALK, Elias ; DIXON, Simon ; WIDMER, Gerhard: On the Evaluation of Perceptual Similarity Measures for Music. In: *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFx)*. London, September 2003

[Pee05] PEETERS, Geoffrey: Time variable tempo detection and beat marking. In: *Proc. of the International Computer Music Conference (ICMC)*. Barcelona, September 2005

[Pfo05] PFORDRESHER, Peter Q.: Auditory Feedback in Music Performance: The Role of Melodic Structure and Musical Skill. In: *Journal of Experimental Psychology* 31 (2005), No. 6, pp. 1331–1345

[PFW05] PAMPALK, Elias ; FLEXER, Arthur ; WIDMER, Gerhard: Improvements of audio-based music similarity and genre classification. In: *Proc. of the 6th International Conference on Music Information Retrieval (ISMIR)*. London, September 2005

[PL92] PUCKETTE, Miller ; LIPPE, Cort: Score Following in Practice. In: *Proc. of the International Computer Music Conference (ICMC)*. San Francisco, 1992

[Pov77] POVEL, Dirk-Jan: Temporal Structure of Performed Music. Some Preliminary Observations. In: *Acta Psychologica* Vol. 41, 1977, pp. 309–320

[PP01] PFEIFFER, Silvia ; PARKER, Conrad: bewdy, Maaate! In: *Presentation at the Australian Linux Conference*. Sydney, January 2001

[PP02] Pfordresher, Peter Q. ; Palmer, Caroline: Effects of delayed auditory feedback on timing of music performance. In: *Psychological Research* 16 (2002), pp. 71–79

[Puc95] Puckette, Miller: Score Following using the sung voice. In: *Proc. of the International Computer Music Conference (ICMC)*. Banff, 1995

[Rap99] Raphael, Christopher: Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 21 (1999), No. 4, pp. 360–370

[Rap01] Raphael, Christopher: A Probabilistic Expert System for Automatic Musical Accompaniment. In: *Journal of Computational and Graphical Statistics* 10 (2001), No. 3, pp. 487–512

[Rap04] Raphael, Christopher: A Hybrid Graphical Model for Aligning Polyphonic Audio with Musical Scores. In: *Proc. of the 5th International Conference on Music Information Retrieval (ISMIR)*. Barcelona, October 2004

[Rap07] Rapoport, Eliezer: The Marvels of the Human Voice: Poem-Melody-Vocal Performance. In: *Orbis MusicÆ* 14 (2007), pp. 7–36

[Ras79] Rasch, Rudolf A.: Synchronization in Performed Ensemble Music. In: *Acustica* 43 (1979), pp. 121–131

[RB93] Riedmiller, Martin ; Braun, Heinrich: A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: *Proc. of the IEEE International Conference on Neural Networks*. San Francisco, March/April 1993

[Rep90] Repp, Bruno H.: Patterns of Expressive Timing in Performances of a Beethoven Minuet by Nineteen Famous Pianists. In: *Journal of the Acoustical Society of America (JASA)* 88 (1990), No. 2, pp. 622–641

[Rep92] Repp, Bruno H.: Diversity and commonality in music performance: An analysis of timing microstructure in Schumann's 'Träumerei'. In: *Journal of the Acoustical Society of America (JASA)* 92 (1992), No. 5, pp. 2546–2568

[Rep94] Repp, Bruno H.: On Determining the Basic Tempo of an Expressive Music Performance. In: *Psychology of Music* 22 (1994), No. 2, pp. 157–167

[Rep95] Repp, Bruno H.: Quantitative Effects of Global Tempo on Expressive Timing in Music Performance: Some Perceptual Evidence. In: *Music Perception* 13 (1995), No. 3, pp. 39–57

[Rep96a] Repp, Bruno H.: Patterns of note onset asynchronies in expressive piano performance. In: *Journal of the Acoustical Society of America (JASA)* 100 (1996), No. 6, pp. 3917–3932

[Rep96b] Repp, Bruno H.: Pedal Timing and Tempo in Expressive Piano Performance: A Preliminary Investigation. In: *Psychology of Music* 24 (1996), No. 2, pp. 199–221

[Rep96c] Repp, Bruno H.: The Art of Inaccuracy: Why Pianists' Errors are Difficult to Hear. In: *Music Perception* 14 (1996), No. 2, pp. 161–184

[Rep96d] Repp, Bruno H.: The dynamics of expressive piano performance: Schumann's 'Träumerei' revisited. In: *Journal of the Acoustical Society of America (JASA)* 100 (1996), No. 1, pp. 641–650

[Rep97a] Repp, Bruno H.: Acoustics, Perception, and Production of Legato Articulation on a Computer-Controlled Grand Piano. In: *Journal of the Acoustical Society of America (JASA)* 102 (1997), No. 3, pp. 1878–1890

[Rep97b] Repp, Bruno H.: Expressive Timing in a Debussy Prelude: A Comparison of Student and Expert Pianists. In: *Musicae Scientiae* 1 (1997), No. 2, pp. 257–268

[Rep97c] Repp, Bruno H.: The effect of tempo on pedal timing in piano performance. In: *Psychological Research* 60 (1997), No. 3, pp. 164–172

[Rep98] Repp, Bruno H.: A microcosm of musical expression. I. Quantitative analysis of pianists' timing in the initial measures of Chopin's Etude in E major. In: *Journal of the Acoustical Society of America (JASA)* 104 (1998), No. 2, pp. 1085–1100

[Rep99a] Repp, Bruno H.: A microcosm of musical expression. II. Quantitative analysis of pianists' dynamics in the initial measures of Chopin's Etude in E major. In: *Journal of the Acoustical Society of America (JASA)* 105 (1999), No. 3, pp. 1972–1988

[Rep99b] Repp, Bruno H.: A microcosm of musical expression. III. Contributions of timing and dynamics to the aesthetic impression of pianists' performances of the initial measures of Chopin's Etude in E Major. In: *Journal of the Acoustical Society of America (JASA)* 106 (1999), No. 1, pp. 469–478

[Rep99c] Repp, Bruno H.: Detecting deviations from metronomic timing in music: Effects of perceptual structure on the mental timekeeper. In: *Perception & Psychophysics* 61 (1999), No. 3, pp. 529–548

[Rep99d] Repp, Bruno H.: Effects of Auditory Feedback Deprivation on Expressive Piano Performance. In: *Music Perception* 16 (1999), No. 4, pp. 409–438

[Reu95] Reuter, Christoph: *Der Einschwingvorgang nichtperkussiver Musikinstrumente*. Frankfurt : Peter Lang, 1995

[Rin02] Rink, John: Analysis and (or?) performance. In: Rink, John (Ed.): *Musical Performance – A Guide to Understanding*. Cambridge : Cambridge University Press, 2002. – Fifth printing 2006

[RJ93] Rabiner, Lawrence ; Juang, Biin-Hwang: *Fundamentals of Speech Recognition*. New Jersey : Prentice Hall, 1993

[RP82] Rasch, Rudolf A. ; Plomb, Reinier: The Perception of Musical Tones. In: Deutsch, Diana (Ed.): *The Psychology of Music*. New York : Academic Press, 1982

[RPK08] Ramirez, Rafael ; Perez, Alfonso ; Kersten, Stefan: Performer Identification in Celtic Violin Recordings. In: *Proc. of the 9th International Conference on Music Information Retrieval (ISMIR)*. Philadelphia, September 2008

[Ryy04] Ryynänen, Matti: *Probabilistic Modelling of Note Events in the Transcription of Monophonic Melodies*, Tampere University of Technology, Master's Thesis, March 2004

[S+08] Spitzak, Bill et al.: *FLTK – Fast Light Toolkit*. Version: 2008. `http://www.fltk.org`, Retrieved on: 2008 July 8th

[SAF83] Sundberg, Johan ; Askenfelt, Anders ; Frydén, Lars: Musical Performance: A Synthesis-by-Rule Approach. In: *Computer Music Journal* 7 (1983), No. 1, pp. 37–43

[SAH79] Schroeder, M. R. ; Atal, B. S. ; Hall, J. L.: Optimizing digital speech coders by exploiting masking properties of the human ear. In: *Journal of the Acoustical Society of America (JASA)* 66 (1979), No. 6, pp. 1647–1652

[San08] Sandell, Greg: *Definitions of the word ´Timbre´*. Version: 2008. `http://www.zainea.com/timbre.htm`, Retrieved on: 2008 April 12th

[Sap07] Sapp, Craig Stuart: Comparative Analysis of Multiple Musical Performances. In: *Proc. of the 8th International Conference on Music Information Retrieval (ISMIR)*. Wien, September 2007

[Sap08] Sapp, Craig Stuart: Hybrid Numeric/Rank Similarity Metrics for Musical Performance Analysis. In: *Proc. of the 9th International Conference on Music Information Retrieval (ISMIR)*. Philadelphia, September 2008

[Sch40] Schoen, Max: *The Psychology of Music*. New York : The Ronald Press Company, 1940

[Sch85] Schloss, W. A.: *On the Automatic Transcription of Percussive Music – From Acoustic Signal to High-Level Analysis*. Stanford, Stanford University, Center for Computer Research in Music and Acoustics (CCRMA), Dissertation, 1985

[Sch95] Scheirer, Eric D.: *Extracting Expressive Performance Information from Recorded Music*, Massachusetts Institute of Technology, Master's Thesis, September 1995

[Sch98] Scheirer, Eric D.: Tempo and beat analysis of acoustic musical signals. In: *Journal of the Acoustical Society of America (JASA)* 103 (1998), No. 1, pp. 588–601

[Sch03a] Scherer, Klaus R.: Why Music does not Produce Basic Emotions: Pleading for a new Approach to Measuring the Emotional Effects of Music. In: *Proc. of the Stockholm Music Acoustics Conference (SMAC)*. Stockholm, August 2003

[Sch03b] Schwarz, Diemo: Requirements for Music Notation regarding Music-to-Score Alignment and Score Following / IRCAM. Paris, 2003. – Tech Report. – available online (8/2004): http://www.interactivemusicnetwork.org/mpeg-ahg/mn-req-alignment.pdf

[Sch04] Schubert, Emery: Modeling Perceived Emotion With Continuous Musical Features. In: *Music Perception* 21 (2004), No. 4, pp. 561–585

[Sea02] Seashore, Carl E.: A Voice Tonoscope. In: *Studies in Psychology* 3 (1902), pp. 18–28

[Sea38] Seashore, Carl E.: *Psychology of Music*. New York : McGraw-Hill, 1938. – Nachdruck 1967, Dover Publications, New York

[Seg06] Segnini Sequera, Rodrigo: Timbrescape: a Musical Timbre and Structure Visualization Method using Tristimulus Data. In:

Proc. of the 9th International Conference on Music Perception and Cognition (ICMPC). Bologna, August 2006

[SG84] Smith, Julius O. ; Gosset, Phil: A flexible sampling-rate conversion method. In: *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* Vol. 2. San Diego, March 1984

[Sha84] Shaffer, L. Henry: Timing in Solo and Duet Piano Performances. In: *The Quarterly Journal of Experimental Psychology* 36A (1984), pp. 577–595

[Shi01] Shifres, Favio: Three expressive strategies of expert performance of an excerpt by J.S. Bach. In: *MikroPolyphonie* Vol. 17, 2001

[SL01] Sloboda, John A. ; Lehmann, Andreas C.: Tracking Performance Correlates of Changes in Perceived Intensity of Emotion During Different Interpretation of a Chopin Piano Prelude. In: *Music Perception* 19 (2001), No. 1, pp. 87–120

[Sla98] Slaney, Malcolm: Auditory Toolbox – Version 2 / Interval Research Corporation. Wien, 1998 (TR-1998-010). – Tech Report. – available online (5/2008): http://cobweb.ecn.purdue.edu/~malcolm/interval/1998-010/

[Slo82] Sloboda, John A.: Music Performance. In: Deutsch, Diana (Ed.): *The Psychology of Music*. New York : Academic Press, 1982

[Slo85] Sloboda, John A.: *The Musical Mind – The cognitive psychology of music*. Oxford : Oxford University Press, 1985 (Oxford Psychology Series 5). – Reprinted 2004

[SN03] Soulodre, Gilbert A. ; Norcross, Scott G.: Objective Measures of Loudness. In: *Proc. of the 115th AES Convention, Preprint 5896*. New York : Audio Engineering Society, October 2003

[SRS03] Soulez, Ferréol ; Rodet, Xavier ; Schwarz, Diemo: Improving polyphonic and poly-instrumental music to score alignment. In: *Proc. of the 4th International Conference on Music Information Retrieval (ISMIR)*. Baltimore, October 2003

[SSKS04] Shalev-Shwartz, Shai ; Keshet, Joseph ; Singer, Yoram: Learning to Align Polyphonic Music. In: *Proc. of the 5th International Conference on Music Information Retrieval (ISMIR)*. Barcelona, October 2004

[Sta01] Stamatatos, Efstathios: A Computational Model for Discriminating Music Performers. In: *Proc. of the MOSART workshop*. Barcelona, November 2001

[Sta08] Stanford University: *SndLib*. Version: 2008. `http://ccrma.stanford.edu/software/snd/sndlib/`, Retrieved on: 2008 April 30th

[Ste08] Steinberg AG: *Virtual Studio Technology*. Version: 2008. `http://ygrabit.steinberg.de`, Retrieved on: 2008 July 8th

[Sto04] Stolla, Jochen: *Abbild und Autonomie – Zur Klangbildgestaltung bei Aufnahmen klassischer Musik 1950–1994*. Marburg : Tectum Verlag, 2004

[Stu90] Stumpf, Carl: *Tonpsychologie II*. Hilversum / Amsterdam : Knuf and Bonset, 1890. – Reprint 1965

[SW06] Schubert, Emery ; Wolfe, Joe: Does Timbral Brightness Scale with Frequency and Spectral Centroid. In: *Acustica* 92 (2006), pp. 820–825

[TADH00] Timmers, Renee ; Ashley, Richard ; Desain, Peter ; Heijink, Hank: The influence of musical context on tempo rubato. In: *Journal of New Music Research* 29 (2000), No. 2

[Tag03] Taguti, Tomoyasu: Mapping a Physical Correlate of Loudness into the Velocity Space of MIDI-Controlled Piano Tones. In: *Proc. of the Stockholm Music Acoustics Conference (SMAC)*. Stockholm, August 2003

[TC00] Tzanetakis, George ; Cook, Perry: MARSYAS: A Framework for Audio Analysis. In: *Organised Sound* 4 (2000), No. 3

[TC02] Tzanetakis, George ; Cook, Perry: Musical Genre Classification of Audio Signals. In: *IEEE Trans. on Speech and Audio Processing* 10 (2002), No. 5, pp. 293–302

[TCV03] Timmers, Renee ; Camurri, Antonio ; Volpe, Gualtiero: Performance Cues for Listeners' Emotional Engagement. In: *Proc. of the Stockholm Music Acoustics Conference (SMAC)*. Stockholm, August 2003

[TE03] Turetsky, Robert J. ; Ellis, Daniel P.W.: Ground-Truth Transcriptions of Real Music from Force-Aligned MIDI Syntheses. In: *Proc. of the 4th International Conference on Music Information Retrieval (ISMIR)*. Baltimore, October 2003

[TEC02] Tzanetakis, George ; Ermolinskyi, Andrey ; Cook, Perry: Pitch Histograms in Audio and Symbolic Music Information Retrieval. In: *Proc. of the 3rd International Conference on Music Information Retrieval (ISMIR).* Paris, October 2002

[Ter79] Terhardt, E.: Calculating Virtual Pitch. In: *Hearing Research* 1 (1979), pp. 155–182

[TH02] Timmers, Renee ; Honing, Henkjan: On music performance, theories, measurement and diversity. In: *Cognitive Processing* 1-2 (2002)

[Tim01] Timmers, Renee: Context-sensitive evaluation of expression. In: *Proc. of the Workshop on Current Research Directions in Computer Music.* Barcelona, November 2001

[Tim05] Timmers, Renee: Predicting the similarity between expressive performances of music from measurements of tempo and dynamics. In: *Journal of the Acoustical Society of America (JASA)* 117 (2005), No. 1

[Tim07a] Timmers, Renee: Communication of (e)motion through performance: Two case studies. In: *Orbis MusicÆ* 14 (2007), pp. 116–140

[Tim07b] Timmers, Renee: Perception of music performance on historical and modern commercial recordings. In: *Journal of the Acoustical Society of America (JASA)* 122 (2007), No. 5

[TMCV06] Timmers, Renee ; Marolt, Matija ; Camurri, Antonio ; Volpe, Gualtiero: Listeners' emotional engagement with performances of a Scriabin étude: an explorative case study. In: *Psychology of Music* 34 (2006), No. 4, pp. 481–510

[Tod92] Todd, Neil P. McAngus: The Dynamics of Dynamics: A Model of Musical Expression. In: *Journal of the Acoustical Society of America (JASA)* 91 (1992), No. 6, pp. 3540–3550

[Tod93] Todd, Neil P. McAngus: Vestibular Feedback in Musical Performance. In: *Music Perception* 10 (1993), No. 3, pp. 379–382

[Tod95] Todd, Neil P. McAngus: The Kinematics of Musical Expression. In: *Journal of the Acoustical Society of America (JASA)* 97 (1995), No. 3, pp. 1940–1949

[TWV07] Thompson, Sam ; Williamson, Aaron ; Valentine, Elizabeth: Time-Dependent Characteristics of Performance Evaluation. In: *Music Perception* 25 (2007), No. 1, pp. 13–29

[Van95] VANTOMME, Jason D.: Score Following by Temporal Pattern. In: *Computer Music Journal* 19 (1995), No. 3, pp. 50–59

[Ver84] VERCOE, Barry: The Synthetic Performer in the Context of Live Performance. In: *Proc. of the International Computer Music Conference (ICMC)*. Paris, 1984, pp. 199–200

[Vit67] VITERBI, Andrew J.: Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. In: *IEEE Trans. on Information Theory* 13 (1967), pp. 260–269

[VP85] VERCOE, Barry ; PUCKETTE, Miller: Synthetic Rehearsal: Training the Synthetic Performer. In: *Proc. of the International Computer Music Conference (ICMC)*. Vancouver, 1985, pp. 275–278

[WAD+01] WINDSOR, W. Luke ; AARTS, Rinus ; DESAIN, Peter ; HEIJINK, Hank ; TIMMERS, Renee: The timing of grace notes in skilled musical performance at different tempi: a preliminary case study. In: *Psychology of Music* 29 (2001), pp. 149–169

[Wal02] WALLS, Peter: Historical performance and the modern performer. In: RINK, John (Ed.): *Musical Performance – A Guide to Understanding*. Cambridge : Cambridge University Press, 2002. – Fifth printing 2006

[Wal04] WALKER, Timothy M.: *Instrumental Difference in Characteristics of expressive musical performance*. Columbus, The Ohio State University, Dissertation, 2004

[WDPB06] WINDSOR, W. Luke ; DESAIN, Peter ; PENEL, Amandine ; BORKENT, Michiel: A structurally guided method for the decomposition of expression in music performance. In: *Journal of the Acoustical Society of America (JASA)* 119 (2006), No. 2, pp. 1182–1193

[Wes78] WESSEL, David L.: Low Dimensional Control of Musical Timbre. In: *Proc. of the 59th AES Convention, Preprint 1337*. Hamburg : Audio Engineering Society, February/March 1978

[WF02] WEINZIERL, Stefan ; FRANKE, Christoph: 'Lotte, ein Schwindel!' – Geschichte und Praxis des Musikschnitts am Beispiel von Beethovens 9. Symphonie. In: *Proc. of the VDT International Audio Convention (22. Tonmeistertagung)*. Hannover, November 2002

[Wid95a] WIDMER, Gerhard: A Machine Learning Analysis of Expressive Timing in Pianists' Performances of Schumann's 'Träumerei'. In:

Proc. of the Stockholm Symposium on Generative Grammars for Music Performance. Stockholm, May 1995

[Wid95b] WIDMER, Gerhard: Modeling the Rational Basis of Musical Expression. In: *Computer Music Journal* 19 (1995), No. 2, pp. 76–96

[Wid98a] WIDMER, Gerhard: Applications of Machine Learning to Music Research: Empirical Investigations into the Phenomenon of Musical Expression. In: MICHALSKI, Ryszad S. (Ed.) ; BRATKO, Ivan (Ed.) ; KUBAT, Miroslav (Ed.): *Machine Learning, Data Mining, and Knowledge Discovery: Methods and Applications*. Chichester, United Kingdom : Wiley, 1998, pp. 269–293

[Wid98b] WIDMER, Gerhard: In Search of the Horowitz Factor: Interim Report on a Musical Discovery Project. In: MICHALSKI, Ryszad S. (Ed.) ; BRATKO, Ivan (Ed.) ; KUBAT, Miroslav (Ed.): *Proc. of the 5th International Conference on Discovery Science (DS)*. London, UK : Springer-Verlag, 1998, pp. 13–21

[Wid02] WIDMER, Gerhard: Machine Discoveries: A Few Simple, Robust Local Expression Principles. In: *Journal of New Music Research* 31 (2002), No. 1, pp. 37–50

[Wöl04] WÖLLNER, Clemens: Expressive Timing and intensity profiles in mental performances. In: *Proc. of the 8th International Conference on Music Perception & Cognition (ICMPC)*. Evanston, August 2004

[WPD06] WOODRUFF, John ; PARDO, Bryan ; DANNENBERG, Roger B.: Remixing Stereo Music with Score-Informed Source Separation. In: *Proc. of the 7th International Conference on Music Information Retrieval (ISMIR)*. Victoria, October 2006

[Wri08] WRIGHT, Matthew J.: *The Shape of an Instant: Measuring and Modeling Perceptual Attack Time with Probability Density Functions*. Stanford, Stanford University, Dissertation, 2008

[WT03] WIDMER, Gerhard ; TOBUDIC, Asmir: Playing Mozart by Analogy: Learning Multi-level Timing and Dynamics Strategies. In: *Journal of New Music Research* 32 (2003), No. 3, pp. 259–268

[WZ04] WIDMER, Gerhard ; ZANON, Patrick: Automatic Recognition of Famous Artists by Machine / Österreichisches Forschungsinstitut für Artificial Intelligence (ÖFAI). Wien, 2004 (TR-2004-04). – Tech Report. – available online (8/2004): http://www.ai.univie.ac.at/cgi-bin/tr-online?number+2004-04

[ZF67] Zwicker, Eberhard ; Feldtkeller, Richard: *Das Ohr als Nachrichtenempfänger*. 2nd Edition. Stuttgart : S. Hirzel Verlag, 1967

[ZF99] Zwicker, Eberhard ; Fastl, Hugo: *Psychoacoustics. Facts and Models*. 2nd Edition. Springer, 1999

[ZKG05] Zhu, Yongweil ; Kankanhalli, Mohan S. ; Gao, Sheng: Music Key Detection for Musical Audio. In: *Proc. of the 11th International Multimedia Modelling Conference*. Melbourne, January 2005

[Zöl97] Zölzer, Udo: *Digitale Audiosignalverarbeitung*. 2nd Edition. Stuttgart : Teubner, 1997

[ZW03] Zanon, Patrick ; Widmer, Gerhard: Learning to Recognize Famous Pianists with Machine Learning Techniques. In: *Proc. of the Stockholm Music Acoustics Conference (SMAC)*. Stockholm, August 2003