Automated Methods for Audio-Based Music Analysis with Applications to Musicology

Verena Konz

Department of Computer Science
Saarland University
66123 Saarbrücken, Germany

Dissertation zur Erlangung des Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
der Naturwissenschaftlich-Technischen Fakultät I
der Universität des Saarlandes, May 31, 2012

Betreuender Hochschullehrer / Supervisor:

Prof. Dr. Meinard Müller
Universität des Saarlandes & MPI Informatik, Saarbrücken, Germany

Gutachter / Reviewers:

Prof. Dr. rer. nat. Meinard Müller
Universität des Saarlandes & MPI Informatik, Saarbrücken, Germany

Prof. Dr. rer. nat. Hans-Peter Seidel
MPI Informatik, Saarbrücken, Germany

Prof. Dr. phil. Rainer Kleinertz
Universität des Saarlandes, Saarbrücken, Germany

Dekan / Dean:

Univ.-Prof. Mark Groves
Universität des Saarlandes, Saarbrücken, Germany

Verena Konz
Cluster of Excellence, Geb. E1.7
Campus E1.7
66123 Saarbrücken
Germany
vkonz@mpi-inf.mpg.de

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, _____

(Verena Konz)

Acknowledgements

First, I want to thank my supervisor Prof. Dr. Meinard Müller for his support. I would also like to thank Prof. Dr. Hans-Peter Seidel for providing us the excellent research environment. Furthermore, I want to thank Prof. Dr. Rainer Kleinertz for our interdisciplinary collaboration. In this context, I would also like to thank Florian Henri Besthorn and Stephanie Klauk. I am grateful to my former and present colleagues in the Multimedia Information Retrieval and Music Processing Group: Nanzhu Jiang, Thomas Prätzlich, Zhe Zuo, Andi Scharfstein, Jonathan Driedger, Peter Grosche, Andreas Baak, and Thomas Helten. Furthermore, I would like to thank our collaborating colleagues of the Multimedia Signal Processing Group at Bonn University, in particular, Prof. Dr. Michael Clausen, Sebastian Ewert, Christian Fremerey, David Damm, and Verena Thomas. For the collaboration with the Hochschule für Musik Saar I want to thank Prof. Thomas Duis and Wolfgang Bogler. In this context, I thank all music students for the participation in our experiments and in the Disklavier recordings, as well as all people who were involved in the recordings, in particular, Fedele Antonicelli, Dr. Vlora Arifi-Müller, and Philip Thelen. In addition, I thank all people working in the administration, in particular, the secretaries of the Max-Planck-Institut für Informatik Sabine Budde and Ellen Fries, and the people working in the Office of the Cluster of Excellence. Furthermore, I am grateful to my family. Finally, I want to thank Robert for his support.

Abstract

This thesis contributes to bridging the gap between music information retrieval (MIR) and musicology. We present several automated methods for music analysis, which are motivated by concrete application scenarios being of central importance in musicology. In this context, the automated music analysis is performed on the basis of audio material. Here, one reason is that for a given piece of music usually many different recorded performances exist. The availability of multiple versions of a piece of music is exploited in this thesis to stabilize analysis results. We show how the presented automated methods open up new possibilities for supporting musicologists in their work. Furthermore, we introduce novel interdisciplinary concepts which facilitate the collaboration between computer scientists and musicologists. Based on these concepts, we demonstrate how MIR researchers and musicologists may greatly benefit from each other in an interdisciplinary collaboration. Firstly, we present a fully automatic approach for the extraction of tempo parameters from audio recordings and show to which extent this approach may support musicologists in analyzing recorded performances. Secondly, we introduce novel user interfaces which are aimed at encouraging the exchange between computer science and musicology. In this context, we indicate the potential of computer-based methods in music education by testing and evaluating a novel MIR user interface at the University of Music Saarbrücken. Furthermore, we show how a novel multi-perspective user interface allows for interactively viewing and evaluating version-dependent analysis results and opens up new possibilities for interdisciplinary collaborations. Thirdly, we present a cross-version approach for harmonic analysis of audio recordings and demonstrate how this approach enables musicologists to explore harmonic structures even across large music corpora. Here, one simple yet important conceptual contribution is to convert the physical time axis of an audio recording into a performance-independent musical time axis given in bars.

Kurzzusammenfassung

Diese Arbeit trägt dazu bei, die Brücke zwischen der automatisierten Musikverarbeitung und der Musikwissenschaft zu schlagen. Ausgehend von Anwendungen, die in der Musikwissenschaft von zentraler Bedeutung sind, stellen wir verschiedene automatisierte Verfahren vor. Die automatisierte Musikanalyse wird hierbei auf der Basis von Audiodaten durchgeführt. Ein Grund hierfür ist, dass zu einem gegebenen Musikstück üblicherweise viele verschiedene Aufnahmen existieren. Die Verfügbarkeit mehrerer Versionen zu ein und demselben Musikstück wird in dieser Arbeit ausgenutzt, um Analyseresultate zu stabilisieren. Wir demonstrieren, inwieweit die vorgestellten automatisierten Methoden neue Möglichkeiten eröffnen, Musikwissenschaftler in ihrer Arbeit zu unterstützen. Außerdem führen wir neue interdisziplinäre Konzepte ein, die die Kollaboration zwischen Informatikern und Musikwissenschaftlern erleichtern. Auf der Basis dieser Konzepte zeigen wir, dass Informatiker und Musikwissenschaftler im Rahmen einer interdisziplinären Kollaboration erheblich voneinander profitieren können. Erstens stellen wir ein vollautomatisches Verfahren zur Extraktion von Tempoparametern aus Audioaufnahmen vor und zeigen, inwieweit dieses Verfahren Musikwissenschaftler bei der Interpretationsanalyse verschiedener Aufnahmen unterstützen kann. Zweitens führen wir neuartige Benutzerschnittstellen ein, die darauf abzielen, den Austausch zwischen der Informatik und der Musikwissenschaft zu fördern. In diesem Zusammenhang testen und evaluieren wir eine Benutzerschnittstelle an der Hochschule für Musik Saar und deuten auf diese Weise das Potential computer-basierter Methoden im Bereich der Musikerziehung an. Weiterhin stellen wir eine neuartige Benutzerschnittstelle vor, die es auf interaktive Weise ermöglicht, verschiedene Sichtweisen auf versionsabhängige Analyseresultate einzunehmen und diese auszuwerten. Diese Benutzerschnittstelle eröffnet neue Möglichkeiten für interdisziplinäre Kollaborationen. Drittens zeigen wir, wie eine cross-version harmonische Analyse es Musikwissenschaftlern ermöglicht, harmonische Strukturen über riesige musikalische Werkzyklen hinweg zu ergründen. In diesem Zusammenhang ist ein einfacher aber wichtiger konzeptueller Beitrag, die physikalische Zeitachse einer Audioaufnahme in eine versionsunabhängige musikalische Zeitachse gegeben in Takten zu verwandeln.

Summary

This thesis aims to bridge the gap between music information retrieval (MIR) and musicology. We present several automated methods for music analysis, which are motivated by concrete application scenarios being of central importance in musicology. In this context, the automated music analysis is performed on the basis of audio material. Here, one reason is that for a given piece of music usually many different recorded performances exist. The availability of multiple versions of a piece of music is exploited in this thesis to stabilize analysis results. We show how the presented automated methods open up new possibilities for supporting musicologists in their work. Furthermore, we introduce novel interdisciplinary concepts which facilitate the collaboration between computer scientists and musicologists. Based on these concepts, we demonstrate how MIR researchers and musicologists may greatly benefit from each other in an interdisciplinary collaboration.

Firstly, we present a fully automatic approach for the extraction of tempo parameters from audio recordings. Recorded performances of a piece of music are characterized by the individual playing style and the personal interpretation of the performer. The analysis of performance aspects across different recorded performances, which is referred to as performance analysis, constitutes an important task in musicology. Here, the different recorded performances are typically annotated in a manual process, which is prohibitive in view of large audio collections. The fully automatic approach presented in this thesis enables performing the analysis of temporal parameters of recorded performances on an unprecedented scale. In our approach, we exploit score-like MIDI information along with the audio to be analyzed. Using score-audio synchronization techniques, we automatically derive temporal information from the audio recording. This information is given in the form of a tempo curve revealing the relative tempo difference between the audio recording and the MIDI reference on the musically meaningful time axis in bars. As shown by our experiments on harmony-based Western music, our approach allows for capturing the overall tempo flow and, for certain classes of music, even finer expressive tempo nuances. Finally, we demonstrate the potential and the limitations of our automated approach and investigate to which extent it may support musicologists in analyzing recorded performances.

Secondly, we present novel computer-based interfaces which are aimed at encouraging the exchange between computer science and musicology. In this context, we report on an experiment conducted at the University of Music Saarbrücken with the goal of introducing a novel user interface to music education. Here, we not only tested and evaluated our interface in a setting of practical relevance, but also indicated the potential of MIR methods in music education. Furthermore, we introduce various novel functionalities for

a multi-perspective user interface that opens up new possibilities for viewing, interacting, and evaluating version-dependent analysis results. Here, we exploit the fact that for a given piece of music, there typically exist multiple music representations, including different recorded performances or score-like MIDI representations. Our interface then allows a user to interactively generate unifying views of the analysis results across the different available versions. In this way, consistencies and inconsistencies across the version-dependent analysis results can be easily located by the user. The new evaluation and navigation possibilities of this user interface enable interdisciplinary collaborations, where musicologists employ their musical knowledge and trained ear to conveniently evaluate version-dependent analysis results obtained by MIR methods.

Thirdly, we introduce a cross-version approach, which analyzes the harmonic properties of several audio versions synchronously. The computer-based harmonic analysis is referred to as chord labeling and is of central importance in the field of MIR. Chord labeling procedures are typically evaluated on large audio collections by comparing the automatically extracted chord labels to manually generated ground truth annotations. Here, a piece to be analyzed is typically represented by a specific audio recording which possesses version-dependent characteristics. Another major problem arises from the fact that audio-based recognition results refer to the physical time axis in seconds of the considered audio recording, whereas score-based analysis results obtained by music experts refer to a musical time axis given in bars. This simple fact alone often makes it difficult to get musicologists involved in the evaluation process of audio-based chord labeling procedures. The presented cross-version approach for chord labeling aims to overcome the strong dependency of chord labeling results on a specific version. We show that using a cross-version approach stabilizes the chord labeling result in the sense that inconsistencies indicate version-dependent characteristics, whereas consistencies across several versions indicate harmonically stable passages in the piece of music. In particular, we show that consistently labeled passages often correspond to correctly labeled passages. Our experiments document that the cross-version labeling procedure significantly increases the precision of the result while keeping the recall at a relatively high level. Furthermore, we describe how to transform the time axis of analysis results obtained from audio recordings to a common musical time axis given in bars. This not only facilitates a convenient evaluation by a musicologist, but also allows for comparing analysis results across different recorded performances. We introduce a powerful visualization, which reveals the harmonically stable passages on a musical time axis specified in bars, and demonstrate how this cross-version visualization may serve musicologists as a supportive tool for exploring harmonic structures. Finally, analyzing tonal centers across the entire corpus of Beethoven's piano sonatas, we show how a cross-version approach enables large-scale harmonic analyses.

Zusammenfassung

Diese Arbeit trägt dazu bei, die Brücke zwischen der automatisierten Musikverarbeitung und der Musikwissenschaft zu schlagen. Ausgehend von Anwendungen, die in der Musikwissenschaft von zentraler Bedeutung sind, stellen wir verschiedene automatisierte Verfahren vor. Die automatisierte Musikanalyse wird hierbei auf der Basis von Audiodaten durchgeführt. Ein Grund hierfür ist, dass zu einem gegebenen Musikstück üblicherweise viele verschiedene Aufnahmen existieren. Die Verfügbarkeit mehrerer Versionen zu ein und demselben Musikstück wird in dieser Arbeit ausgenutzt, um Analyseresultate zu stabilisieren. Wir demonstrieren, inwieweit die vorgestellten automatisierten Methoden neue Möglichkeiten eröffnen, Musikwissenschaftler in ihrer Arbeit zu unterstützen. Außerdem führen wir neue interdisziplinäre Konzepte ein, die die Kollaboration zwischen Informatikern und Musikwissenschaftlern erleichtern. Auf der Basis dieser Konzepte zeigen wir, wie Informatiker und Musikwissenschaftler im Rahmen einer interdisziplinären Kollaboration erheblich voneinander profitieren können.

Erstens stellen wir ein vollautomatisches Verfahren zur Extraktion von Tempoparametern aus Audioaufnahmen vor. Verschiedene Einspielungen eines Musikstückes unterscheiden sich durch den individuellen Stil und die persönliche Interpretation des Musikers. Die Analyse von Performance-Aspekten, die auch als Interpretationsforschung bezeichnet wird, stellt einen wichtigen Forschungsbereich der Musikwissenschaft dar. In diesem Zusammenhang werden die verschiedenen Aufnahmen eines Musikstückes üblicherweise manuell annotiert, was hinderlich im Hinblick auf grosse Audiodatenbestände ist. Das vollautomatisierte Verfahren, das in dieser Arbeit vorgestellt wird, ermöglicht die Analyse zeitlicher Parameter auf der Basis von Audioaufnahmen in einer bislang nicht möglichen Art und Weise. In unserem Verfahren nutzen wir zu einer vorhandenen Audioaufnahme partiturähnliche MIDI-Information aus. Unter Einsatz von Partitur-Audio Synchronisationstechniken leiten wir automatisiert Tempoinformation aus der Audioaufnahme ab. Diese Information ist als Tempokurve gegeben, die die relativen Tempounterschiede zwischen der Audioaufnahme und der MIDI-Referenz auf einer musikalischen Zeitachse in Takten wiedergibt. Unsere Experimente auf der Basis von harmoniebasierter westlicher Musik zeigen, dass unser Verfahren den globalen Tempoverlauf sowie für bestimmte Klassen von Musik sogar feinere Temponuancen erfassen kann. Abschließend zeigen wir das Potential und die Grenzen unseres automatisierten Verfahrens auf und untersuchen, inwieweit es Musikwissenschaftler bei der Interpretationsanalyse unterstützen kann.

Zweitens stellen wir neuartige computerbasierte Benutzerschnittstellen vor, die darauf ausgerichtet sind, den Austausch zwischen den beiden Gebieten zu fördern. In diesem Zusammenhang berichten wir über ein an der Hochschule für Musik Saar durchgeführtes

Experiment, das darauf abzielte eine neue Benutzerschnittstelle in der Musikerziehung einzuführen. Hierbei haben wir die Benutzerschnittstelle in einem anwendungsrelevanten Umfeld getestet und ausgewertet und darüberhinaus das Potential computerbasierter Methoden in der Musikerziehung angedeutet. Weiterhin stellen wir neuartige Funktionalitäten für eine multi-perspektivische Benutzerschnittstelle vor, die neue Möglichkeiten eröffnet, versionsabhängige Analyseresultate zu betrachten, auszuwerten und mit ihnen zu interagieren. Hierbei nutzen wir die Tatsache aus, dass zu einem gegebenen Musikstück üblicherweise mehrere Musikdarstellungen existieren, wie verschiedene Aufnahmen oder partiturähnliche MIDI-Darstellungen. Unsere Benutzerschnittstelle ermöglicht es dem Nutzer, in interaktiver Weise vereinheitlichende Sichtweisen auf die Analyseresultate über verschiedene Versionen hinweg einzunehmen. Auf diese Weise können Konsistenzen und Inkonsistenzen in den versionsabhängigen Analyseresultaten leicht durch den Nutzer lokalisiert werden. Die neuen Evaluations- und Navigationsmöglichkeiten dieser Benutzerschnittstelle ermöglichen interdisziplinäre Kollaborationen, in denen Musikwissenschaftler ihr musikalisches Wissen und ausgebildetes Gehör einsetzen können, um auf angenehme Art und Weise versionsabhängige Analyseresultate computerbasierter Methoden auszuwerten.

Drittens, führen wir ein Cross-Version-Verfahren ein, das die harmonischen Eigenschaften verschiedener Audioversionen synchron analysiert. Die computer-basierte harmonische Analyse, die als Chord Labeling bezeichnet wird, ist von zentraler Wichtigkeit in der automatisierten Musikverarbeitung. Chord Labeling-Verfahren werden üblicherweise auf großen Audiodatenbeständen ausgewertet, indem die automatisch extrahierten Akkordlabel mit manuell erstellten Ground Truth-Annotationen verglichen werden. Hierbei wird das zu analysierende Stück üblicherweise durch eine bestimmte Audioaufnahme repräsentiert, die versionsabhängige Eigenschaften aufweist. Ein weiteres bedeutendes Problem basiert auf der Tatsache, dass sich audiobasierte Ergebnisse auf die physikalische Zeitachse der betrachteten Audioaufnahme in Sekunden beziehen, wohingegen die auf dem Notentext basierenden Analyseresultate eines Musikexperten sich auf eine musikalische Zeitachse in Takten beziehen. Allein diese Tatsache gestaltet es oft schwierig, Musikwissenschaftler in die Evaluierung audiobasierter Chord Labeling-Verfahren einzubeziehen. Das vorgestellte Cross-Version Chord Labeling-Verfahren ist darauf ausgerichtet, die starke Abhängigkeit der Chord Labeling-Ergebnisse von einer bestimmten Version zu überwinden. Wir zeigen, dass der Einsatz eines Cross-Version-Verfahrens das Chord Labeling-Ergebnis in der Weise stabilisiert, dass Inkonsistenzen auf versionsabhängige Eigenschaften hindeuten, während Konsistenzen über verschiedene Versionen hinweg harmonisch stabile Passagen in dem musikalischen Werk repräsentieren. Insbesondere zeigen wir, dass konsistente Bereiche oft korrekten Bereichen des Analyseresultates entsprechen. Unsere Experimente belegen, dass unter Verwendung des Cross Version-Verfahrens die Precision bedeutend ansteigt, wobei der Recall gleichzeitig auf einem relativ hohen Niveau verbleibt. Weiterhin beschreiben wir, wie die Zeitachse der aus Audioaufnahmen gewonnenen Analyseresultate in eine gemeinsame musikalische Zeitachse, gegeben in Takten, umgewandelt werden kann. Dies erleichtert nicht nur eine angemessene Auswertung eines Musikwissenschaftlers sondern ermöglicht außerdem, Analyseresultate über verschiedene Aufnahmen hinweg miteinander zu vergleichen. Wir führen eine mächtige Visualisierung ein, die die harmonisch stabilen Passagen auf einer musikalischen Zeitachse in Takten anzeigt, und demonstrieren wie diese Cross-Version-Visualisierung Musikwis-

senschaftlern als unterstützendes Hilfsmittel dienen kann, um harmonische Strukturen zu ergründen. Indem wir das Auftreten tonaler Zentren über den gesamten Werkzyklus von Beethovens Klaviersonaten untersuchen, zeigen wir abschliessend, dass unser Cross-Version-Verfahren grossangelegte harmonische Analysen ermöglicht.

Contents

Chapter 1

Introduction

This thesis deals with the introduction of novel automated methods for the analysis of music which are motivated by concrete applications in musicology. In the following, we first describe our motivation and formulate the general goal of our research (Section 1.1). In Section 1.2 we indicate the contributions of the particular chapters of this thesis before presenting in Section 1.3 an overview of the author's related publications. Finally, the structure of this thesis is described in Section 1.4.

1.1 Motivation

In the last years, music information retrieval (MIR) has become an active research field. In this context, numerous novel computer-based methods for extracting musically meaningful information from audio recordings have been developed. However, these methods often lack the applicability to music sciences. They are rarely geared to the needs of musicologists and often do not fit into the scientific context of musicology. Although the field of MIR opens the possibility of performing interdisciplinary research, computer scientists and musicologists rarely collaborate and benefit from each other. There exists still a large gap between computer science and musicology.

One reason is a lack of communication and mutual understanding between musicologists and computer scientists. On the one hand, musicologists are often not aware of novel developments in MIR. On the other hand, computer scientists often do not have an adequate musical background to comprehend the musical relevance of the analysis results. Automated methods usually require a strong background in computer science so that traditional musicologists have difficulties in applying them. Another source of mutual incomprehension may be the fact that analysis results obtained from automated methods are often based on simplifying model assumptions. Finally, methodologies used in computer science and in musicology fundamentally differ from each other so that novel concepts are needed allowing for a transfer from one field to the other. For example, many of the automated procedures are evaluated on the basis of recorded audio material, whereas musicologists typically work on the basis of symbolic music representations. Here, a major problem arises from the fact that audio-based results refer to the physical time axis given in sec-

onds of the considered audio recording, whereas score-based analysis results typically refer to a musical time axis given in bars. As a consequence, such computer-based analysis results are only of limited use for musicologists.

This thesis aims at the development of computer-based methods which can be directly applied to music sciences. Starting from a concrete application scenario of central importance in musicology, we introduce various automated methods which open new ways for supporting musicologists in their work.

Furthermore, our goal is to bridge the gap between computer science and musicology. Collaborating with music experts we contribute to establishing communication between the two fields and show how computer scientists and musicologists can benefit from each other in the context of interdisciplinary research. In this context, we introduce various novel concepts which support interdisciplinary collaborations.

1.2 Contributions

This thesis deals with the development of computer-based methods which are motivated by concrete application scenarios in musicology. We show in three different areas how our novel automated methods may support musicologists in their work. In Part I, we introduce a fully automatic approach for the extraction of tempo parameters from audio recordings and investigate to which extent this approach may support musicologists in analyzing recorded performances. In Part II, we present novel user interfaces which open up new possibilities for interdisciplinary collaborations between computer scientists and musicologists. Finally, Part III deals with the harmonic analysis of audio recordings, where we demonstrate how a cross-version approach may support musicologists in exploring harmonic structures even across large music corpora. In the following, we summarize the particular chapters and indicate the respective contributions.

The main contributions of Part I which deals with tempo analysis, are contained in Chapter 3. Here, we present a novel approach towards extracting temporal performance attributes from music recordings in a fully automated fashion. We exploit the fact that for many pieces there exists a kind of "neutral" representation in the form of a musical score (or MIDI file) that explicitly provides the musical onset and pitch information of all occurring note events. Using music synchronization techniques, we temporally align these note events with their corresponding physical occurrences in the music recording. As our main contribution, we describe various algorithms for deriving tempo curves from these alignments which reveal the relative tempo differences between the actual performance and the neutral reference representation. We have evaluated the quality of the automatically extracted tempo curves on harmony-based Western music of various genres. Besides a manual inspection of a representative selection of real music performances, we have also conducted a quantitative evaluation on synthetic audio material generated from randomly warped MIDI files. Our experiments indicate that our automated methods yield accurate estimations of the overall tempo flow and, for certain classes of music such as piano music, of even finer expressive tempo nuances.

Part II concerns the topic of user interaction. The main contributions of this part are

presented in Chapters 5 and 6. In Chapter 5, we report on an investigation with the objective of introducing a novel MIR interface to music education. In collaboration with the University of Music Saarbrücken we conducted an experiment consisting of several steps. First, nine piano students were recorded playing the same piece of music, the first movement of Beethoven's Pathétique Sonata Op. 13, on the same piano and under the same recording conditions. In the next step, the nine audio recordings were temporally aligned and integrated in a user interface referred to as Interpretation Switcher [16, 54], which allows for synchronous playback of the different performances. Upon using this interface, the music students were then asked to analyze the anonymised performances according to a well-designed questionnaire. There are a number of achievements of our experiment. Firstly, we tested and evaluated our interface in a setting of practical relevance, thus indicating the potential of MIR methods in music education. Secondly, we generated royalty free music recordings without any copyright restrictions, which can be used freely for research purposes. Thirdly, using a Yamaha Disklavier for our experiments, we also obtained MIDI data (which was actually not used in the investigation described in this paper) along with audio recordings. Such MIDI-audio pairs can be used as ground truth material for various MIR tasks [54]. Finally, we generated many different interpretations of the same piece, which yields valuable data for tasks such as automated performance analysis [88].

In Chapter 6, we introduce a user interface that facilitates novel ways of viewing, comparing, and evaluating analysis results obtained from different methods and computed on the basis of different music representations. Here, we exploit the fact that for a given piece of music one often has multiple, closely-related sources of information, including audio recordings of different performances and score-like representations including MIDI versions. Our interface combines and extends the functionality of known user interfaces for inter- and intra-document navigation [9, 11, 21, 80]. The technical backbone of our interface is the *Interpretation Switcher* [16], which allows a user to select several recordings of the same piece of music and, during playback, to seamlessly switch between these versions (inter-document navigation). We extended this switcher to additionally visualize version-dependent annotations such as chord labels or structure blocks, which can be used for intra-document navigation similar to [21]. As one main contribution, we introduce different modes for adjusting the version-dependent timelines of the music representations. Furthermore, our interface allows for interactively generating multi-perspective views across the different version-dependent analysis results disclosing consistencies and inconsistencies. This allows a user to conveniently locate, playback, and compare musically interesting passages, which not only makes evaluation and annotation easier but also deepens the listener's understanding of the annotations and the underlying audio material. Here, our interface not only allows a technically unexperienced user to interact with the music analysis results and the audio material, but also opens up new possibilities for enriching music education using signal processing techniques.

Part III deals with harmonic analysis, where mainly Chapters 8, 9, 10, 11, and 12 present novel contributions. The computer-based harmonic analysis of audio recordings is one of the central tasks in MIR and is referred to as chord labeling. The evaluation of chord labeling procedures is typically performed on large audio collections, where the automatically extracted chord labels are compared to manually generated ground truth annotations.

Here, the piece to be analyzed is typically represented by an audio recording, which possesses version-dependent characteristics. For example, specific instruments are used, which have instrument-dependent sound properties, e. g., concerning the energy distributions in the harmonics. Similarly, room acoustics and other recording conditions may have a significant impact on the audio signal's spectral properties. Finally, by emphasizing certain voices or suppressing others, a musician can change the sound in order to shape the piece of music. As a consequence, the chord labeling results strongly depend on specific characteristics of the considered audio recording. Another major problem arises from the fact, that audio-based recognition results refer to the physical time axis given in seconds of the considered audio recording, whereas score-based analysis results obtained by music experts typically refer to a musical time axis given in bars. This simple fact alone makes it often difficult to get musicologists involved into the evaluation process of audio-based music analysis. For example, for the evaluation of chord labeling procedures, ground truth annotations are required. While the manual generation of audio-based annotations is a tedious and time-consuming process musicians are trained to derive chord labels by means of printed sheet music. Such labels, however, are only of limited use for the evaluation of audio-based recognition results. First research efforts have been directed towards the use of score-based ground truth labels for audio-based chord recognition, where it turned out that incorporating such ground truth labels may significantly improve machine learning methods for chord recognition [44, 51].

In Chapter 8, we introduce a cross-version chord recognition approach. By exploiting the fact that for a musical work there often exist a large number of different audio recordings as well as symbolic representations, we analyze the available versions independently using some automated chord labeling procedure and employ a late-fusion approach to merge the version-dependent analysis results. Here, the idea is to overcome the strong dependency of chord labeling results on a specific version. We observe that more or less random decisions in the automated chord labeling typically differ across several versions. Such passages often correspond to harmonically instable passages leading to inconsistencies. In contrast, consistencies across several versions typically indicate harmonically stable passages. As another contribution, we describe how to transform the time axis of analysis results obtained from audio recordings to a common musical time axis given in bars. This not only facilitates a convenient evaluation by a musicologist, but also allows for comparing analysis results across different recorded performances. Finally, we introduce a powerful visualization which is based on the cross-version chord labeling (another interesting approach for visualizing harmonic structures of tonal music has been suggested in [74]). The cross-version visualization indicates the harmonically stable passages in an intuitive and non-technical way leading the user to passages dominated by a certain key also referred to as tonal centers. Furthermore, in the case that score-based ground truth labels are also provided, the visualization allows for an in-depth error analysis of chord labeling procedures.

In Chapter 9 we show how a cross-version approach serves for the evaluation of two MIDI-based chord labelers using annotations given for corresponding audio recordings. As main contribution, we present a qualitative evaluation of the two chord labeling procedures. Performing an in-depth error analysis we classify possible error sources and, furthermore, illustrate the respective error source by means of concrete song examples. This qualitative

error analysis not only indicates limitations of the employed symbolic chord labelers but also deepens the understanding for the underlying music material.

In Chapter 10, we show that consistently labeled passages across several versions often correspond to correct labeling results. Consequently, one can exploit the consistency information to significantly increase the precision of the result while keeping the recall at a relatively high level, which can be regarded as a stabilization of the labeling procedure. Furthermore, we show that our cross-version approach is conceptually different to a constraint-based approach, where only chord labels are considered that are particularly close to a given chord model. Unlike our cross-version approach, using such simple constraints leads to a significant loss in recall.

As our main contribution in Chapter 11, we present a detailed case study on Beethoven's Sonata Op. 57, the so-called Appassionata. Here, in a collaboration with musicologists, our cross-version visualization is used as a helpful tool for exploring harmonic structures demonstrating how computer-based methods and visualizations may support musicologists in their work.

In Chapter 12, we demonstrate how our cross-version approach enables for large-scale analyses of harmonic structures. Performing an analysis of tonal centers across the entire corpus of Beethoven's piano sonatas, we reveal commonalities, differences and trends in the appearance of tonal centers. In this way, we show how our cross-version approach may support musicologists in investigating tonal centers across large music corpora.

1.3 Related Publications

This thesis is based on various publications, which are listed below in chronological order. Furthermore, for each publication we indicate how it is related to the thesis.

[37] Verena Konz, Meinard Müller, and Andi Scharfstein, *Extracting expressive tempo curves from music recordings*, in Proceedings of the 35th International Conference on Acoustics (NAG/DAGA), Rotterdam, The Netherlands, 2009.

[61] Meinard Müller, Verena Konz, Andi Scharfstein, Sebastian Ewert, and Michael Clausen, *Towards Automated Extraction of Tempo Parameters from Expressive Music Recordings*, in Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR), Kobe, Japan, 2009, pp. 69–74.

[37] and [61] deal with the automated extraction of tempo parameters from audio recordings which is discussed in detail in Chapter 3.

[33] Verena Konz and Meinard Müller, *Introducing the Interpretation Switcher Interface to Music Education*, in Proceedings of the 2nd International Conference on Computer Supported Education (CSEDU), Valencia, Spain, 2010, pp. 135–140.

[33] describes an experiment conducted at the University of Music Saarbrücken with the goal to introduce a novel MIR user interface to music education. This experiment is presented in Chapter 5 of this thesis.

[55] Meinard Müller, Michael Clausen, Verena Konz, Sebastian Ewert, and Christian Fremerey, *A Multimodal Way of Experiencing and Exploring Music*, Interdisciplinary Science Reviews (ISR), 35 (2010), pp. 138–153.

In [55] we show how music synchronization techniques can be integrated into novel user interfaces that allow the user to access and explore music in all its different facets thus enhancing human involvement with music and deepening music understanding. In particular, we discuss three different case studies, where automated synchronization methods play an important role for supporting the user in experiencing and exploring music. Two of the three case studies are closely related to this thesis. One case study describes the experiment which is part of Chapter 5. A second case study shows how synchronization can be used for the automated extraction of tempo parameters from audio recordings, which is in the center of [61] and described in detail in Chapter 3.

[35] Verena Konz, Meinard Müller, and Sebastian Ewert, *Ein Baseline-Experiment zur Klassifizierung von Problemen bei der Akkorderkennung*, in Proceedings of the 36th Deutsche Jahrestagung für Akustik (DAGA), Berlin, Germany, 2010.

In [35] a baseline-experiment is conducted with the goal to classify problems appearing in the context of chord labeling. Compensating for tuning deviations in the chord labeling procedure turns out to be of particular importance. The baseline-experiment showing the importance of tuning in the context of chord labeling is described in Section 7.7.

[36] Verena Konz, Meinard Müller, and Sebastian Ewert, *A Multi-Perspective Evaluation Framework for Chord Recognition*, in Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR), Utrecht, The Netherlands, 2010, pp. 9–14.

The automated extraction of chord labels from audio recordings constitutes a major task in music information retrieval. To evaluate computer-based chord labeling procedures, one requires ground truth annotations for the underlying audio material. However, the manual generation of such annotations on the basis of audio recordings is tedious and time-consuming. On the other hand, trained musicians can easily derive chord labels from symbolic score data. In [36] we describe a procedure that allows for transferring annotations and chord labels from the score domain to the audio domain and vice versa. Using music synchronization techniques, the general idea is to locally warp the annotations of all given data streams onto a common time axis, which then allows for a cross-domain evaluation of the various types of chord labels. As a further contribution of this paper, we extend this principle by introducing a multi-perspective evaluation framework for simultaneously comparing chord recognition results over multiple performances of the same piece of music. In [36] the idea of cross-version harmonic analysis, which is in the center of Chapter 8, is introduced for the first time. The procedure for transferring annotations from the score domain to the audio domain and vice versa is presented in Section 8.1.5.

[60] Meinard Müller, Verena Konz, Nanzhu Jiang, and Zhe Zuo, *A Multi-Perspective User Interface for Music Signal Analysis*, in Proceedings of the International Computer Music Conference (ICMC), Huddersfield, England, UK, 2011, pp. 205–211.

In [60] we introduce various novel functionalities for a user interface that opens up new possibilities for viewing, comparing, interacting, and evaluating analysis results within a multi-perspective framework and that bridges the gap between signal processing and music sciences. This publication is part of Chapter 6.

[30] Nanzhu Jiang, Peter Grosche, Verena Konz, and Meinard Müller, *Analyzing Chroma Feature Types for Automated Chord Recognition*, in Proceedings of the Audio Engineering Society Conference (AES), Ilmenau, Germany, 2011.

In [30], the role of the feature extraction step within the recognition pipeline of various chord recognition procedures based on template matching strategies and hidden Markov models is analyzed. In particular, numerous experiments are described which show how the various procedures depend on the type of the underlying chroma feature as well as on parameters that control temporal and spectral aspects. Parts of [30] are described in Chapter 7. In particular, the experiment in which the dependency of the chord recognition results on the underlying feature type is investigated is described in detail in Section 7.6.

[59] Meinard Müller and Verena Konz, *Automatisierte Methoden zur Unterstützung der Interpretationsforschung*, in Gemessene Interpretation, Heinz von Loesch and Stefan Weinzierl, eds., vol. 4 of Klang und Begriff, Schott Verlag, 2011, pp. 193–204.

In [59] the potential and the limitations of automated methods are discussed. In particular, we exemplarily show to which extent automated methods for extracting tempo parameters from audio recordings may support a musicologist in analyzing recorded performances.

[34] Verena Konz and Meinard Müller, *A Cross-Version Approach for Harmonic Analysis of Music Recordings*, in Multimodal Music Processing (Dagstuhl Seminar 11041), Dagstuhl Follow-Ups, 3 (2012), pp. 53–71.

In [34] we present a cross-version approach for harmonic analysis of audio recordings which is part of Chapter 8. Furthermore, we show that by analyzing the harmonic properties of several audio versions synchronously one can achieve a stabilization of the chord labeling results in the sense that inconsistencies indicate version-dependent characteristics or musically problematic passages, whereas consistencies across several versions indicate harmonically stable passages in the piece of music. In particular, we show that consistently labeled passages often correspond to correctly labeled passages. Our experiments document that the cross-version labeling procedure significantly increases the precision of the result while keeping the recall at a relatively high level. The stabilization of audio chord labeling is in the center of Chapter 10.

[14] Sebastian Ewert, Meinard Müller, Verena Konz, Daniel Müllensiefen, and Geraint Wiggins, *Towards Cross-Version Harmonic Analysis of Music*, IEEE Transactions on Multimedia, 2012, to appear.

In [14] we realize the idea of cross-version harmonic analysis to automatically evaluate MIDI-based chord labeling procedures using annotations given for corresponding audio recordings. To this end, one needs reliable synchronization procedures that automatically establish the musical relationship between the multiple versions of a

given piece. This becomes a hard problem when there are significant local deviations in these versions. In [14] a novel late-fusion approach that combines different alignment procedures in order to identify reliable parts in synchronization results is introduced which is not part of this thesis. The cross-version comparison of the various chord labeling results is then performed only on the basis of the reliable parts. Finally, we present a qualitative evaluation of the two symbolic chord labelers, where we classify possible error sources and illustrate the respective error source by means of concrete song examples. This qualitative evaluation not only indicates limitations of the employed chord labeling strategies but also deepens the understanding of the underlying music material. The cross-version evaluation of the two MIDI-based chord labelers is part of Chapter 9.

Under Review

Verena Konz, Meinard Müller and Rainer Kleinertz, *A Cross-Version Chord Labelling Approach for Exploring Harmonic Structures—A Case Study on Beethovens Appassionata—*, submitted to the Journal of New Music Research.

In this paper, we present a case study on Beethoven's Appassionata in order to demonstrate how computer-based methods may assist musicologists when performing harmonic analyses. Using the cross-version visualization we perform a detailed harmonic analysis of the Appassionata, where it turns out that that the consistencies in the labeling results across different versions typically correspond to harmonically stable passages, thus being of musical relevance. This shows that our visualisation can be used as a supportive tool for exploring harmonic structures and constitutes a source of inspiration. The case study on Beethoven's Appassionata is described in detail in Chapter 11.

1.4 Outline

This thesis is organized as follows. It is structured into three different parts, where Part I deals with tempo analysis. Here, we first present an overview about music synchronization which is a concept of fundamental importance for this thesis (Chapter 2). Afterwards, in Chapter 3, we introduce a fully automatic approach for extracting tempo parameters from audio recordings using synchronization techniques. Furthermore, we indicate the potential and the limitations of such automated methods for supporting musicologists in analyzing recorded performances. The human interaction with computer-based interfaces is in the center of Part II. In Chapter 5, we describe an experiment conducted at the University of Music Saarbrücken which aimed at introducing a novel MIR interface to music education. Then, in Chapter 6, we present a novel multi-perspective user interface which opens up new possibilities for viewing, comparing, interacting and evaluating analysis results and that bridges the gap between signal processing and music sciences. Harmonic analysis is the central topic of Part III. Here, we first give an overview of the chord labeling task, where we describe typical approaches and feature types used in the context of chord labeling as well as two experiments highlighting the importance of features and the significance of tuning (Chapter 7). Afterwards, in Chapter 8, we introduce the concept of cross-version

chord labeling. This concept is applied to the cross-version evaluation of two MIDI-based chord labelers using annotations given for corresponding audio recordings. In Chapter 10, we show that employing a cross-version approach one can achieve a stabilization of the chord labeling results. In Chapter 11, we then present a case study on Beethoven's Appassionata in which the cross-version visualization is exemplarily used for performing a detailed harmonic analysis of this musical work. Here, our cross-version visualization turns out to be a helpful tool for supporting musicologists in exploring harmonic structures. In Chapter 12, we analyze harmonic structures across the entire corpus of Beethoven's piano sonatas demonstrating how the cross-version approach allows for large-scale harmonic analyses. Finally, we conclude in Chapter 13 by reflecting on a meaningful use of automated methods in the context of interdisciplinary research and indicating our vision of the development of MIR in the future.

Part I

Tempo Analysis

Chapter 2

Music Synchronization

In this chapter, we describe the concept of music synchronization which is of central importance for this thesis. In particular, the fully automatic approach for the extraction of tempo parameters from audio recordings as well as the concept of cross-version harmonic analysis presented in Chapters 3 and 8, respectively, are based on music synchronization techniques.

In the following, we follow [55, 61]. A musical work is far from simple or singular. In particular, there may exist various audio recordings, MIDI files, digitized sheet music, and other symbolic representations. The general goal of *music synchronization* is to automatically link the various data streams thus interrelating the multiple information sets related to a given musical work [29, 54]. More precisely, *synchronization* is taken to mean a procedure which, for a given position in one representation of a piece of music, determines the corresponding position within another representation. The result of a synchronization process is illustrated by Figure 2.1 in the form of red bidirectional arrows. Here, a MIDI representation is synchronized with an audio recording. Automated music synchronization constitutes a challenging research field since one has to account for a multitude of aspects such as the data format, the genre, the instrumentation, or differences in parameters such as tempo, articulation and dynamics that result from expressiveness in performances. In the design of synchronization algorithms, one has to deal with a delicate trade-off between robustness, temporal resolution, alignment quality, and computational complexity.

In order to synchronize two different music representations, one typically proceeds in two steps, which are explained next. For details, we refer to [54]. In the first step, the two music representations are transformed into sequences of suitable features, say $X := (x_1, x_2, \ldots, x_N)$ and $Y := (y_1, y_2, \ldots, y_M)$, respectively. Here, on the one hand, the feature representations should show a large degree of robustness to variations that are to be left unconsidered in the comparison. On the other hand, the feature representations should capture characteristic information that suffice to accomplish the synchronization tasks. In this context, chroma-based music features have turned out to be a powerful tool for synchronizing harmony-based music, see [2, 18, 54, 57]. Here, the chroma refer to the 12 traditional pitch classes of the equal-tempered scale encoded by the attributes $\mathrm{C}, \mathrm{C}^\sharp, \mathrm{D}, \ldots$,B. Representing the short-time content of a music representation in each of the 12 pitch classes, chroma features show a large degree of robustness to variations in

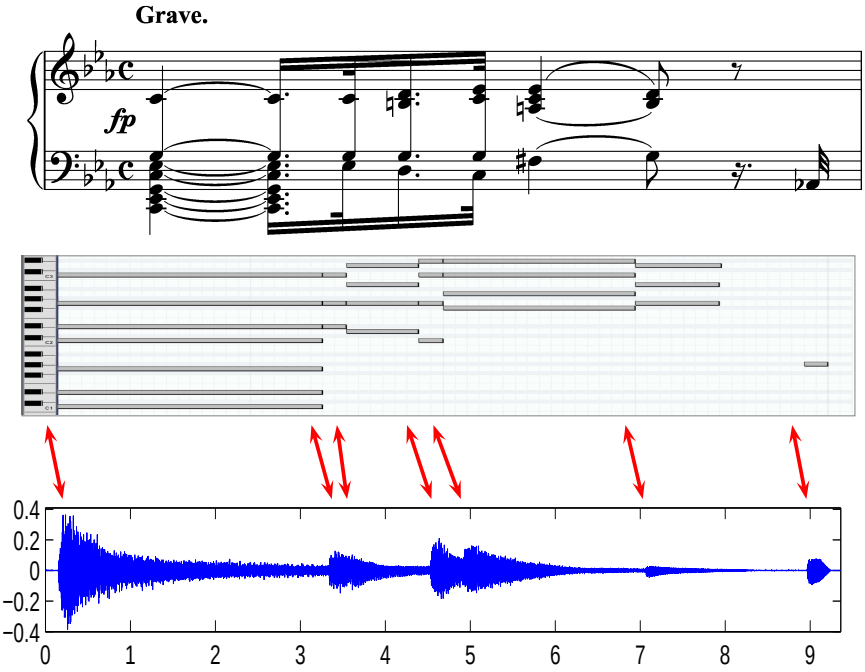

Figure 2.1. First measure of Beethoven's Pathétique Sonata Op. 13. The MIDI-audio alignment is indicated by the arrows.

timbre and dynamics, while keeping sufficient information to characterize harmony-based music.

In the second step, the derived feature sequences have to be brought into temporal correspondence to account for temporal variations in the two music representations to be synchronized. An important technique for computing such a correspondence is dynamic time warping (DTW), which is a well-known technique to find an optimal alignment between two given (time-dependent) sequences under certain restrictions. Intuitively, the alignment can be thought of a linking structure indicated by red bidirectional arrows as shown in Figure 2.1. These arrows encode how the sequences are to be warped (in a non-linear fashion) to match each other. Therefore, from the feature sequences, an $N \times M$ cost matrix C is built up by evaluating a local cost measure c for each pair of features, i. e., $C(n,m) = c(x_n, y_m)$ for $n \in [1:N] := \{1,2,\ldots,N\}$ and $m \in [1:M]$. Each tuple $p = (n,m)$ is called a *cell* of the matrix. A (global) *alignment path* is a sequence $(p_1,\ldots,p_L)$ of length L with $p_\ell \in [1:N] \times [1:M]$ for $\ell \in [1:L]$ satisfying $p_1 = (1,1)$, $p_L = (N,M)$ and $p_{\ell+1} - p_\ell \in \Sigma$ for $\ell \in [1:L-1]$. Here, $\Sigma = \{(1,0),(0,1),(1,1)\}$ denotes the set of admissible step sizes. The *cost* of a path $(p_1,\ldots,p_L)$ is defined as $\sum_{\ell=1}^{L} C(p_\ell)$. Then, a cost-minimizing alignment path, which constitutes the final synchronization result, is computed from C via dynamic programming.

For a detailed account on DTW and music synchronization we refer to [29, 54] and the references therein. Based on this general strategy, we employ a multiscale synchronization algorithm based on high-resolution audio features as described in [13]. This approach, which combines the high temporal accuracy of onset features with the robustness of chroma features, generally yields robust music alignments of high temporal accuracy.

Chapter 3

Extracting Tempo Parameters from Audio Recordings

A performance of a piece of music heavily depends on the musician's or conductor's individual vision and personal interpretation of the given musical score. As basis for the analysis of artistic idiosyncrasies, one requires accurate annotations that reveal the exact timing and intensity of the various note events occurring in the performances. In the case of audio recordings, this annotation is often done manually, which is prohibitive in view of large music collections. In this chapter, we present a fully automatic approach for extracting temporal information from a music recording using score-audio synchronization techniques. This information is given in the form of a tempo curve that reveals the relative tempo difference between an actual performance and some reference representation of the underlying musical piece. As shown by our experiments on harmony-based Western music, our approach allows for capturing the overall tempo flow and for certain classes of music even finer expressive tempo nuances. The results presented in this chapter have been published in [37, 59, 61].

The chapter is organized as follows. First, we present an overview of related work (Section 3.1). Then, we introduce various algorithms for extracting tempo curves from expressive music recordings (Section 3.2). Our experiments are described in Section 3.3, before we discuss the potential and the limitations of automated methods (Section 3.4). Finally, we conclude with prospects on future work (Section 3.5).

3.1 Related Work

Musicians give a piece of music their personal touch by continuously varying tempo, dynamics, and articulation. Instead of playing mechanically they speed up at some places and slow down at others in order to shape a piece of music. Similarly, they continuously change the sound intensity and stress certain notes. Such performance issues are of fundamental importance for the understanding and perception of music. The automated analysis of different interpretations, also referred to as *performance analysis*, has become an active field of research [39, 73, 88]. Here, one goal is to find commonalities between

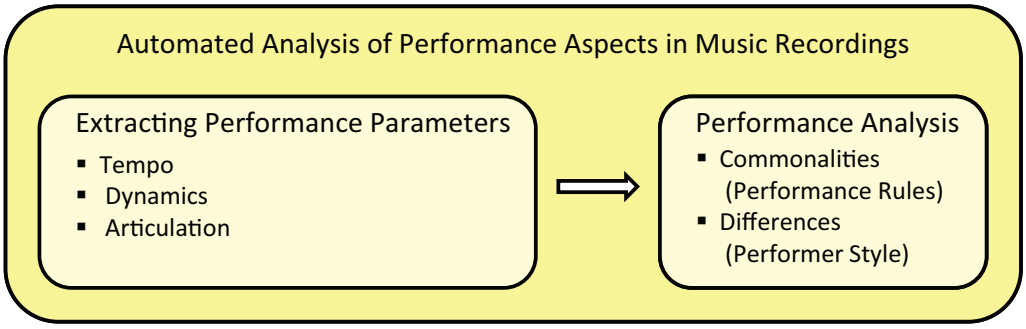

Figure 3.1. Automated analysis of performance aspects in music recordings.

different interpretations, which allow for the derivation of general performance rules. A kind of orthogonal goal is to capture what is characteristic for the style of a particular interpreter. Before one can analyze a specific performance, one requires the information about when and how the notes of the underlying piece of music are actually played, see Figure 3.1. Therefore, as the first step of performance analysis, one has to annotate the performance by means of suitable attributes that make explicit the exact timing and intensity of the various note events. The extraction of such performance attributes constitutes a challenging problem, in particular in the case of audio recordings.

Many researchers manually annotate the audio material by marking salient data points in the audio stream. Using novel music analysis interfaces such as the Sonic Visualiser [80], experienced annotators can locate note onsets very accurately even in complex audio material [73, 87]. However, being very labor-intensive, such a manual process is prohibitive in view of large audio collections. Another way to generate accurate annotations is to use a computer-monitored *player piano*. Equipped with optical sensors and electromechanical devices, such pianos allow for recording the key movements along with the acoustic audio data, from which one directly obtains the desired note onset information [87, 88]. The advantage of this approach is that it produces precise annotations, where the symbolic note onsets perfectly align with the physical onset times. The obvious disadvantage is that special-purpose hardware is needed during the recording of the piece. In particular, conventional audio material taken from CD recordings cannot be annotated in this way. Therefore, the most preferable method is to automatically extract the necessary performance aspects directly from a given audio recording. Here, automated approaches such as *beat tracking* [10] and *onset detection* [3] are used to estimate the precise timings of note events within the recording. Even though great research efforts have been directed towards such tasks, the results are still unsatisfactory, in particular for music with weak onsets and strongly varying beat patterns. In practice, semi-automatic approaches are often used, where one first roughly computes beat timings using beat tracking software, which are then adjusted manually to yield precise beat onsets.

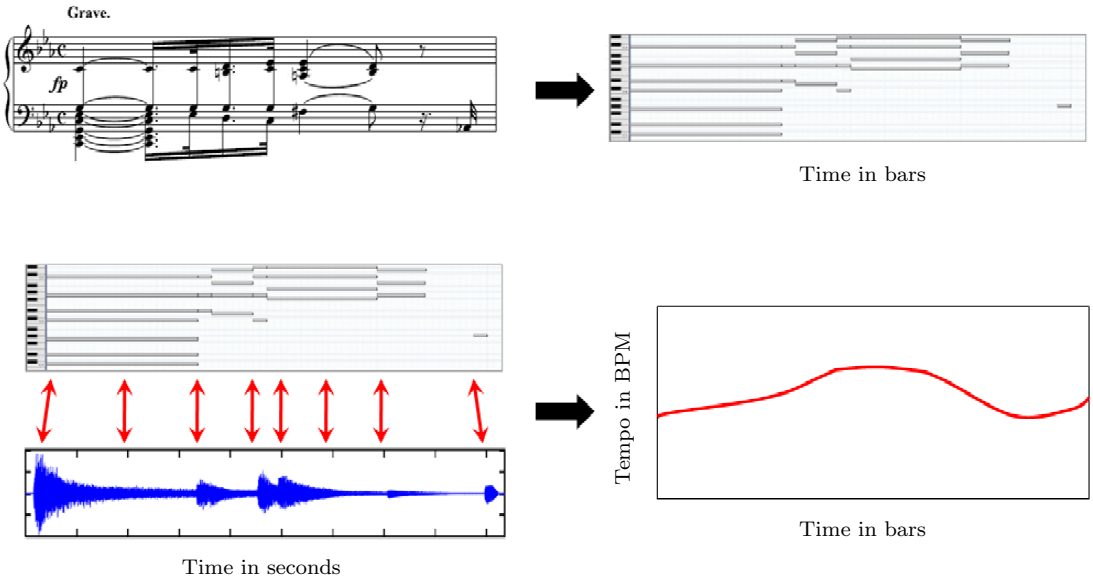

Figure 3.2. Automated extraction of tempo curves using synchronization techniques illustrated by means of the first bar of Beethoven's Piano Sonata Op. 13 (Pathétique). From the score an uninterpreted MIDI representation is derived, which is synchronized with the considered audio recording. Then, the MIDI-audio alignment is used to derive a tempo curve.

3.2 Computation of Tempo Curves

The feeling of pulse and rhythm is one of the central components of music and closely relates to what one generally refers to as tempo. In order to define some notion of tempo, one requires a proper reference to measure against. For example, Western music is often structured in terms of bars and beats, which allows for organizing and sectioning musical events over time. Based on a fixed time signature, one can then define the tempo as the number of beats per minute (BPM). Obviously, this definition requires a regular and steady musical beat or pulse over a certain period in time. Also, the very process of measurement is not as well-defined as one may think. Which musical entities (e. g., note onsets) characterize a pulse? How precisely can these entities be measured before getting drowned in noise? How many pulses or beats are needed to obtain a meaningful tempo estimation? With these questions, we want to indicate that the notion of tempo is far from being well-defined. Different representations of timing and tempo are presented in [28].

Figure 3.2 now shows a schematic overview of the automated procedure for computing tempo curves. Here, we assume that we are given a "neutral" MIDI file, where the notes are played with a constant tempo in a purely mechanical way. Such a MIDI file can be generated from a score using a fixed global tempo (measured in BPM), see Figure 3.2 (top). In the following, we refer to this MIDI file as *reference representation* of the underlying piece of music. Assuming that the time signature of the piece is known, one can recover bar and beat positions from MIDI time positions. Given a specific performance to be analyzed in the form of an audio recording, we first use music synchronization techniques to compute a MIDI-audio alignment path as described in Section 2. From this path we

derive a *tempo curve* that describes for each time position within the MIDI reference (given in seconds or bars) the tempo of the performance (given as a multiplicative factor of the reference tempo or in BPM), see Fig 3.2 (bottom). Figure 3.5 and Figure 3.6 show some tempo curves for various performances.

Intuitively, the value of the tempo curve at a certain reference position corresponds to the slope of the alignment path at that position. However, due to discretization and alignment errors, one needs numerically robust procedures to extract the tempo information by using average values over suitable time windows. In the following, we describe three different approaches for computing tempo curves using a fixed window size (Section 3.2.1), an adaptive window size (Section 3.2.2), and a combined approach (Section 3.2.3).

3.2.1 Fixed Window Size

Recall from Section 2 that the alignment path $p = (p_1, \ldots, p_L)$ between the MIDI reference and the performance is computed on the basis of the feature sequences $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_M)$. Note that one can recover beat and bar positions from the indices $n \in [1 : N]$ of the reference feature sequence, since the MIDI representation has constant tempo and the feature rate is assumed to be constant.

To compute the tempo of the performance at a specific reference position $n \in [1 : N]$, we basically proceed as follows. First, we choose a neighborhood of n given by indices n_1 and n_2 with $n_1 \leq n \leq n_2$. Using the alignment path, we compute the indices m_1 and m_2 aligned with n_1 and n_2, respectively. Then, the tempo at n is defined as quotient $\frac{n_2-n_1+1}{m_2-m_1+1}$. The main parameter to be chosen in this procedure is the size of the neighborhood. Furthermore, there are some technical details to be dealt with. Firstly, the boundary cases at the beginning and end of the reference need special care. To avoid boundary problems, we extend the alignment path p to the left and right by setting $p_\ell := (\ell, \ell)$ for $\ell < 1$ and $p_\ell := (N + \ell - L, M + \ell - L)$ for $\ell > L$. Secondly, the indices m_1 and m_2 are in general not uniquely determined. Generally, an alignment path p may assign more than one index $m \in [1 : M]$ to a given index $n \in [1 : N]$. To enforce uniqueness, we chose the minimal index over all possible indices. More precisely, we define a function $\varphi_p : \mathbb{Z} \to [1 : M]$ by setting

$$\varphi_p(n) := \min\{m \in [1 : M] \mid \exists \ell \in \mathbb{Z} : p_\ell = (n, m)\}.$$

We now give the technical details of the sketched procedure for the case that the neighborhoods are of a fixed window (FW) size $w \in \mathbb{N}$. The resulting tempo curve is denoted by $\tau_w^{\mathrm{FW}} : [1 : N] \to \mathbb{R}_{\geq 0}$. For a given alignment path p and an index $n \in [1 : N]$, we define

$$n_1 := n - \left\lfloor \tfrac{w-1}{2} \right\rfloor \quad \text{and} \quad n_2 := n + \left\lceil \tfrac{w-1}{2} \right\rceil. \tag{3.1}$$

Then $w = n_2 - n_1 + 1$ and the tempo at reference position n is defined by

$$\tau_w^{\mathrm{FW}}(n) = \frac{w}{\varphi_p(n_2) - \varphi_p(n_1) + 1}. \tag{3.2}$$

The tempo curve τ_w^{FW} crucially depends on the window size w. Using a small window allows for capturing sudden tempo changes. However, in this case the tempo curve becomes

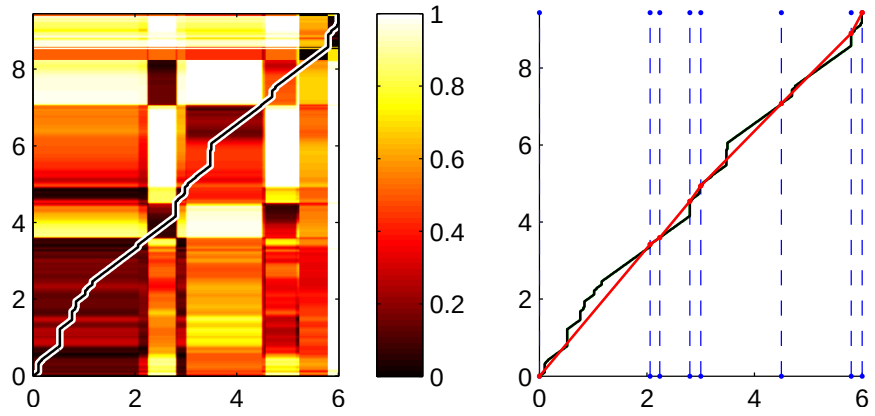

Figure 3.3. Left: Cost matrix and cost-minimizing alignment path for the Beethoven example shown in Figure 3.2. The reference representation (MIDI) corresponds to the horizontal and the performance (audio) to the vertical axis. **Right:** Original (black) and onset-rectified alignment path (red). The MIDI note onset positions are indicated by the blue vertical lines.

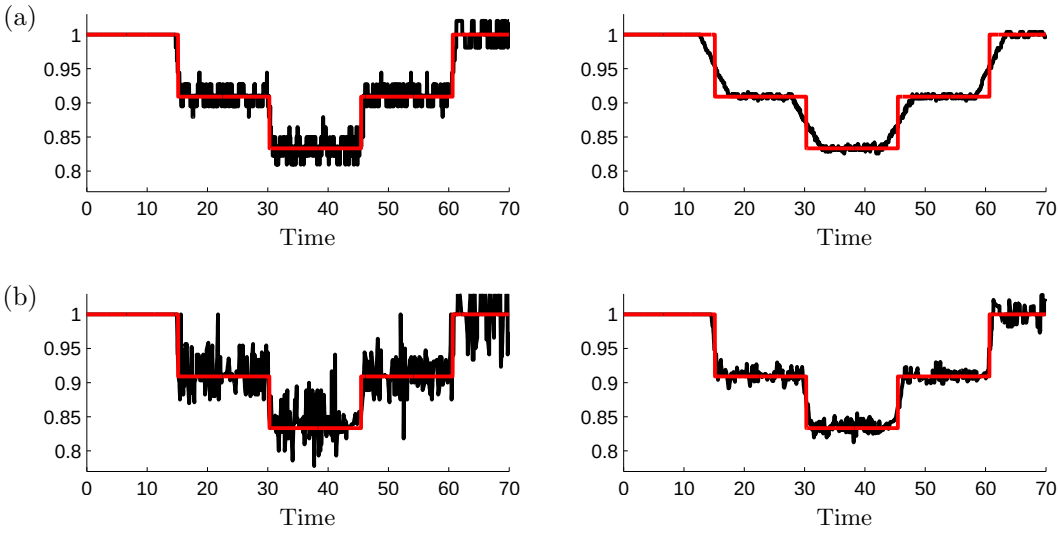

Figure 3.4. Ground truth tempo curve (step function) and various computed tempo curves. **(a)** τ_w^{FW} using a fixed window size with small w (left) and large w (right). **(b)** τ_v^{AW} using an adaptive window size with small v (left) and large v (right).

sensible to inaccuracies in the alignment path and synchronization errors. In contrast, using a larger window smooths out possible inaccuracies, while limiting the ability to accurately pick up local phenomena. This effect is also illustrated by Figure 3.4 (a), where the performance is synthesized from a temporally warped MIDI reference. We continue this discussion in Section 3.3.

3.2.2 Adaptive Window Size

Using a window of fixed size does not account for specific musical properties of the piece of music. We now introduce an approach using an adaptive window size, which is based

on the assumption that note onsets are the main source for inducing tempo information. Intuitively, in passages where notes are played in quick succession one may obtain an accurate tempo estimation even when using only a small time window. In contrast, in passages where only few notes are played one needs a much larger window to obtain a meaningful tempo estimation.

We now formalize this idea. We assume that the note onsets of the MIDI reference are given in terms of feature indices. Furthermore, for notes with the same onset position we only list one of these indices. Let $O = \{o_1, \ldots, o_K\} \subseteq [1 : N]$ be the set of onset positions with $1 \leq o_1 < o_2 < \ldots < o_K \leq N$. The distance between two neighboring onset positions is referred to as inter onset interval (IOI). Now, when computing the tempo curve at position $n \in [1 : N]$, the neighborhood of n is specified not in terms of a fixed number w of feature indices but in terms of a fixed number $v \in \mathbb{N}$ of IOIs. This defines an onset-dependent adaptive window (AW). More precisely, let $\tau_v^{\mathrm{AW}} : [1 : N] \to \mathbb{R}_{\geq 0}$ denote the tempo function to be computed. To avoid boundary problems, we extended the set O to the left and right by setting $o_k := o_1 + k - 1$ for $k < 1$ and $o_k := o_K + k - K$ for $k > K$. First, we compute τ_v^{AW} for all indices n that correspond to onset positions. To this end, let $n = o_k$. Then we define

$$k_1 := k - \left\lfloor \tfrac{v-1}{2} \right\rfloor \quad \text{and} \quad k_2 := k + \left\lceil \tfrac{v-1}{2} \right\rceil.$$

Setting $n_1 := o_{k_1}$ and $n_2 := o_{k_2}$, the tempo at reference position $n = o_k$ is defined as

$$\tau_v^{\mathrm{AW}}(n) := \frac{n_2 - n_1 + 1}{\varphi_p(n_2) - \varphi_p(n_1) + 1}. \tag{3.3}$$

Note that, opposed to (3.2), the window size $n_2 - n_1 + 1$ is no longer fixed but depends on the sizes of the neighboring IOIs around the position $n = o_k$. Finally, $\tau_v^{\mathrm{AW}}(n)$ is defined by a simple linear interpolation for the remaining indices $n \in [1 : N] \setminus O$. Similar to the case of a fixed window size, the tempo curve τ_v^{AW} crucially depends on the number v of IOIs, see Figure 3.4 (b). The properties of the various tempo curves are discussed in detail in Section 3.3.

3.2.3 Combined Strategy

So far, we have introduced two different approaches using on the one hand a fixed window size and on the other hand an onset-dependent adaptive window size for computing average slopes of the alignment path. Combining ideas from both approaches, we now present a third strategy, where we first rectify the alignment path using onset information and then apply the FW-approach on the rectified path for computing the tempo curve. As in Section 3.2.2, let $O = \{o_1, \ldots, o_K\} \subseteq [1 : N]$ be the set of onsets. By possibly extending this set, we may assume that $o_1 = 1$ and $o_K = N$. Now, within each IOI given by two neighboring onsets $n_1 := o_k$ and $n_2 := o_{k+1}$, $k \in [1 : K-1]$, we modify the alignment path p as follows. Let $\ell_1, \ell_2 \in [1 : L]$ be the indices with $p_{\ell_1} = (n_1, \varphi_p(n_1))$ and $p_{\ell_2} = (n_2, \varphi_p(n_2))$, respectively. While keeping the cells p_{ℓ_1} and p_{ℓ_2}, we replace the cells $p_{\ell_1 + 1}, \ldots, p_{\ell_2 - 1}$ by cells obtained from a suitably sampled linear function having the slope $\frac{n_2 - n_1 + 1}{\varphi_p(n_2) - \varphi_p(n_1) + 1}$. Here, in the sampling, we ensure that the step size condition given by Σ is fulfilled, see Section 2. The resulting rectification is illustrated by Figure 3.3 (right). Using the rectified

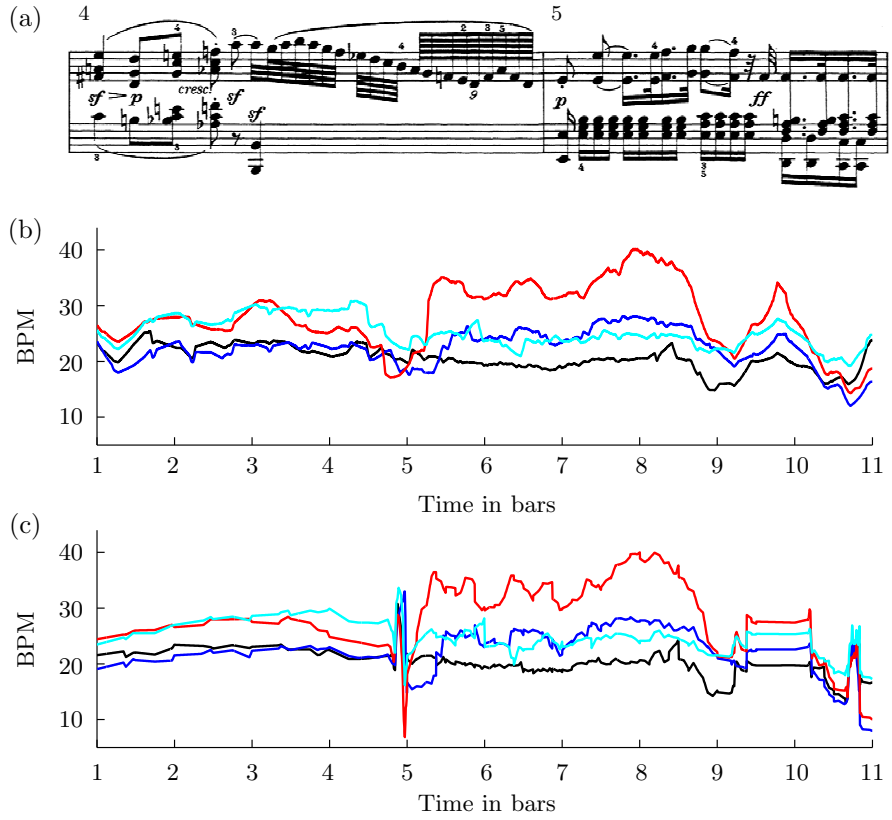

Figure 3.5. Tempo curves of four different interpretations played by different pianists of the first ten bars (slow introductory theme marked *Grave*) of Beethoven's Pathétique Sonata Op. 13. **(a)** Score of bars 4 and 5. **(b)** Tempo curves τ_w^{FWR} for $w \propto 3$ seconds. **(c)** Tempo curves τ_v^{AW} for $v = 10$ IOIs.

alignment path, we then compute the tempo curve using a fixed window size $w \in \mathbb{N}$ as described in Section 3.2.1. The resulting tempo curve is denoted by τ_w^{FWR}. This third approach, as our experiments show, generally yields more robust and accurate tempo estimations than the other two approaches.

3.3 Experiments

In this section, we first discuss some representative examples and then report on a systematic evaluation based on temporally warped music. In the following, we specify the window size w in terms of seconds instead of samples. For example, by writing $w \propto 3$ seconds, we mean that $w \in \mathbb{N}$ is a window size with respect to the feature rate corresponding to 3 seconds of the underlying audio.

In our first example, we consider Beethoven's Pathétique Sonata Op. 13. The first ten bars correspond to the slow introductory theme marked *Grave*. For these bar, Figure 3.5 (b) shows the tempo curves τ_w^{FWR} for four different performances using the combined strategy with a window size $w \propto 3$ seconds. From these curves, one can read off global and local tempo characteristics. For example, the curves reveal the various tempi chosen by the

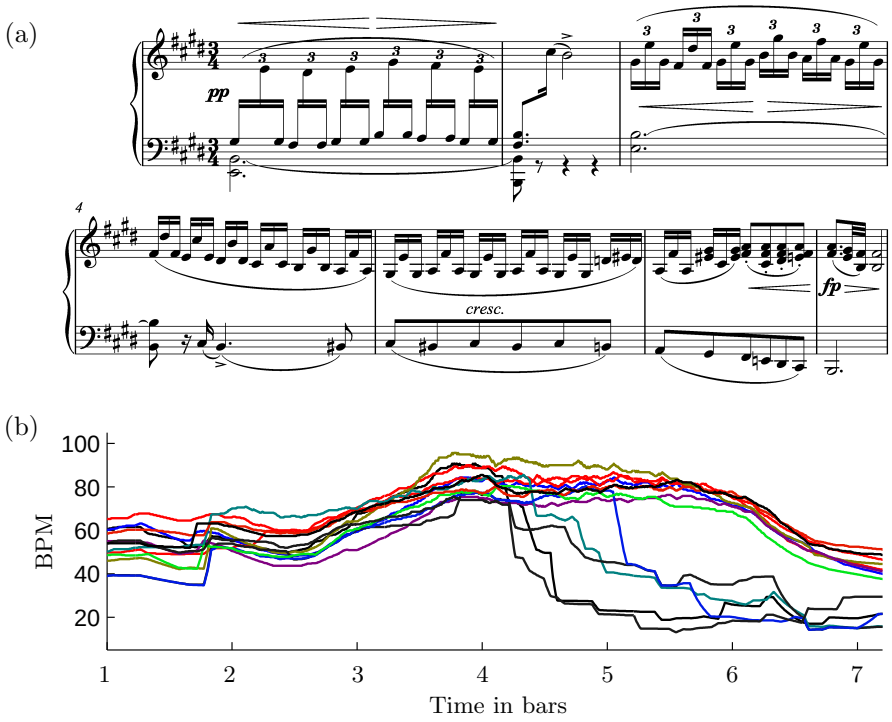

Figure 3.6. Tempo curves of 13 different performances of the beginning of the Schubert Lied *Der Lindenbaum*. **(a)** Score of bars 1 to 7. **(b)** Tempo curves τ_w^{FWR} for $w \propto 3$ seconds.

pianists, ranging from roughly 20 to 30 BPM. One of the pianists (red curve) significantly speeds up after bar 5, whereas the other pianists use a more balanced tempo throughout the introduction. It is striking that all four pianists significantly slow down in bar 8, then accelerate in bar 9, before slowing down again in bar 10. Musically, the last slow-down corresponds to the fermata at the end of bar 10, which concludes the *Grave*. Similarly, the curves indicate a ritardando in all four performances towards the end of bar 4. In this passages, there is a run of 64^{th} notes with a closing nonuplet, see Figure 3.5 (a). Using a fixed window size, the ritardando effect is smoothed out to a large extent, see Figure 3.5 (b). However, having many consecutive note onsets within a short passage, the ritardando becomes much more visible when using tempo curves with an onset-dependent adaptive window size. This is illustrated by Figure 3.5 (c), which shows the four tempo curves τ_v^{AW} with $v = 10$ IOIs.

As a second example, we consider the Schubert Lied *Der Lindenbaum* (D. 911 No. 5). The first seven bars (piano introduction) are shown in Figure 3.6 (a). Using the combined strategy with a window size $w \propto 3$ seconds, we computed tempo curves for 13 different interpretations, see Figure 3.6 (b). As shown by the curves, all interpretations exhibit an accelerando in the first few bars followed by a ritardando towards the end of the introduction. Interestingly, some of the pianists start with the ritardando in bar 4 already, whereas most of the other pianists play a less pronounced ritardando in bar 6. These examples indicate that our automatically extracted tempo curves are accurate enough for revealing interesting performance characteristics.

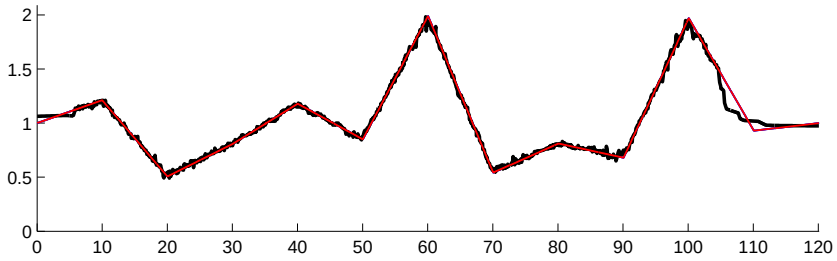

Figure 3.7. Piecewise linear ground-truth tempo curve (red) and computed tempo curves (black).

In view of a more quantitative evaluation, we computed tempo curves using different approaches and parameters on a corpus of harmony-based Western music of various genres. To allow for a reproduction of our experiments, we used pieces from the RWC music database [22]. In the following, we consider 15 representative pieces, which are listed in Table 3.1. These pieces include five classical piano pieces, five classical pieces of various instrumentations (full orchestra, strings, flute, voice) as well as five jazz pieces and pop songs. To automatically determine the accuracy of our tempo extraction procedures, we temporally modified MIDI files for each of the 15 pieces. To this end, we generated continuous piecewise linear tempo curves τ^{GT}, referred to as *ground-truth tempo curves*. These curves have a constant slope on segments of roughly 10 seconds of duration, where the slopes are randomly generated either using a value $v \in [1:2]$ (corresponding to an accelerando) or using a value $v \in [1/2:1]$ (corresponding to a ritardando). These values cover a range of tempo changes of $\pm 100\%$ of the reference tempo. Intuitively, the ground-truth tempo curves simulate on each segment a gradual transition between two tempi to mimic ritardandi and accelerandi. For an example, we refer to Figure 3.7. We then temporally warped each of the original MIDI files with respect to a ground-truth tempo curve τ^{GT} and generated from the modified MIDI file an audio version using a high-quality synthesizer. Finally, we computed tempo curves using the original MIDI files as reference and the warped audio versions as performances.

To determine the accuracy of a computed tempo curve τ, we compared it with the corresponding ground-truth tempo curve τ^{GT}. Here, the idea is to bar deviations by *scale* rather than by *absolute value*. Therefore, as distance function, we use the average multiplicative difference and standard deviation (both measured in percent) of τ and τ^{GT}. More precisely, we define

$$\mu(\tau, \tau^{\mathrm{GT}}) = 100 \cdot \frac{1}{N} \cdot \sum_{n=1}^{N} \big(2^{|\log_2(\tau(n)/\tau^{\mathrm{GT}}(n))|} - 1\big).$$

Similarly, we define the standard deviation $\sigma(\tau, \tau^{\mathrm{GT}})$. For example, one obtains $\mu(\tau, \tau^{\mathrm{GT}}) = 100\%$ in the case $\tau = 2 \cdot \tau^{\mathrm{GT}}$ (double tempo) and in the case $\tau = \frac{1}{2} \cdot \tau^{\mathrm{GT}}$ (half tempo). Similarly, a computed tempo of 110 BPM or 90.9 BPM would imply a mean error of $\mu = 10\%$ assuming a ground-truth tempo of 100 BPM.

In a first experiment, we computed the curves τ_w^{FW} and τ_w^{FWR} with $w \propto 4$ seconds as well as τ_v^{AW} with $v = 10$ IOIs for each of the 15 pieces. Table 3.1 shows the mean error

RWC ID (Comp./Int., Instr.)	**FW**		**AW**		**FWR**	
	μ	σ	μ	σ	μ	σ
C025 (Bach, piano)	3.29	7.30	2.60	5.05	1.59	2.86
C028 (Beethoven, piano)	3.24	6.98	6.36	21.14	2.66	6.72
C031 (Chopin, piano)	3.32	7.72	2.77	4.76	1.75	3.42
C032 (Chopin, piano)	2.54	4.17	3.05	4.67	1.56	2.34
C029 (Schumann, piano)	4.52	8.86	4.18	5.97	2.44	5.13
C003 (Beethoven, orchestra)	4.20	5.39	10.58	22.97	3.56	4.79
C015 (Borodin, strings)	2.44	2.85	4.68	9.85	2.25	2.71
C022 (Brahms, orchestra)	1.70	1.95	2.41	2.96	1.31	1.66
C044 (Rimski-K., flute/piano)	1.62	2.59	2.47	4.27	1.61	2.58
C048 (Schubert, voice/piano)	2.61	3.27	3.95	7.76	2.07	2.98
J001 (Nakamura, piano)	1.44	1.87	1.44	2.43	1.03	1.59
J038 (HH Band, big band)	2.24	2.96	3.20	5.41	1.91	2.74
J041 (Umitsuki, sax/bass/perc.)	1.88	2.40	3.75	4.69	1.72	2.34
P031 (Nagayama, electronic)	2.01	2.42	8.35	14.89	1.94	2.39
P093 (Burke, voice/guitar)	2.50	3.26	6.21	14.74	2.34	3.13
Average over all	2.64	4.27	4.40	8.77	1.98	3.16

Table 3.1. Tempo curve evaluation using the approaches FW and FWR (with $w \propto 4$ seconds) and AW (with $v = 10$ IOIs). The table shows for each of the 15 pieces the mean error μ and standard deviation σ (given in percent) of the computed tempo curves and the ground truth tempo curve. For generating the ground-truth tempo curves, MIDI segments of 10 seconds were used.

μ and standard deviation σ between the computed tempo curves and the ground truth tempo curves. For example, for the Schubert Lied *Der Lindenbaum* with identifier C048, the mean error between the computed tempo curve τ_w^{FW} and the ground-truth tempo τ^{GT} amounts to 2.61%. This error decreases to 2.07% when using the FWR-approach based on the rectified alignment path. Looking at the average mean error over all pieces, one can notice that the error amounts to 2.64% for the FW-approach, 4.40% for the AW-approach, and 1.98% for the FWR-approach. For example, assuming a tempo of 100 BPM, the last number implies a mean difference of less than 2 BPM between the computed tempo and the actual tempo.

In general, the FWR-approach yields the best tempo estimation, whereas the AW-approach often produces poorer results. Even though the onset information is of crucial importance for estimating local tempo nuances, the AW-approach relies on accurate alignment paths that correctly align the note onsets. Synchronization approaches as described in [13] can produce highly accurate alignments in the case of music with pronounced note attacks. For example, this is the case for piano music. In contrast, such information is often missing in string or general orchestral music. This is the reason why the purely onset-based AW-strategy yields a relatively poor tempo estimation with a mean error of 10.58% for Beethoven's Fifth Symphony (identifier C003). On the other hand, using a fixed window size without relying on onset information, local alignment errors cancel each other out, which results in better tempo estimations. E. g., the error drops to 3.56% for Beethoven's Fifth Symphony when using the FWR-approach.

Finally, we investigated the dependency of the accuracy of the tempo estimation on the window size. We generated strongly fluctuating ground-truth tempo curves using MIDI

w [sec]	**FW**		**FWR**		v [IOI]	**AW**	
	μ	σ	μ	σ		μ	σ
1	10.62	49.88	5.58	12.47	2	14.50	31.00
2	5.37	14.21	3.58	6.16	4	9.54	23.44
3	4.39	6.90	3.42	5.34	6	7.34	17.34
4	4.62	6.52	3.99	5.74	8	6.18	12.99
5	5.48	7.08	5.06	6.63	10	5.65	10.66
6	6.79	8.02	6.52	7.74	12	5.46	9.48
7	8.40	9.19	8.22	9.00	16	5.54	8.20
8	10.15	10.51	10.03	10.38	20	5.98	8.09

Table 3.2. Tempo curve evaluation using the approaches FW, AW, and FWR with various window sizes w (given in seconds) and v (given in IOIs). The table shows the average values over all 15 pieces, see Table 3.1. For generating the ground-truth tempo curves, MIDI segments of 5 seconds were used.

segments of only 5 seconds length (instead of 10 seconds as in the last experiment). For the corresponding synthesized audio files, we computed tempo curves for various window sizes. The mean errors averaged over all 15 pieces are shown in Table 3.2. The numbers show that the mean error is minimized when using medium-sized windows. E. g., in the FWR-approach, the smallest error of 3.42% is attained for a window size of $w \propto 3$ seconds. Actually, the window size constitutes a trade-off between robustness and temporal resolution. On the one hand, using a larger window, possible alignment errors cancel each other out, thus resulting in a gain of robustness. On the other hand, sudden tempo changes and fine agogic nuances can be recovered more accurately when using a smaller window.

3.4 Potential and Limitations of Automated Methods

The presented automated approach for extracting tempo parameters from audio recordings (see Section 3.2) yields accurate and robust estimations of the overall tempo progression. However, when dealing with automated methods one generally has to take into account limitations of the employed methods. A meaningful use of automated methods requires the knowledge about such weaknesses and limitations. In particular, in an interdisciplinary context, where musicologists apply automated methods to musicological tasks, the awareness of limitations of the underlying methods is indispensable. In this section, we investigate to which extent our approach for extracting temporal information may support musicologists in analyzing recorded performances. Performing a case study on the first movement of Beethoven's Sonata Op. 57, the so-called Appassionata, we exemplarily demonstrate in which cases automated procedures for extracting temporal parameters reach their limits.

Figure 3.8 shows the automated extraction of tempo curves in a schematic overview, where three different recorded performances (played by the pianists Duis, Barenboim and Brendel) of the first four bars of Beethoven's Appassionata are considered. For the tempo computation in this case study we choose the FWR-approach (described in Section 3.2.3), where first the alignment path is rectified by using onset information and then the tempo curve is computed using a fixed window size $w \propto 8$ seconds.

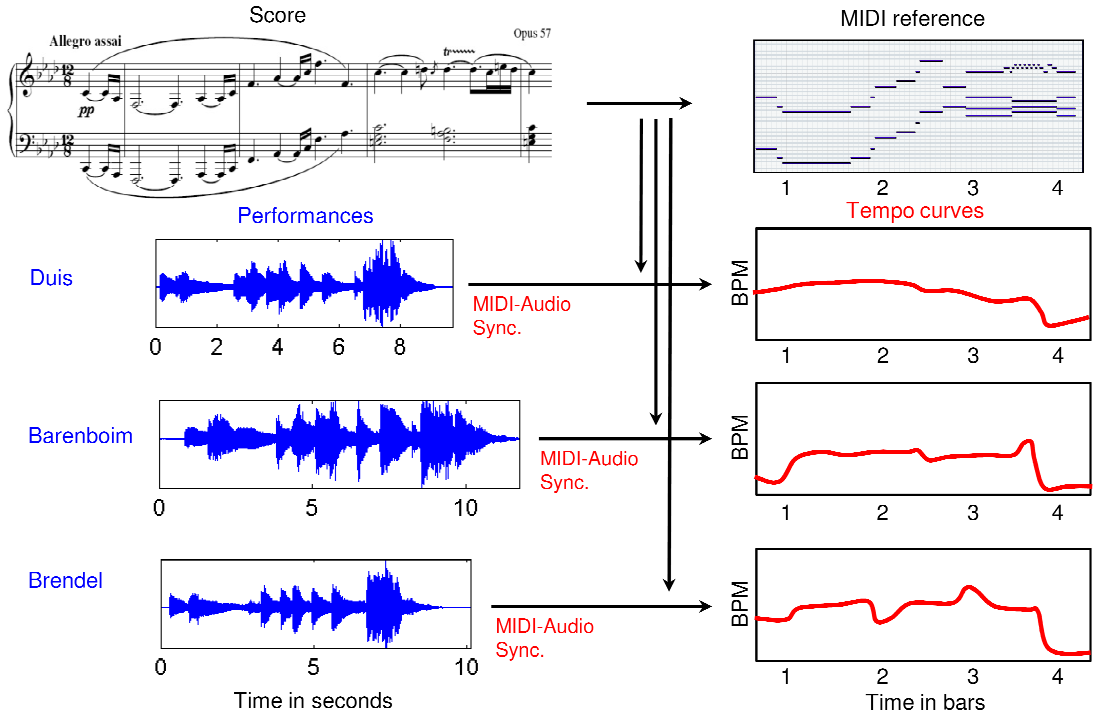

Figure 3.8. Schematic overview of the automated procedure for extracting tempo curves for three different recorded performances of the first movement of Beethoven's Appassionata (bb. 1-3).

The tempo curves reveal global and local characteristics of the different recorded performances. For example, it is obvious that Barenboim plays in the first bars much slower than the two other pianists. Furthermore, the tempo curves show that all pianists significantly slow down towards the end (bb. 3-4). In fact, the score reveals the musical reasons for this ritardando: In bar 4 the first appearance of the main theme ends with a half cadence, where the diminished seventh chord (b. 3) resolves into the dominant (b. 4), which is supported by the ritardando of all pianists.

As a further example Figure 3.9 shows tempo curves for five different recorded performances of the beginning of the Appassionata (bb. 1-24). Obviously, the overall tempi of the different performances extremely vary. While Gieseking plays the first movement with an average tempo of approximately 100 BPM, Gould chooses a much slower overall tempo, namely 60 BPM. Besides the overall tempo, the tempo curves reveal local tempo variations of the pianists. For example, all pianists accelerate in bar 17. This agogic freedom can be explained by the ascending chord line which reaches its climax in bar 18, significantly contributing to the `F` minor character of the main theme.

As the examples show, automatically derived tempo curves reflect very well the approximate tempo of the considered recorded performance. However, in which cases do such automated methods reach their limits? Figure 3.10 shows tempo curves for the first movement of the Appassionata where two different recorded performances (Gieseking and Zilberstein) are considered. Here, for each recorded performance an automatically derived tempo curve (red) as well as a manually generated tempo curve (black) are visualized. Ob-

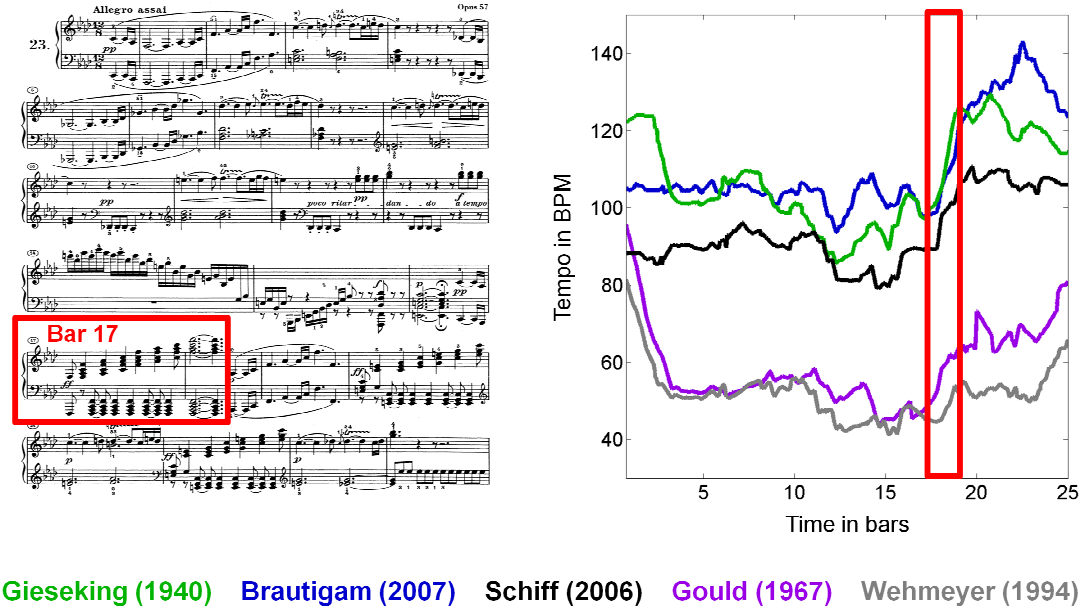

Figure 3.9. Automatically derived tempo curves for five different recorded performances of the first movement of Beethoven's Appassionata (bb. 1-24).

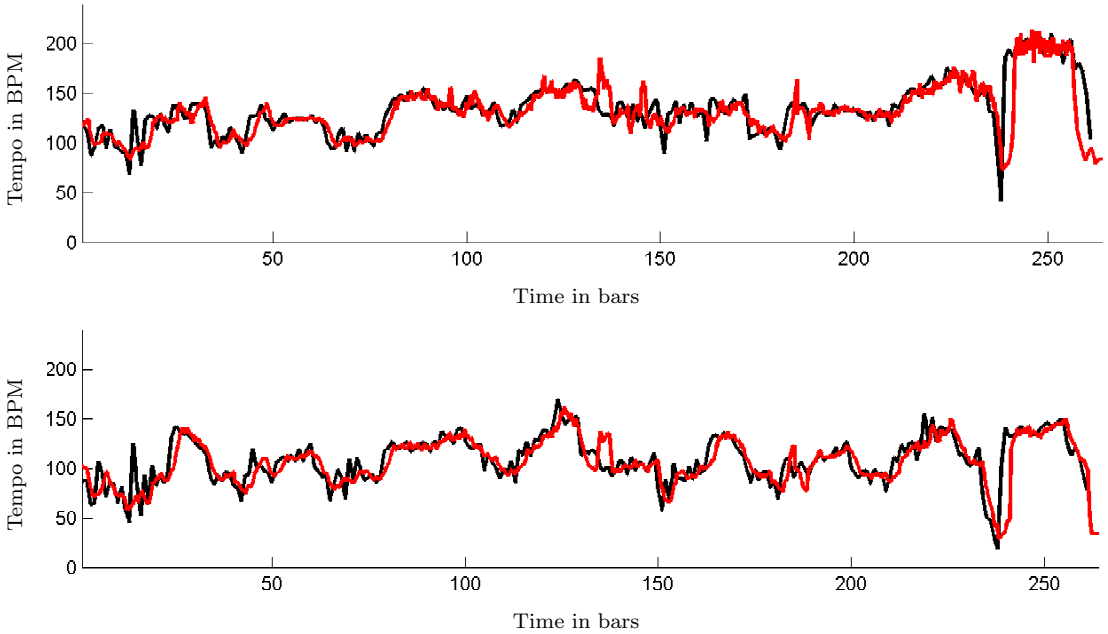

Figure 3.10. Tempo curves for recorded performances of Gieseking (top) and Zilberstein (bottom) of the first movement of Beethoven's Appassionata. For each recorded performance an automatically derived tempo curve (red) as well as a manually generated tempo curve (black) is visualized.

viously, the tempo curves for Gieseking and Zilberstein behave very similar. Both pianists seem to share a common tendency of tempo shaping. However, differences appear in the use of agogics, which is characteristic for the personal style of the performer. Such agogic deviations of the tempo are often expressed in fine tempo nuances. In particular, capturing these tempo nuances is difficult for automated methods. This becomes apparent if one

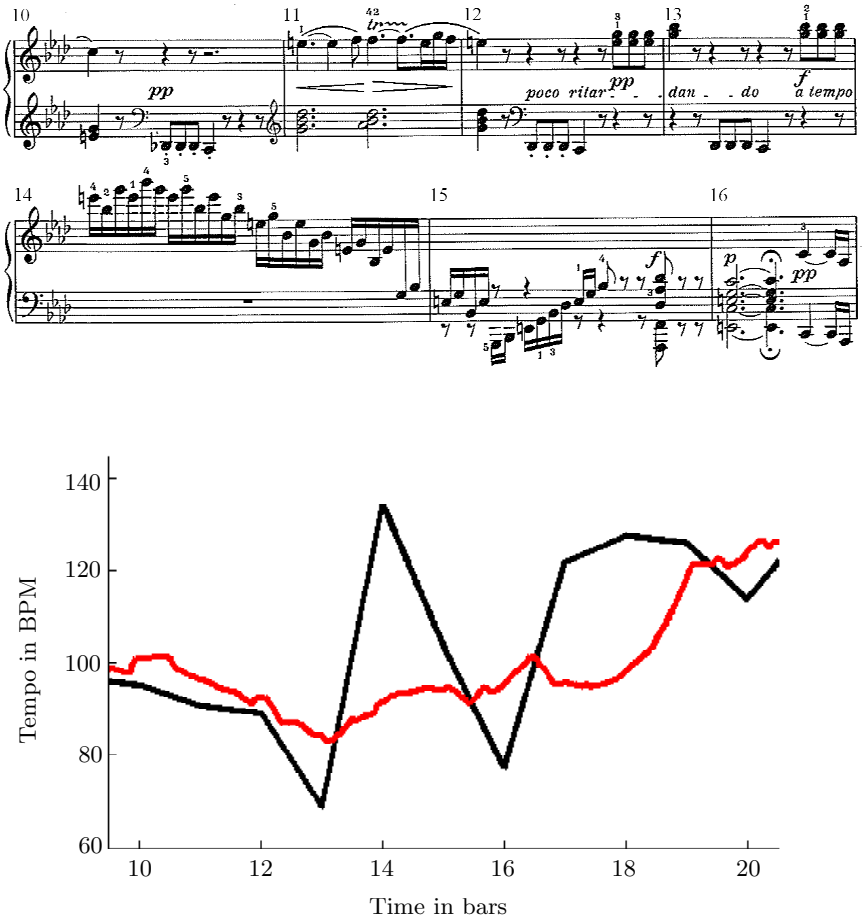

Figure 3.11. Excerpt of the tempo curves for Gieseking's performance (bb. 10-20). An automatically derived tempo curve (red) as well as a manually generated tempo curve (black) is visualized.

compares the automatically derived tempo curves with the manually generated curves. While the global tempo is reflected well by the automatically derived tempo curve, local tempo deviations, which are visible in the manually generated curve, are not precisely captured by the automatically derived curve.

In the following, we exemplarily discuss by means of two excerpts of Gieseking's recording in which situations automated methods for deriving tempo curves tend to fail. Figure 3.11 shows an excerpt of the tempo curves for bars 10-20. One directly notices that the manually generated curve exhibits large deviations of the global tempo which can be explained by the following musical reasons: For example, the slow down from 90 to 70 BPM in bars 12-13 refers to the fading fate motif, being commented in the score with *poco ritardando*. Subsequently, bars 13-14 are characterized by a rapid increase in tempo from 70 to 135 BPM, followed by a decrease of tempo in bars 14-16 from approximately 135 to 75 BPM. The fate motif leads into a rapidly executed 16th figuration (commented with *a tempo*), which leads into a half cadence with fermata in bar 16. Then, a variant of the main theme entries being accompanied by an increase in tempo from approximately 75 to 120 BPM in bars 16-17. While these sudden tempo deviations (ranging from 70 to 135 BPM) can be

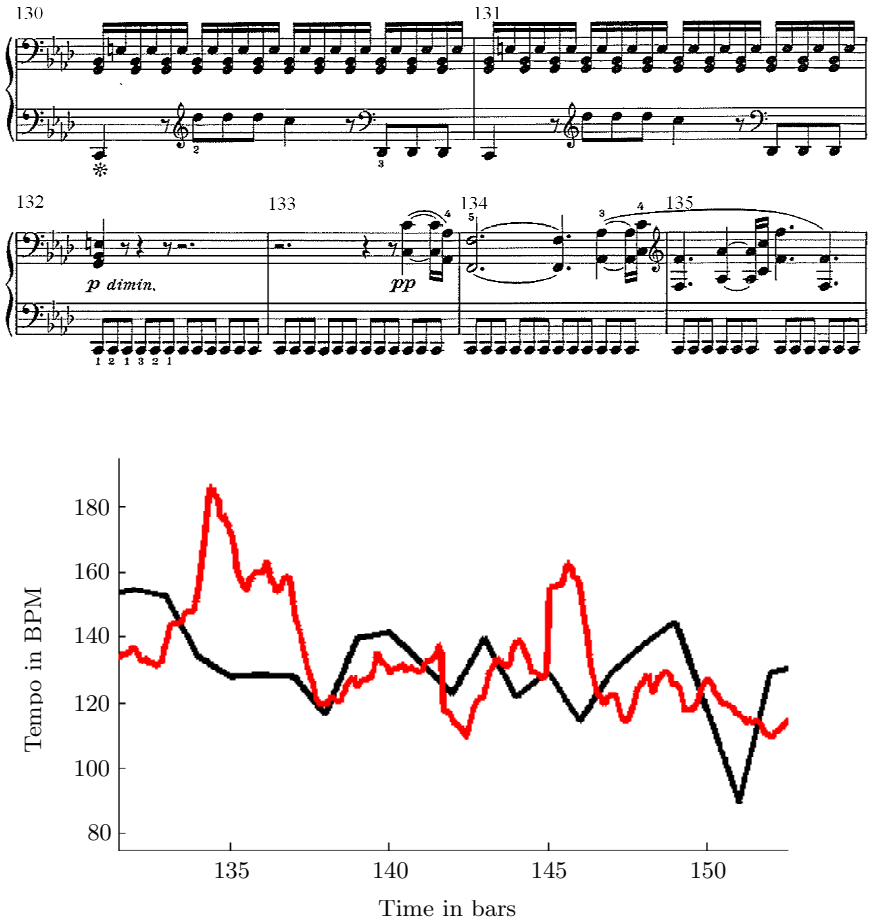

Figure 3.12. Excerpt of the tempo curves for Gieseking's performance (bb. 132-152). An automatically derived tempo curve (red) as well as a manually generated tempo curve (black) is visualized.

directly observed in the manually generated curve, they are completely smoothed out in the automatically derived curve (indicating a medium tempo of approximately 100 BPM).

This is a fundamental problem when dealing with automated methods. As already described in Section 3.3, the window size constitutes a delicate tradeoff between accuracy and robustness. Choosing a relatively large window $w \propto 8$ seconds in the present case study, the fine agogic nuances in bars 10-20 can not be captured by the automatically derived tempo curve.

As a second problem one has to deal with inaccuracies in the underlying synchronization procedure which may result in inaccuracies in the tempo computation. Synchronization problems often appear in the case that the respective musical passage has a uniform or repetitive character, for example if the same harmony is present for a long section or if a note repetition appears, e. g., in the form of a basso ostinato. In such cases the MIDI-audio synchronization may fail due to the uniformity of the musical material. Figure 3.12 shows an example, where inaccuracies in the underlying synchronization lead to inaccurate values in the tempo curve. The automatically derived curve obviously oscillates around the man-

ually generated curve. This typically indicates a failure of the underlying synchronization procedure. A closer look at the score reveals that here indeed the previously described uniformity of the musical material exists. In bars 130-131 the constantly present tremolo of the right hand is alternately complemented to a dominant seventh chord on C and a dominant seventh chord on E♭ by the fate motif of the left hand. Bars 132 and 133 are merely filled with a C appearing as basso ostinato, before over this bass the recapitulation enters with the main theme in octaves. Similarly, the subsequent bars are characterized by a basso ostinato over which constantly present harmonies sound. This explains the inaccuracies in the underlying synchronization which are reflected as subsequent errors in the automatically derived tempo curve.

3.5 Conclusions

In this chapter, we have introduced automated methods for extracting tempo curves from expressive music recordings by comparing the performances with neutral reference representations. In particular when using a combined strategy that incorporates note onset information, we obtain accurate and robust estimations of the overall tempo progression. Here, the window size constitutes a delicate trade-off between susceptibility to alignment errors and sensibility towards timing nuances of the performance. In practice, it becomes a difficult problem to determine whether a given change in the tempo curve is due to an alignment error or whether it is the result of an actual tempo change in the performance. Here, one idea for future work is to use tempo curves as a means for revealing problematic passages in the music representations where synchronization errors may have occurred with high probability. Furthermore, it is of crucial importance to further improve the temporal accuracy of synchronization strategies. This constitutes a challenging research problem in particular for music with less pronounced onset information, smooth note transitions, and rhythmic fluctuation.

Finally, we exemplarily indicated the potential of the presented automated methods for the musicological analysis of recorded performances. By the automatization of measuring and annotation procedures computer science may efficiently support musicologists, in particular, in view of large music data collections. However, every process of automatization has certain limitations. As a consequence, the critical treatment of the automatically computed results is indispensable for musicologists when using computer-based methods.

Part II

User Interaction

Chapter 4

Interfaces in MIR

4.1 Related Work

In the field of MIR, the development of technologies and interfaces for music exploration and analysis has been an active research area [9, 11, 20, 21, 54, 80]. In the following, we give a short overview of some selected MIR user interfaces. The Sonic Visualiser is a system for viewing and analyzing the contents of music audio files [80]. For example, it enables for loading audio files and viewing the waveform or other visualizations as e. g. the spectrogram along with the audio. There exist various Vamp plugins for the Sonic Visualiser. The first to be mentioned here is the MATCH plugin, a system for temporally aligning several versions of a piece of music [11]. MATCH allows for switching from one version to the other and in this way for comparing different recorded performances of the same piece of music. Furthermore, the Chordino plugin allows for extracting chords from an audio file and displaying the harmonic information along with it [7]. The SmartMusicKiosk represents an interface for intra-document navigation [20, 21]. It automatically detects the chorus or other key parts of a pop song resulting in a visualisation of the song structure. This enables the user to directly jump to the desired part of the song. The Songle interface is a web service which allows users to retrieve, browse and annotate songs on the web [23]. The interface displays automatically derived annotations along with a given song. By navigating within the song the user has the possibility to correct errors in the automatically derived annotations. Finally, the sheet music interface described in [9] automatically aligns note events within a sheet music representation to corresponding note events within an audio recording. In this way, the interface highlights the corresponding bars within the sheet music while playing back the audio recording. Furthermore, the user can jump to a certain position in the audio by clicking on the corresponding bar.

4.2 The Interpretation Switcher

The *SyncPlayer* system is an advanced audio player for multimodal presentation, browsing, and retrieval of music data [16]. One of the available plugins for the SyncPlayer, referred to as *Interpretation Switcher*, is the MIR interface which is of central importance

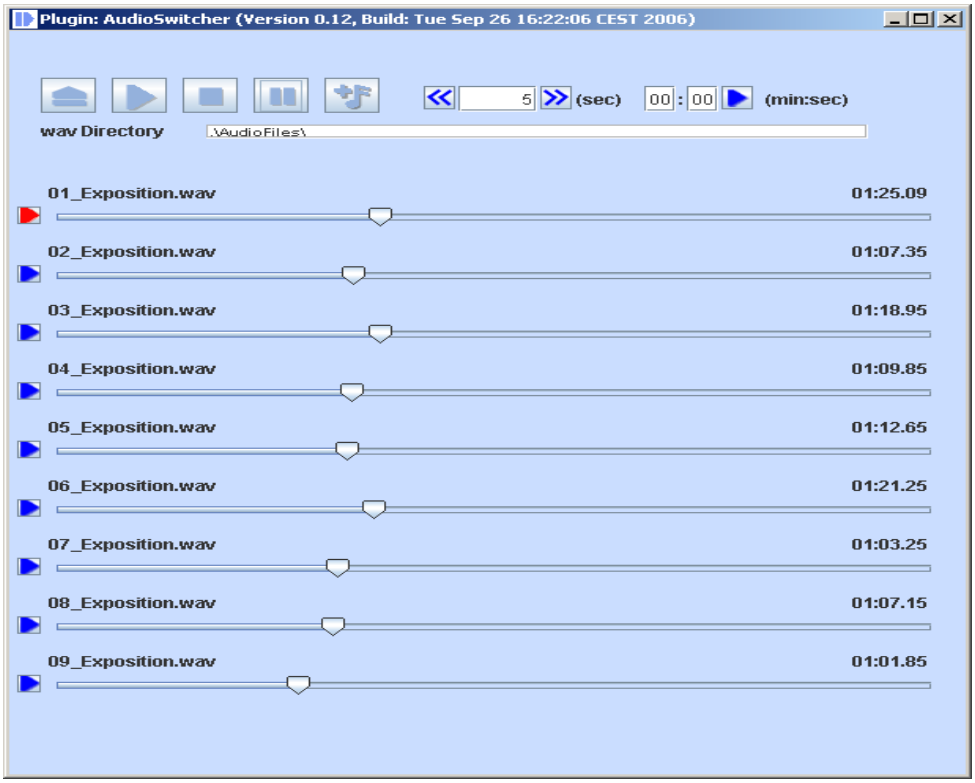

Figure 4.1. Instance of the Interpretation Switcher plugin of the SyncPlayer for synchronous playback of different audio recordings of the same piece of music. In this example, nine different recordings of the exposition of Beethoven's Pathétique Sonata are opened.

in Chapters 5 and 6, see Figure 4.1. It provides a similar functionality as the previously described MATCH plugin for the Sonic Visualiser. The Interpretation Switcher allows the user to select several recordings of the same piece, which have previously been synchronized [54]. Each of the selected recordings is represented by a slider bar indicating the current playback position with respect to the recording's particular time scale. The audio recording that is currently used for playback, in the following referred to as reference recording, is represented by a red marker. The slider of the reference recording moves at constant speed while the sliders of the other recordings move according to the relative tempo variations with respect to the reference. The reference recording may be changed at any time simply by clicking on the respective marker located on the left of each slider. The playback of the new reference recording then starts at the time position that musically corresponds to the last playback position of the former reference. In this way, the user may listen to a specific recording by activating a slider bar and then, at any time during playback, seamlessly switch to any of the other versions (inter-document navigation). One can also jump to any position within any of the recordings by directly selecting a position of the respective slider, which automatically triggers a switch of the reference to the respective recording. Note that the SyncPlayer provides many more functionalities comprising plugins for inter- and intra-document browsing and retrieval as well as data visualization and analysis.

Chapter 5

Introducing the Interpretation Switcher Interface to Music Education

In the field of MIR, great efforts have been directed towards the development of technologies and interfaces that allow users to access and explore music on an unprecedented scale. On the other hand, musicians and music teachers are often still skeptical about the benefits of computer-based methods in music education. In this chapter, we report on an experiment conducted at the University of Music Saarbrücken with the goal to introduce a novel MIR user interface, referred to as Interpretation Switcher, to music education and to get feedback from music experts. To this end, we asked nine music students to analyze different performances of the same piece of music according to a well designed questionnaire, using the novel switching functionality of our interface. Doing so, we not only tested and evaluated our interface in a setting of practical relevance, but also indicated the potential of MIR methods in music education. The results presented in this chapter have been published in [33, 55].

This chapter is organized as follows. First, we give an overview on related work (Section 5.1). Afterwards, we describe the setup of our experiment (Section 5.2) before presenting the evaluation in Section 5.3 and concluding in Section 5.4.

5.1 Related Work

Computers have become an indispensable tool for storing, processing, and generating music. Even though computer-based methods and interfaces are ubiquitously used for music synthesis, there is still a reluctance in using computers for music analysis and music education. Research in computer-assisted music education already started at the end of the 1960s mainly in the USA and the United Kingdom [5, 31, 79, 81, 25]. For example, computers have served as a tool for creative music-making. Furthermore, the method of Computer-Assisted-Instruction (CAI), where students are taught a particular skill by a computer, has been applied in areas like music theory or aural training. Various

studies have been conducted to investigate the effect of CAI-based methods within music education [79, 31], and the usefulness of such methods seems to be a controversial issue.

In the field of music information retrieval (MIR), the development of technologies and interfaces for music exploration and analysis has been an active research area [9, 11, 20, 54, 80]. However, these technologies and interfaces are often evaluated in the own lab environment, where people are familiar with computers. Though, for building up MIR systems of practical relevance one needs broader feedback, in particular from music experts. Hence, for a user-centered analysis, it is necessary to conduct "real" user studies that ensure a natural setting [42].

There are many applications in the context of music education that may benefit from the above mentioned MIR-based technologies and interfaces. However, musicians and music teachers are often still reluctant in using novel computer-assisted methods and novel MIR interfaces in their lessons. Furthermore, many of the available MIR interfaces are still too complicated lacking the necessary user-friendliness and robustness to be operable by non-experts. Under such circumstances, it remains a challenge to raise the interest of music educators for using, testing, and participating in the development of novel MIR interfaces and for discussing possible application scenarios.

5.2 Experimental Setup

5.2.1 Piece of Music

For our experiment, we chose the first movement of Beethoven's Pathétique Sonata Op. 13. This piece of music appeared to be a good choice for various reasons. Firstly, for the Pathétique Sonata numerous detailed descriptions and scientific literature exists. Secondly, being a very popular and famous work, the Pathétique belongs to the standard repertoire of many pianists. Hence, there are numerous audio recordings for this piece. The third and most important reason for choosing the Pathétique is that it is very rich in contrast concerning tempo as well as dynamics.

To make the latter point clear, we describe the Pathétique's exposition in more detail. Beginning with the slow introductory theme marked *Grave* (bb. 1-10, see Figure 5.1 (a)), the work starts very dramatically. The introduction is characterized by its contrasts in dynamics—fortissimo passages are followed by subito piano and vice versa. This contrast in dynamics is underlined by contrasts in rhythmics, articulation, and atmosphere. Ending with the chromatic run, the introduction leads in the first theme (bb. 11-27, see Figure 5.1 (b)) of the sonata which is characterized by the tremolo in octavos in the left hand giving it a dramatic touch. In contrast to the dramatic first theme, the second theme (bb. 51-88, see Figure 5.1 (c)) sounds more playful. It is based on the call and response principle and is characterized by a play with articulation.

As described above, one can find many contrasting elements in the exposition of the first movement of the Pathétique: Contrasts in the shaping of the two themes, contrasts in dynamics, and contrasts in atmospheres. In addition, there is an abrupt change in tempo at the beginning of the first theme (b. 11), where the introductory *Grave* leads in the

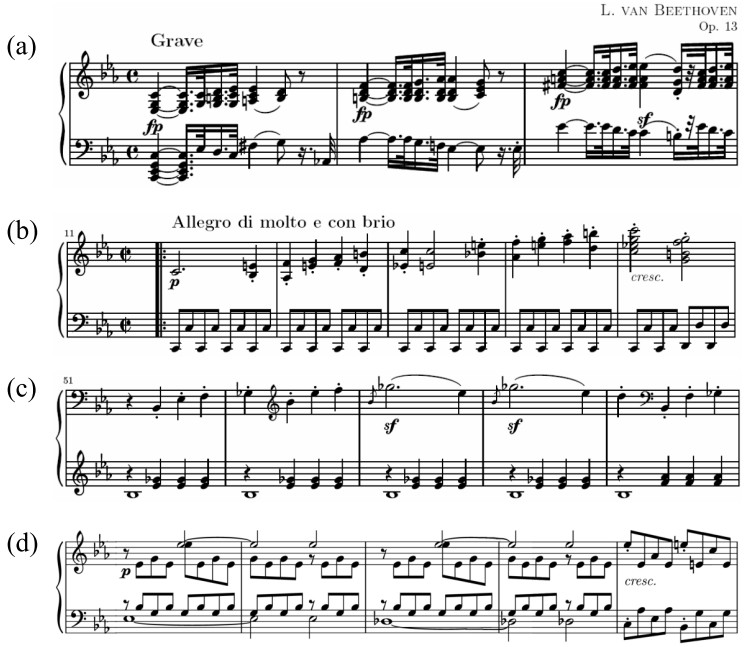

Figure 5.1. First movement of Beethoven's Pathétique Sonata Op. 13 (score obtained from (Mutopia Project, 2009)). **(a)** Beginning of the introduction (Section A, bb. 1 ff.) **(b)** First theme (bb. 11 ff.) **(c)** Second theme (bb. 51 ff.) **(d)** Section B (bb. 89 ff.)

actual exposition marked *Allegro di molto e con brio*. Because of its musical richness, the Pathétique offers the pianist a wide range of possibilities for shaping the piece with respect to dynamics, tempo, and agogics. Therefore, this piece is very well suited in view of the musical evaluation within our experiment.

5.2.2 Performance and Recording Setup

The recordings of our experiment were performed at the University of Music Saarbrücken. Nine students of different study paths from the piano class of Prof. Thomas Duis were asked to play the first movement of the Pathétique. Being in several training states, they were on different performance levels. All students played on the same instrument under the same recording conditions on two different days (Friday, 06.02.2009 and Monday, 09.02.2009). In the recording sessions, only the performer, the technical staff, and the scientific investigators were present in the room—the other performers were not allowed to listen to their fellow students. Using two microphones, we did not achieve the quality of a recording studio. However, we obtained audio recordings of a sufficient quality in view of our experiments. Furthermore, using a Yamaha Disklavier, we also generated MIDI data along with the audio recordings. Actually, the MIDI files were not used in our experiments, but as mentioned before they are useful for later projects.

5.2.3 MIR User Interface

The MIR interface used in the following experiment is the *Interpretation Switcher*, which is described in Section 4.2. In our experiment, we restrict ourself basically to the switching functionality with the motivation to keep the interface as simple and intuitive as possible to avoid any rejections from the users. As Figure 4.1 shows, the Interpretation Switcher looks like a standard audio player with the only difference that more than one slider control bar is available. After a short explanation of the main switching functionality, none of the students reported on difficulties in using our interface.

5.2.4 Survey and Questionnaire

Subsequent to the last recording session, the nine different performances were aligned and integrated in our Interpretation Switcher. Then, we conducted our survey in the evening of the second recording day (Monday, 09.02.2009). Eight music students participated in the survey, seven of whom were also among the nine performers. The different interpretations were anonymised within the interface and the participants listened to the recordings for the first time.

Each participant was provided with a computer running the Interpretation Switcher and with earphones. After a short introduction of the interface's switching functionality, the participants received a questionnaire having one hour for answering the questions. This questionnaire consisted of two main parts. In the first part, the students had to listen, to compare, and to rate the nine different interpretations with respect to various performance aspects. Here, the questions were designed in such a way that the students naturally started to use the switching functionality of the interface, thus getting familiar with the Interpretation Switcher in a concrete application of musical relevance. In the second part, they were then asked to give feedback on the usefulness and operability of the interface itself.

The questions of the first part of the questionnaire referred to different sections of the first movement of the Pathétique. As a kind of warming up, we started with a short section (Section A), which only consisted of the first three bars, see Figure 5.1 (a). This section was cut out from the nine aligned performances and presented to the students by the Interpretation Switcher interface. Even being rather short, Section A already offers the pianists a wide range of interpretation so that the comparison of the different performances constitutes a musically interesting task. In the first question (A1), the participants had to rate the nine different interpretations of Section A with respect to the three musical aspects dynamics, articulation, and agogics. Here, the rating scale ranged between 1 and 10, where 1 means poor and 10 excellent. In addition, they had to rate their total impression of this section's performances using the same scale. Afterwards, in question A2, they had to identify their own interpretation (if applicable) only by means of Section A. Then, the performances of Section A were closed and a different section (Section B) was presented by the Interpretation Switcher to them. Here, Section B consisted of the technically more involved bb. 89-100, see Figure 5.1 (d). The students then had to answer corresponding questions (B1, B2).

At the beginning of the questionnaire, the students were confronted with different performances of relatively short sections. Here, only switching between the performances was required to properly answer the questions—jumping back and forth within a performance was not necessary. In this way, the students became familiar with the basic switching functionality of the interface. In the next stage, they were presented with the nine performances of the entire exposition. They now had to rate their total impression of the first theme (bb. 11 ff., see Figure 5.1 (b)), of the second theme (bb. 51 ff., see Figure 5.1 (c)), and of the entire exposition (questions E1, E2 and E5). Here the new challenge concerning the use of the Interpretation Switcher was not only to switch between the different performances but also to find the corresponding entry points of the two themes within the recordings. Another task (E3), was to order the nine different interpretations with respect to the tempo (beginning with the slowest, ending with the fastest) in the second theme. With this task the students had to constantly switch between and jump within the performances, being forced to use the functionality of the interface extensively. In question E4, again, they had to identify their own performance (if applicable) now having the entire exposition at their disposal.

After finishing the questions on music aspects, in the second part of the questionnaire the participants were asked to evaluate the Interpretation Switcher interface. Here, the idea was to let the participants first use the interface in an application scenario to gather practical experience without knowing about the final interface evaluation. In the first question (S1), they should rate the user-friendliness and the degree of usability of the Interpretation Switcher on the above described scale from 1 to 10. We then wanted to know if there were any problems while using the interface (S2). Furthermore, the students were asked to comment on possible improvements and to propose additional functionalities they would have liked when working on the first part of the questionnaire (S3). In a last question (S4), they should sketch possible application scenarios where they could imagine to use MIR user interfaces such as the Interpretation Switcher.

5.3 Evaluation

5.3.1 Performance Evaluation

In the first part of the questionnaire, the participants had to analyze and compare the different performances against each other. Table 5.1 presents the results of question A1, where they had to rate the nine different performances of Section A with regard to dynamics, articulation, agogics, and in total. The first row of Table 5.1 shows the number of the respective performance; the values of each column correspond to the respective performance. The second row shows the ratings with regard to dynamics averaged over the eight participants. For example, the first performance was rated with a score of $\mu = 6.63$ on average. The third row shows the standard deviation, which is $\sigma = 1.19$ for the first performance. The following rows of Table 5.1 are to be read in the same fashion. For example, the participants rated the sixth performance on average with $\mu = 6.88$ ($\sigma = 1.36$) with respect to articulation, whereas the overall impression of this performance amounts to $\mu = 6.38$ (σ=1.51). As we can see, the eighth performance was ranked highest with respect to dynamics ($\mu = 6.75$), whereas the second one with respect to articulation ($\mu = 7.00$).

Table 5.1. Evaluation results of question A1 (Figure 5.1 (a)). The average ratings μ along with the standard deviations σ are shown for the nine performances of Section A with regard to various musical aspects.

		1	2	3	4	5	6	7	8	9
Dynamics	μ	6.63	6.25	6.38	6.13	5.75	6.38	5.38	6.75	6.63
	σ	1.19	1.49	1.69	1.36	1.67	1.69	1.77	1.49	1.60
Articulation	μ	6.13	7.00	6.25	6.38	6.13	6.88	5.38	5.88	6.00
	σ	1.81	1.69	1.83	1.30	1.64	1.36	2.20	1.36	1.41
Agogics	μ	6.13	6.50	5.13	6.38	5.63	6.50	4.75	5.75	6.13
	σ	1.73	1.69	2.53	1.92	1.41	1.41	2.12	1.28	0.99
Overall	μ	6.25	6.25	6.00	6.25	5.88	6.38	5.25	6.13	6.00
	σ	1.58	1.67	2.14	1.28	1.25	1.51	1.58	1.55	1.20

Table 5.2. Evaluation results of question B1 (Figure 5.1 (d)).

		1	2	3	4	5	6	7	8	9
Dynamics	μ	5.38	6.38	7.13	6.38	5.63	7.13	5.88	4.75	4.13
	σ	1.41	1.69	1.55	1.69	1.41	1.13	1.36	2.12	1.36
Articulation	μ	5.38	6.00	6.25	6.38	4.88	6.25	5.13	4.63	4.75
	σ	1.85	1.20	1.04	1.60	1.89	1.58	1.89	1.69	2.19
Agogics	μ	5.13	6.00	6.88	6.75	5.13	6.75	5.38	4.88	4.00
	σ	2.42	0.76	0.99	1.39	1.81	1.28	1.51	1.64	1.51
Overall	μ	5.25	5.88	6.75	6.63	5.00	6.63	5.63	5.00	4.25
	σ	1.75	0.99	1.49	1.41	1.51	1.41	1.69	1.51	1.39

The overall rankings for Section A are relatively close together, which may show that the section was too short for giving a well-founded evaluation or that it was played similarly by all students.

Analogously, Table 5.2 shows the results of question B1. Here, the best performance concerning the overall impression is the third one ($\mu = 6.75$), whereas the worst performance is the ninth one ($\mu = 4.25$). Actually, the ninth performance was ranked worst with respect to all musical aspects. The reason for the poor rating is that the performing student struggled significantly with the technically more involved Section B, thus neglecting the musical shaping. This may also explain, why the given scores between the performances differ to a much larger degree for Section B than for Section A. Finally, Table 5.3 presents the results of questions E1, E2 and E5, where only the overall impression had to be rated. Here, the first, third, and second performances were rated best with regard to the first theme ($\mu = 6.88$), the second theme ($\mu = 6.75$), and the entire exposition ($\mu = 7.00$), respectively. Again, the ninth performance was rated worst with regard to all three categories. Interestingly, there does not exist a clear winner performance concerning all different musical aspects and themes.

Table 5.3. Evaluation results of question E1 (first theme, Figure 5.1 (b)), question E2 (second theme, Figure 5.1 (c)), and question E5 (entire exposition).

		1	2	3	4	5	6	7	8	9
1. Theme	μ	6.88	6.25	5.88	5.63	4.88	5.25	5.50	5.25	4.75
	σ	1.13	1.39	1.25	2.13	2.30	2.19	2.45	2.12	1.83
2. Theme	μ	6.25	6.38	6.75	5.63	5.00	5.25	6.38	5.13	4.50
	σ	1.83	1.06	1.39	1.30	1.60	1.04	1.85	1.96	1.60
Exposition	μ	6.50	7.00	6.88	5.63	4.75	5.13	5.50	5.63	4.38
	σ	1.41	1.51	1.46	1.85	1.58	1.81	2.00	1.77	1.60

5.3.2 Interface Evaluation

In the second part of the questionnaire, the students were asked about the operability and usefulness of the Interpretation Switcher. As mentioned before, none of them had serious problems in using the interface, which is also reflected by a high average rating of $\mu = 7.63$ given for the user-friendliness of the interface. Only one of the participants gave a low score of 4 explaining a relatively large standard deviation of $\sigma = 2.07$. As it turned out, the reason for this was that the student was pressured for time and not really in the mood of participating in our experiment. Actually, this student also admitted that she has had no time for properly practicing the piece, resulting in performance number nine with the lowest score, see Table 5.3. Most of the other participants emphasized that they found the handling and functioning of the Interpretation Switcher very intuitive, even music students who have had only little experience with computers. Furthermore, most students found the Interpretation Switcher very useful for tasks such as performance analysis, music comparison, and other analysis tasks. Here, the average rating amounted to $\mu = 7.13$ with standard deviation $\sigma = 1.89$.

After the general rating, the students were also asked to freely comment on problems, possible improvements, additional functionalities, and possible application scenarios (S2, S3, S4). At this point they all confirmed that they have had no problems while using the Interpretation Switcher interface. However, two students noted that the interface could have reacted faster while switching between the respective performances. One student would have appreciated to have an additional functionality for displaying the musical score during playback. Also user-defined auxiliary markers that can be freely fixed, adjusted, and removed along the various slider control bars should be introduced for additional orientation and navigation purposes. All but one of them affirmed that they could imagine to use the Interpretation Switcher within their studies or even for private use. In particular, they said that the interface may be useful in the context of special seminars, where the comparison of different performances play an important role. One student was enthusiastic about the features offered by the Interpretation Switcher. He usually records his piano lessons in order to listen to and to study his own playing afterwards. Here, he would significantly benefit from novel switching and navigation functionalities for comparing and analyzing the recorded audio material. Also, the Interpretation Switcher could be very useful for compactly documenting the learning progress over a longer period in time. For example, it could synchronously present the various performances of a specific musical section recorded in different piano lessons over the semester.

5.4 Conclusions

In this chapter, we presented a first experiment conducted at the University of Music Saarbrücken with the main objective of introducing MIR user interfaces with novel switching and navigation functionalities to music teachers and students. Even though this group tends to be skeptical about using computer-based methods in music education, most participants affirmed the usefulness of our interface for comparing and analyzing performances or simply for music listening and enjoyment. Testing and evaluating our interface within a concrete application of practical relevance, we not only made a new group of prospective users acquainted with MIR methods but also obtained valuable feedback from music experts.

The presented experiment only constitutes the beginning of a planned collaboration with music educators and students, who are usually not aware of the developments in music information retrieval. For the future, we plan to conduct similar experiments on a larger scale. One further idea is to participate regularly in the lessons of piano students to record their playing. We then plan to process (segment, classify, synchronize) the audio material automatically and to suitably integrate it in our Interpretation Switcher to document and analyze the students' learning process.

Finally, we plan to develop and combine various additional functionalities. For example, as mentioned by one of the participants, an additional sheet music interface for presenting the musical score while playing back associated audio material would be helpful. Actually, such functionalities have been presented in [9]. Furthermore, we will integrate additional functionalities for inter- and intra-document music browsing including the possibility of setting user-defined auxiliary markers as well as pre-computed markers that reflect the musical form of the piece [16, 20]. In introducing novel functionalities, one main challenge will be to keep the operability of the interface as intuitive as possible to avoid rejections from the users' side.

Chapter 6

A Multi-Perspective User Interface for Music Signal Analysis

In view of the exploding distribution of digitized audio material, computer-based methods have become indispensable for processing and analyzing the content of music signals. To evaluate analysis results obtained by automated methods, one requires manually generated high-quality labeled data and the feedback by music experts. In this chapter, we introduce various novel functionalities for a user interface that opens up new possibilities for viewing, comparing, interacting, and evaluating analysis results within a multi-perspective framework and bridges the gap between signal processing and music sciences. Here, we exploit the fact that a given piece of music may have multiple, closely-related sources of information including different audio recordings and score-like MIDI representations. Our interface then allows a user to interactively generate unifying views of the analysis results across the available music representations. Disclosing musically relevant consistencies and inconsistencies, these views not only afford new evaluation and navigation possibilities but also deepen a user's understanding of the underlying musical material. The results of this chapter have been published in [60].

This chapter is organized as follows. First, we present the underlying user interface (Section 6.1), and introduce three different modes for representing the timelines of the versions (Section 6.2). Afterwards, we illustrate the effect of the three timeline modes using Beethoven's Pathétique Sonata as an example for a case study (Section 6.3). In Section 6.4, we introduce a novel functionality of our interface, which allows for generating multi-perspective views across different version-dependent analysis results before indicating various application scenarios (Section 6.5). Finally, we demonstrate how our interface may serve as a helpful tool for ear training (Section 6.6).

6.1 Extension of the Interpretation Switcher

The technical backbone of our user interface is referred to as *Interpretation Switcher*, which has emerged from the previously developed *SyncPlayer* system [16] and is described in Section 4.2. This interface allows a user to select several recordings of the same piece of

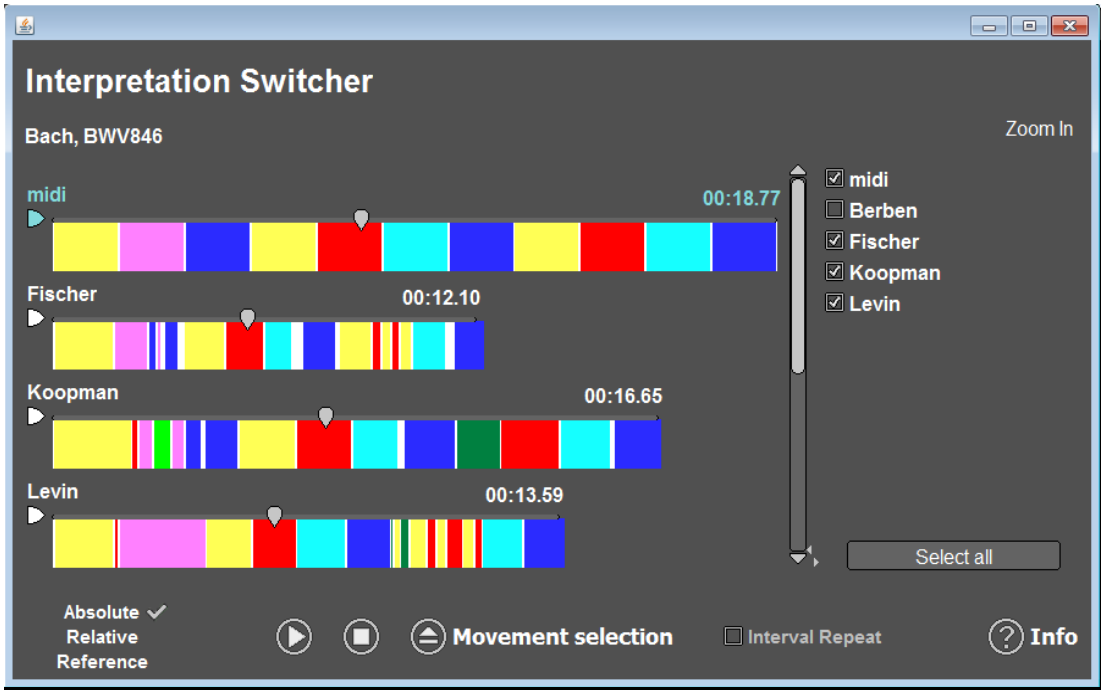

Figure 6.1. Interpretation Switcher opened with four different versions (MIDI file and three audio recordings) of the first eleven bars of Bach's Prelude in C Major (BWV 846). The annotations correspond to version-dependent chord labels (generated manually for the MIDI version and automatically for the audio versions). In the right part of the interface, the user may select any subset of the available versions (here, four out of five versions are selected).

music, which have previously been synchronized [54]. Each of the recordings is represented by a slider bar indicating the current playback position with respect to the recording's particular timeline, see Figure 6.1. The user may listen to a specific recording by activating a slider bar and then, at any time during playback, seamlessly switch to any of the other versions (inter-document navigation).

In addition to the switching functionality, we have extended the Interpretation Switcher to also indicate available version-dependent annotations below each individual slider bar, where labeled segments are represented by color-coded blocks. Such annotations may encode the chord labels generated manually or obtained by some automated chord recognition procedure [77]. Or, such annotations may correspond to the repetitive structure or the musical form, which may have been extracted from the respective recording using automated structure analysis procedures [68]. Based on these annotations, the Interpretation Switcher also facilitates intra-document navigation, where the user can directly jump to the beginning of any structural element simply by clicking on the corresponding block, see Figure 6.1.

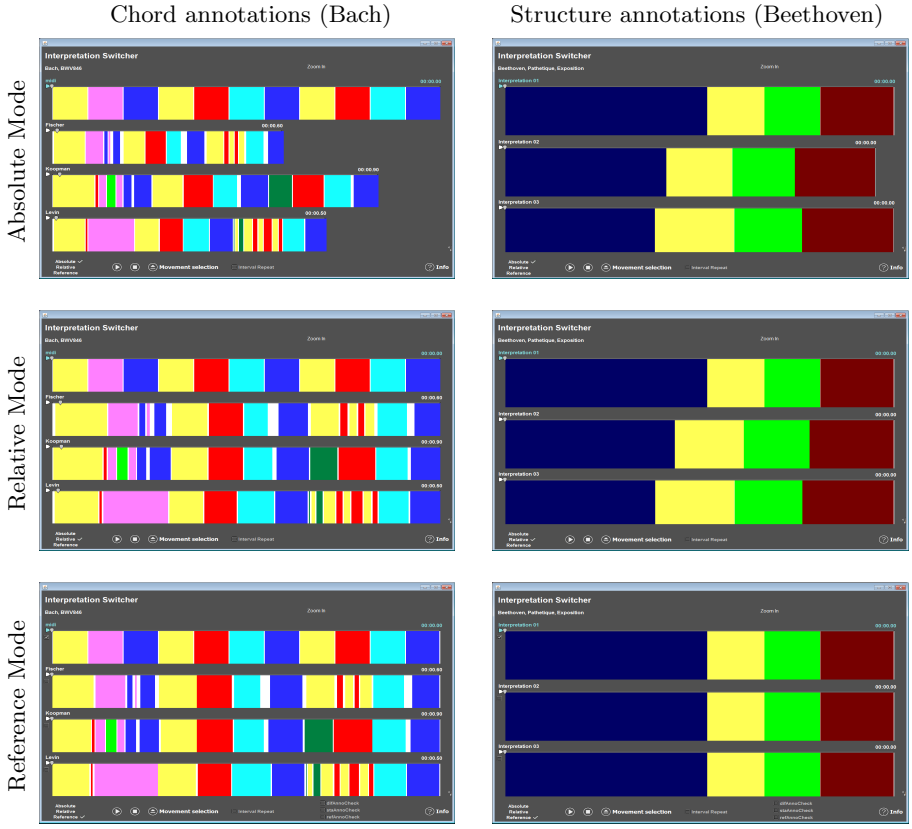

Figure 6.2. Different timeline modes showing annotations in the absolute mode (**top**), the relative mode (**middle**) and the reference mode (**bottom**) using the first versions as reference, respectively. The left column continues the Bach example from Figure 6.1. The right column shows the Interpretation Switcher opened with three different recordings of the exposition of Beethoven's Pathétique Sonata. Here, the annotations correspond to structural information indicating four musically meaningful parts of the exposition.

6.2 Timeline Modes

We have further extended the functionalities of the Interpretation Switcher by realizing three different modes for representing the timelines of the versions. In the *absolute mode*, each timeline encodes absolute timing, where the length of a particular slider bar is proportional to the duration of the respective version, see Figure 6.2 (top). In the *relative mode*, each timeline encodes relative timing, where the length of all slider bars coincide, see Figure 6.2 (middle). In other words, in the relative mode all timelines are linearly stretched to yield the same length. The third mode, which is referred to as *reference mode*, is the most interesting one. Here, an arbitrary but fixed version can be selected to act as a reference. Then, all timelines of the other versions are temporally warped to run synchronously to the reference timeline, see Figure 6.2 (bottom).

One feature of our timeline adjustment functionality is that the annotations indicated below the slider bars are also adjusted according to the respective mode. Thus, the different timeline modes allow for generating different views on these annotations. For

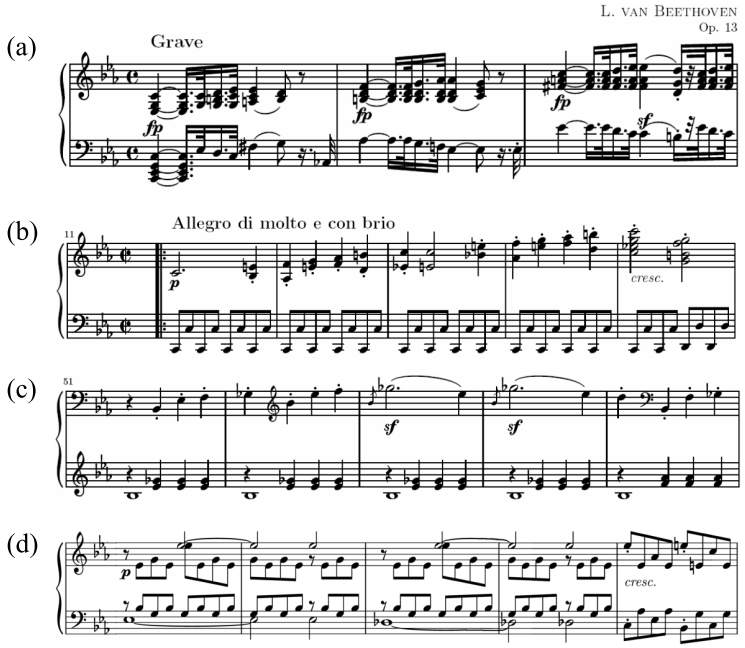

Figure 6.3. First movement of Beethoven's Pathétique Sonata Op. 13 (score obtained from [63]). **(a)** Beginning of the introduction (Part A, bb. 1 ff.) **(b)** First theme (Part B, bb. 11 ff.) **(c)** Second theme (Part C, bb. 51 ff.) **(d)** Part D (bb. 89 ff.)

example, using the reference mode, all annotations are temporally warped onto a common timeline, which then facilitates a direct comparison of the annotations across the versions. This is a very useful feature, in particular when the reference corresponds to ground-truth annotations. Furthermore, when the reference corresponds to an uninterpreted MIDI version representing a musical score, the reference mode allows for presenting all version-dependent annotations with respect to a musically meaningful timeline, where time is given in bars and bars rather than seconds.

6.3 Case Study

In the following case study, we exemplarily discuss the effect of the different timeline modes by means of the Beethoven example in Figure 6.2, right column. Here, the Interpretation Switcher is opened with three different recordings of the exposition of Beethoven's Pathétique Sonata Op. 13 for which structure annotations are indicated. For each recording the respective structure annotation consists of four blocks (A (blue), B (yellow), C (green) and D (red)), which correspond to musically meaningful parts of the exposition.

To better understand the structure annotations shown in Figure 6.2, right column, we refer to the description of the Pathétique's exposition in Section 5.2.1. For convenience, Figure 6.3 shows again the score of the considered parts of the exposition. The first block (A) of the structure annotation of each recording corresponds to the introduction of the exposition, see Figure 6.3a. The introduction leads in the first theme (bb. 11 ff.,

see Figure 6.3b) of the sonata, corresponding to the second block (B) of the structure annotation. The second theme (bb. 51 ff., see Figure 6.3c), corresponds to the third block (C) of the structure annotation. The last block (D) of the structure annotation refers to the fourth part of the exposition, introduced by a third theme in E flat major (bb. 89 ff., see Figure 6.3d).

Now, we again consider the three recordings of the Pathétique Sonata. The three different timeline modes of the Interpretation Switcher allow for generating different views on the structure annotations. Firstly, using the absolute mode (see Figure 6.2, right column, top), where each timeline encodes absolute timing, enables to visually compare the absolute durations of the three recordings in an intuitive way. For example, one directly observes that the lengths of the first and the third slider bar roughly agree with each other, whereas the second slider bar is noticeably shorter. In other words, Pianist 1 and Pianist 3 choose a slower overall tempo in their performances of the exposition (resulting in a total duration of 224 seconds), whereas Pianist 2 plays the exposition much faster (resulting in a total duration of only 213 seconds), see also Table 6.1.

Secondly, the relative mode (see Figure 6.2, right column, middle) allows for visually comparing the relative durations of the particular structure blocks with respect to the total durations of the recordings. In this way, performance characteristics concerning the tempo shaping in the four parts can be investigated easily. For example, one can notice that Pianist 1 plays the introduction of the exposition (Part A) rather slowly compared to the two other pianists (covering 52.2% of the duration of his/her whole performance). On the contrary, Pianist 3 plays the introduction much faster so that its duration amounts to only 38.4% of the total duration. However, one observes that Pianist 1 plays all the three subsequent parts (B, C, D) faster than the two other pianists, see Table 6.1. Indeed, Pianist 1 plays the introduction in a slow and expressive way but changes to a faster tempo level at the actual beginning of the exposition (Part B).

Thirdly, in the reference mode (see Figure 6.2, right column, bottom) all timelines are temporally warped to run synchronously to the reference timeline, where every recording can be selected to act as a reference. In this example, the first recording serves as the reference. The reference mode allows now for a direct comparison of the annotations across the recordings. One directly notices that the annotations of the three recordings agree with each other. Here, the underlying reason is that the structure annotations are consistent across the recordings and perfectly reflect the musical structure of the exposition. Actually, in this example, the annotations were generated manually. However, the situation changes when annotations are computed by automated procedures for each of the versions independently. Then, one typically encounters analysis errors and inconsistencies, which become apparent in the Bach example (Figure 6.2, left column). This example will be described in more detail in the subsequent section.

6.4 Multi-Perspective Views

As a further contribution, we have realized a functionality that facilitates the generation of multi-perspective views across different version-dependent analysis results. We discuss this functionality by means of our Bach example, where we consider a score-like uninter-

		Exp.	A	B	C	D
Pianist 1	Time[sec]	224	117	33	32	42
	Rel. time[%]	100	52.2	14.7	14.3	18.8
Pianist 2	Time[sec]	213	93	38	36	46
	Rel. time[%]	100	43.7	17.8	16.9	21.6
Pianist 3	Time[sec]	224	86	46	39	53
	Rel. time[%]	100	38.4	20.5	17.4	23.7

Table 6.1. Absolute and relative time durations for the structural parts of the Pathétique's exposition. The table shows for each performance the absolute durations (in seconds) and the relative durations (in %) of the considered structural parts (A, B, C, D) with respect to the total duration of the respective performance.

preted MIDI file (with manually generated ground-truth chord labels) and three audio recordings (with automatically extracted chord labels). Applying the interface's zooming functionality, Figure 6.4b shows the version-dependent chord labels of the Bach example (bb. 7-9) in the reference mode, which enables for a simultaneous comparison of the various chord labels over multiple versions of the same piece of music. The various colors correspond to the different labels. In our example, the first slider bar corresponds to the MIDI version which, in this example, is used as the reference. (However, note that *any* version may be selected to serve as the reference.) Here, bar 7 is labeled as G major (blue), bar 8 as C major (yellow), and bar 9 as A minor (red). The white color indicates unannotated passages. In the reference mode, the interface allows for placing a copy of the reference annotations below each of the version-dependent annotations. Furthermore, the pairwise consistencies (indicated by white) and inconsistencies (indicated by black) across annotations can be visualized.

Such a multi-perspective view is shown in Figure 6.4c. Here, a tripartite panel is associated to each version showing the original version-dependent annotations (top), the pairwise consistency information (middle), and the reference annotations (bottom). For example, this view immediately reveals that there are inconsistent annotations in bar 8, which is labeled as C major (yellow) in the first version (reference) and labeled as E minor (green) in the third version. Actually, this misclassification has musical reasons: in bar 8 a C major seventh chord is played, which is simplified to C major in the manual annotation. However, due to the added seventh (B) all the tones for E minor (E,G,B) are also present leading to the misclassification E minor.

Additionally, the interface can also provide statistics that indicate the degree of consistency with respect to the reference across all available versions. These statistics are visualized as an additional gray-scaled panel as shown at the bottom of Figure 6.4c. Here, the degree of consistency is reflected by the luminance of the grayscale. In particular, a white entry at a given reference time position indicates that all chord labels agree with the reference label across all versions, whereas a black entry indicates that all non-reference chord labels differ from the reference label. This visualization points the user to problematic passages, which were labeled inconsistently. These inconsistencies may be due to weaknesses of the used labeling procedure (analysis errors), to synchronization inaccuracies, or to musical ambiguities in the piece of music (ill-posed problem, inadequate model assumptions). In our Bach example, the multi-perspective view reveals that for bar 7 the chord labels (blue) agree across all versions (except for some smaller inconsistencies at the left bound-

Figure 6.4. Various multi-perspective views for the Bach example zooming into bars 7 to 9 of the first eleven bars as shown in Figure 6.2. **(a)**: Score of bars 7 to 9. **(b)**: Interpretation Switcher in the reference mode (using the first version as reference). **(c)**: Multi-perspective view showing copies of the reference annotations below each of the version-dependent annotations and the pairwise consistencies (white) and inconsistencies (black). The bottom visualizes the degree of consistency (gray values) across all versions.

ary that may stem from synchronization inaccuracies). On the contrary, for bb. 8-9, the multi-perspective view indicates several inconsistencies. Hence, these two bars seem to be problematic passages in the piece of music. Actually, looking at the score one finds out that seventh chords are present in both bars (C major seventh in bar 8, A minor seventh in bar 9), which produces a certain chord ambiguity resulting in misclassifications.

Our interface offers various ways, a user can interactively modify the views including zooming and selection options. In particular, *every* version can be selected to serve as a reference, where the view immediately adjusts upon selection. Note that in this case not only the timeline is changed, but also the copied reference annotations are replaced and the statistics are recomputed. As another feature, for a given set of versions, one can select an arbitrary subset to be considered in the multi-perspective view. For example, in Figure 6.1, four of the five available versions are selected.

6.5 Applications

In this section, we indicate various application scenarios for our advanced Interpretation Switcher Interface. First of all, as indicated in the previous section, our user interface may serve as a valuable tool for the evaluation of automated music analysis and labeling procedures. Using the reference mode, a multi-perspective view can be generated that yields a synchronized and compact overview of version-dependent analysis results across multiple music representations of a given piece of music. Here, annotation consistencies and inconsistencies can be visualized in a pairwise mode, where each version is compared with the reference separately, as well as in a comprehensive mode comprising all versions. Here, inconsistencies typically point to misclassifications that may be due to analysis errors of automated methods or to intrinsic musical ambiguities. On top of the visual feedback, our interface allows for immediate playback of any position within any version simply by clicking on a color-coded block. Such a block visually represents either a labeled segment or a derived segment that indicates consistency information. This allows a user to easily identify interesting musical passages by means of the visual cues and then to playback the corresponding underlying acoustic material. Having such audio-visual navigation and feedback functionalities, a researcher is greatly supported in performing an in-depth error analysis while deepening his or her understanding of the underlying musical material.

At this point, we want to emphasize that our multi-view evaluation interface may yield interesting information even in the case that no ground-truth annotations are available. For example, in chord recognition, most research is evaluated on the basis of a corpus of Beatles songs, for which high-quality manual chord transcripts have been prepared [27]. However, such special-purpose manual annotations are rarely available. Therefore, one may exploit the fact that one often has large quantities of different versions (e. g., various performances) of a given piece of music, which present opportunities for generating substitutes for manual ground-truth using music synchronization techniques [29, 54, 85]. First multi-perspective approaches to automatically evaluate algorithms have been applied to chord recognition [36] and to beat tracking [24]. In this context, our user interface supports such approaches by supplying immediate visual and acoustic feedback.

As another major benefit, our Interpretation Switcher alleviates interdisciplinary research by bridging the gap between music information retrieval and music sciences. Usually, MIR methods are evaluated by MIR researchers in their own lab environment, and music experts are rarely incorporated in the evaluation process. Here, one reason is the lack of communication between MIR researchers, who often do not have an adequate musical background, and music experts, who are often reluctant in using novel computer-assisted

methods. Our interface allows even a technically unexperienced user to perform an error analysis of automatically generated annotations. Being pointed to problematic passages by the interface, a music expert can employ his or her musical knowledge and trained ear for an in-depth audio-visual analysis of specific passages. This process can be supported by an MIR researcher who provides the knowledge about the details of the employed annotation methods. In this way, our interface opens the way for an interdisciplinary collaboration, which, on the one hand, supports the MIR researcher in improving the employed methods using the valuable feedback from the music expert, and, on the other hand, familiarizes the music expert with novel computer-assisted methods and interfaces.

For the future, we plan to apply our advanced Interpretation Switcher to support interdisciplinary research going far beyond evaluation. In the context of musicology, one project consists in determining tonal centers (i. e., passages dominated by a certain key) within a large musical work or even entire music corpora, see Chapter 12. Here, first experiments show that our multi-perspective audio-visual navigation functionalities considerably alleviates the work of musicologists. As a second interdisciplinary project, we have started to introduce computer-based methods into the context of music education, see Chapter 5. Here, our user interface may help to conduct more user-centered analyses of MIR methods within natural, music-oriented settings [42].

6.6 The Ear Training Plugin

In this section, we illustrate how our multi-view interface may serve as an interactive tool for ear training. First, in Section 6.6.1 we give an introduction to ear training. Afterwards, in Section 6.6.2 we describe the desirable functionalities of an ear training plugin based on the multi-view interface described in Section 6.4.

6.6.1 Ear Training

Ear Training is an important subject at academies of music. First of all, every candidate has to pass an ear training test in the entrance examination. Furthermore, the students of all study paths have to attend an ear training course for several years and to pass a final exam. Ear training denotes the procedure of instructing the ear in order to determine intervals, pitches and rhythms. The usual method in ear training is the dictation, where the student has to transcribe the music by listening to it. There exist several levels of dictations, e. g. dictations for one or for multiple voices, tonal or atonal dictations. By training the ear, the perception of music is sharpened resulting in a deeper understanding of musical structures, which are valuable skills in the context of improvisation, composing, playing or analyzing music. Only a few musicians possess absolute pitch ability, whereas most of them use relative pitch ability. Therefore, the musical context in which a musical event sounds is indispensable for identifying it. Furthermore, the musician has to be informed about the initial musical event. A usual ear training lesson proceeds as follows: The teacher plays a certain dictate on the piano and the students try to transcribe it. Afterwards, the transcription is compared to the original. Passages, where the student's solution deviates from the original are played again by the teacher. Here, one didactic

method is to play the student's wrong transcription in comparison to the original. In this way, the student can directly learn from his/her error. Usually, ear training requires plenty of practice and discipline. However, for ear training exercises one generally needs a second person who fills the role of the teacher playing the dictate. It is hardly possible to train the ear by oneself. In the following, we illustrate how an ear training plugin for the Interpretation Switcher may open up new possibilities for interactively training the ear.

6.6.2 Functionalities of the Ear Training Plugin

We assume that a certain piece of music is opened in the Interpretation Switcher using the reference mode. Here, the first slider bar represents a MIDI version and the second slider bar an audio version of the considered piece of music. Additionally, the annotation panel below the MIDI version allows for indicating the ground truth annotation, whereas the panel below the audio version serves for visualizing the manually annotated chord labels by the student. In the following, we consider only the 24 major and minor chords, where every chord corresponds to a fixed color in the visualization.

In the beginning, the ground truth annotation is hidden so that the two annotation panels are white colored. However, the borders of the chord segments are indicated by vertical black lines (see Figure 6.5a). Now, the task for the student is to manually annotate the chords by listening to the piece of music. By right-clicking within a certain segment of the audio version's annotation panel, a menu bar opens, showing the 24 major and minor chords. By clicking on one of these 24 chords, the student can select the chord for the corresponding segment. After the selection, the respective segment colors corresponding to the selected chord and the shorthand of the chord is written as a string within the segment (see Figure 6.5b).

Having manually annotated all the chords, the student can now compare his/her annotation to the ground truth annotation. By activating the mode *Solve* the ground truth annotation is blend in, visualized by the respective colors and shorthands (see Figure 6.5c). Using the reference mode, the student can now directly compare his/her annotation with the ground truth annotation. In the mode *refAnnoCheck* this comparison is facilitated since copies of the ground truth annotation are placed above each of the two anntotations (see Figure 6.5d). Furthermore, by activating the mode *diffAnnoCheck* the differences between the two annotations are visualized in black (see Figure 6.5e). In this way, the student can easily locate wrongly annotated passages and by clicking on the respective segment listen to the previously identified problematic passage.

As the process of listening plays a crucial role in ear training, we now demonstrate how sonifications of the chord annotations can support the student's learning process. We therefore assume that for each of the 24 considered chords a prestored sonification exists. Activating the *errorListenCheck* mode, the acoustic playback of the two slider bars changes. While the first slider bar plays back by default the MIDI version of the piece of music, it plays back in the *errorListenCheck* mode the chord progression corresponding to the ground truth annotation using the respective sonified chords. The second slider bar plays back by default the audio version of the piece of music. However, in the *errorListenCheck* mode it plays back the chord progression corresponding to the manual annotation

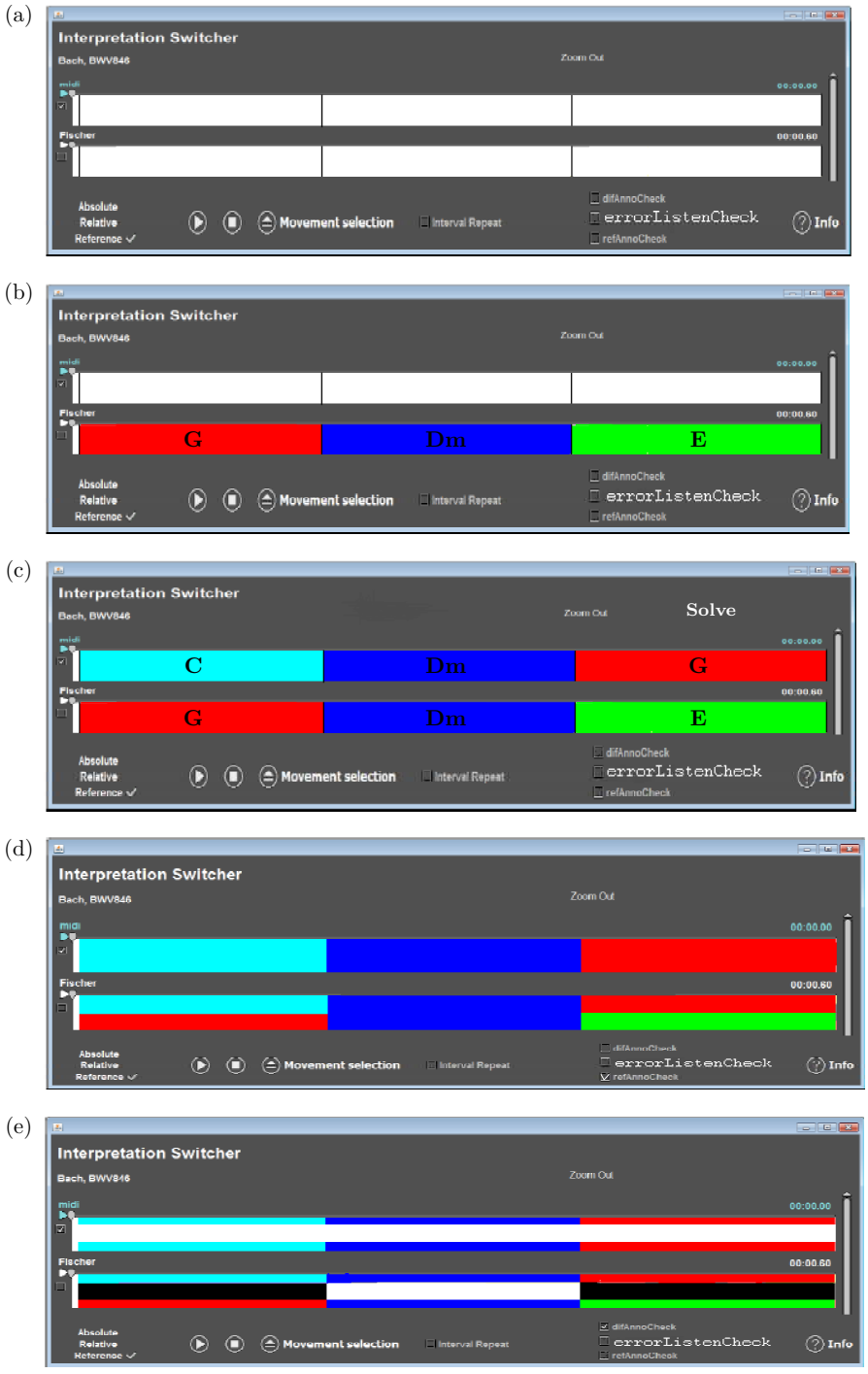

Figure 6.5. Different instances of the Ear Training Plugin. (a) Start setting. The borders of the chord segments are indicated by vertical black lines. (b) The annotation panel below the audio version's slider bar enables the student for manually annotating the chords by listening to the piece of music. (c) The ground truth annotation is blend in below the MIDI version. (d) Copies of the ground truth annotation are placed above each of the two annotations. (e) Deviations from the ground truth annotation are visualized in black between the two annotations.

of the student using the prestored sonified chords. The *errorListenCheck* mode supports the student's learning process in two ways: Firstly, listening to chord progressions instead of the original composition simplifies the identification of the underlying chords. Secondly, the common didactic method, which allows the student to compare by listening his/her own possibly wrong annotation with the ground truth annotation, is realized. Using the switching functionality of the interface, the student can listen in the musical context the wrong manual annotation in comparison to the actual chord progression. This possibility may realize a real learning progress in ear training.

Part III

Harmonic Analysis

Chapter 7

Chord Labeling

Automated chord labeling, which deals with the computer-based harmonic analysis of audio recordings, is one of the central tasks in the field of music information retrieval. In this chapter, we first introduce the topic of chord labeling by formulating the task and giving an overview of related work (Section 7.1). Afterwards, we describe different approaches for chord labeling: a template-based (Section 7.2), a Gaussian-based (Section 7.3) and an HMM-based approach (Section 7.4). Then, we refer to the feature extraction step and describe different types of chroma features which are of interest in the context of chord labeling (Section 7.5). In a subsequent experiment we demonstrate the importance of the features by analyzing the behavior of several chord labeling procedures in dependency of various feature types (Section 7.6). Finally, we investigate the role of tuning in the context of chord labeling by conducting a baseline experiment which shows that balancing out tuning deviations has a great impact on the chord labeling results (Section 7.7). This chapter is mainly based on [30], Section 7.1 to Section 7.6 basically follow [30]. The results presented in Section 7.7 have been published in [35].

7.1 Related Work

In recent years automated chord labeling has been of increasing interest in the field of MIR see, e. g., [4, 6, 8, 17, 26, 27, 40, 47, 48, 64, 65, 66, 70, 77, 86]. Harmony is a fundamental attribute of Western tonal music and the succession of chords over time often forms the basis of a piece of music, where a chord denotes the simultaneous sound of three or more pitches. Harmonic progressions are not only of musical importance, but also constitute a powerful mid-level representation for the underlying musical signal and can be applied for various tasks such as music segmentation, cover song identification, or audio matching [76, 62].

In the following, we address to the problem of audio-based chord labeling, where the chord labeling task consists in first splitting up the recording into segments and then assigning a chord label to each segment. The segmentation specifies the start time and end time of a chord, and the chord label specifies which chord is played during this time period. Most chord labeling procedures proceed in a similar fashion. In the first step, the given music

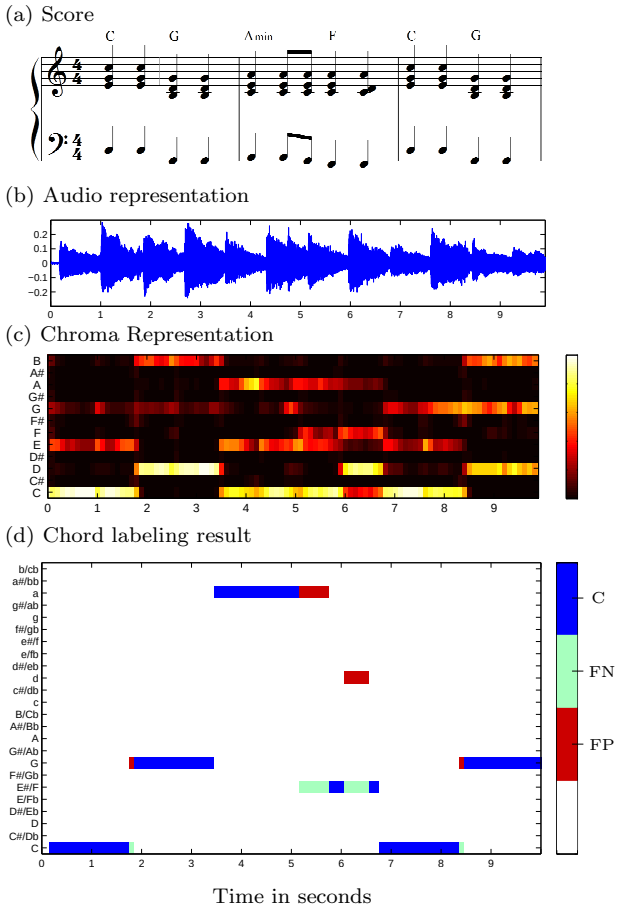

Figure 7.1. Chord labeling task illustrated by the first measures of the Beatles song "Let It Be". **(a):** Score of the first three measures. **(b):** Audio representation of these measures. **(c):** Chroma representation. **(d):** Chord labeling result indicating correct (C), false negative (FN), and false positive (FP) labels.

recording is transformed into a sequence $X = (x_1, x_2, \ldots, x_N)$ of feature vectors $x_n \in \mathcal{F}$, $n \in [1 : N] := \{1, \ldots, N\}$. Here, $\mathcal{F}$ denotes a suitable feature space. Most recognition systems are based on so-called chroma features or pitch class profiles, which we discuss in detail in Section 7.5. In the second step, using suitable pattern matching techniques, each feature vector x_n is mapped to a chord label $\lambda_{x_n} \in \Lambda$, see Figure 7.1(d). Here, Λ denotes a suitably defined set of all possible chord labels. In the following, we consider the case that Λ consists of the twelve major and minor triads, i. e.,

$$\Lambda = \{\mathtt{C}, \mathtt{C}^\sharp, \ldots, \mathtt{B}, \mathtt{Cm}, \mathtt{C}^\sharp\mathtt{m}, \ldots, \mathtt{Bm}\}. \tag{7.1}$$

The restriction to these 24 chord classes, even though problematic from a musical point of view, is often made in the chord labeling literature.

There are many ways of performing the pattern matching step. The first possibility is to use template-based matching strategies [17, 64], a simple template-based labeling strategy is described in Section 7.2. Currently, most chord labeling approaches employ hidden Markov models (HMMs) which also account for the temporal context in the classification stage [77, 86, 6, 40, 70]. In Section 7.4, we will summarize an HMM-based chord labeling

procedure. Furthermore, more complex Bayesian networks have been suggested for chord labeling [47].

After the pattern matching step, further post-filtering techniques are applied to smooth out local misclassifications. In the case that HMMs are used, the pattern matching and temporal filtering steps are jointly performed within one optimization procedure.

Even though numerous procedures for automated chord labeling have been described in the literature, the delicate interplay of the various feature extraction, filtering, and pattern matching components is still not sufficiently investigated and understood. The situation is complicated by the fact that the components' behavior may crucially depend on a variety of parameters that allow for adjusting temporal, spectral, or dynamical aspects. In [65], the influence of various aspects and parameters of a typical HMM-based chord recognizer is investigated. In [6], a detailed investigation is described to better understand the interrelation of different chord labeling components with a focus on the impact of filtering and pattern matching strategies. However, the impact of different feature extraction strategies was not investigated being left for future work. In Section 7.6, we continue this strand of research by analyzing the impact of various types of chroma features in the context of the chord labeling task.

7.2 Template-Based Chord Labeling

One way to perform the pattern matching step, is to use a template-based labeling strategy. Here, the idea is to pre-compute a set $\mathcal{T} \subset \mathcal{F}$ of templates that correspond to the set of chord labels. The elements of $\mathcal{T}$ are denoted by $\mathbf{t}_\lambda \in \mathcal{T}$, $\lambda \in \Lambda$. Intuitively, each template is given in the form of a kind of prototype chroma vector that corresponds to a specific musical chord. Furthermore, we fix a distance measure $d : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ that allows for comparing different chroma features. In the following, we use the cosine measure defined by

$$d(x, y) = 1 - \frac{\langle x|y\rangle}{\|x\| \cdot \|y\|}, \tag{7.2}$$

for $x, y \in \mathcal{F} \setminus \{0\}$. In the case $x = 0$ or $y = 0$, we set $d(x, y) = 1$. Here, $\|\cdot\|$ denotes the Euclidean norm (also referred to as ℓ^2-norm).

Then, the template-based chord recognition procedure consists in assigning the chord label that minimizes the distance between the corresponding template and the given feature vector $x = x_n$:

$$\lambda_x := \operatorname*{argmin}_{\lambda \in \Lambda} d(\mathbf{t}_\lambda, x). \tag{7.3}$$

Note that this procedure works in a purely framewise fashion without considering any temporal context.

There are several strategies for determining suitable chord templates based on musical knowledge or learning procedures using labeled training data. In the following, we consider binary templates and averaged templates. The set $\mathcal{T}^b$ consists of 24 binary templates, each of which being a 12-dimensional binary vector with three non-zero entries equal to one. These non-zero entries correspond to the three chromas the corresponding chord

is composed of. For example, the binary template corresponding to the major chord $\mathtt{C} = \{\mathrm{C}, \mathrm{E}, \mathrm{G}\}$ is given by

$$\mathbf{t}^b_{\mathtt{C}} = (1,0,0,0,1,0,0,1,0,0,0,0)^T. \tag{7.4}$$

Furthermore, the set $\mathcal{T}^a$ consists of averaged templates, which are learned from training material by averaging suitable chroma vectors obtained from labeled audio data. For example, the averaged template $\mathbf{t}^a_{\mathtt{C}}$ is obtained by averaging all chroma vectors from the training set labeled as C.

The two template-based chord recognition approaches are denoted by $\mathtt{T}^{\mathtt{b}}$ and $\mathtt{T}^{\mathtt{a}}$, respectively.

7.3 Gaussian-Based Approach

Next, we introduce a chord recognition procedure based on Gaussian distributions. Here, the chord templates are replaced by chord models each specified by a multivariate Gaussian distribution given in terms of a mean vector μ and a covariance matrix Σ. As for the averaged templates, μ and Σ are learned from labeled audio data. Then, the distance of a given chroma vector to a chord model is expressed by a Gaussian probability value and the assigned label is determined by the probability-maximizing chord model (instead of the cost-minimizing chord template), see [6]. The Gaussian-based chord recognition approach is denoted by `GP`.

7.4 HMM-Based Approach

Finally, we summarize an HMM-based chord recognition procedure, which was originally suggested by Sheh and Ellis [77] and is now the most widely used chord labeling approach. The strength of this approach is that HMMs also account for the temporal context in the classification stage, which can be considered as a kind of context-aware filtering of the matching results. To this end, in addition to the Gaussian models, one needs transition probabilities that express the likelihood of passing over from one chord label to any of the other chord labels. These probabilities are given by a transition matrix $\Omega \in [0,1]^{24\times 24}$, which can be specified manually based on musical knowledge or automatically by using a training procedure reverting to suitable training material. For the labeling procedure, one then needs a Viterbi decoding algorithm to determine a chord label sequence that jointly maximizes the output probabilities defined by the Gaussian distributions and the transition probabilities, see [77]. The determination of the transition matrix also plays a crucial role in the chord recognition context and has been studied in various contributions [4, 6, 65]. In our experiments, we determine Ω using training data with annotated chord labels, see Section 7.6.1. The HMM-based chord recognition approach is denoted by `HMM`.

The HMM-based approach employed in our experiments in Section 7.6 uses more general graphical models. Since the focus of our evaluation lies on the feature side, we revert to a basic variant. More advanced recognizers are introduced in [6, 47, 86].

7.5 Feature Extraction

Chroma-based audio features, sometimes also referred to as pitch class profiles, are a well-established tool in processing and analyzing music data [2, 18, 54] and were introduced to the chord recognition task by Fujishima [17]. As already mentioned in Chapter 2, the chroma correspond to the set $\{\mathrm{C}, \mathrm{C}^\sharp, \mathrm{D}, \ldots, \mathrm{B}\}$ that consists of the twelve pitch spelling attributes as used in Western music notation, assuming the equal-tempered scale. A chroma vector can be represented as a 12-dimensional vector $x = (x(1), x(2), \ldots, x(12))^T$, where $x(1)$ corresponds to chroma C, $x(2)$ to chroma $\mathrm{C}^\sharp$, and so on. Normalized chroma-based features indicate the short-time energy distribution among the twelve chroma and closely correlate to the harmonic progression of the underlying piece. This is the reason why basically every chord recognition procedure relies on some type of chroma feature.

There are many ways for computing chroma features. For example, the transformation of an audio recording into a chroma representation (or chromagram) may be performed either by using short-time Fourier transforms in combination with binning strategies [2] or by employing suitable multirate filter banks [54]. Furthermore, the properties of chroma features can be changed by introducing suitable post- and pre-processing steps modifying spectral, temporal, and dynamical aspects. This leads to a large number of feature types which can behave quite differently depending on the subsequent analysis task. In this section, we summarize the types of chroma features that will be used in the subsequent experiments in Section 7.6. Note that there are many more chroma variants. However, our selection covers interesting variants that demonstrate the importance of the feature extraction step.

7.5.1 Pitch Features

As basis for the chroma feature extraction, we first decompose a given audio signal into 88 frequency bands with center frequencies corresponding to the pitches $A0$ to $C8$ (MIDI pitches $p = 21$ to $p = 108$). For deriving this decomposition, we use a multirate filter bank consisting of elliptic filters as described in [54]. Then, for each subband, we compute the short-time mean-square power (i. e., the samples of each subband output are squared) using a rectangular window of a fixed length and an overlap of 50 %. In the following, we use a window length of 200 milliseconds leading to a feature rate of 10 Hz (10 features per second). The resulting features, which we denote as `Pitch`, measure the local energy content of each pitch subband and indicate the presence of certain musical notes within the audio signal, see [54] for further details. To account for tuning problems, we employ a tuning strategy similar to [18]. To this end, one computes an average spectral vector and estimates the tuning deviation parameter from the maximum spectral coefficient. This tuning deviation parameter is then used to suitably shift the center frequencies of the subband-filters of the above multirate filter bank. A similar approach is described in [58].

7.5.2 CP Feature

From the `Pitch` representation, one can obtain a chroma representation by simply adding up the corresponding values that belong to the same chroma. To archive invariance in dynamics, we normalize each chroma vector with respect to the Euclidean norm. The resulting features are referred to as *Chroma-Pitch* denoted by `CP`.

7.5.3 CLP Features

To account for the logarithmic sensation of sound intensity [43, 90], one often applies a logarithmic compression when computing audio features [32]. To this end, the local energy values e of the pitch representation are logarithmized before deriving the chroma representation. Here, each entry e is replaced by the value $\log(\eta \cdot e+1)$, where η is a suitable positive constant. Then, the chroma values are computed as explained in Section 7.5.2. The resulting features, which depend on the compression parameter η, are referred to as *Chroma-Log-Pitch* denoted by $\texttt{CLP}[\eta]$.

7.5.4 CENS Features

Adding a further degree of abstraction by considering short-time statistics over energy distributions within the chroma bands, one obtains CENS (Chroma Energy Normalized Statistics) features, which constitute a family of scalable and robust audio features. These features have turned out to be very useful in audio matching and retrieval applications [62, 38]. In computing CENS features, a quantization is applied based on logarithmically chosen thresholds. This introduces some kind of logarithmic compression similar to the $\texttt{CLP}[\eta]$ features. Furthermore, these features allow for introducing a temporal smoothing. Here, feature vectors are averaged using a sliding window technique depending on a window size denoted by w (given in frames) and a downsampling factor denoted by d, see [54] for details. In the following, we do not change the feature rate and consider only the case $d = 1$ (no downsampling). Therefore, the resulting feature only depends on the parameter w and is denoted by $\texttt{CENS}[w]$.

7.5.5 CRP Features

To boost the degree of timbre invariance, a novel family of chroma-based audio features has been introduced in [56]. The general idea is to discard timbre-related information in a similar fashion as pitch-related information is discarded in the computation of mel-frequency cepstral coefficients (MFCCs). Starting with the `Pitch` features, one first applies a logarithmic compression and transforms the logarithmized pitch representation using a DCT. Then, one only keeps the upper coefficients of the resulting pitch-frequency cepstral coefficients (PFCCs), applies an inverse DCT, and finally projects the resulting pitch vectors onto 12-dimensional chroma vectors. These vectors are referred to as CRP (Chroma DCT-Reduced log Pitch) features. The upper coefficients to be kept are specified by a parameter $p \in [1 : 120]$. In our experiments, we use $p = 55$. Furthermore, similar to

the CENS[w] features, we apply temporal smoothing by introducing a window parameter w that is used to average the CRP features in a band-wise fashion. The resulting features are denoted by CRP[w].

7.5.6 CISP Features

Finally, we use a chroma type, where the tonal components are enhanced and the spectral resolution is increased by considering instantaneous frequencies. These features were originally introduced by Ellis and have been used in the chord recognition context as well as for cover song identification [12]. The basis for these features is a spectrogram.[1] To enhance the spectral resolution, the instantaneous frequency for each coefficient is estimated exploiting the phase information. Furthermore, based on the instantaneous frequencies, a separation of noise and harmonic components is performed and only harmonic components are preserved. Finally, to account for tuning deviations, the mapping of spectral coefficients to chroma bins is globally adjusted by up to ± 0.5 semitones to minimize the deviations of the instantaneous frequency values from the chroma bin centers using a histogram-based technique. To obtain the final features, denoted by CISP, adjacent frames are averaged in 100 ms windows to yield a feature rate of 10 Hz.

7.6 Importance of Features

In this section, we examine the behavior of the four chord labeling procedures described in Section 7.2, 7.3, and 7.4 in dependence on the underlying feature types. We start by describing the experimental setup (including the data collection and evaluation measure) in Section 7.6.1 and then report on the experiments in Section 7.6.2.

7.6.1 Experimental Setup

In our experiments, we use a collection of Beatles songs, which is a widely used benchmark dataset with publicly available ground-truth chord annotations [46]. Although this dataset is limited to only one artist, the results still show certain tendencies of the chord recognition accuracies. The collection, which we denote as $\mathcal{D}$, consists of 180 songs. We further partition $\mathcal{D}$ into three sub-collections $\mathcal{D}_k$, $k \in \{1, 2, 3\}$, by first ordering the recordings alphabetically according to the songs' titles, and then by putting the first 60 recordings into $\mathcal{D}_1$, the second 60 recordings into $\mathcal{D}_2$, and the last 60 recordings into $\mathcal{D}_3$.

The original annotations supplied by Harte [27] were reduced to the 24 chord labels following the widely spread convention of the MIREX Audio Chord Estimation task [52]. Here, only the first two intervals of each chord are considered, where augmented chords are mapped to major chords and diminished chords to minor chords. In some cases, there are

[1] In our experiments, we use an implementation available in the ISP toolbox http://kom.aau.dk/project/isound/. Here, discrete Fourier transforms are calculated over windowed frames of length 93 ms with 75% overlap. Consequently, each frame corresponds to 23 ms of the audio and each coefficient covers a frequency range of 10.8 Hz.

passages where no meaningful chord information exists. Such regions are annotated as "N" and are left unconsidered in our evaluation (i. e., having no influence on the recognition accuracy).

In our evaluation, we first quantize and segment the chord annotations to match the frames being specified by the feature extraction step. The evaluation is then performed framewise using standard precision and recall measures by comparing the automatically generated labels with the reference labels. More precisely, a reference label is considered *correct* (C) if it agrees with the computed label, otherwise it is called a *false negative* (FN). Each incorrectly computed label is called a *false positive* (FP), see also Figure 7.1(d). From this one obtains precision, recall, and F-measure defined by

$$\mathrm{P} = \frac{\mathrm{C}}{\mathrm{C}+\mathrm{FP}}, \quad \mathrm{R} = \frac{\mathrm{C}}{\mathrm{C}+\mathrm{FN}}, \quad \mathrm{F} = \frac{2\cdot\mathrm{P}\cdot\mathrm{R}}{\mathrm{P}+\mathrm{R}} \tag{7.5}$$

for each song.

In our evaluation, we employ a 3-fold cross validation. Here, two of the three sub-collections are used to train the recognizer that is then tested on the remaining one. F-measure values are averaged over all songs of the respective sub-collection $\mathcal{D}_k$. The final F-measure for the overall dataset $\mathcal{D}$ is the mean of the values obtained for the three sub-collections.

For determining the averaged templates to be used in $\mathtt{T}^{\mathtt{a}}$ as well as μ and Σ to be used in `GP` and `HMM`, we revert to the observation by Goto [19] that the twelve cyclic shifts of a 12-dimensional chroma vector correspond to the twelve possible transpositions. Therefore, exploiting the reference chord labels, we first transpose all chroma features to `C` or `Cm`, then determine the models for these two chords, and finally obtain models for all 24 chords by suitably transposing the `C` and `Cm` models. This procedure guarantees the same amount of training data for all major and minor chords, respectively. To generate the transition matrix Ω, we first determine for each frame the corresponding reference label. Then, for all $\lambda_i, \lambda_j \in \Lambda$ we define the transition probabilities $\Omega(\lambda_i, \lambda_j) = \frac{C(\lambda_i,\lambda_j)}{\sum_{\lambda_k\in\Lambda} C(\lambda_i,\lambda_k)}$. Here, $C(\lambda_i, \lambda_j)$ specifies the number of chord transitions from label λ_i to the label λ_j, and $\sum_{\lambda_k\in\Lambda} C(\lambda_i, \lambda_k)$ serves as a normalization counting the transition from λ_i to all labels $\lambda_k \in \Lambda$ including itself.

7.6.2 Dependency on Feature Type

In the following experiment, the dependency of the chord recognition results on the underlying feature type is investigated. Figure 7.2 summarizes the results of the evaluation for the five different feature types in combination with the four recognizers. In this experiment, we use the compression parameter $\eta = 100$ for `CLP`$[\eta]$ and the window parameters $w = 1$ and $w = 11$ for `CENS`$[w]$ and `CRP`$[w]$.

To better understand the influence of the 3-fold cross validation used in our experiments, Figure 7.2(a)-(c) shows the recognition accuracies for the three folds independently. Averaging over the results of the three folds, one obtains the final results of the cross validation shown in Figure 7.2(d). The F-measure values for the different parts of $\mathcal{D}$ are very consistent, e. g., using `CP` together with `HMM` leads to F = 0.531 for $\mathcal{D}_3$ (Figure 7.2(a)), F = 0.528

(a)

	T^b	T^a	GP	HMM
CP	0.449	0.427	0.434	0.531
CLP[100]	0.540	0.607	0.616	0.716
CENS[1]	0.458	0.435	0.439	0.581
CENS[11]	0.534	0.552	0.562	0.633
CRP[1]	0.523	0.573	0.573	0.708
CRP[11]	0.607	0.659	0.661	0.712
CISP	0.426	0.482	0.505	0.707

(b)

	T^b	T^a	GP	HMM
CP	0.476	0.416	0.423	0.528
CLP[100]	0.566	0.610	0.612	0.720
CENS[1]	0.474	0.425	0.420	0.588
CENS[11]	0.565	0.557	0.567	0.650
CRP[1]	0.532	0.592	0.591	0.729
CRP[11]	0.615	0.676	0.678	0.733
CISP	0.436	0.484	0.501	0.721

(c)

	T^b	T^a	GP	HMM
CP	0.456	0.412	0.408	0.522
CLP[100]	0.552	0.613	0.618	0.738
CENS[1]	0.441	0.430	0.413	0.583
CENS[11]	0.540	0.552	0.543	0.631
CRP[1]	0.529	0.583	0.577	0.711
CRP[11]	0.615	0.665	0.657	0.714
CISP	0.425	0.480	0.490	0.713

(d)

	T^b	T^a	GP	HMM
CP	0.460	0.418	0.421	0.527
CLP[100]	0.553	0.610	0.615	**0.725**
CENS[1]	0.458	0.430	0.424	0.584
CENS[11]	0.546	0.554	0.557	0.638
CRP[1]	0.528	0.583	0.581	0.716
CRP[11]	**0.612**	**0.667**	**0.665**	0.720
CISP	0.429	0.482	0.499	0.714

(e)

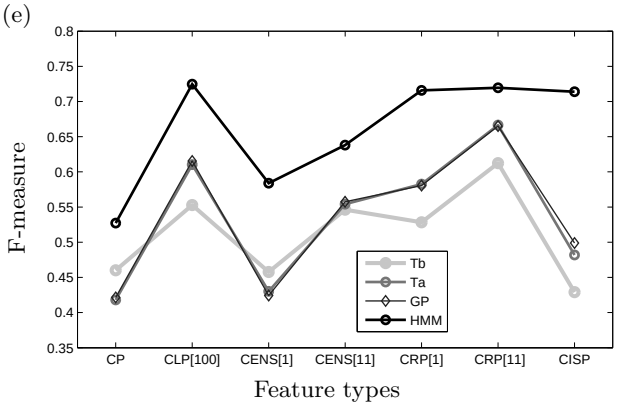

Figure 7.2. Dependency of recognition rate on feature type using **(a):** Training: $\mathcal{D}_1 \cup \mathcal{D}_2$, Test: $\mathcal{D}_3$, **(b):** Training: $\mathcal{D}_1 \cup \mathcal{D}_3$, Test: $\mathcal{D}_2$, **(c):** Training: $\mathcal{D}_2 \cup \mathcal{D}_3$, Test: $\mathcal{D}_1$, and **(d):** 3-fold cross validation on $\mathcal{D}$ (averaged over (a),(b) and (c)). **(e):** Visual representation of (d).

for $\mathcal{D}_2$ (Figure 7.2(b)), F = 0.522 for $\mathcal{D}_1$ (Figure 7.2(c)), and in average F = 0.527 for the entire dataset $\mathcal{D}$ (Figure 7.2(d)). This indicates that the partition and selection of training and testing data only has a marginal influence on the overall chord recognition results for this particular dataset.

In the following experiments, we only revert to the average values of the cross validation. As Figure 7.2(d) reveals, the chord recognition accuracies depend on the complexity of

the respective recognizer. For example, in the case of CLP[100], one obtains an F-measure value of F = 0.553 for the basic binary template-based method $\mathtt{T}^{\mathtt{b}}$. Considering training data to learn averaged templates, the accuracy of $\mathtt{T}^{\mathtt{a}}$ is increased to F = 0.610. Further adding covariance information as used in GP gives only slight improvements (F = 0.615). However, when using the most advanced method HMM one gets the highest accuracy of F = 0.725. The reason for this is that HMM introduces a context-aware smoothing in the classification stage. Considering the temporal context of chords leads to better results in comparison to the methods working in a purely framewise fashion.

Our results also reveal that the chord recognition quality substantially depends on the used feature type and implementation details. For example, using the most basic feature CP results in very low F-measure values (e. g., F = 0.527 for CP with HMM) regardless of which recognizer is used. However, simply applying a logarithmic compression enhancing weaker components of the feature leads to a significant increases in F-measure (e. g., F = 0.725 for CLP[100] with HMM).

CENS[1] shows a very similar behavior as CP ($w = 1$ actually disables the temporal smoothing on the feature side). This indicates that the internal quantization of these features is not beneficial for chord recognition. However, when applying a temporal smoothing by setting the window parameter $w = 11$ (corresponding to one second) the recognition accuracy significantly increases for all recognizers. This effect is even noticeable for HMM which already involves a smoothing in the classification step (F = 0.584 for CENS[1] and F = 0.638 for CENS[11]).

CRP[1] is designed to boost timbre invariance. These features already incorporate an internal logarithmic compression leading to similar results as for CLP[100] (F = 0.716 for CRP[1] with HMM). Further adding temporal smoothing on the feature side, the F-measure increases to F = 0.720 for CRP[11] with HMM. In particular, these features lead to high F-measures, even in the case of the simple framewise recognizers (e. g., F = 0.612 in the case of $\mathtt{T}^{\mathtt{b}}$). Here, a carefully designed feature seems to lessen the influence of the recognizer on the chord recognition accuracy, see also Figure 7.2(e) for a visual representation of the recognition results.

CISP attempts to emphasize harmonic components of the signal. This should improve the chord recognition quality for all recognizers. However, in practice, CISP shows a special behavior. On the one hand, using CISP results in high F-measure values in combination with HMM (F = 0.714). On the other hand, combining CISP features with any of the framewise recognizers $\mathtt{T}^{\mathtt{b}}$, $\mathtt{T}^{\mathtt{a}}$, and GP results in very low F-measure values (e. g., F = 0.429 for $\mathtt{T}^{\mathtt{b}}$). Here, one reason is that for this feature type the intensities of chroma bands corresponding to chord notes are only slightly more pronounced than those corresponding to non-chord notes. In general, the ratio of chroma intensities of chord notes to those of non-chord notes seems to have a large influence on the chord recognition results. In particular the frame-wise recognizers tend to be very sensitive to this ratio. Here, high ratios (as in CP and CENS[w]) as well as low ratios (as in CISP) lead to poor recognition results. HMM, however, is able to compensate for the low intensity ratios of CISP, but not for the high intensity ratios of CP and CENS[w], see Figure 7.2(e).

The results of the experiments discussed in this section show that the choice of the chroma feature has a significant influence on the different recognition procedures. Even the most

advanced recognizer HMM has a substantial dependence on the underlying feature type. Note that using an HMM-based recognizer in combination with a poor choice of chroma feature leads to results of lower quality than using a basic recognizer with a good feature (e. g., F = 0.527 for HMM with CP but F = 0.553 for $\mathtt{T}^{\flat}$ with CLP[100]). In particular, a logarithmic compression of the intensities as well as a temporal smoothing on the feature side have a beneficial effect, regardless of the recognizer used.

7.7 Importance of Tuning

The tuning in Western Music typically uses the equal-tempered scale, where the octave is divided into 12 equal semitones. Here, the pitch reference for the tuning is the concert pitch, which is set at 440 Hz and corresponds to A4. However, the exact setting of the concert pitch depends on the performing ensemble and may vary greatly. For example, in historical recordings often a lower tuning is used. Here, the concert pitch may range between 440 Hz and 415 Hz. However, modern orchestras tend to use a higher concert pitch ranging between 440 and 444 Hz. As a consequence, for the extraction of harmonic content from audio recordings it may be of great importance to consider the aspect of tuning in the chord labeling procedure.

In the following section, we conduct a simple brute-force baseline experiment to investigate the role of tuning in the context of chord labeling. Based on a template-based chord labeler as described in 7.2, we perform the chord labeling task by using different versions of chroma features. In addition to the twelve possible cyclic chroma shifts we use six differently shifted pitch filter banks to account for the fractional semitone shifts $0, 0.25, 0.33, 0.5, 0.67, 0.75$, for a description of the original filter bank see [54]. Here, using a shifted filterbank simulates a retuning of the audio. Altogether, this amounts to 72 chroma feature versions sampling the space of the chromatic scale. A similar strategy has been used e. g. by Gómez [18] when using 24-bin or 36-bin pitch class profiles.

Performing a frame-wise evaluation on the Beatles dataset $\mathcal{D}$ consisting of 180 songs (described in Section 7.6.1) we compute F-measures for the chord labeling task for all of the resulting 72 chroma versions. Then, we consider the chroma version which maximizes the F-measure. Table 7.1 shows the chord labeling results for eight selected songs as well as in average for all 180 songs. Here, F_{orig} refers to the F-measure using the original chroma features, whereas F_{tune} refers to the best possible F-measure. Furthermore, $Tune$ indicates the optimal tuning difference in semitones of the audio relative to the annotations using a suitably shifted filter bank. For example, for the song *Lovely Rita* the tuning difference amounts to 0.67 semitones according to our computations. As Table 7.1 shows, balancing out tuning deviations has a strong impact on the chord labeling result. For example, for the song *Lovely Rita* the F-measure amounts to only 3% using the original filter bank, whereas it amounts to 64.2% using a shifted filter bank of 0.67 semitones. Figure 7.3 shows the visualization of the chord labeling result for the beginning of this song. Here, on top the original chroma features are used, whereas on the bottom the shifted pitch filter bank is used. The visualizations clearly show that in the beginning of this song using the original filter bank almost all chords are misclassified, whereas using the shifted filter bank most of the chords are correctly classified.

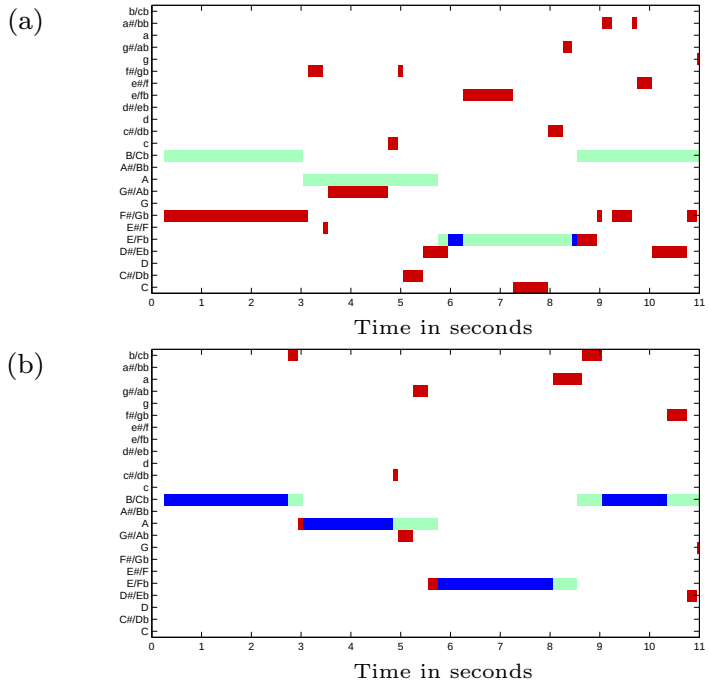

Figure 7.3. Visualization of the chord labeling result for the beginning of the Beatles song *Lovely Rita*. **(a)** Using the original filter bank, **(b)** Using a shifted pitch filter bank of 0.67 semitones.

Song	Tune	F_{orig}	F_{tune}
Lovely Rita	0.67	0.030	0.642
Strawberry Fields Forever	0.25	0.519	0.547
Wild Honey Pie	0.50	0.078	0.366
Ticket To Ride	0.25	0.361	0.604
Another Girl	0.67	0.181	0.446
Boys	0.33	0.304	0.434
You've Got To Hide Your Love Away	0.33	0.577	0.704
Do You Want To Know A Secret	0.50	0.346	0.503
Average		0.300	0.531
Average (180 Songs)		0.526	0.559

Table 7.1. Importance of balancing out tuning deviations in the context of chord labeling. The table shows F-measures for eight selected Beatles songs, in average, and in average for all 180 songs in the dataset. Here, F_{orig} refers to the F-measure using original chroma features and F_{Tune} to the maximal F-measure using a shifted pitch filter bank. $Tune$ indicates the optimal tuning difference in semitones of the audio relative to the annotations using a suitably shifted filter bank.

In Summary, balancing out the tuning deviations leads to an increase of the F-measure from 30% to 53.1% in average for the eight selected songs. Averaging over all 180 songs the F-measure increases by 3%. Note, that the above described experiment is really brute-force so that we do not focus on the absolute F-measure values. However, our experiments highlight the importance of a global re-tuning in the context of chord labeling. Considering only integer semitone shifts one may get poor F-measures throughout all shifts so that the consideration of fractional shifts is indispensible.

Chapter 8

Cross-Version Harmonic Analysis

The evaluation of chord labeling procedures is typically performed on large audio collections where the automatically extracted chord labels are compared to manually generated ground truth annotations. Here, a piece to be analyzed is typically represented by an audio recording produced under certain recording conditions, played on specific instruments and characterized by the individual styles of the musicians. As a consequence, the obtained chord labeling results are strongly influenced by version-dependent characteristics. Another major problem arises from the fact that audio-based recognition results refer to the physical time axis in seconds of the considered audio recording, whereas score-based analysis results obtained by music experts refer to a musical time axis given in bars. This simple fact alone makes it often difficult to get musicologists involved into the evaluation process of audio-based music analysis.

In this chapter, we introduce a cross-version approach, which uses synchronization techniques to analyze the harmonic properties of several audio versions synchronously. Here, the idea is to overcome the strong dependency of chord labeling results on a specific version. In particular, it turns out that consistencies across several versions indicate harmonically stable passages in the piece of music, whereas inconsistencies indicate version-dependent characteristics. Furthermore, we describe how to transform the time axis of analysis results obtained from audio recordings to a common musical time axis given in bars. This not only facilitates a convenient evaluation by a musicologist, but also allows for comparing analysis results across different recorded performances. Finally, we introduce a powerful visualization, which reveals the harmonically stable passages on a musical time axis specified in bars. The results of this chapter have been published in [34, 36].

The chapter is organized as follows. First, we present the cross-version framework (Section 8.1). In our experiments (Section 8.2), we exemplarily investigate the harmonic stability of consistently labeled passages. Furthermore, we demonstrate how the cross-version visualization facilitates a better understanding of classification errors in the case that score-based ground truth labels are provided by a music expert. Finally, we conclude in Section 8.3.

8.1 Cross-Version Framework

We now describe the cross-version chord labeling procedure shown in a schematic overview in Figure 8.1. At this point, we emphasize that our approach is not meant to be of technical nature, and we refer to [6, 47] for an overview of state-of-the-art chord labeling procedures. Instead, we introduce a simple yet powerful paradigm which exploits the availability of different versions of a given piece of music.

In the following, we first describe how synchronization procedures can be used to transform the time axis of audio-based analysis results to a performance-independent musical time axis (Section 8.1.1). Afterwards, we present the employed chord labeling procedure (Section 8.1.2), introduce the concept of cross-version chord labeling (Section 8.1.3) and illustrate the usefulness of our cross-version visualization by means of several music examples (Section 8.1.4). Finally, we describe an alternative procedure for transforming the time axis of audio-based analysis results to a common musical time axis (Section 8.1.5).

8.1.1 Musical Time Axis

The alignment techniques described in Chapter 2 can be used to transform the time axis of audio-based analysis results to a common musical time axis, see Figure 8.1 for an overview. To this end, we assume that for a certain piece of music we are given a MIDI representation of the musical score, where the MIDI time axis follows a musically meaningful time axis in bars. Such a MIDI file can be obtained by automatically exporting a score in computer-readable format, which in turn can be generated by applying OMR (optical music recognition) software to scanned sheet music, see Figure 8.1a. Now, given an audio recording of the same piece of music, one can apply music synchronization procedures to establish temporal links between the timelines of the MIDI representation and the audio version.

This linking information allows for transferring bar or beat positions from the MIDI timeline to corresponding time positions (given in seconds) of the audio timeline. Then, the audio timeline can be partitioned into segments each corresponding to e. g. one musical beat or bar. Based on this musically meaningful segmentation, beat- or bar-synchronous audio features can be determined. Here, each feature vector corresponds to a musically meaningful time unit that is independent of the respective recorded performance. We will use such synchronized features to directly compare the chord labeling results across the different versions.

An alternative procedure for transforming the time axis of audio-based analysis results to a common musical time axis is described in 8.1.5. This procedure allows for transferring annotations and chord labels from the score domain to the audio domain and vice versa. Using music synchronization techniques, the general idea is to locally warp the annotations of all given data streams onto a common time axis, which then allows for a cross-domain evaluation of the various types of chord labels.

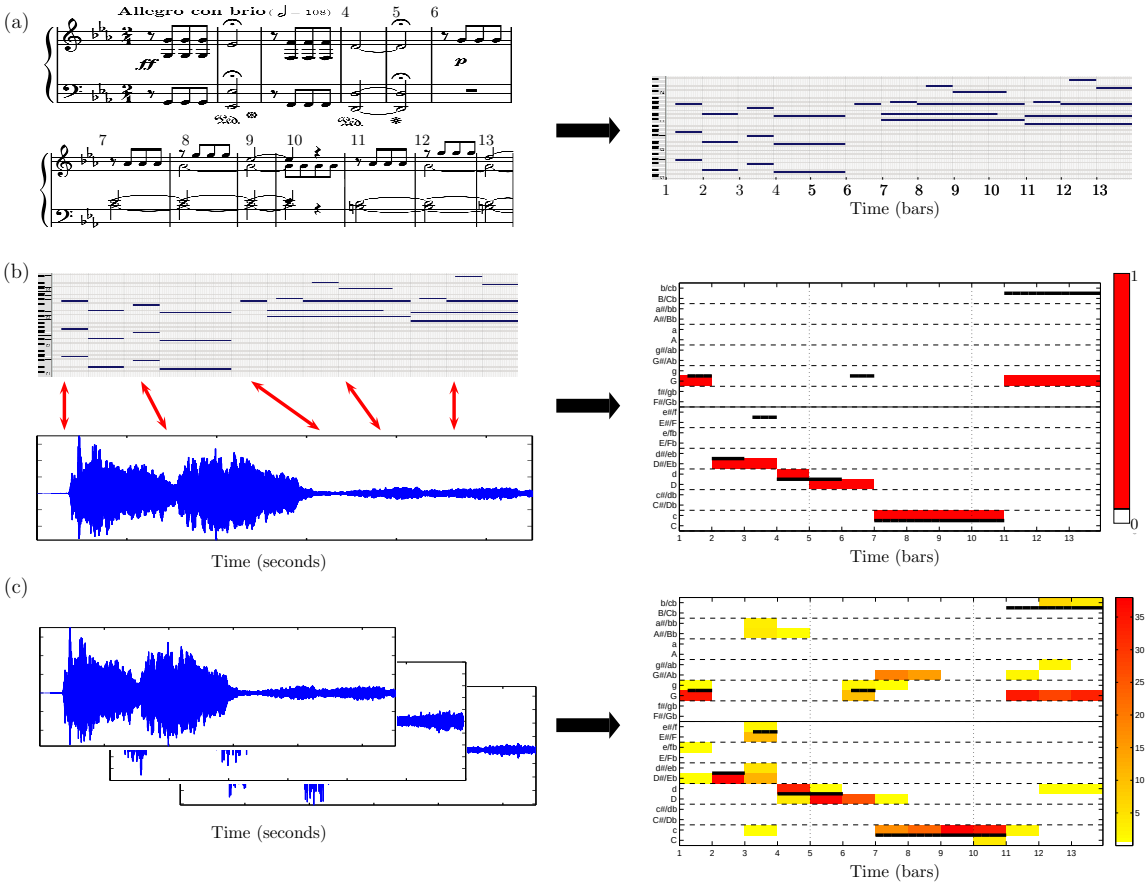

Figure 8.1. Schematic overview of the employed cross-version framework. Here, the beginning of Beethoven's Fifth (bb. 1-13) is used as an example. (a) Export of the score to a neutral MIDI representation. Here, the score corresponds to a piano reduction of Beethoven's Fifth. (b) Visualization of the automatically derived chord labels for a specific audio recording. The time axis in bars is obtained by synchronizing the audio recording with the MIDI representation. The horizontal black lines in the visualization represent the bassline extracted from the MIDI representation. (c) Cross-version visualization (38 different audio recordings). The horizontal black lines in the visualization represent the bassline extracted from the MIDI representation.

8.1.2 Chord Labeling

The chord labeling is then performed on the basis of the synchronized chroma features, where we furthermore apply a tuning estimation to balance out possible deviations of the performances from standard tuning [18, 47]. Note that numerous chord labeling procedures have been described in the literature. State-of-the-art chord recognizers typically employ statistical models such as hidden Markov models [40, 77, 86] or more general graphical models [47] to incorporate smoothness priors and temporal continuity into the recognition process, see Chapter 7. Since the respective chord labeling procedure is not in the focus of this chapter, we use a basic template-based chord labeling procedure [17] as described in Section 7.2, which better illustrates the kind of information that is enhanced and stabilized by our cross-version strategy. However, note that more complex chord recognizers can be used instead.

In the following, we consider 24 chord categories comprising the twelve major and the twelve minor chords, following the conventions as used for MIREX 2011 [53]. Let Λ denote the set of these 24 categories, then for each $\lambda \in \Lambda$ we define a binary template $\mathbf{t}_\lambda$ that corresponds to the respective chord. The template-based chord labeling procedure consists in assigning to each frame (here, exemplarily, we use a bar-wise frame level) the chord label that minimizes a predefined distance d (in our implementation, we use the cosine distance) between the corresponding template and a given feature vector referred to as x:

$$\lambda_x := \underset{\lambda \in \Lambda}{\operatorname{argmin}}\ d(\mathbf{t}_\lambda, x). \tag{8.1}$$

As result, we obtain for each audio version a sequence of automatically extracted bar-wise chord labels. Figure 8.1b shows the automatically extracted chord labels for a specific audio recording of the first 13 bars of Beethoven's Symphony No. 5, Op. 67, the so-called Beethoven's Fifth. The vertical axis represents the 24 chord categories, where major and minor chords with the same root note are visualized next to each other. Capital letters correspond to major chords, whereas lower case letters correspond to minor chords. The horizontal axis represents the time axis given in bars. The automatically derived chord labels are shown in red, e. g., the chord label for bar 1 corresponds to G major, whereas the chord label for bar 2 corresponds to E♭ major. As the bassline of a harmonic progression plays an important role for the understanding of harmonic structures, we have visualized it as an additional information in the middle of the corresponding major and minor chord having the bassline as root note. The bassline is automatically extracted from the MIDI representation by determining the lowest of all present MIDI notes at every point in time.

8.1.3 Cross-Version Chord Labeling

As previously mentioned, the chord labeling results not only depend on the piece of music but also on the acoustic and artistic characteristics of the specific audio recording. To alleviate the dependence on such characteristics, one can exploit the fact that for classical pieces of music usually many different recorded performances exist. Here, our idea is to perform the chord labeling across several versions of a given piece of music and then to resolve the dependency of the chord labels on a specific version by using some kind of late-fusion strategy. Since the automatically extracted chord labels for the different performances are given bar-wise, one can overlay the performance-specific chord labels for all considered recorded performances resulting in a cross-version visualization. Figure 8.1c shows a cross-version visualization for the beginning of Beethoven's Fifth (bb. 1-13), where 38 different performances are considered. The color-scale ranging from bright yellow to dark red indicates the degree of consistency of the chord labels across the various performances, where red entries point to consistencies and yellow entries to inconsistencies. For example, bar 2 is labeled highly consistently, whereas bar 3 is labeled inconsistently across the considered performances.

In this way, the cross-version visualization directly reveals chord label consistencies and inconsistencies across the different performances giving a deeper insight into the chord labeling procedure as well as the underlying music material. As we will show, consistently labeled passages generally correspond to harmonically stable passages, which are clearly

dominated by a certain key. In some cases, consistencies may also point to consistent misclassifications which might be taken as an indicator for inadequacies of the underlying chord labeling model. For example, considering only 24 major and minor chords, it is obvious that more complex chords such as, e. g., diminished chords can not be captured. In contrast, inconsistencies generally point to harmonically instable passages or ambiguities in the underlying music material. For example, incomplete chords as well as additional notes such as trills, appoggiaturas or suspended notes lead to chord ambiguities causing an inconsistent labeling across the different performances.

8.1.4 Examples

To illustrate our cross-version approach, we now discuss some real-world music examples. We first refer to the introductory bars of Beethoven's Fifth (see Figure 8.1). Figure 8.1b shows the visualization of the automatically derived chord labels for a specific audio recording. Following the time axis in bars, the visualization allows for a direct comparison to the score. As the score reveals the first five bars (bb. 1-5) do not contain complete triads. Instead, the characteristic "fate motif" appears, which is presented in octaves in unison. The visualization shows that the automatically derived chord labels for these introductory bars, aside from bar 3, are meaningful in the sense that they represent chords having the presented note of the respective bar as root note. However, in bar 3, where `f` is played in unison, `E♭` major is detected. This might be an indicator for inaccuracies in the synchronization since the previous bar (b. 2) is dominated by the note `e♭`. The same problem appears in bar 6. Bars 7-10 are then labeled as `C` minor. A closer look at the score reveals that in this passage (bb. 8-10) `C` minor is clearly present. However, in the beginning of this passage (b. 7) `C` minor with suspended sixth (`a♭`) leads into the `C` minor chord (bb. 8-10). In fact, `C` minor with suspended sixth corresponds to the notes of `A♭` major. However, the suspended sixth (`a♭`) is played in a very soft way in the considered recording, which might be the reason for the detection of `C` minor. Bars 11-13 then are labeled in a meaningful way as `G` major.

The cross-version visualization (Figure 8.1c) now directly reveals consistently and inconsistently labeled passages. For example, one observes the following highly consistently labeled passages, which may correspond to harmonically stable passages: bars 1-2, 4-5 and 8-13. As previously described, bars 1-2 and 4-5 refer to the fate motif in unison, thus not containing complete triads. These bars are now consistently labeled as a chord having the respective note of the considered bar as root note. Comparing bars 8-13 to the score shows that they indeed correspond to passages being clearly dominated by a certain harmony. Bars 8-10 are consistently labeled correctly as `C` minor reflecting the harmonic stability of this passage, which is clearly dominated by a `C` minor triad. Similarly, bars 11-13 are correctly identified by the visualization as harmonically stable, being dominated by `G` major. In contrast, one directly observes that bar 3 is labeled inconsistently. This inconsistent labeling may be due to local inaccuracies in the underlying synchronization procedure. For a larger amount of recordings this bar is labeled as `F` major (or as `E♭` major) having as root the note presented in unison in this bar (or in the previous bar). In fact, bar 3 was already misclassified as `E♭` major considering a single audio recording before. The cross-version visualization now clearly identifies this bar to be problematic in view

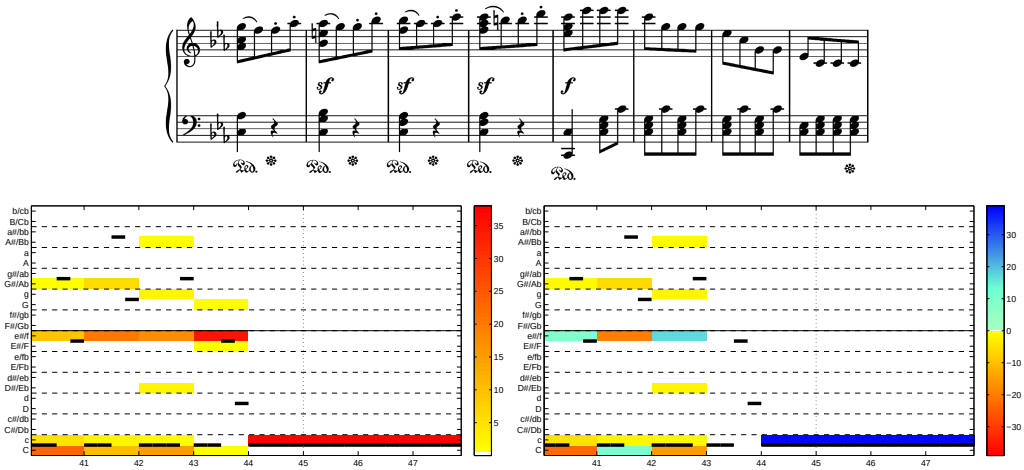

Figure 8.2. Cross-version visualization for Beethoven's Fifth (bb. 40-47). Here, 38 different audio recordings are considered. **Left:** Cross-version visualization of the automatically derived chord labels. **Right:** Cross-version visualization, where the automatically derived chord labels are overlayed with score-based ground truth chord labels.

of the underlying synchronization procedure. Finally, bar 7 attracts attention since it is labeled for approximately half of the recordings as C minor and as A♭ major for the other half. Here, C minor with suspended sixth (a♭) is present, which indeed sounds equivalently to A♭ major. Since the suspended a♭ is usually played in a soft way, for many recordings (including the previously discussed specific recording) this bar is misclassifed as C minor. However, the cross-version visualization shows that for the largest part of recordings this bar is correctly classified (with regard to the sound) as A♭ major.

As the previously discussed example shows, ground truth data is not necessarily needed to derive valuable information from the cross-version visualization concerning the employed chord labeling procedure as well as the underlying music material. However, assuming the case that score-based ground truth labels are provided by a trained musician, this information can be easily incorporated into our cross-version approach, see Figure 8.2. In this way, errors (deviations from the ground truth) can be subdivided into errors being specific to a certain audio version (inconsistent misclassifications) and errors independent of a specific version (consistent misclassifications). While inconsistent misclassifications may point to ambiguities in the underlying music material, consistent misclassifications may point to inadequacies in the underlying chord labeling framework.

We now exemplarily show how the cross-version visualization may serve as a useful tool, in the case that score-based ground truth labels are provided. The ground truth annotation has been generated by a music expert on the bar-level using the shorthands and conventions proposed by Harte et al. [27], see Section 8.2.1. Figure 8.2 shows the cross-version visualization for a different excerpt of Beethoven's Fifth (bb. 40-47). On the left, the previously introduced visualization is shown, where the automatically derived cross-version chord labels are visualized without considering ground truth chord labels. On the right, an extension of this cross-version visualization is presented, where the cross-version chord labels are compared to score-based ground truth labels. In this visualization we now distinguish two different color scales: one color scale ranging from dark blue to bright green

and the previously introduced color scale ranging from dark red to yellow. The first color scale from blue to green serves two purposes. Firstly, it encodes the score-based ground truth chord labels. Secondly, it shows the degree of consistency between the automatically generated audio labels and the score labels. For example, the dark blue entries in bars 44-47 show, that a `C` minor chord is specified in the score-based ground truth labels, and all automatically derived chord labels coincide with the score label here. In contrast, the bright green entry in bar 40 shows that the score-based chord label corresponds to `F` minor, but most of the automatically derived chord labels differ from the score label, specifying a `C` major chord. Analogously, the second color scale from dark red to yellow also fulfills two purposes. Firstly, it encodes the automatically derived chord labels that differ from the score-based labels. Secondly, it measures the universality of an error. For example, in bars 44-47 there are no red or yellow entries, since the score-based labels and the automatically derived labels coincide here. However, in bar 40 most automatically derived chord labels differ from the score-based labels. Here most chord labels specify a `C` major chord.

8.1.5 Procedure for Transferring Annotations

In Section 8.1.1 we have described how to transform the time axis of audio-based analysis results to a common musical time axis using beat- or bar-synchronous features. In the following, we present an alternative procedure, which allows for transferring annotations and chord labels from the score domain to the audio domain and vice versa. Here, we first compute the chord annotations on the physical time axis in seconds of the respective audio version before applying synchronization procedures to transform the time axis by locally warping all computed chord annotations of the several audio versions onto a common musical time axis, see Figure 8.3.

Therefore, we assume that we are given a MIDI representation of the musical score, where the MIDI time axis follows a musically meaningful time axis in bars. Additionally, chord labels manually annotated by a trained musician on the basis of a score are given as well as labels automatically derived from the audio recording via some computer-based method. In the first step, we derive CENS features from the MIDI as well as from the audio, say $X := (x_1, x_2, \ldots, x_N)$ and $Y := (y_1, y_2, \ldots, y_M)$, respectively. Since each CENS feature corresponds to a time frame, we can also create two binary chord vector sequences, $A := (A_1, \ldots, A_N)$ and $B := (B_1, \ldots, B_M)$, which encode the given chord labels in a framewise fashion. Here, $A_n, B_m \in \{0,1\}^d$ for $n \in [1:N]$ and $m \in [1:M]$. The constant d equates the number of considered chords, in our case $d = 24$, since we consider in the following only the 24 major and minor chords. A value of one in a vector component encodes the chord prevalent in the corresponding time frame. Using the first four bars of Chopin's Mazurka Op. 68 No. 3 as an example, we illustrate the sequences A for the score and B for the audio in Figure8.3(b) and 8.3(c), respectively. Note that in Figure8.3(b) the time is expressed in terms of bars, while in Figure8.3(c) the time is given in seconds. This different notion of time prevents a comparison of A and B at this point.

The next step consists of synchronizing the two CENS features sequences X and Y as mentioned in Section 2. The resulting alignment path $p = (p_1, \ldots, p_L)$ encodes temporal correspondences between elements of X and Y. Following the same time frame division, the alignment path also encodes correspondences between the sequences A and B. Using

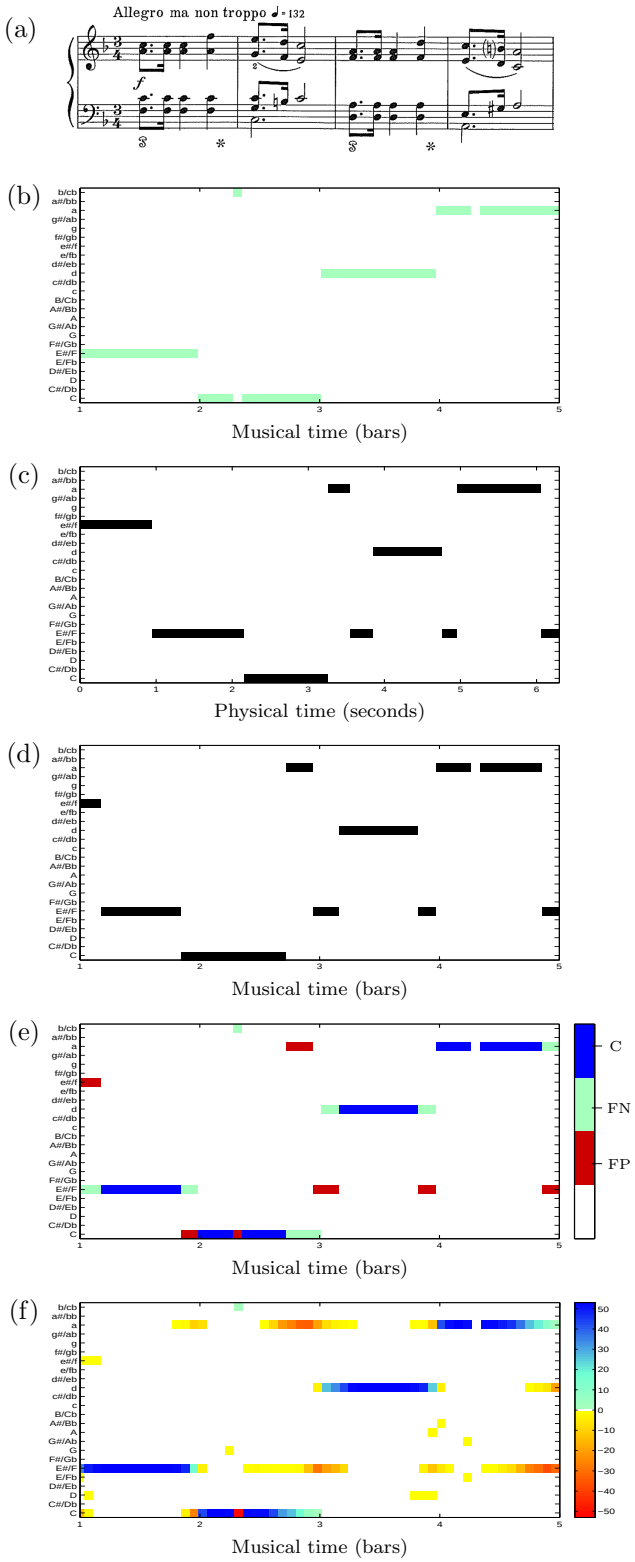

Figure 8.3. Various chord annotations visualized for the Chopin Mazurka Op. 68 No. 3 (F major), bars 1-4. **(a)** Score. **(b)** Score-based ground truth chord labels. **(c)** Automatically derived audio chord labels (physical time axis). **(d)** Warped audio chord labels (musical time axis). **(e)** Overlayed score and audio chord labels. **(f)** Cross-version visualization, where the automatically derived audio chord labels for 51 different recorded performances are overlayed with score-based ground truth chord labels.

this linking information, we locally stretch and contract the audio chord vector sequence B according to the warping information supplied by p. Here, we have to consider two cases. In the first case, p contains a subsequence of the form

$$(n, m), (n + 1, m), \ldots, (n + \ell - 1, m)$$

for some $\ell \in \mathbb{N}$, i. e., the ℓ score-related vectors $A_n, \ldots, A_{n+\ell-1}$ are aligned to the single audio-related vector B_m. In this case, we duplicate the vector B_m by taking ℓ copies of it. In the second case, p contains a subsequence of the form

$$(n, m), (n, m + 1), \ldots, (n, m + \ell - 1)$$

for some $\ell \in \mathbb{N}$, i. e., the score-related vector A_n is aligned to ℓ audio-related vectors $B_m, \ldots, B_{m+\ell-1}$. In this case, we replace the ℓ vectors by the vector $B_{m+\lfloor \ell/2 \rfloor}$. The resulting warped version of B is denoted by $\bar{B}$. Note that the length of $\bar{B}$ equals the length N of A, see Figure 8.3(d). For the visualization we set all vectors in $\bar{B}$ to 0, where no groundtruth chord label is available, as for example in the middle of bar 4, see Figure 8.3(d).

Overall, we have now converted the physical time axis of the audio chord vector sequence B to the musically meaningful bar axis, as used for A. Finally, we can visualize the differences between the score-based and the audio-based chord labels by overlaying A and $\bar{B}$, see Figure 8.3(e). Blue entries now indicate areas, where the ground truth labels and the audio chord labels coincide. On the contrary, green and red encode the differences between the chord labels. Here, green entries correspond to the ground truth chord labels derived from the score, whereas red entries correspond to the audio chord labels. For example, at the beginning of bar 2 the score as well as the audio chord labels indicate a `C` major chord. On the contrary, at the end of bar 2 there is a `C` major chord specified in the score, while the chord labels derived from the audio incorrectly specify an `A` minor chord.

In Figure 8.3(f) the cross-version visualization for the first four bars of the Chopin Mazurka is represented. Here, we warped the automatically generated chord labels for 51 different audio recordings onto the musical time axis using the steps described above. By overlaying the resulting chord vector sequences $\bar{B}$ for all versions, we obtain a cross-version visualization as introduced in 8.1.3.

8.2 Experiments

In the following, we first describe the score-based ground truth annotations used in our experiments (Section 8.2.1). Based on several music examples, we then discuss the harmonic stability of consistently labeled passages in the cross-version visualization (Section 8.2.2). Finally, we demonstrate how the visualization may be used for an in-depth analysis of chord recognition errors (Section 8.2.3).

8.2.1 Annotations

For the cross-version evaluation we manually annotated the chords for the following four pieces of Western classical music: Chopin's Mazurka Op. 68 No. 3, Bach's Prelude in `C`

major BWV 846, the first movement of Beethoven's Fifth Symphony, Op. 67, and the first movement of Beethoven's Op. 27 No. 2, the so-called "Moonlight Sonata". Using the underlying score, the annotations were created on the beat-level, for Chopin's Mazurka even on a finer level. In the case of a bar-wise evaluation, the beat-wise given annotation is transformed into a bar-wise annotation, by assigning to each bar the chord label existing for more than half of the considered bar. In this context, bars for which no prevalent chord label exists are left unannotated. The format and naming conventions used for the annotation were proposed by Harte et al. [27]. The annotator paid much attention to capture even slight differences between adjacent chords. Hence, the bass tone as well as missing or added tones in chords are marked explicitly using the corresponding shorthands. As we consider in the evaluation only the 24 major and minor chords, we have to map the given chord labels to one of these 24 chords in a meaningful way. To this end, we employ the interval comparison of the dyad, which was used in MIREX 2011 [53] and takes into account only the first two intervals of each chord. Thus, augmented and diminished chords are mapped to major and minor, respectively, as well as any other label having a major or minor third as its first interval.

8.2.2 Harmonic Stability

By means of two music examples, we now exemplarily investigate the harmonic stability of consistently labeled passages. As a first example, we choose again bars 40-47 of Beethoven's Fifth, which already served as example in Section 8.1.4. The cross-version visualization of the automatically derived chord labels (see Figure 8.2, left) reveals two highly consistently labeled passages: bar 43, labeled highly consistently as `F` minor, and bars 44-47, which are labeled as `C` minor across all considered recorded performances. Comparing to the score, bars 44-47 indeed turn out to be a harmonically stable passage which is clearly dominated by `C` minor. Consequently, this highly consistently labeled passage is labeled correctly, which is shown in the visualization, where the automatically derived chord labels are compared to score-based ground truth labels (see Figure 8.2, right). In contrast, bar 43 is labeled consistently as `F` minor (see Figure 8.2, left), but comparing to the score one finds out that besides of an `F` minor chord two additional notes (`b` and `d`) are contained in this bar, suggesting the dominant `G` major. Therefore, a clear assignment of a triad is not possible on the bar level. This is also the reason that there is no score-based label assigned to this bar in the ground truth annotation (see Figure 8.2, right). The remaining bars are labeled rather inconsistently indicating harmonic instability or ambiguities in the underlying music material (see Figure 8.2, left). A closer look at the score reveals that these bars are characterized by suspended notes on the first beat. These additional notes which do not belong to the underlying chords are mainly responsible for the inconsistent labeling. The comparison with the score-based ground truth annotation reveals that for bars 40 and 41 indeed most of the automatically derived chord labels differ from the ground truth annotation (see Figure 8.2, right).

Figure 8.4 shows the cross-version visualization for an excerpt of Bach's Prelude BWV 846 in `C` major (bb. 11-15), where five different recorded performances are considered. The visualization reveals 3 bars which are labeled correctly with high consistency (b. 11, b. 13, and b. 15) and two bars, which are misclassified for most of the considered audio versions

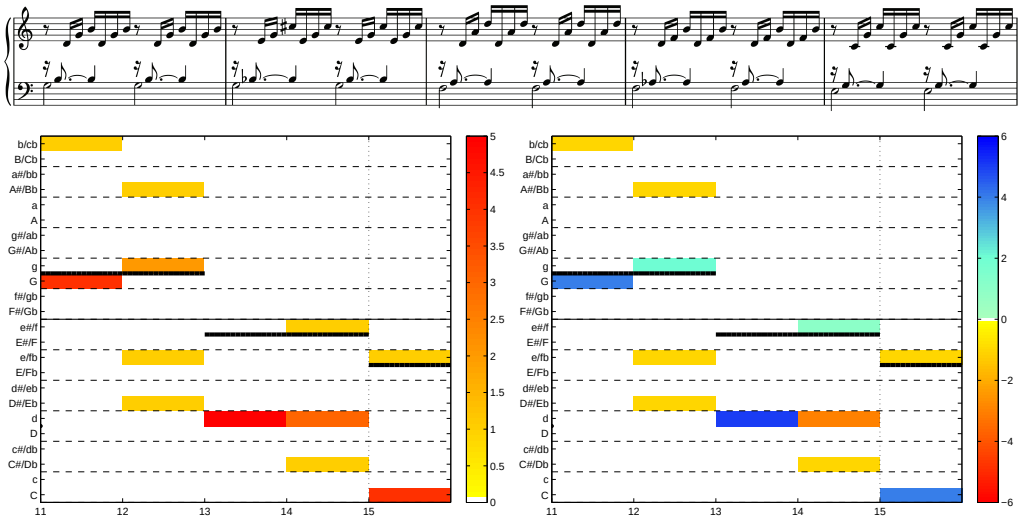

Figure 8.4. Cross-version visualization for Bach's Prelude BWV 846 in C major (bb. 11-15). Here, five different audio recordings are considered. **Left:** Cross-version visualization of the automatically derived chord labels. **Right:** Cross-version visualization, where the automatically derived chord labels are overlayed with score-based ground truth chord labels.

(b. 12 and b. 14). Comparing to the score one finds out that the correctly labeled passages indeed correspond to bars, where clear major or minor chords are present. In contrast, bars 12 and 14 are problematic in the sense that they contain diminished seventh chords which can not be assigned in a meaningful way to one of the considered 24 major and minor chords, thus producing misclassifications. In this case, an extension of the considered chord categories to also include diminished seventh chords might solve the problem.

8.2.3 In-Depth Error Analysis

None of the currently available automatic chord labeling approaches works perfectly. Errors can either be caused by the inherent ambiguity in chord labeling, or by a weakness special to the employed chord labeler. An in-depth analysis allowing for a distinction between these error sources is a very hard and time-consuming task. We show how this process can be supported and accelerated using the evaluation and visualization framework presented in Section 8.1. In the following examples we use the procedure described in Section 8.1.5 for transforming the time axis of the audio-based analysis results to a common musical time axis.

We start our in-depth error analysis with Chopin's Mazurka Op. 68 No. 3, which already served as example in Section 8.1.5. Figure 8.5 shows again the chord labeling result for a specific audio recording as well as the cross-version visualization for the first four bars of the piece. We first refer to the visualization of the chord labeling result for a specific audio recording, which clearly reveals various chord recognition errors, see Figure 8.5(e). Making use of the musical time axis, these errors can now easily be traced back to the corresponding position in the score and analyzed further. For example, at the beginning of the piece, the score-based ground truth annotation corresponds to F major, whereas the computed audio-based annotation corresponds to F minor. A mix-up of major and

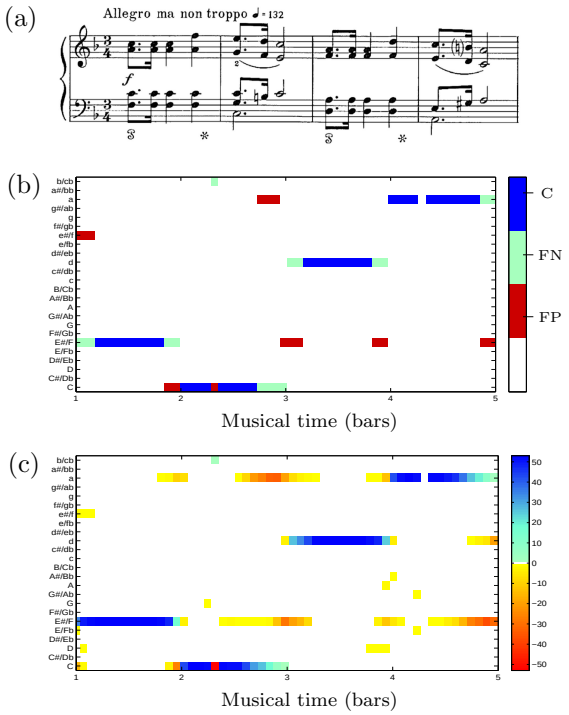

Figure 8.5. Chopin Mazurka Op. 68 No. 3, bars 1-4. **(a)** Score. **(b)** Visualization of the chord labeling result for a specific audio recording. **(c)** Cross-version visualization, where the automatically derived chord labels are overlayed with score-based ground truth chord labels.

minor often appears in the chord recognition task. The next misclassification occurs at the end of bar 1, where the ground truth still corresponds to `F` major, but the computed annotation specifies a `C` major, which is actually the subsequent chord in the ground truth. This may be a boundary problem or an error in the synchronization.

In the middle of bar 2, we note that the ground truth chord is `B` minor, whereas the computed chord is `C` major. Having a look at the score, one can see that the chord in question is actually a B diminished chord. Due to the reduction of the manual annotation to major/minor chords, this chord is mapped to a `B` minor chord in the ground truth. Causing a misclassification here, this is often a problem in the major/minor evaluation based on the comparison of the dyad.

The next misclassifications are due to the musical ambiguity of chords. At the end of bar 2 we observe in the score a `C` major chord, where the fifth is missing. Comparing on the dyad level, this chord is mapped to a `C` major chord in the ground truth. However, all the notes of the chord (`c`, `e`) are also part of an `A` minor chord, which is actually computed at this position. A similar problem occurs at the beginning and at the end of bar 3, where the ground truth annotation corresponds to `D` minor, whereas the computed annotation corresponds to `F` major. The same phenomenon appears a last time at the end of bar 4, where `F` major is recognized instead of `A` minor. This phenomenon is caused by ambiguities inherent to the chord labeling task and constitutes a very common problem. The chords in classical music rarely are pure major or minor chords, because tones are often missing

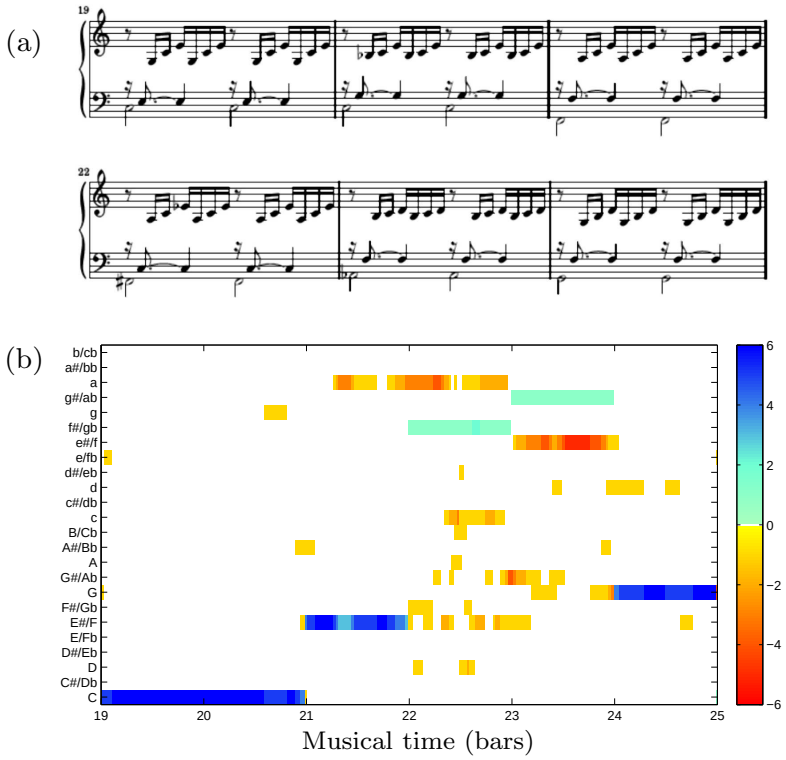

Figure 8.6. Bach BWV 846, bars 19-24. **(a)** Score, **(b)** Cross-version visualization, where the automatically derived chord labels are overlayed with score-based ground truth labels.

or added. Hence, the recognition as well as the manual annotation process become a hard task.

Next, we illustrate what kind of additional information our cross-version visualization can provide compared to the previously discussed visualization that only makes use of a single audio recording. Looking for consistencies and inconsistencies across the chord recognition results for 51 different audio recordings, see Figure 8.5(f), it is possible to classify and investigate single errors even further. For example, the misclassified `F` minor chord in the beginning of bar 1 (see Figure 8.5(e)) seems to be an exception for the specific recording. This can be clearly seen from the cross-version visualization where only for a few of the 51 audio recordings `F` minor is computed instead of `F` major. Also, the misclassification at the end of bar 4 (`F` major instead of `A` minor) is not consistent across all considered audio recordings. On the contrary, some of the misclassifications which we observed in the case of one audio recording (Figure 8.5(e)), are consistently misclassified for most of the other audio recordings. For example, the diminished chord in the middle of bar 2, the chord ambiguity problem occuring at the end of bar 2 (`A` minor instead of `C` major), the beginning of bar 3 (`F` major instead of `D` minor) and the end of bar 4 (`F` major instead of `A` minor). Overall, the cross-version visualization allows for a classification of recognition errors into those specific to a recording and those independent of a recording.

As a further example we now consider the famous Bach Prelude in `C` major, BWV 846. The cross-version visualization for 5 different audio recordings for bars 19-24 (see Figure 8.6) again reflects the chord recognition problems related to diminished chords. At the beginning of the excerpt (bb. 19-21) and at the end (b. 24) the chord recognition result

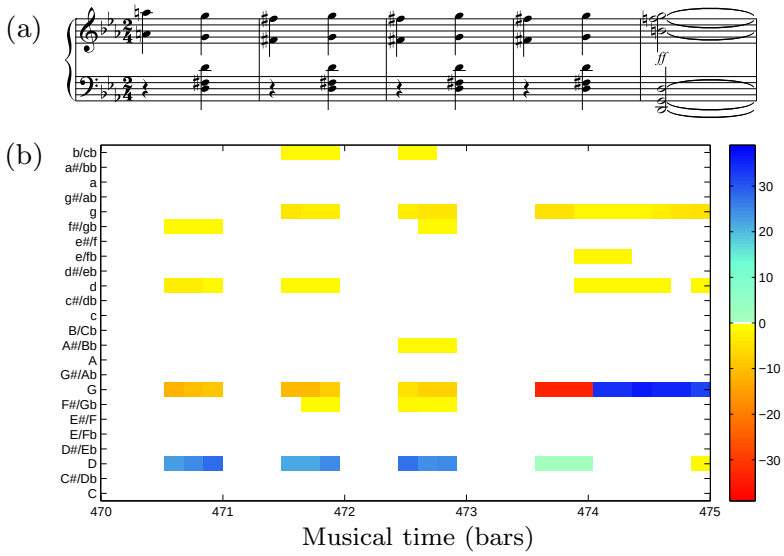

Figure 8.7. Beethoven's Fifth, bars 470-474. **(a)** Score, **(b)** Cross-version visualization, where the automatically derived chord labels are overlayed with score-based ground truth labels.

for all audio recordings consistently agrees more or less with the ground truth. However, one can observe two passages with green entries in bars 22-23. Looking at the corresponding position in the score, we find two diminished seventh chords, in bar 22 an F♯ : dim7 and in bar 23 an A♭ : dim7. Due to the reduction to major/minor chords these two chords are mapped to F♯ minor and A♭ minor in the ground truth annotation, respectively, see Figure 8.6. However, in most audio recordings an A minor chord is detected instead of F♯ : dim7, having two tones (a and c) in common. And instead of the A♭ : dim7 chord an F minor chord is found, for which even all three tones are present (f, a♭ and c) due to the additional passing note C in the A♭. While the seventh chord in bar 20 is recognized well for all recordings, we see that in bar 21 the F major seventh chord was mistaken for an A minor chord, again due to chord ambiguity reasons.

As a last example we now consider the first movement of Beethoven's Fifth Symphony in 37 different audio recordings. Actually, this piece of music is much more complicated in terms of harmonic aspects than the previously considered Chopin and Bach examples. In the Beethoven example, we can often find the musical principles of suspension, passing notes or "unisono" passages. Here, the automatic chord recognition as well as the manual annotation are challenging and ambiguous tasks. One example for the use of nonharmonic tones in chords can be found in bars 470-474, visualized in Figure 8.7. Looking at the score, we observe in the left hand a D major chord with a missing fifth (bb. 470-473), but in the right hand a g is added in octaves to this D major chord. Being the fourth of d, the g can be seen as a nonharmonic tone in D major. This causes a chord misclassification for about 15 recordings, where G major or alternatively G minor is computed. On the contrary, the G seventh chord in bar 474 is recognized very well for all recordings. Note that the first beats of bars 470-474 are not manually annotated, since the octaves do not represent meaningful chords.

Another example of a musical pattern that is found to be extremely problematic in the chord recognition task, is the principle of suspension. We illustrate the problems related

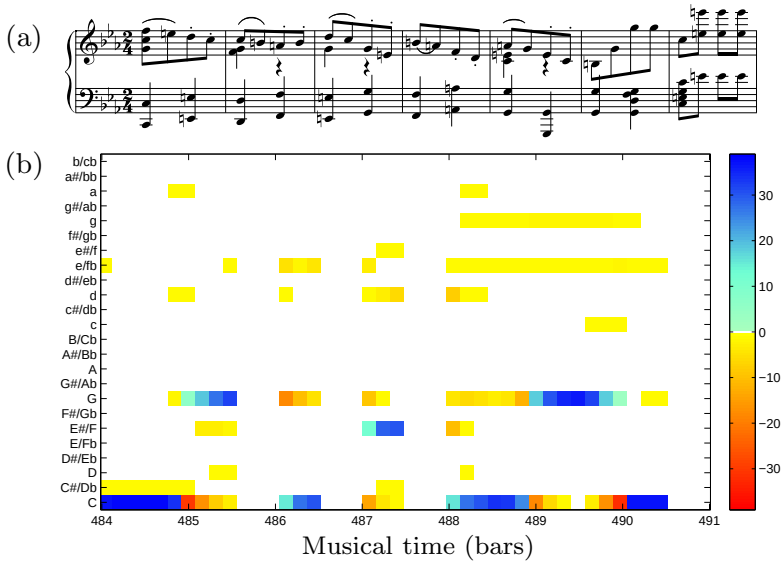

Figure 8.8. Beethoven's Fifth bars 484-490. **(a)** Score, **(b)** Cross-version visualization, where the automatically derived chord labels are overlayed with score-based ground truth labels.

to this musical characteristic using another excerpt (bb. 484-490) of Beethoven's Fifth, see Figure 8.8. In each of bars 484-488, one can find a suspension on the first eighth, which resolves into a major chord on the second eighth. This musical characteristic can easily be spotted in the cross-version visualization. Here, we see that at the beginning of each bar the number of audio recordings for which the computed annotation agrees with the ground truth is very low and gets higher afterwards. In bar 490 finally the first complete pure major chord is reached. Note that the second beats of bars 485-487 consist of passing notes to the next suspension. Hence, a meaningful chord cannot be assigned resulting in several beats missing a ground truth annotation.

8.3 Conclusions

In this chapter, we presented a cross-version approach which allows for comparing chord label annotations across different performances and across different domains (e. g. symbolic, MIDI, audio). In Chapter 9, such a cross-version approach will serve as basis for evaluating two MIDI-based chord labeling procedures using annotations given for corresponding audio recordings. Our experiments indicate, that consistently labeled passages across different performances often correspond to harmonically stable passages, whereas inconsistencies point to harmonically instable regions in the piece of music. In fact, analyzing the harmonic properties of several audio versions synchronously, one can achieve a stabilization of the chord labeling results, which will be of central importance in Chapter 10. Presenting the cross-version analysis results on a musically meaningful time axis in bars also helps to make the analysis results better accessible to music experts. Firstly, the presented approach allows for involving musicologists in the evaluation process of automated chord labeling procedures. For example, the cross-version visualization opens the way for an interdisciplinary collaboration, where musicologists may greatly support

computer scientists in performing an in-depth error analysis of the employed chord labeling procedure based on the score. Secondly, the cross-version visualization may serve musicologists as a helpful tool for exploring harmonic structures of a musical work, which we will illustrate in our case study on Beethoven's Appassionata presented in Chapter 11.

Chapter 9

Cross-Version Evaluation

For a given piece of music, there often exist multiple versions belonging to the symbolic (e. g. MIDI representations), acoustic (audio recordings), or visual (sheet music) domain. Each type of information allows for applying specialized, domain-specific approaches to music analysis tasks. In this chapter, we realize the idea of cross-version harmonic analysis (introduced in Chapter 8) to automatically evaluate two MIDI-based chord labeling procedures using annotations given for corresponding audio recordings. Using a novel late-fusion approach that combines different alignment procedures in order to identify reliable parts in synchronization results, the cross-version comparison of the various chord labeling results is performed only on the basis of the reliable parts. We show how inconsistencies in these results across the different versions allow for a quantitative and qualitative evaluation. In particular, we perform an in-depth error analysis of the two symbolic chord labelers, classify possible error sources, and illustrate the respective error source by means of concrete song examples. This qualitative error analysis not only indicates limitations of the employed chord labeling strategies but also deepens the understanding of the underlying music material. This chapter is based on [14].

The chapter is organized as follows. In Section 9.1 we present an overview on symbolic chord labeling and in particular, describe the two symbolic chord labelers, which are used in the evaluation presented in Section 9.2. Finally, we conclude in Section 9.3.

9.1 Symbolic Chord Labeling

Symbolic chord labeling deals with the computer-based harmonic analysis of symbolic music data. In the field of symbolic chord labeling, only few procedures have been introduced [50, 67, 69, 75, 82, 89]. Early symbolic chord labeling procedures are typically rule-based and refer to musical knowledge of Western Classical music [50, 89]. For example, in [50] a method for performing roman numeral analysis from symbolic music data is introduced. Furthermore, Sleator and Temperley proposed several procedures for performing harmonic analyses which are now part of the Melisma Music Analyzer [78]. The Melisma system is one of the two chord labelers which are used in our evaluation and described in more detail in Section 9.1.1. Hidden Markov models have also been used

in the context of symbolic chord labeling, see [69]. In the following, we shortly describe the two symbolic chord labeling procedures which we will evaluate in Section 9.2. For a detailed description of the two symbolic chord labelers we refer to [15].

9.1.1 Temperley's Melisma

The first symbolic chord labeling procedure which is used in our evaluation is referred to as *Melisma* [82]. The input of Melisma is a list of MIDI pitches including the onset and offset times as well as some metrical information.[1] Based on this information, the module *harmony* estimates the root note of the chord in a certain time window using a system of preference rules [41, 82]. Afterwards, the module *key* derives keys for all given segments using a Bayesian model [83] before in a last step the actual chord labels are produced in the form of a roman numeral analysis.[2] Temperley's system is a combination of different procedures: Preference rule systems, a Bayesian algorithm and the final procedure for labeling the corresponding chord given the key and the root note. Furthermore, it depends on different classes of parameters: hard-coded parameters, user-defined parameters and learned parameters.

For our evaluation described in Section 9.2 we use information about root note, mode (major, minor, unspecified) and the fifth (perfect, diminished, unspecified) as well as the onset and offset times. This finally results in considering the three possible chord classes major, minor and diminished.

9.1.2 A Bayesian Model Selection Algorithm

The second symbolic chord labeler used in our evaluation is referred to as *RLM*. This system follows a Bayesian approach for chord labeling [71]. Here, all relevant parameters are part of the same modeling procedure and the most likely chord model is determined by Bayesian model selection. Since *RLM* focuses mainly on popular music it assumes only triad chords in the following six possible chord classes: major, minor, diminished, augmented, sus2, and sus4.

9.2 Evaluation

Exploiting the availability of multiple versions of a given piece of music, we have suggested the concept of a cross-version analysis for comparing and/or combining analysis results

[1] In our experiments, we provided *harmony* with the information about quarter and sixteenth notes given by the MIDI file instead of deriving metrical information by the *meter* program.

[2] Note, that key's output is a roman numeral analysis, whereas chord labels are only contained in key's internal data structure. Testing key's roman numeral analysis for popular music showed, unsurprisingly, that many chords were uninterpretable resulting in the assigned label *Chr* (standing for chromatic and indicating that it is not possible to derive the respective chord from a major or a minor scale by adding thirds to one of the scale notes.) Therefore, we by-passed the roman numeral analysis in order to access the chord labels of key's internal data structure.

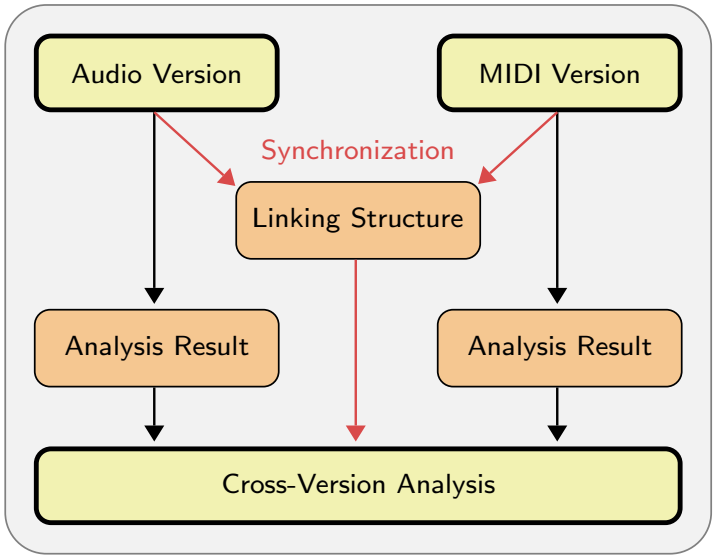

Figure 9.1. Cross-version music analysis based on synchronization techniques (from [14]).

across the versions (see Chapter 8). We now exemplarily apply this concept to automatically evaluate the two MIDI-based chord labelers *RLM* and *Melisma* from Section 9.1 whose performance has not been clear so far, since no ground truth annotations for MIDI versions have been available on a larger scale. We evaluate the two symbolic chord labelers on the well-known Beatles dataset, where chord annotations are available for corresponding audio recordings (Section 9.2.1). Figure 9.1 shows the cross-version analysis procedure in a schematic overview.

In the following, we first describe the experimental setup (Section 9.2.1) and introduce a cross-version visualization (Section 9.2.2). Afterwards, we present a quantitative evaluation (Section 9.2.3) before demonstrating how the cross-version visualization can greatly support a user in a qualitative analysis of the recognition errors (Section 9.2.4).

9.2.1 Experimental Setup

In our evaluation we exploit the audio data chord annotations provided by Christopher Harte, who manually annotated all 180 songs of the 12 Beatles studio albums [27]. Harte's annotations are generally accepted as the de-facto standard for evaluating audio-based chord labeling methods. Transferring these annotations from the acoustic to the symbolic domain allows for an efficient reuse of the existing ground truth for the evaluations of symbolic chord labelers. Furthermore, having a common set of ground truth across all available musical domains presents a starting point to identify exactly those positions in a piece where a method relying on one music representation has the advantage over another method, and to investigate the underlying musical reasons.

Our evaluation dataset consists of 112 songs out of the 180 songs. For these 112 songs we not only have an audio recording with annotated chord labels, but also a corresponding MIDI version. Given a MIDI file and a corresponding audio recording, we start our evaluation by computing a MIDI-audio alignment.

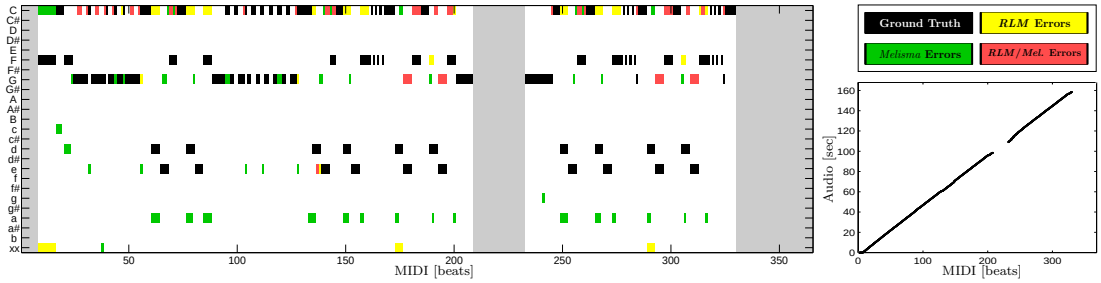

Figure 9.2. Cross-version chord evaluation for the song *Getting Better*. **Left:** Overlay of two MIDI-based chord labeling results (*Melisma* and *RLM*) and manually generated audio-based chord labels. In the visualization the ground truth labels (black), the errors of *Melisma* (green), the errors of *RLM* (yellow), and errors appearing for *Melisma* and *RLM* (red) are indicated. **Right:** Consistency alignment (horizontal axis specifies MIDI time in beats and vertical axis specifies audio time in seconds).

Because the MIDI versions often differ significantly, at a local level, from the audio recordings, we cannot simply employ global synchronization techniques. Therefore, we employ a consistency alignment, which identifies those sections that can be aligned reliably. The main idea of this method is to employ a late-fusion approach that combines several types of conceptually different alignment strategies instead of relying on one single strategy. Looking for consistencies and inconsistencies across the synchronization results, the method automatically classifies the alignments locally as reliable or critical. In Figure 9.2 (right) the consistency alignment for the song *Getting Better* is visualized, showing that one section in the MIDI file (between beat 210 and 230) could not be reliably aligned to a corresponding section in the audio. For a detailed description of the consistency alignment we refer to [15].

Using the linking information provided by the alignment, we compute for each MIDI beat the corresponding position in the audio version. Using this linking information, we then transfer the audio-based chord labels to the MIDI version. If more than one audio chord label exists in the audio segment associated with a MIDI beat, we simply choose the predominant chord label as MIDI annotation. As the result, we obtain a beatwise chord label annotation for the MIDI version.

For our evaluation, we compare the transferred ground truth annotations to the automatically generated chord labels obtained from *Melisma* and *RLM* on the basis of the 12 major and the 12 minor chords. Therefore, using the interval comparison of the triad as used for MIREX 2010 [52], all ground truth chord labels are mapped to one of these 24 chords. Here, both a seventh chord and a major seventh chord are mapped to the corresponding major chord. However, augmented, diminished or other more complex chords cannot be reduced to either major or minor and therefore are omitted from the evaluation.

9.2.2 Visualization

Using synchronization techniques allows for visualizing different chord recognition results simultaneously for multiple versions. Such cross-version visualizations turn out to be a powerful tool for not only analyzing the chord label results but also for better understanding the underlying music material [36]. We introduce our visualization concept by means of

a concrete example shown in Figure 9.2. Here, the chord labels generated by *Melisma* and *RLM* are visualized along with the transferred ground truth annotations using a common MIDI time axis given in beats (horizontal axis). The vertical axis represents the 24 major and minor chords, starting with the 12 major chords and continuing with the 12 minor chords. Associated with each beat is a black entry representing the ground truth chord label that we transferred to the MIDI files. For example, in Figure 9.2, a `G` major chord label is assigned to beat 50. The colored entries in the figure are used to indicate where the two automatic chord labelers differ from the manual annotation. Here, yellow and green entries indicate that *RLM* and *Melisma* differ from the manual annotation, respectively. For example, in the beginning of the song the green entries show that *Melisma* detected a `C` major chord, while the ground truth specified an `F` major chord. If a chord labeler generated a chord label that cannot be reduced to either major or minor, then this is indicated by a colored entry in the 'xx' row. For example, in the beginning of the song *RLM* detected a complex chord corresponding to a yellow entry in the 'xx' row. Sometimes, both automatic chord labelers differ from the ground truth, but agree on the same chord label. Such consistent deviations from the ground truth are marked in red. An example can be found around beat 200, where both automatic chord labelers specify a `C` major chord instead of an `F` major chord in the ground truth. Furthermore, areas in the figure with a gray background indicate beats for that no ground truth is available. For example, in Figure 9.2, this can be observed between beat 210 and 230. Here, our consistency alignment, given on the right in the figure, shows that this section in the MIDI file could not be reliably aligned to a corresponding section in the audio. Furthermore, a ground truth annotation might also be unavailable for a beat if the chord label at that position is irreducible to major or minor—for example, if the chord label specifies an augmented chord.

Overall, our visualization allows for the identification of two different classes of inconsistencies. On the one hand, red entries in the visualization reveal positions, where the two chord labelers consistently differ from the ground truth. Here, the reason for the error may be of extrinsic or musical nature, independent of the specific chord labeler. On the other hand, yellow and green entries indicate intrinsic errors of the respective chord labeler. Thus, our visualization constitutes a useful tool to identify interesting or problematic passages in the audio recording.

9.2.3 Quantitative Evaluation

We now quantitatively evaluate the two MIDI-based chord labelers. Table 9.1 presents the results for nine exemplarily chosen songs as well as an average over all 112 pieces in our database. For each song, the precision values of *Melisma* and *RLM* are listed. Here, precision indicates the percentage of the manually annotated beats correctly classified by the respective chord labeler. Also, the alignment coverage (AC), which specifies the percentage of the MIDI version that has been aligned to the respective audio version, is listed.

As can be seen from Table 9.1, the precision of *RLM*, averaged over all 112 songs, is 82%, whereas that of *Melisma* is only 72%. Using Bayesian model selection, *RLM* seems to be more data adaptive and performs better in our experiments than *Melisma*, depending

Table 9.1. Results of the cross-version chord evaluation for *RLM* and *Melisma*. The four columns indicate the piece/dataset, the alignment coverage (AC), as well as the precision (Prec) for the two methods.

Piece	AC	Prec *RLM*	Prec *Melisma*
AnotherGirl	97	98	60
DoctorRobert	99	76	60
EightDaysAWeek	99	92	74
EverybodysTryingToBeMyBab	95	71	85
GettingBetter	83	60	52
GoodDaySunshine	82	85	55
InMyLife	97	90	75
IWannaBeYourMan	91	61	42
Money	56	38	11
Average	89	75	57
Average over all 112 songs	86	82	72

on some hard-coded parameters. Furthermore, *Melisma* is tuned towards classical music, whereas *RLM* focuses on popular music, which might be advantageous with regard to the Beatles dataset.

Even though such a quantitative evaluation gives a general indication on the algorithms' performances, it is not very helpful for the understanding of the algorithmic or musical reasons of the recognition errors. We now show how our visualization framework can be used for a more in-depth analysis of the chord recognition results.

9.2.4 Qualitative Evaluation

Our cross-version visualization directly reveals two different types of errors: *extrinsic errors* that are independent of the employed chord labeling strategy (marked by red entries) as well as *intrinsic errors* of the two chord labelers (marked by yellow and green entries). In the following, we further detail on this observation by exemplarily performing a qualitative error analysis by means of some concrete song examples.

First, we discuss some typical intrinsic errors of the two chord labelers. For *Melisma*, it turned out that one main error source consists in confusing major and minor. Here, the song *Another Girl* (Figure 9.3) serves as an example. As can be clearly seen from the visualization, *Melisma* recognizes most of the time `A` minor instead of `A` major. On the contrary, most of *RLM*'s errors are produced by specifying a complex chord label instead of a major or minor label in the ground truth. For example, looking at the song *Doctor Robert* (Figure 9.4), one notices that an `A` major chord is annotated from beat 1 to beat 57 in the ground truth, whereas *RLM* often specifies a more complex chord corresponding to the 'xx' row. Taking into account six different chord classes (major, minor, diminished, augmented, sus2, sus4), *RLM* is susceptible to choose such a complex chord label instead of a simple major or minor chord label. Here, a manual inspection revealed that also simplifying assumptions in the manually generated audio annotations (taken as ground truth) and the reduction process are sources for confusion and ambiguity. For example, in the song *Doctor Robert* `A` major and `A sus4` are played alternately. However, the manual

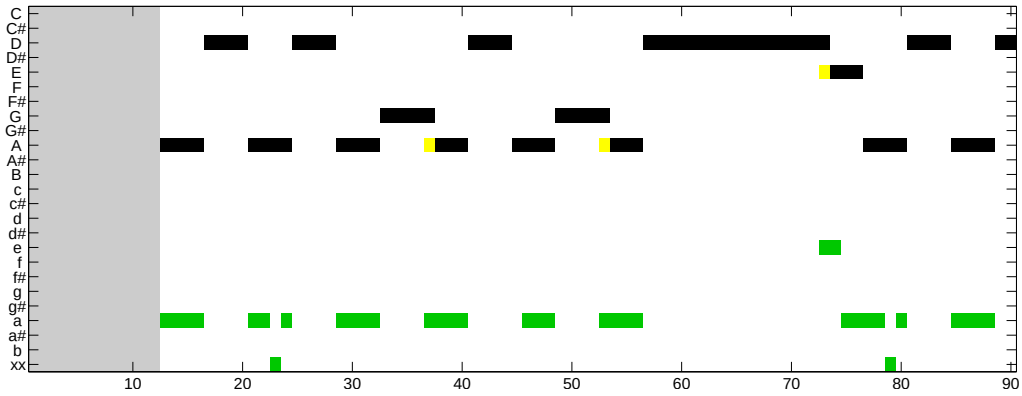

Figure 9.3. Cross-version chord label visualization for the song *Another Girl* (Beat 1-90).

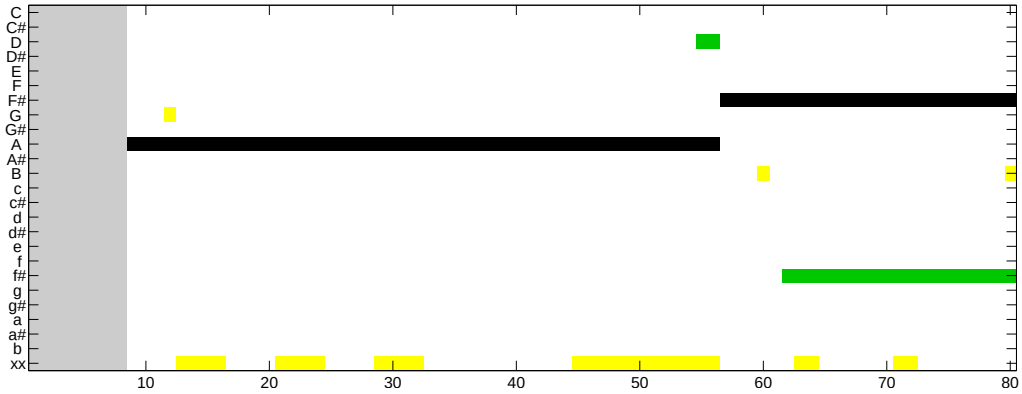

Figure 9.4. Cross-version chord label visualization for the song *Doctor Robert* (Beat 1-80).

annotation simply corresponds to `A` major so that *RLM*'s classification as an `A sus4` chord (which is actually correct) appears as a misclassification. Furthermore, also in the *Doctor Robert* (Figure 9.4) example, *Melisma*'s confusion of major and minor appears again, where `F♯` minor is recognized instead of `F♯` major from beat 62 to beat 80.

The second type of error sources are extrinsic errors, which are errors that appear consistently for both chord labelers (marked by red entries). Such consistent misclassifications may appear for several reasons. Having performed an in-depth error analysis allows us to categorize these errors into the following four subclasses. Firstly, a consistent misclassification can appear due to errors in the synchronization. Looking at the visualization of the song *In My Life* (Figure 9.5), one can see the three consistent misclassifications in red. Appearing at the borders of the determined chord labels and specifying either the previous or the subsequent ground truth label as chord label, these misclassifications are characteristic for inaccuracies in the synchronization.

Secondly, inaccuracies in the manual ground truth annotations can be responsible for consistent misclassifications. Here, it should be noted that Harte's manual annotations are created for the guitar accompaniment, so further voices (for instance, the melody voice) remain unconsidered in the manual chord labels. However, many Beatles songs contain passages where the chord of the guitar accompaniment is overlayed with another chord

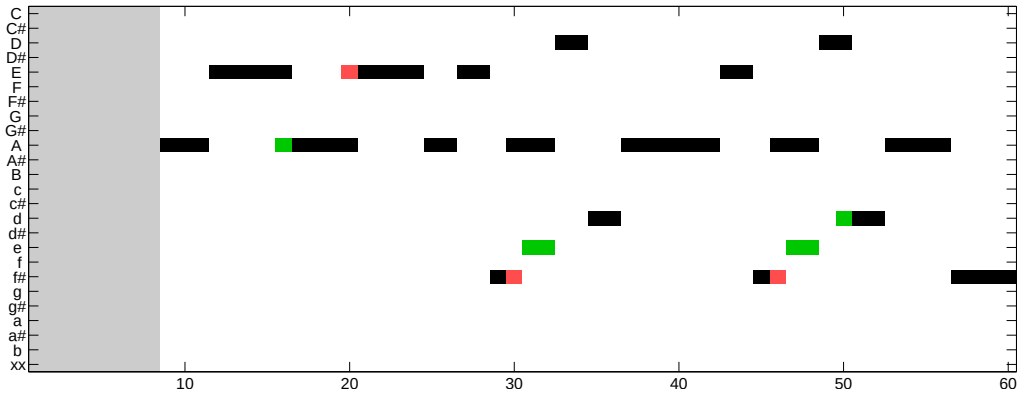

Figure 9.5. Cross-version chord label visualization for the song *In My Life* (Beat 1-60).

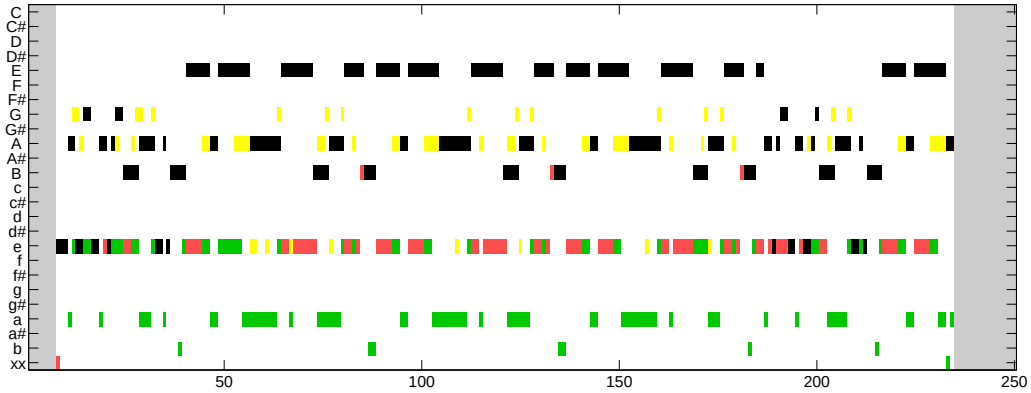

Figure 9.6. Cross-version chord label visualization for the song *Money* (Beat 1-250).

played by other instrument. For example, in *Money* (Figure 9.6), E major and E minor are sometimes overlaid[3]. Here, the ground truth specifies E major, referring only to the guitar chord. However, both chord labellers specify E minor instead of E major most of the time, this being the predominant chord at these positions.

Thirdly, a harmonic difference between the MIDI and the audio version may lead to a consistent deviation of the two chord labelers from the ground truth. In the song example *Good Day Sunshine* (Figure 9.7), one observes the three red entries at E minor around the beats 55, 70 and 110, whereas the ground truth specifies E major. These consistent misclassifications are due to a deviation of the MIDI version from the audio version. In the MIDI version an important tone in the leading voice is changed from originally g♯ (which is part of E major) to g (which is part of E minor), leading to the misclassification E minor for both chord labellers.

Finally, as the fourth subclass, we detected errors that are caused by musical reasons. For example, the use of suspensions or the presence of passing notes and other nonharmonic tones often lead to local chord ambiguities. In particular, the leading voice often contains

[3]Arguably, this bluesy chord is neither minor nor major; it is an example of why chord classifiers need to develop beyond the simplistic assumption that everything is based on triads.

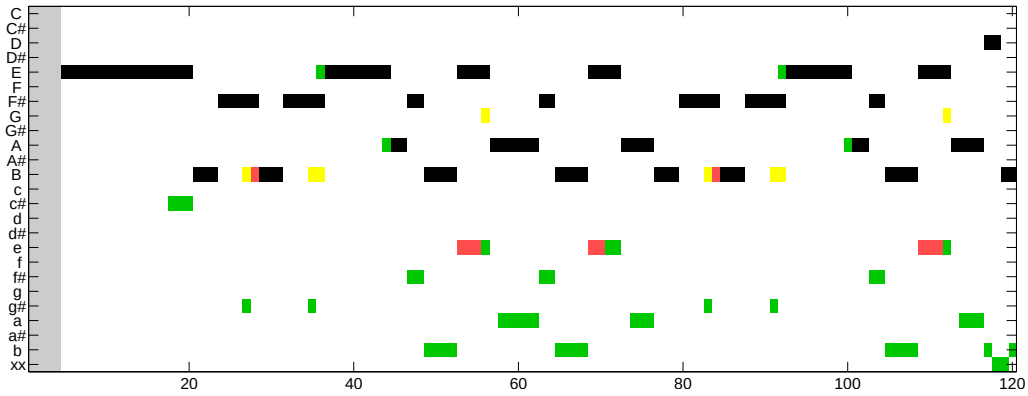

Figure 9.7. Cross-version chord label visualization for the song *Good Day Sunshine* (Beat 1-120).

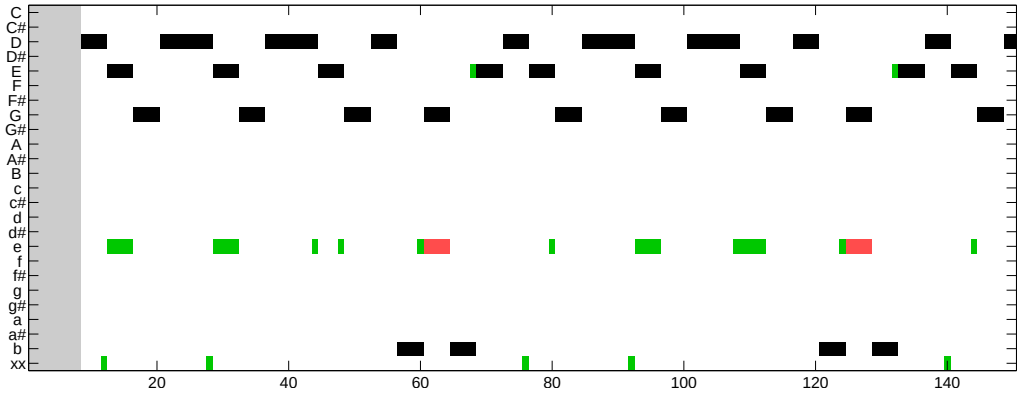

Figure 9.8. Cross-version chord label visualization for the song *Eight Days A Week* (Beat 1-150).

nonharmonic tones with regard to the underlying harmony. Precisely this phenomenon appears e. g. in the song *Eight Days A Week* (Figure 9.8) at beat 60, where the underlying chord is `G` major, which is also the labeled chord in the ground truth. However, both chord labelers specify an `E` minor chord here. This is due to the nonharmonic tone `e` in the leading voice, which, together with the tones `g` and `b`, forms an `E` minor chord.

Having classified the errors appearing in the chord labeling process, we now consider two further examples, which illustrate combinations of the error sources described above. For example, in *Everybody's Trying To Be My Baby* (Figure 9.9), *RLM* specifies the beats 107 to 123 as `E` minor, whereas *Melisma* agrees with the ground truth, labeling an `E` major chord. At first sight, it looks as though *RLM* mixed up major and minor. However, the reason for the misclassification is a harmonic difference between the MIDI and the audio version. In the MIDI version, an `E` minor seventh chord sounds in this passage, but in the audio version and therefore in the ground truth an `E` major chord is present. Therefore, the `E` minor classification of *RLM* is correct, whereas the `E` major classification of *Melisma* turns out as a misclassification being caused by its typical error source, the confusion of major and minor.

In the visualization of the next example, *I Wanna Be Your Man* (Figure 9.10), one observes two large passages of misclassification, the first from beat 8 to 75 and the second from beat

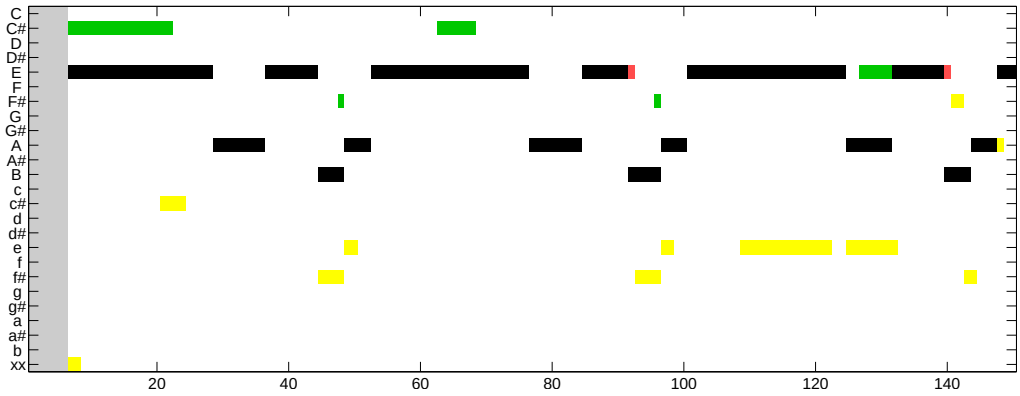

Figure 9.9. Cross-version chord label visualization for the song *Everybody's Trying To Be My Baby* (Beat 1-150).

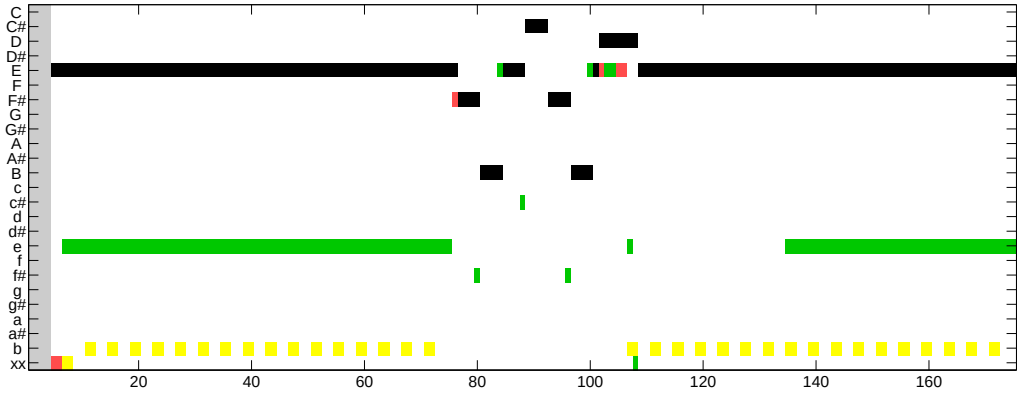

Figure 9.10. Cross-version chord label visualization for the song *I Wanna Be Your Man* (Beat 1-175).

135 to 175. Here, both chord labellers deviate from the ground truth, which specifies `E` major. *Melisma* labels an `E` minor chord, whereas *RLM* detects a `B` minor chord. Actually, the reasons for the misclassification of the respective chord labeller differ. On the one hand, in this passage there is a permanent harmonic change between an `E` major chord and an `E` minor seventh chord. Due to inaccuracies in the manual annotation, the ground truth specifies `E` major for the whole passage, thus neglecting the `E` minor seventh chord, which is also present in the MIDI version. This inaccuracy in the manual annotations is the reason for the misclassification of *Melisma*, labelling `E` minor. On the other hand, we can find the following nonharmonic tones in the leading voice: `b`, `d` and `f♯`, which form a `B` minor chord. This ambiguity is responsible for the misclassification of *RLM*, which specifies `B` minor.

9.3 Conclusions

In this chapter, we demonstrated the utility of our cross-version framework by exemplarily applying the concept of cross-version harmonic analysis for the evaluation of MIDI-based

chord labeling methods using audio-based ground truth annotations. Performing an in-depth analysis of error sources, we indicated limitations of the employed chord labeling strategies and exemplified how our framework facilitates interdisciplinary research. Visualizations and interfaces based on our framework allow even a technically unexperienced user to perform an error analysis of automatically generated annotations. This opens the way for a collaboration between computer scientists, providing the knowledge about the underlying chord labeling strategies, and musicologists providing a trained ear and the expertise about harmonic relations.

Chapter 10

Stabilizing Audio Chord Labeling

In this chapter, we show that analyzing the harmonic properties of several audio versions synchronously (by using a cross-version approach as introduced in Chapter 8) stabilizes the chord labeling result in the sense that inconsistencies indicate version-dependent characteristics, whereas consistencies across several versions indicate harmonically stable passages in the piece of music. In particular, we show that consistently labeled passages often correspond to correctly labeled passages. Our experiments document that the cross-version labeling procedure significantly increases the precision of the result while keeping the recall at a relatively high level. The results of this chapter have been published in [34].

This chapter is organized as follows. First, we introduce a cross-version voting strategy (Section 10.1) before presenting a simple constraint-based strategy which uses a single version (Section 10.2). In our experiments (Section 10.3), a comparison of these two strategies demonstrates that our voting strategy is conceptually different from simply imposing stricter conditions in the template-based approach. Finally, we conclude in Section 10.4.

10.1 Cross-Version Voting Strategy

By overlaying the chord labeling results as described in Section 8.1.3 for the first 19 bars of Beethoven's Piano Sonata Op. 27 No. 2, the so-called Moonlight Sonata, considering seven different audio versions, we obtain a cross-version visualization, see Figure 10.1b. The cross-version strategy now reveals consistencies and inconsistencies in the chord labeling across all audio versions. For example, one directly notices that the misclassification in bar 10, when considering a specific audio version (see Figure 10.1e), seems to be version-dependent. Considering several audio versions, bar 10 is more or less consistently labeled correctly as `E` minor. In contrast, a more consistent misclassification (`C` major instead of `E` minor was labeled for four versions) can be found in bar 16.

In the following experiment, we investigate to which extent the consistency information across several audio versions may be exploited to stabilize chord labeling. In the *majority voting strategy* we keep for each bar exactly one of the automatically extracted chord labels,

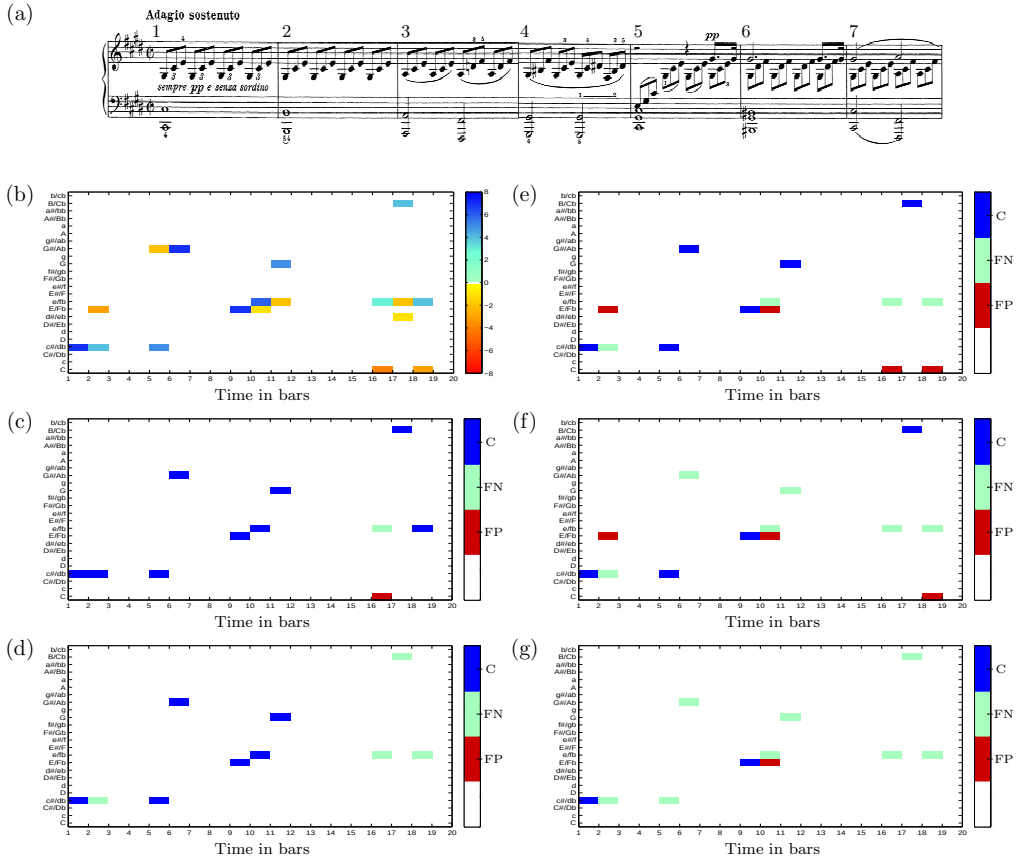

Figure 10.1. Visualization of the chord labeling result for Beethoven's Moonlight Sonata (bb. 1-19). In the left column (b-d) the cross-version voting strategy is used considering seven performances, whereas in the right column (e-g) the constraint-based strategy is used considering only a single audio recording (Barenboim). Bars, for which no score-based ground truth label exists (since the clear assignment of a harmony is not possible), are left unconsidered in the evaluation. (a) Score of bars 1-7. (b) Visualization of consistencies and inconsistencies in the cross-version analysis. (c) Cross-version majority voting strategy. (d) Cross-version voting strategy with $\nu = 0.5$. (e) Basic strategy. (f) Constraint-based strategy with $\gamma = 0.3$. (g) Constraint-based strategy with $\gamma = 0.1$.

namely the most consistent chord label across all versions. All remaining audio chord labels are left unconsidered in the evaluation. This results in a visualization which is shown in Figure 10.1c. Blue entries (correct: C) now indicate areas, where the audio chord label agrees with the ground truth chord label. In contrast, green and red entries encode the differences between the chord labels. Here, red entries (false positives: FP) correspond to the audio chord labels, whereas green entries (false negatives: FN) correspond to the ground truth labels. As one directly notices, besides one misclassification in bar 16, the above mentioned highly consistent error, all chords are now correctly classified resulting in a significant increase of precision.

In the next step, we further constrain the degree of consistency by introducing a consistency parameter $\nu \in [0, 1]$. To this end, we consider only bars which are labeled consistently for more than $(\nu \cdot 100)\%$ of the audio versions. All other bars are left unannotated.

For example, $\nu = 0.5$ signifies that we keep in the evaluation only passages, where for more than 50% of the audio versions the extracted chord labels agree. Figure 10.1d shows the visualization of the chord labeling result for $\nu = 0.5$, where the voting procedure succeeds in eliminating all misclassifications. At the same time only three correct classifications are taken out of the evaluation. In this way, the precision further increases (amounting to 100% in Figure 10.1d), while the recall still remains on a relatively high level (amounting to 60% in Figure 10.1d).

As the example described above shows, the cross-version voting approach succeeds in significantly increasing the precision, while keeping the recall at a relatively high level. For a quantitative evaluation of the cross-version voting strategy we refer to the experiments described in Section 10.3.

10.2 Constraint-Based Strategy

To better illustrate the potential of our cross-version voting strategy, we now consider a constraint-based stabilizing procedure. Using the template-based approach described in Section 8.1.2, the automatically derived chord label for a given bar is defined by the template having the minimal distance to the feature vector, in the following referred to as *basic strategy*. Figure 10.1e shows a visualization of the chord labeling result. As the visualization reveals the first bar is correctly identified as C♯ minor, whereas bar 2 is misclassified, being identified as E major although being labeled as C♯ minor in the ground truth. Here, a C♯ minor 7th chord is present in the ground truth, being mapped to C♯ minor. In fact, this seventh chord contains all the tones for E major, which explains the misclassification.

As we can see from the example, using the basic strategy, it obviously happens that for bars containing complex chords none of the given 24 templates fits well to the present feature vector. Here, the chord template of minimal distance may have a rather large distance to the feature vector. To counteract this case, we now introduce a parameter $\gamma \in [0, 1]$, which represents an upper threshold for the distance between the assigned chord template and the feature vector. In this way, we obtain a constraint-based procedure, where only chord labels λ are kept for which

$$d(\mathbf{t}_\lambda, x) < \gamma. \tag{10.1}$$

All feature vectors x that have a larger distance than γ to any of the chord templates are left unannotated. In the following experiment, the idea is to successively decrease the parameter γ in order to investigate its influence on the chord labeling result.

Figure 10.1f shows the visualization for $\gamma = 0.3$. Obviously, one misclassification (bb. 16) is now taken out of the evaluation. However, at the same time two previously correctly classified chords (bb. 6, bb. 11) are left unconsidered in the evaluation, resulting in a decrease of the recall. Here, again seventh chords are present being correctly classified but having a relatively large distance to the template vector. Further decreasing the parameter γ is accompanied by a dramatical loss in recall while the precision increases moderately (Figure 10.1g). For quantitative results of the evaluation of the constraint-based strategy we refer to the experiments shown in Figure 10.2.

Composer	Piece	# (Versions)	Identifier
Bach	Prelude C Major BWV 846	5	'Bach'
Beethoven	Moonlight Sonata Op. 27 No. 2 (first movement)	7	'BeetM'
Beethoven	Fifth Symphony Op. 67 (first movement)	38	'Beet5'
Chopin	Mazurka Op. 68 No. 3	49	'Chopin'

Table 10.1. Overview of the pieces and number of versions used in our experiments.

10.3 Experiments

In this section we quantitatively evaluate the various chord labeling strategies using a dataset that comprises four classical pieces of music, see Table 10.1. At this point, we want to emphasize that our main object is not in increasing the F-measure, defined below. Instead, in the application we have in mind, we are interested in finding passages, where one obtains correct chord labels with high guarantee. Therefore, our aim is to increase the precision, however, without losing too much of the recall.

In the following, we denote the automatically derived audio chord labels as L_a, and the ground truth chord labels as L_{gt}. For our bar-wise evaluation, we use precision (P), recall (R) and F-measure (F) defined as follows:

$$P = \frac{\#(L_a \cap L_{gt})}{\#L_a}, \quad R = \frac{\#(L_a \cap L_{gt})}{\#L_{gt}}, \quad F = 2 \cdot \frac{P \cdot R}{P + R}. \tag{10.2}$$

We first discuss the cross-version voting strategy. Figure 10.2 shows curves for P, R and F for the four pieces in the dataset, where the horizontal axis now represents the parameter ν ranging between 0.5 and 0.8 except for the position labeled by 'Maj' corresponding to the majority voting strategy. First of all, one notices that performing the chord labeling across several versions using the majority voting strategy, precision, recall and F-measure already improve by 10-30% in comparison to the basic strategy based on a specific version (see 'Min' in Figure 10.2).

Furthermore, for all four examples the precision rapidly increases, so that for $\nu = 0.5$ already a high precision is reached: 95% (Bach), 94% (BeetM), 100% (Chopin) and 77% (Beet5). At the same time the recall remains on a rather high level, still amounting to 59% (Bach), 77% (BeetM), 76% (Chopin) and 63% (Beet5). In this way, our experiments show that consistently labeled passages across several versions often correspond to correctly labeled passages. Increasing the consistency parameter ν further increases the precision values, while the recall still remains at acceptably high levels. In summary, exploiting the consistency information of the chord labels across several versions succeeds in stabilizing the chord labeling, resulting in a significant increase of precision without loosing too much of the recall.

We now compare these results with the ones obtained from the constraint-based strategy. Figure 10.2 shows curves for P, R, and F for the four pieces in our dataset. Here, P, R,

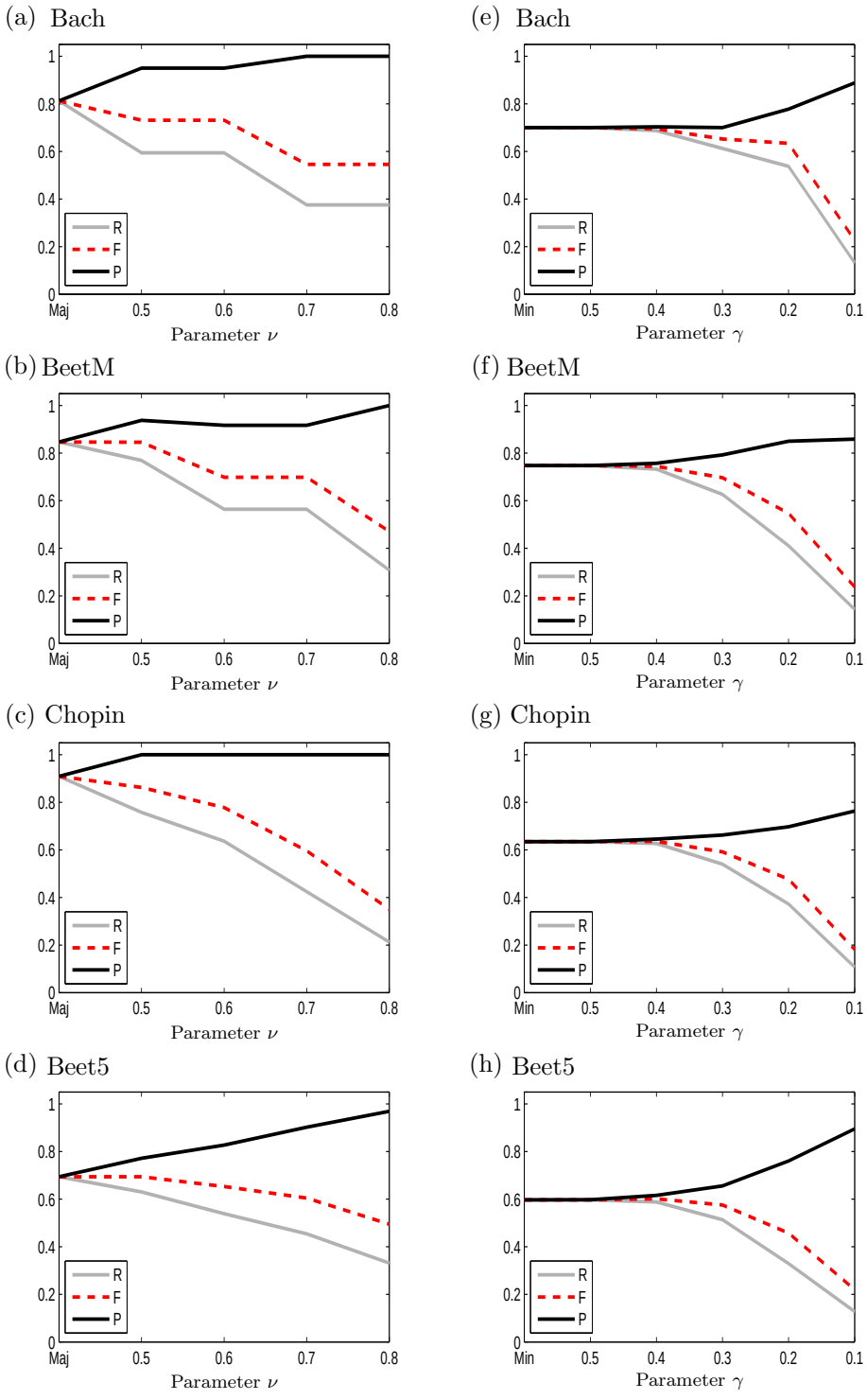

Figure 10.2. **Left:** Cross-version voting strategy. Curves for precision (P), recall (R) and F-measure (F) using the majority voting strategy (Maj), and four different consistency parameters ν from 0.5 to 0.8. **Right:** Constraint-based strategy based on a specific version. Curves for the mean value of precision (P), recall (R) and F-measure (F) using the basic strategy (Min) and five different settings for γ from 0.5 to 0.1.

	Mean	**Min**	**Max**
Bach	0.7000	0.4375	0.8750
BeetM	0.7473	0.6923	0.8718
Chopin	0.6345	0.4545	1.0000
Beet5	0.5967	0.5282	0.8345

Table 10.2. Basic chord labeling based on specific versions. The table shows mean, minimum and maximum F-measures over all recorded performances of a given piece.

and F correspond to mean values, which are obtained by first applying the constraint-based strategy on every version in the dataset separately and then averaging over all these versions.

In the visualization, the horizontal axis represents the parameter γ ranging between 0.5 and 0.1 except for the position labeled by 'Min' corresponding to the basic labeling strategy. As one directly notices, there is a clear tendency visible for all four examples in our database. For increasing γ the precision also slowly increases reaching a high value of roughly 80% for $\gamma = 0.1$. However, at the same time the recall dramatically drops down to roughly 10% for $\gamma = 0.1$. Obviously, using the constraint-based strategy one can also increase precision values as misclassifications are taken out of the evaluation, however at the same time previously correct classifications are excluded resulting in a declining recall. Because of the dramatic loss of recall, this simple constraint-based strategy is not suited for stabilizing the chord labeling results.

Furthermore, our experiments reveal that performing the chord labeling based on a specific audio recording, the version-dependent results can vary greatly. This is shown by Table 10.2 indicating the mean F-measure, as well as the minimal and maximal F-measure achieved over all available recordings when using the basic labeling strategy (there was also a MIDI-synthesized version in each of the four groups). For example, the F-measure for one version of Bach amounts to 43.75%, corresponding to the minimal F-measure over all versions, whereas for another version the F-measure amounts to 87.5%, corresponding to the maximal F-measure over all versions. The average F-measure over the five versions amounts to 70%. These strong variations of the chord labeling results across different versions can not be explained by tuning effects, as we compensated for possible tuning deviations in the feature extraction step. A manual inspection showed that, for most cases, musical ambiguities are responsible for strong differences between the version-dependent results.

10.4 Conclusions

In this chapter, we showed that consistently labeled passages across several versions often correspond to correctly labeled passages. As a consequence, the cross-version visualization may serve musicologists as a helpful tool for exploring harmonic structures of a musical work. This will be demonstrated in our case study on Beethoven's Appassionata (Chapter 11), where we use the visualization as a source of inspiration for a detailed harmonic analysis.

Because of their high reliability, cross-version chord labels may be an alternative to manually generated ground truth labels. This may particularly hold for large-scale harmonic analyses on the basis of huge corpora of recorded music. In Chapter 12, we will apply our cross-version framework on the entire corpus of Beethoven's piano sonatas in order to explore harmonic structures across different movements for some of the sonatas. Here, our automated methods help to investigate which tonal centers occur in a specific sonata and how they are functionally related to each other. In this context, a structure-oriented analysis, which analyzes tonal centers according to the different form parts of the classical sonata form, is of great musicological meaning as each such part is characterized by a specific occurrence of certain harmonies. Performing this analysis across the complete corpus of Beethoven's piano sonatas, we will quantify and better understand from a music-historical perspective how Beethoven has applied tonal centers in his work.

As for future work, we need to perform more detailed quantitative evaluations to verify our hypothesis that our cross-version approach indeed leads to a stabilization of the chord labeling results. Furthermore, we plan to use our automated framework for exploring harmonic structures across even larger and more complex corpora of musical works, such as the corpus of Wagner's operas.

Chapter 11

Exploring Harmonic Structures: Case Study on Beethoven's Appassionata

In this chapter, we present a case study on Beethoven's Sonata Op. 57, the so-called "Appassionata" in order to demonstrate how the cross-version visualization introduced in Chapter 8 may serve musicologists as a supportive tool for exploring harmonic structures. Performing a detailed harmonic analysis of the Appassionata, it turns out that consistencies in the labeling results across different versions typically correspond to harmonically stable passages, thus being of musical relevance. Revealing the harmonically stable passages in an intuitive and non-technical way on the musically meaningful time axis given in bars, the cross-version visualization leads the user to passages dominated by a certain key, also referred to as tonal centers. On the other hand, the visualization reveals harmonically instable passages that typically contain the classification errors.

In the following, we first discuss the cross-version visualization for the beginning of the Appassionata (Section 11.1). Afterwards, we give a short musical description of the Appassionata explaining the choice of this work for our case study (Section 11.2). Inspired by the cross-version visualization, we then present an in-depth harmonic analysis of the Appassionata discussing the different form parts of sonata form separately (Section 11.3), before commenting on consistencies and inconsistencies (Section 11.4). Thereafter, we exemplarily show the importance of adjusting the model assumptions to the considered application scenario (Section 11.5) and investigate to which extent the cross-version visualization may support a subjective analysis of the overall form by a musicologist (Section 11.6). Finally, we conclude in Section 11.7.

11.1 Cross-Version Visualization

For the following case study we use the cross-version chord labeling approach described in Section 8.1. For convenience, Figure 11.1 presents the employed cross-version procedure again in a schematic overview, now for the beginning of Beethoven's Appassionata (bb. 1-

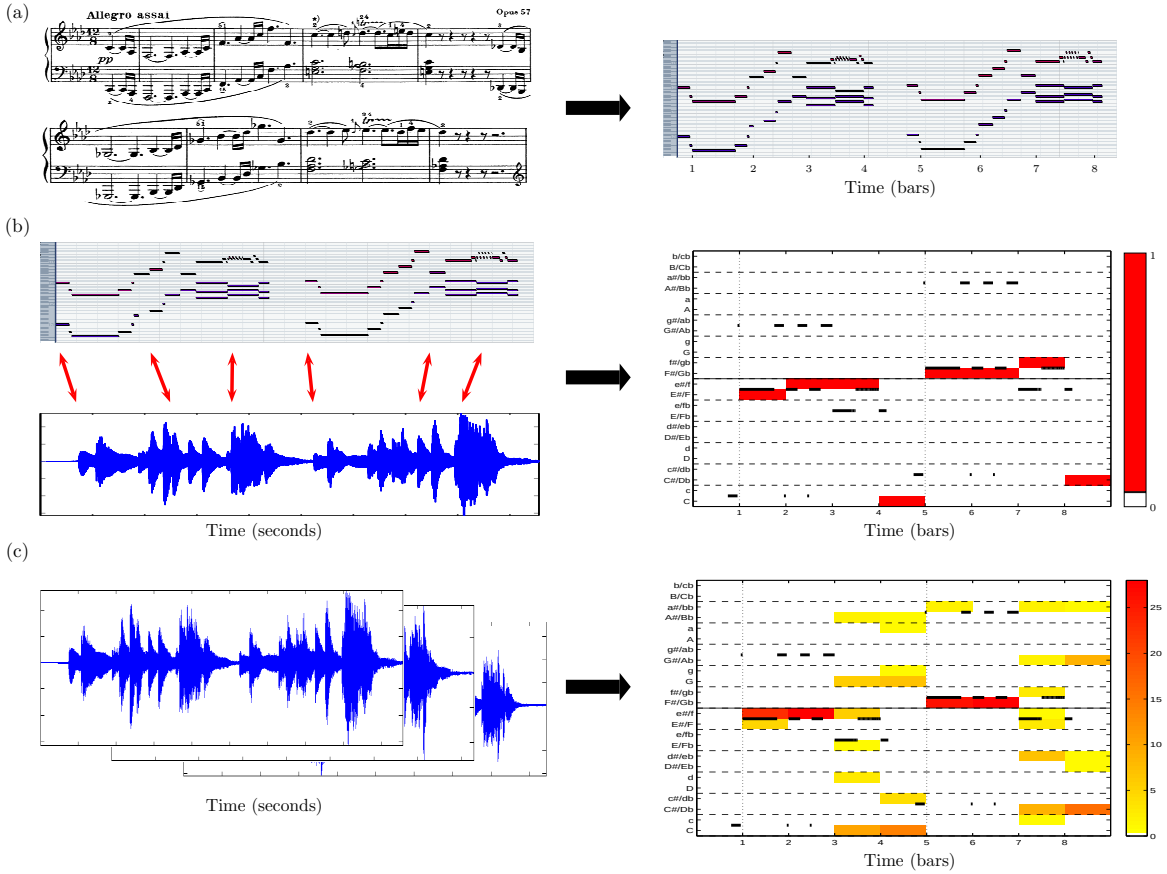

Figure 11.1. Schematic overview of the employed cross-version framework. Here, the beginning of the first movement of the Appassionata (bb. 1-8) is used as an example. (a) Export of the score to a neutral MIDI representation. (b) visualization of the automatically derived chord labels for a specific audio recording. The time axis in bars is obtained by synchronizing the audio recording with the MIDI representation. (c) Cross-version visualization (28 different audio recordings).

8).

The automatically extracted chord labels for a specific audio recording are visualized in Figure 11.1b. Following the time axis in bars, the visualization allows for a direct comparison of the automatically derived chord labels with the score. For example, one notices that F major is detected in bar 1. As the score shows, this is clearly a misclassification, since this bar contains a broken F minor chord. Actually, F major and F minor only differ in the third (a vs. $\mathtt{a}^{\flat}$), which is played very softly in the specific recording, resulting in this misclassification. In bar 2, where the broken F minor chord is continued, F minor then is correctly classified. The automatically derived chord label for bar 3 corresponds again to F minor. Referring to the score, one verifies that first a clear C major chord is present here, followed by a diminished seventh chord. Considering only 24 chord categories in the labelling procedure, the detection of this more complex chord is not really meaningful. This explains the misclassification F minor in this bar. In contrast, bars 5–6 are correctly labelled as $\mathtt{G}^{\flat}$ major. Here, again a clear arpeggio of $\mathtt{G}^{\flat}$ major is present. However, in bar 7 the next misclassification appears due to the diminished seventh chord, similarly to bar 3. Bar 8, musically corresponding to a clear $\mathtt{D}^{\flat}$ major chord, is again correctly classified.

Figure 11.1c now shows the cross-version visualization for the first eight bars of the Appassionata, where 28 different recorded performances are considered. The degree of consistency across the various performances is indicated by a color-scale ranging from bright yellow to dark red. Here, red entries indicate consistently labelled passages, whereas yellow entries point to inconsistencies.

For example, the visualization reveals two passages, where the labelling is of high consistency: bars 1–2 and bars 5–6, which are consistently labelled as F minor and G♭ major, respectively. Looking at the score, one notices that these two passages correspond to an arpeggio of the detected chords. In fact, these two passages reflect the only two harmonically stable passages within the first eight bars. However, considering only a single audio recording, bar 1 was misclassified as F major, see Figure 11.1b. Furthermore, the visualization reveals passages where the labelling is of low consistency. For example, bar 3 and bar 7 are labelled inconsistently, indicating harmonically instable passages. In fact, this is where the above mentioned diminished seventh chords appear, which are not part of the 24 considered chord categories and so mainly responsible for the inconsistent labelling. Two more or less consistently labelled passages are represented by bar 4 and bar 8. For example, bar 4 is labelled consistently for 14 performances as C major. Comparing to the score, one notices that this bar starts with a clear C major chord. However, it includes the upbeat to bar 5, introducing the G♭ major chord, which causes misclassifications in some of the recorded performances. Bar 8 is consistently labelled as D♭ major across 16 performances. Surprisingly, for ten performances this bar is misclassified as A♭ major.

11.2 Musical Work

To demonstrate the possibilities, problems and future perspectives of our method we present an analysis of the first movement of Beethoven's "Appassionata" sonata [84]. (This common title is used even if it does not stem from Beethoven but from the editor of a four-hand arrangement, in 1838.) The choice of this piece is based on several aspects. Firstly, Beethoven's piano sonatas, especially those of the middle period, present harmonic properties which are neither too complicated nor too simple. Secondly, these sonatas—and above all the Appassionata—are available in a great number of different recorded performances. Finally, for the assessment of musicological purposes of our method, it was important for our choice that the Appassionata is an extremely interesting work in its formal properties. As our research focuses not only on automated harmonic analysis, but also on the possibilities presented by our automated analysis for the comprehension, interpretation and demonstration of form aspects in large-scale works, an analysis of the Appassionata might serve as a paradigm for future research.

The sonata movement is clearly divided into four quite similar sections (see Table 1): exposition (bb. 1–65), development (bb. 65–135), recapitulation (bb. 135–204) and coda (bb. 204–262). In contrast to almost all other sonatas and symphonies of this period, the exposition is not repeated. Consequently, the bars really indicate the extension of the piece in time, whereas in more "conventional" sonata movements the exposition has to be counted twice.

Form part	bb.	Subparts	bb.	Musical description
Exposition (A)	1–65	A1	1–24	first group
		A2	24–35	transition
		A3	35–51	second group
		A4	51–65	cadential group
Development (B)	65–135	B1	65–93	first group
		B2	93–109	transition
		B3	109–123	second group
		B4	123–135	cadential group
Recapitulation (C)	135–204	C1	135–163	first group
		C2	163–174	transition
		C3	174–190	second group
		C4	190–204	cadential group
Coda (D)	204–262	D1	204–210	first group
		D2	210–218	second group
		D3	218–239	transition
		D4	239–262	second group

Table 11.1. Structural overview of the first movement of the Appassionata. The table shows for each musical form part (A, B, C, D) the corresponding structural subparts.

11.3 Harmonic Analysis

Before entering a discussion of the labelling results as shown in Figures 11.2–11.6, it has to be stressed that our main goal is not to maximise the "correctness" of the identified chord labels. Instead, we want to demonstrate that our cross-version visualization is a useful tool for the harmonic analysis of a piece of music. By performing an in-depth harmonic analysis of the Appassionata, we investigate to which extent the harmonic structures provided by our visualization correspond to the analysis results obtained by "traditional" musicologists. Figure 11.2 shows the visualization of our automated analysis as an overview. The four form parts A, B, C and D as well as the subparts are marked in the visualization by vertical black lines. In the following sections, the four form parts are discussed separately, starting with the exposition.

11.3.1 Exposition

Figure 11.3 shows the cross-version visualization for the exposition of the sonata. The visualization provides information in multiple ways. Concerning the overall harmonic structure of the exposition, it becomes evident that the exposition is divided into four subparts, which, besides other differences not considered here, are different in their harmonic structure. These four subparts correspond to the conventional subparts of sonata expositions: first group (often called "first theme", A1), transition (A2), second group (or "second theme", A3) and cadential group leading in minor keys usually to the related major key (A4).

A1 refers to the first group. Here, several red passages can be seen indicating that they are consistently labelled across all recorded performances. This might be an indicator for harmonically stable bars. So, for example, the first two bars present a triad arpeggiatura in `F` minor (bb. 1–2), which is despite its agitated rhythm a harmonically stable passage.

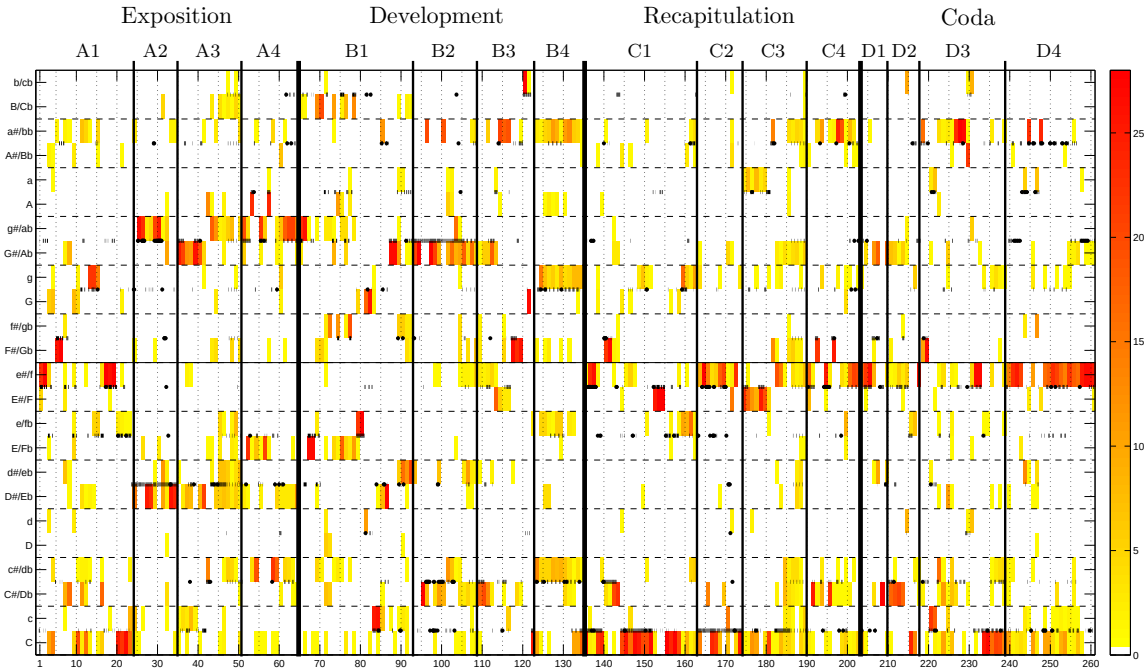

Figure 11.2. Cross-version visualization for the first movement of Beethoven's Appassionata Sonata.

This is reflected in the visualization by the dark red entries in the F minor row, showing a high degree of consistency of the chord labels for these bars across all performances. The following bar (b. 3) seems to be harmonically less stable, being labelled inconsistently for different recorded performances. In fact, the score shows that this bar includes a C major chord in the first half followed by a diminished seventh chord in the second half. Choosing a bar-wise window in the chord recognition procedure and considering only the 24 major and minor chords in our framework, this bar is identified as harmonically not homogeneous. For a majority of recordings bar 3 is labelled as C major, for others it is labelled as D minor, E major, F minor, G major and B♭ major. The next harmonically stable passage (bb. 5–6) is consistently labelled as G♭ major. Here, the triad arpeggiatura of the first two bars is repeated a semitone higher in G♭ major.

The second part of the exposition (A2), the so-called transition, modulates to the key of the second group (A3), A♭ major. The visualization shows two tonal centres here: E♭ major, the dominant of the destination key A♭ major, and A♭ minor, the minor variant of this key. An eye-catching feature is the pedal point e♭, which is constantly present during the entire part A2.

The transition (A2) is followed by the second group (A3). Here, the visualization indicates a larger homogeneous section: The first 6 bars (bb. 35–40, presenting what is usually called "second theme") are consistently labelled as A♭ major. In contrast, bar 42 is labelled as D♭ minor for approximately half of the recordings and as B♭♭/A major for approximately the other half because of the two successive harmonies D♭ minor and B♭♭ major in this bar. The following bars 45–50 then are uniformly colored in yellow presenting a completely new pattern in the visualization. The reason for these apparent difficulties in the identifi-

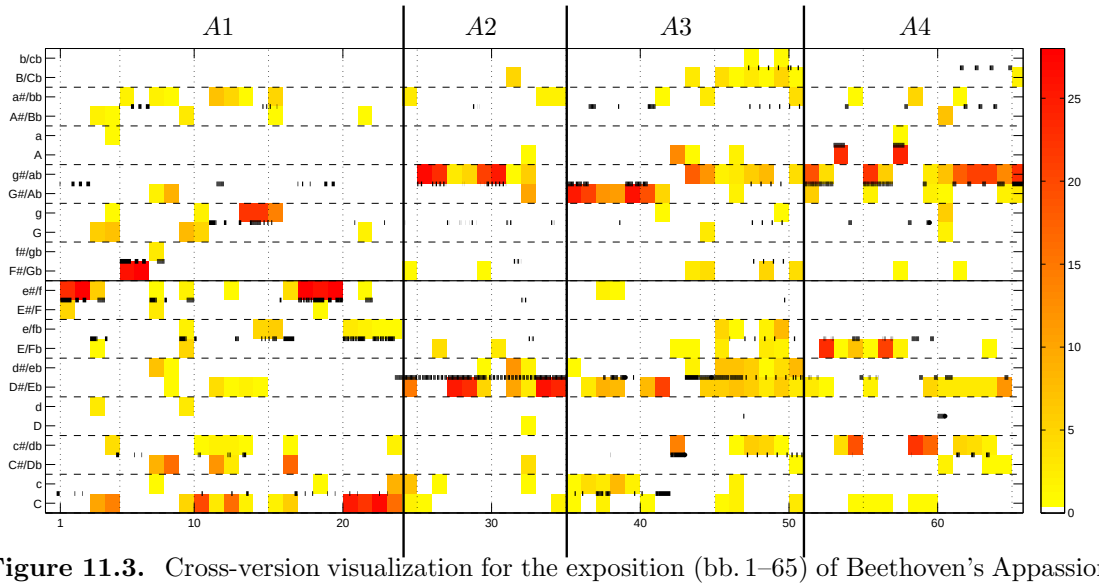

Figure 11.3. Cross-version visualization for the exposition (bb. 1–65) of Beethoven's Appassionata Sonata.

cation of a concrete harmony is that bars 45–46 present only a single note with trill and appoggiaturas and bars 47–50 a mere scale in A♭ minor with chromatic elements. This scale, leading to the cadential group (A4), is reflected in the unusually "jumping" bassline in the visualization.

The cadential group (A4) shows a clear predominance of A♭ minor, the minor variant of the relative key A♭ major. The first three bars are consistently labelled correctly as A♭ minor (b. 51), F♭ major (b. 52) and B♭♭ major (b. 53, enharmonically changed as A major). Similarly, bars 55–57 are consistently labelled correctly. Bars 54 and 58, labelled mainly as D♭ minor, present a special problem. Even for a musicologist it is not easy to immediately understand what happens here. On the fourth beat of bar 53 the B♭♭ major chord is changed into the same chord with its minor seventh (a♭♭ enharmonically changed to g natural) and minor ninth (c♭♭ enharmonically changed to b♭), omitting the fundamental b♭♭. If B♭♭ major is the dominant chord of E♭♭ major, this diminished seventh chord would have the same function being read as d♭–f♭–a♭♭–c♭♭. As the class of diminished seventh chords can be interpreted in four different directions (in this case: d♭ could be the leading tone to E♭♭/D major or minor, f♭ to G♭♭/F major or minor, a♭♭ to B♭♭♭/A♭ major or minor), Beethoven makes use of this polyvalence, interpreting this chord (sounding in the context of B♭♭ major as d♭–f♭–a♭♭–c♭♭) as d♭–f♭–g–b♭, which is the dominant of A♭ major and minor. This enharmonic change becomes evident only in the course of bars 54 and 58 on the fourth beat with the chord of E♭7 major, which on the second and third beat is obscured by the suspended fourth and minor ninth. This leads to the erroneous labelling of this chord in our procedure mainly as D♭ minor. In bar 59 E♭7 major with suspended fourth and minor ninth is repeated, which is only labelled correctly as E♭ major for some recordings, whereas a majority is labelled as D♭ minor, due to the repeated occurrence of d♭, a♭ and f♭. The last five bars (bb. 61–65) then are consistently labelled as A♭ minor.

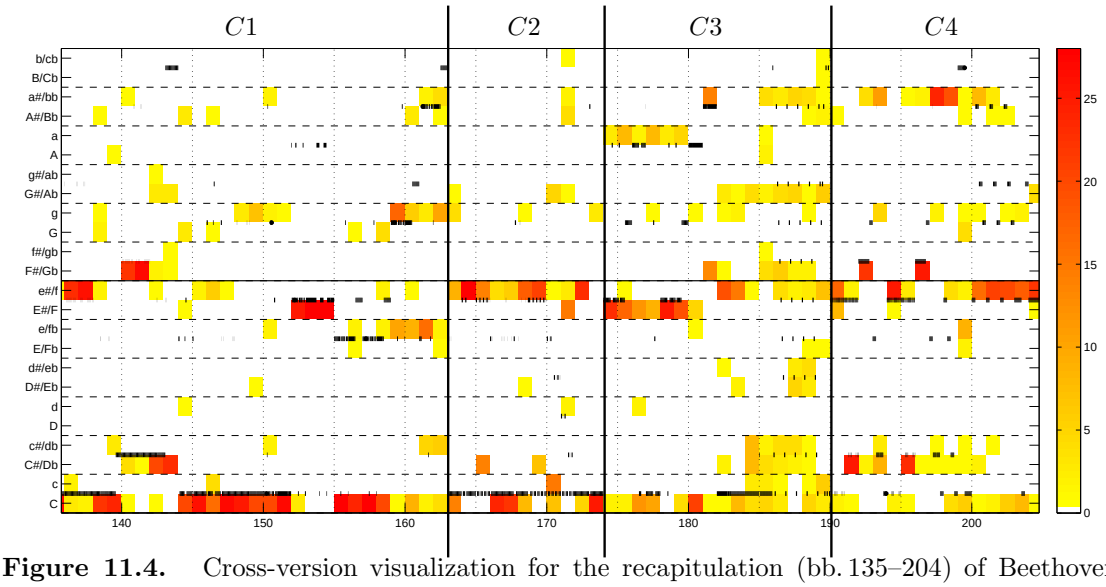

Figure 11.4. Cross-version visualization for the recapitulation (bb. 135–204) of Beethoven's Appassionata Sonata.

11.3.2 Recapitulation

As in sonata form the recapitulation (C) usually is a return of the exposition (A) with characteristic harmonic changes, it makes sense to compare these two parts in our visualization. Figure 11.4 shows the harmonic structure of the recapitulation, which—corresponding to the exposition—is divided in four parts: The first group (C1), the transition (C2), the second group (C3) and the cadential group (C4). At first glance, the harmonic structure of C1 seems to be similar to A1. Several differences, however, can be observed. So, for instance, two pedals `c` (bb. 135–139) and `d`$^\flat$ (bb. 139–142) are striking, which did not appear in A1. Furthermore, the continuation of the "first theme" (bb. 144–151) is likewise highlighted by the pedal `c` in contrast to the exposition. It can be seen that this passage is consistently labelled as `C` major, whereas the labelling for the corresponding passage in the exposition varied. Another difference becomes obvious in bars 152–154, where `F` major is present instead of `F` minor in the exposition. Here, a variant of the "first theme" consisting of ascending triads is presented in the major tonic, anticipating already the "second theme", which in the recapitulation appears in `F` major (C3, bb. 174 ff.).

C2, C3 and C4 are transposed a minor third downwards in comparison to the exposition. Hence, in C2, which leads to C3 in the major tonic `F` major, the two tonal stable keys are now the dominant `C` major and the tonic `F` minor. The second group is then presented in the major tonic `F` major (C3, bb. 174 ff.). Finally, the cadential group (C4) re-establishes the tonic `F` minor.

11.3.3 Development

The development (B) traditionally is the central section of sonata form where modulations and thematic work take place. As new keys are a characteristic property of this section, it usually shows a higher degree of tonal instability. A closer look at our visualization,

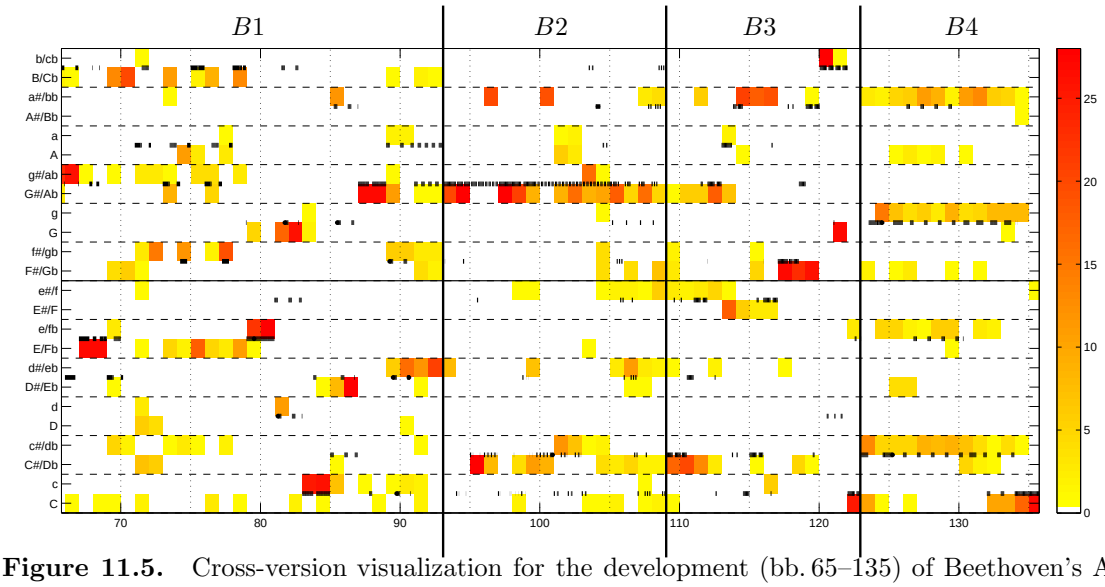

Figure 11.5. Cross-version visualization for the development (bb. 65–135) of Beethoven's Appassionata Sonata.

however, reveals that the Appassionata's development is not as different in its harmonic structure from the exposition and the recapitulation as might be expected. Again four parts can be distinguished which surprisingly resemble the four parts of the exposition: first group (B1), transition (B2), second group (B3), cadential group (B4). Certainly, especially the first part differs harmonically from the exposition, which is expressed by the red and dark yellow passages in Figure 11.5: The descending triad of the very beginning (often called the "main theme") is presented firstly in A♭ minor (the tonality in which the exposition ends), here notated as G♯ minor (bb. 65f.), then in F♭ major (bb. 67–68; notated as E major) and so on. After a series of sequenced repetitions of the "interrupting" motif from bars 3–4, the development properly starts with a series of sequences of the "main theme" which range from E minor (bb. 79–80), G7 major (bb. 81–82, shown as G major), C minor (bb. 83–84), E♭7 major (bb. 85–86, shown as E♭ major), to A♭ major (bb. 87–88), see Figure 11.5. Beethoven then—strangely enough—repeats the transition (B2; bb. 93–109) with its pedal, now on a♭, leading to the entrance of the "second theme" (B3; bb. 109–122) in Db major. This section is followed by a sort of cadential group (B4; bb. 123–134), which is almost entirely based on the diminished seventh chord e–g–b♭–d♭. Since we only consider 24 chord categories in our procedure, this diminished seventh chord leads to an inconsistent labelling in our visualization. Interestingly, the inconsistent labelling concentrates more or less on chords which have as root note one of the notes belonging to the diminished seventh chord. We will show in Section 11.5 how to account for diminished seventh chords by suitably extending the set of chord categories. This diminished seventh chord is changed to C7 major with suspended minor ninth (b. 132) before finally, C major is reached (b. 134) preparing the entrance of the recapitulation (C).

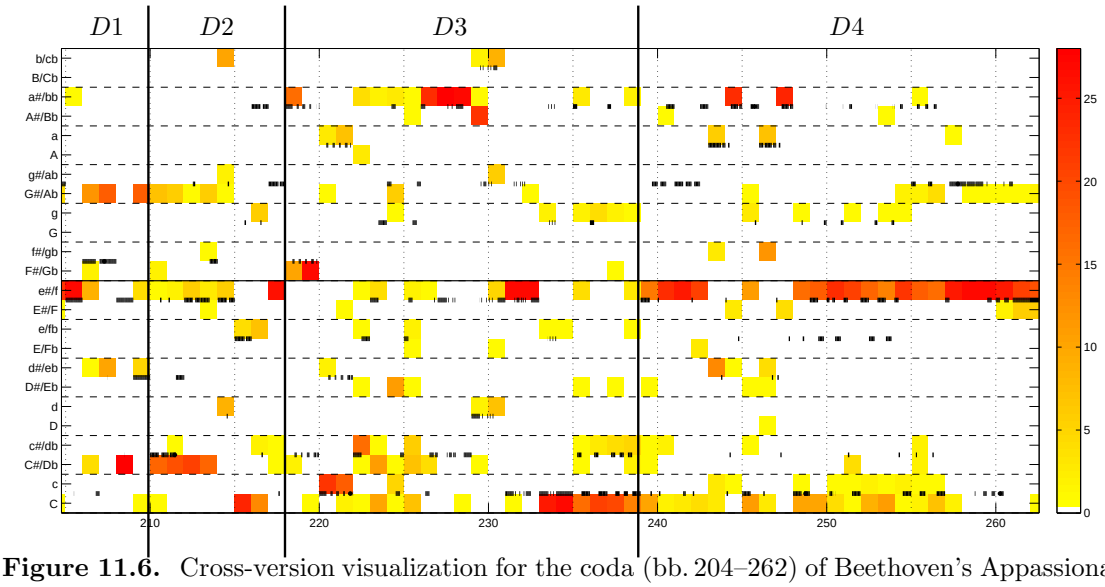

Figure 11.6. Cross-version visualization for the coda (bb. 204–262) of Beethoven's Appassionata Sonata.

11.3.4 Coda

The last large scale part of this first movement of the Appassionata is the coda, which, having a length of 58 bars, is of equal importance as the other three parts. Its tonal centre can be seen easily in Figure 11.6: the tonic `F` minor. In particular, D4 is clearly dominated by this key, emphasising the tonic at the end of the movement. Here again, the second group is present, now drawn into the sombre tonic of the movement as a whole, `F` minor.

11.4 Consistencies and Inconsistencies

The aim of our case study is not to present a presumably perfect automated analysis, but to discuss the potential and also the problems of our procedure. Not only consistencies (red), but also inconsistencies (yellow) in the visualization reveal interesting aspects of musical relevance. Apparently, the "red" bars indicate musical sections where a certain chord is clearly predominant. Even if we take possible errors of our approach into account, several aspects revealed by the visualization are surprising. For example, before the recapitulation (C), the tonic `F` minor (or major) and its dominant `C` major are present only in the first section of the exposition (A1): `F` minor in bars 1–2 and 17–19, `C` major in bars 10 and 20–23. A closer look at the score shows that this is indeed correct. Apparently all "red" bars can be related clearly to the indicated key and, vice versa, there seems to be no other bar with a clear triadic structure, which is not shown in red. The absence of `F` minor and major and its dominant `C` major seems to be a characteristic property of this sonata movement. Figure 11.2 clearly shows that even in the first group of the exposition (A1) these chords are scarcely present and that from the transition (A2) onwards they disappear completely until their return in the first group of the recapitulation (C1). In the recapitulation's second group (C3), `F` major plays a significant role, whereas `F` minor is pre-dominant in the coda, especially in its final part (D4). `F` minor appears like the

centre of gravity, circled by a multitude of other harmonies, which come closer and closer and finally fall back on this centre.

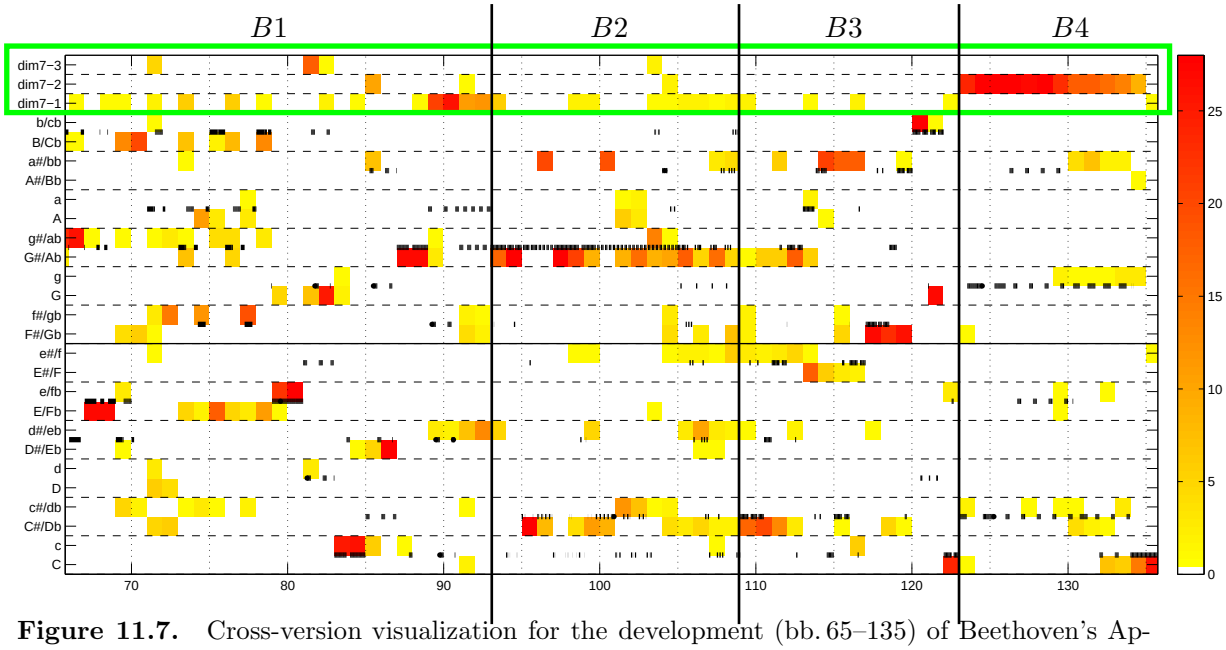

Figure 11.7. Cross-version visualization for the development (bb. 65–135) of Beethoven's Appassionata Sonata. Here, 27 different chord templates (12 major, 12 minor, 3 diminished seventh chords) are considered.

The next question is why in our visualization many bars of this sonata movement are shown in yellow (indicating inconsistency). Generally, this may be either a problem of our approach (e. g. local misclassifications) or a characteristic of the actual harmonies, or both. One reason may be a plurality of chords within the chosen frame. In this case, the change of the frame size from—in our case—one bar to a half bar or a dotted crotchet (quarter note) may solve this problem. Other possible reasons may be additional notes in the chords, such as appoggiaturas, trills or suspended notes, furthermore, incomplete chords or chords that do not fit into the set of 24 chord categories. It may be concluded that chords, which do not at all appear in our visualization, are likely to be more or less absent. Thus, our visualization reveals the surprising fact that even tonalities that might be expected in this `F` minor sonata movement are absent, for instance $\mathtt{B}^{\flat}$ major, `G` major, `G` minor, `D` major and `D` minor.

11.5 Model Assumptions

In our case study we have chosen certain model assumptions as simple as possible in order to better illustrate the behaviour of automated methods. Note that, e. g., the set of 24 chord categories or the basic template-based chord labelling procedure can be easily replaced by other more complex or even simpler model assumptions. In general, the underlying model assumptions should be adapted to the respective application, and the appropriateness of the model should be discussed with musicologists. For example, in a specific scenario the set of 24 chord categories may be inappropriate. Here, prior knowledge

can be used to either restrict or extend this set to better match a specific application. In the following, we give an example arising from our case study, which shows that the adjustment of the model assumptions to the considered application scenario is of great importance.

As our harmonic analysis in Section 11.3 has shown, the diminished seventh chords consisting of three minor thirds do not fit with any of the 24 major and minor chords. In our visualization, this necessarily leads to inconsistent patterns, thus concealing the importance of this chord in this piece. This can best be seen in the last section of the development (B4), where the diminished seventh chord e–g–b♭–d♭ dominates the piece, see Figure 11.5.

In order to also account for the diminished seventh chords, one can extend the set Λ of chord categories to comprise 27 elements: 12 major chords, 12 minor chords, and the 3 diminished seventh chords (up to enharmonic equivalence). In the following, these three chords are denoted by `dim7-1` (c–e♭–g♭–a), `dim7-2` (c♯–e–g–b♭) and `dim7-3` (d–f–a♭–b), respectively. Then, 27 binary templates $\mathbf{t}_\lambda$ are used instead of the 24 which were used before. Figure 11.7 shows the cross-version visualization for the development, where now 27 different chord categories are considered. The vertical axis again represents the chord categories, where now the 3 diminished seventh chords `dim7-1`, `dim7-2`, and `dim7-3` are arranged above the 24 major and minor chords (see the green box).

As the visualization directly reveals, the diminished seventh chord in the last section (B4, bb. 123–131) is now consistently labelled in the expected way, namely as `dim7-2`. Especially for bars 123–129 the chord labels across almost all versions agree on this chord, so that almost no yellow passages are visible any longer. Furthermore, the visualization now shows a second passage, bar 89–90, which is consistently labelled as the diminished seventh chord `dim7-1`. Comparing this to the score, one finds out that indeed a clear `dim7-1` is present in these bars. Consequently, the extension of the chord categories has led, at least in this example, to a clear identification of passages where diminished seventh chords are present. Furthermore, the consistently labelled passages, which were present when considering only 24 chord categories, are still present when considering now 27 categories. However, note that further enlarging the set of possible chord categories may also lead to a deterioration of the results. Having more categories to choose from, also increases the chance of misclassifications and hence of inconsistencies.

11.6 Aspects of Large-Scale Form

The visualization of the first movement of Beethoven's Appassionata as seen in Figure 11.2 should not be regarded as the final result of our automated harmonic analysis, but as a starting point for a deeper comprehension of this composition. Analysis, be it in fields such as chemistry, physics or in the arts, is always a reduction to more or less isolated aspects which then may be helpful for a better understanding of the whole. An additional advantage of our approach is the possibility to visualize the harmonic structure of entire movements. So, Figure 11.2 gives an overview of the first movement of the Appassionata. The movement consists of four quite similar form parts. This has been observed before, e. g. by Theodor W. Adorno, who compared the construction of this movement to four

stanzas [1]. This, however, is misleading, as a strophic form would suppose a more or less closed form of each "stanza". Figures 11.2–11.6, however, show that all form parts open with a section in which several keys are reached (A1, B1, C1, D1) and—with the significant exception of the development (B), which ends on a diminished seventh chord (dominant to the tonic F minor) in B4, see Figure 11.5—only at the end come to a clear harmonic position. Thus, all four parts seem to go through a process of establishing a key. Furthermore the visualization reveals that the "second theme" appears in the exposition in A♭ major (A3, bb. 35 ff.), in the development in D♭ major (B3, bb. 109 ff.), in the recapitulation in F major (C3, bb. 174 ff.) and finally in the coda twice: in D2 again in D♭ major as in the development, and then finally in D4 (bb. 239 ff.) "più allegro" in F minor, being drawn here from its former "far regions" into the F minor key of the entire movement. Our analysis shows that this key is clearly dominant only in this final section of the sonata movement. Here, at the end, the "second theme" is revealed as the actual "main theme" of this movement, whereas the "first theme" (or "first group") has a more introductory character. Three times this "introduction" and the subsequent transition with its pedal lead towards a sort of a visionary "second theme", before turning to a minor key (A and C) or a diminished seventh chord (B). Only at the end, in the coda, this "second theme" is drawn into the main key; it loses its visionary character, gaining tonal authenticity.

11.7 Conclusions and Perspectives

This case study on Beethoven's Appassionata should be seen as an example scenario to illustrate how automated methods and visualizations may assist musicologists in their work while serving as a source of inspiration. Contrary, by means of a concrete and detailed harmonic analysis, we indicated challenges and limitations of automated chord recognizers.

The visualization of harmonically stable and instable passages on a musically meaningful time axis (given in bars and not in seconds as is usually the case when analysing recorded performances) often reveals harmonically relevant structures, which can then be studied in greater detail based on a score representation. The presented automated approach allows for large-scale harmonic analyses on the basis of extensive recorded music corpora. In this way, our approach may efficiently support harmonic analyses of entire work cycles. At present, we are working on the entire corpus of Beethoven's piano sonatas, see Chapter 12. Here, our visualizations serve as the starting point for a comparison of harmonic structures within particular sonatas and also across different movements and pieces.

Finally, we want to emphasise two advantages of the presented procedure. Firstly, the visualization is a meaningful reduction of the score. The principle of reducing tonal music has already been established by methods using e. g. Schenkerian analysis [45]. The reading of a score requires high musical skills, and—what is even more important—the analysis of harmonic developments in a score is obfuscated by a huge amount of different information including melody, thematic processes, phrasing, dynamics, and instrumentation. The procedure presented here enables a considerably easier reading of harmonic developments and may serve as a useful tool, which, by the way of reduction and large-scale visualization, may offer new insights. Secondly, it becomes possible to perform harmonic analyses with

a speed and a degree of objectivity which is not feasible in a purely manual fashion. Traditional analysis is based on more or less "subjective" impressions of reading, performing or listening with all its multiple features—certainly adequate forms of reception. On the contrary, the results of our cross-version chord labelling approach are on the one hand clearly a reduction, but on the other hand can claim to be not an interpretation but to be based on physical facts. The intention is not the reduction of musicology to "objectivity", but to use objectively verifiable facts for the reassessment of established interpretations and the opening of new perspectives.

Chapter 12

Large-Scale Analysis of Harmonic Structures

In this chapter, we show how our cross-version approach for harmonic analysis enables for large-scale analyses of harmonic structures. Based on a cross-version voting strategy, as presented in Chapter 10, we derive statistics of the appearance of tonal centers across the entire corpus of Beethoven's piano sonatas. Performing a structure-oriented analysis of tonal centers according to the different parts of sonata form, we investigate the appearance of tonal centers in the different form parts of particular piano sonatas. Furthermore, computing and visualizing the statistics of tonal centers relatively to the key of the considered sonata, we are able to compare the statistics across different sonatas. Analyzing the average appearance of tonal centers for the sonatas of the early, middle and late period, separately, we reveal commonalities, differences and trends in the appearance of tonal centers across the entire work cycle.

This chapter is organized as follows. First, we describe the scenario of Beethoven's piano sonatas (Section 12.1). In our experiments, we use a cross-version voting strategy for deriving statistics of the appearance of tonal centers, which we analyze across the complete work cycle (Section 12.2). Finally, we conclude in Section 12.3.

12.1 Description of the Scenario

In this section, we decribe the scenario our experiments are based on. First, we give an introduction to the work cycle of Beethoven's piano sonatas (Section 12.1.1). Afterwards, we describe the dataset used in our experiments (Section 12.1.2). Finally, we introduce the Meta-MIDI annotation format, which enables for a musically meaningful evaluation across the entire work cycle (Section 12.1.3).

12.1.1 Beethoven's Piano Sonatas

The corpus of Beethoven's piano sonatas can be seen as one of the highlights in music history. Until now, numerous musicological studies have been performed which investigate several aspects of Beethoven's piano sonatas [49, 72, 84]. Furthermore, these sonatas belong to the standard repertoire of many pianists resulting in numerous recordings of the complete work cycle. The corpus comprises 32 different sonatas, which generally contain three to four different movements — there are also sonatas which only consist of two movements. From a historical point of view, the 32 sonatas are often divided into three different classes corresponding to three time phases in which they were composed. This division into three phases is based on a certain development in the compositional properties across all 32 sonatas. As the transition between the three phases is smooth, there does not exist a unique division. In the following, we assume the division into three phases as shown in Table 12.1. In the first phase (1795-1800), which corresponds to the early period of Beethoven's piano sonatas, Beethoven takes up prevalent Classical principles and develops them further. In the second phase (1801-1814) musical form principles are already questioned and an individualization of the movements starts. Finally, the third phase (1816-1822) comprising Beethoven's late sonatas is characterized by the release of prevalent rules concerning musical form, by complex compositional properties as well as high technical requirements of the performing pianist.

In our experiments, we will focus on the first movements of Beethoven's piano sonatas due to the following reasons. The first movements of Classical sonatas are from the musicological point of view of particular interest. Furthermore, they typically follow sonata form which means that they can be subdivided into certain form parts: exposition, development, recapitulation and coda. These form parts again follow certain musical principles concerning structural aspects as well as harmonic aspects. In the following, we shortly summarize harmonic principles of Classical sonata form which are fundamental for the subsequent evaluation. In this context, we have to distinguish between sonatas in major and sonatas in minor. We first refer to sonatas in major. Here, in the exposition the first theme is represented in the tonic, whereas the second theme is represented in the dominant. Afterwards, in the development typically distant keys are reached which were not present before. In the recapitulation the first and the second theme are then presented both in the tonic which resolves the previous harmonic conflict between the two themes. As a consequence, the tonic is typically the prevalent key in the recapitulation. If a coda exists, the tonic is usually stabilized again resulting in a predominance of tonic and dominant. For sonatas in minor the main difference is that in the exposition the second theme is usually presented in the tonic parallel and not in the dominant as in the major case. Consequently, the main keys appearing in the exposition of a minor sonata are the tonic and the tonic parallel.

12.1.2 Dataset

In our experiments, we perform the cross-version chord labeling for the first movements of the entire corpus of Beethoven's piano sonatas. Table 12.1 shows an overview of all 32 sonatas, where for each sonata the number of the sonata, the opus number, the respec-

No.	Piece	Name	# bars	Length in seconds							Remarks	
				1	2	3	4	5	6	7		
1	**Op002No1**	-	304	329	226	209	378	210	337	260	i)	–
2	**Op002No2**	-	668	410	439	651	435	378	416	444	i)	–
3	**Op002No3**	-	347	603	623	619	619	586	617	639		+
4	**Op007**	-	497	525	477	510	520	445	498	425		+
5	**Op010No1**	-	388	351	353	329	348	291	323	316		+
6	**Op010No2**	-	404	374	366	517	323	296	342	302	i)	–
7	**Op010No3**	-	468	413	428	428	414	384	473	415		+
8	**Op013**	-	431	513	572	526	580	494	508	602	ii)	+
9	**Op014No1**	-	222	631	408	381	445	374	365	408		+
10	**Op014No2**	-	263	400	431	466	476	414	394	420		+
11	**Op022**	-	267	419	447	470	444	408	465	426		+
12	**Op026**	-	219	434	526	435	519	408	436	414	iv)	+
13	**Op027No1**	-	106	332	335	302	276	302	321	292	iii), iv)	+
14	**Op027No2**	Moonlight	69	369	395	304	362	384	315	359	iv)	+
15	**Op028**	-	622	637	675	540	649	538	546	517		+
16	**Op031No1**	-	436	382	394	424	368	355	388	372		+
17	**Op031No2**	-	320	512	526	499	535	425	540	461	ii)	+
18	**Op031No3**	-	341	516	524	547	533	471	482	510		+
19	**Op049No1**	-	143	251	285	317	-	296	221	214		+
20	**Op049No2**	-	174	277	284	288	290	256	267	273		+
21	**Op053**	Waldstein	387	680	692	619	649	564	627	579		+
22	**Op054**	-	154	350	357	315	370	312	346	307	iv)	+
23	**Op057**	Appassionata	262	578	639	637	-	455	575	449		+
24	**Op078**	-	206	435	413	440	443	377	388	406	iii)	+
25	**Op079**	-	372	281	278	310	285	267	253	253		+
26	**Op081a**	Les Adieux	308	448	470	423	441	359	424	452	ii), iii)	+
27	**Op090**	-	245	346	373	379	348	271	322	248		+
28	**Op101**	-	102	267	269	215	253	220	247	211		+
29	**Op106**	Hammerklavier	530	660	784	694	702	568	679	578		+
30	**Op109**	-	101	254	259	224	234	198	210	201	ii), iii)	+
31	**Op110**	-	116	400	453	366	410	363	378	345		+
32	**Op111**	-	209	604	601	543	542	516	547	547	ii)	+

Table 12.1. Overview of the dataset comprising the first movements of the 32 piano sonatas by Beethoven. The table shows for each movement the title of the work, the number of bars, the lengths of the seven considered versions (1: Ashkenazy, 2: Barenboim, 3: Bilson, 4: Brendel, 5: Gulda, 6: Jando, and 7:MIDI) and some important remarks (i) Structural differences between the versions, ii) MIDI version contains part-wise tempo changes, iii) Time signature changes, iv) Not in sonata form). The last column indicates if the movement is used in the experiments (+) or not (–). The borders of the three compositional phases are indicated by double horizontal lines.

tive name, and the number of bars is indicated. For each sonata movement we consider seven different versions (six recorded performances and one MIDI representation). The table shows for each of the versions the respective length in seconds. Furthermore, certain

characteristics of the respective sonata movement are indicated in the form of remarks. First, for three of the sonatas the available versions structurally differ from each other in the sense that certain form parts are repeated only in some of the versions. Therefore, these three movements are not considered in the following evaluation. Second, some of the sonata movements contain subsequent parts which extremely vary concerning the underlying tempo. Our experiments showed that these sudden tempo changes lead to inaccuracies in the underlying synchronization procedure. Therefore, we adapted the tempo of the respective MIDI version so that it contains sudden tempo changes but follows a constant tempo for the respective parts. Third, some of the sonatas contain time signature changes which is on the one hand of musical interest, on the other hand makes the extraction of bar borders from the MIDI version problematic since the length of a bar is not constant. Fourth, four of the sonata movements in the second phase do not follow sonata form.

12.1.3 Meta-MIDI Annotation Format

In our cross-version approach, we transform the time axis of audio-based analysis results to a musically meaningful time axis in bars. Therefore, in a first step, we have to extract bar or beat positions from the MIDI representation. However, for a given piece of music the extracted beat positions from a MIDI file often deviate from the musical beat positions since the MIDI format typically does not contain any musical information about the length of a beat. Furthermore, there is no general convention to specify upbeats in a MIDI file. Problems appear also for the extraction of bar borders in the case that a piece of music contains time signature changes. To account for these problems and to enable for a musically meaningful evaluation across the corpus of Beethoven's piano sonatas, we developed the Meta-MIDI annotation format. This format allows to store meta information about the considered piece of music as e. g. time signature changes, upbeat information, the length of a musical beat, changes in the length of a musical beat within the piece, the number of measures and structural annotations, as e. g. annotations of the different parts of sonata form. The format is designed in such a way, that it enables musicologists to conveniently annotate the considered piece of music based on the score.

12.2 Experiments

In the following experiments, we first describe how to derive statistics of tonal centers using a cross-version voting strategy (Section 12.2.1). Afterwards, we exemplarily analyze statistics of tonal centers according to the different form parts of sonata form for several piano sonatas (Section 12.2.2). Finally, we perform an analysis of the appearance of tonal centers across the three phases of the work cycle (Section 12.2.3).

12.2.1 Statistics of Tonal Centers

We now use a cross-version voting strategy as described in Section 10.1. We therefore, introduce a consistency parameter $\nu \in [0, 1]$ and consider only bars which are labeled

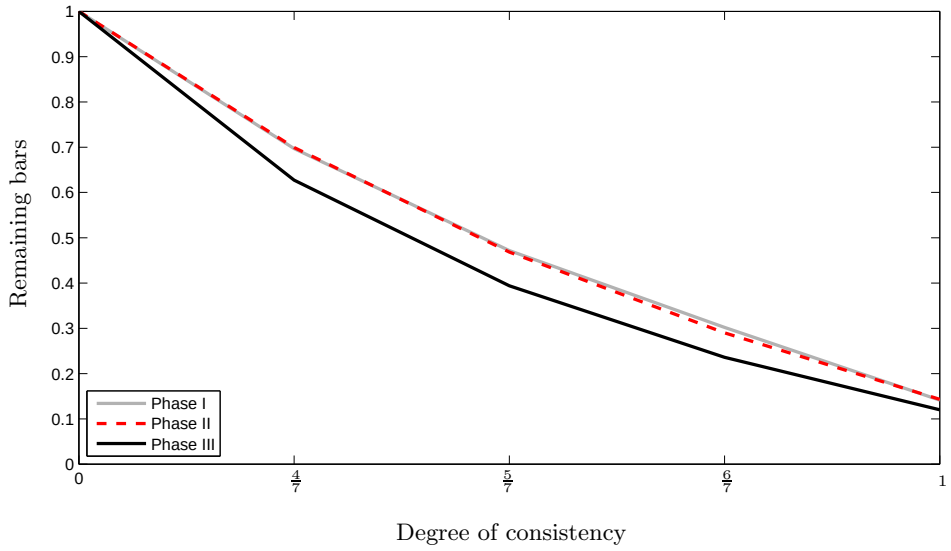

Figure 12.1. Proportion of remaining bars in dependency of the considered degree of consistency in the cross-version voting strategy. The three curves show the average proportion of remaining bars for the three phases based on different consistency parameters ν from $\frac{4}{7}$ to 1.

consistently for more than $(\nu \cdot 100)\%$ of the versions, whereas all other bars are left unannotated. As we have shown, increasing the consistency parameter ν leads to an increase of the precision values. In this context, our experiments in Section 10.3 documented that a consistency parameter $\nu = 0.5$ already led to high precision values of more than 90% for three of four considered pieces.

After having applied the cross-version voting strategy, the remaining bars tend to be harmonically stable and correspond with high probability to correctly labeled bars. Figure 12.1 shows the number of remaining, i. e. consistently labeled bars, in dependency of the consistency parameter ν. Here, the horizontal axis corresponds to the degree of consistency, whereas the vertical axis shows the number of remaining bars in proportion to the number of all bars of the piece. Each of the three curves shows for a certain phase for different consistency parameters ν the mean value of remaining bars over all sonatas contained in this phase. For example, considering only bars which are consistently labeled for at least four out of seven versions approximately 70% of the bars remain in average for the sonatas of Phase I. In other words, for the sonatas in Phase I 70% of the bars are consistently labeled across four out of seven considered versions. Figure 12.1 shows that for all three phases the number of remaining bars decreases with increasing consistency parameter ν. Considering only bars being labeled consistently across all versions, for all three phases approximately 15% of the bars are left. Furthermore, a comparison of the three curves shows that the curve for Phase I and Phase II are very similar to each other. However, the curve for Phase III strongly differs from the two other curves. As the visualization reveals the late sonatas of Phase III contain in average approximately 5% less consistently labeled passages than the sonatas in the two earlier periods. Since the sonatas of the late period are characterized by complex harmonic structures tonal centers may appear less often in Phase III than in the two other phases.

Based on the cross-version voting strategy, we now compute statistics which show the

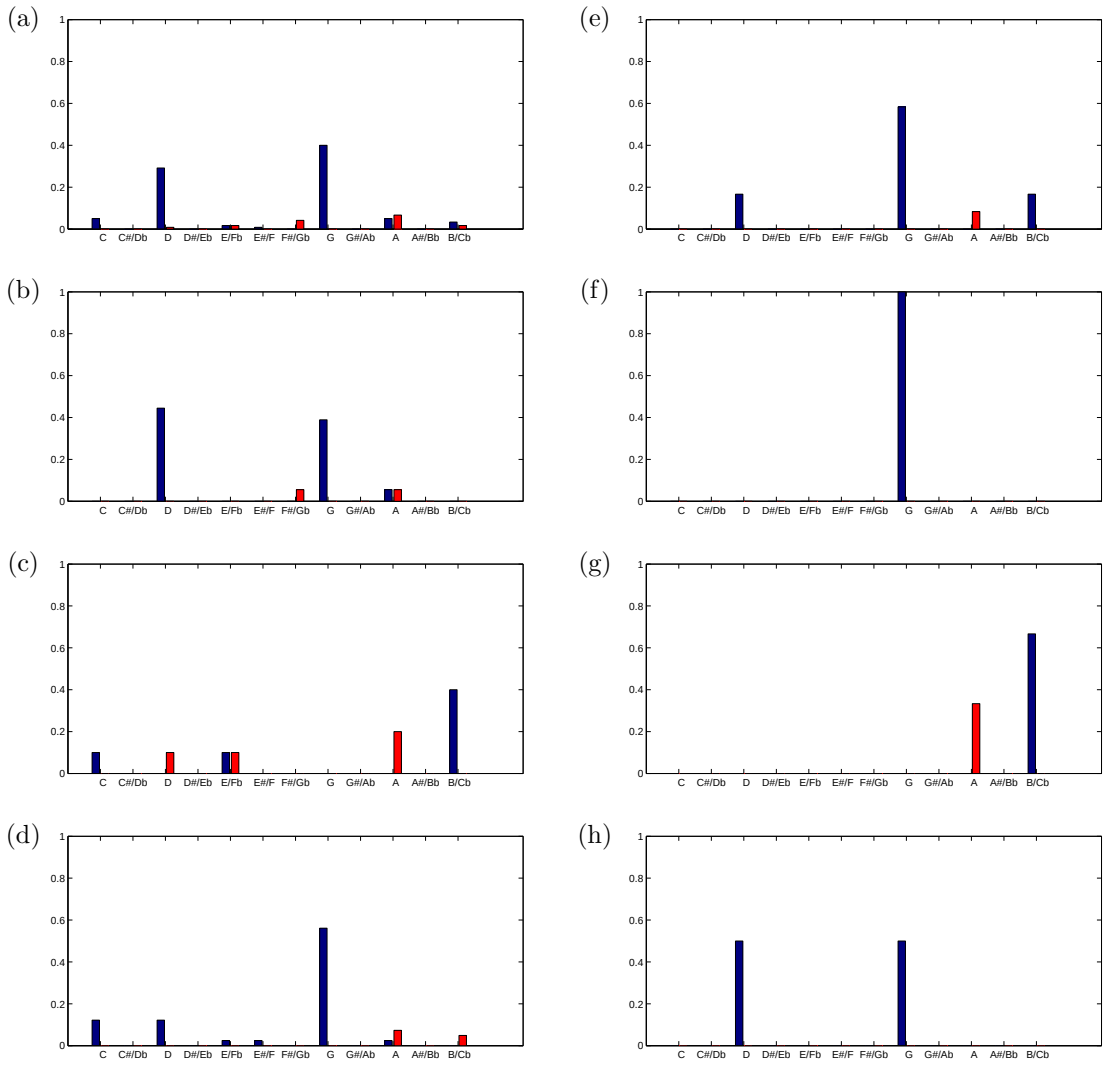

Figure 12.2. Statistics of tonal centers for **Op049No2** based on two different consistency parameters ν. The left column (a-d) shows statistics for $\nu = 0.5$, whereas in the right column (e-h) $\nu = 1$ is used. (a) Statistics for the entire movement, $\nu = 0.5$. (b) Statistics for the exposition, $\nu = 0.5$. (c) Statistics for the development, $\nu = 0.5$. (d) Statistics for the recapitulation, $\nu = 0.5$. (e) Statistics for the entire movement, $\nu = 1$. (f) Statistics for the exposition, $\nu = 1$. (g) Statistics for the development, $\nu = 1$. (h) Statistics for the recapitulation, $\nu = 1$.

distribution of the 24 major and minor chords among the consistently labeled bars for a certain consistency degree ν. In this way, we aim to measure the proportion of the 24 keys among the appearing tonal centers.

In the following, we investigate the influence of increasing the consistency parameter ν on the statistics of tonal centers. Figure 12.2 shows statistics for several form parts of **Op049No2** using two different parameter settings for ν. In the left column we used the consistency parameter $\nu = 0.5$, whereas in the right column $\nu = 1$. The vertical axis of the visualizations shows the proportion of the chords among all consistently labeled bars. The horizontal axis corresponds to the 12 root notes of the 24 chords, from C to B.

The respective major and minor chords are visualized next to each other, where blue bars correspond to major chords and red bars to minor chords.

We now compare the statistics for $\nu = 0.5$ with the statistics for $\nu = 1$. The visualizations show that considering a rather low consistency parameter $\nu = 0.5$ the distributions of tonal centers spread. For example, the visualization of the statistics for the entire movement, see Figure 12.2a, clearly reveals the tonal centers D major (29%) and G major (41%). However, there are many other keys which appear in the visualization as possible tonal centers but amount to less than 10%: C major, D minor, E major, E minor, F major, F♯ minor, A major, A minor, B major, and B minor. Using the maximal consistency parameter $\nu = 1$ a kind of stabilization of the statistics is achieved. The chords, which were only scarcely present using a lower consistency degree now disappear and only the chords which are consistently labeled across all versions remain. In this way, only four of the previously twelve appearing tonal centers are left: D major, G major, A minor and B major. The same effect can be observed for the exposition, development and recapitulation of the sonata, see Figure 12.2. This shows that computing chord statistics based on the cross-version voting strategy leads to a meaningful identification of tonal centers.

12.2.2 Examples

In the following experiments, we set $\nu = 0.7$ so that we consider only bars which are labeled consistently for more than 70% of all versions, i.e. which are labeled consistently for five out of the seven considered versions. This parameter setting seems to be a good choice since we aim on the one hand to investigate passages of high harmonic stability and on the other hand we want to keep still a sufficient number of bars. As Figure 12.1 shows, for $\nu = 0.7$ the remaining bars for Phase I and Phase II still amount in average to 50% and for Phase III to 40% of all bars.

As a first example we use again **Op049No2**. Figure 12.3 shows statistics of tonal centers for this movement based on the consistency parameter $\nu = 0.7$. In Figure 12.3a, the distribution of the 24 chords among the consistently labeled bars is visualized for the entire first movement. As the visualization directly reveals, the main tonal centers for the first movement of this sonata are G major (48%) and D major (32%). This is the distribution of keys one would expect for a typical Classical sonata in major. The tonic represented in this case by G major and the dominant represented by D major are of central harmonic importance in sonata form.

We now perform a structure-oriented analysis of tonal centers, where we consider each part of sonata form separately. Performing the analysis of tonal centers according to the different form parts is of great musicological meaning since each form part is characterized by the appearance of certain harmonies. Figure 12.3b shows the distribution of the keys for the exposition. Similarly to the distribution of the entire movement the tonic G major (47%) and the dominant D major (42%) are the prevalent tonal centers. However, one notices that the dominant D major appears in the exposition more often as tonal center reaching a value of 42% in comparison to 32% considering the entire movement. This can be explained by the fact, that the dominant as the key of the second theme is of central importance for the exposition, whereas the distribution of the entire movement also takes

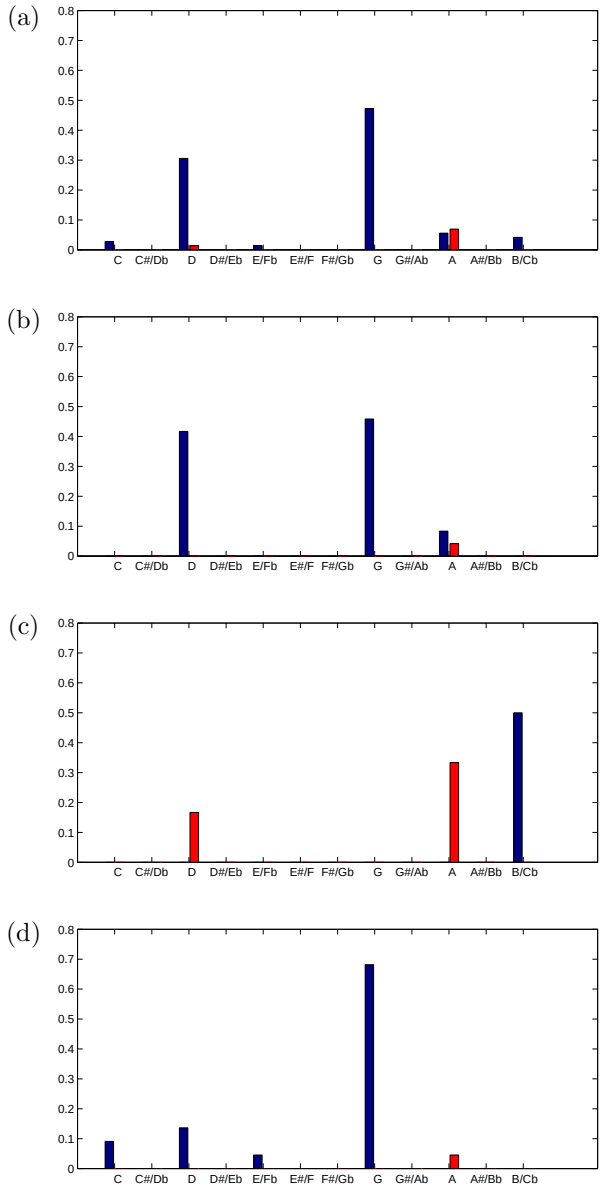

Figure 12.3. Statistics of tonal centers for **Op049No2** based on the consistency parameter $\nu = 0.7$. (a) Statistics for the entire movement. (b) Statistics for the exposition. (c) Statistics for the development. (d) Statistics for the recapitulation.

into account the appearance of the second theme in the tonic in the recapitulation. A comparison to the cross-version visualization, see Figure 12.4, shows that indeed `G` major and `D` major are the prevalent chords in the exposition.

Figure 12.3c shows the distribution for the development of the sonata. Here, `B` major (50%), `A` minor (33%) and `D` minor (17%) appear as main tonal centers. Furthermore, according to our statistics these keys were not reached before in the exposition. The cross-version visualization, see Figure 12.4, underlines the appearance of these tonal centers. In

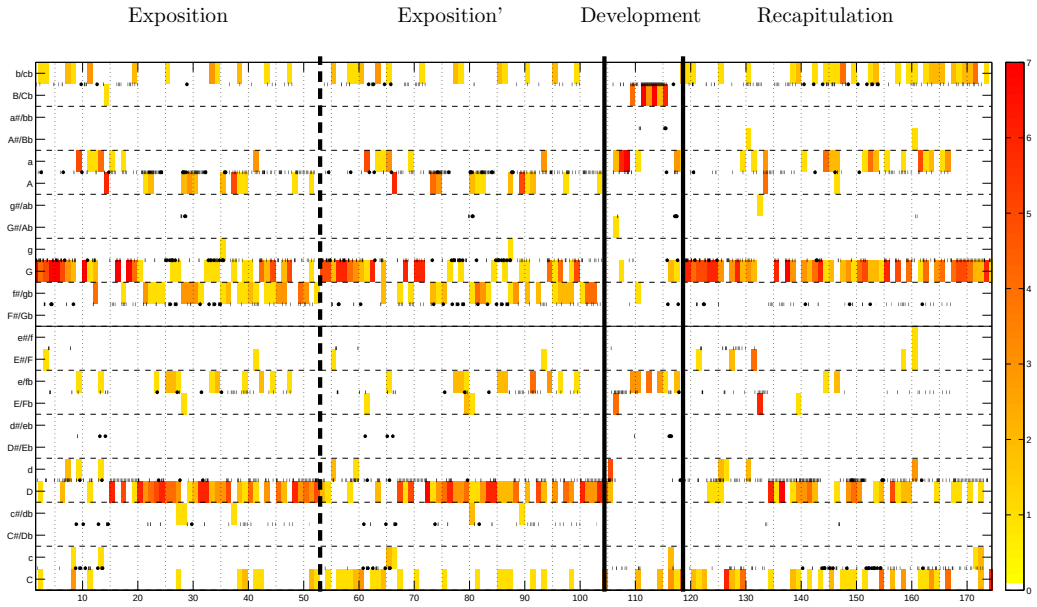

Figure 12.4. Cross-version visualization for **Op049No2**.

fact, B major seems to be the main tonal center appearing in bars 111-115.

Finally, the statistics for the recapitulation are shown in Figure 12.3d. It is obvious, that the tonic G major appears now as the main tonal center with a proportion of 70%. Here, the reason is that the first and the second theme are now both presented in the tonic, which is revealed by the cross-version visualization, see Figure 12.4.

The previously described example has turned out to be a sonata following the rules of Classical sonata form. In the following, we exemplarily show how the visualization of the statistics of tonal centers helps to identify sonatas which deviate from certain rules or more generally said, which are characterized by individual harmonic properties.

As next example we consider **Op028**. Figure 12.5 shows statistics of tonal centers appearing in the different form parts of the sonata. In the left column the previously introduced visualization is shown for the entire movement, the exposition, the development, the recapitulation and the coda. Here, the keys are denoted in an absolute way, facilitating the deeper analysis of tonal centers in the cross-version visualization. However, for a comparison of different piano sonatas in different keys, which we actually aim to, it is meaningful to compute and visualize the keys of the tonal centers in a relative way based on tonal functions. Such a visualization of the statistics is shown in the right column of Figure 12.5. Here, the horizontal axis again shows the 24 major and minor chords but now relatively to the key of the considered sonata. Hence, the axis is shifted in the sense that it starts with the root note of the tonic. The axis labels indicate on the one hand a semitone index, which corresponds to the number of semitones of the interval between the root note of the tonic and the root note of the respective chord. On the other hand, the axis labels indicate the scale degree, here, the degree of the major scale. For example I, IV, and V correspond to the tonic, subdominant and dominant, respectively.

Op028 is a sonata in D major. As the statistics of tonal centers for the exposition in

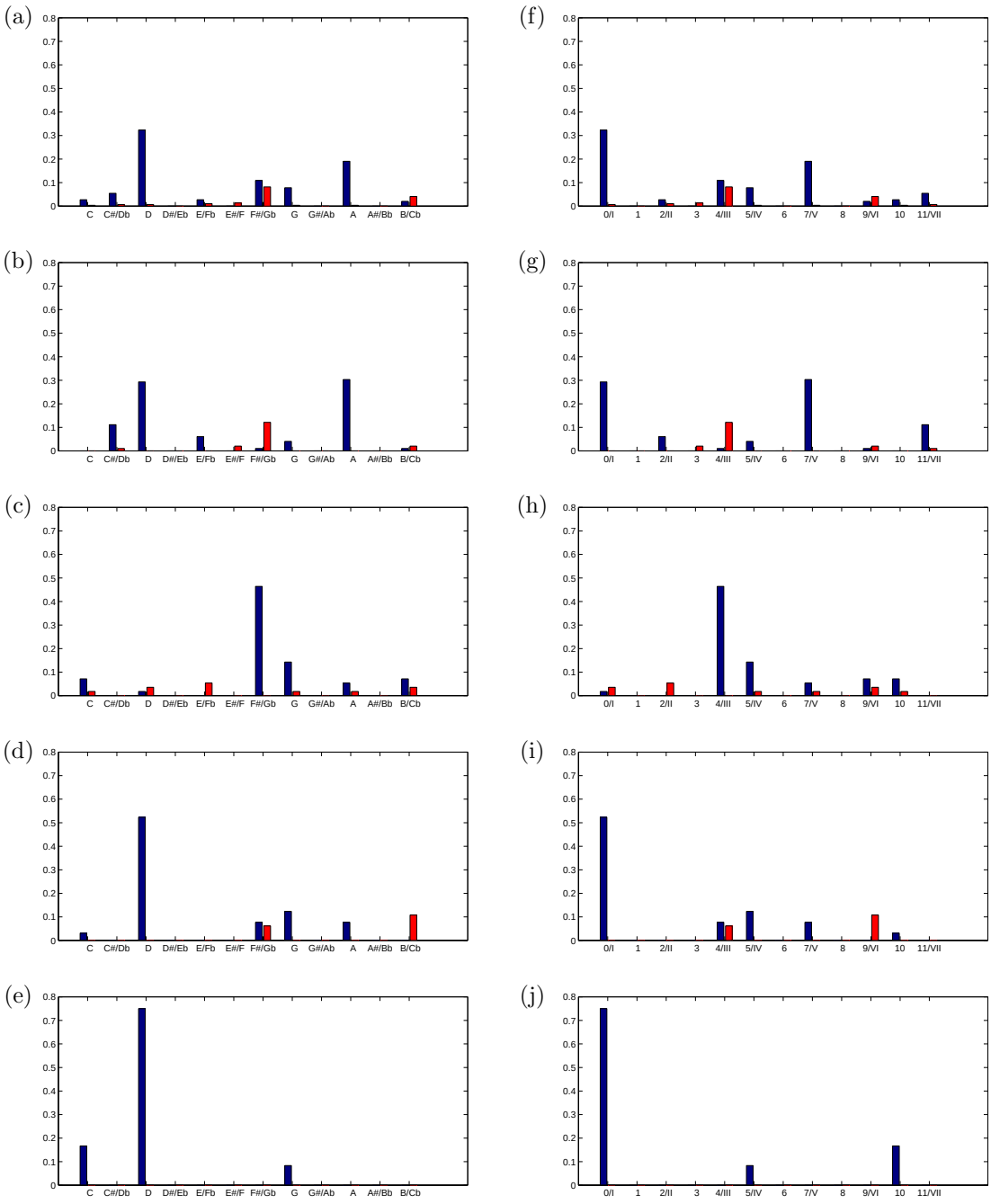

Figure 12.5. Statistics of tonal centers for **Op028**. In the left column (a-e) the chords are visualized in an absolute way, whereas in the right column (f-j) the chords are visualized relatively to the key of the considered sonata (D major). (a) Statistics for the entire movement. (b) Statistics for the exposition. (c) Statistics for the development. (d) Statistics for the recapitulation. (e) Statistics for the coda. (f) Statistics for the entire movement. (g) Statistics for the exposition. (h) Statistics for the development. (i) Statistics for the recapitulation. (j) Statistics for the coda.

Figure 12.5b and Figure 12.5g show, the main tonal centers in the exposition are the tonic D major (29%) and the dominant A major (30%). However, two further tonal centers

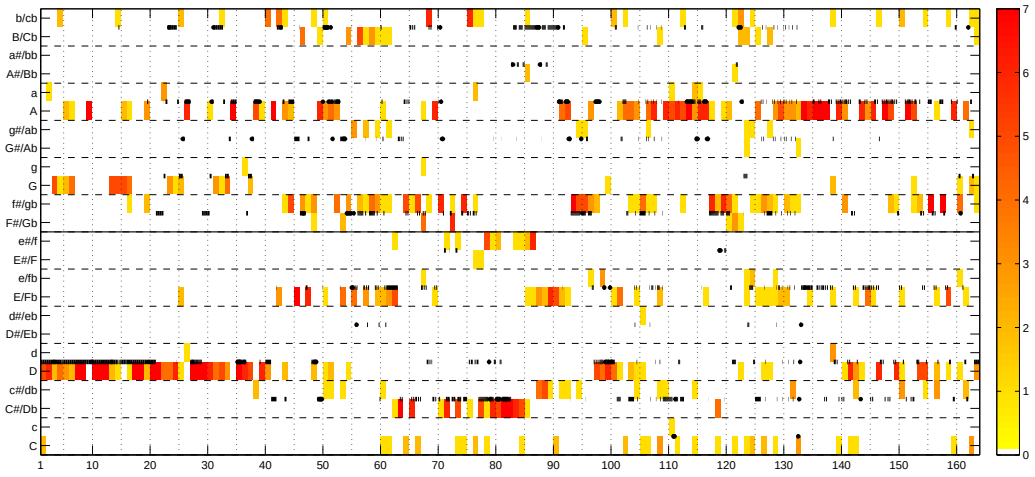

Figure 12.6. Cross-version visualization for the exposition of **Op028**.

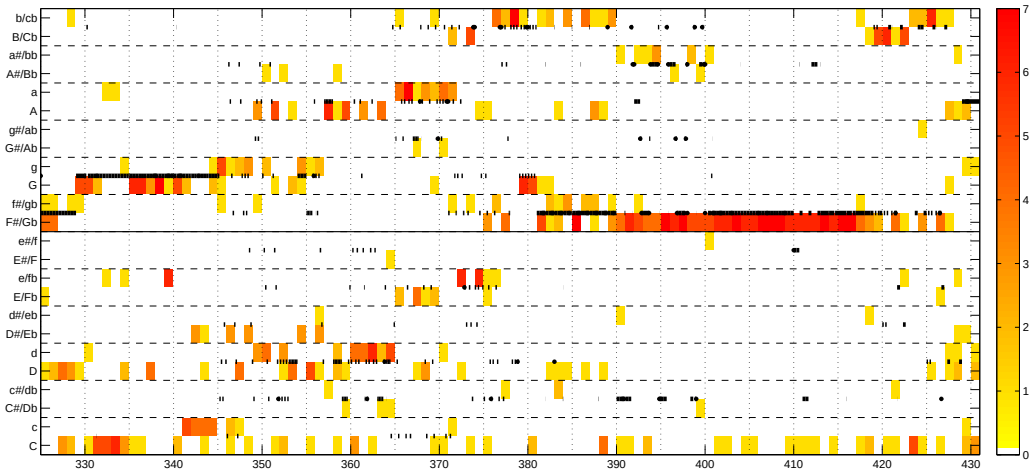

Figure 12.7. Cross-version visualization for the development of **Op028**.

appear in the visualization: the counter parallel F♯ minor (semitone index 4, degree III) amounting to 12% and its dominant C♯ major (semitone index 11), which corresponds to the major chord on degree VII of the D major scale and amounts to 11%. The appearance of these two tonal centers seems to be characteristic for this sonata. We now analyze the four appearing tonal centers in the cross-version visualization, see Figure 12.6. As the visualization reveals, the first theme (bb. 1-39) is clearly dominated by the tonic D major. The second theme starts in bar 63. From this bar on, the cross-version visualization reveals a consistent labeling for C♯ major and F♯ minor in change. A comparison with the score shows that indeed the second theme in its first appearance (bb. 63-76) is dominated by F♯ minor and its dominant C♯ major. It only contains a concluding bar in the dominant A major. In bar 77 a long cantabile enters starting clearly in C♯ major before it continues in bar 91 in the actual dominant A major. Now the main tonal centers are A major, in change with F♯ minor and D major. Finally, in bar 101 the dominant A major appears as the main tonal center until the end of the exposition.

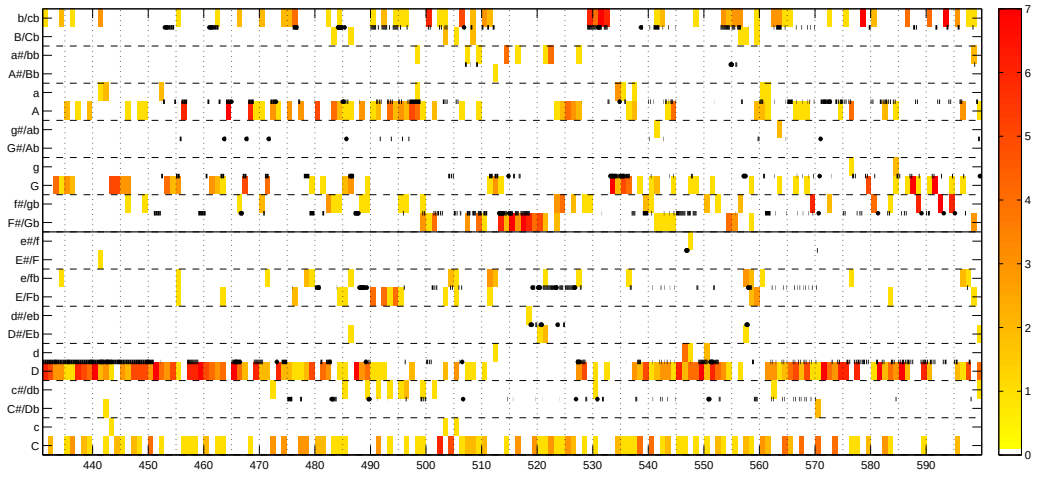

Figure 12.8. Cross-version visualization for the recapitulation of **Op028**.

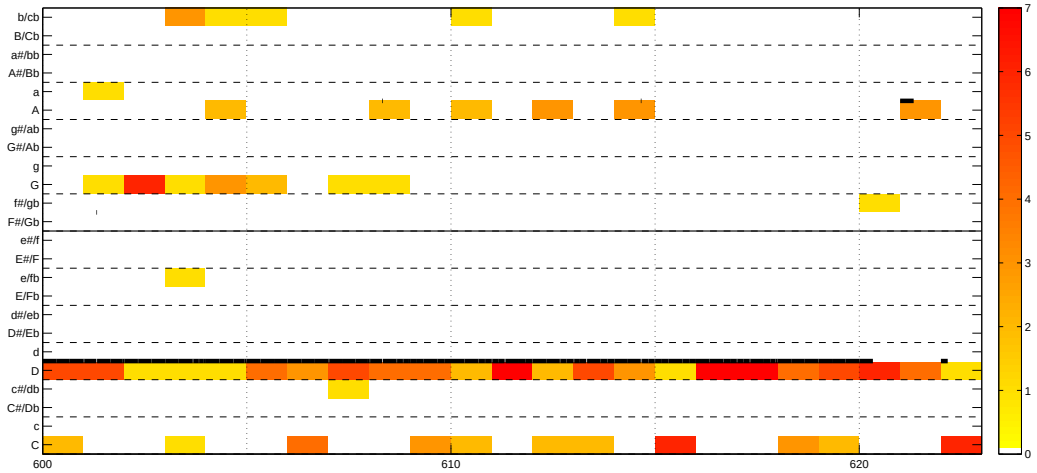

Figure 12.9. Cross-version visualization for the coda of **Op028**.

The main tonal center of the development shown by the statistics in Figure 12.5c and Figure 12.5h seems to be F♯ major (47%). A comparison with the cross-version visualization, see Figure 12.7, shows that F♯ major appears as tonal center from bars 389 to 418. In fact, the score reveals that in this passage a clear F♯ major chord is present.

The statistics for the recapitulation, see Figure 12.5d and Figure 12.5i, show that it is obviously dominated by the tonic D major (52%). A comparison with the cross-version visualization in Figure 12.8 reveals that the second theme now finally appears in the tonic from bar 545 on. However, in its first appearance (bb. 499 ff.) it is mainly colored by the keys F♯ major and B minor amounting to 7% and 11%, respectively.

The statistics for the coda, see Figure 12.5e and Figure 12.5j, reveal the tonic D major as main tonal center. 75% of the consistently labeled bars obviously correspond to D major. The cross-version visualization, see Figure 12.9, as well as the score affirm the clearly presence of D major in the entire coda showing that finally the tonic is stabilized in the

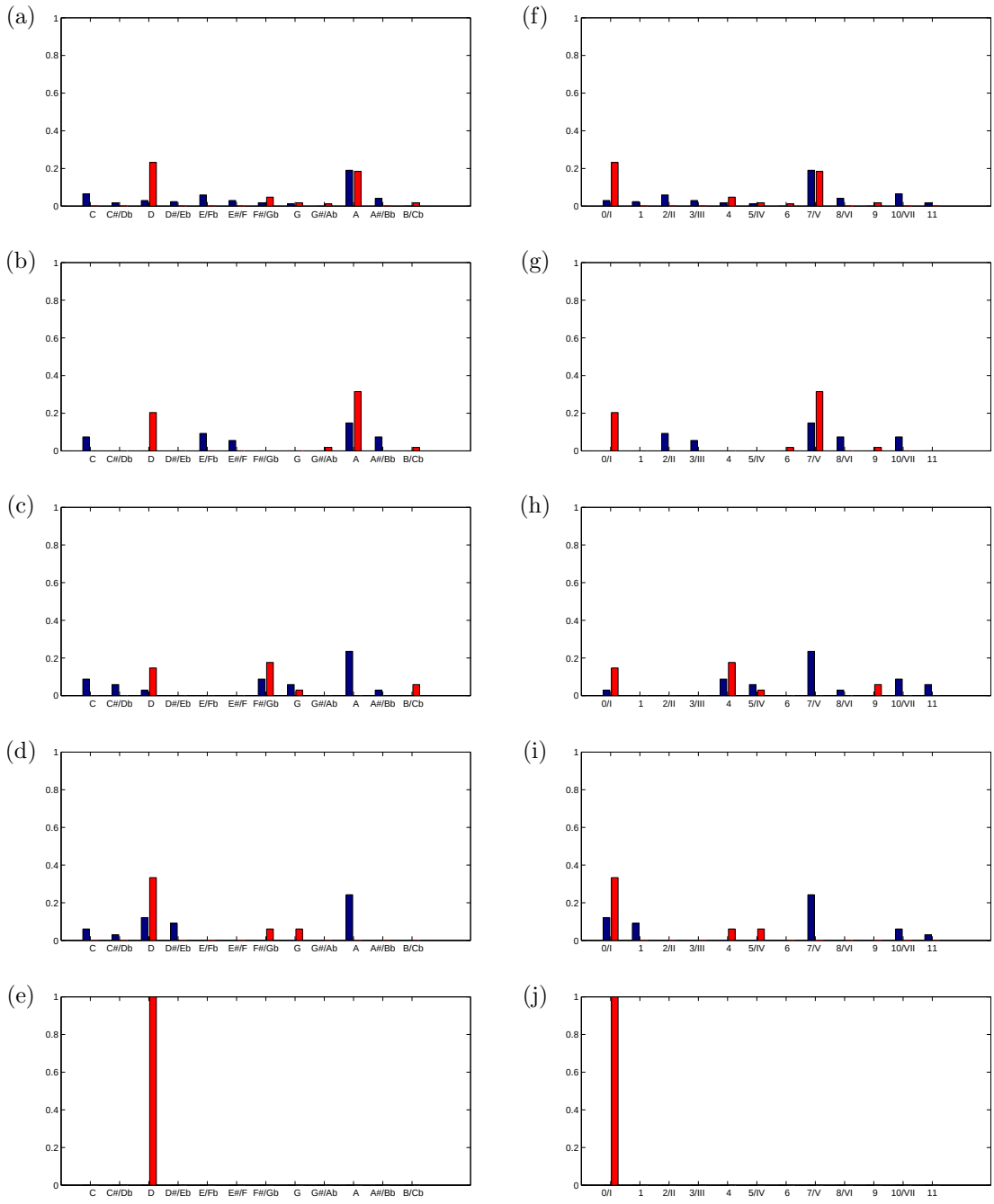

Figure 12.10. Statistics of tonal centers for **Op031No2**. In the left column (a-e) the chords are visualized in an absolute way, whereas in the right column (f-j) the chords are visualized relatively to the key of the considered sonata (`D` minor). (a) Statistics for the entire movement. (b) Statistics for the exposition. (c) Statistics for the development. (d) Statistics for the recapitulation. (e) Statistics for the coda. (f) Statistics for the entire movement. (g) Statistics for the exposition. (h) Statistics for the development. (i) Statistics for the recapitulation. (j) Statistics for the coda.

end of the movement.

As an example for a sonata in minor, we now consider **Op031No2**. Figure 12.10 presents statistics of tonal centers for this sonata. In the left column, the keys are visualized absolutely, whereas in the right column the keys are visualized relatively to the key of the sonata, D minor. Note, that the degrees indicated by the horizontal axis now refer to the degrees of the minor scale. The visualizations of the statistics for the entire movement, see Figure 12.10a and Figure 12.10f, reveal as main tonal centers the tonic D minor (24%), the dominant A major (19%) and the minor variant of the dominant, A minor (18%). The high proportion of the last mentioned chord is a little surprising. Referring to the statistics of tonal centers for the exposition, see Figure 12.10b and Figure 12.10g, we notice that obviously, the minor dominant A minor plays an important role in the exposition amounting to 31%, aside from the tonic D minor (21%). A comparison to the cross-version visualization and the score reveals that, indeed, the second theme appears in the exposition in the minor dominant A minor— for sonatas in minor the second theme is typically represented in the exposition in the tonic parallel.

In the development, the main tonal centers are the tonic D minor (17%), the dominant A major (24%) and the parallel of the dominant F♯ minor (semitone index 4, 18%), see Figure 12.10c and Figure 12.10h.

In the recapitulation the tonic now is represented as the main tonal center (35%) together with the dominant A major amounting to 23%, see Figure 12.10d and Figure 12.10i. A comparison with the score shows that the second theme is now likewise represented in the tonic.

Finally, the coda clearly reveals the tonic D minor (100%) as tonal center, see Figure 12.10e and Figure 12.10j. As the score reveals the coda stabilizes the tonic in the end of the movement, containing D minor as the clear tonal center.

12.2.3 Tonal Centers across the Three Phases

After having investigated tonal centers for particular sonatas we now aim to perform the analysis of tonal centers across the three phases. In this context, our goal is to reveal commonalities, differences and trends in the appearance of tonal centers across the entire corpus of Beethoven's piano sonatas. Furthermore, we will analyze the tonal centers for the different parts of sonata form separately, revealing trends in the appearance of tonal centers in the different form parts across the three phases.

Table 12.1 shows the division of the sonatas into the three phases. Since in the following evaluation we are interested in a structure-oriented analysis of tonal centers we keep for the three phases only the sonata movements which follow sonata form. Furthermore, as sonatas in major and sonatas in minor follow different harmonic principles, we perform a separate evaluation of the major and the minor sonatas. In this way, we evaluate in total 25 piano sonatas, 18 major sonatas, among 6, 8 and 4 belong to Phase I, Phase II, and Phase III, respectively, and 7 minor sonatas among 2, 4 and 1 belong to Phase I, Phase II, and Phase III, respectively.

Based on the computation of statistics of tonal centers relatively to the key of the considered sonata we are able to directly compare the appearance of tonal centers across the

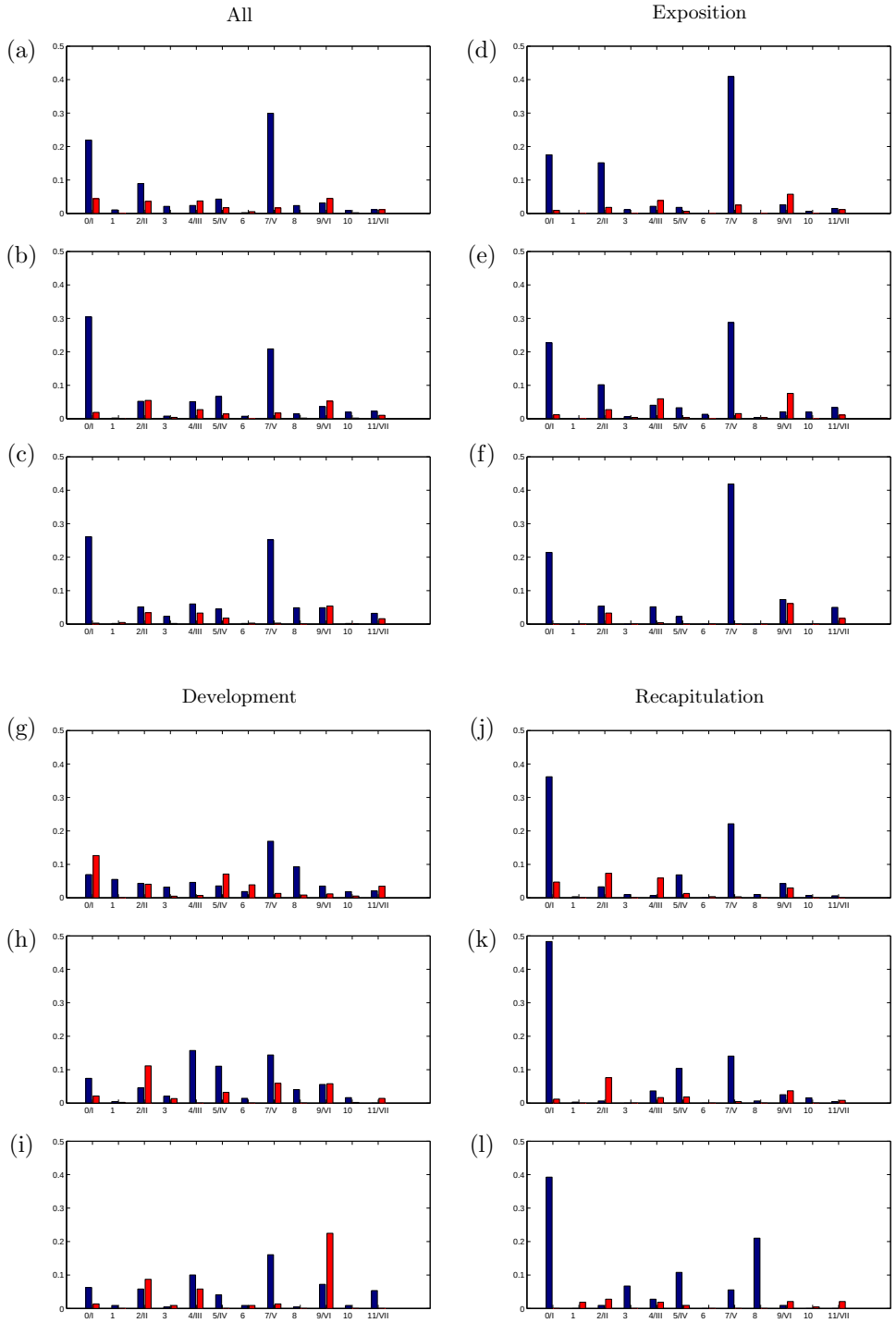

Figure 12.11. Mean distributions of tonal centers across the three phases for sonatas in major considering the entire movements (a-c), the expositions (d-f), the developments (g-i), the recapitulations (j-l). (a) Phase I (entire movements). (b) Phase II (entire movements). (c) Phase III (entire movements). (d) Phase I (expositions). (e) Phase II (expositions). (f) Phase III (expositions). (g) Phase I (developments). (h) Phase II (developments). (i) Phase III (developments). (j) Phase I (recapitulations). (k) Phase II (recapitulations). (l) Phase III (recapitulations).

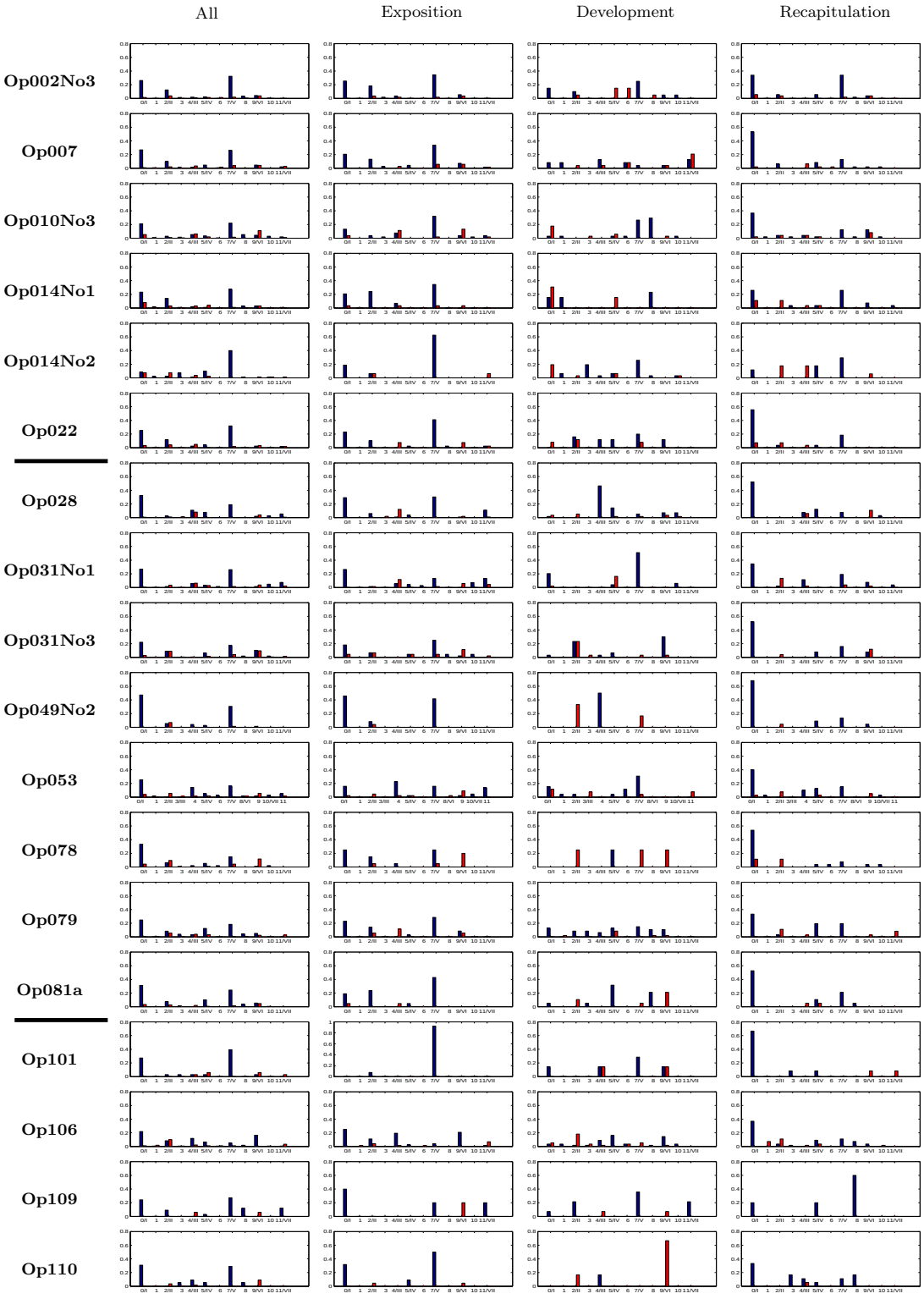

Figure 12.12. Overview of the statistics of tonal centers for all considered sonatas in major. Here, statistics are shown for the entire movement, and the different form parts exposition, development, and recapitulation. The horizontal black lines on the left indicate the borders of the phases.

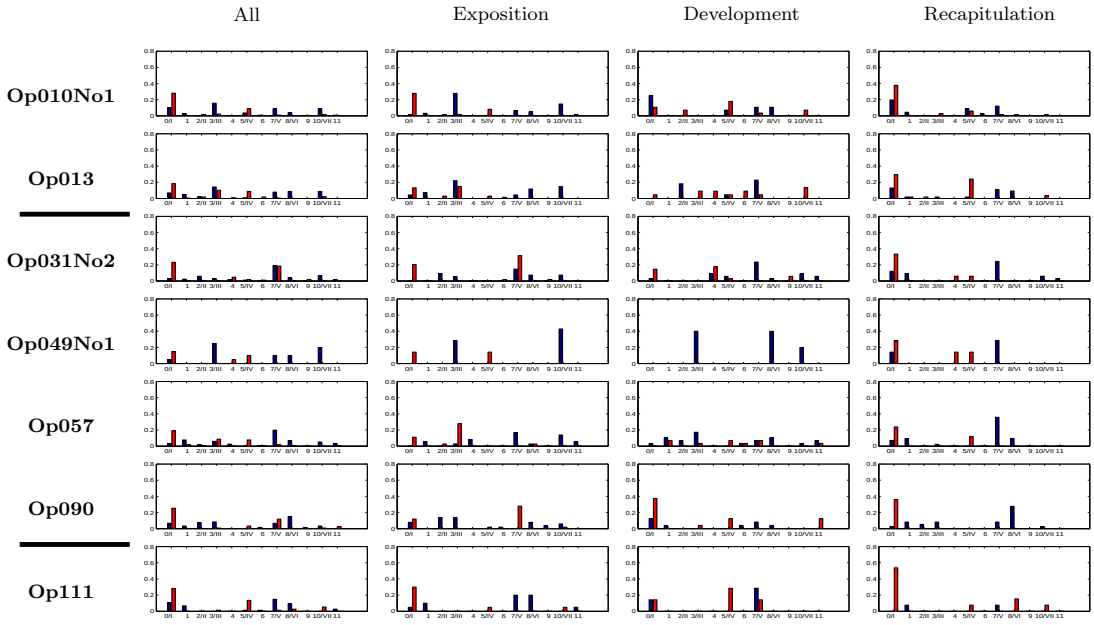

Figure 12.13. Overview of the statistics of tonal centers for all considered sonatas in minor. Here, statistics are shown for the entire movement, and the different form parts exposition, development, and recapitulation. The horizontal black lines on the left indicate the borders of the phases.

entire corpus of piano sonatas. In a first step, we compute the distribution of the 24 keys among the consistently labeled bars based on the consistency parameter $\nu = 0.7$. In a second step, we compute a mean distribution of the 24 keys for each of the three phases separately, by averaging over the distributions of the sonatas contained in the respective phase. In this way, we obtain statistics about the average appearance of the 24 keys as tonal centers for the sonatas in a particular phase.

We start our evaluation with the sonatas in major. Figure 12.11 shows mean distributions of tonal centers for the three phases and the different form parts. For a comparison to the distributions of tonal centers for the particular sonatas of the three phases, which might deviate from the mean distribution, Figure 12.12 presents the distributions of tonal centers for all movements and form parts in an overview. We now first refer to statistics concerning the entire sonata movements. Figures 12.11a, b, and c show the mean distributions of tonal centers for Phase I, Phase II, and Phase III, respectively. Comparing the statistics for the three phases shows that they do not substantially differ from each other. Across all three phases, the tonic and the dominant seem to be the clear tonal centers of the entire sonata movements.

We now perform the analysis for the different parts of sonata form separately. In Figures 12.11d, e, and f the mean distributions of tonal centers for Phase I, Phase II, and Phase III, respectively, are visualized, where only the exposition is considered. As the visualizations reveal, the tonic and the dominant, as the keys of the first and the second theme, again seem to be of central importance across all phases. However, one notices that the proportion of the secondary dominant (semitone index 2, major chord on degree II) decreases across the three phases, amounting to 14% in Phase I, 10% in Phase II and 6% in Phase III. The secondary dominant, as the dominant of the dominant, may be im-

portant in the harmonic context of the second theme. The decrease of its appearance as tonal center possibly is an indicator of a process of blurring the clear harmonic structures from the early sonatas to the late sonatas.

Figures 12.11g, h, and i show the mean distributions of tonal centers for Phase I, Phase II, and Phase III, respectively, where the development is considered. First, one notices that the distributions of tonal centers for all three phases widely spread. This is not surprising, since the development is typically characterized by the appearance of novel harmonies which were not reached in the exposition. Therefore, tonal centers of the development may vary greatly across the sonatas. However, the multitude of tonal centers decreases across the three phases so that the sonatas of Phase III seem to agree more or less on the appearance of certain tonal centers, mainly the dominant (17%) and the tonic parallel (semitone index 9, degree VI, 23%). Furthermore, one notices that the appearance of the minor tonic, corresponding to the key appearing the second often in Phase I (13%) decreases across the three phases to amount only 2% in Phase III. On the contrary, the appearance of the tonic parallel (semitone index 9, degree VI) increases across the three phases. It amounts to 2% in Phase I, 6% in Phase II and finally it seems to be of great importance for the sonatas of Phase III, amounting to 23%.

Figures 12.11j, k, and l show the mean distributions of tonal centers for Phase I, Phase II, and Phase III, respectively, where the recapitulation is considered. The visualizations mainly reveal two interesting aspects. First, the dominant seems to become less important across the three phases, decreasing from 22% in Phase I, to 14% in Phase II and 5% in Phase III. Second, the rather distant chord functionally corresponding to the parallel of the minor subdominant (semitone index 8), which amounts to only 1% and 0.05% in Phase I and Phase II, respectively, seems to be of great importance for the sonatas in Phase III, reaching a proportion of 21%.

Since the available sonatas in minor only amount to 7 sonatas among 2, 4 and 1 belong to Phase I, Phase II, and Phase III, respectively, the mean distributions for the particular phases might be statistically not relevant. However, Figure 12.13 shows the statistics of tonal centers for all minor sonatas in an overview. Here, a comparison of tonal centers across the sonatas reveals individual characteristics of certain sonatas. For example, **Op049No1** is the only sonata for which in the development only tonal centers in major keys are reached. Furthermore, in the exposition of **Op090** the minor dominant appears as main tonal center as it is the case in the previously discussed **Op031No2**. A comparison with the score shows, that indeed, in **Op090** the second theme is likewise presented in the minor dominant similar to **Op031No2**.

12.3 Conclusions

In this chapter, we showed how the cross-version approach for harmonic analysis may efficiently support musicologists in analyzing tonal centers across large music corpora. Analyzing the occurrence of tonal centers for the different form parts of sonata form across the entire work cycle of Beethoven's piano sonatas, we revealed commonalities, differences and trends in the appearance of tonal centers. In this way, we demonstrated how the presented statistics of tonal centers may offer musicologists new perspectives of

analysis, in particular across large music corpora. In the future, we plan to analyze tonal centers across even larger and more complex musical works, as e. g. the corpus of Wagner's operas. Here, due to the vast amount of data, a purely manual harmonic analysis is hardly possible. Furthermore, being characterized by complex harmonies and rich orchestrations, the detection of large-scale harmonic relations within and across the operas becomes a challenging task.

Chapter 13

Conclusion

This thesis has presented several automated methods and novel concepts which are aimed at bridging the gap between MIR and musicology. We now first reflect on general problems which appear when dealing with automated methods in this interdisciplinary context and then present our vision of a meaningful development of MIR in the future.

Automated methods do not work flawlessly—when dealing with automated methods one always has to account for possible inaccuracies in the underlying extraction step. Here, a further problem is that it is often unclear if an existing error is due to inadequacies in the underlying automated procedure or due to intrinsic musical reasons. For example, the automated extraction of tempo parameters has shown that an obvious change in the tempo curve may either point to inaccuracies in the underlying synchronization procedure or to an actual tempo change of the performer. A possibility to smooth out extraction errors is to perform large-scale analyses as applied in the context of our cross-version approach for harmonic analysis. Here, multiple versions of the same piece of music were exploited to stabilize the analysis results so that consistencies across the analysis results are indicators for correct labeling results. In the future, such a cross-version approach might also be of interest in the context of the extraction of tempo parameters from audio recordings. Here, consistencies in the tempo curves across several recorded performances may show common tendencies in the tempo shaping across all performers. By comparing individual tempo curves for specific recorded performances with the cross-version tempo analysis result, commonalities and differences in the playing style of several performers, which is the actual aim of performance analysis, could possibly be identified.

The possibility of smoothing out local inaccuracies in the analysis results is opposed to a detailed analysis, which is often of particular interest in musicology. Performing such fine-grained analyses is often not possible with current automated methods. For example, fine tempo nuances can not be revealed by the tempo curve based on the described automated procedure for extracting tempo parameters. However, these fine agogic deviations of the underlying tempo are of particular interest for analyzing a performer's individual playing style.

A further fundamental problem is that automated methods are often applied although the definition of the underlying task is not clear and, furthermore, from a musical point of

view not meaningful. For example, a beat-wise computation of tempo is not meaningful when considering musical works of the Romantic period, where the tempo shaping on a rather coarser level, i.e. measured over a larger time window, is of interest. Furthermore, extracting chords from a piece of music which consists of broken chords does not make sense when considering a time window being too small to capture all particular notes of the chord. Therefore, a musically meaningful definition of the task is of great importance for deriving significant information from MIR methods. To ensure that the employed framework is musically meaningful the definition of the task and the choice of appropriate model assumptions should be discussed in collaboration with musicologists.

Furthermore, problems appear for the evaluation of automated methods due to the lack of appropriate ground truth data. For example, for chord labeling only ground truth annotations for popular music are available, whereas from the musicological point of view harmonic structures appearing in Classical music are of particular interest. In this thesis, we therefore generated the ground truth chord annotations by ourselves. However, to enable evaluations on large audio collections of Classical music, it would be desirable that corresponding chord annotations are available. Here, our proposed method for transferring score-based ground truth annotations to audio-based annotations and vice versa enables a music expert to conveniently annotate a piece of music based on the score. This opens the way for the generation of score-based ground truth chord annotations for large audio collections in the Classical domain.

A further barrier for the development of automated methods which are applicable to musicology is the lack of communication between the two fields which fundamentally differ from each other. On the one hand, musicologists are often skeptical about the benefits of computer-based methods. Furthermore, they are not aware of novel developments in MIR and do not have a strong background in computer science which is often required when dealing with automated methods. On the other hand, computer scientists often lack in the musical background to comprehend the musical relevance of their analysis results. Furthermore, the methodologies of the two fields are fundamentally different so that novel concepts are needed which allow for transferring between them and alleviating in this way the collaboration. For example, our transformation of the physical time axis of audio-based analysis results to a musically meaningful time axis in bars is a simple yet important conceptual contribution, which only then makes the analysis results interpretable for musicologists.

To summarize, this thesis contributed to bridging the gap between computer science and musicology. We showed how in an interdisciplinary collaboration musicologists and computer scientists can greatly benefit from each other. Since automated methods allow for large-scale analyses across music corpora on an unprecedented scale, they are able to significantly support musicologists in their work. However, the role of the human is indispensable when dealing with automated methods. In this context, the major responsibilities of the human are the definition of the task, the choice of the model assumptions and the interpretation of the analysis results of the automated methods being aware of possible extraction errors.

We now indicate our vision of future interdisciplinary research in the field of MIR. For a successful collaboration between computer scientists and musicologists in the future, the

development of further fundamental concepts allowing for an exchange between the two fields is indispensable. In a collaboration based on these concepts, it is important that the task is defined by musicologists and computer scientists together. Here, musicologists ensure that the task is of musical importance, whereas computer scientists are responsible for estimating the feasibility of the proposed task with automated methods. Similarly, the choice of appropriate model assumptions for the considered application scenario has to be discussed in collaboration. In this way, musicologists and computer scientist may greatly benefit from each other. On the one hand, computer scientists may benefit from music experts by incorporating them in the evaluation process of automated methods. In this context, the present thesis showed the importance of interdisciplinary user interfaces which should be designed in a user-friendly way. Using such interfaces for the evaluation, musicologists do not need to know any details about the underlying automated procedures and can employ their musical knowledge and trained ear for conveniently performing an in-depth error analysis of the employed automated methods. This evaluation by a music expert may greatly support computer scientists in improving the underlying automated method. On the other hand, musicologists can be considerably supported by automated methods allowing for efficient analyses across large music corpora. Here, again appropriate user interfaces or visualizations are meaningful for allowing musicologists a convenient interpretation of the automatically derived results.

In summary, the field of MIR offers great possibilities for performing interdisciplinary collaborations in the future, where computer scientists and musicologists may considerably benefit from each other. Due to the fundamental difference of the two fields and a lack of communication it remains a challenge to encourage the exchange between computer science and musicology. However, in this thesis, we contributed to bridging this gap and furthermore, we showed the novel possibilities and the importance of performing interdisciplinary research.

Bibliography

[1] Theodor W. Adorno. *The Philosophy of Music*. Stanford University Press, 1998.

[2] Mark A. Bartsch and Gregory H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on Multimedia*, 7(1):96–104, February 2005.

[3] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, 2005.

[4] Juan Pablo Bello and Jeremy Pickens. A robust mid-level representation for harmonic content in music signals. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 304–311, London, UK, 2005.

[5] Andrew Brown. *Computers in Music Education*. Routledge, 2007.

[6] Taemin Cho, Ron J. Weiss, and Juan Pablo Bello. Exploring common variations in state of the art chord recognition systems. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 1–8, Barcelona, Spain, 2010.

[7] Chordino and NNLS Chroma. `http://www.isophonics.org/nnls-chroma`, Retrieved 28.03.2012.

[8] Ching-Hua Chuan and Elaine Chew. Audio key finding: Considerations in system design and case studies on Chopin's 24 Preludes. *EURASIP Journal on Advances in Signal Processing*, 2007:1–15, 2007.

[9] David Damm, Christian Fremerey, Frank Kurth, Meinard Müller, and Michael Clausen. Multimodal presentation and browsing of music. In *Proceedings of the 10th International Conference on Multimodal Interfaces (ICMI)*, pages 205–208, Chania, Crete, Greece, October 2008.

[10] Simon Dixon. Evaluation of the audio beat tracking system beatroot. *Journal of New Music Research*, 36:39–50, 2007.

[11] Simon Dixon and Gerhard Widmer. MATCH: A music alignment tool chest. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, GB, 2005.

[12] Daniel P. W. Ellis and Graham. E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, Honolulu, Hawaii, USA, April 2007.

[13] Sebastian Ewert, Meinard Müller, and Peter Grosche. High resolution audio synchronization using chroma onset features. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1869–1872, Taipei, Taiwan, 2009.

[14] Sebastian Ewert, Meinard Müller, Verena Konz, Daniel Müllensiefen, and Geraint Wiggins. Towards cross-version harmonic analysis of music. *IEEE Transactions on Multimedia*, 2012, to appear.

[15] Sebastian Ewert, Meinard Müller, Verena Konz, Daniel Müllensiefen, and Geraint Wiggins. Towards cross-version harmonic analysis of music. *IEEE Transactions on Multimedia*, 2012, to appear.

[16] Christian Fremerey, Frank Kurth, Meinard Müller, and Michael Clausen. A demonstration of the SyncPlayer system. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 131–132, Vienna, Austria, September 2007.

[17] Takuya Fujishima. Realtime chord recognition of musical sound: A system using common lisp music. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 464–467, Beijing, 1999.

[18] Emilia Gómez. Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing*, 18(3):294–304, 2006.

[19] Masataka Goto. A chorus-section detecting method for musical audio signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 437–440, Hong Kong, China, 2003.

[20] Masataka Goto. SmartMusicKIOSK: Music listening station with chorus-search function. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 31–40, 2003.

[21] Masataka Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5):1783–1794, 2006.

[22] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC music database: Popular, classical and jazz music databases. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Paris, France, 2002.

[23] Masataka Goto, Kazuyoshi Yoshii, Hiromasa Fujihara, Matthias Mauch, and Tomoyasu Nakano. Songle: A web service for active music listening improved by user contributions. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 311–316, Miami, USA, 2011.

[24] Peter Grosche, Meinard Müller, and Craig Stuart Sapp. What makes beat tracking difficult? A case study on Chopin Mazurkas. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 649–654, Utrecht, The Netherlands, 2010.

[25] Christian Guetl and Richard Parncutt. An interactive tool for training and testing musical auditory skills. In *Proc. ED-MEDIA*, pages 5229–5237, 2008.

[26] Christopher Harte and Mark Sandler. Automatic chord identification using a quantised chromagram. In *Proceedings of the 118th AES Convention*, Barcelona, Spain, 2005.

[27] Christopher Harte, Mark Sandler, Samer Abdallah, and Emilia Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 66–71, London, GB, 2005.

[28] Henkjan Honing. From time to time: The representation of timing and tempo. *Computer Music Journal*, 25(3):50–61, 2001.

[29] Ning Hu, Roger B. Dannenberg, and George Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, US, October 2003.

[30] Nanzhu Jiang, Peter Grosche, Verena Konz, and Meinard Müller. Analyzing chroma feature types for automated chord recognition. In *Proceedings of the Audio Engineering Society Conference (AES)*, Ilmenau, Germany, 2011.

[31] Zsuzsanna Kiraly. Solfeggio 1: a vertical ear training instruction assisted by the computer. *International Journal of Music Education*, 40(1):41–58, 2003.

[32] Anssi P. Klapuri, Antti J. Eronen, and Jaakko Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1):342–355, 2006.

[33] Verena Konz and Meinard Müller. Introducing the Interpretation Switcher interface to music education. In *Proceedings of the 2nd International Conference on Computer Supported Education (CSEDU)*, pages 135–140, Valencia, Spain, 2010.

[34] Verena Konz and Meinard Müller. A cross-version approach for harmonic analysis of music recordings. *Multimodal Music Processing (Dagstuhl Seminar 11041), Dagstuhl Follow-Ups*, 3:53–71, 2012.

[35] Verena Konz, Meinard Müller, and Sebastian Ewert. Ein Baseline-Experiment zur Klassifizierung von Problemen bei der Akkorderkennung. In *Proceedings of the 36th Deutsche Jahrestagung für Akustik (DAGA)*, Berlin, Germany, 2010.

[36] Verena Konz, Meinard Müller, and Sebastian Ewert. A multi-perspective evaluation framework for chord recognition. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 9–14, Utrecht, The Netherlands, 2010.

[37] Verena Konz, Meinard Müller, and Andi Scharfstein. Extracting expressive tempo curves from music recordings. In *Proceedings of the 35th International Conference on Acoustics (NAG/DAGA)*, Rotterdam, The Netherlands, 2009.

[38] Frank Kurth and Meinard Müller. Efficient index-based audio matching. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):382–395, February 2008.

[39] Jörg Langner and Werner Goebl. Visualizing expressive performance in tempo-loudness space. *Computer Music Journal*, 27(4):69–83, 2003.

[40] Kyogu Lee and Malcolm Slaney. A unified system for chord transcription and key extraction using hidden Markov models. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Vienna, AT, 2007.

[41] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1983.

[42] Micheline Lesaffre, Marc Leman, Bernard De Baets, Hans De Meyer, Liesbeth De Voogdt, and Jean-Pierre Martens. How potential users of music search and retrieval systems describe the semantic quality of music. *Journal of the American Society For Information Science and Technology*, 59(5):1–13, 2008.

[43] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, Plymouth, Massachusetts, 2000.

[44] Robert Macrae and Simon Dixon. Guitar tab mining, analysis and ranking. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 453–458, Miami, USA, 2011.

[45] Alan Marsden. Schenkerian analysis by computer: A proof of concept. *Journal of New Music Research*, 39(3):269–289, 2010.

[46] Matthias Mauch, C. Cannam, M. Davies, Simon Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler. OMRAS2 metadata project 2009. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Kobe, Japan, 2009.

[47] Matthias Mauch and Simon Dixon. Simultaneous estimation of chords and musical context from audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1280–1289, 2010.

[48] Matthias Mauch, Daniel Müllensiefen, Simon Dixon, and Geraint Wiggins. Can statistical language models be used for the analysis of harmonic progressions? In *Proceedings of the International Conference of Music Perception and Cognition (ICMPC)*, Sapporo, Japan, 2008.

[49] Siegfried Mauser. *Beethovens Klaviersonaten*. Verlag C. H. Beck, 2008.

[50] John H. Maxwell. An expert system for harmonic analysis of tonal music. In *Understanding Music with AI*, pages 335–353. MIT Press, 1992.

[51] Matt McVicar, Yizhao Ni, Raul Santos-Rodriguez, and Tijl De Bie. Using online chord databases to enhance chord recognition. *Journal of New Music Research*, 40(2):139–152, 2011.

[52] MIREX 2010. Audio Chord Estimation Subtask. `http://www.music-ir.org/mirex/wiki/2010:Audio_Chord_Estimation`, Retrieved 17.09.2010.

[53] MIREX 2011. Audio Chord Estimation Subtask. `http://www.music-ir.org/mirex/wiki/2011:Audio_Chord_Estimation`, Retrieved 02.03.2012.

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

[55] Meinard Müller, Michael Clausen, Verena Konz, Sebastian Ewert, and Christian Fremerey. A multimodal way of experiencing and exploring music. *Interdisciplinary Science Reviews (ISR)*, 35(2):138–153, 2010.

[56] Meinard Müller and Sebastian Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):649–662, 2010.

[57] Meinard Müller and Sebastian Ewert. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, page to appear, Miami, USA, 2011.

[58] Meinard Müller, Peter Grosche, and Frans Wiering. Robust segmentation and annotation of folk song recordings. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 735–740, Kobe, Japan, October 2009.

[59] Meinard Müller and Verena Konz. Automatisierte Methoden zur Unterstützung der Interpretationsforschung. In Heinz von Loesch and Stefan Weinzierl, editors, *Gemessene Interpretation*, volume 4 of *Klang und Begriff*, pages 193–204. Schott Verlag, 2011.

[60] Meinard Müller, Verena Konz, Nanzhu Jiang, and Zhe Zuo. A multi-perspective user interface for music signal analysis. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 205–211, Huddersfield, England, UK, 2011.

[61] Meinard Müller, Verena Konz, Andi Scharfstein, Sebastian Ewert, and Michael Clausen. Towards automated extraction of tempo parameters from expressive music recordings. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 69–74, Kobe, Japan, October 2009.

[62] Meinard Müller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 288–295, 2005.

[63] Mutopia Project. `http://www.mutopiaproject.org`, Retrieved 12.05.2009.

[64] Laurent Oudre, Yves Grenier, and Cédric Févotte. Template-based chord recognition: Influence of the chord types. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, Kobe, Japan, 2009.

[65] Hélène Papadopoulos and Geoffroy Peeters. Large-scale study of chord estimation algorithms based on chroma representation and HMM. In *Content-Based Multimedia Indexing (CBMI)*, pages 53–60, 2007.

[66] Hélène Papadopoulos and Geoffroy Peeters. Joint estimation of chords and downbeats from an audio signal. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):138–152, 2011.

[67] Brian Pardo and William Birmingham. The chordal analysis of tonal music. Technical report cse-tr-439-01, University of Michigan, Dept. of Electrical Engineering and Computer Science, 2001.

[68] Jouni Paulus, Meinard Müller, and Anssi Klapuri. Audio-based music structure analysis. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 625–636, Utrecht, The Netherlands, 2010.

[69] Christopher Raphael and Josh Stoddard. Functional harmonic analysis using probabilistic models. *Computer Music Journal*, 28(3):45–52, 2004.

[70] Jeremy T. Reed, Yushi Ueda, Sabato Siniscalchi, Yuki Uchiyama, Shigeki Sagayama, and Chin-Hui Lee. Minimum classification error training to improve isolated chord recognition. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 609–614, Kobe, Japan, 2009.

[71] Christophe Rhodes, David Lewis, and Daniel Müllensiefen. Bayesian model selection for harmonic labelling. In *Proceedings of the International Conference on Mathematics and Computation in Music (MCM). Revised Selected Papers (Communications in Computer and Information Science)*, pages 107–116. Springer, 2009.

[72] Charles Rosen. *Beethoven's Piano Sonatas: A Short Companion*. Yale University Press, 2002.

[73] Craig Stuart Sapp. Comparative analysis of multiple musical performances. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 497–500, Vienna, Austria, 2007.

[74] Craig Stuart Sapp. *Computational Methods for the Analysis of Musical Structure*. PhD thesis, Stanford University, USA, May 2011.

[75] Ricardo Scholz and Geber Ramalho. COCHONUT: Recognizing complex chords from MIDI guitar sequences. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 27–32, 2008.

[76] J. Serrà, E. Gómez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech and Language Processing*, 16:1138–1151, October 2008.

[77] Alexander Sheh and Daniel P. W. Ellis. Chord segmentation and recognition using EM-trained hidden Markov models. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 185–191, Baltimore, USA, 2003.

[78] Daniel Sleator and David Temperley. The Melisma Music Analyzer. `http://www.link.cs.cmu.edu/music-analysis/`, 2003.

[79] Kenneth H. Smith. The effect of computer-assisted instruction and field independence on the development of rhythm sight-reading skills of middle school instrumental students. *International Journal of Music Education*, 27(1):59–68, 2009.

[80] Sonic Visualiser. `http://www.sonicvisualiser.org/`, Retrieved 12.05.2009.

[81] Robin S. Stevens. The best of both worlds: An eclectic approach to the use of computer technology in music education. *International Journal of Music Education*, 17(1):24–36, 1991.

[82] David Temperley. *The Cognition of Basic Musical Structures*. MIT Press, 2001.

[83] David Temperley. *Music and Probability*. MIT Press, 2007.

[84] Donald Francis Tovey. *A Companion to Beethoven's Pianoforte Sonatas*. The Associated Board of the Royal Schools of Music, 1998.

[85] Robert J. Turetsky and Daniel P.W. Ellis. Ground-truth transcriptions of real music from force-aligned MIDI syntheses. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 135–141, Baltimore, USA, 2003.

[86] Yushi Ueda, Yuuki Uchiyama, Takuya Nishimoto, Nobutaka Ono, and Shigeki Sagayama. HMM-based approach for automatic chord detection using refined acoustic features. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5518–5521, Dallas, USA, 2010.

[87] Gerhard Widmer. Machine discoveries: A few simple, robust local expression principles. *Journal of New Music Research*, 31(1):37–50, 2003.

[88] Gerhard Widmer, Simon Dixon, Werner Goebl, Elias Pampalk, and Asmir Tobudic. In search of the Horowitz factor. *AI Magazine*, 24(3):111–130, 2003.

[89] Terry Winograd. Linguistics and the computer analysis of tonal harmony. *Journal of Music Theory*, 12:2–49, 1968.

[90] Eberhard Zwicker and Hugo Fastl. *Psychoacoustics, facts and models*. Springer Verlag, New York, NY, US, 1990.