Signal Processing Methods for Beat Tracking, Music Segmentation, and Audio Retrieval

Peter M. Grosche

Max-Planck-Institut für Informatik
Saarbrücken, Germany

Dissertation zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultät I
der Universität des Saarlandes, 2012

Dekan / Dean:

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

Datum des Kolloquiums / Date of Defense:

09. November 2012 / November 9th, 2012

Mitglieder des Prüfungsausschusses / Members of the Examination Board

Prof. Dr. Christian Theobalt, MPI Informatik, Saarbrücken, Chair

Prof. Dr. Meinard Müller, International Audio Laboratories Erlangen, Reviewer

Prof. Dr. Hans-Peter Seidel, MPI Informatik, Saarbrücken, Reviewer

Dr. Carsten Stoll, MPI Informatik, Saarbrücken, Reporter

Betreuender Hochschullehrer / Supervisor:

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

Peter Matthias Grosche
MPI Informatik
Campus E1.4
66123 Saarbrücken
Germany
`pgrosche@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, 9. November 2012

Peter M. Grosche

Acknowledgements

This work was supported by the DFG Cluster of Excellence on “Multimodal Computing and Interaction” at Saarland University and the Max-Planck-Institut Informatik in Saarbrücken.

I thank Prof. Dr. Meinard Müller for the opportunity to do challenging research in such an exciting field and Prof. Dr. Hans-Peter Seidel for providing an excellent research environment. Special thanks for support, advice, and encouragement go to all colleagues in the Multimedia Information Retrieval and Music Processing Group, all members of the Computer Graphics Department at MPII, the outstanding administrative staff of the Cluster of Excellence and AG4, and the members of the working group of Prof. Dr. Clausen, University of Bonn.

Für meine zwei Mädels.

Abstract

The goal of music information retrieval (MIR) is to develop novel strategies and techniques for organizing, exploring, accessing, and understanding music data in an efficient manner. The conversion of waveform-based audio data into semantically meaningful feature representations by the use of digital signal processing techniques is at the center of MIR and constitutes a difficult field of research because of the complexity and diversity of music signals. In this thesis, we introduce novel signal processing methods that allow for extracting musically meaningful information from audio signals. As main strategy, we exploit musical knowledge about the signals' properties to derive feature representations that show a significant degree of robustness against musical variations but still exhibit a high musical expressiveness. We apply this general strategy to three different areas of MIR: Firstly, we introduce novel techniques for extracting tempo and beat information, where we particularly consider challenging music with changing tempo and soft note onsets. Secondly, we present novel algorithms for the automated segmentation and analysis of folk song field recordings, where one has to cope with significant fluctuations in intonation and tempo as well as recording artifacts. Thirdly, we explore a cross-version approach to content-based music retrieval based on the query-by-example paradigm. In all three areas, we focus on application scenarios where strong musical variations make the extraction of musically meaningful information a challenging task.

Zusammenfassung

Ziel der automatisierten Musikverarbeitung ist die Entwicklung neuer Strategien und Techniken zur effizienten Organisation großer Musiksammlungen. Ein Schwerpunkt liegt in der Anwendung von Methoden der digitalen Signalverarbeitung zur Umwandlung von Audiosignalen in musikalisch aussagekräftige Merkmalsdarstellungen. Große Herausforderungen bei dieser Aufgabe ergeben sich aus der Komplexität und Vielschichtigkeit der Musiksignale. In dieser Arbeit werden neuartige Methoden vorgestellt, mit deren Hilfe musikalisch interpretierbare Information aus Musiksignalen extrahiert werden kann. Hierbei besteht eine grundlegende Strategie in der konsequenten Ausnutzung musikalischen Vorwissens, um Merkmalsdarstellungen abzuleiten die zum einen ein hohes Maß an Robustheit gegenüber musikalischen Variationen und zum anderen eine hohe musikalische Ausdruckskraft besitzen. Dieses Prinzip wenden wir auf drei verschieden Aufgabenstellungen an: Erstens stellen wir neuartige Ansätze zur Extraktion von Tempo- und Beat-Information aus Audiosignalen vor, die insbesondere auf anspruchsvolle Szenarien mit wechselnden Tempo und weichen Notenanfängen angewendet werden. Zweitens tragen wir mit neuartigen Algorithmen zur Segmentierung und Analyse von Feldaufnahmen von Volksliedern unter Vorliegen großer Intonationsschwankungen bei. Drittens enwickeln wir effiziente Verfahren zur inhaltsbasierten Suche in großen Datenbeständen mit dem Ziel, verschiedene Interpretationen eines Musikstückes zu detektieren. In allen betrachteten Szenarien richten wir unser Augenmerk insbesondere auf die Fälle in denen auf Grund erheblicher musikalischer Variationen die Extraktion musikalisch aussagekräftiger Informationen eine große Herausforderung darstellt.

Contents

Chapter 1

Introduction

Music plays an exceptional role in our society. The everyday lives of billions of people worldwide are notably affected by the omnipresence of music, e. g., by its widespread use in mass media, its ubiquitous presence in public places, and its essential role in entertainment or social activities such as music creation and dance. In the last decades, the way how music is produced, stored, accessed, distributed, and consumed underwent a radical change. Nowadays, large music collections containing millions of audio documents in digital form are at any moment accessible from anywhere around the world. Personal music collections easily comprise ten thousands of songs adding up to over 1000 hours of playback time. Stored on portable audio devices, personal music collections have become the daily companion of many people. Such abundance of digital music content, together with the relative ease of access, not only fosters that nowadays more music is consumed than ever before, but, in turn, also requires novel strategies and modes of access that allow users to organize and explore large music collections as well as to discover novel songs and artists in a convenient and enjoyable way. As a consequence, information technology is now deeply interwoven with almost every aspect of music consumption and production.

One main goal in the field of *music information retrieval* (MIR) is to develop tools that enrich the experience of users when interacting with music—be it for music production, music organization, music consumption, or music analysis. Intensive research has been conducted with the goal to develop automated methods for extracting musically meaningful information from music in all its different facets. As *audio* is the most natural form of music, the conversion of waveform-based audio data into semantically meaningful feature representations by the use of digital signal processing techniques is at the center of MIR. Music signal processing constitutes a difficult field of research because of the complexity and diversity of music signals. When dealing with specific audio domains such as speech or music, the understanding of acoustic, linguistic, and musical properties is of foremost importance for extracting meaningful and semantically interpretable information [125]. For example, language models play an outstanding role in speech processing and are an essential part in modern speech recognition systems. Similarly, music signals are by no means chaotic, or random. Quite contrary, music exhibits strong regularities, is highly structured, and follows certain “rules”. As a result, when analyzing music signals, one has to account for various musical dimensions such as pitch, harmony, timbre, and rhythm.

Exploiting musical knowledge and model assumptions, various mid-level representations have been proposed that robustly capture and reveal musically meaningful information concealed in the audio waveform.

One key aspect of music, however, is that the rules are not strict but leave a lot of room for artistic freedom in the realization by a performer. In the case of strong musical variations, the model assumptions are often not completely satisfied or even violated. In such cases, the extraction of musically meaningful information becomes a very challenging problem. For example, the aspects of tempo and beat are of fundamental importance for understanding and interacting with music [139]. It is the *beat*, the steady pulse that drives music forward and provides the temporal framework of a piece of music [166]. Intuitively, the beat can be described as a sequence of perceived pulses that are equally spaced in time. The beat corresponds to the pulse a human taps along when listening to music [112]. The term *tempo* then refers to the rate of the pulse. When listening to a piece of music, most humans are able to tap to the musical beat without difficulty. Exploiting knowledge about beat and tempo one can employ signal processing methods for transferring the cognitive process into an automated beat tracking system. Typically, such a system can cope with modern pop and rock music with a strong beat and steady tempo, where the model assumptions are typically satisfied. For classical music, however, the rules are less strictly followed. Musicians do not play mechanically at a fixed tempo, but form their interpretation of a music piece by constantly changing the tempo, slowing down at certain positions, or accelerating to create tension. As a consequence, extracting the beat locations from highly expressive performances of, e.g., romantic piano music is a very challenging task.

Another musical key concept is pitch. Pitch is a perceptual attribute which allows the ordering of sounds on a frequency-related scale extending from low to high [101; 103]. Exploiting the fact that most Western music is based on the equal-tempered scale, signal processing approaches allow for decomposing the signals into musically meaningful logarithmically spaced frequency bands corresponding to the pitch scale [123]. Exploiting such musical knowledge on the frequency content, one again relies on the fact that the musicians stick to the rules—an unrealistic assumption. For example, in the case of field recordings of folk songs, one typically has to deal with recordings performed by non-professional elderly singers that have significant problems with the intonation, fluctuating with their voices even over several semitones throughout a song. In that scenario, imposing strict pitch model assumptions results in the extraction of meaningless audio features and requires a careful adaption of the model assumptions to the actual musical content. The main challenge lies in incorporating robustness to musical variations without sacrificing the musical expressiveness of the feature representations.

The superordinate goal of this thesis is to introduce novel music signal processing methods that particularly address the key characteristics of music signals. Firstly, we exploit knowledge about the musical properties of the signals to derive compact and precise feature representations that reveal musically meaningful and highly expressive information. Furthermore, in this thesis, we particularly focus on the challenging cases where musical variations lead to not completely satisfied or even violated model assumptions. As main goal, we introduce compact feature representations that show a significantly increased robustness against musical variations but still exhibit a very high musical expressiveness.

1.1 Contributions

This thesis introduces various music signal processing approaches that contribute to three areas of MIR. Firstly, Part I of this thesis deals with the extraction of tempo and beat information in particular for complex music with changing tempo and soft note onsets. Secondly, Part II contributes to the segmentation and performance analysis of field recordings of folk songs that are performed by singers with serious intonation problems under poor recording conditions. Thirdly, Part III of this thesis covers content-based music retrieval following the query-by-example paradigm. In particular, we address scalability issues in a cross-version retrieval scenario where strong musical variations occur.

In Part I of the thesis, we address the aspects of tempo and beat. Because tempo and beat are of fundamental musical importance, the automated extraction of this information from music recordings is a central topic in the field of MIR. In recent years, various different algorithmic solutions for automatically extracting beat position from audio recordings have been proposed that can handle modern pop and rock music with a strong beat and steady tempo. For non-percussive music with soft note onsets, however, the extraction of beat and tempo information becomes a difficult problem. Even more challenging becomes the detection of local periodic patterns in the presence of tempo changes as typically occurring in highly expressive performances of, e.g., romantic piano music. In Chapter 2, as first contribution of Part I, we introduce a novel mid-level representation that captures musically meaningful local pulse information even for the case of music exhibiting tempo changes. Our main idea is to derive for each time position a sinusoidal kernel that best explains the local periodic nature of a previously extracted (possibly very noisy) note onset representation. Then, we employ an overlap-add technique accumulating all these kernels over time to obtain a single function that reveals the *predominant local pulse* (PLP). Our concept introduces a high degree of robustness to noise and distortions resulting from weak and blurry onsets. Furthermore, the resulting PLP curve reveals the local pulse information even in the presence of continuous tempo changes and indicates a kind of confidence in the periodicity estimation. We show how our PLP concept can be used as a flexible tool for enhancing state-of-the-art tempo estimation and beat tracking procedures. The practical relevance is demonstrated by extensive experiments based on challenging music recordings of various genres.

As it turns out, our PLP concept is capable of capturing continuous tempo changes as implied by ritardando or accelerando. However, especially in the case of expressive performances, current beat tracking approaches still have significant problems to accurately capture local tempo deviations and beat positions. In Chapter 3, as second contribution of Part I, we introduce a novel evaluation framework for detecting critical passages in a piece of music that are prone to tracking errors. Our idea is to look for consistencies in the beat tracking results over multiple performances of the same underlying piece. Our investigation does not analyze beat tracking performance for entire recordings or even collections of recordings, but provides information about critical passages within a given piece where the tracking errors occur. As another contribution, we further classify the critical passages by specifying musical properties of certain beats that frequently evoke tracking errors. Finally, considering three conceptually different beat tracking procedures, we conduct a case study on the basis of a challenging test set of five Chopin Mazurkas containing in average over 50 performances for each piece. Our experimental results not only make the limitations of state-of-the-art beat trackers explicit but also deepen the understanding of the underlying music material.

The tempo and in particular the local changes of the tempo are a key characteristic of a music performance. Instead of playing mechanically musicians speed up at some places and slow down at others in order to shape a piece of music. Furthermore, local changes of the tempo indicate boundaries of structural elements of music recordings. As indicated above, the detection of locally periodic patterns becomes a challenging problem in the case that the music recording reveals significant tempo changes. Furthermore, the existence of various pulse levels such as measure, tactus, and tatum often makes the determination of absolute tempo problematic. In Chapter 4, as third contribution of Part I, we generalize the concept of *tempograms* encoding local tempo information using two different methods for periodicity analysis. In particular, we avoid the error-prone determination of an explicit tempo value. As a result, the obtained mid-level representations are highly robust to extraction errors. As further contribution, we introduce the concept of *cyclic tempograms*. Similar to the well-known chroma features where pitches differing by octaves are identified, we identify tempi differing by a power of two to derive the cyclic tempograms. The resulting mid-level representation robustly reveals local tempo characteristics of music signals in a compact form and is invariant to changes in the pulse level. In summary, the novel concepts introduced in Part I of the thesis enhance state-of-the-art in beat tracking and tempo estimation in particular in the case of complex music with significant musical variations and give a better understanding of musical reasons for the shortcomings of current solutions.

In Part II of this thesis, we are dealing with applications of music signal processing to automatically segmenting field recordings of folk songs. Generally, a folk song is referred to as a song that is sung by the common people of a region or culture during work or social activities. As a result, folk music is closely related to the musical culture of a specific nation or region. Even though folk songs have been passed down mainly by oral tradition, most musicologists study the relation between folk songs on the basis of score-based transcriptions. Due to the complexity of audio recordings, once having the transcriptions, the original recorded tunes are often no longer considered, although they still may contain valuable information. It is the object of this part of the thesis to indicate how the original recordings can be made more easily accessible for folk song researches and listeners by bridging the gap between the symbolic and the audio domain. In Chapter 5, as first contribution of Part II, we introduce an automated approach for segmenting folk song recordings that consist of several repetitions of the same tune into its constituent stanzas. As main idea, we introduce a reference-based segmentation procedure that exploits the existence of a symbolically given transcription of an idealized stanza. Performed by elderly non-professional singers under poor recording conditions, the main challenge arises from the fact that most singers often deviate significantly from the expected pitches and have serious problems with the intonation. Even worse, their voices often fluctuate by several semitones downwards or upwards across the various stanzas of the same recording. As one main contribution, we introduce a combination of robust audio features along with various cleaning and audio matching strategies to account for such deviations and inaccuracies in the audio recordings. As it turns out, the reference-based segmentation procedure yields accurate segmentation results even in the presence of strong deviations. However, one drawback of this approach is that it crucially depends on the availability of a manually generated reference transcription.

In Chapter 6, as second contribution of Part II, we introduce a reference-free segmentation procedure, which is driven by an audio thumbnailing procedure based on self similarity matrices (SSMs). The main idea is to identify the most repetitive segment in a given recording which can then take over the role of the reference transcription in the segmentation procedure. As further contribution, for handling the strong temporal and spectral variations occurring in the field recordings, we introduce various enhancement strategies to absorb a high degree of these deviations and deformations already on the feature and SSM level. Our experiments show that the reference-free segmentation results are comparable to the ones obtained by the reference-based method.

The generated relations and structural information can then be utilized to create novel navigation and retrieval interfaces which assist folk song researchers or listeners in conveniently accessing, comparing, and analyzing the audio recordings. Furthermore, the generated segmentations can also be used to automatically locate and capture interesting performance aspects that are lost in the notated form of the song. As third contribution of Part II, in Chapter 7, various techniques are presented that allow for analyzing temporal and melodic variations within the stanzas of the recorded folk song material. It is important to note that variabilities and inconsistencies may be, to a significant extent, properties of the repertoire and not necessarily errors of the singers. To measure such deviations and variations within the acoustic audio material, we use a multimodal approach by exploiting the existence of a symbolically given transcription of an idealized stanza. Then, a novel method is proposed that allows for capturing temporal and melodic characteristics and variations of the various stanzas of a recorded song in a compact and semantically interpretable matrix representation, which we refer to as *chroma template*. In particular, the chroma templates reveal consistent and inconsistent aspects across the various stanzas of a recorded song in the form of an explicit and semantically interpretable matrix representation. Altogether, our framework allows for capturing differences in various musical dimensions such as tempo, key, tuning, and melody. As further contribution, we present an application of an user interface that assists folk song researchers in conveniently accessing, listening, and in particular comparing the individual stanzas of a given field recording. In combination, the techniques presented in Part II of the thesis make the actual field recordings more accessible to folk song researcher and constitute a first step towards including the actual recordings and the enclosed performance aspects into folk song research.

In Part III of the thesis, we are dealing with content-based music retrieval. The rapidly growing corpus of digital audio material requires novel retrieval strategies for exploring large music collections and discovering new music. Traditional retrieval strategies rely on metadata that describe the actual audio content in words. In the case that such textual descriptions are not available, one requires content-based retrieval strategies which only utilize the raw audio material. In Chapter 8, we give an overview on content-based retrieval strategies that follow the query-by-example paradigm: given an audio fragment as query, the task is to retrieve all documents that are somehow similar or related to the query from a music collection. Such strategies can be loosely classified according to their *specificity*, which refers to the degree of similarity between the query and the database documents. High specificity refers to a strict notion of similarity, whereas low specificity to a rather vague one. Furthermore, we introduce a second classification principle based on *granularity*, where one distinguishes between fragment-level and document-level retrieval. Using a classification scheme based on specificity and granularity, we identify various classes of retrieval scenarios, which comprise *audio identification*, *audio matching*, and *version identification*. For these three important classes, we give an overview of representative state-of-the-art approaches, which also illustrate the sometimes subtle but crucial differences between the retrieval scenarios. Finally, we give an outlook on an user-oriented retrieval system, which combines the various retrieval strategies in a unified framework.

Furthermore, as main technical contribution of Part III, we deal with the question on how to accelerate cross-version music retrieval. The general goal of cross-version music retrieval is to identify all versions of a given piece of music by means of a short query audio fragment. In particular, we address the fundamental issue on how to build efficient retrieval systems of lower specificity by employing indexing procedures that still exhibit a high degree of robustness against musical variations in the versions. In Chapter 9, we investigate to which extent well-established audio fingerprints, which aim at identifying a specific audio recording, can be modified to also deal with more musical variations between different versions of a piece of music. To this end, we exploit musical knowledge to replace the traditional peak fingerprints based on a spectrogram by peak fingerprints based on other more "musical" feature representations derived from the spectrogram. Our systematic experiments show that such modified peak fingerprints allow for a robust identification of different versions and performances of the same piece of music if the query length is at least 15 seconds. This indicates that highly efficient audio fingerprinting techniques can also be applied to accelerate mid-specific retrieval tasks such as audio matching or cover song identification.

In Chapter 10, we investigate how cross-version retrieval can be accelerated by employing index structures that are based on a shingling approach. To this end, the audio material is split up into small overlapping shingles that consist of short chroma feature subsequences. These shingles are indexed using locality sensitive hashing. Our main idea is to use a shingling approach, where an individual shingle covers a relatively large portion of the audio material (between 10 and 30 seconds). Compared to short shingles, such large shingles have a higher musical relevance so that a much lower number of shingles suffices to characterize a given piece of music. However, increasing the size of a shingle comes at the cost of increasing the dimensionality and possibly loosing robustness to variations. We systematically investigate the delicate trade-off between the query length, feature parameters, shingle dimension, and index settings. In particular, we show that large shingles can still be indexed using locality sensitive hashing with only a small degradation in retrieval quality. In summary, the contributions of Part III of the thesis give valuable insights and indicate solutions that are of fundamental importance for building efficient cross-version retrieval systems that scale to millions of songs and at the same time exhibit a high degree of robustness against musical variations.

1.2 Included Publications

The main contributions of this thesis have been previously published as articles in journals and conference proceedings related to the field of music signal processing.

The contributions of Part I of the thesis have been presented in the publications [73; 72; 71] (related to Chapter 2), in [79] (Chapter 3), and in the publication [77] (Chapter 4). Furthermore, main functionality of the presented techniques has been released in the form of a MATLAB toolbox [74] (Appendix A).

[73] Peter Grosche and Meinard Müller. Extracting predominant local pulse information from music recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1688–1701, 2011.

[71] Peter Grosche and Meinard Müller. Computing predominant local periodicity information in music recordings. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 33–36, New Paltz, NY, USA, 2009.

[72] Peter Grosche and Meinard Müller. A mid-level representation for capturing dominant tempo and pulse information in music recordings. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 189–194, Kobe, Japan, 2009.

[79] 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.

[77] Peter Grosche, Meinard Müller, and Frank Kurth. Cyclic tempogram – a mid-level tempo representation for music signals. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5522–5525, Dallas, Texas, USA, 2010.

[74] Peter Grosche and Meinard Müller. Tempogram toolbox: Matlab implementations for tempo and pulse analysis of music recordings. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, Miami, FL, USA, 2011, late-breaking contribution.

The contributions Part II of the thesis have been presented in the publications [130; 132; 131; 128].

[130] 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, 2009.

[131] Meinard Müller, Peter Grosche, and Frans Wiering. Towards automated processing of folk song recordings. In Eleanor Selfridge-Field, Frans Wiering, and Geraint A. Wiggins, editors, *Knowledge representation for intelligent music processing*, number 09051 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[132] Meinard Müller, Peter Grosche, and Frans Wiering. Automated analysis of performance variations in folk song recordings. In *Proceedings of the International Conference on Multimedia Information Retrieval (MIR)*, pages 247–256, Philadelphia, PA, USA, 2010.

[128] Meinard Müller and Peter Grosche. Automated segmentation of folk song field recordings. In *Proceedings of the 10th ITG Conference on Speech Communication*, Braunschweig, Germany, 2012.

The contributions Part III of the thesis have been presented in the publications [80] (Chapter 8), in [76] (Chapter 9), and in [75] (Chapter 10).

[80] Peter Grosche, Meinard Müller, and Joan Serrà. Audio content-based music retrieval. In Meinard Müller, Masataka Goto, and Markus Schedl, editors, *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups*, pages 157–174. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2012.

[76] Peter Grosche and Meinard Müller. Toward musically-motivated audio fingerprints. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 93–96, Kyoto, Japan, 2012.

[75] Peter Grosche and Meinard Müller. Toward characteristic audio shingles for efficient cross-version music retrieval. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 473–476, Kyoto, Japan, 2012.

1.3 Supplemental Publications

The following publications by the author are also related to music signal processing but are not further considered in this thesis.

[163] Joan Serrà, Meinard Müller, Peter Grosche, and Josep Lluis Arcos. Unsupervised detection of music boundaries by time series structure features. In *Proceedings of the AAAI International Conference on Artificial Intelligence*, Toronto, Ontario, Canada, 2012.

[81] Peter Grosche, Björn Schuller, Meinard Müller, and Gerhard Rigoll. Automatic transcription of recorded music. *Acta Acustica united with Acustica*, 98(2):199–215, 2012.

[93] Nanzhu Jiang, Peter Grosche, Verena Konz, and Meinard Müller. Analyzing chroma feature types for automated chord recognition. In *Proceedings of the 42nd AES Conference on Semantic Audio*, Ilmenau, Germany, 2011.

[129] Meinard Müller, Peter Grosche, and Nanzhu Jiang. A segment-based fitness measure for capturing repetitive structures of music recordings. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, pages 615–620, Miami, FL, USA, 2011.

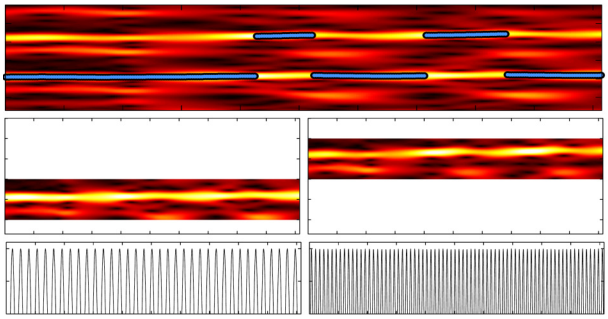
[156] Hendrik Schreiber, Peter Grosche, and Meinard Müller. A re-ordering strategy for accelerating index-based audio fingerprinting. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, pages 127–132, Miami, FL, USA, 2011.

[78] Peter Grosche, Meinard Müller, and Frank Kurth. Tempobasierte Segmentierung von Musikaufnahmen. In *Proceedings of the 36th Deutsche Jahrestagung für Akustik (DAGA)*, Berlin, Germany, 2010.

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

Part I

Beat Tracking and Tempo Estimation



Chapter 2

Predominant Local Pulse Estimation

Most approaches to tempo estimation and beat tracking proceed in two steps. In the first step, positions of note onsets within the music signal are estimated. Here, most approaches capture changes of the signal's energy or spectrum and derive a so-called *novelty curve*. The peaks of such a curve yield good indicators for note onset candidates [7; 23; 189]. In the second step, the novelty curve is analyzed to detect reoccurring patterns and quasi-periodic pulse trains [154; 146; 31; 44]. For non-percussive music with soft note onsets, however, novelty curves provide noisy and irregular information about onset candidates, which makes the extraction of beat and tempo information a difficult problem. Even more challenging becomes the detection of local periodic patterns in the presence of tempo changes.

In this chapter, we introduce a novel approach that allows for a robust extraction of musically meaningful local pulse information even for the case of complex music. Intuitively speaking, our idea is to construct a mid-level representation that explains the local periodic nature of a given (possibly very noisy) onset representation without determining explicit note onset positions. More precisely, starting with a novelty curve, we determine for each time position a sinusoidal kernel that best captures the local peak structure of the novelty curve. Since these kernels localize well in time, even continuous tempo variations and local changes of the pulse level can be handled. Now, instead of looking at the local kernels individually, our crucial idea is to employ an overlap-add technique by accumulating all local kernels over time. As a result, one obtains a single curve that can be regarded as a local periodicity enhancement of the original novelty curve. Revealing *predominant local pulse* (PLP) information, this curve is referred to as *PLP curve.*

Our PLP concept yields a powerful mid-level representation that can be applied as a flexible tool for various music analysis tasks. In particular, we discuss in detail how the PLP concept can be applied for improving on tempo estimation as well as for validating the local tempo estimates. Furthermore, we show that state-of-the-art beat trackers can be improved when using a PLP-enhanced novelty representation. Here, one important feature of our work is that we particularly consider music recordings that reveal changes in tempo, whereas most of the previous tempo estimation and beat tracking approaches assume a (more or less) constant tempo throughout the recording. As it turns out, our PLP concept is capable of capturing continuous tempo changes as implied by ritardando or accelerando. However, as our approach relies on the assumption of a locally quasi-periodic behavior of the signal, it reaches its limits in the presence of strong local tempo distortions as found in highly expressive music (e. g. romantic piano music). To demonstrate the practical relevance of our PLP concept, we have conducted extensive experiments based on several music datasets consisting of 688 recordings amounting to more than 36 hours of annotated audio material. The datasets cover various genres including popular music, Jazz music and classical music.

The remainder of this chapter is organized as follows. In Section 2.1, we review related work and discuss relevant state-of-the-art concepts. In Section 2.2, we then give an overview of our PLP concept. Subsequently, we elaborate on the mathematical details of our variant of a novelty curve (Section 2.3), tempograms (Section 2.4), the determination of the optimal periodicity kernels (Section 2.5), and the computation of the PLP curves (Section 2.6). Then, we discuss general properties of PLP curves (Section 2.7) and describe an iterative approach (Section 2.8). The applications to tempo estimation and beat tracking as well as the corresponding experiments are discussed in Section 2.9. Conclusions of this chapter are given in Section 2.10.

2.1 Related Work

In general, the beat is a perceptual phenomenon and perceptual beat times do not necessarily coincide with physical beat times [42]. Furthermore, the perception of beats varies between listeners. However, beat positions typically go along with note onsets or percussive events. Therefore, in most tempo and beat tracking approaches, the first step consists in locating such events in the given signal—a task often referred to as *onset detection* or *novelty detection*. To determine the physical starting times of the notes occurring in the music recording, the general idea is to capture changes of certain properties of the signal to derive a *novelty curve*. The peaks of this curve indicate candidates for note onsets.

Many different methods for computing novelty curves have been proposed, see [7; 23; 39] for an overview. When playing a note, the onset typically goes along with a sudden increase of the signal's energy. In the case of a pronounced attack phase, note onset candidates may be determined by locating time positions, where the signal's amplitude envelope starts to increase [7; 67]. Much more challenging, however, is the detection of onsets in the case of non-percussive music, where one has to deal with soft onsets or blurred note transitions. This is often the case for classical music dominated by string instruments. As a result, more refined methods have to be used for computing a novelty curve, e. g., by analyzing the signal's spectral content [88; 7; 189; 50], pitch [88; 189; 24], harmony [47; 61], or phase [88; 7; 86]. To handle the variety of signal types, a combination of novelty curves and signal features can improve the detection accuracy [88; 35; 189; 169; 50]. Also supervised classification approaches were proposed [108; 50].

Furthermore, in complex polyphonic mixtures of music, simultaneously occurring events of high intensities lead to masking effects that prevent any observation of an energy increase of a low intensity onset. To circumvent these masking effects, detection functions were proposed that analyze the signal in a bandwise fashion [100] to extract transients occurring in certain frequency regions of the signal. As a side-effect of a sudden energy increase, there appears an accompanying broadband noise burst in the signal's spectrum. This effect is mostly masked by the signal's energy in lower frequency regions but well detectable in the higher frequency regions [118] of the spectrum. Here, logarithmic compression [100] and spectral whitening [167] are techniques for enhancing the high-frequency information. Some of these approaches are employed for computing our novelty curves, see Section 2.3.

To derive the beat period and the tempo from a novelty curve, one strategy is to explicitly determine note onset positions and then to reveal the structure of these events. For the selection of onset candidates, one typically employs peak picking strategies based on adaptive thresholding [7]. Each pair of note onset positions then defines an inter-onset-interval (IOI). Considering suitable histograms or probabilities of the occurring IOIs, one may derive hypotheses on the beat period and tempo [40; 37; 67; 159; 33]. The idea is that IOIs frequently appear at integer multiples and fractions of the beat period. Similarly, one may compute the autocorrelation of the extracted onset times [61] to derive the beat period. The drawback of these approaches is that they rely on an explicit localization of a discrete set of note onsets—a fragile and error-prone step. In particular, in the case of weak and blurry onsets the selection of the relevant peaks of the novelty curve that correspond to true note onsets becomes a difficult or even infeasible problem.

Avoiding the explicit extraction of note onset, the novelty curves can directly be analyzed with respect to reoccurring or quasi-periodic patterns. Here, generally speaking, one can distinguish between three different methods for measuring periodicities. The autocorrelation method allows for detecting periodic self-similarities by comparing a novelty curve with time-shifted copies [31; 44; 145; 146; 160; 36]. Another widely used method is based on a bank of comb filter resonators, where a novelty curve is compared with templates consisting of equally spaced spikes representing various frequencies [102; 154]. Similarly, one can use a short-time Fourier transform [146; 147; 187] or a non-stationary Gabor transform [89] to derive a frequency representation of the novelty curve. Here, the novelty curve is compared with sinusoidal templates representing specific frequencies. Each of the methods reveals periodicities of the underlying novelty curve, from which one can estimate the tempo or beat. The characteristics of the periodicities typically change over time and can be visualized by means of spectrogram-like representations referred to as *tempogram* [21], *rhythmogram* [92], or *beat spectrogram* [54].

More challenging becomes the detection of periodic patterns in the case that the music recordings reveal significant tempo changes. This often occurs in performances of classical music as a result of ritardandi, accelerandi, fermatas, and so on [37]. Furthermore, the extraction problem is complicated by the fact that the notions of tempo and beat are ill-defined and highly subjective due to the complex hierarchical structure of rhythm [139; 66]. For example, there are various levels that contribute to the human perception of tempo and beat. Typically, previous work focuses on determining musical pulses on the *tactus* (the foot tapping rate or beat [112]) level [44; 146; 31], but only few approaches exist for analyzing the signal on the measure level [61; 102; 148; 137] or finer tatum level [159; 141; 34]. The *tatum* or *temporal atom* refers to the fastest repetition rate of musically meaningful accents occurring in the signal [13]. Various approaches have been suggested that simultaneously analyze different pulse levels [148; 160; 27; 68; 102].

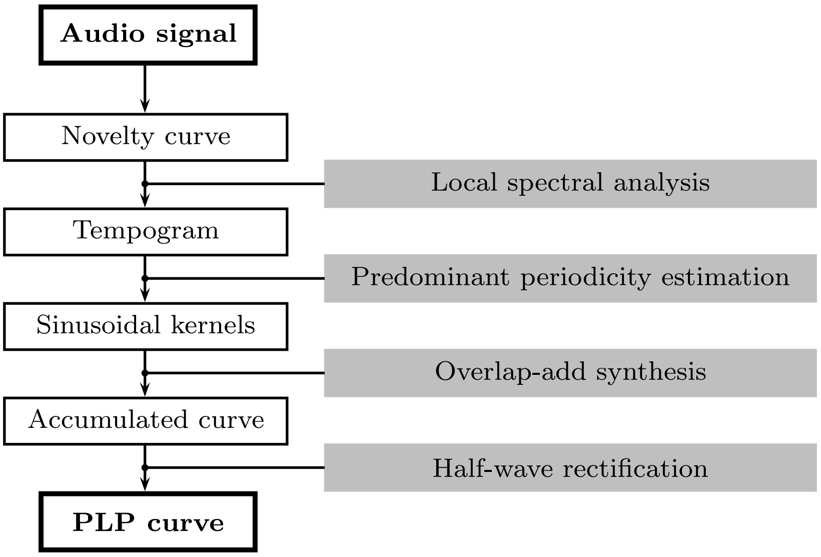


Figure 2.1: Flowchart of the steps involved in the PLP computation.

2.2 Overview of the PLP Concept

We now give an overview of the steps involved in the PLP computation, see Figure 2.1 for a schematic overview and Figure 2.2 for an example. The input of our procedure consists of a spike-like novelty curve, see Figure 2.2a. In the first step, we derive a time-pulse representation, referred to as *tempogram*, by performing a local spectral analysis of the novelty curve, see Figure 2.2b. Here, we avoid the explicit determination of note onsets, which generally is an error-prone and fragile step. Then, from the tempogram, we determine for each time position the sinusoidal periodicity kernel that best explains the local periodic nature of the novelty curve in terms of period (frequency) and timing (phase), see Figure 2.2c. Since there may be a number of outliers among these kernels, one usually obtains unstable information when looking at these kernels in a one-by-one fashion. Therefore, as one main idea of our approach, we use an overlap-add technique by accumulating all these kernels over time to obtain a single curve, see Figure 2.3b. In a final step, we apply a half-wave rectification (only considering the positive part of the curve) to obtain the mid-level representation we refer to as predominant local pulse (PLP) curve, see Figure 2.3c. As it turns out, such PLP curves are robust to outliers and reveal musically meaningful periodicity information even when starting with relatively poor onset information.

2.3 Novelty Curve

Our PLP concept is based on a novelty curve as typically used for note onset detection tasks. We now describe the approach for computing novelty curves used in our experiments. In our variant, we combine ideas and fundamental concepts of various state-of-the-

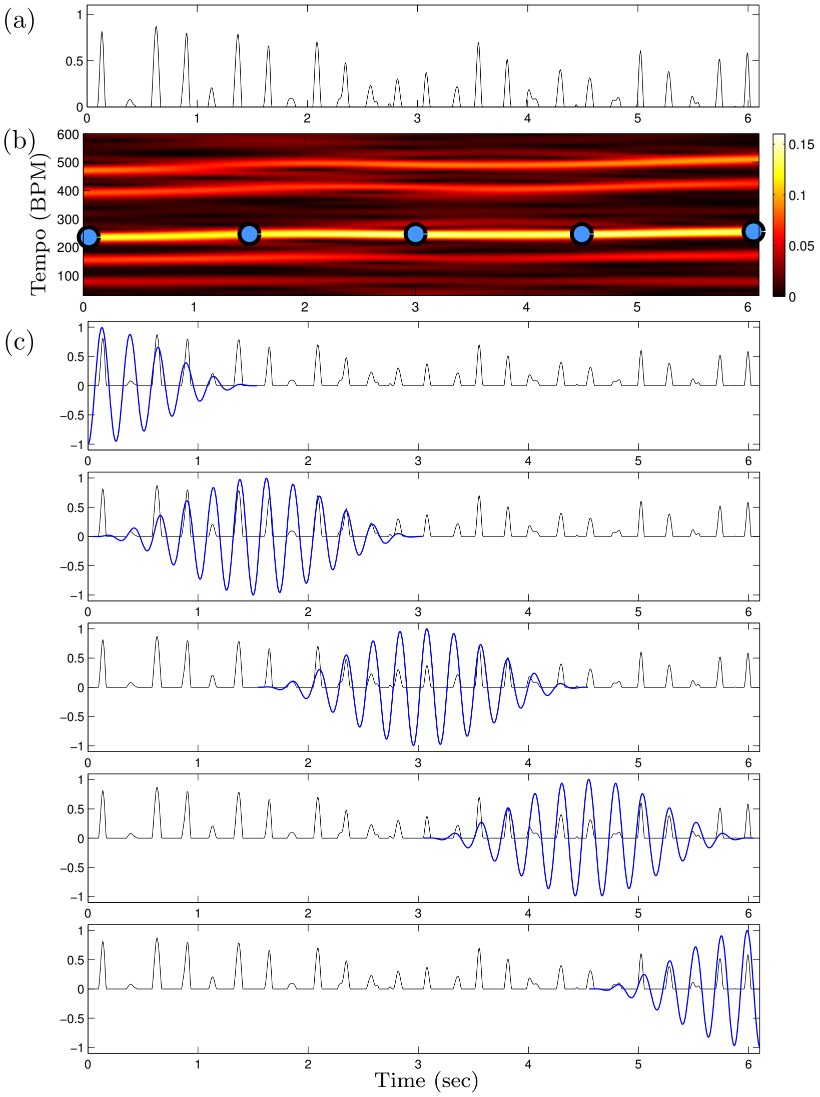


Figure 2.2: Illustration of the estimation of optimal periodicity kernels. **(a)** Novelty curve Δ. **(b)** Magnitude tempogram $|\mathcal{T}|$ with maxima (indicated by circles) shown at five time positions t. **(c)** Optimal sinusoidal kernels κ_t (using a kernel size of 3 seconds) corresponding to the maxima. Note how the kernels capture the local peak structure of the novelty curve in terms of frequency and phase.

art methods [7; 100; 102; 189]. Our novelty curve is particularly designed for also revealing meaningful note onset information for complex music, such as orchestral pieces dominated by string instruments. Note, however, that the particular design of the novelty curve is not the focus of this thesis. The mid-level representations as introduced in the following are designed to work even for noisy novelty curves with a poor peak structure. Naturally, the overall result may be improved by employing more refined novelty curves as suggested in [88; 189; 50].

Recall from Section 2.1 that a note onset typically goes along with a sudden change of the signal's energy and spectral content. In order to extract such changes, given a music

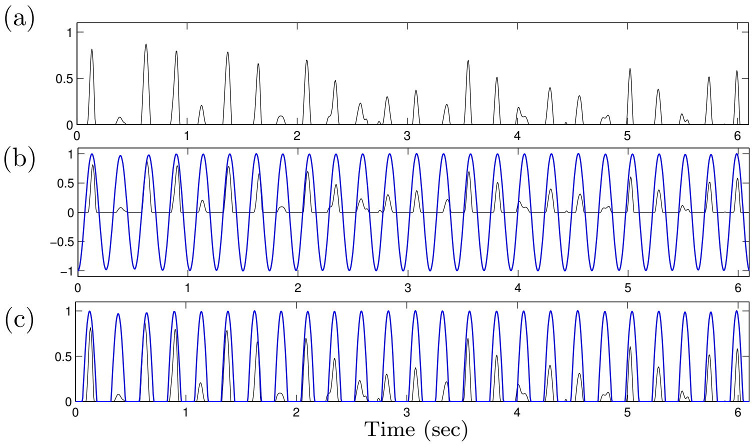


Figure 2.3: Illustration of the PLP computation from the optimal periodicity kernels shown in Figure 2.2c. **(a)** Novelty curve Δ. **(b)** Accumulation of all kernels (overlap-add). **(c)** PLP curve Γ obtained after half-wave rectification.

recording, a short-time Fourier transform is used to obtain a spectrogram

$$X = (X(k,t))_{k,t}$$

with $k \in [1:K]$ and $t \in [1:T]$. Here, K denotes the number of Fourier coefficients, T denotes the number of frames, and $X(k,t)$ denotes the k^{th} Fourier coefficient for time frame t. In our implementation, the discrete Fourier transforms are calculated over Hann-windowed frames of length 46 ms with 50% overlap. Consequently, each time parameter t corresponds to 23 ms of the audio recording.

Note that the Fourier coefficients of X are linearly spaced on the frequency axis. Using suitable binning strategies, various approaches switch over to a logarithmically spaced frequency axis, e. g., by using mel-frequency bands or pitch bands, see [100]. Here, we keep the linear frequency axis, since it puts greater emphasis on the high-frequency regions of the signal, thus accentuating noise bursts that are typically visible in the high-frequency spectrum. Similar strategies for accentuating the high frequency content for onset detection are proposed in [118; 23].

In the next step, we apply a logarithm to the magnitude spectrogram $|X|$ of the signal yielding

$$Y := \log(1 + C \cdot |X|)$$

for a suitable constant $C > 1$, see [100; 102]. Such a compression step not only accounts for the logarithmic sensation of sound intensity but also allows for adjusting the dynamic range of the signal to enhance the clarity of weaker transients, especially in the high-frequency regions. In our experiments, we use the value $C = 1000$, but our results as well as the findings reported by Klapuri et al. [102] show that the specific choice of C does not effect the final result in a substantial way. The effect of this compression step is illustrated by Figure 2.4 for a recording of Beethoven's Fifth Symphony. Figure 2.4a shows the piano reduced version of the first 12 measures of the score. The audio recording is an orchestral version conducted by Bernstein. Figure 2.4c shows the magnitude spectrogram $|X|$ and

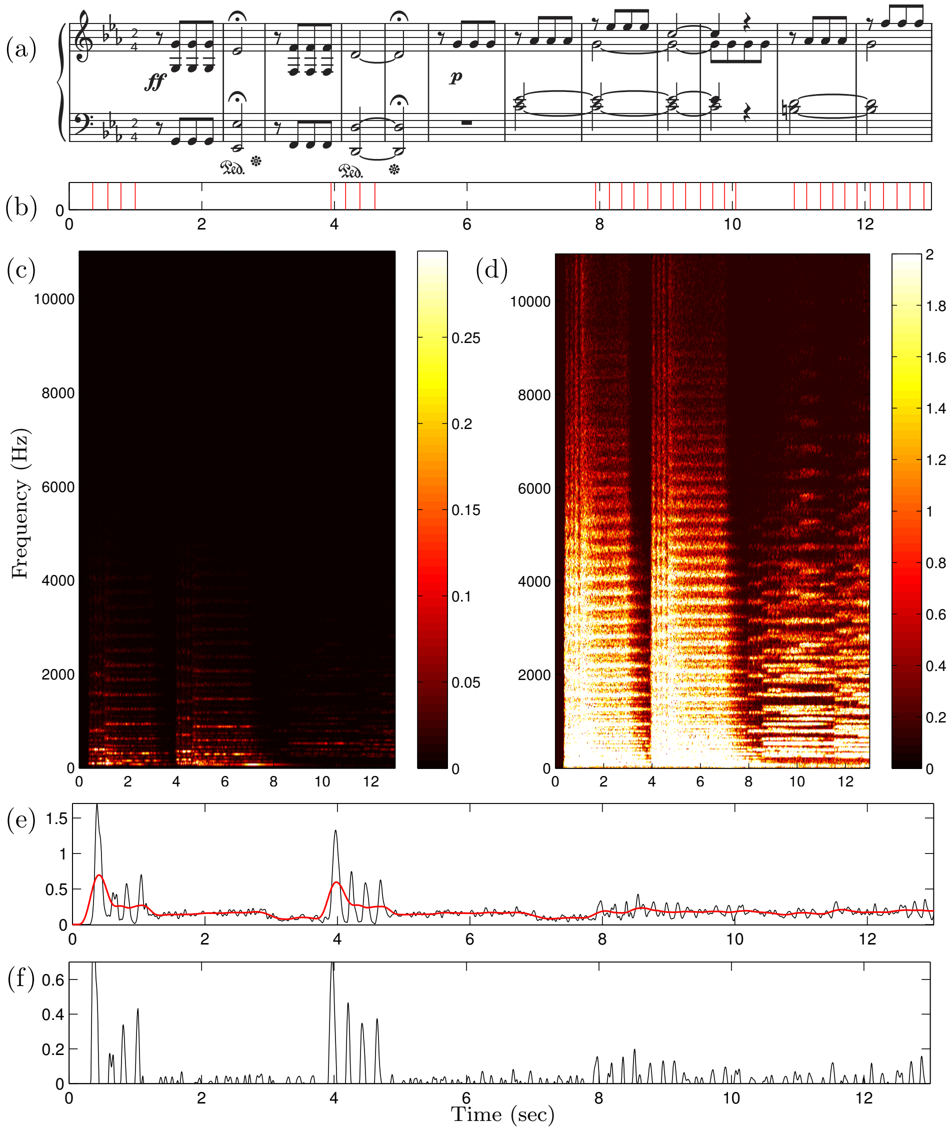


Figure 2.4: First 12 measures of Beethoven's Symphony No. 5 (Op. 67). **(a)** Score representation (in a piano reduced version). **(b)** Annotated reference onsets (for an orchestral audio recording conducted by Bernstein). **(c)** Magnitude spectrogram $|X|$. **(d)** Logarithmically compressed magnitude spectrogram Y. **(e)** Novelty curve $\bar{\Delta}$ and local mean (red curve). **(f)** Novelty curve Δ.

Figure 2.4d the compressed spectrogram Y using $C = 1000$. As a result of the logarithmic compression, events with low intensities are considerably enhanced in Y, especially in the high frequency range.

To obtain a novelty curve, we basically apply a first order differentiator to compute the discrete temporal derivative of the compressed spectrum Y. In the following, we only consider note onsets (positive derivative) and not note offsets (negative derivative). Therefore, we sum up only over positive intensity changes to obtain the novelty function

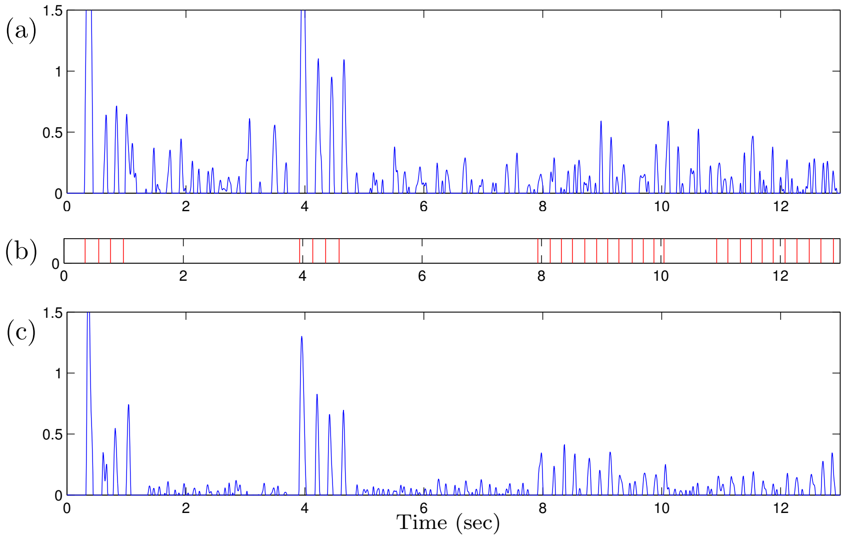


Figure 2.5: Illustrating the effect of the logarithmic compression on the resulting novelty curves. **(a)** Novelty curve based on the magnitude spectrogram $|X|$ (see Figure 2.4c). **(b)** Manually annotated reference onsets. **(c)** Novelty curve Δ based on the logarithmically compressed magnitude spectrogram Y (see Figure 2.4d).

$\bar{\Delta} : [1 : T - 1] \to \mathbb{R}$:

$$\bar{\Delta}(t) := \sum_{k=1}^{K} |Y(k, t + 1) - Y(k, t)|_{\geq 0}. \tag{2.1}$$

for $t \in [1 : T - 1]$, where $|x|_{\geq 0} := x$ for a non-negative real number x and $|x|_{\geq 0} := 0$ for a negative real number x. Figure 2.4e shows the resulting curve for the Beethoven example. To obtain our final novelty function Δ, we subtract the local mean (red curve in Figure 2.4e) from $\bar{\Delta}$ and only keep the positive part (half-wave rectification), see Figure 2.4f. In our implementation, we actually use a higher-order smoothed differentiator [2]. Furthermore, we process the spectrum in a bandwise fashion using 5 bands. Similar as in [154] these bands are logarithmically spaced and non-overlapping. Each band is roughly one octave wide. The lowest band covers the frequencies from 0 Hz to 500 Hz, the highest band from 4000 Hz to 11025 Hz. The resulting 5 novelty curves are summed up to yield the final novelty function.

The resulting novelty curve for our Beethoven example reveals the note onset candidates in the form of impulse-like spikes. Actually, this piece constitutes a great challenge for onset detection as, besides very dominant note onsets in the fortissimo section at the beginning of the piece (measures 1-5), there are soft and blurred note onsets in the piano section which is mainly played by strings (measures 6-12). This is also reflected by the novelty curve shown in Figure 2.4f. The strong onsets in the fortissimo section result in very pronounced peaks. The soft onsets in the piano section (seconds 8-13), however, are much more difficult to be distinguished from the spurious peaks not related to any note onsets. In this context, the logarithmic compression plays a major role. Figure 2.5 compares the novelty curve Δ with a novelty curve directly derived from the magnitude spectrogram $|X|$ without applying a logarithmic compression. Actually, omitting the logarithmic compression (Figure 2.5a) results in a very noisy novelty curve that does not reveal musically meaningful onset information in the piano section. The novelty curve Δ (Figure 2.5b), however, still possesses a regular peak structure in the problematic sections. This clearly illustrates the benefits of the compression step. Note that the logarithmic compression of the spectrogram gives higher weight to an absolute intensity difference within a quiet region of the signal than within a louder region, which follows the psychoacoustic principle that a just-noticeable change in intensity is roughly proportional to the absolute intensity [51]. Furthermore, the compression leads to a better temporal localization of the onset, because the highest relative slope of the attack phase approaches the actual onset position and noticeably reduces the influence of amplitude changes (e.g. tremolo) in high intensity regions. Further examples of our novelty curve are discussed in Section 2.7.

The variant of a novelty curve described in this section combines important design principles and ideas of various approaches proposed in the literature. The basic idea of considering temporal differences of a spectrogram representation is well known from the *spectral flux* novelty curve, see [7]. This strategy works particularly well for percussive note onsets but is not suitable for less pronounced onsets (see Figure 2.5a). One well known variant of the spectral flux strategy is the *complex domain method* as proposed in [8]. Here, magnitude and phase information is combined in a single novelty curve to emphasize weak note onsets and smooth note transitions. In our experiments, the logarithmic compression has a similar effect as jointly considering magnitude and phase, but showed more robust results in many examples. Another advantage of our approach is that the compression constant C allows for adjusting the compression. The combination of magnitude compression and phase information did not lead to a further increase in robustness.

2.4 Tempogram

A novelty curve typically reveals the note onset candidates in the form of impulse-like spikes. Because of extraction errors and local tempo variations, the spikes may be noisy and irregularly spaced over time. Dealing with spiky novelty curves, autocorrelation methods [44] as well as comb filter techniques [154] may have difficulties in capturing the quasi-periodic information. This is due to the fact that spiky structures are hard to identify by means of spiky analysis functions in the presence of irregularities. In such cases, smoothly spread analysis functions such as sinusoids are better suited to detect locally distorted quasi-periodic patterns. Therefore, similar to [146], we use a short-time Fourier transform to analyze the local periodic structure of the novelty curves.

The novelty curve as described in Section 2.3 is simply a function $\Delta : [1 : T] \to \mathbb{R}$ indicating note onset candidates in the form of peaks, where $[1 : T] := \{1, 2, \ldots, T\}$, for some $T \in \mathbb{N}$, represents the sampled time axis with respect to a fixed sampling rate. To avoid boundary problems, we assume that Δ is defined on $\mathbb{Z}$ by setting $\Delta(t) := 0$ for $t \in \mathbb{Z} \setminus [1 : T]$. Furthermore, we fix a window function $W : \mathbb{Z} \to \mathbb{R}$ centered at $t = 0$ with support $[-N : N]$ for some $N \in \mathbb{N}$. In the following, we use a Hann window of size $2N+1$, which is normalized to yield $\sum_{t \in \mathbb{Z}} W(s-t) = 1$ for all $s \in [1 : T]$. Then, for a frequency parameter $\omega \in \mathbb{R}_{\geq 0}$, the complex Fourier coefficient $\mathcal{F}(t, \omega)$ is defined by

$$\mathcal{F}(t, \omega) = \sum_{n \in \mathbb{Z}} \Delta(n) \cdot W(n-t) \cdot \mathrm{e}^{-2\pi i \omega n} \ . \tag{2.2}$$

Note that the frequency ω corresponds to the period $1/\omega$. In the context of music, we rather think of tempo measured in beats per minutes (BPM) than of frequency measured in Hertz (Hz). Therefore, we use a tempo parameter τ satisfying the equation $\tau = 60 \cdot \omega$.

Similar to a spectrogram, which yields a time-frequency representation, a *tempogram* is a two-dimensional *time-pulse representation* indicating the strength of a local pulse over time, see also [21; 146]. Here, intuitively, a *pulse* can be thought of a periodic sequence of accents, spikes or impulses. We specify the periodicity of a pulse in terms of a tempo value (in BPM). Now, let $\Theta \subset \mathbb{R}_{>0}$ be a finite set of tempo parameters. Then, we model a tempogram as a function $\mathcal{T} : [1 : T] \times \Theta \to \mathbb{C}$ defined by

$$\mathcal{T}(t, \tau) = \mathcal{F}(t, \tau/60) \ . \tag{2.3}$$

For an example, we refer to Figure 2.2b, which shows the magnitude tempogram $|\mathcal{T}|$ for the novelty curve shown in Figure 2.2a. Intuitively, the magnitude tempogram indicates for each time position how well the novelty curve can be locally represented by a pulse track of a given tempo. Note that the complex-valued tempogram contains not only magnitude information, but phase information as well. In our experiments, we mostly compute $\mathcal{T}$ using the set $\Theta = [30 : 600]$ covering the (integer) musical tempi between 30 and 600 BPM. Here, the bounds are motivated by the assumption that only events showing a temporal separation between roughly 100 ms (600 BPM) and 2 seconds (30 BPM) contribute to the perception of tempo [139]. This tempo range requires a spectral analysis of high resolution in the lower frequency range. Therefore, a straightforward FFT is not suitable. However, since only relatively few frequency bands (tempo values) are needed for the tempogram, computing the required Fourier coefficients individually according to Eq. (2.2) has still a reasonable computational complexity. Typically, we set W to be a Hann window with the size $2N + 1$ corresponding to 4-12 seconds of the audio. The overlap of adjacent windows is adjusted to yield a frame rate of 5 Hz (five frames per second). For a more detailed explanation and a general overview on different tempogram representations, we refer to Chapter 4.

2.5 Predominant Local Periodicity

We now make use of both, the magnitudes and the phases given by $\mathcal{T}$, to derive a mid-level representation that captures the *predominant local pulse* (PLP) of the underlying music signal. Here, the term *predominant pulse* refers to the pulse that is most noticeable in the novelty curve in terms of intensity. Furthermore, our representation is *local* in the sense that it yields the predominant pulse for each time position, thus making local tempo information explicit.

For each $t \in [1 : T]$ we compute the tempo parameter $\tau_t \in \Theta$ that maximizes the magnitude of $\mathcal{T}(t, \tau)$:

$$\tau_t := \operatorname*{argmax}_{\tau \in \Theta} |\mathcal{T}(t, \tau)| \ . \tag{2.4}$$

Figure 2.2b exemplarily shows the predominant local periodicity τ_t for five $t \in [1:T]$ of the magnitude tempogram. The corresponding phase φ_t is defined by [123]:

$$\varphi_t := \frac{1}{2\pi} \arccos\left(\frac{\mathrm{Re}(\mathcal{T}(t,\tau_t))}{|\mathcal{T}(t,\tau_t)|}\right). \tag{2.5}$$

Using τ_t and φ_t, the optimal sinusoidal kernel $\kappa_t : \mathbb{Z} \to \mathbb{R}$ for $t \in [1:T]$ is defined as the windowed sinusoid

$$\kappa_t(n) := W(n-t)\cos(2\pi(n \cdot \tau_t/60 - \varphi_t)) \tag{2.6}$$

for $n \in \mathbb{Z}$ and the same window function W as used for the tempogram computation in Eq. (2.2). Figure 2.2c shows the five optimal sinusoidal kernels for the five time parameters indicated in Figure 2.2b using a Hann window of three seconds. Intuitively, the sinusoid κ_t best explains the local periodic nature of the novelty curve at time position t with respect to the set Θ. The period $60/\tau_t$ corresponds to the predominant periodicity of the novelty curve and the phase information φ_t takes care of accurately aligning the maxima of κ_t and the peaks of the novelty curve. The properties of the kernels κ_t depend not only on the quality of the novelty curve, but also on the window size $2N+1$ of W and the set of frequencies Θ. Increasing the parameter N yields more robust estimates for τ_t at the cost of temporal flexibility. In the following, this duration is referred to as *kernel size* (KS) and is specified in seconds.

2.6 PLP Curve

The estimation of optimal periodicity kernels in regions with a strongly corrupted peak structure is problematic. This particularly holds in the case of small kernel sizes. To make the periodicity estimation more robust, our idea is to apply an overlap-add technique, where we accumulate these kernels over all time positions to form a single function instead of looking at the kernels in a one-by-one fashion. Furthermore, we only consider the positive part of the resulting curve (half-wave rectification). More precisely, we define a function $\Gamma : [1:T] \to \mathbb{R}_{\geq 0}$ as follows:

$$\Gamma(n) = \left| \sum_{t \in [1:T]} \kappa_t(n) \right|_{\geq 0} \tag{2.7}$$

for $n \in [1:T]$, where $|x|_{\geq 0} := x$ for a non-negative real number x and $|x|_{\geq 0} := 0$ for a negative real number x. The resulting function is our mid-level representation referred to as *PLP curve*. Figure 2.3b shows the accumulated curve for the five optimal periodicity kernels shown in Figure 2.2c. Note, how the maxima of the periodicity kernels not only align well with the peaks of the novelty curve, but also with the maxima of neighboring kernels in the overlapping areas, which leads to constructive interferences. Furthermore note that, because of the normalization of the window W (see Section 2.4), the values of the curve lie in the interval $[-1,1]$ and a local maximum is close to the value one if and only if the overlapping kernels align well. From this, the final PLP curve Γ is obtained through half-wave rectification, see Figure 2.3c.

Note that taking the framewise maximum as in Eq. (2.4) has its assets and drawbacks. On the one hand, it allows the PLP curve to quickly adjust to even sudden changes in tempo and in the dominating pulse level, see Figure 2.6 for an example. On the other hand, taking the framewise maximum may lead to unwanted jumps such as random switches between tempo octaves in the tempo trajectory defined by the maximizing tempo parameter. Here, instead of simply using the context-independent framewise maximum, one may use optimization techniques based on dynamic programming to obtain a context-sensitive smooth tempo trajectory [146; 3]. Similarly, one may constrain the set Θ of tempo parameters in the maximization covering only tempo parameters in a suitable neighborhood of an expected (average) tempo value. Because of the subsequent accumulation step, a small number of outliers does not effect the overall properties of the PLP curve. A larger number of outliers or unwanted switches between tempo octaves, however, may deteriorate the result. Our PLP framework allows for incorporating additional constraints and smoothness strategies in the kernel selection to adjust the properties of the resulting PLP curve according to the requirements of a specific application. The issue of kernel selection will be further discussed in Section 2.7 and Section 2.9.6.

2.7 Discussion of Properties

We now discuss various properties of PLP curves based on representative examples to demonstrate the benefits of our concept. For an extensive quantitative analysis, we refer to Section 2.9.

As first example, we consider the Waltz No. 2 from Dimitri Shostakovich's Suite for Variety Orchestra No. 1. Figure 2.6a shows an excerpt (measures 25 to 36) of a piano reduced version of the score of this piece. The audio recording in this example is an orchestral version conducted by Yablonsky. (The audio excerpt corresponding to measures 25 to 36 has a duration of ten seconds.) The manually annotated reference onset positions in this audio excerpt are indicated by the vertical lines in Figure 2.6b. The novelty curve for this except is shown in Figure 2.6c. Note that the peaks of this curve strongly correlate with the onset positions. However, the first beats (downbeats) in this 3/4 Waltz are played softly by non-percussive instruments leading to relatively weak and blurred onsets, whereas the second and third beats are played staccato supported by percussive instruments. As a result, the peaks of the novelty curve corresponding to downbeats are hardly visible or even missing, whereas peaks corresponding to the percussive beats are much more pronounced.

Figure 2.6d shows the magnitude tempogram computed from the novelty curve using a kernel size KS = 3 sec. Obviously, this tempogram indicates a significant tempo at 210 BPM throughout the audio excerpt, which actually corresponds to the quarter note pulse (tactus level) of the piece. Note that this tempo is clearly indicated despite of poor and missing peaks in the novelty curve. Furthermore, the magnitude tempogram additionally reveals high intensities at 420 BPM, which corresponds to the double tempo or eighth note pulse (tatum) level of the piece. Looking at the score, one would expect a predominant tempo which corresponds to the tactus level (score reveals quarter note pulse) for measures 25-28, 31/32 and 35/36 and to the tatum level (score reveals eighth note pulse) for measures 29/30 and 33/34. Indeed, this is exactly reflected by the lines

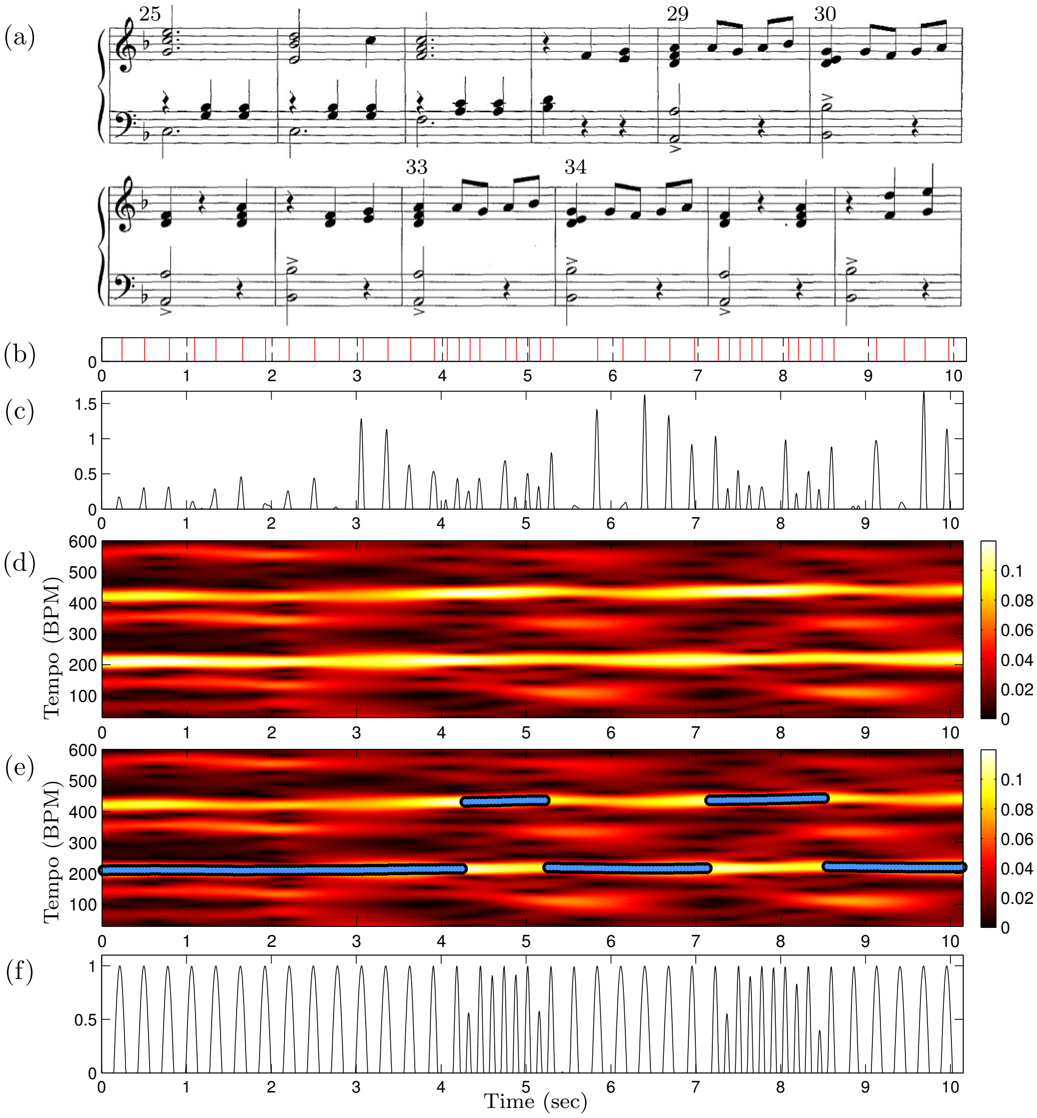


Figure 2.6: Excerpt of Shostakovich's Waltz No. 2 from the *Suite for Variety Orchestra No. 1.* **(a)** Score representation of measures 25 to 36 (in a piano reduced version). **(b)** Annotated reference onsets (for an orchestral audio recording conducted by Yablonsky). **(c)** Novelty curve Δ. **(d)** Magnitude tempogram $|\mathcal{T}|$ using $\Theta = [30 : 600]$. **(e)** Magnitude tempogram $|\mathcal{T}|$ with indication of the predominant tempo. **(f)** PLP curve Γ.

in Figure 2.6e, which indicate the predominant tempo (maximum intensity) for each time position. Note that one has a pulse level switch to the tatum level exactly for the seconds 4-5 (measures 29/30) and seconds 7-8 (measures 33/34).

The PLP curve Γ shown in Figure 2.6f is obtained from the local tempo estimates. Note that the predominant pulse positions are clearly indicated by the peaks of the PLP curve even though some of the expected peaks where missing in the original novelty curve. Also, the switches between the tactus and tatum level are captured by the PLP curve. In other words, the PLP curve can be regarded as a local periodicity enhancement of the original

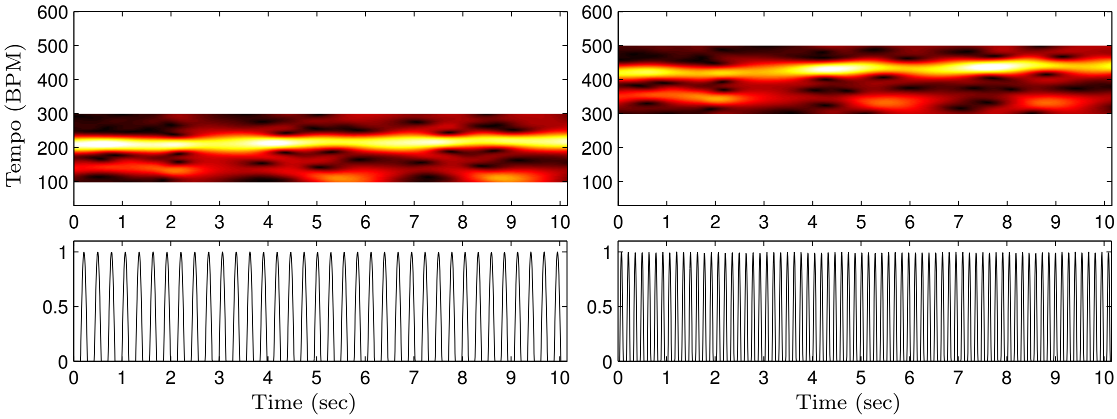


Figure 2.7: Magnitude tempogram and resulting PLP curve using a constrained tempo set for the Shostakovich example shown in Figure 2.6. **Left:** $\Theta = [100 : 300]$ (quarter note tempo range). **Right:** $\Theta = [300 : 500]$ (eighth note tempo range).

novelty curve, where the predominant pulse level is taken into account. Although our concept is designed to reveal such locally predominant information, for some applications the local nature of these estimates might not be desirable. Actually, our PLP framework allows for incorporating prior knowledge on the expected tempo range to exhibit information on different pulse levels. Here, the idea is to constrain the set Θ of tempo parameters in the maximization, see Eq. (2.4). For example, using a constrained set $\Theta = [100 : 300]$ instead of the original set $\Theta = [30 : 600]$, one obtains the tempogram and PLP curve shown in Figure 2.7 on the left. In this case, the PLP curve correctly reveals the quarter note (tactus) pulse positions with a tempo of 210 BPM. Similarly, using the set $\Theta = [300 : 500]$ reveals the eighth (tatum) note pulse positions and the corresponding tempo of 420 BPM, see Figure 2.7 on the right. In other words, in the case there is a dominant pulse of (possibly varying) tempo within the specified tempo range Θ, the PLP curve yields a good pulse tracking on the corresponding pulse level.

As second example, we again consider the orchestral version of Beethoven's Fifth Symphony conducted by Bernstein. Figure 2.8a shows the piano reduced version of the first 12 measures of the score. Recall that this piece constitutes a great challenge for novelty detection as there are soft and blurred note onsets in the piano section which is mainly played by strings. In particular, the height of a peak in the resulting novelty curve (see Figure 2.8b) is not necessarily a good indicator for the relevance of the peak. However, even though corrupted, the peak structure still possesses some local periodic regularities. These regularities are captured by the periodicity kernels and revealed in the magnitude tempogram shown in Figure 2.8c. Here, at the beginning (second 0 to 6), a tempo of roughly 280 BPM dominates the tempogram. During the second fermata (second 6-7) the tempogram does not show any pronounced tempo. However, in the piano section, the tempogram again indicates a dominating tempo of roughly 300 BPM, which actually corresponds to the eighth note pulse level. Finally, Figure 2.8d shows the PLP curve Γ. Note that the peaks of Γ align well with the musically relevant onset positions. While note onset positions in the fortissimo section can be directly determined from the original novelty curve, this becomes problematic for the onsets in the piano section. However,

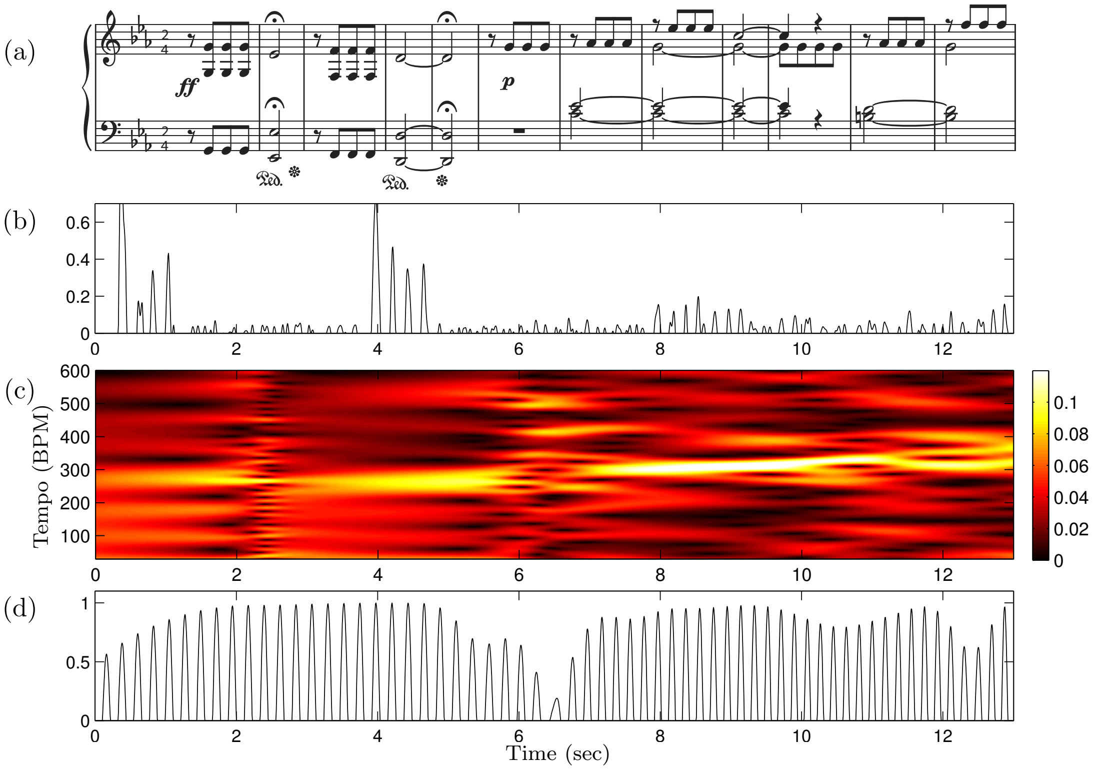


Figure 2.8: First 12 measures of Beethoven's Symphony No. 5 (Op. 67). **(a)** Piano reduced version of the score. **(b)** Novelty curve Δ. **(c)** Magnitude tempogram $|\mathcal{T}|$. **(d)** PLP curve Γ.

exploiting that the note onsets lie on a local rhythmic grid, the PLP curve is capable of capturing meaningful onset information even in the piano passage.

As another important property of our concept, a PLP curve not only reveals positions of predominant pulses but also indicates a kind of confidence in the estimation. Note that the amplitudes of the periodicity kernels do not depend on the amplitude of the novelty curve. This makes a PLP curve invariant under changes in dynamics of the underlying music signal. Recall that we estimate the periodicity kernels using a sliding window technique and add up the kernels over all considered time positions. Since neighboring kernels overlap, constructive and destructive interference phenomena in the overlapping regions influence the amplitude of the resulting PLP curve Γ. Consistent local tempo estimates result in consistent kernels, which in turn produce constructive interferences in the overlap-add synthesis. In such regions, the peaks of the PLP curve assume a value close to one. In contrast, random local tempo estimates result in inconsistent kernels, which in turn cause destructive interferences and lower values of Γ. In Figure 2.8d, this effect is visible in the fermata section (seconds 5 to 8). In Section 2.9.4, we show how this property of PLP curves can be used to detect problematic passages in audio recordings.

Finally, we give a first indication in which way our PLP concept is capable of capturing local tempo changes. To this end, we distinguish between two types of tempo changes. The first type concerns moderate and continuous tempo changes as typically implied by an accelerando or ritardando. To simulate such tempo changes, we generated a pulse train of increasing tempo in the first half and of decreasing tempo in the second half. Figure 2.9

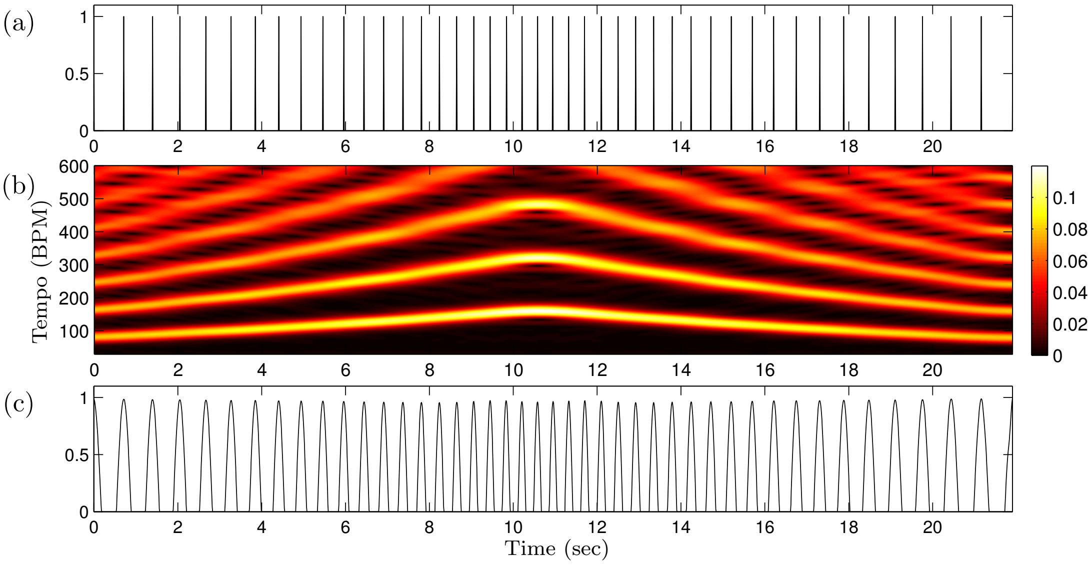


Figure 2.9: Behavior of PLP curves under continuous tempo changes (accelerando, ritardando). **(a)** Impulse train of increasing tempo (80 to 160 BPM, first part) and decreasing tempo (160 to 80 BPM, second part). **(b)** Magnitude tempogram $|\mathcal{T}|$ for KS = 4 sec. **(c)** PLP curve.

shows the resulting novelty curve, the magnitude tempogram, and the PLP curve. As this example indicates, the PLP curve captures well such types of continuous tempo changes—even the amplitude of the PLP curve indicates a high confidence of the estimation. The second type concerns strong local tempo distortions as found in highly expressive music (e. g. romantic piano music). To simulate such tempo changes, we first generated a pulse train of constant tempo (160 BPM) and then locally displaced the impulses in a random fashion. Figure 2.10 shows the resulting novelty curve, the magnitude tempogram, and the PLP curve. As this example indicates, the PLP curve fails to capture such extreme distortions—also note the low confidence values. This is not surprising since our PLP concept relies on the assumption of a locally quasi-periodic behavior of the signal. Using a constrained tempo set $\Theta = [110 : 220]$ (similar effects are obtained by using a context-sensitive smooth tempo trajectory, see Section 2.9.6), one obtains an improved PLP curve as shown in Figure 2.10d. However, note that the quasi-periodically spaced peaks of the PLP curve often deviate from the real pulse positions. In Section 2.9, we will further discuss these issues using romantic piano music as extreme example.

2.8 Iterative Refinement of Local Pulse Estimates

In this section, we indicate how the PLP concept can be applied in an iterative fashion to stabilize local tempo estimations. As example, we consider Brahm's Hungarian Dance No. 5. Figure 2.11 shows a piano reduced score of measures 26-38. The audio recording is an orchestral version conducted by Ormandi. This excerpt is very challenging because of several abrupt changes in tempo. Additionally, the novelty curve is rather noisy because of many weak note onsets played by strings. Figure 2.11a-c show the extracted novelty curve, the tempogram, and the extracted tempo, respectively. Despite of poor note onset

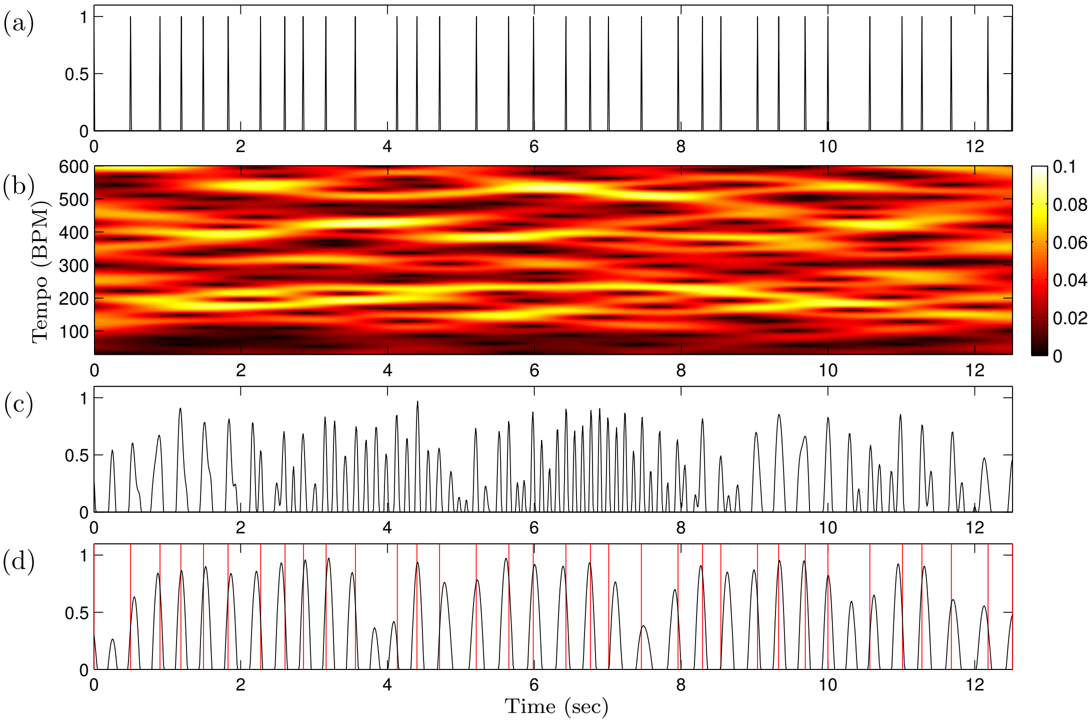


Figure 2.10: Behavior of PLP curves under strong local tempo distortions. **(a)** Impulse train of constant tempo (160 BPM) with random local distortions. **(b)** Magnitude tempogram $|\mathcal{T}|$. **(c)** PLP curve. **(d)** PLP curve for the constrained tempo set $\Theta = [110 : 220]$. Ground-truth pulse positions are indicated by vertical lines.

information, the tempogram correctly captures the predominant eighth note pulse and the tempo for most time positions. A manual inspection reveals that the excerpt starts with a tempo of 180 BPM (measures 26-28, seconds 0-4), then abruptly changes to 280 BPM (measures 29-32, seconds 4-6), and continues with 150 BPM (measures 33-38, seconds 6-18).

Due to the corrupted novelty curve and the rather diffuse tempogram, the extraction of the predominant sinusoidal kernels is problematic. However, accumulating all these kernels leads to an elimination of many of the extraction errors. The peaks of the resulting PLP curve Γ (Figure 2.12a) correctly indicate the musically relevant eighth note pulse positions in the novelty curve. Again, the lower amplitude of Γ in the region of the sudden tempo change indicates the lower confidence in the periodicity estimation. As noted above, PLP curves can be regarded as a periodicity enhancement of the original novelty curve. Based on this observation, we compute a second tempogram now based on the PLP instead of the original novelty curve. Comparing the resulting tempogram (Figure 2.12b) with the original tempogram (Figure 2.11b), one can note a significant cleaning effect, where only the tempo information of the dominant pulse (and its harmonics) is maintained. This example shows how our PLP concept can be used in an iterative framework to stabilize local tempo estimations. Finally, Figure 2.13a shows the manually generated ground truth onsets as well as the resulting tempogram (using the onsets as idealized novelty curve). Comparing the three tempograms and local tempo estimates of Figure 2.11, Figure 2.12,

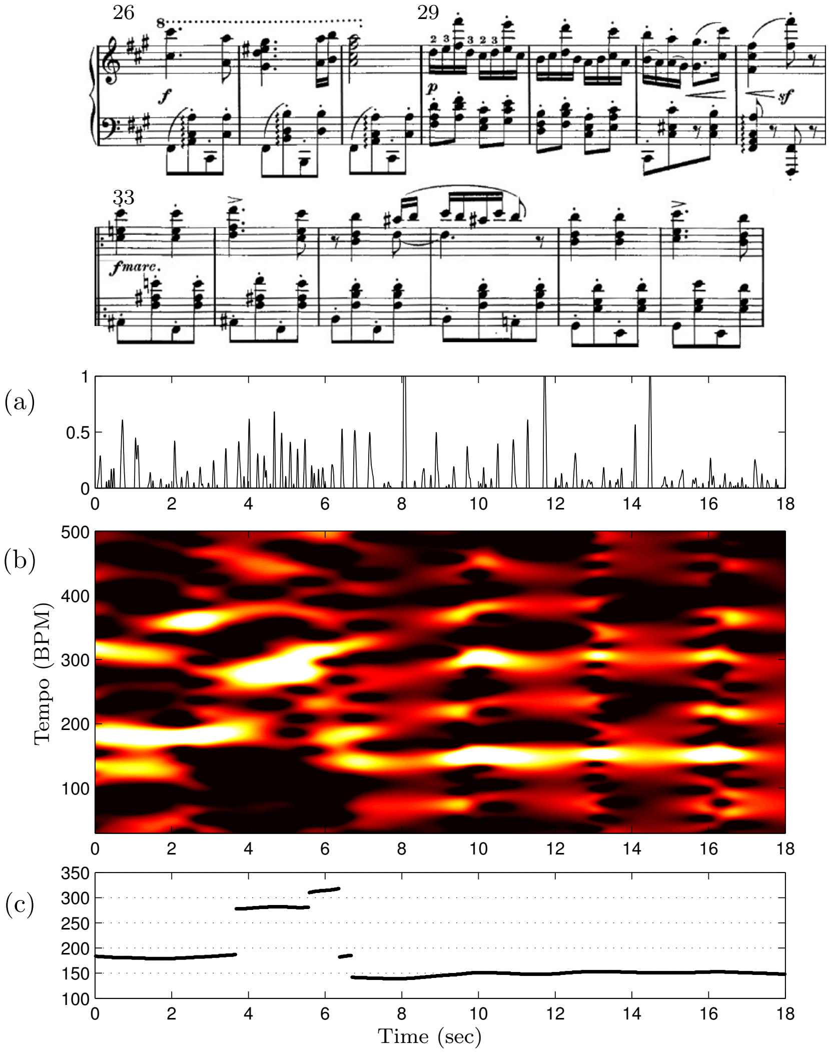


Figure 2.11: Excerpt of an orchestral version conducted by Ormandy of Brahms's Hungarian Dance No. 5. The score shows measures 26 to 38 in a piano reduced version. **(a)** Novelty curve Δ. **(b)** Tempogram derived from Δ using KS = 4 sec. **(c)** Estimated tempo.

and Figure 2.13 again indicates the robustness of PLP curves to noisy input data and outliers.

2.9 Experiments

In the last sections, we have discussed various properties of the PLP concept by means of several challenging music examples. In this section, we report on various experiments to demonstrate how our PLP concept can be applied for improving and stabilizing tempo estimation and beat tracking. We start with describing two baseline experiments in the context of tempo estimation and note onset detection (Section 2.9.1). We then continue

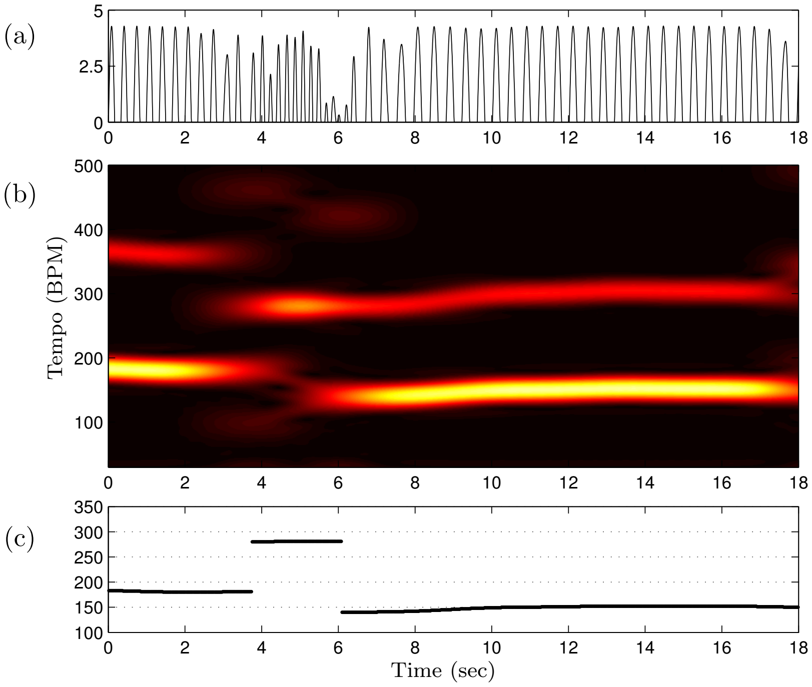


Figure 2.12: PLP concept applied in an iterative fashion for the Brahms example as shown in Figure 2.11. **(a)** PLP curve Γ. **(b)** Tempogram derived from Γ using KS = 4 sec. **(c)** Estimated tempo.

with describing our datasets which consist of real audio material and are used in the subsequent experiments (Section 2.9.2), report on our extensive tempo estimation experiments (Section 2.9.3), and show how our PLP concept can be used to measure the confidence of the estimated tempo values (Section 2.9.4). Subsequently, we address the task of beat tracking, which extends tempo estimation in the sense that it additionally considers the phase of the pulses. In Section 2.9.5, we start by reviewing a state-of-the-art beat tracker used in our experiments. Then we report on various experiments, showing that the combined usage of PLP curves with original novelty information significantly improves beat tracking results (Section 2.9.6). Finally, we introduce a novel beat tracking evaluation measure that considers beats in their temporal context (Section 2.9.7).

2.9.1 Baseline Experiments

Before describing our evaluation based on real audio data, we report on two baseline experiments. Firstly, we describe a baseline experiment using synthesized audio material, where we show that our PLP concept can locally estimate the tempo even in the presence of continuous tempo changes. This extends previous approaches to tempo estimation [47; 121] where often one global tempo for the entire recording is determined and used for the evaluation. Secondly, we describe a baseline experiment to note onset detection, showing that the PLP curves reveal musically meaningful pulse positions.

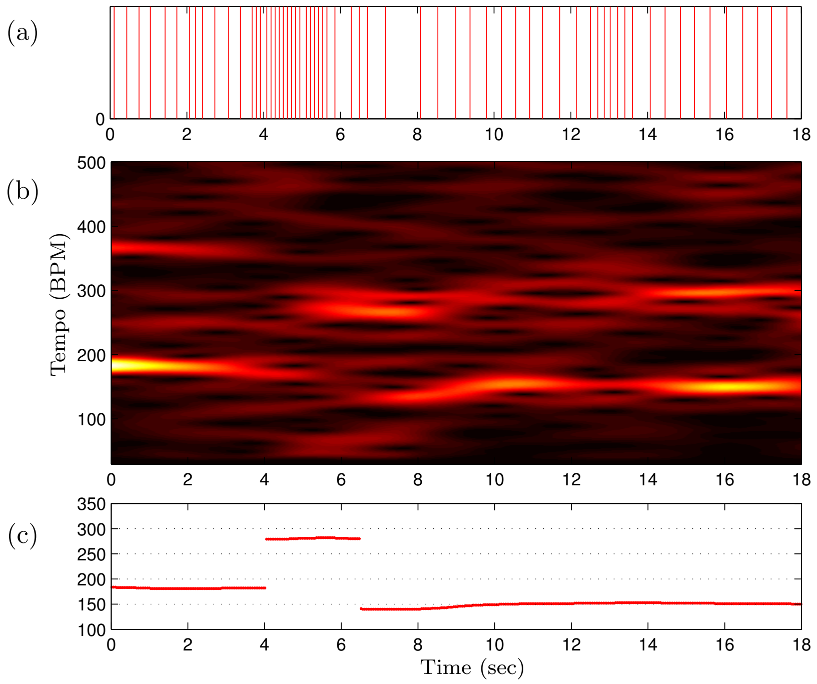


Figure 2.13: Ground truth representation for the Brahms example as shown in Figure 2.11 and Figure 2.12. **(a)** Ground-truth pulses. **(b)** Tempogram derived from these pulses using KS = 4 sec. **(c)** Estimated tempo.

Baseline Experiment to Tempo Estimation

In Section 2.7, we indicated that our PLP concept can handle continuous tempo changes, see also Figure 2.9. We now give a quantitative evaluation to confirm this property. To this end, we use a representative set of ten pieces from the RWC music database [64] consisting of five classical pieces, three jazz, and two popular pieces, see Table 2.1 (first column). The pieces have different instrumentations containing percussive as well as non-percussive passages of high rhythmic complexity. Using the MIDI files supplied by [64], we manually determined the pulse level that dominates the piece (making the simplistic assumption that the predominant pulse does not change throughout the piece) and set the tempo to a constant value with regard to this pulse, see Table 2.1 (second and third columns). The resulting MIDI files are referred to as *original MIDIs*. To simulate continuous tempo changes as implied by accelerandi and ritardandi, we divided the original MIDIs into 20-seconds segments and alternately applied to each segment a continuous speed up or slow down (referred to as *warping procedure*) so that the resulting tempo of the dominant pulse fluctuates between +30% and −30% of the original tempo. The resulting MIDI files are referred to as *distorted MIDIs*. Finally, audio files were generated from the original and distorted MIDIs using a high-quality synthesizer.

To evaluate the tempo extraction capability of our PLP concept, we proceed as follows. Given an original MIDI, let τ denote the tempo and let Θ be the set of integer tempo parameters covering the tempo range of ±40% of the original tempo τ. This coarse tempo range reflects the prior knowledge of the respective pulse level (in this experiment, we do

			original MIDI				distorted MIDI			
Piece	Tempo	Level	4	6	8	12	4	6	8	12
C003	360	1/16	74.5	81.6	83.7	85.4	73.9	81.1	83.3	86.2
C015	320	1/16	71.4	78.5	82.5	89.2	61.8	67.3	71.2	76.0
C022	240	1/8	95.9	100.0	100.0	100.0	95.0	98.1	99.4	89.2
C025	240	1/16	99.6	100.0	100.0	100.0	99.6	100.0	100.0	96.2
C044	180	1/8	95.7	100.0	100.0	100.0	82.6	85.4	77.4	59.8
J001	300	1/16	43.1	54.0	60.6	67.4	37.8	48.4	52.7	52.7
J038	360	1/12	98.6	99.7	100.0	100.0	99.2	99.8	100.0	96.7
J041	315	1/12	97.4	98.4	99.2	99.7	95.8	96.6	97.1	95.5
P031	260	1/8	92.2	93.0	93.6	94.7	92.7	93.7	93.9	93.5
P093	180	1/8	97.4	100.0	100.0	100.0	96.4	100.0	100.0	100.0
average:			86.6	90.5	92.0	93.6	83.5	87.1	87.5	84.6
average (after iteration):			89.2	92.0	93.0	95.2	86.0	88.8	88.5	83.1

Table 2.1: Percentage of correctly estimated local tempi using original MIDI files (constant tempo) and distorted MIDI files for different kernel sizes KS $= 4, 6, 8, 12$ sec.

not want to deal with tempo octave confusions) and comprises the tempo values of the distorted MIDI. Based on Θ, we compute for each time position t the maximizing tempo parameter $\tau_t \in \Theta$ as defined in Eq. (2.4) for the original MIDI using various kernel sizes. We consider the local tempo estimate τ_t *correct*, if it falls within a 2% deviation of the original tempo τ. The left part of Table 2.1 shows the percentage of correctly estimated local tempi for each piece. Note that, even having a constant tempo, there are time positions with incorrect tempo estimates. Here, one reason is that for certain passages the pulse level or the onset information is not suited or simply not sufficient for yielding good local tempo estimations, e. g., caused by musical rests or local rhythmic offsets. For example, for the piece C003 (Beethoven's Fifth), the tempo estimation is correct for 74.5% of the time parameters when using a kernel size (KS) of 4 sec. Assuming a constant tempo, it is not surprising that the tempo estimation stabilizes when using a longer kernel. In case of C003, the percentage increases to 85.4% for KS = 12 sec.

In any case, the tempo estimates for the original MIDIs with constant tempo only serve as reference values for the second part of our experiment. Using the distorted MIDIs, we again compute the maximizing tempo parameter $\tau_t \in \Theta$ for each time position. Now, these values are compared to the time-dependent distorted tempo values that can be determined from the warping procedure. Analogous to the left part, the right part of Table 2.1 shows the percentage of correctly estimated local tempi for the distorted case. The crucial point is that even when using the distorted MIDIs, the quality of the tempo estimations only slightly decreases. For example, in the case of C003, the tempo estimation is correct for 73.9% of the time parameters when using a kernel size of 4 sec (compared to 74.5% in the original case). Averaging over all pieces, the percentage decreases from 86.6% (original MIDIs) to 83.5% (distorted MIDIs), for KS = 4 sec. This clearly demonstrates that our concept allows for capturing even significant tempo changes. As mentioned above, using longer kernels naturally stabilizes the tempo estimation in the case of constant tempo. This, however, does not hold when having music with constantly changing tempo. For example, looking at the results for the distorted MIDI of C044 (Rimski-Korsakov, The Flight of the Bumble Bee), we can note a drop from 82.6% (4 sec kernel) to 59.8% (12 sec kernel).

Furthermore, we investigated the iterative approach already sketched for the Brahms example in Section 2.8. Here, we use the PLP curve as basis for computing a second tempogram from which the tempo estimates are derived. As indicated by the last line of Table 2.1, this iteration indeed yields an improvement for the tempo estimation for the original as well as the distorted MIDI files. For example, in the distorted case with KS = 4 sec the estimation rate raises from 83.5% (tempogram based on Δ) to 86.0% (tempogram based on Γ).

Baseline Experiment to Onset Detection

As discussed in Section 2.7, a PLP curve can be seen as a local periodicity enhancement of the original novelty curve where the peaks of the PLP curve indicate likely pulse positions. We now describe an application of our PLP concept to the task of onset detection in order to evaluate the quality of the PLP peaks.

Recall from Section 2.1 that most approaches for onset detection proceed in two steps. First, a novelty curve Δ is extracted from the given music signal. Then, to determine the note onsets, one tries to locate the positions of relevant peaks of Δ. For music with soft note onsets or strong fluctuation such as vibrato, however, the discrimination of relevant and spurious peaks becomes nearly impossible. Here, additional model assumptions on the rhythmic nature of the music signal may be exploited to support note onset detection. The musical motivation is that the periodic structure of notes plays an important role in the sensation of note onsets. In particular, weak note onsets may only be perceptible within a rhythmic context. For example, in [34] a global rhythmic structure is determined in terms of IOI statistics of previously extracted note onset candidates. Then, assuming constant tempo, this structure is used to determine the relevant onset positions.

Following these lines, we also exploit the quasi-periodic nature of note onsets in our PLP-based enhancement strategy. However, in our approach, we avoid the fragile peak picking step at an early stage. Furthermore, we do not presuppose a global rhythmic structure but only assume local periodicity thus allowing tempo changes. More precisely, given a novelty curve Δ, we compute a PLP curve Γ. Being local quasi-periodic, the peak positions of Γ define a grid of pulse positions where note onset positions are likely to occur. Furthermore, the peak structure of Γ is much more pronounced than in Δ. Actually, the peaks are simply the local maxima and peak picking becomes a straightforward task. Assuming that note onsets are likely to lie on the PLP-defined grid, we select all peak positions of Γ as detected onsets.

We compare the pulse positions obtained from the PLP-curve with the onsets extracted from the novelty curve using a peak picking strategy [7]. In the experiment, we use two evaluation datasets containing audio recordings along with manually labeled onset positions used as reference onsets. The first dataset is publicly available [113] and has been used in the evaluation of various onset detection algorithms, e.g., [189]. This dataset, in the following referred to as `PUBLIC`, consists of 242 seconds of audio (17 music excerpts of different genre) with 671 labeled onsets. The second dataset particularly contains classical music with soft onsets and significant tempo changes. This dataset, in the following referred to as `PRIVATE`, consists of 201 seconds of audio with 569 manually labeled onsets.

		PUBLIC			PRIVATE		
Curve	KS	P	R	F	P	R	F
Δ		0.783	**0.821**	0.793	0.694	**0.732**	0.698
Γ	4 sec	0.591	**0.933**	0.695	0.588	**0.913**	0.679
Γ	6 sec	0.599	**0.955**	0.705	0.599	**0.907**	0.689
Γ	8 sec	0.597	**0.944**	0.701	0.588	**0.877**	0.674

Table 2.2: Mean precision P, recall R, and F-measure F values for the onset detection task using the novelty curve Δ and PLP curves Γ of kernel sizes $\mathrm{KS} = 4, 6, 8$ sec.

Following the MIREX 2011 Audio Onset Detection evaluation procedure[1], each reference onset is considered a *correct detection* (CD) if there is a detected onset within an error tolerance of 50 ms, otherwise a *false negative* (FN). Each detected onset outside of all tolerance regions is called a *false positive* (FP). The corresponding number of onsets is denoted NCD, NFN, and NFP, respectively. From this one obtains the precision, recall, and F-measure defined by

$$\mathrm{P} = \frac{\mathrm{NCD}}{\mathrm{NCD} + \mathrm{NFP}}, \ \mathrm{R} = \frac{\mathrm{NCD}}{\mathrm{NCD} + \mathrm{NFN}}, \ \mathrm{F} = \frac{2 \cdot \mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}} \ . \tag{2.8}$$

These values are computed separately for each piece. The final values are obtained by averaging over all pieces of the respective dataset.

Table 2.2 shows the resulting average P, R, and F values for the original novelty curve Δ as well as for the PLP curve Γ using periodicity kernels of different sizes. As the results show, our PLP-concept indeed reveals musically meaningful pulse positions. For example, using the PLP curve Γ with a kernel size 4 seconds instead of the original novelty curve Δ, the mean recall R increases from 0.821 to 0.933 for the PUBLIC set and from 0.732 to 0.913 for the PRIVATE set. This shows that a vast majority of the relevant note onsets indeed lie on the PLP-defined pulse grid. Especially for the PRIVATE set, the PLP curve allows for inferring a large number of soft note onsets that are missed when using the original novelty curve. On the other side, the precision values for Γ are lower than those for Δ. This is not surprising, since in our PLP-based approach we select all peak positions of Γ. Even though most note onsets fall on PLP peaks, not all PLP peaks necessarily correspond to note onsets. For example, in the Shostakovich example shown in Figure 2.6, the PLP curve infers three onset positions for measures 31 and 35, respectively. In these measures, however, the second beats correspond to rests without any note onsets. Similarly, in our Beethoven example shown in Figure 2.8d, the fermata passage is periodically filled with non-relevant pulses. On the other hand, all relevant onsets in the soft piano section (measures 8-13) are identified correctly. These string onsets can not be recovered correctly using the original novelty curve. This experiment indicates that our PLP curve indeed reveals musically meaningful pulse positions. As a consequence, our concept allows for recovering soft note onsets.

Finally, we look at the influence of the kernel size on the detection quality. Here, note that most of the excerpts in the PUBLIC dataset have a constant tempo. Therefore, using a kernel size of 6 seconds instead of 4 seconds, the kernel estimation is more robust leading to an increase of recall (from $R = 0.933$ to $R = 0.955$). Contrary, the PRIVATE dataset

[1] http://www.music-ir.org/mirex/wiki/2011:Audio_Onset_Detection

	Audio	Length	Beats	Unannotated	Mean Tempo	Std. Tempo
Dataset	[#]	[sec]	[#]	[sec]	[BPM]	[%]
BEATLES	179	28831	52729	1422	116.7	3.3
RWC-POP	100	24406	43659	0	111.7	1.1
RWC-JAZZ	50	13434	19021	0	89.7	4.5
RWC-CLASSIC	61	19741	32733	725	104.8	15.2
MAZURKA	298	45177	85163	1462	126.0	24.6

Table 2.3: The five beat-annotated datasets used in our experiments. The first four columns indicate the name, the number of audio recordings, the total length, and the total number of annotated beat positions.

contains classic music with many tempo changes. Here, kernels of smaller sizes are better suited for adjusting the local periodicity estimations to the changing tempo.

2.9.2 Audio Datasets

For our subsequent experiments and evaluations, we use five different datasets that consists of real audio recordings (opposed to the synthesized audio material used in Section 2.9.1) and comprise music of various genres and complexities. For all audio recordings, manually generated beat annotations are available. The first collection BEATLES consists of the 12 studio albums by "The Beatles" containing a total number of 179 recordings[2] of Rock/Pop music [119]. Furthermore, we use audio recordings from the RWC Music Database [64], which consists of subcollections of different genres. From this database, we use the three subcollections RWC-POP, RWC-CLASSIC, and RWC-JAZZ containing a total number of 211 recordings. Our fifth dataset MAZURKA consists of piano recordings taken from a collection of 2700 recorded performances for the 49 Mazurkas by Frédéric Chopin. These recordings were collected in the Mazurka Project[3]. For 298 of the 2700 recordings, manually generated beat annotations exist, which have been previously used for the purpose of performance analysis [152]. The dataset MAZURKA consists of exactly these 298 recordings (corresponding to five of the 49 Mazurkas).

Table 2.3 gives an overview of the five different datasets. The first four columns of the table indicate the name of the dataset, the number of contained audio recordings, the total length of all audio recordings, and the total number of annotated beat positions. Some recordings contain passages where no meaningful notion of a beat is perceivable. For example, the datasets MAZURKA and RWC-CLASSIC contain some audio files with long passages of silence. Furthermore, in BEATLES, some songs contain noise-like improvisational passages where the musicians refrain from following any rhythmic pattern. All these passages have not been annotated and are left unconsidered in our evaluation (if not stated otherwise). The fifth column (Unannotated) of Table 2.3 indicates the total length of the unannotated passages.

From the beat positions, one can directly derive the local tempo given in beats per minute

[2] Actually, there are 180 songs, but for the song "Revolution 9" no annotations were available. This song is a collage of vocal and music sound clips without any meaningful notion of a beat.

[3] http://mazurka.org.uk/

Dataset	4 sec	6 sec	8 sec	12 sec
`BEATLES`	94.1	95.4	95.9	96.3
`RWC-POP`	95.3	96.7	97.3	98.0
`RWC-JAZZ`	81.8	85.4	86.6	87.2
`RWC-CLASSIC`	70.4	70.9	70.3	68.7
`MAZURKA`	44.5	40.1	37.3	34.3

Table 2.4: Percentage of correctly estimated local tempi (±4% tolerance) for the five datasets using the kernel sizes $\mathrm{KS} = 4, 6, 8, 12$ sec.

(BPM). The last two columns of Table 2.3 indicate the piecewise mean tempo (in BPM) and standard deviation (in percent) averaged over all recordings of the respective dataset. Note that popular music is often played with constant tempo. This is also indicated by the small values for the standard deviation (e. g., 1.1% for `RWC-POP`). In contrast, classical music often reveals significant tempo changes, which is indicated by higher values for the standard deviation (e. g., 15.2% for `RWC-CLASSIC`). These changes can be abrupt as a result of a changing tempo marking (e. g., from *Andante* to *Allegro*) or continuous as indicated by tempo marks such as ritardando or accelerando. Another source for tempo changes is the artistic freedom a musician often takes when interpreting a piece of music. In particular, for romantic piano music such as the Chopin Mazurkas, the tempo consistently and significantly changes from one beat to the next, resulting in pulse sequences similar to the one shown in Figure 2.10a.

2.9.3 Tempo Estimation Experiments

Continuing the evaluation of Section 2.9.1, we now analyze the tempo estimation capability of our approach on the basis of real audio recordings. To this end, we generate a reference tempo curve for each audio recording of our datasets from the available beat annotations. Here, we first compute the local tempo on the quarter-note level, which is determined by the given inter-beat intervals. The regions before the first beat and after the last beat are left unconsidered in the evaluation. As the tempo values on such a fine temporal level tend to be too noisy, we further smooth the resulting tempo values by considering for each time position the averaged tempo over a range of three consecutive beat intervals. Using the same sampled time axis $[1 : T]$ as in Section 2.4, we obtain a tempo curve $\tau^{\mathrm{R}} : [1 : T] \to \mathbb{R}_{\geq 0}$ that encodes the local reference tempo for each time position. Now, for each time position t, we compute the maximizing tempo parameter $\tau_t \in \Theta$ as defined in Eq. (2.4). Leaving the problem of tempo octave confusion unconsidered, we say that an estimated local tempo τ_t is *correct*, if it falls within ±4% of an integer multiple[4] $k \in [1, 2, \ldots, 5, 6]$ of the reference tempo $\tau^{\mathrm{R}}(t)$. Here, we choose a tolerance of ±4% as used in [47]. For each recording, we then compute the percentage of correctly estimated tempi and average these values over all recordings of a given dataset.

[4] In general, confusion with integer fractions $k \in [1/2, 1/3, 1/4, \ldots]$ of the tempo may occur, too. However, it can be shown that Fourier-based tempograms (as opposed to, e. g. autocorrelation-based tempograms) respond to tempo harmonics (integer multiples) but suppress tempo subharmonics (integer fractions), see Chapter 4. Since we use Fourier-based tempograms, we only consider confusion with tempo harmonics.

Table 2.4 shows the evaluation results of the local tempo estimation for the five datasets and for different kernel sizes. For popular music, one generally obtains high estimation rates, e. g., an average rate of 94.1% for `BEATLES` and 95.3% for `RWC-POP` when using a kernel size (KS) of 4 seconds. Having constant tempo for most parts, the rates even increase when using longer kernel sizes. For the `RWC-JAZZ` dataset, the rate is 81.8% (KS = 4 sec). This lower rate is partly due to passages with soft onsets and complex rhythmic patterns. Using longer kernels, the tempo can be correctly identified even for some of these passages leading to a significantly higher rate of 87.2% (KS = 12 sec). The situation becomes more complex for classical music, where one has much lower rates, e. g., 70.4% (KS = 4 sec) for `RWC-CLASSIC`. Here, a manual inspection reveals two major reasons leading to degradations in the estimation rates. The first reason is again the existence of passages with soft onsets—here, longer kernel sizes help in stabilizing the tempo estimation. The second reason is that for many recordings of classical music one has significant local tempo fluctuation caused by the artistic freedom a musician takes. In such passages, the model assumption of local quasi-periodicity is strongly violated—even within a window of 4 seconds the tempo may significantly change by more than 50% percent, see also Figure 2.10. Here, it is difficult for the local periodicity kernels to capture meaningful periodic behavior. For such passages, increasing the kernel size has a negative effect on the tempo estimation. In other words, the increase of the kernel size is beneficial for the first type of degradation and detrimental for the second type of degradation. For `RWC-CLASSIC`, these two effects neutralize each other yielding similar estimation rates for all kernel sizes. However, for the `MAZURKA` dataset, one mainly has to deal with degradations of the second type. Containing highly expressive romantic piano music, the estimation rate is 44.5% when using KS = 4 sec. The rate becomes even worse when increasing the kernel size, e. g., 34.3% for KS = 12 sec. This type of music reveals the limitations of a purely onset-based tempo estimation approach—actually, for such music the notion of local tempo becomes problematic even from a musical point of view, see Section 2.9.4 for a continuation of this discussion.

2.9.4 Confidence and Limitations

The results for the local tempo estimation significantly degrade in the case that the assumption of local quasi-periodicity is violated. We now show how the PLP concept allows for detecting such problematic passages automatically. As mentioned in Section 2.7, constructive and destructive interference phenomena in the overlap-add synthesis influence the amplitude of the resulting PLP curve $\Gamma : [1 : T] \to [0, 1]$. Locally consistent tempo estimations result in amplitude values for the peaks close to one, whereas inconsistent kernel estimations result in lower values. We now exploit this property of Γ to derive a confidence measure for the tempo estimation. To this end, we fix a confidence threshold $\theta \in [0, 1]$ and a length parameter λ. Then, a time interval $I \subseteq [1 : T]$ of length λ is called *reliable* if *all* peaks (local maxima) of Γ positioned in I have a value above θ, otherwise I is called *unreliable*. The idea is that when I contains at least one peak of lower amplitude, there are inconsistent kernel estimates that make a tempo estimation in I unreliable. Finally, we define the subset $I(\theta, \lambda) \subseteq [1 : T]$ to be the union of all reliable intervals of length λ.

We show that $I(\theta, \lambda)$ indeed corresponds to passages yielding reliable tempo estimates by

Database	0.95	0.90	0.80	0.70	0 (All)
BEATLES	98.5 (59.4)	98.1 (62.9)	97.5 (64.3)	97.2 (66.2)	89.0 (100)
RWC-POP	99.5 (66.5)	99.2 (67.2)	99.1 (69.3)	98.8 (72.5)	92.8 (100)
RWC-JAZZ	94.2 (35.0)	91.4 (40.0)	89.8 (43.8)	89.6 (47.4)	79.0 (100)
RWC-CLASSIC	89.4 (31.4)	84.7 (38.5)	82.4 (43.7)	81.8 (47.1)	67.6 (100)
MAZURKA	74.1 (6.4)	69.2 (11.8)	65.6 (17.8)	62.4 (22.0)	42.0 (100)

Table 2.5: Percentage of correctly estimated local tempi for the five datasets using restricted regions $I(\theta, \lambda)$. The parameters are $\lambda = \mathrm{KS} = 4$ sec and $\theta = 0.95, 0.90, 0.80, 0.70$. The unrestricted case (last column) corresponds to $\theta = 0$. The relative size of $I(\theta, \lambda)$ (in percent) is specified in parentheses.

conducting experiments based on the five datasets of Table 2.3. This time, we include all time positions in the evaluation, even the previously excluded regions without any beat annotations and the regions before the first and after the last beats. Since no meaningful tempo can be assigned to these regions, all estimates within these regions are considered wrong in the evaluation. Here, our motivation is that these regions should automatically be classified as unreliable. The last column of Table 2.5 shows the estimation rates using a kernel size of 4 sec. Naturally, including unannotated regions, the rates are lower compared to the ones reported in the first column of Table 2.4. For example, for the dataset BEATLES, one now has a rate of 89.0% instead of 94.1%.

In our experiments, we use an interval length of $\lambda = 4$ sec corresponding to the kernel size $\mathrm{KS} = 4$ sec. We then compute $I(\theta, \lambda)$ for a fixed threshold θ and evaluate the tempo estimates only on the restricted region $I(\theta, \lambda) \subseteq [1 : T]$. Table 2.5 shows the percentages of correctly estimated local tempi within $I(\theta, \lambda)$ for various thresholds $\theta \in \{0.95, 0.9, 0.8, 0.7\}$. Furthermore, the size of $I(\theta, \lambda)$ relative to $[1 : T]$ is indicated in parentheses (given in percent). For example, for the dataset BEATLES, the restricted region $I(\theta, \lambda)$ with $\theta = 0.95$ covers in average 59.4% of all time positions, while the estimation rate amounts to 98.5%. Lowering the threshold θ, the region $I(\theta, \lambda)$ increases, while the estimation rate decreases. The values of the last column can be seen as the special case $\theta = 0$ resulting in the unrestricted case $I(\theta, \lambda) = [1 : T]$.

Also, for the other datasets, the estimation rates significantly improve when using the restricted region $I(\theta, \lambda)$. In particular, for RWC-POP, the estimation error drops to less than one percent when using $\theta \geq 0.8$, while still covering more than two thirds of all time positions. Actually, for popular music, most of the unreliable regions result from pulse level changes (see Figure 2.6f) rather than poor tempo estimates. Also, for the classical music dataset RWC-CLASSIC, the estimation rates increase significantly reaching 89.4% for $\theta = 0.95$. However, in this case the restricted regions only cover one third (31.4%) of the time positions. This is even worse for the MAZURKA dataset, where only 6.4% are left when using $\theta = 0.95$. Figure 2.14 illustrates one main problem that arises when dealing with highly expressive music where the assumption of local quasi-periodicity is often violated. The passage shows significant tempo fluctuations of the interpretation of the Mazurka Op. 30-2 as indicated by the reference tempo curve τ^{R} in Figure 2.14a. Indeed, the PLP curve allows for detecting regions of locally consistent tempo estimates (indicated by the thick blue lines in Figure 2.14b). For these regions the local tempo estimates largely overlap with the reference tempo, see Figure 2.14a.

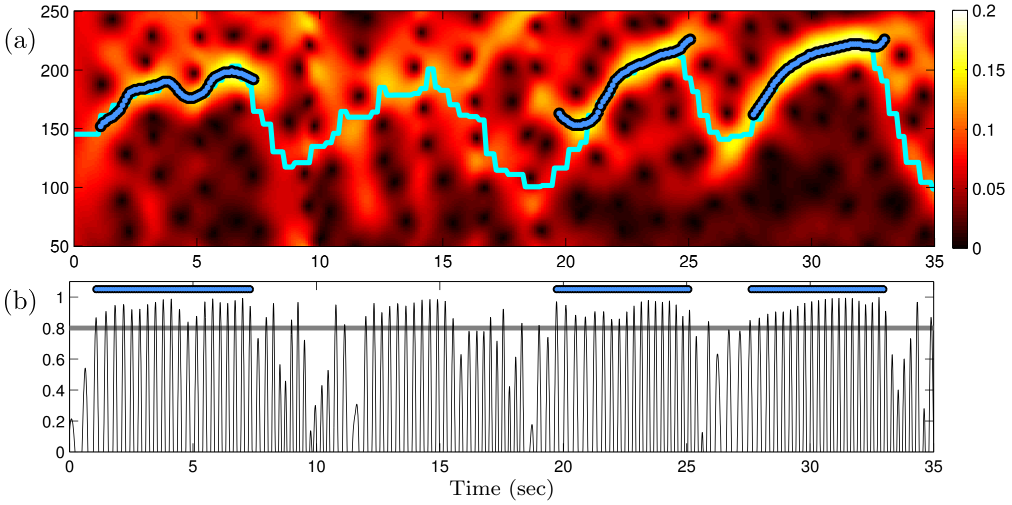


Figure 2.14: Tempo estimation for a highly expressive recording (pid9065-14) of Chopin's Mazurka Op30-2. **(a)** Magnitude tempogram $|\mathcal{T}|$ of the first 35 seconds using $\Theta = [50 : 250]$ with the reference tempo τ^{R} (cyan) and the tempo estimates (thick blue) on $I(\theta, \lambda)$. **(b)** PLP curve and restricted region $I(\theta, \lambda)$ (blue) for $\theta = 0.8$ and $\lambda = 4$ sec.

2.9.5 Dynamic Programming Beat Tracking

In the following, we summarize the state-of-the-art beat tracking procedure as introduced in [44]. This procedure is used in the following beat-tracking experiments. The input of the algorithm consists of a novelty-like function $\Lambda : [1 : T] \to \mathbb{R}$ (indicating note onset positions) as well as a number $\rho \in \mathbb{Z}$ that yields an estimate of a global (average) beat period $\rho \in \mathbb{Z}$. Assuming a roughly constant tempo, the difference δ of two neighboring beats should be close to ρ. To measure the distance between δ and ρ, a neighborhood function $N_\rho : \mathbb{N} \to \mathbb{R}$

$$N_\rho(\delta) := -\big(\log_2(\delta/\rho)\big)^2$$

is introduced. This function takes the maximum value of 0 for $\delta = \rho$ and is symmetric on a log-time axis. Now, the task is to estimate a sequence $B = (b_1, b_2, \ldots, b_K)$, for some suitable $K \in \mathbb{N}$, of monotonously increasing beat positions $b_k \in [1 : T]$ satisfying two conditions. On the one hand, the value $\Lambda(b_k)$ should be large for all $k \in [1 : K]$, and, on the other hand, the beat intervals $\delta = b_k - b_{k-1}$ should be close to ρ. To this end, one defines the score $S(B)$ of a beat sequence $B = (b_1, b_2, \ldots, b_K)$ by

$$S(B) = \sum_{k=1}^{K} \Lambda(b_k) + \alpha \sum_{k=2}^{K} N_\rho(b_k - b_{k-1}), \tag{2.9}$$

where the weight $\alpha \in \mathbb{R}$ balances out the two conditions. In our experiments, $\alpha = 5$ turned out to yield a suitable trade-off. Finally, the beat sequence maximizing S yields the solution of the beat tracking problem. The score-maximizing beat sequence can be obtained by a straightforward dynamic programming (DP) approach, see [44] for details. Therefore, in the following, we refer to this procedure as *DP beat tracking*.

	Peak Picking				DP Beat Tracking		
Dataset	Δ	Γ	$\Gamma^{\pm 40}$	Γ^{DP}	Δ	Γ	Ψ
BEATLES	0.619	0.593	0.671	0.663	0.826	0.741	**0.861**
RWC-POP	0.554	0.507	0.610	0.579	0.786	0.752	**0.819**
RWC-JAZZ	0.453	0.411	0.407	0.407	0.514	**0.573**	0.533
RWC-CLASSIC	0.532	0.514	0.521	0.528	0.618	0.609	**0.644**
MAZURKA	**0.757**	0.618	0.731	0.685	0.641	0.651	0.684

Table 2.6: Average F-measures for various beat tracking approaches using an error tolerance of 70 ms.

2.9.6 Beat Tracking Experiments

We now report on various beat tracking experiments conducted on the five audio datasets described in Section 2.9.2. We consider two different approaches to beat tracking. In the first approach, which serves as baseline, we simply perform peak picking based on an adaptive thresholding strategy [7] and define the beat positions to be the detected peak positions. In the second approach, we use the DP beat tracking procedure summarized in Section 2.9.5.

For each of these two approaches, we compare the beat tracking results for five different curves using the original novelty curve Δ, the PLP curve Γ, a constrained PLP curve $\Gamma^{\pm 40}$, a PLP curve Γ^{DP} based on a smooth tempo curve, as well as a combined novelty/PLP curve denoted by Ψ. Here, the PLP curve Γ is computed using $\Theta = [30 : 600]$ and $\mathrm{KS} = 4$ sec). For the constrained PLP curve $\Gamma^{\pm 40}$, we use a tempo set covering ±40% of the mean tempo of the audio recording, where we assume that a rough estimate of this tempo is given. The curve Γ^{DP} is obtained by first computing a smoothed tempo trajectory based on dynamic programming as described in [3; 187] and then by using these tempo values in the PLP computation instead of the maximizing values, cf. Eq. (2.4). Finally, the combined curve Ψ is defined as $\Psi = (\Delta^{\mathrm{norm}} + \Gamma)/2$, where Δ^{norm} denotes a locally normalized version of Δ that assumes values in the interval $[0, 1]$ (as the PLP curve). The normalization is obtained using a sliding maximum filter of length 4 sec (as for the kernel size).

In a first evaluation, we use the same F-measure as for onset detection experiment (Section 2.9.1). A reference beat is considered a correct detection if there is a detected beat within an error tolerance of 70 ms. The same tolerance value is suggested in the literature [29] and used in the MIREX 2011 Audio Beat Tracking evaluation procedure[5]. Then, precision, recall, and F-measure are defined as in Eq. (2.8).

Table 2.6 shows the F-measure values for both beat tracking approaches in combination with different curves. Using peak picking based on Δ one obtains an F-measure of $\mathrm{F} = 0.619$ for the dataset BEATLES. Actually, for most music, beat positions go along with onset positions. Consequently, onset positions typically lie on beat positions or on positions corresponding to higher pulse levels. Therefore, even the simple onset detection procedure already yields reasonable F-measures (resulting from a very high recall and a moderate precision). At first sight, it may be surprising that when using peak picking on Γ, one

[5] http://www.music-ir.org/mirex/wiki/2011:Audio_Beat_Tracking

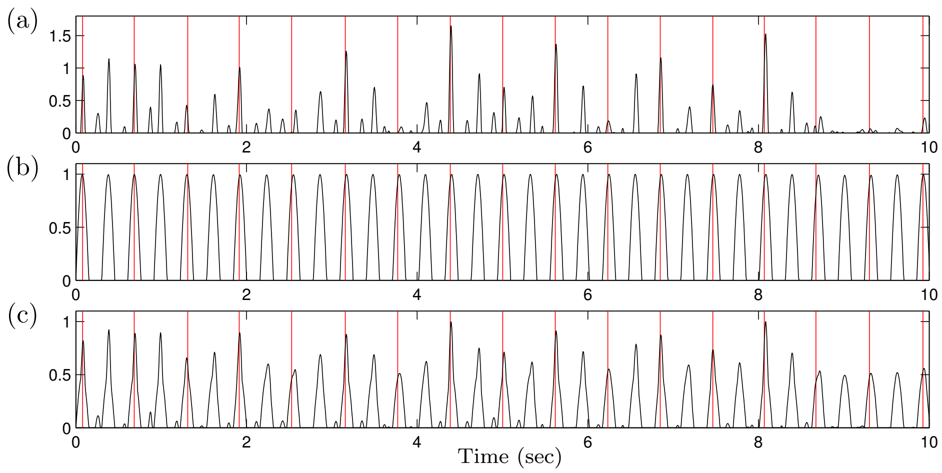


Figure 2.15: Illustration of the different curves used in the beat tracking experiments with ground truth beat positions shown as vertical lines. **(a)** Novelty curve Δ. **(b)** PLP curve Γ. **(c)** Combined curve Ψ.

obtains slightly lower F-measure values (e. g. F = 0.593 for BEATLES). Here, note that the peak positions of Γ define a locally periodic pulse grid, where beat positions are likely to occur. As our experiments show, the number of false negatives is reduced in comparison with Δ (leading to a higher recall). However, not all PLP peaks necessarily correspond to beats. Typically, the predominant pulse corresponds to the tatum pulse leading to many false positives (low precision). The situation already improves when using $\Gamma^{\pm 40}$. Constraining the pulse to the correct pulse level reduces the number of false positives (e. g. F = 0.671 for BEATLES). Employing a smooth tempo trajectory for computing Γ^{DP} has a very similar effect (e. g. F = 0.663 for Beatles).

Using the DP beat tracker, the F-measures significantly improve. In general, the best results are achieved when using the DP beat tracker with the combined curve Ψ. In particular, Ψ leads to better results than the usual approach exclusively based on Δ. For example, in the case of BEATLES, the F-measure increases from F = 0.826 for Δ to F = 0.861 for Ψ. Using Γ alone, the results seem to degrade (F = 0.741). A manual investigation shows that the PLP curve robustly provides information about likely pulse positions typically at the tatum level, whereas the beat positions correspond to the tactus level. This often results in half-beat shifts (typically one period on the tatum level) in the beat tracking result. In other words, since the peak values are invariant to dynamics, Γ is generally not capable of discriminating between on-beats and off-beats, see Figure 2.15b. The problem of half-beat shifts is not as prominent when using the original novelty curve, since the onset peaks of Δ on the on-beat positions are often more pronounced than the ones on the off-beat positions, see Figure 2.15a. However, the novelty curve Δ often reveals passages with noisy and missing peaks. Here, the combination Ψ inherits the robustness from Γ and the discriminative power from Δ, yielding the best overall beat tracking results, see last column of Table 2.6. Figure 2.15c illustrates the gain achieved through the combined usage of Γ and Δ.

The evaluation based on the simple F-measure has several weaknesses. First, even completely random false positives only slightly degrade the F-measure. Here, one reason is that the F-measure only moderately punishes peak positions, even if they are not musically meaningful. As a consequence, simple peak picking on Δ, even though ignoring any notion of a beat concept, seems to yield good results. In particular for the MAZURKA dataset, a peak picking based on Δ seems to outperform all other strategies ($\mathrm{F} = 0.757$). Furthermore, this evaluation measure does not account for the issue of half-beat shifts. Finally, evaluating the beats individually, the temporal context of beat tracking is ignored. In Section 2.9.7, we tackle these problems by introducing a novel context-sensitive evaluation measure.

2.9.7 Context-Sensitive Evaluation

In the evaluation measure considered so far, the beat positions were evaluated one by one. However, when tapping to the beat of music a listener obviously requires the temporal context of several consecutive beats. Therefore, in evaluating beat tracking procedures, it seems natural to consider beats in the temporal context instead of looking at the beat positions individually [29]. To account for these temporal dependencies, we now introduce a context-sensitive evaluation measure. Let $R = (r_1, r_2, \ldots, r_M)$ be the sequence of monotonously increasing reference beat positions $r_m \in [1 : T]$, $m \in [1 : M]$. Similarly, let $B = (b_1, b_2, \ldots, b_K)$ be the sequence of monotonously increasing detected beat positions $b_k \in [1 : T]$, $k \in [1 : K]$. Furthermore, let $L \in \mathbb{N}$ be a parameter that specifies the temporal context measured in beats, and let ε be the error tolerance corresponding to 70 milliseconds. Then a reference beat r_m is considered an *L-correct detection*, if there exists a subsequence $r_j, \ldots, r_{j+L-1}$ of R containing r_m (i. e. $m \in [j : j + L - 1]$) as well as a subsequence $b_i, \ldots, b_{i+L-1}$ of B such that

$$|r_{j+\ell} - b_{i+\ell}| \leq \varepsilon$$

for all $\ell \in [0 : L - 1]$. Intuitively, for a beat being considered L-correct, one requires an entire track consisting of L consecutive detected beats that match (up to the error tolerance ε) to a track of L consecutive reference beats. Here, a single outlier in the detected beats already destroys this property. Let M^L be the number of L-correct references beats. Then, we define the context-sensitive recall $\mathrm{R}^L := M^L/M$, precision $\mathrm{P}^L := M^L/K$ and F-measure $\mathrm{F}^L := (2 \cdot \mathrm{P}^L \cdot \mathrm{R}^L)/(\mathrm{P}^L + \mathrm{R}^L)$.

In our evaluation, we use the parameter $L = 4$ corresponding to four consecutive beats (roughly a measure). Table 2.7a shows the resulting context-sensitive F^L-measures for the same experiments as described in the last section. Now, the weaknesses of a simple peak picking strategy based on Δ or Γ become obvious. Compared to the previous F-measure (cf. Table 2.6), the F^L-measures drop significantly for all datasets. In particular for popular music, these measures are close to zero (e. g. $\mathrm{F}^L = 0.015$ for RWC-POP and Δ), which indicates that basically no four consecutive beats are detected without any intervening spurious peaks. Actually, the situation already improves significantly when using the constrained PLP curve $\Gamma^{\pm 40}$ (e. g. $\mathrm{F}^L = 0.486$ for RWC-POP). This shows that the PLP curve captures meaningful local beat information when restricted to the desired pulse level. Γ^{DP} once again obtains similar results. For example, in the case of BEATLES $\mathrm{F}^L = 0.554$ for $\Gamma^{\pm 40}$ and $\mathrm{F}^L = 0.555$ for Γ^{DP}. For MAZURKA, however, exhibiting many abrupt tempo changes, Γ^{DP} leads to lower F^L-measures ($\mathrm{F}^L = 0.484$) than $\Gamma^{\pm 40}$ ($\mathrm{F}^L = 0.539$). Here,

(a)

	Peak Picking				DP Beat Tracking		
Dataset	Δ	Γ	$\Gamma^{\pm 40}$	Γ^{DP}	Δ	Γ	Ψ
BEATLES	0.050	0.044	0.554	0.555	0.789	0.708	**0.824**
RWC-POP	0.015	0.005	0.486	0.444	0.757	0.743	**0.808**
RWC-JAZZ	0.014	0.003	0.253	0.231	0.414	**0.535**	0.493
RWC-CLASSIC	0.124	0.118	0.381	0.393	0.536	0.528	**0.560**
MAZURKA	0.238	0.225	**0.539**	0.484	0.451	0.479	0.508

(b)

	Peak Picking				DP Beat Tracking		
Dataset	Δ	Γ	$\Gamma^{\pm 40}$	Γ^{DP}	Δ	Γ	Ψ
BEATLES	0.050	0.044	0.597	0.592	0.902	0.909	**0.926**
RWC-POP	0.016	0.005	0.528	0.494	0.881	0.917	**0.923**
RWC-JAZZ	0.014	0.003	0.282	0.259	0.564	0.705	**0.708**
RWC-CLASSIC	0.125	0.119	0.404	0.420	0.638	0.633	**0.661**
MAZURKA	0.238	0.226	**0.540**	0.486	0.466	0.528	0.527

Table 2.7: Beat tracking results based on context-sensitive evaluation measures ($L = 4$). **(a)** F^L-measures. **(b)** Half-shift invariant $\tilde{\mathrm{F}}^L$-measures.

simply choosing the local maximum from the constrained tempo set allows for locally adapting to the strongly varying tempo. Actually, peak picking on $\Gamma^{\pm 40}$ leads to the best results for this dataset. For all other datasets, however, employing the DP beat tracking procedure improves the results. In particular, for popular music with only moderate tempo changes, the stricter F^L-measures come close to the simple F-measures (e. g., $\mathrm{F}^L = 0.824$ compared to $\mathrm{F} = 0.861$ for BEATLES and Ψ).

To investigate the role of half-beat shifts as discussed in Section 2.9.6, we make the evaluation measure invariant to such errors. To this end, we shift the sequence R of reference beats by one half-beat to the right (replacing r_m by $\tilde{r}_m := (r_{m+1} + r_m)/2$) to obtain a sequence $\tilde{R}$. Then the reference beat r_m is considered correct if r_m is L-correct w. r. t. R or if $\tilde{r}_m$ is L-correct w. r. t. $\tilde{R}$. As before, we define recall and precision to obtain a half-shift invariant F-measure denoted by $\tilde{\mathrm{F}}^L$. Table 2.7b shows the corresponding evaluation results. In particular for the DP tracking approach, one obtains a significant increase in the evaluation measures for all datasets. For example, for BEATLES and Ψ, one has $\tilde{\mathrm{F}}^L = 0.926$ opposed to $\mathrm{F}^L = 0.824$, which shows that half-beat shifts are a common problem in beat tracking. Actually, even humans sometimes perceive beats on off-beat positions, in particular for syncopal passages with strong off-beat events. This also explains the strong increase in the case of Jazz music ($\tilde{\mathrm{F}}^L = 0.708$ opposed to $\mathrm{F}^L = 0.493$ for RWC-JAZZ and Ψ), where one often encounters syncopal elements. For DP tracking based on Γ, the improvements are most noticeable over all datasets. As Γ is invariant to dynamics, the half-shift beat confusion is very distinctive, see Figure 2.15b.

Finally, we note that the context-sensitive evaluation measures much better reveal the kind of improvements introduced by our PLP-concept, which tends to suppress spurious peaks. For both approaches, peak picking and DP beat tracking, one obtains the best results when using a PLP-based enhancement.

2.10 Conclusion

In this chapter, we introduced a novel concept for deriving musically meaningful local pulse information from possibly noisy onset information. Opposed to previous approaches that assume constant tempo, the main benefit of our PLP mid-level representation is that it can locally adjust to changes in tempo as long as the underlying music signal possesses some quasi-periodicity. In our representation, we do not aim at extracting pulses at a specific level. Instead, a PLP curve is able to locally switch to the dominating pulse level, which typically is the tatum level. Furthermore, our concept allows for integrating additional knowledge in form of a tempo range to enforce pulse detection on a specific level. Conducting extensive experiments based on well-known datasets of different genres, we have shown that our PLP concept constitutes a powerful tool for tempo estimation and beat tracking. Furthermore, initial experiments also revealed that PLP curves are suitable for supporting higher-level music processing tasks such as music synchronization [49], meter estimation [102], as well as pulse-adaptive feature design [45] and audio segmentation [114].

Even for classical music with soft onsets, we were able to extract useful tempo and beat information. However, for highly expressive interpretations of romantic music, the assumption of local quasi-periodicity is often violated leading to poor results. At least, our PLP concept yields a confidence measure to reveal such problematic passages. Highly-expressive music also reveals the limits of purely onset-oriented tempo and beat tracking procedures. Here, future work is concerned with jointly considering additional musical aspects regarding meter, harmony, polyphony, or structure in order to support and stabilize tempo and beat tracking, see [138; 37; 61; 27; 148] for first approaches towards this direction.

Chapter 3

A Case Study on Chopin Mazurkas

In the last chapter, we introduced a novel concept for deriving musically meaningful local pulse information. As it turned out, highly-expressive music reveals the limits of state-of-the-art tempo and beat tracking procedures. To better understand the shortcomings of beat tracking methods, significant efforts have been made to compare and investigate the performance of different strategies on common datasets [43; 188; 121; 69; 38]. However, most approaches were limited to comparing the different methods by specifying evaluation measures that refer to an entire recording or even an entire collection of recordings. Such globally oriented evaluations do not provide any information on the critical passages within a piece where the tracking errors occur. Thus, no conclusions can be drawn from these experiments about possible *musical reasons* that lie behind the beat tracking errors. A first analysis of *musical properties* influencing the beat tracking quality was conducted by Dixon [38], who proposed quantitative measures for the rhythmic complexity and for variations in tempo and timings. However, no larger evaluations were carried out to show a correlation between these theoretical measures and the actual beat tracking quality.

In this chapter, we introduce a novel evaluation framework that exploits the existence of different performances available for a given piece of music. In our case study we revert to a collection of recordings for the Chopin Mazurkas containing in average over 50 performances for each piece. Based on a local, beat-wise histogram, we simultaneously determine consistencies of beat tracking errors over many performances. The underlying assumption is, that tracking errors consistently occurring in many performances of a piece are likely caused by musical properties of the piece, rather than physical properties of a specific performance. These consistencies indicate musically critical passages in the underlying piece, rather than a specific performance that are prone to tracking errors. As a further contribution, we classify the beats of the critical passages by introducing various types of beats such as non-event beats, ornamented beats, weak bass beats, or constant harmony beats. Each such beat class stands for a musical performance-independent property that frequently evokes beat tracking errors. In our experiments, we evaluated three conceptually different beat tracking procedures on a corpus consisting of 298 audio recordings corresponding to five different Mazurkas. For each recording, the tracking results were compared with

ID	Composer	Piece	#(Meas.)	#(Beats)	#(Perf.)
M17-4	Chopin	Op. 17, No. 4	132	396	62
M24-2	Chopin	Op. 24, No. 2	120	360	64
M30-2	Chopin	Op. 30, No. 2	65	193	34
M63-3	Chopin	Op. 63, No. 3	77	229	88
M68-3	Chopin	Op. 68, No. 3	61	181	50

Table 3.1: The five Chopin Mazurkas and their identifiers used in our study. The last three columns indicate the number of measures, beats, and performances available for the respective piece.

manually annotated ground-truth beat positions. Our local evaluation framework and detailed analysis explicitly indicates various limitations of current state-of-the-art beat trackers, thus laying the basis for future improvements and research directions.

This chapter is organized as follows: In Section 3.1, we formalize and discuss the beat tracking problem. In Section 3.2, we describe the underlying music material and specify various beat classes. After summarizing the three beat tracking strategies (Section 3.3) used in our case study and introducing the evaluation measure (Section 3.4), we report on the experimental results in Section 3.5. Finally, we conclude in Section 3.6 with a discussion of future research directions.

3.1 Specification of the Beat Tracking Problem

For a given piece of music, let N denote the number of *musical beats*. Enumerating all beats, we identify the set of musical beats with the set $\mathcal{B} = [1 : N] := \{1, 2, \ldots, N\}$. Given a performance of the piece in the form of an audio recording, the musical beats correspond to specific physical time positions within the audio file. Let $\pi : \mathcal{B} \to \mathbb{R}$ be the mapping that assigns each musical beat $b \in \mathcal{B}$ to the time position $\pi(b)$ of its occurrence in the performance. In the following, a time position $\pi(b)$ is referred to as *physical beat* or simply as *beat* of the performance. Then, the task of *beat tracking* is to recover the set $\{\pi(b) \mid b \in \mathcal{B}\}$ of all beats from a given audio recording.

Note that this specification of the beat tracking problem is somewhat simplistic, as we only consider physical beats that are defined by onset events. More generally, a beat is a perceptual phenomenon and perceptual beat times do not necessarily coincide with physical beat times [41]. Furthermore, the perception of beats varies between listeners.

For determining physical beat times, we now discuss some of the problems, one has to deal with in practice. Typically, a beat goes along with a note onset revealed by an increase of the signal's energy or a change in the spectral content. However, in particular for non-percussive music, one often has soft note onsets, which lead to blurred note transitions rather than sharp note onset positions. In such cases, there are no precise timings of note events within the audio recording, and the assignment of exact physical beat positions becomes problematic. This issue is aggravated in the presence of tempo changes and expressive tempo nuances (e.g., ritardando and accelerando).

ID	$\lvert\mathcal{B}\rvert$	$\lvert\mathcal{B}_1\rvert$	$\lvert\mathcal{B}_2\rvert$	$\lvert\mathcal{B}_3\rvert$	$\lvert\mathcal{B}_4\rvert$	$\lvert\mathcal{B}_5\rvert$	$\lvert\mathcal{B}_*\rvert$
M17-4	396	9	8	51	88	0	154
M24-2	360	10	8	22	4	12	55
M30-2	193	2	8	13	65	0	82
M63-3	229	1	7	9	36	0	47
M68-3	181	17	7	0	14	12	37

Table 3.2: The number of musical beats in each of the different beat classes defined in Section 3.2. Each beat may be a member of more than one class.

Besides such physical reasons, there may also be a number of musical reasons for beat tracking becoming a challenging task. For example, there may be beats with no note event going along with them. Here, a human may still perceive a steady beat, but the automatic specification of physical beat positions is quite problematic, in particular in passages of varying tempo where interpolation is not straightforward. Furthermore, auxiliary note onsets can cause difficulty or ambiguity in defining a specific physical beat time. In music such as the Chopin Mazurkas, the main melody is often embellished by ornamented notes such as trills, grace notes, or arpeggios. Also, for the sake of expressiveness, the notes of a chord need not be played at the same time, but slightly displaced in time. This renders a precise definition of a physical beat position impossible.

3.2 Five Mazurkas by Frédéric Chopin

The Mazurka Project[1] has collected over 2700 recorded performances for 49 Mazurkas by Frédéric Chopin, ranging from the early stages of music recording (Grünfeld 1902) until today [153]. In our case study, we use 298 recordings corresponding to five of the 49 Mazurkas, see Table 3.1. For each of theses recordings the beat positions were annotated manually [153]. These annotations are used as ground truth in our experiments. Furthermore, Humdrum and MIDI files of the underlying musical scores for each performance are provided, representing the pieces in an uninterpreted symbolic format.

In addition to the physical beat annotations of the performances, we created musical annotations by grouping the musical beats $\mathcal{B}$ in five different beat classes $\mathcal{B}_1$ to $\mathcal{B}_5$. Each of these classes represents a musical property that typically constitutes a problem for determining the beat positions. The colors refer to Figure 3.4, Figure 3.5, and Figure 3.6.

- **Non-event beats $\mathcal{B}_1$ (black)**: Beats that do not coincide with any note events, see Figure 3.1a.
- **Boundary beats $\mathcal{B}_2$ (blue)**: Beats of the first measure and last measure of the piece.
- **Ornamented beats $\mathcal{B}_3$ (red)**: Beats that coincide with ornaments such as trills, grace notes, or arpeggios, see Figure 3.1b.
- **Weak bass beats $\mathcal{B}_4$ (cyan)**: Beats where only the left hand is played, see Figure 3.1e.

[1] mazurka.org.uk

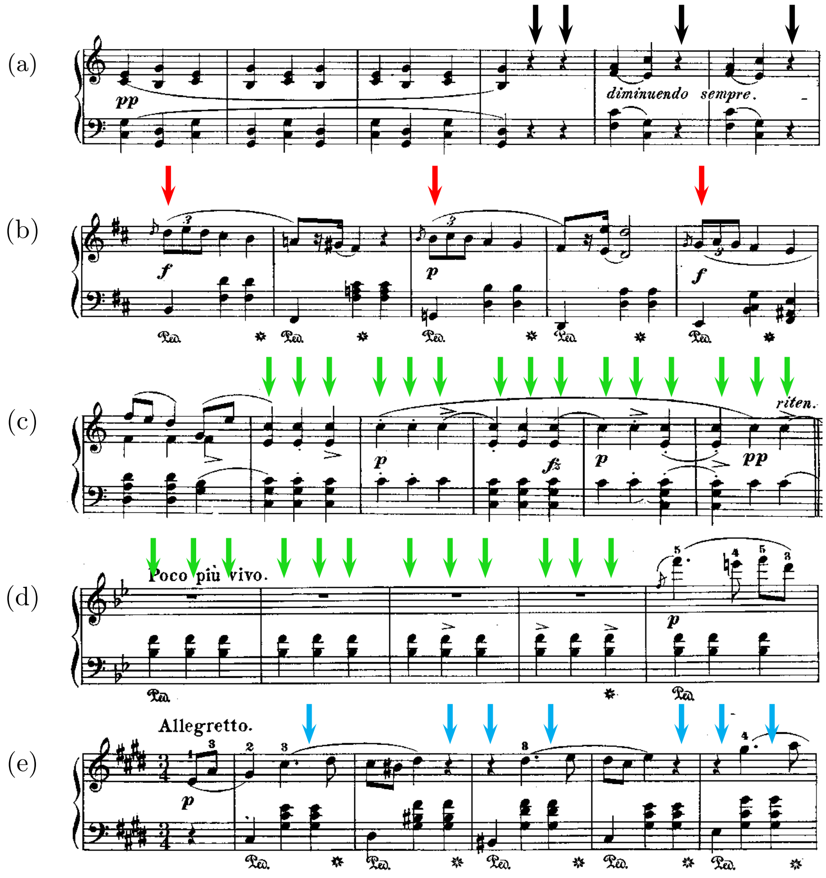


Figure 3.1: Scores of example passages for the different beat classes introduced in Section 3.2. **(a)** Non-event beats ($\mathcal{B}_1$) in M24-2, **(b)** Ornamented beats ($\mathcal{B}_3$) in M30-2, **(c)** Constant harmony beats ($\mathcal{B}_5$) in M24-2, **(d)** Constant harmony beats ($\mathcal{B}_5$) in M68-3, and **(e)** Weak bass beats ($\mathcal{B}_4$) in M63-3.

- **Constant harmony beats $\mathcal{B}_5$ (green)**: Beats that correspond to consecutive repetitions of the same chord, see Figure 3.1(c-d).

Furthermore, let $\mathcal{B}_* := \cup_{k=1}^{5} \mathcal{B}_k$ denote the union of the five beat classes. Table 3.2 details for each Mazurka the number of beats assigned to the respective beat classes. Note that the beat classes need not be disjoint, i.e., each beat may be assigned to more than one class. In Section 3.5, we discuss the beat classes and their implications on the beat tracking results in more detail.

3.3 Beat Tracking Strategies

In our experiments we use three different beat trackers, see Section 2.1 for an overview. Firstly, we directly use the onset candidates extracted from a novelty curve as explained in Section 2.3. Figure 3.2c shows a novelty curve for an excerpt of M17-4 (identifier explained in Table 3.1). Using a peak picking strategy [7] note onsets can be extracted from this

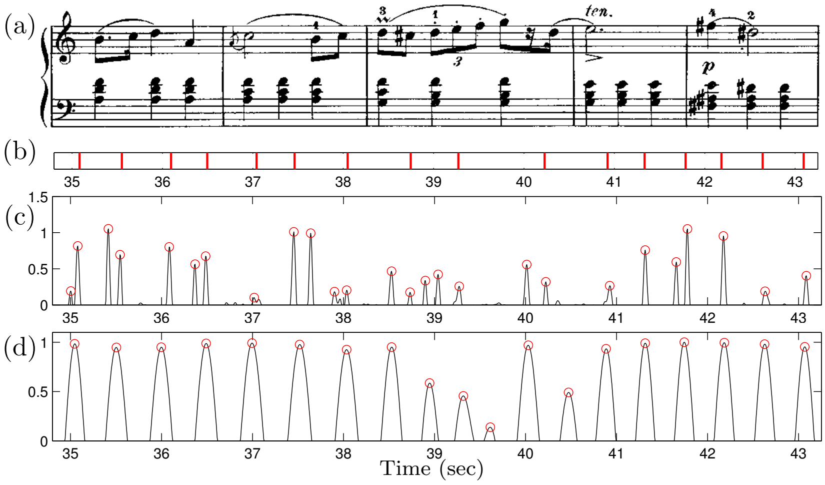


Figure 3.2: Representations for an excerpt of M17-4. **(a)** Score representation of beats 60 to 74. **(b)** Annotated ground truth beats for the performance pid50534-05 by Horowitz (1985). **(c)** Novelty curve (note onset candidates indicated by circles). **(d)** PLP curve (beat candidates indicated by circles).

curve. In this method, referred to as ONSET in the following sections, each detected note onset is considered as a beat position. Secondly, we compute a PLP curve from the novelty curve as introduced in Section 2.6 and consider the PLP peak positions as beat positions. In the following, this approach is referred to as PLP. We use a window size of three seconds and initialize the tempo estimation with the mean of the annotated tempo. More precisely, we define the global tempo range for each performance covering one octave around the mean tempo, e.g., for a mean tempo of 120 BPM, tempo estimates in the range [90 : 180] are valid. This prevents tempo doubling or halving errors and robustly allows for investigating beat tracking errors, rather than tempo estimation errors. The third beat tracking method (SYNC) employs the MIDI file available for each piece. This MIDI file can be regarded as additional knowledge, including the pitch, onset time and duration of each note. Using suitable synchronization techniques [49] on the basis of coarse harmonic and very precise onset information, we identify for each musical event of the piece (given by the MIDI file) the corresponding physical position within a performance. This coordination of MIDI events to the audio is then used to determine the beat positions in a performance and simplifies the beat tracking task to an alignment problem, where the number of beats and the sequence of note events is given as prior knowledge.

3.4 Evaluation on the Beat Level

Many evaluation measures have been proposed to quantify the performance of beat tracking systems by comparing the beat positions determined by a beat tracking algorithm and annotated ground truth beats. An extensive review of evaluation measures is given in [29]. These measures can be divided into two groups. Firstly, measures that analyze each beat position separately and secondly, measures that take the tempo and metrical levels into account [31; 30; 102; 121]. While the latter gives a better estimate of how well a *sequence*

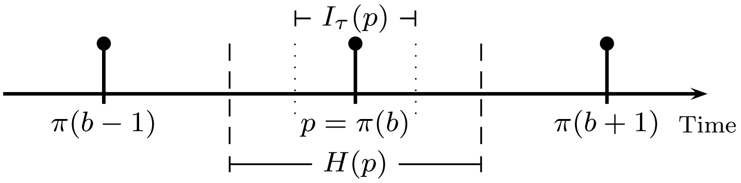


Figure 3.3: Illustration of the τ-neighborhood $I_\tau(p)$ and the half-beat neighborhood $H(p)$ of a beat $p = \pi(b)$, $b \in \mathcal{B}$.

of retrieved beats correlates with the manual annotation, it does not give any insight into the beat tracking performance at a specific beat of the piece.

In our evaluation, we consider the beat tracking quality on the beat-level of a piece and combine the results of all performances available for this piece. This allows for detecting beats that are prone to errors in many performances. For a given performance, let $\Pi := \{\pi(b) \,|\, b \in \mathcal{B}\}$ be the set of manually determined physical beats, which are used as ground truth. Furthermore, let $\Phi \subset \mathbb{R}$ be the set of beat candidates obtained from a beat tracking procedure. Given a tolerance parameter $\tau > 0$, we define the τ-neighborhood $I_\tau(p) \subset \mathbb{R}$ of a beat $p \in \Pi$ to be the interval of length 2τ centered at p, see Figure 3.3. We say that a beat p has been *identified* if there is a beat candidate $q \in \Phi$ in the τ-neighborhood of p, i.e., $q \in \Phi \cap I_\tau(p)$. Let $\Pi_{\text{id}} \subset \Pi$ be the set of all identified beats. Furthermore, we say that a beat candidate $q \in \Phi$ is *correct* if q lies in the τ-neighborhood $I_\tau(p)$ of some beat $p \in \Pi$ and there is no other beat candidate lying in $I_\tau(p)$ that is closer to p than q. Let $\Phi_{\text{co}} \subset \Phi$ be the set of all correct beat candidates. We then define the precision $\mathrm{P} = \mathrm{P}_\tau$, the recall $\mathrm{R} = \mathrm{R}_\tau$, and F-measure $\mathrm{F} = \mathrm{F}_\tau$ as [29]

$$\mathrm{P} = \frac{|\Phi_{\text{co}}|}{|\Phi|}, \quad \mathrm{R} = \frac{|\Pi_{\text{id}}|}{|\Pi|}, \quad \mathrm{F} = \frac{2 \cdot \mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}} \,. \tag{3.1}$$

Table 3.3 shows the results of various beat tracking procedures on the Mazurka data. As it turns out, the F-measure is a relatively soft evaluation measure that only moderately punishes additional, non-correct beat candidates. As a consequence, the simple onset-based beat tracker seems to outperform most other beat trackers. As for the Mazurka data, many note onsets coincide with beats, the onset detection leads to a high recall, while having only a moderate deduction in the precision.

We now introduce a novel evaluation measure that punishes non-correct beat candidates, which are often musically meaningless, more heavily. To this end, we define a *half-beat neighborhood* $H(p)$ of a beat $p = \pi(b) \in \Pi$ to be the interval ranging from $\frac{\pi(b-1)-\pi(b)}{2}$ (or $\pi(b)$ for $b = 1$) to $\frac{\pi(b+1)-\pi(b)}{2}$ (or $\pi(b)$ for $b = N$), see Figure 3.3. Then, we say that a beat $b \in \mathcal{B}$ has been *strongly identified* if there is a beat candidate $q \in \Phi$ with $q \in \Phi \cap I_\tau(p)$ and if $H(p) \cap \Phi = \{q\}$ for $p = \pi(b)$. In other words, q is the only beat candidate in the half-beat neighborhood of p. Let $\Pi_{\text{stid}} \subset \Pi$ be the set of all strongly identified beats, then we define the *beat accuracy* $A = A_\tau$ to be

$$A = \frac{|\Pi_{\text{stid}}|}{|\Pi|}. \tag{3.2}$$

	SYNC	ONSET				PLP			
ID	P/R/F/A	P	R	F	A	P	R	F	A
M17-4	0.837	0.552	0.958	0.697	0.479	0.615	0.743	0.672	0.639
M24-2	0.931	0.758	0.956	0.845	0.703	0.798	0.940	0.862	0.854
M30-2	0.900	0.692	0.975	0.809	0.623	0.726	0.900	0.803	0.788
M63-3	0.890	0.560	0.975	0.706	0.414	0.597	0.744	0.661	0.631
M68-3	0.875	0.671	0.885	0.758	0.507	0.634	0.755	0.689	0.674
Mean:	**0.890**	0.634	0.952	**0.754**	0.535	0.665	0.806	**0.728**	0.729

	MIREX				Our Methods		
Method	DRP3	GP2	OGM2	TL	SYNC	ONSET	PLP
F	0.678	0.547	0.321	0.449	**0.890**	**0.754**	**0.728**

Table 3.3: Comparison of the beat tracking performance of the three strategies used in this chapter and the MIREX 2009 results based on the evaluation metrics Precision P, Recall R, F-measure F and the beat accuracy A.

3.5 Experimental Results

We now discuss the experimental results obtained using our evaluation framework and explain the relations between the beat tracking results and the beat classes introduced in Section 3.2.

We start with discussing Table 3.3. Here, the results of the different beat tracking approaches for all performances of the five Mazurkas are summarized, together with some results from the MIREX 2009 beat tracking task[2]. All beat trackers used in our evaluation yield better results for the Mazurkas than all trackers used in the MIREX evaluation. As noted before, the F-measure only moderately punishes additional beats. In consequence, ONSET (F = 0.754) seems to outperform all other methods, except SYNC (F = 0.890). In contrast, the introduced beat accuracy A punishes false positives more heavily, leading to $A = 0.535$ for ONSET, which is significantly lower than for PLP ($A = 0.729$) and SYNC ($A = 0.890$). For SYNC, the evaluation metrics P, R, F, and A are equivalent because the number of detected beats is always correct. Furthermore, SYNC is able to considerably outperform the other strategies. This is not surprising, as it is equipped with additional knowledge in the form of the MIDI file.

There are some obvious differences in the beat tracking results of the individual Mazurkas caused by the musical reasons explained in [38]. First of all, all methods deliver the best result for M24-2. This piece is rather simple, with many quarter notes in the dominant melody line. M17-4 is the most challenging for all three trackers because of a frequent use of ornaments and trills and many beat positions that are not reflected in the dominating melody line. For the ONSET tracker, M63-3 constitutes a challenge ($A = 0.414$), although this piece can be handled well by the SYNC tracker. Here, a large number of notes that do not fall on beat positions provoke many false positives. This also leads to a low accuracy of PLP ($A = 0.631$).

Going beyond this evaluation on a piece-level, Figure 3.4, Figure 3.5, and Figure 3.6

[2] www.music-ir.org/mirex/wiki/2009:Audio_Beat_Tracking_Results

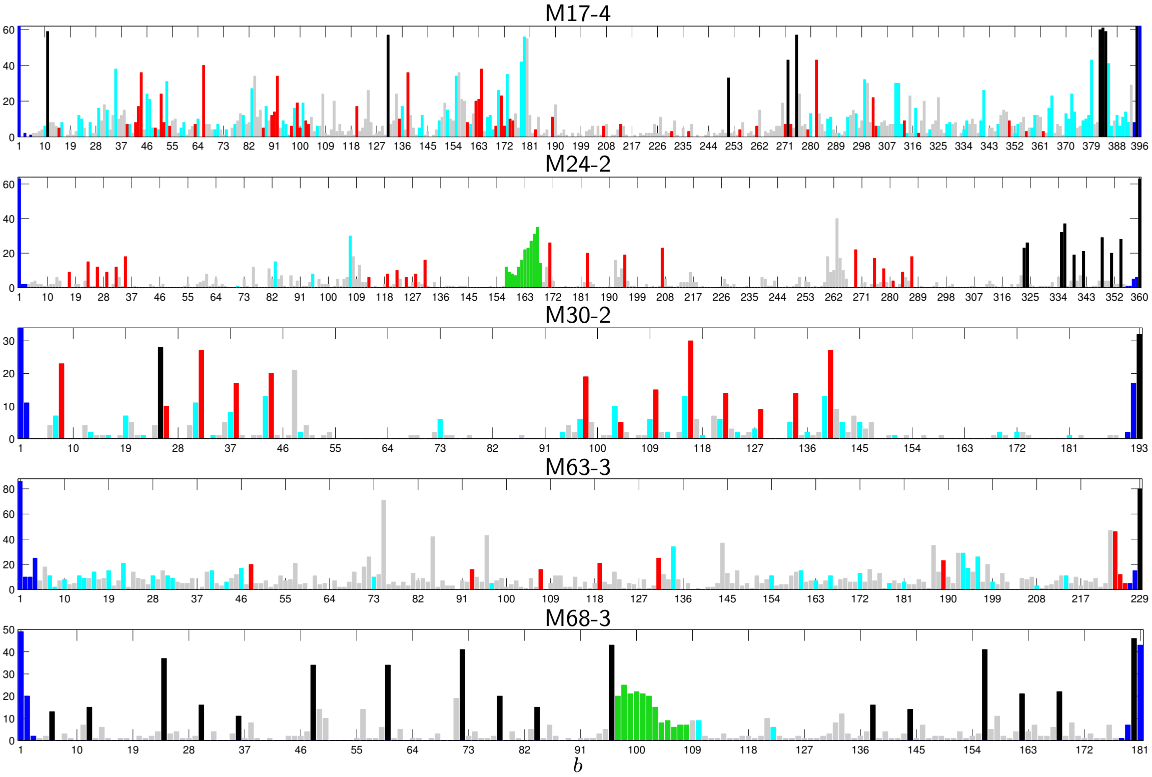


Figure 3.4: The beat error histogram for the synchronization based beat tracking (SYNC) shows for how many performances of each of the five Mazurkas a beat b is not identified. The different colors of the bars encode the beat class $\mathcal{B}$ a beat is assigned to, see Section 3.2.

ID	$\mathcal{B}$	$\mathcal{B}\backslash\mathcal{B}_1$	$\mathcal{B}\backslash\mathcal{B}_2$	$\mathcal{B}\backslash\mathcal{B}_3$	$\mathcal{B}\backslash\mathcal{B}_4$	$\mathcal{B}\backslash\mathcal{B}_5$	$\mathcal{B}\backslash\mathcal{B}_*$
M17-4	0.837	0.852	0.842	0.843	0.854	0.837	0.898
M24-2	0.931	0.940	0.936	0.941	0.933	0.939	0.968
M30-2	0.900	0.900	0.903	0.931	0.905	0.900	0.959
M63-3	0.890	0.890	0.898	0.895	0.895	0.890	0.911
M68-3	0.875	0.910	0.889	0.875	0.875	0.887	0.948
Mean:	0.890	0.898	0.894	0.897	0.894	0.892	0.925

Table 3.4: Beat accuracy A results comparing the different beat classes for SYNC: For all beats $\mathcal{B}$, excluding non-event beats $\mathcal{B}_1$, boundary beats $\mathcal{B}_2$, ornamented beats $\mathcal{B}_3$, weak bass beats $\mathcal{B}_4$, constant harmony beats $\mathcal{B}_5$, and the union $\mathcal{B}_*$.

illustrate the beat-level beat tracking results of our evaluation framework for the SYNC, PLP, and ONSET strategy, respectively. Here, for each beat $b \in \mathcal{B}$ of a piece, the bar encodes for how many of the performances of this piece the beat was not *strongly identified* (see Section 3.4). High bars indicate beats that are incorrectly identified in many performances, low bars indicate beats that are identified in most performances without problems. As a consequence, this representation allows for investigating the musical properties leading to beat errors. More precisely, beats that are consistently wrong over a large number of performances of the same piece are likely to be caused by musical properties of the piece, rather than physical properties of a specific performance. For example, for the tracking

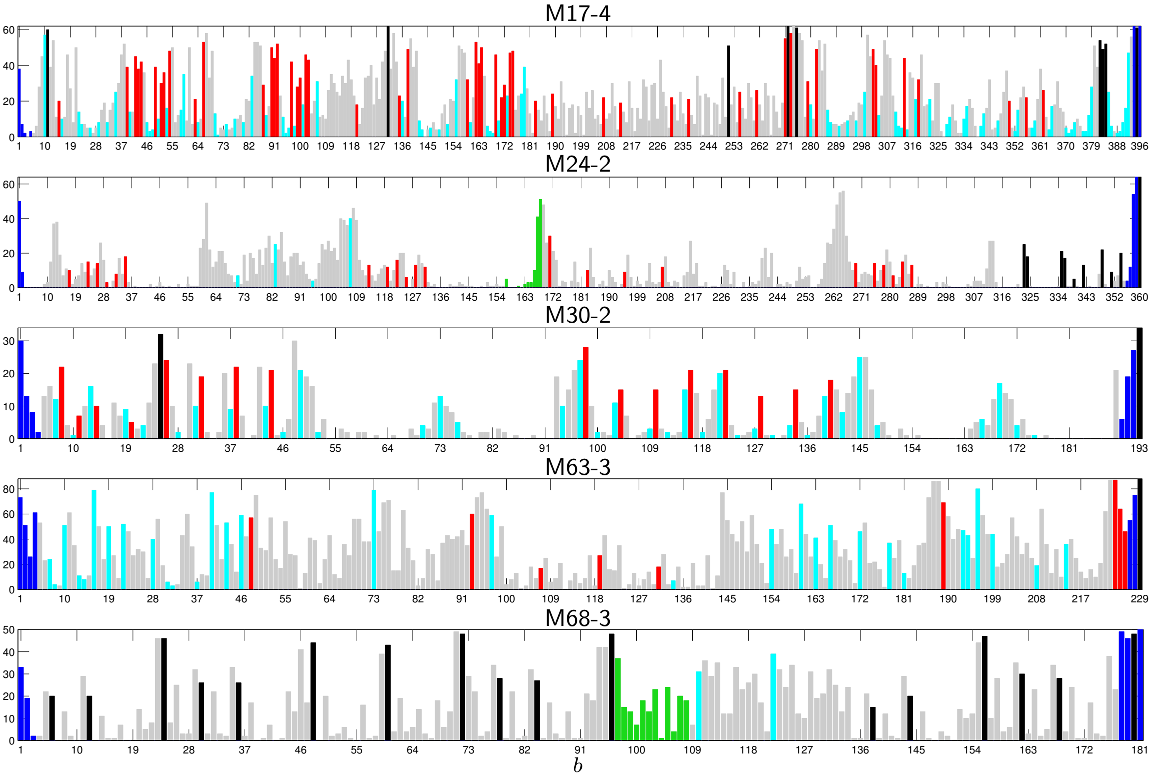


Figure 3.5: The beat error histogram for the PLP tracker shows for how many performances of each of the five Mazurkas a beat b is not identified. The different colors of the bars encode the beat class $\mathcal{B}$ a beat is assigned to, see Section 3.2.

ID	$\mathcal{B}$	$\mathcal{B}\backslash\mathcal{B}_1$	$\mathcal{B}\backslash\mathcal{B}_2$	$\mathcal{B}\backslash\mathcal{B}_3$	$\mathcal{B}\backslash\mathcal{B}_4$	$\mathcal{B}\backslash\mathcal{B}_5$	$\mathcal{B}\backslash\mathcal{B}_*$
M17-4	0.639	0.650	0.641	0.671	0.593	0.639	0.649
M24-2	0.854	0.857	0.862	0.857	0.856	0.854	0.873
M30-2	0.788	0.788	0.794	0.814	0.772	0.788	0.822
M63-3	0.631	0.631	0.638	0.639	0.647	0.631	0.668
M68-3	0.674	0.705	0.689	0.674	0.678	0.674	0.733
Mean:	0.729	0.735	0.734	0.739	0.723	0.729	0.751

Table 3.5: Beat accuracy A results comparing the different beat classes for PLP: For all beats $\mathcal{B}$, excluding non-event beats $\mathcal{B}_1$, boundary beats $\mathcal{B}_2$, ornamented beats $\mathcal{B}_3$, weak bass beats $\mathcal{B}_4$, constant harmony beats $\mathcal{B}_5$, and the union $\mathcal{B}_*$.

strategies SYNC and PLP and all five pieces, the first and last beats are incorrectly identified in almost all performances, as shown by the blue bars ($\mathcal{B}_2$). This is caused by boundary problems and adaption times of the algorithms. For ONSET, this effect is less pronounced as only local information is used for determining beat positions. As a result, there is no adaption time for ONSET.

Furthermore, there is a number of significant high bars within all pieces. The SYNC strategy for M68-3 (see Figure 3.4) exhibits a number of isolated black bars. These non-event beats do not fall on any note-event ($\mathcal{B}_1$). As stated in Section 3.1, especially when dealing with expressive music, simple interpolation techniques do not work to infer these beat

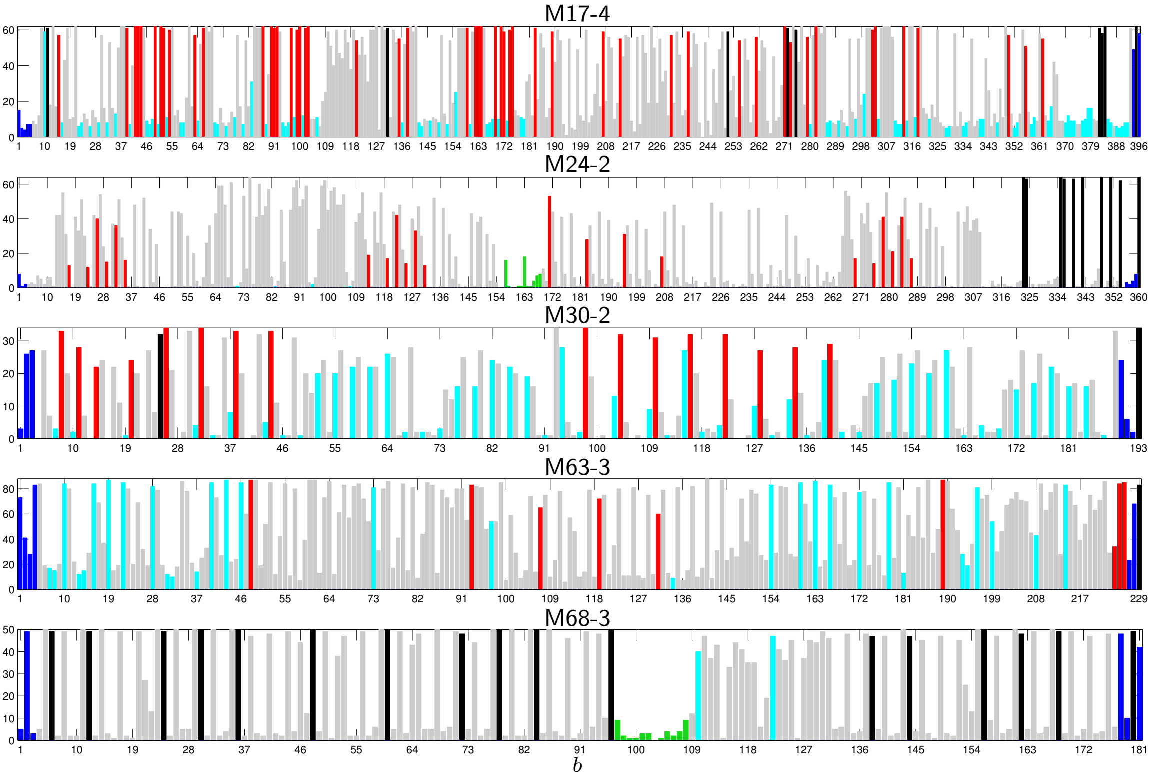


Figure 3.6: The beat error histogram for the ONSET tracker shows for how many performances of each of the five Mazurkas a beat b is not identified. The different colors of the bars encode the beat class $\mathcal{B}$ a beat is assigned to, see Section 3.2.

positions automatically. The same beat positions are problematic in the PLP strategy, see Figure 3.5 and in particular in ONSET, see Figure 3.6. For M30-2 (Figure 3.4) most of the high bars within the piece are assigned to $\mathcal{B}_3$ (red). These beats, which coincide with ornaments such as trills, grace notes, or arpeggios are physically not well defined and hard to determine. For the Mazurkas, chords are often played on-beat by the left hand. However, for notes of lower pitch, onset detection is problematic, especially when played softly. As a consequence, beats that only coincide with a bass note or chord, but without any note being played in the main melody, are a frequent source for errors. This is reflected by the cyan bars ($\mathcal{B}_3$) frequently occurring in M17-4 (Figure 3.4). Finally, $\mathcal{B}_5$ (green) contains beats falling on consecutive repetitions of the same chord. This constitutes a challenge for the onset detection, especially when played softly. Both M24-2 and M68-3 exhibit a region of green bars that are incorrectly tracked by the SYNC (Figure 3.4) and PLP (Figure 3.5) trackers.

As mentioned in Section 3.3, PLP can not handle tempo changes well. As a consequence, many of the beat errors for PLP that are not assigned to any beat class (e.g., M24-2 in Figure 3.5, $b = [260 : 264]$) are caused by sudden tempo changes appearing in many of the performances. However, these are considered a performance-dependent property, rather than a piece-dependent musical property and are not classified in a beat class.

Tables 3.4 and 3.5 summarize the effect of each beat class on the piece-level results.

Here, the mean beat accuracy is reported for each of the five Mazurkas, when excluding the beats of a certain class. For example, M30-2 contains many beats of $\mathcal{B}_3$. Excluding these ornamented beats from the evaluation, the overall beat accuracy increases from $A = 0.900$ to $A = 0.931$ for SYNC (Table 3.4) and from 0.788 to 0.814 for PLP (Table 3.5). The challenge of M68-3 however, are non-event beats ($\mathcal{B}_1$). Leaving out these beats, the accuracy increases from 0.875 to 0.910 for SYNC and from 0.674 to 0.705 for PLP.

Aside from musical properties of a piece causing beat errors, physical properties of certain performances make beat tracking difficult. In the following, we exemplarily compare the beat tracking results of the performances of M63-3. Figure 3.7 shows the beat accuracy A for all 88 performances available for this piece. In case of the SYNC tracker, the beat accuracy for most of the performances is in the range of $0.8 - 0.9$, with only few exceptions that deviate significantly (Figure 3.7a). In particular, Michalowski's 1933 performance with index 39 (pid9083-16) shows a low accuracy of only $A = 0.589$ due to a poor condition of the original recording which contains a low signal-to-noise ratio and many clicks. The low accuracy ($A = 0.716$) of performance 1 (Csalog 1996, pid1263b-12) is caused by a high amount of reverberation, which makes a precise determination of the beat positions hard. The poor result of performance 81 (Zak 1951, pid918713-20) is caused by a detuning of the piano. Compensating for this tuning effect, the synchronization results and thus, the beat accuracy improves from $A = 0.767$ to $A = 0.906$. As it turns out, ONSET tends to be even more sensitive to bad recording conditions. Again, performance 39 shows an extremely low accuracy ($A = 0.087$), however, there are more recordings with a very low accuracy (70, 71, 79, 80, 57, and 58). Further inspection shows that all of these recordings contain noise, especially clicks and crackling, which proves devastating for onset detectors and leads to a high number of false positives. Although onset detection is problematic for low quality recordings, the PLP approach shows a different behavior. Here, the periodicity enhancement of the novelty curve [72] provides a cleaning effect and is able to eliminate many spurious peaks caused by recording artifacts and leads to a higher beat accuracy. However, other performances suffer from a low accuracy (performances 29, 30, and 77). As it turns out, these examples exhibit extreme local tempo changes that can not be captured well by the PLP approach, which relies on a constant tempo within the analysis window. On the other hand, some performances show a noticeably higher accuracy (2, 5, 11, 31, 74, and 87). All oft these recordings are played in a rather constant tempo.

3.6 Further Notes

Our experiments indicate that our approach of considering multiple performances simultaneously for a given piece of music for the beat tracking task yields a better understanding not only of the algorithms' behavior but also of the underlying music material. The understanding and consideration of the physical and musical properties that make beat tracking difficult is of essential importance for improving the performance of beat tracking approaches. Exploiting the knowledge of the musical properties leading to beat tracking errors one can design more advanced audio features. For example, in case of the Chopin Mazurkas the tempo and beat is often revealed only by the left hand, whereas the right hand often has an improvisatory character. For this kind of music, one may achieve improvements when separating the recording into melody (right hand) and accompaniment

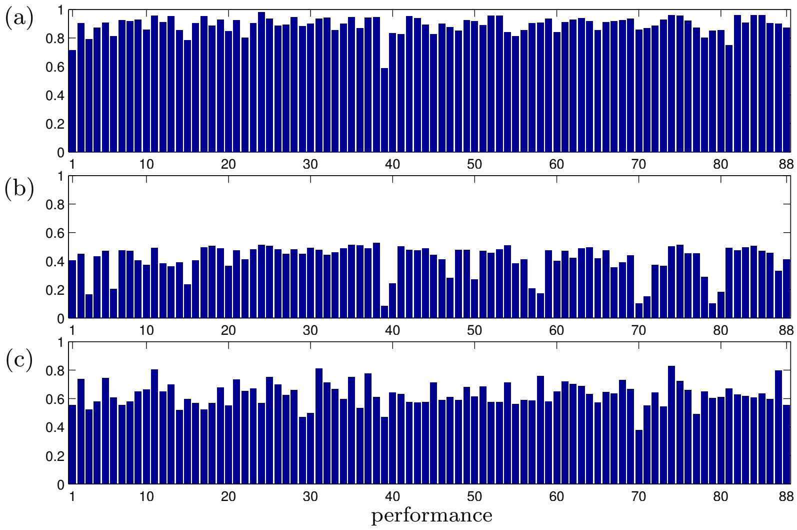


Figure 3.7: Beat accuracy A for the beat tracker SYNC **(a)**, ONSET **(b)**, and PLP **(c)** of all 88 performances of M63-3.

(left hand) using source separation techniques as proposed in [48]. Analyzing the voices individually, the quality of the novelty curve can be enhanced to alleviate the negative effect of the ornamented beats or weak bass beats.

In [85], our concept has now been adopted for investigating beat tracking results obtained by different beat tracking approaches on the same dataset. Considering inconstistencies in the beat tracking results obtained from the different procedures, the authors apply this approach also to music recordings without any manually annotated ground truth. Here, the underlying assumption is that inconsistencies in the beat tracking result indicate problematic examples. Consistencies across the different trackers, however, are a result of a correctly tracked beat. As a result, this approach allows for detecting difficult *recordings* that are a challenge for beat tracking algorithms.

Chapter 4

Tempo-Related Audio Features

Our experiments in the preceding chapters indicate that the extraction of tempo and beat information is a challenging problem. In particular, in the case of weak note onsets and significant tempo changes, determining explicit tempo values is an error-prone step. In this chapter, we introduce various robust *mid-level* representations that capture local tempo characteristics of music signals. Instead of extracting explicit tempo information from the music recordings (which is an error-prone step), the mid-level representations reveal information about local changes of the tempo. First, we generalize the concept of tempograms. Tempogram representations derived from a novelty curve already played a major role in the computation of the PLP curves as introduced in Chapter 2. In addition to the tempogram based on a Fourier transform, we now introduce a second variant based on an autocorrelation function. As it turnes out, the autocorrelation tempogram naturally complements the Fourier tempogram [146].

An important property of musical rhythm is that there are various pulse levels that contribute to the human perception of tempo such as the measure, tactus, and tatum levels [102], see Section 2.1. As an analogy, these different levels may be compared to the existence of harmonics in the pitch context. Inspired by the concept of chroma features, we introduce the concept of *cyclic tempograms*, where the idea is to form tempo equivalence classes by identifying tempi that differ by a power of two. Originally suggested in [104] we formalize and expand this concept in this chapter. The resulting cyclic tempo features constitute a robust mid-level representation that reveals local tempo characteristics of music signals while being invariant to changes in the pulse level. Being the tempo-based counterpart of the harmony-based chromagrams, cyclic tempograms are suitable for music analysis and retrieval tasks, where harmony-based, timbre-based, and rhythm-based criteria are not relevant or applicable.

The remainder of this chapter is organized as follows. First, in Section 4.1, we generalize the concept of tempograms and describe the two variants. In Section 4.2, we introduce the novel concept of cyclic tempograms. In Section 4.3, we sketch various applications of the resulting mid-level representations and finally conclude in Section 4.4.

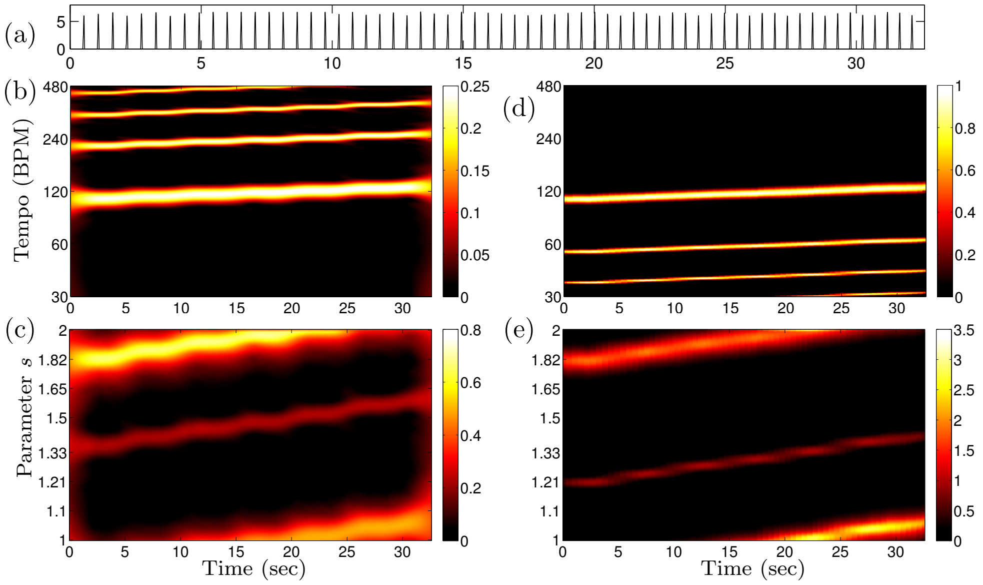


Figure 4.1: **(a)** Novelty curve of click track of increasing tempo (110 to 130 BPM). **(b)** Fourier tempogram (showing harmonics). **(c)** Cyclic tempogram $\mathcal{C}_{60}$ induced by (b). **(d)** Autocorrelation tempogram (showing subharmonics). **(e)** Cyclic tempogram $\mathcal{C}_{60}$ induced by (d).

4.1 Tempogram Representations

In general, a *tempogram* (similar to a spectrogram that is a time-frequency representation) is a time-tempo representation of a given time-dependent signal. Mathematically, a tempogram is a mapping $\mathcal{T} : \mathbb{R} \times \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ depending on a time parameter $t \in \mathbb{R}$ measured in seconds and a tempo parameter $\tau \in \mathbb{R}_{>0}$ measured in beats per minute (BPM). Intuitively, the value $\mathcal{T}(t, \tau)$ indicates to which extend a pulse of tempo τ is present at time t. For example, let us suppose that a music signal has a dominant tempo of 120 BMP around position $t = 20$ seconds. Then the resulting tempogram $\mathcal{T}$ should have a large value $\mathcal{T}(t, \tau)$ for $\tau = 120$ and $t = 20$. Because of the ambiguity concerning the pulse levels, one typically also has large values of $\mathcal{T}$ for integer multiples $\tau, 2\tau, 3\tau, \ldots$ (referred to as *harmonics* of τ) and integer fractions $\tau, \tau/2, \tau/3, \ldots$ (referred to as *subharmonics* of τ). For an illustration, we refer to Figure 4.1, which shows various tempograms for a click track of increasing tempo.

For computing tempograms, one typically first extracts a novelty curve Δ as introduced in Section 2.3. The peaks of this curve yield good indicators for note onsets. In a second step, local periodic patterns are derived from the novelty curve. Here, we discuss two different methods that yield tempograms with harmonics (Fourier tempogram, Section 4.1.1) and with subharmonics (autocorrelation tempogram, Section 4.1.2), respectively.

4.1.1 Fourier Tempogram

As first strategy, we analyze the novelty curve Δ with respect to local periodic patterns using a short-time Fourier transform. To this end, we fix a window function $W : \mathbb{Z} \to \mathbb{R}$

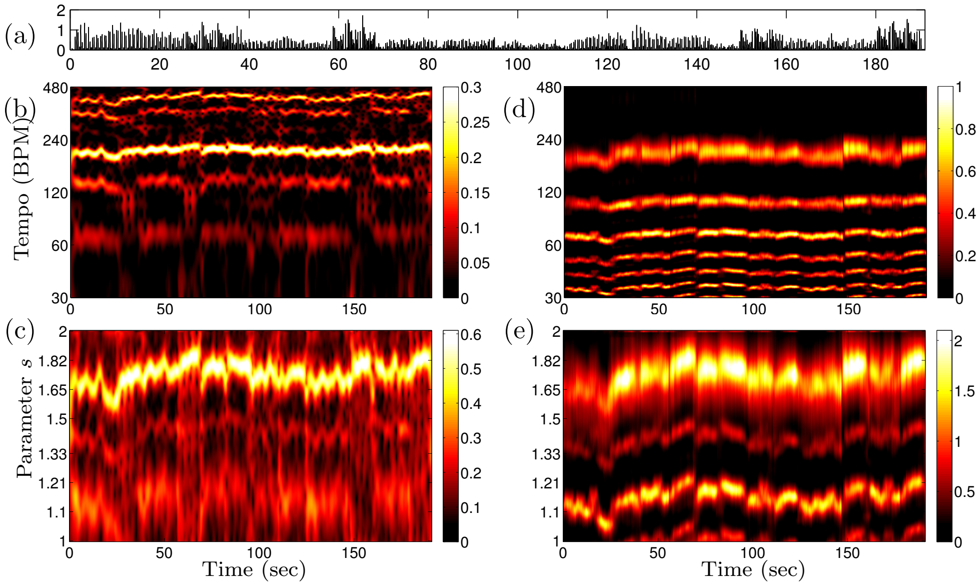


Figure 4.2: **(a)** Novelty curve Δ of an audio recording of a Waltz by Shostakovich. **(b)** Fourier tempogram $\mathcal{T}^{\mathrm{F}}$. **(c)** Cyclic tempogram $\mathcal{C}^{\mathrm{F}}_{60}$. **(d)** Autocorrelation tempogram $\mathcal{T}^{\mathrm{A}}$. **(e)** Cyclic tempogram $\mathcal{C}^{\mathrm{A}}_{60}$.

centered at $t = 0$ with support $[-N : N]$. In our experiments, we use a Hann window of size $2N + 1$ corresponding to six seconds of the audio recording. Then, for a frequency parameter $\omega \in \mathbb{R}_{\geq 0}$, the complex Fourier coefficient $\mathcal{F}(t, \omega)$ is defined by

$$\mathcal{F}(t,\omega) = \sum_{n \in \mathbb{Z}} \Delta(n) \cdot W(n-t) \cdot \mathrm{e}^{-2\pi i \omega n} \ . \tag{4.1}$$

In the musical context, we rather think of tempo measured in beats per minutes (BPM) than of frequency measured in Hertz (Hz). Therefore, we use a tempo parameter τ satisfying the equation $\tau = 60 \cdot \omega$. Furthermore, we compute the tempi only for a finite set $\Theta \subset \mathbb{R}_{>0}$. In our implementation, we cover four tempo octaves ranging from $\tau = 30$ to $\tau = 480$. Furthermore, we sample this interval in a logarithmic fashion covering each tempo octave by M samples, where the integer M determines the tempo resolution. Then, the discrete *Fourier tempogram* $\mathcal{T}^{\mathrm{F}} : \mathbb{Z} \times \Theta \to \mathbb{R}_{\geq 0}$ is defined by

$$\mathcal{T}^{\mathrm{F}}(t,\tau) = |\mathcal{F}(t, \tau/60)|. \tag{4.2}$$

As an example, Figure 4.2b shows the tempogram $\mathcal{T}^{\mathrm{F}}$ of a recording of a Waltz by Shostakovich. In $\mathcal{T}^{\mathrm{F}}$, the tempo on the beat level (roughly $\tau = 216$ BPM) and the second harmonics of this tempo are dominant. However, the tempo on the measure level of the three-quarter Waltz (roughly 72 BPM, third subharmonics of $\tau = 216$) is hardly noticeable. Actually, since the novelty curve Δ locally behaves like a track of positive clicks, it is not hard to see that Fourier analysis responds to harmonics but suppresses subharmonics, see also [145].

4.1.2 Autocorrelation Tempogram

In the context of tempo estimation, also autocorrelation-based methods are widely used to estimate local periodicities [44]. Since these methods, as it turns out, respond to subharmonics while suppressing harmonics, they ideally complement Fourier-based methods, see [145]. To obtain a discrete *autocorrelation tempogram*, we proceed as follows. Again, we fix a window function $W : \mathbb{Z} \to \mathbb{R}$ centered at $t = 0$ with support $[-N : N]$, $N \in \mathbb{N}$. This time, we use a box window of size $2N + 1$ corresponding to six seconds of the underlying music recording. The local autocorrelation is then computed by comparing the windowed novelty curve with time shifted copies of itself. More precisely, we use the unbiased local autocorrelation

$$\mathcal{A}(t, \ell) = \frac{\sum_{n \in \mathbb{Z}} \Delta(n) \cdot W(n-t)\Delta(n+\ell) \cdot W(n-t+\ell)}{2N + 1 - \ell}, \tag{4.3}$$

for time $t \in \mathbb{Z}$ and time lag $\ell \in [0 : N]$. Each time parameter $t \in \mathbb{Z}$ of the novelty curve corresponds to r seconds of the audio (in our implementation we used $r = 0.023$).

Then, the lag ℓ corresponds to the tempo

$$\tau = 60/(r \cdot \ell)$$

in BPM. We therefore define the *autocorrelation tempogram* $\mathcal{T}^{\mathrm{A}}$ by

$$\mathcal{T}^{\mathrm{A}}(t, \tau) = \mathcal{A}(t, \ell). \tag{4.4}$$

for each tempo $\tau = 60/(r \cdot \ell)$, $\ell \in [1 : N]$. Finally, using standard resampling and interpolation techniques applied to the tempo domain, we derive an autocorrelation tempogram $\mathcal{T}^{\mathrm{A}} : \mathbb{Z} \times \Theta \to \mathbb{R}_{\geq 0}$ that is defined on the same tempo set Θ as the Fourier tempogram $\mathcal{T}^{\mathrm{F}}$, see Section 4.1.1. The tempogram $\mathcal{T}^{\mathrm{A}}$ for our Shostakovich example is shown in Figure 4.2d, which clearly indicates the subharmonics. This fact is also illustrated by comparing the Fourier tempogram shown in Figure 4.2b and the autocorrelation tempogram shown in Figure 4.2d.

4.2 Cyclic Tempograms

The different pulse levels as present in the audio recordings and revealed by the tempograms (either harmonics or subharmonics) lead to octave confusions when determining absolute tempo information, see Section 2.1. To reduce the impact of such kind of tempo confusions, we apply a similar strategy as in the computation of chroma features [4]. Recall that two pitches having fundamental frequencies f_1 and f_2 are considered as octave equivalent, if they are related by $f_1 = 2^k f_2$ for some $k \in \mathbb{Z}$. Similarly, we say that two tempi τ_1 and τ_2 are *octave equivalent*, if they are related by $\tau_1 = 2^k \tau_2$ for some $k \in \mathbb{Z}$. Then, for a given tempo parameter τ, the resulting tempo equivalence class is denoted by $[\tau]$. For example, for $\tau = 120$ one has $[\tau] = \{\ldots, 30, 60, 120, 240, 480 \ldots\}$. Now, the *cyclic tempogram* $\mathcal{C}$ induced by $\mathcal{T}$ is defined by

$$\mathcal{C}(t, [\tau]) := \sum_{\lambda \in [\tau]} \mathcal{T}(t, \lambda) \ . \tag{4.5}$$

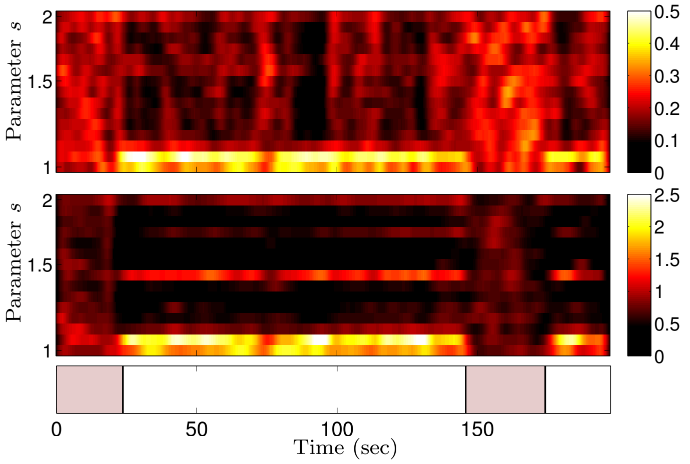


Figure 4.3: Cyclic tempogram $\mathcal{C}_{60}^{\mathrm{F}}$ (top) and $\mathcal{C}_{60}^{\mathrm{A}}$ (middle) with $M = 15$ as well as tempo-based segmentations (bottom) for *In The Year 2525* by Zager and Evans. Intro and interlude are annotated.

Note that the tempo equivalence classes topologically correspond to a circle. Fixing a reference tempo ρ (e.g., $\rho = 60$ BPM), the cyclic tempogram can be represented by a mapping $\mathcal{C}_\rho : \mathbb{R} \times \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ defined by

$$\mathcal{C}_\rho(t, s) := \mathcal{C}(t, [s \cdot \rho]), \tag{4.6}$$

for $t \in \mathbb{R}$ and $s \in \mathbb{R}_{>0}$. Note that $\mathcal{C}_\rho(t, s) = \mathcal{C}_\rho(t, 2^k s)$ for $k \in \mathbb{Z}$ and $\mathcal{C}_\rho$ is completely determined by its values $s \in [1, 2)$. Here, we use the representation $\mathcal{C}_\rho$ with $\rho = 60$. As illustration, Figure 4.1 shows various tempograms for a click track of increasing tempo with a tempo increasing from $\tau = 110$ to $\tau = 130$ BPM. Figure 4.1b shows a Fourier tempogram with harmonics and Figure 4.1c the resulting cyclic tempogram. As in the pitch context, the tempo class $[3\tau]$ is referred to as the *tempo dominant* and corresponds to the third harmonics 3τ. In Figure 4.1c, the tempo dominant is visible as the increasing line in the middle. Similarly, Figure 4.1d shows an autocorrelation tempogram with subharmonics and Figure 4.1e the resulting cyclic tempogram. Here, the tempo class $[\tau/3]$ is referred to as the *tempo subdominant* and corresponds to the third subharmonics $\tau/3$, see the increasing line in the middle of Figure 4.1e.

For computing cyclic tempograms, recall that the tempo parameter set Θ introduced in Section 4.1.1 comprises four tempo octaves ranging from $\tau = 30$ to $\tau = 480$, where each octave is covered by M logarithmically spaced samples. Therefore, one obtains a discrete cyclic tempogram $\mathcal{C}^{\mathrm{F}}$ (resp. $\mathcal{C}^{\mathrm{A}}$) from the tempogram $\mathcal{T}^{\mathrm{F}}$ (resp. $\mathcal{T}^{\mathrm{A}}$) simply by adding up the corresponding values of the four octaves as described in Eq. (4.5). Using a reference tempo of $\rho = 60$ BPM, we obtain the cyclic tempogram $\mathcal{C}_{60}^{\mathrm{F}}$ (resp. $\mathcal{C}_{60}^{\mathrm{A}}$). Note that these discrete cyclic tempograms are M-dimensional, where the cylic tempo axis is sampled at M positions. For our Shostakovich example, Figure 4.2c (resp. Figure 4.2e) shows the discrete cyclic tempogram $\mathcal{C}_{60}^{\mathrm{F}}$ (resp. $\mathcal{C}_{60}^{\mathrm{A}}$), where we used a time resolution of $r = 0.023$ seconds and a tempo resolution of $M = 120$. Note that the subharmonic tempo at measure level corresponding to roughly 72 BPM ($s = 1.2$) is clearly visible in $\mathcal{C}_{60}^{\mathrm{A}}$, but not in $\mathcal{C}_{60}^{\mathrm{F}}$.

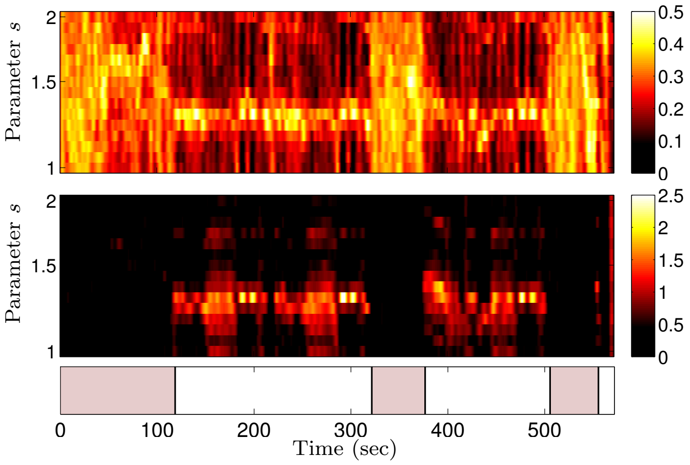


Figure 4.4: Cyclic tempogram $\mathcal{C}_{60}^{\mathrm{F}}$ (top) and $\mathcal{C}_{60}^{\mathrm{A}}$ (middle) with $M = 15$ as well as tempo-based segmentations (bottom) for Piano Sonata Op. 13 (Pathétique) by Beethoven performed by Barenboim. All *Grave* parts are annotated.

4.3 Applications to Music Segmentation

As mentioned before, the cyclic tempograms are the tempo-based counterparts of the harmony-based chromagrams. Compared to usual tempograms, the cyclic versions are more robust to tempo ambiguities that are caused by the various pulse levels. Furthermore, one can simulate changes in tempo simply by cyclically shifting a cyclic tempogram. Note that this is similar to the property of chromagrams, which can be cyclically shifted to simulate modulations in pitch. As one further advantage, even low-dimensional versions of discrete cyclic tempograms still bear valuable local tempo information of the underlying musical signal.

To illustrate the potential of our concept, we sketch how cyclic tempograms can be used for automated music segmentation, which is a central task in the field of music information retrieval [123; 91; 114]. Actually, there are many different strategies for segmenting music signals such as novelty-based, repetition-based, and homogeneity-based strategies. In the latter, the idea is to partition the music signal into segments that are homogenous with regard to a specific musical property [114]. In this context, timbre-related audio features such as MFCCs or spectral envelopes are frequently used, resulting in timbre-based segmentations. Similarly, using chroma-based audio features results in harmony-based segmentations. We now indicate, how our cyclic tempograms can be applied to obtain tempo-based segmentations (using a simple two-class clustering procedure for illustration). In the following examples, we use low-dimensional versions of $\mathcal{C}_{60}^{\mathrm{A}}$ and $\mathcal{C}_{60}^{\mathrm{F}}$ based on $M = 15$ different tempo classes. In our first example, we consider the song *In The Year 2525* by Zager and Evans. This song starts with a slow intro and contains a slow interlude of the same tempo. The remaining parts (basically eight repetitions of the chorus section) are played in a different, much faster tempo. As can be seen in Figure 4.3, both cyclic tempograms, $\mathcal{C}_{60}^{\mathrm{F}}$ and $\mathcal{C}_{60}^{\mathrm{A}}$, allow for separating the slow from the fast parts. As second

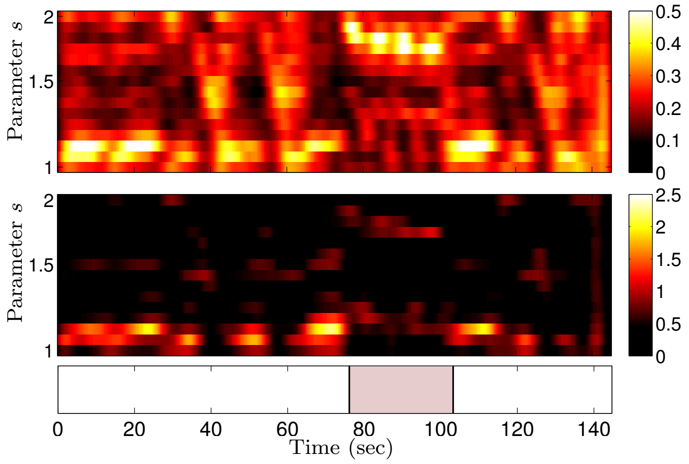


Figure 4.5: Cyclic tempogram $\mathcal{C}^{\mathrm{F}}_{60}$ (top) and $\mathcal{C}^{\mathrm{A}}_{60}$ (middle) with $M = 15$ as well as tempo-based segmentations (bottom) for Hungarian Dance No. 5 by Brahms.

example, we consider a recording of the first movement of Beethoven's Piano Sonata Op. 13 (Pathétique). After a dramatic *Grave* introduction, the piece continues with *Allegro di molto e con brio*. However, it returns twice to *Grave*—at the beginning of the development section as well as in the coda. Using a purely tempo-based segmentation, the occurrences of the three *Grave* sections can be recovered, see Figure 4.4. Here, in particular the autocorrelation tempogram $\mathcal{C}^{\mathrm{A}}_{60}$ yields a clear discrimination. Finally, as a third example, we consider a piano version of Brahms' Hungarian Dance No. 5, a piece with many abrupt changes in tempo. This property is well reflected by the cyclic tempograms shown in Figure 4.5. In particular, the Fourier tempogram $\mathcal{C}^{\mathrm{F}}_{60}$ separates well the slow middle part from the other, much faster parts.

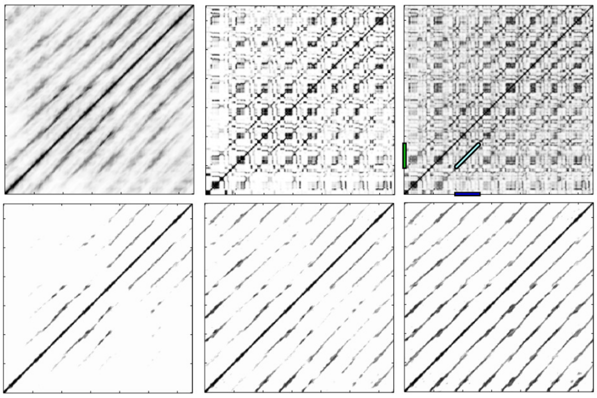
4.4 Further Notes

As shown in this chapter, tempogram representations constitute a class of powerful mid-level representations that reveal local tempo information for music with significant tempo changes while being invariant to extraction errors. Being the tempo-based counterpart of the harmony-based chromagrams, cyclic tempograms are suitable for music analysis and retrieval tasks, where harmony-based criteria are not relevant. In the three example discussed in this chapter, the cyclic tempograms yield musically meaningful segmentations purely based on a low-dimensional representation of tempo. Actually, these segments can not be recovered using MFCCs or chroma features, since the homogeneity assumption does not hold with regard to timbre or harmony.

For the future, one could integrate our concept of cyclic tempo features into a segmentation and structure extraction framework. In practice, various strategies based on different musical dimensions are needed to cope with the richness and diversity of music [91; 142]. In this context, our features reveal musically meaningful segmentation purely based on tempo information, while being invariant to rhythmic [92], harmonic [84], and timbre [114] properties. Furthermore, having low-dimensional tempo features (in the order of the 12-20 dimensions of chroma and MFCC features), makes it possible to employ index-based range and nearest neighbor searches, which is important in view of efficient music retrieval.

Part II

Music Segmentation



Chapter 5

Reference-Based Folk Song Segmentation

Generally, a folk song is referred to as a song that is sung by the common people of a region or culture during work or social activities. As a result, folk music is closely related to the musical culture of a specific nation or region. Since many decades, significant efforts have been carried out to assemble and study large collections of folk songs [95; 178; 94] which are not only part of the nations' cultural heritage but also allow musicologists to conduct folk song research on a large scale. Among others, researchers are interested in reconstructing and understanding the genetic relations between variants of folk songs as well as discovering musical connections and distinctions between different national or regional cultures [178; 95; 94].

Even though folk songs were typically transmitted only orally without any fixed symbolic notation, most of the folk song research is conducted on the basis of notated music material, which is obtained by transcribing recorded tunes into symbolic, score-based music representations. These transcriptions are often idealized and tend to represent the presumed intention of the singer rather than the actual performance. After the transcription, the audio recordings are often no longer studied in the actual research. This seems somewhat surprising, since one of the most important characteristics of folk songs is that they are part of oral culture. Therefore, one may conjecture that performance aspects enclosed in the recorded audio material are likely to bear valuable information, which is no longer contained in the transcriptions. Furthermore, even though the notated music material may be more suitable for classifying and identifying folk songs using automated methods, the user may want to listen to the original recordings rather than to synthesized versions of the transcribed tunes.

One reason for folk song researchers to focus on symbolic representations is that, due to its massive data volume and complexity, audio material is generally hard to deal with. In a specific folk song recording, musically relevant information such as the occurring notes (specified by musical onset times, pitches, and durations), the melody, or the rhythm are not given explicitly, but are somehow hidden in the waveform of the audio signal.

It is the object of this chapter to indicate how the original recordings can be made more easily accessible for folk song researches and listeners by bridging the gap between the symbolic and the audio domain. In particular, we present a procedure for automatically segmenting a given folk song recording that consists of several repetitions of the same tune into its individual stanzas. More precisely, for most folk songs a tune is repeated over and over again with changing lyrics. A typical field recording therefore consists of a sequence $A_1 A_2 \ldots A_K$ of stanzas A_k, $k \in [1:K] := \{1, 2, \ldots, K\}$, where each A_k corresponds to the same tune. Given a field recording, the segmentation task consists in identifying the temporal boundaries of the various stanzas. In this chapter, we introduce a reference-based segmentation algorithm that employs a manually transcribed reference stanza. The segmentation is then achieved by locally comparing the field recording with the reference stanza by means of a suitable distance function.

Main challenges arise from the fact that the folk songs are performed by elderly non-professional singers under poor recording conditions. The singers often deviate significantly from the expected pitches and have serious problems with the intonation. Even worse, their voices often fluctuate by several semitones downwards or upwards across the various stanzas of the same recording. As our main contribution, we introduce a combination of robust audio features along with various cleaning and audio matching strategies to account for such musical variations and inaccuracies. Our evaluation based on folk song recordings shows that we obtain a reliable segmentation even in the presence of strong musical variations.

The remainder of this chapter is organized as follows. In Section 5.1, we give an overview on computational folk song research and introduce the Dutch folk song collection *Onder de groene linde* (OGL). In Section 5.2, we give a short introduction to chroma features, which lay the basis for our analysis. Then, we describe a distance function for comparing chroma features (Section 5.3) and show how the segmentation of the audio recordings is derived (Section 5.4). In Section 5.5, as one main contribution of this chapter, we describe various enhancement strategies for achieving robustness to the aforementioned pitch fluctuations and recording artifacts. Then, in Section 5.6, we report on our systematic experiments conducted on the OGL collection. Finally, further notes are given in Section 5.7.

5.1 Background on Folk Song Research

Folk songs are typically performed by common people of a region or culture during work or recreation. These songs are generally not fixed by written scores but are learned and transmitted by listening to and participating in performance. Systematic research on folk song traditions started in the 19th century. At first researchers wrote down folk songs in music notation at performance time, but from an early date onwards performances were recorded using available technologies. Over more than a century of research, enormous amounts of folk song data have been assembled. Since the late 1990s, digitization of folk songs has become a matter of course. See [25] for an overview of European collections.

Digitized folk songs offer interesting challenges for computational research, and the availability of extensive folk song material requires computational methods for large-scale musicological investigation of this data. Much interdisciplinary research into such methods has been has been carried out within the context of music information retrieval (MIR).

An important challenge is to create computational methods that contribute to a better musical understanding of the repertoire [177]. For example, using computational methods, motivic relationships between different folk song repertoires are studied in [94]. Within individual traditions, the notion of tune family is important. Tune families consist of melodies that are considered to be historically related through the process of oral transmission. In the WITCHCRAFT project, computational models for tune families are investigated in order to create a melody search engine for Dutch folk songs [179; 186]. In the creation of such models aspects from music cognition play an important role. The representation of a song in human memory is not literal. During performance, the actual appearance of the song is recreated. Melodies thus tend to change over time and between performers. But even within a single performance of a strophic song interesting variations of the melody may be found.

By systematically studying entire collections of folk songs, researchers try to reconstruct and understand the genetic relation between variants of folk songs with the goal to discover musical connections and distinctions between different national or regional cultures [95; 178]. To support such research, several databases of encoded folk song melodies have been assembled, the best known of which is the Essen folk song database,[1] which currently contains roughly 20000 folk songs from a variety of sources and cultures. This collection has also been widely used in MIR research. For a survey of folk song research we refer to [178].

Even though folk songs are typically orally transmitted in performance, much of the research is conducted on the basis of notated musical material and leaves potentially valuable performance aspects enclosed in the recorded audio material out of consideration. However, various folk song collections contain a considerable amount of audio data, which has not yet been explored at a larger scale. An important step in unlocking such collections of orally transmitted folk songs is the creation of content-based search engines which allow users to browse and navigate within these collections on the basis of the different musical dimensions. The engines should enable a user to search for encoded data using advanced melodic similarity methods. Furthermore, it should also be possible to not only visually present the retrieved items, but also to supply the corresponding audio recordings for acoustic playback. One way of solving this problem is to create robust alignments between retrieved encodings (for example in MIDI format) and the audio recordings. The segmentation and annotation procedure described in the following exactly accomplishes this task.

Since folk songs are part of oral culture, one may conjecture that performance aspects enclosed in the recorded audio material are likely to bear valuable information, which is no longer contained in the transcriptions. Performance analysis has become increasingly important in musicological research and in music psychology. In folk song research (or more widely, in ethnomusicological research) computational methods are beginning to be applied to audio recordings as well. Examples are the study of African tone scales [122] and Turkish rhythms [87]. Comparing the various stanzas of a folksong allows for studying performance and melodic variation within a single performance of a folk song.

In the Netherlands, folk song ballads (strophic, narrative songs) have been extensively col-

[1] http://www.esac-data.org/

lected and studied. A long-term effort to record these songs was started by Will Scheepers in the early 1950s, and it was continued by Ate Doornbosch until the 1990s [70]. Their field recordings were usually broadcasted in the radio program *Onder de groene linde* (Under the green lime tree). Listeners were encouraged to contact Doornbosch if they knew more about the songs. Doornbosch would then record their version and broadcast it. In this manner a collection, in the following referred to as *OGL collection*, was created that not only represents part of the Dutch cultural heritage but also documents the textual and melodic variation resulting from oral transmission.

At the time of the recording, ballad singing had already largely disappeared from popular culture. Ballads were widely sung during manual work until the first decades of the 20th century. The tradition came to an end as a consequence of two innovations: the radio and the mechanization of manual labor. Decades later, when the recordings were made, the mostly female, elderly singers often had to delve deeply in their memories to retrieve the melodies. The effect is often audible in the recordings: there are numerous false starts, and it is evident that singers regularly began to feel comfortable about their performance only after a few strophes. Part of the effect may have been caused by the fact the recordings were generally made from solo performances at home, whereas the original performance setting would often have been a group of singers performing during work.

The OGL collection, which is currently hosted at the Meertens Institute in Amsterdam, is available through the *Nederlandse Liederenbank* (NLB)[2]. The database also gives access to very rich metadata, including date and location of recording, information about the singer, and classification by tune family and (textual) topic. The OGL collection contains 7277 audio recordings, which have been digitized as MP3 files (stereo, 160 kbit/s, 44.1 kHz). Nearly all of the field recordings are monophonic and comprise a large number of stanzas (often more than 10 stanzas). When the collection was assembled, melodies were transcribed on paper by experts. Usually only one stanza is given in music notation, but variants from other stanzas are regularly included. The transcriptions are often idealized and tend to represent the presumed intention of the singer rather than the actual performance. For a large number of melodies, transcribed stanzas are available in various symbolic formats including LilyPond[3] and Humdrum [158], from which MIDI representations have been generated (with a tempo set at 120 BPM for the quarter note). At this date, around 2500 folk songs from OGL have been encoded. In addition, the encoded corpus contains 1400 folk songs from written sources, and 1900 instrumental melodies from written, historical sources, bringing the total number of encoded melodies at approximately 5800. A detailed description of the encoded corpus is provided in [180].

5.2 Chroma-Based Audio Features

In our segmentation procedure, we assume that we are given a transcription of a reference tune in the form of a MIDI file. Recall from Section 5.1 that this is exactly the situation we have with the songs of the OGL collection. In the first step, we transform the MIDI reference as well as the audio recording into a common mid-level representation. Here, we

[2] Dutch Song Database, http://www.liederenbank.nl

[3] www.lilypond.org

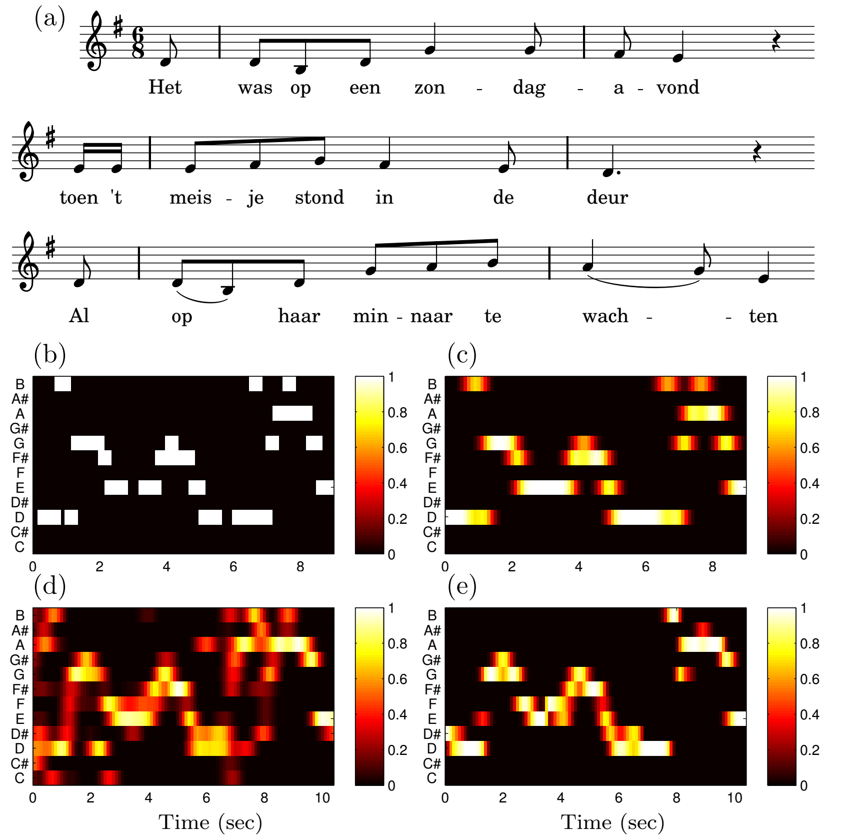


Figure 5.1: Representations of the beginning of the first stanza of the folk song OGL27517. **(a)** Score representation of the manually generated reference transcription. **(b)** Chromagram of the MIDI representation of the transcription. **(c)** Smoothed MIDI chromagram (CENS feature). **(d)** Chromagram of an audio recording (CENS feature). **(e)** F0-enhanced chromagram as will be introduced as first enhancement strategy in Section 5.5.

use the well-known chroma representation, as described in this section.

Chroma features have turned out to be a powerful mid-level representation for relating harmony-based music, see [4; 6; 90; 123; 143; 162; 164]. Assuming the equal-tempered scale, the term *chroma* refers to the elements of 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. Note that in the equal-tempered scale, different pitch spellings such $\mathrm{C}^\sharp$ and $\mathrm{D}^\flat$ refer to the same chroma. 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. Representing the short-time energy content of the signal in each of the 12 pitch classes, chroma features do not only account for the close octave relationship in both melody and harmony as it is prominent in Western music, but also introduce a high degree of robustness to variations in timbre and articulation [4]. Furthermore, normalizing the features makes them invariant to dynamic variations.

It is straightforward to transform a MIDI representation into a chroma representation or *chromagram*. Using the explicit MIDI pitch and timing information one basically identifies

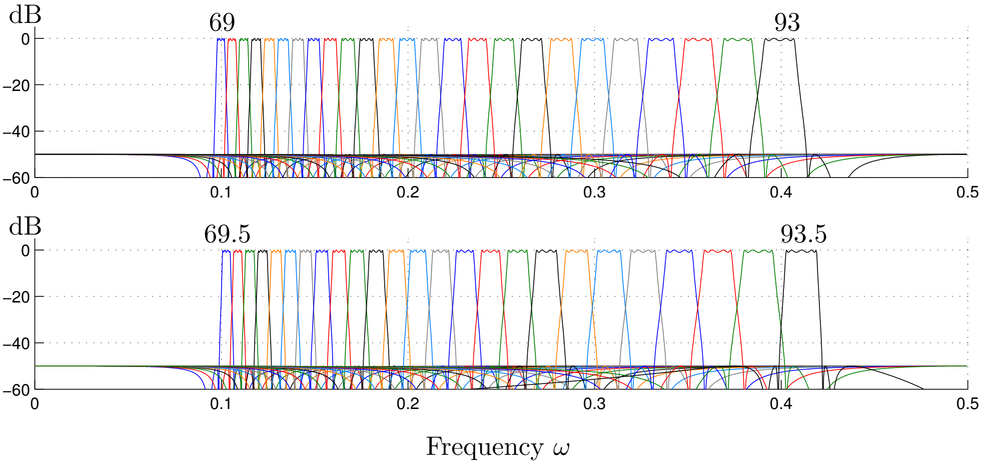


Figure 5.2: Magnitude responses in dB for some of the pitch filters of the multirate pitch filter bank used for the chroma computation. **Top:** Filters corresponding to MIDI pitches $p \in [69 : 93]$ (with respect to the sampling rate 4410 Hz). **Bottom:** Filters shifted half a semitone upwards.

pitches that belong to the same chroma class within a sliding window of a fixed size, see [90]. Disregarding information on dynamics, we derive a binary chromagram assuming only the values 0 and 1.[4] Furthermore, dealing with monophonic tunes, one has for each frame at most one non-zero chroma entry that is equal to 1. Figure 5.1 shows various representations for the folk song OGL27517. Figure 5.1b shows a chromagram of a MIDI reference corresponding to the score shown in Figure 5.1a. In the following, the chromagram of the reference transcription is referred to as *reference chromagram* or *MIDI chromagram*.

For transforming an audio recording into a chromagram, one has to revert to signal processing techniques. There are many ways of computing and enhancing chroma features, which results in a large number of chroma variants with different properties [4; 58; 59; 123]. Most chroma implementations are based on short-time Fourier transforms in combination with binning strategies [4; 59]. We use chroma features obtained from a pitch decomposition using a multirate filter bank as described in [123]. A given audio signal is first decomposed into 88 frequency bands with center frequencies f_p corresponding to the pitches $A0$ to $C8$ (MIDI pitches $p = 21$ to $p = 108$), where

$$f_p = 440 \text{ Hz} \cdot 2^{\frac{p}{12}} \ . \tag{5.1}$$

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 measure the local energy content of each pitch subband and indicate the presence of certain musical notes within the audio signal, see [123] for further details.

The employed pitch filters possess a relatively wide passband, while still properly separating adjacent notes thanks to sharp cutoffs in the transition bands, see Figure 5.2.

[4] Information about note intesities is not captured by the reference transcriptions.

Actually, the pitch filters are robust to deviations of up to ±25 cents[5] from the respective note's center frequency. The pitch filters will play an important role in the following sections. We then obtain a chroma representation by simply adding up the corresponding values that belong to the same chroma. To archive invariance in dynamics, chroma vectors are normalized with respect to the Euclidean norm (signal energy). The resulting chroma features are further processed by applying suitable quantization, smoothing, and downsampling operations resulting in some enhanced chroma features referred to as CENS (Chroma Energy Normalized Statistics). An implementation of these features is available online[6] and described in [127]. Adding a further degree of abstraction by considering short-time statistics over energy distributions within the chroma bands, CENS features constitute a family of scalable and robust audio features and have turned out to be very useful in audio matching and retrieval applications [135; 105]. These features allow for introducing a temporal smoothing. To this end, 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 [123] for details. In our experiments, we average feature vectors over a window corresponding to one second of the audio and a feature resolution of 10 Hz (10 features per second). Figure 5.1c shows the resulting smoothed version of the reference (MIDI) chromagram shown in Figure 5.1b. Figure 5.1d shows the final smoothed chromagram (CENS) for one of the five stanzas of the audio recording. For technical details, we refer to the cited literature.

5.3 Distance Function

On the basis of the chroma representations, the idea is to locally compare the reference with the audio recording by means of a suitable distance function. This distance function expresses the distance of the MIDI reference chromagram with suitable subsegments of the audio chromagram while being invariant to temporal variations between the reference and the various stanzas of the audio recording. More precisely, let $X = (x_1, x_2, \ldots, x_K)$ be the sequence of chroma features obtained from the MIDI reference and let $Y = (y_1, y_2, \ldots, y_L)$ be the one obtained from the audio recording as explained in Section 5.2. The resulting features $X(k) := x_k$, $k \in [1:K] := \{1, 2, \ldots, K\}$, and $Y(\ell) := y_\ell$, $\ell \in [1:L]$, are normalized 12-dimensional vectors. We define the distance function $\Delta := \Delta_{X,Y} : [1:L] \to \mathbb{R} \cup \{\infty\}$ with respect to X and Y using a variant of dynamic time warping (DTW):

$$\Delta(\ell) := \frac{1}{K} \min_{a \in [1:\ell]} \Big(\mathrm{DTW}\big(X\,,\, Y(a:\ell)\big) \Big), \tag{5.2}$$

where $Y(a:\ell)$ denotes the subsequence of Y starting at index a and ending at index $\ell \in [1:L]$. Furthermore, $\mathrm{DTW}(X, Y(a:\ell))$ denotes the DTW distance between X and $Y(a:\ell)$ with respect to a suitable local cost measure (in our case, the cosine distance). The distance function Δ can be computed efficiently using dynamic programming. For details on DTW and the distance function, we refer to [123]. The interpretation of Δ is as follows: a small value $\Delta(\ell)$ for some $\ell \in [1:L]$ indicates that the subsequence of

[5] The *cent* is a logarithmic unit to measure musical intervals. The semitone interval of the equally-tempered scale equals 100 cents.

[6] www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/

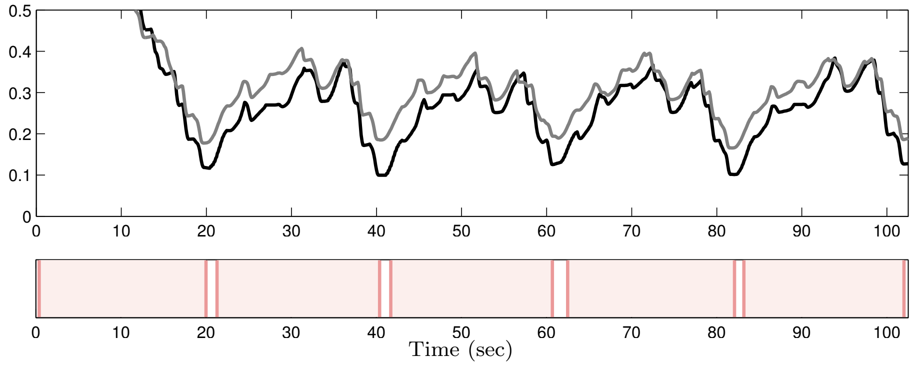


Figure 5.3: Top: Distance function Δ for OGL27517 using original chroma features (gray) and F0-enhanced chroma features (black). **Bottom:** Resulting segmentation.

Y starting at index a_ℓ (with $a_\ell \in [1 : \ell]$ denoting the minimizing index in Eq. (5.2)) and ending at index ℓ is similar to X. Here, the index a_ℓ can be recovered by a simple backtracking algorithm within the DTW computation procedure. The distance function Δ for OGL27517 is shown in Figure 5.3 as gray curve. The five pronounced minima of Δ indicate the endings of the five stanzas of the audio recording.

5.4 Segmentation of the Audio Recording

Recall that we assume that a folk song audio recording basically consists of a number of repeating stanzas. Exploiting the existence of a MIDI reference and assuming the repetitive structure of the recording, we apply the following simple greedy segmentation strategy. Using the distance function Δ, we look for the index $\ell \in [1 : L]$ minimizing Δ and compute the starting index a_ℓ. Then, the interval $S_1 := [a_\ell : \ell]$ constitutes the first *segment*. The value $\Delta(\ell)$ is referred to as the *cost* of the segment. To avoid large overlaps between the various segments to be computed, we exclude a neighborhood $[L_\ell : R_\ell] \subset [1 : L]$ around the index ℓ from further consideration. In our strategy, we set $L_\ell := \max(1, \ell - \frac{2}{3}K)$ and $R_\ell := \min(L, \ell + \frac{2}{3}K)$, thus excluding a range of two thirds of the reference length to the left as well as to the right of ℓ. To achieve the exclusion, we modify Δ simply by setting $\Delta(m) := \infty$ for $m \in [L_\ell : R_\ell]$. To determine the next segment S_2, the same procedure is repeated using the modified distance function, and so on. This results in a sequence of segments $S_1, S_2, S_3, \ldots$. The procedure is repeated until all values of the modified Δ lie above a suitably chosen *quality threshold* $\tau > 0$. Let N denote the number of resulting segments, then $S_1, S_2, \ldots, S_N$ constitutes the final segmentation result, see Figure 5.3 for an illustration.

5.5 Enhancement Strategies

This basic segmentation approach works well as long as the singer roughly follows the reference tune and stays in tune. However, for the field recordings, this is an unrealistic assumption. In particular, most singers have significant problems with the intonation.

Their voices often fluctuate by several semitones downwards or upwards across the various stanzas of the same recording. In this section, we show how the segmentation procedure can be improved to account for poor recording conditions, intonation problems, and pitch fluctuations.

Recall that the comparison of the MIDI reference and the audio recording is performed on the basis of chroma representations. Therefore, the segmentation algorithm described so far only works well in the case that the MIDI reference and the audio recording are in the same musical key. Furthermore, the singer has to stick roughly to the pitches of the well-tempered scale. Both assumptions are violated for most of the songs. Even worse, the singers often fluctuate with their voice by several semitones within a single recording. This often leads to poor local minima or even completely useless distance functions as illustrated Figure 5.4. To deal with local and global pitch deviations as well as with poor recording conditions, we use a combination of various enhancement strategies.

5.5.1 F0-Enhanced Chromagrams

In our first strategy, we enhance the quality of the chroma features similar to [59; 46] by picking only dominant spectral coefficients, which results in a significant attenuation of noise components. Dealing with monophonic music, we can go even one step further by only picking spectral components that correspond to the fundamental frequency (F0). More precisely, we use a modified autocorrelation method as suggested in [32] to estimate the fundamental frequency for each audio frame. For each frame, we then determine the MIDI pitch $p \in [1 : 120]$ having center frequency

$$f_p = 2^{\frac{p-69}{12}} \cdot 440 \text{ Hz}$$

that is closest to the estimated fundamental frequency. Next, in the pitch decomposition used for the chroma computation (as explained in Section 5.2), we assign energy only to the pitch subband that corresponds to the determined MIDI pitch—all other pitch subbands are set to zero within this frame. Finally, the resulting sparse pitch representation is projected onto a chroma representation and smoothed as explained in Section 5.2. The F0-based pitch assignment is capable of suppressing most of the noise resulting from poor recording conditions and local pitch deviations caused by the singers' intonation problems as well as vibrato are compensated to a substantial degree. The cleaning effect on the resulting chromagram, which is also referred to as *F0-enhanced chromagram*, is illustrated by Figure 5.1e, showing the F0-enhanced variant of the audio chromagram (see Figure 5.1d). This enhancement strategy leads to audio chromagrams that exhibit a much higher similarity to the reference chromagram (see Figure 5.1c) than the original chromagrams. As a result, the desired local minima of the distance function Δ, which are crucial in our segmentation procedure, become more pronounced. This effect is also illustrated by the distance functions shown in Figure 5.3.

Even though the folk song recordings are monophonic, the F0 estimation is often not accurate enough in view of applications such as automated transcription. However, using chroma representations, octave errors as typical in F0 estimations become irrelevant.

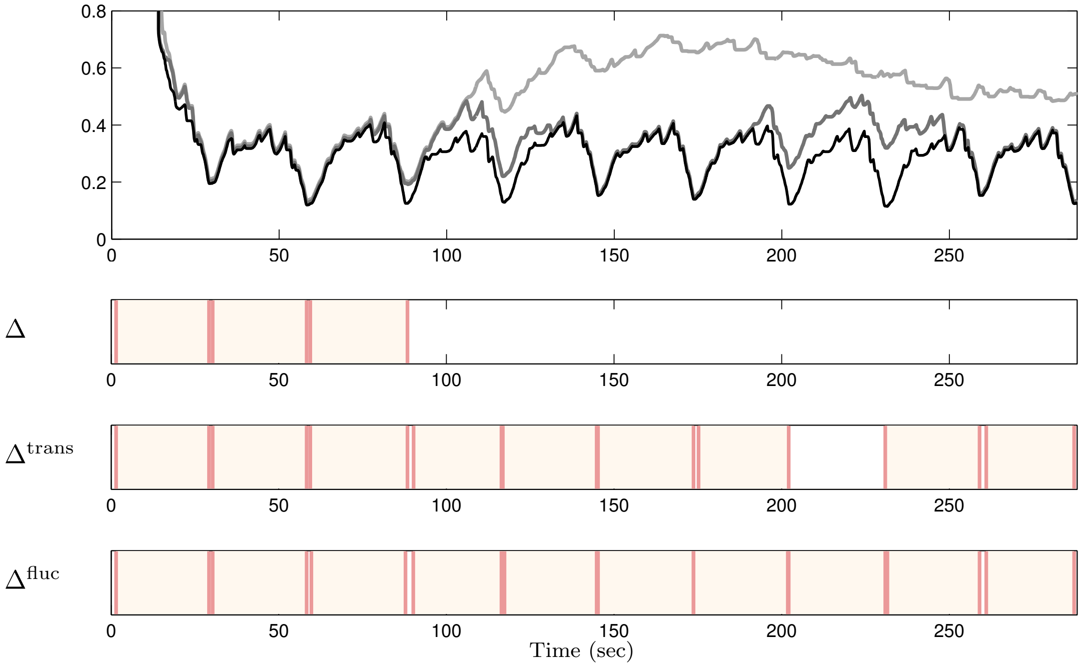


Figure 5.4: Distance functions Δ (light gray), Δ^{trans} (dark gray), and Δ^{fluc} (black) as well as the resulting segmentations for the song OGL25010.

5.5.2 Transposition-Invariant Distance Function

Next, we show how to deal with global pitch deviations and continuous fluctuation across several semitones. To account for a global difference in key between the MIDI reference and the audio recording, we revert to the observation by Goto [62] that the twelve cyclic shifts of a 12-dimensional chroma vector naturally correspond to the twelve possible transpositions. Therefore, it suffices to determine the shift index that minimizes the chroma distance of the audio recording and MIDI reference and then to cyclically shift the audio chromagram according to this index. Note that instead of shifting the audio chromagram, one can also shift the MIDI chromagram in the inverse direction. The minimizing shift index can be determined either globally by using averaged chroma vectors as suggested in [162] or locally by computing twelve different distance functions for the twelve shifts, which are then minimized to obtain a single transposition-invariant distance functions. We detail on the latter strategy, since it also solves part of the problem having a fluctuating voice within the audio recording. A similar strategy was used in [124] to achieve transposition invariance for music structure analysis tasks.

We simulate the various pitch shifts by considering all twelve possible cyclic shifts of the MIDI reference chromagram. We then compute a separate distance function for each of the shifted reference chromagrams and the original audio chromagram. Finally, we minimize the twelve resulting distance functions, say $\Delta^0, \ldots, \Delta^{11}$, to obtain a single *transposition-invariant* distance function $\Delta^{\mathrm{trans}} : [1 : L] \to \mathbb{R} \cup \{\infty\}$:

$$\Delta^{\mathrm{trans}}(\ell) := \min_{i \in [0:11]} \Big(\Delta^i(\ell)\Big). \tag{5.3}$$

Stanza	1	2	3	4	5	6	7	8	9	10
12 shift	5	5	5	4	4	4	4	3	3	3
24 shift	5.0	5.0	4.5	4.5	4.0	4.0	3.5	3.5	3.0	3.0

Table 5.1: Shift indices (cyclically shifting the audio chromagrams upwards) used for transposing the various stanzas of the audio recording of OGL25010 to optimally match the MIDI reference, see also Figure 5.4. The shift indices are given in semitones (obtained by Δ^{trans}) and in half semitones (obtained by Δ^{fluc}).

Figure 5.4 shows the resulting function Δ^{trans} for a folk song recording with strong fluctuations. In contrast to the original distance function Δ, the function Δ^{trans} exhibits a number of significant local minima that correctly indicate the segmentation boundaries of the stanzas.

5.5.3 Fluctuation-Invariant Distance Function

So far, we have accounted for transpositions that refer to the pitch scale of the equal-tempered scale. However, the above mentioned voice fluctuation are fluent in frequency and do not stick to a strict pitch grid. Recall from Section 5.2 that our pitch filters can cope with fluctuations of up to ±25 cents. To cope with pitch deviations between 25 and 50 cents, we employ a second filter bank, in the following referred to as *half-shifted filter bank*, where all pitch filters are shifted by half a semitone (50 cents) upwards, see Figure 5.2. Using the half-shifted filter bank, one can compute a second chromagram, referred to as *half-shifted chromagram*. A similar strategy is suggested in [59; 162] where generalized chroma representations with 24 or 36 bins (instead of the usual 12 bins) are derived from a short-time Fourier transform. Now, using the original chromagram as well as the half-shifted chromagram in combination with the respective 12 cyclic shifts, one obtains 24 different distance functions in the same way as described above. Minimization over the 24 functions yields a single function Δ^{fluc} referred to as *fluctuation-invariant distance function*. The improvements achieved by this novel distance function are illustrated by Figure 5.4. In regions with a bad intonation, the local minima of Δ^{fluc} are much more significant than those of Δ^{trans}. Table 5.1 shows the optimal shift indices derived from the transposition and fluctuation-invariant strategies, where the decreasing indices indicate to which extend the singer's voice rises across the various stanzas of the song.

5.6 Experiments

Our evaluation is based on a dataset consisting of 47 representative folk song recordings selected from the OGL collection described in Section 5.1. The audio dataset has a total length of 156 minutes, where each of the recorded song consists of 4 to 34 stanzas amounting to a total number of 465 stanzas. The recordings reveal significant deteriorations concerning the audio quality as well as the singer's performance. Furthermore, in various recordings, the tunes are overlayed with sounds such as ringing bells, singing birds, or barking dogs, and sometimes the songs are interrupted by remarks of the singers.

Strategy	F0	P	R	F	α	β	γ
Δ	$-$	0.898	0.628	0.739	0.338	0.467	0.713
Δ	$+$	0.884	0.688	0.774	0.288	0.447	0.624
Δ^{trans}	$-$	0.866	0.817	0.841	0.294	0.430	0.677
Δ^{trans}	$+$	0.890	0.890	0.890	0.229	0.402	0.559
Δ^{fluc}	$-$	0.899	0.901	0.900	0.266	0.409	0.641
Δ^{fluc}	$+$	0.912	0.940	0.926	0.189	0.374	0.494

Table 5.2: Performance measures for the reference-based segmentation procedure using the tolerance parameter $\delta = 2$ and the quality threshold $\tau = 0.4$. The second column indicates whether original ($-$) or F0-enhanced ($+$) chromagrams are used.

We manually annotated all audio recordings by specifying the segment boundaries of the stanzas' occurrences in the recordings. Since for most cases the end of a stanza more or less coincides with the beginning of the next stanza and since the beginnings are more important in view of retrieval and navigation applications, we only consider the starting boundaries of the segments in our evaluation. In the following, these boundaries are referred to as *ground truth boundaries*.

To assess the quality of the final segmentation result, we use precision and recall values. To this end, we check to what extent the 465 manually annotated stanzas within the evaluation dataset have been identified correctly by the segmentation procedure. More precisely, we say that a computed starting boundary is a *true positive*, if it coincidences with a ground truth boundary up to a small tolerance given by a parameter δ measured in seconds. Otherwise, the computed boundary is referred to as a *false positive*. Furthermore, a ground truth boundary that is not in a δ-neighborhood of a computed boundary is referred to as a *false negative*. We then compute the precision P and the recall R for the set of computed boundaries with respect to the ground truth boundaries. From these values one obtains the F-measure

$$\mathrm{F} := 2 \cdot \mathrm{P} \cdot \mathrm{R}/(\mathrm{P} + \mathrm{R}) \; .$$

Table 5.2 shows the PR-based performance measures of our reference-based segmentation procedure using different distance functions with original as well as F0-enhanced chromagrams. In this first experiment, the tolerance parameter is set to $\delta = 2$ and the quality threshold to $\tau = 0.4$. Here, a tolerance of up to $\delta = 2$ seconds seems to us an acceptable deviation in view of our intended applications and the accuracy of the annotations. For example, the most basic distance function Δ with original chromagrams yields an F-measure of $\mathrm{F} = 0.739$. Using F0-enhanced chromagrams instead of the original ones results in $\mathrm{F} = 0.774$. The best result of $\mathrm{F} = 0.926$ is obtained when using Δ^{fluc} with F0-enhanced chromagrams. Note that all of our introduced enhancement strategies result in an improvement in the F-measure. In particular, the recall values improve significantly when using the transposition and fluctuation-invariant distance functions.

A manual inspection of the segmentation results showed that most of the false negatives as well as false positives are due to deviations in particular at the stanzas' beginnings. The entry into a new stanza seems to be a problem for some of the singers, who need some seconds before getting stable in intonation and pitch. A typical example is NLB72355.

δ	P	R	F
1	0.637	0.639	0.638
2	0.912	0.940	0.926
3	0.939	0.968	0.953
4	0.950	0.978	0.964
5	0.958	0.987	0.972

τ	P	R	F
0.1	0.987	0.168	0.287
0.2	0.967	0.628	0.761
0.3	0.950	0.860	0.903
0.4	0.912	0.940	0.926
0.5	0.894	0.944	0.918

Table 5.3: Dependency of the PR-based performance measures on the tolerance parameter δ and the quality threshold τ. All values refer to the reference-based segmentation procedure with Δ^{fluc} using F0-enhanced chromagrams. **Left:** PR-based performance measures for various δ and fixed $\tau = 0.4$. **Right:** PR-based performance measures for various τ and fixed $\delta = 2$.

Increasing the tolerance parameter δ, the average quality improves substantially, as indicated by Table 5.3 (left). For example, using $\delta = 3$ instead of $\delta = 2$, the F-measure increase from $F = 0.926$ to $F = 0.953$. Other sources of error are that the transcriptions sometimes differ significantly from what is actually sung, as is the case for NLB72395. Here, as was already mentioned in Section 5.1, the transcripts represent the presumed intention of the singer rather than the actual performance. Finally, structural differences between the various stanzas are a further reason for segmentation errors. In a further experiment, we investigated the role of the quality threshold τ (see Section 5.4) on the final segmentation results, see Table 5.3 (right). Not surprisingly, a small τ yields a high precision and a low recall. Increasing τ, the recall increases at the cost of a decrease in precision. The value $\tau = 0.4$ was chosen, since it constitutes a good trade-off between recall and precision.

Finally, to complement our PR-based evaluation, we introduce a second type of more softer performance measures that indicate the significance of the desired minima of the distance functions. To this end, we consider the distance functions for all songs with respect to a fixed strategy and chroma type. Let α be the average over the cost of all ground truth segments (given by the value of the distance function at the corresponding ending boundary). Furthermore, let β be the average over all values of all distance functions. Then the quotient $\gamma = \alpha/\beta$ is a weak indicator on how well the desired minima (the desired true positives) are separated from possible irrelevant minima (the potential false positives). A low value for γ indicates a good separability property of the distance functions. As for the PR-based evaluation, the soft performance measures shown in Table 5.2 support the usefulness of our enhancement strategies.

5.7 Further Notes

The reference-based segmentation procedure provides robust segmentation results even in the case of strong musical variations in the stanzas. As main ingridient, we introduced enhancement strategies for dealing with the special characteristics of the folk song recordings performed by elderly non-professional solo singers: F0-enhanced chromagrams for efficiently reducing background noise as well as transposition-invariant and fluctuation-invariant chromagrams for handling local transpositions and pitch shifts. However, the presented procedure crucially depends on the availability of a manually generated reference transcription. Recall from Chapter 5.1 that for the 7277 audio recordings contained in OGL, only 2500 are transcribed so far. For other folk song datasets, the situation is even worse. In Chapter 6, we deal with the question on how the segmentation can be done if no MIDI reference is available.

Chapter 6

Reference-Free Folk Song Segmentation

In this chapter, we introduce a reference-free segmentation procedure that does not rely on any reference, thus overcoming the limitations of the reference-based approach introduced in the preceding chapter. Our idea is to apply a recent audio thumbnailing approach described in [129] to identify the most "repetitive" segment in a given recording. This so-called *thumbnail* then takes over the role of the reference. The thumbnailing procedure is built upon suitable audio features and self-similarity matrices (SSM). To cope with the aforementioned variations, we introduce various enhancement strategies to absorb a high-degree of these deviations and deformations already on the feature and SSM level. The evaluation shows that the segmentation results of the reference-free approach are comparable to the ones obtained from the reference-based segmentation procedure introduced in Chapter 5.

The remainder of this chapter is organized as follows. We first describe the self-similarity matrices (Section 6.1). Then, we summarize the audio thumbnailing procedure and explain how the segmentation is obtained (Section 6.2). In Section 6.3, as main contribution of this chapter, we introduce various strategies for enhancing the self-similarity matrices. We report on our segmentation experiments (Section 6.4) and conclude in Section 6.5.

6.1 Self-Similarity Matrices

Most repetition-based approaches to audio structure analysis proceed as follows. In the first step, the music recording is transformed into a sequence $X := (x_1, x_2, \ldots, x_N)$ of feature vectors $x_n \in \mathcal{F}$, $1 \leq n \leq N$, where $\mathcal{F}$ denotes a suitable feature space. We employ chroma features as introduced in Section 5.2. In the second step, based on a similarity measure $\mathbf{s} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$, one obtains an $N \times N$ *self-similarity matrix* (SSM) by comparing the elements of X in a pairwise fashion:

$$\mathcal{S}(n, m) := \mathbf{s}(x_n, x_m),$$

for $n, m \in [1 : N]$. Using normalized features vectors, we simply use the inner product as similarity measure $\mathbf{s}$ yielding a value between 0 and 1 (cosine measure). In the following, a tuple $p = (n, m) \in [1 : N]^2$ is called a *cell* of $\mathcal{S}$, and the value $\mathcal{S}(n, m)$ is referred to as the *score* of the cell p. Introduced to the music context in [53], such matrices have turned out to be a powerful tool for revealing repeating patterns of X. The crucial observation is that each diagonal path (or stripe) of high similarity running in parallel to the main diagonal of $\mathcal{S}$ indicates the similarity of two audio segments (given by the projections of the path onto the vertical and horizontal axis, respectively), see [143].

For example, Figure 6.1a shows an SSM for the first eight stanzas $A_1A_2A_3A_4A_5A_6A_7A_8$ of the field recording OGL19101. The highlighted path encodes the similarity between A_2 and A_3. If the eight segments would be close to being exact repetitions, one would expect a "full" path structure as indicated by Figure 6.1f. However, due to the spectral and temporal deviations between the sung stanzas, the path structure is in general highly distorted and fragmentary. In Section 6.3, we introduce various enhancement strategies to improve on the path structure of the SSM.

6.2 Audio Thumbnailing and Segmentation Procedure

In view of our folk song segmentation task, the enhancement of the self-similarity is one main step in order to achieve robustness to spectral deviations. To deal we temporal deviations, we apply a segmentation approach as proposed in [129]. Since in our scenario the recording basically consists of repetitions of a single tune, the segmentation problem reduces to a thumbnailing problem. In general, the goal of *audio thumbnailing* is to find the most representative and repetitive segment of a given music recordings, see, e. g., [4; 22; 115]. Typically, such a segment should have many (approximate) repetitions, and these repetitions should cover large parts of the recording. Let

$$\alpha = [s : t] \subseteq [1 : N]$$

denote a segment specified by its starting point s, end point t, and length $|\alpha| := t - s + 1$. In [129], a fitness measure is introduced that assigns to each audio segment α a fitness value $\varphi(\alpha) \in \mathbb{R}$ that simultaneously captures two aspects. Firstly, it indicates how well the given segment explains other related segments and, secondly, it indicates how much of the overall music recording is covered by all these related segments. The audio thumbnail is then defined to be the segment α^* having maximal fitness φ over all possible segments.

In the computation of the fitness measure, the main technical idea is to assign to each audio segment α a so-called *optimal path family* over α that simultaneously reveals the relations between α and all other similar segments. Figure 6.2 shows two segments along with their optimal path families, which can be computed efficiently using dynamic programming. One main point is that each path family projected to the vertical axis induces a family of segments, where each element of this family defines a segment similar to α. The induced family of segments then defines a segmentation of the audio recording.

As an example, Figure 6.2 shows path families and induced segment families (vertical axis) for two different segments (horizontal axis) for our running example OGL19101. In

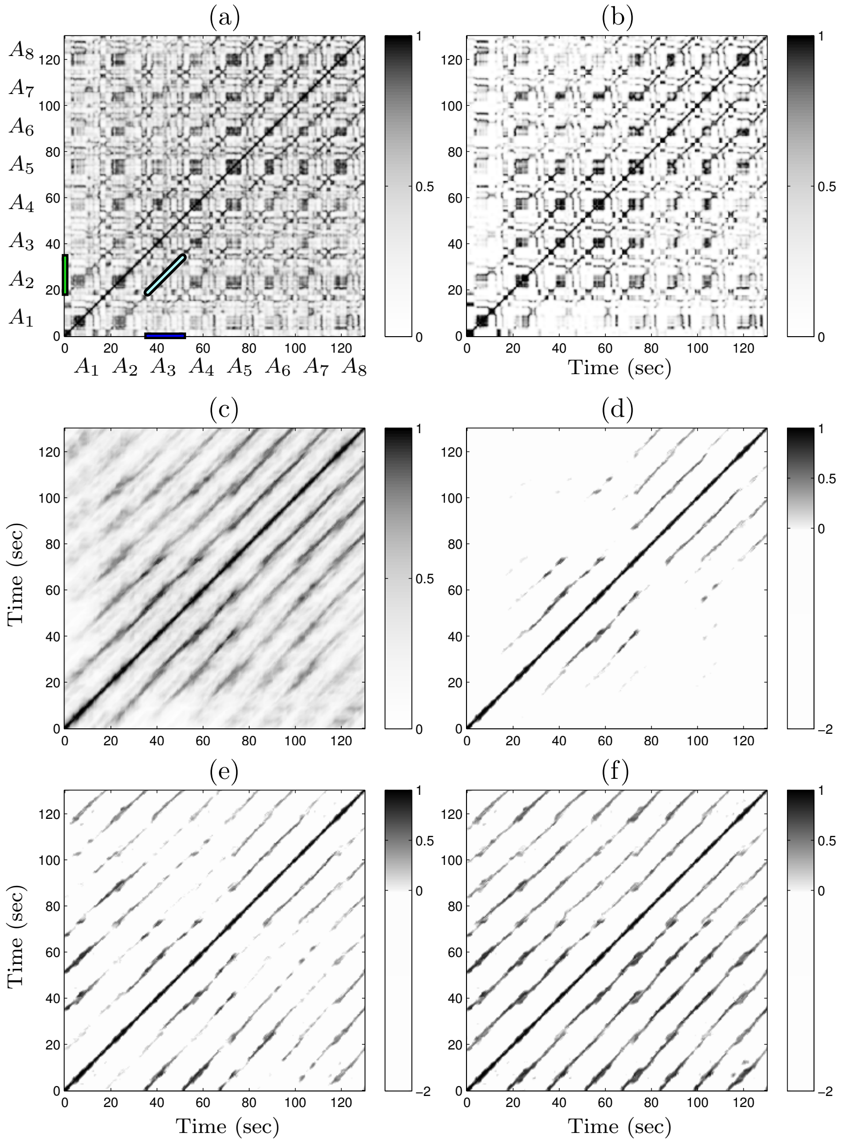


Figure 6.1: Self similarity matrices for the first eight stanzas of the folk song OGL19101. **(a)** SSM computed from CENS features. The highlighted path encodes the similarity of A_3 and A_2. **(b)** SSM computed from F0-enhanced CENS features. **(c)** Path-enhanced SSM. **(d)** Thresholded and normalized SSM $\mathcal{S}$. **(e)** Transposition-invariant SSM $\mathcal{S}^{\text{trans}}$. **(f)** Fluctuation-invariant SSM $\mathcal{S}^{\text{fluc}}$.

Figure 6.2a the segment is $\alpha = [83:98]$, which corresponds to the sixth stanza A_6. The induced segment family consists of eight different segments, which correspond to the eight stanzas $A_1, A_2, \ldots, A_8$. Figure 6.2b shows the path family and induced segment family for $\alpha = [66:98]$, which corresponds to the two subsequent stanzas A_5A_6. Here, the induced segment family consists of four segments corresponding to A_1A_2, A_3A_4, A_5A_6, and A_7A_8. The fitness value of a given segment is derived from the corresponding path family and the values of the underlying SSM. It is designed to slightly favor shorter segments to longer segments, see [129] for further details. In our example, it turns out that the fitness-

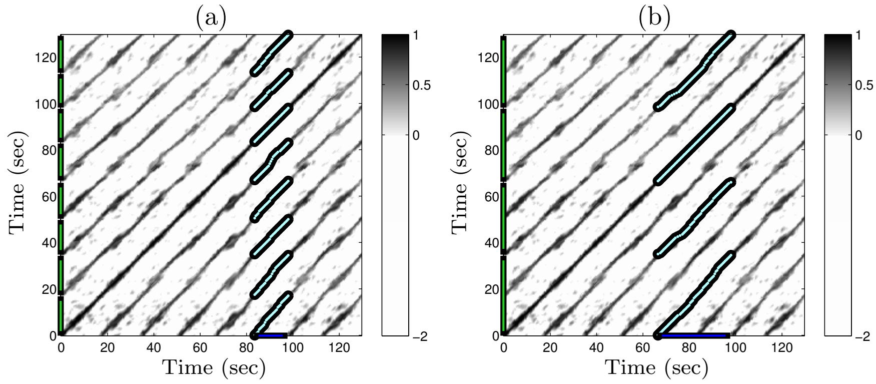


Figure 6.2: Path families and induced segment families for two different segments α for OGL19101. **(a)** $\alpha = [83:98]$ (thumbnail, maximal fitness, corresponding to stanza A_6). **(b)** $\alpha = [66:98]$ (corresponding to stanzas A_5A_6).

maximizing segment is indeed $\alpha^* = [83:98]$. The induced segment family of the fitness maximizing segment is taken as final result of our segmentation problem.

6.3 Enhancement Strategies

Similar as for the reference-based segmentation procedure, we use a combination of various enhancement strategies to deal with local and global pitch deviations as well as with poor recording conditions, see also Section 5.5.

6.3.1 F0-Enhanced Self-Similarity Matrices

Firstly, we again compute F0-enhanced chromagrams by picking only spectral coefficients corresponding to the fundamental frequency (F0) as described in Section 5.5. This results in *F0-enhanced SSMs*, as shown in Figure 6.1b. In comparison to the SSM computed from CENS features (Figure 6.1a), the F0-enhanced SSM exhibits increased robustness against noise and recording artifacts as well as local pitch fluctuations.

6.3.2 Temporal Smoothing

Furthermore, to enhance distorted and fragmented paths of the SSMs, various matrix enhancement strategies have been proposed [6; 134; 144; 164]), where the main idea is to apply some kind of smoothing filter along the direction of the main diagonal having a gradient of $(1, 1)$. This results in an emphasis of diagonal information in $\mathcal{S}$ and a denoising of other structures, This form of filtering, however, typically assumes that the tempo across the music recording is more or less constant and repeating segments have roughly the same length. In the presence of significant tempo differences, however, simply smoothing along the main diagonal may smear out important structural information. To avoid this, we use a strategy that filters the SSM along various gradients as proposed in [134] covering tempo variations of roughly ±30 percent.

Obviously, choosing an appropriate value for the smoothing length parameter constitutes a trade-off between enhancement capability and level of detail. A suitable parameter depends on the kind of audio material.[1] See Figure 6.1c for an illustration and [134] for details.

6.3.3 Thresholding and Normalization

We further process the SSM by suppressing all values that fall below a given threshold. Using normalized chroma features and the cosine measure as similarity measure, all values of the SSM are between 0 and 1. Using a suitable threshold parameter $t > 0$ and a penalty parameter $p \leq 0$, we first set the score values of all cells with a score below t to the value p and then linearly scale the range $[t : 1]$ to $[0 : 1]$, see Figure 6.1d. The thresholding introduces some kind of denoising, whereas the parameter p imposes some additional penalty on all cells of low score. Intuitively, we want to achieve that the relevant path structure lies in the positive part of the resulting SSM, whereas all other cells are given a negative score. Note that different methods can be used for thresholding such as using a predefined threshold or using a relative threshold to enforce a certain percentage of cells to have positive score [162].[2] Again we denote the resulting matrix simply by $\mathcal{S}$.

6.3.4 Transposition and Fluctuation Invariance

As mentioned above, the non-professional singers of the folk songs often deviate significantly from the expected pitches and have serious problems with the intonation. Even worse, their voices often fluctuate by several semitones downwards or upwards across the various stanzas of the same recording. For example, in the case of the OGL19101 recording, the singer's voice constantly increases in pitch while performing the stanzas of this song. As a result, many expected paths of the resulting SSM are weak or even completely missing as illustrated by Figure 6.1d.

One can simulate transpositions (shifts of one or several semitones) on the feature level simply by cyclically shifting a chroma vector along its twelve dimensions [63]. Based on this observation, we adopt the concept of transposition-invariant self-similarity matrices as introduced in [124]. Here, one first computes the similarity between the original feature sequence and each of the twelve cyclically shifted versions of the chromagram resulting in twelve similarity matrices. Then, the *transposition-invariant SSM*, denoted by $\mathcal{S}^{\mathrm{trans}}$, is calculated by taking the point-wise maximum over these twelve matrices. As indicated by Figure 6.1e, many of the missing paths are recovered this way.

The cyclic chroma shifts account for transpositions that correspond to the semitone level of the equal-tempered scale. However, when dealing with the folk song field recordings, one

[1] In our folk song experiments, we use a smoothing length corresponding to 6 seconds. This also takes into account that the length of an individual stanza is above this value.

[2] In our experiments, we choose the threshold in a relative fashion by keeping 40% of the cells having the highest score and set $p = -2$. These values were found experimentally. Slight changes of the parameters' values did not have a significant impact on the final segmentation results.

Strategy	F0	P	R	F
$\mathcal{S}$	$-$	0.668	0.643	0.652
$\mathcal{S}$	$+$	0.734	0.704	0.717
$\mathcal{S}^{\text{trans}}$	$+$	0.821	0.821	0.821
$\mathcal{S}^{\text{fluc}}$	$+$	0.862	0.855	0.860
$\mathcal{S}^{\text{fluc}}$, $\lvert\alpha\rvert \geq 10$	$+$	0.871	0.879	0.872
$\mathcal{S}^{\text{fluc}}$, $\lvert\alpha\rvert \geq 10$ (modified dataset)	$+$	0.954	0.940	0.949
Reference-based method (see Table 5.2)	$+$	0.912	0.940	0.926

Table 6.1: Precision, recall, and F-measure values for the reference-based segmentation method (see Table 5.2) and the reference-free approach using $\delta = 2$.

may have to deal with pitch fluctuations that are fractions of semitones. One strategy may be to introduce an additional tuning estimation step to adjust the frequency bands used in the chroma decomposition [59; 127] and then to compute the SSM from the resulting features. This strategy only works, when one has to deal with a *global* de-tuning that is constant throughout the recording. For the field recordings, however, one often has to deal with *local* pitch fluctuations. Actually, for many recordings such as OGL19101, the singer *continuously* drops or raises with her voice over the various stanzas. This leads to local path distortions and interruptions (see Figure 6.1e). To compensate for such local de-tunings, we further sample the space of semitones using different multirate filter banks corresponding to a shift of 0, 1/4, 1/3, 1/2, 2/3, and 3/4 semitones, respectively, see [127]. Using the resulting six different chromagrams together with the twelve cyclically shifted versions of each of them, we compute 72 similarity matrices as above and then take the point-wise maximum over these matrices to obtain a single *fluctuation-invariant SSM*, denoted by $\mathcal{S}^{\text{fluc}}$. This strategy leads to further improvements as as illustrated by Figure 6.1f, which now shows the expected “full” path structure.

6.4 Experiments

Table 6.1 shows the results obtained for our reference-free segmentation procedure (see Chapter 5) as well as the results of the reference-based method for comparison. For a detailed description of the experimental setup, we refer to Section 5.6. Using the original self-similarity matrix $\mathcal{S}$ derived from the original CENS features to determine the fitness maximizing segment α^*, our reference-free method yields an F-measure value of $\mathrm{F} = 0.652$. Using our F0-enhanced CENS features to increase the robustness against background noise and small local pitch deviations, the F-measure increases to $\mathrm{F} = 0.717$. As mentioned before, dealing with field recordings performed by non-professional singers under poor recording conditions, the matrix enhancement strategies as introduced in Section 6.3 are extremely important for obtaining robust segmentations. In particular, because of the continuous intonation and pitch shifts of the singers, the concepts of transposition and fluctuation invariance significantly improve the segmentation results. For example, using the transposition-invariant SSM $\mathcal{S}^{\text{trans}}$, the F-measure value increases to $\mathrm{F} = 0.821$. Furthermore, when using the fluctuation-invariant SSM $\mathcal{S}^{\text{fluc}}$ that even accounts for shifts corresponding to fractions of a semitone, the F-measure value further increases to $\mathrm{F} = 0.860$.

Assuming some prior knowledge on the minimal length of a stanza, the results can be further improved. For example, to avoid over-segmentation [116], one may consider only segments α satisfying $|\alpha| \geq 10$ seconds, which results in $F = 0.872$, see Table 6.1. This result is still worse than the results obtained from the reference-based approach ($F = 0.926$). Actually, a manual inspection showed that this degradation was mainly caused by four particular recordings, where the segmentation derived from α^* was "phase-shifted" compared to the ground truth. Employing a boundary-based evaluation measure resulted in an F-measure of $F = 0$ for these four recordings. Furthermore, we found out that these phase shifts were caused by the fact that in all of these four recordings the singer completely failed in the first stanza (omitting and confusing entire verse lines). In these cases, the stanza transcript used in the reference-based approach corresponds to the remaining "correct" stanzas. As a result, the reference-based approach can better deal with this issue and is able to recover at least the boundaries of the remaining stanzas.

In a final experiment we simulate a similar behavior by replacing the four recordings using a slightly shortened version, where we omit the first stanzas, respectively. Repeating the previous experiment on this modified dataset produced an F-measure of $F = 0.949$, which is already exceeding the quality obtained by the baseline method. However, there are still some boundaries that are incorrectly detected by our approach. A further investigation revealed that most errors correspond to boundaries that are slightly misplaced and do not fall into the ± 2 seconds tolerance. In many of these cases, there is a short amount of silence between two stanzas, which also introduces some uncertainty to the manually annotated ground-truth boundaries.

6.5 Conclusion

In this chapter, we presented an reference-free approach for automatically segmenting folk song field recordings in a robust way even in the presence of significant temporal and spectral distortions across repeating stanzas. One crucial step in the overall segmentation pipeline was to employ various enhancement strategies that allow for dealing with such distortions already on the feature and SSM levels. Our experiments showed that one obtains good segmentation results having a similar quality as the ones obtained from the reference-based method. Future work in this direction deals with the issue on how the segmentation can be made more robust to structural differences in the stanzas.

The described segmentation task is only a first step towards making the audio material more accessible to performance analysis and folk song research. In the next chapter, we introduce tools that allow a folk song researcher to conveniently screen a large number of field recordings in order to detect and locate interesting and surprising features worth being examined in more detail by domain experts.

Chapter 7

Towards Automated Analysis of Performance Variations

In this chapter, we present various techniques for analyzing the variations within the recorded folk song material. As discussed in the previous chapters, the singers often deviate significantly from the expected pitches. Furthermore, there are also significant temporal and melodic variations between the stanzas belonging to the same folk song recording. It is important to realize that such variabilities and inconsistencies may be, to a significant extent, properties of the repertoire and not necessarily errors of the singers. As the folk songs are part of the oral culture and have been passed down over centuries without any fixed notation, variations introduced by the individual singers are very characteristic for this kind of audio material (see Section 5.1 for a more detailed explanation of folk song characteristics). To measure such deviations and variations within the acoustic audio material, we use a multimodal approach by exploiting the existence of a symbolically given transcription of an idealized stanza.

As one main contribution of this chapter, we propose a novel method for capturing temporal and melodic characteristics of the various stanzas of a recorded song in a compact matrix representation, which we refer to as *chroma template* (CT). The computation of such a chroma template involves several steps. First, we convert the symbolic transcription as well as each stanza of a recorded song into chroma representations. On the basis of these representations, we determine and compensate for the tuning differences between the recorded stanzas using the transcription as reference. To account for temporal variations between the stanzas, we use time warping techniques. Finally, we derive a chroma template by averaging the transposed and warped chroma representations of all recorded stanzas and the reference. The key property of a chroma template is that it reveals consistent and inconsistent melodic performance aspects across the various stanzas. Here, one advantage of our concept is its simplicity, where the information is given in form of an explicit and semantically interpretable matrix representation. We show how our framework can be used to automatically measure variabilities in various musical dimensions including tempo, pitch, and melody. In particular, it allows for directly comparing the realization of different stanzas of a folk song performance. Extracting such information constitutes an important step for making the performance aspects enclosed in the audio material accessible to performance analysis and to folk song research.

The remainder of this chapter is structured as follows. First, in Section 7.1, we introduce and discuss in detail our concept of chroma templates and present various strategies that capture and compensate for variations in intonation and tuning. In Section 7.2, we describe various experiments on performance analysis while discussing our concept by means of a number of representative examples. In Section 7.3, we introduce a user interface that makes the actual folk song recordings more accessible to researchers. As on main idea, the interface allows for intuitively navigating within a folk song recording and comparing the constituent stanzas. Further notes and prospects on future work are given in Section 7.4. Related work is discussed in the respective sections.

7.1 Chroma Templates

In the following, we assume that, for a given folk song, we have an audio recording consisting of various stanzas as well as a transcription of a representative stanza in form of a MIDI file, which will act as a reference. Recall from Section 5.1 that this is exactly the situation we have with the songs of the OGL collection. Furthermore, we assume that a segmentation of the audio recording in its constituent stanzas is available. This segmentation can be derived automatically using the approaches presented in Chapter 5 and Chapter 6. In order to compare the MIDI reference with the individual stanzas of the audio recording, we use chroma features as introduced in Section 5.2. Figure 7.1 shows chroma representations for the song NLB72246. Figure 7.1b shows the chromagram of the MIDI reference corresponding to the score shown in Figure 7.1a. Figure 7.1c shows the chromagram of a single stanza of the audio recording. In the following, we refer to the chromagram of an audio recording as *audio chromagram*. In our implementation, all chromagrams are computed at a feature resolution of 10 Hz (10 features per second). For details, we refer to Section 5.2.

As mentioned above, most singers have significant problems with the intonation. To account for poor recording conditions, intonation problems, and pitch fluctuations we apply the enhancement strategies as described in Section 5.5. First, we enhance the audio chromagram by exploiting the fact that we are dealing with monophonic music. To this end, we estimate the fundamental frequency (F0) for each audio frame and assign energy only to the MIDI pitch with the center frequency that is closest to the estimated fundamental frequency. This results in chromagrams having exactly one non-zero entry in each time frame. The resulting binary chromagram is referred to *F0-enhanced audio chromagram*. By using an F0-based pitch quantization, most of the noise resulting from poor recording conditions is suppressed. Also local pitch deviations caused by the singers' intonation problems as well as vibrato are compensated to a substantial degree. This effect is also visable in Figure 7.1d showing the F0-enhanced version of the audio chromagram as shown in Figure 7.1c.

To account for global differences in key between the MIDI reference and the recorded stanzas, we revert to the observation by Goto [62] that the twelve cyclic shifts of a 12-dimensional chroma vector naturally correspond to the twelve possible transpositions. Therefore, it suffices to determine the cyclic shift index $\iota \in [0 : 11]$ (where shifts are con-

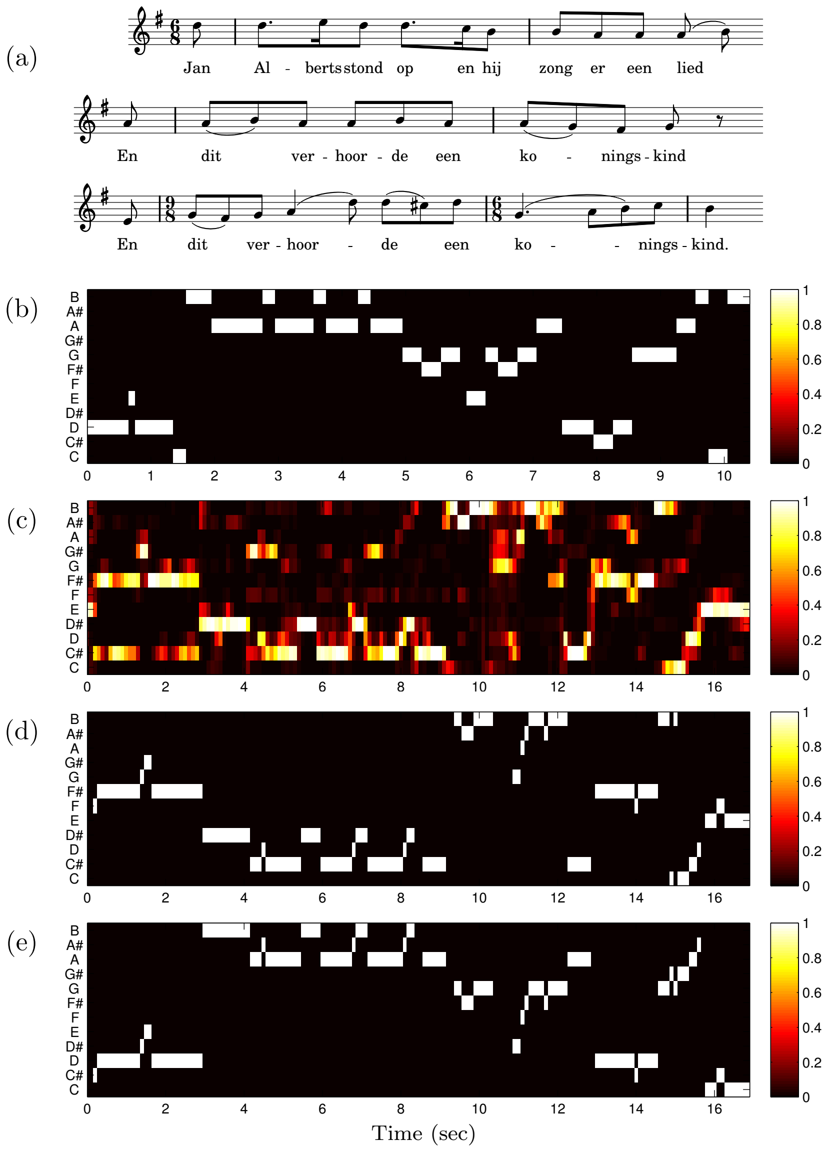


Figure 7.1: Multimodal representation of a stanza of the folk song NLB72246. **(a)** Idealized transcription given in form of a score. **(b)** Reference chromagram of transcription. **(c)** Audio chromagram of a field recording of a single stanza. **(d)** F0-enhanced audio chromagram. **(e)** Transposed F0-enhanced audio chromagram cyclically shifted by eight semitones upwards ($\iota = 8$).

sidered upwards in the direction of increasing pitch) that minimizes the distance between a stanza's audio and reference chromagram and then to cyclically shift the audio chromagram according to this index. Figure 7.1e shows the cyclically shifted by eight semitones ($\iota = 8$) audio chromagram to match the key of the reference. Note the similarities between the two chroma representations after correcting the transposition. The distance measure between the reference chromagram and the audio chromagram is based on dynamic time warping as described in Section 5.3.

So far, we have accounted for transpositions that correspond to integer semitones of the

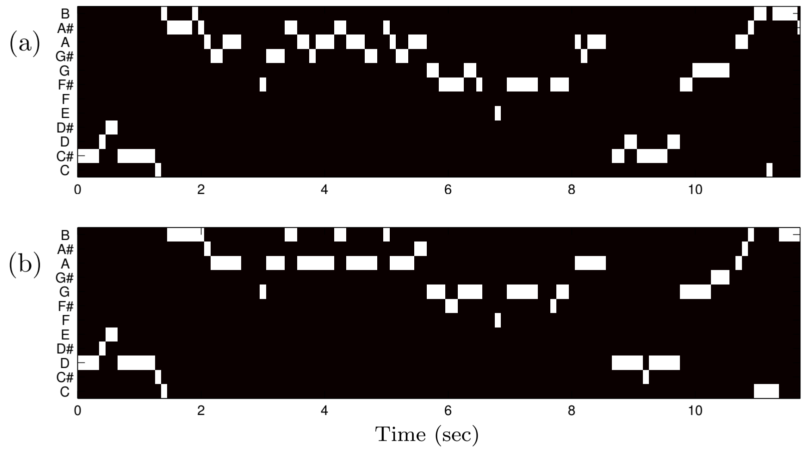


Figure 7.2: Tuned audio chromagrams of a recorded stanza of the folk song NLB72246. **(a)** Audio chromagram with respect to tuning parameter $\tau = 6$. **(b)** Audio chromagram with respect to tuning parameter $\tau = 6.5$.

equal-tempered pitch scale. However, the above mentioned voice fluctuations are fluent in frequency and do not stick to a strict pitch grid. To cope with pitch deviations that are fractions of a semitone, we consider different shifts $\sigma \in [0, 1]$ in the assignment of MIDI pitches and center frequencies as given by Eq. (5.1). More precisely, for a MIDI pitch p, the σ-shifted center frequency $f^\sigma(p)$ is given by

$$f_p^\sigma = 2^{\frac{p-69-\sigma}{12}} \cdot 440 \text{ Hz} . \tag{7.1}$$

Now, in the F0-based pitch quantization as described above, one can use σ-shifted center frequencies for different values σ to account for tuning nuances. In our context, we use four different values $\sigma \in \left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}\right\}$ in combination with the 12 cyclic chroma shifts to obtain 48 different audio chromagrams. Actually, a similar strategy is suggested in [59; 162] where generalized chroma representations with 24 or 36 bins (instead of the usual 12 bins) are derived from a short-time Fourier transform. We then determine the cyclic shift index ι and the shift σ that minimize the distance between the reference chromagram and the resulting audio chromagram. These two minimizing numbers can be expressed by a single rational number

$$\tau := \iota + \sigma \in [0, 12), \tag{7.2}$$

which we refer to as *tuning parameter*. The audio chromagram obtained by applying a tuning parameter is also referred to as *tuned audio chromagram*. Figure 7.2 illustrates the importance of introducing the additional rational shift parameter σ. Here, slight fluctuations around a frequency that lies between the center frequencies of two neighboring pitches leads to oscillations between the two corresponding chroma bands in the resulting audio chromagram, see Figure 7.2a. By applying an additional half-semitone shift ($\sigma = 0.5$) in the pitch quantization step, these oscillations are removed, see Figure 7.2b.

We now show how one can account for temporal and melodic differences by introducing the concept of chroma templates, which reveal consistent and inconsistent performance aspects across the various stanzas. Our concept of chroma templates is similar to the concept of

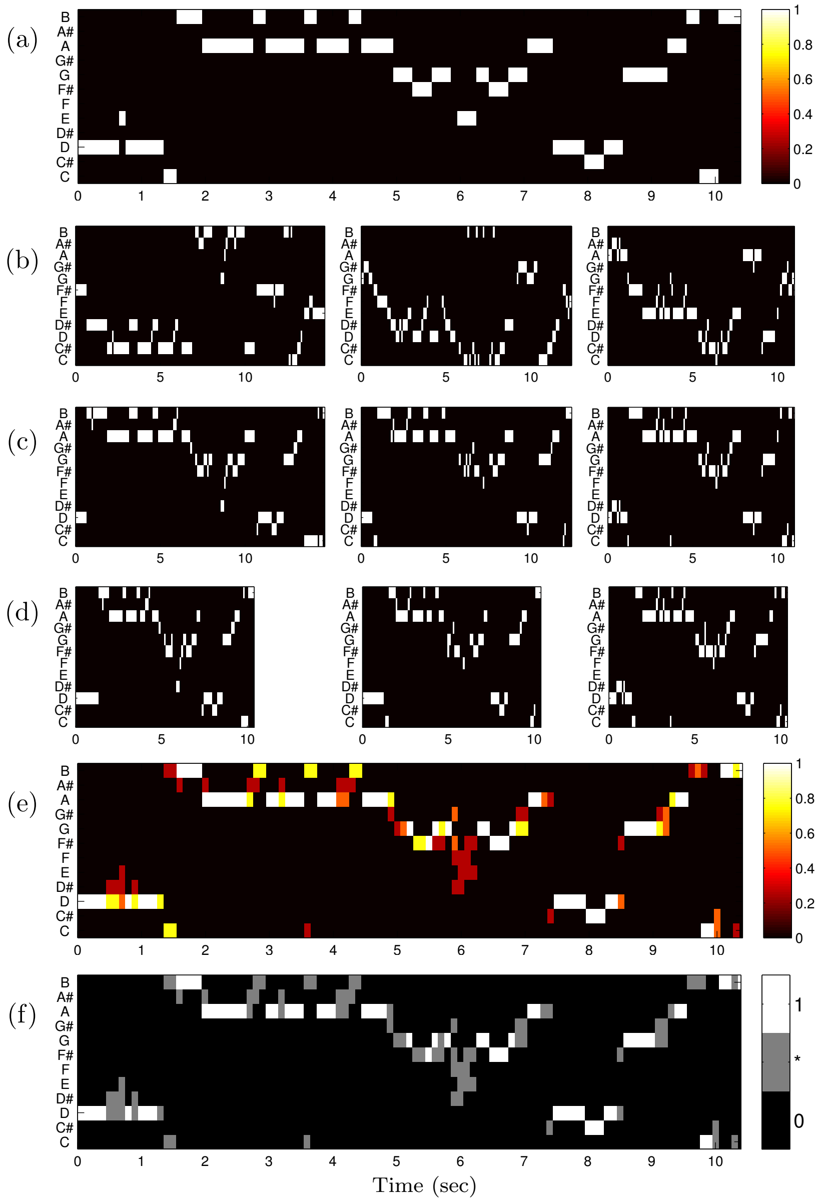


Figure 7.3: Chroma template computation for the folk song NLB72246. **(a)** Reference chromagram. **(b)** Three audio chromagrams. **(c)** Tuned audio chromagrams. **(d)** Warped audio chromagrams. **(e)** Average chromagram obtained by averaging the three audio chromagrams of (d) and the reference of (a). **(f)** Chroma template.

motion templates proposed in [136], which were applied in the context of content-based retrieval of motion capture data. For a fixed folk song, let $Y \in \{0,1\}^{d\times L}$ denote the boolean reference chromagram of dimension $d = 12$ and of length (number of columns) $L \in \mathbb{N}$. Furthermore, we assume that for a given field recording of the song we know the segmentation boundaries of its constituent stanzas. In the following, let N be the number of stanzas and let $X_n \in \{0,1\}^{d\times K_n}$, $n \in [1 : N]$, be the $F0$-enhanced and suitably tuned boolean audio chromagrams, where $K_n \in \mathbb{N}$ denotes the length of X_n. To account for temporal differences, we temporally warp the audio chromagrams to correspond to the reference chromagram Y. Let $X = X_n$ be one of the audio chromagrams of length $K = K_n$. To align X and Y, we employ classical dynamic time warping (DTW) using the Euclidean distance as local cost measure $c : \mathbb{R}^{12} \times \mathbb{R}^{12} \to \mathbb{R}$ to compare two chroma vectors. (Note that when dealing with binary chroma vectors that have at most one nonzero entry, the Euclidean distance equals the Hamming distance.) Recall that a *warping path* is a sequence $p = (p_1, \ldots, p_M)$ with $p_m = (k_m, \ell_m) \in [1 : K] \times [1 : L]$ for $m \in [1 : M]$ satisfying the boundary condition

$$p_1 = (1, 1) \text{ and } p_M = (K, L)$$

as well as the step size condition

$$p_{m+1} - p_m \in \{(1, 0), (0, 1), (1, 1)\}$$

for $m \in [1 : M - 1]$. The total cost of p is defined as $\sum_{m=1}^{M} c(X(k_m), Y(\ell_m))$. Now, let p^* denote a warping path having minimal total cost among all possible warping paths. Then, the DTW distance $\mathrm{DTW}(X, Y)$ between X and Y is defined to be the total cost of p^*. It is well-known that p^* and $\mathrm{DTW}(X, Y)$ can be computed in $O(KL)$ using dynamic programming, see [123; 149] for details. Next, we locally stretch and contract the audio chromagram X 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

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

for some $n \in \mathbb{N}$, i.e., the column $X(k)$ is aligned to the n columns $Y(\ell), \ldots, Y(\ell + n - 1)$ of the reference. In this case, we duplicate the column $X(k)$ by taking n copies of it. In the second case, p^* contains a subsequence of the form

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

for some $n \in \mathbb{N}$, i.e., the n columns $X(k), \ldots, X(k + n - 1)$ are aligned to the single column $Y(\ell)$. In this case, we replace the n columns by a single column by taking the component-wise AND-conjunction $X(k) \wedge \ldots \wedge X(k + n - 1)$. For example, one obtains

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \wedge \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \wedge \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

The resulting warped chromagram is denoted by $\bar{X}$. Note that $\bar{X}$ is still a boolean chromagram and the length of $\bar{X}$ equals the length L of the reference Y, see Figure 7.3d for an example.

After the temporal warping we obtain an optimally tuned and warped audio chromagram for each stanza. Now, we simply average the reference chromagram Y with the warped audio chromagrams $\bar{X}_1, \ldots, \bar{X}_N$ to yield an average chromagram

$$Z := \frac{1}{N+1}\Big(Y + \sum_{n \in [1:N]} \bar{X}_n\Big). \tag{7.3}$$

Note that the average chromagram Z has real-valued entries between zero and one and has the same length L as the reference chromagram. Figure 7.3e shows such an average chromagram obtained from three audio chromagrams and the reference chromagram.

The important observation is that black/white regions of Z indicate periods in time (horizontal axis) where certain chroma bands (vertical axis) consistently assume the same values zero/one in all chromagrams, respectively. By contrast, colored regions indicate inconsistencies mainly resulting from variations in the audio chromagrams (and partly from inappropriate alignments). In other words, the black and white regions encode characteristic aspects that are shared by all chromagrams, whereas the colored regions represent the variations coming from different performances. To make inconsistent aspects more explicit, we further quantize the matrix Z by replacing each entry of Z that is below a threshold δ by zero, each entry that is above $1 - \delta$ by one, and all remaining entries by a *wildcard character* $*$ indicating that the corresponding value is left unspecified, see Figure 7.3f. The resulting quantized matrix is referred to as *chroma template* for the audio chromagrams $X_1, \ldots, X_N$ with respect to the reference chromagram Y. In the following section, we discuss the properties of such chroma templates in detail by means of several representative examples.

7.2 Folk Song Performance Analysis

The analysis of different interpretations, also referred to as *performance analysis*, has become an active research field [37; 111; 152; 184; 185]. Here, one objective is to extract expressive performance aspects such as tempo, dynamics, and articulation from audio recordings. To this end, one needs accurate annotations of the audio material by means of suitable musical parameters including onset times, note duration, sound intensity, or fundamental frequency. To ensure such a high accuracy, annotation is often done manually, which is infeasible in view of analyzing large audio collections. For the folk song scenario, we now sketch how various performance aspects can be derived in a fully automated fashion. In particular, we discuss how one can capture performance aspects and variations regarding tuning, tempo, as well as melody across the various stanzas of a field recording.

For the sake of concreteness, we explain these concepts by means of our running example NLB72246 shown in Figure 7.1a. As discussed in Section 7.1, we first compensate for difference in key and tuning by estimating a tuning parameter τ for each individual stanza of the field recording. This parameter indicates to which extend the stanza's audio chromagram needs to be shifted upwards to optimally agree with the reference chromagram. Figure 7.4b shows the tuning parameter τ for each of the 25 stanzas of the field recording. As can be seen, the tuning parameter almost constantly decreases from stanza to stanza, thus indicating a constant rise of the singer's voice. The singer starts the performance by singing the first stanza roughly $\tau = 7.75$ semitones lower than indicated by the reference transcription. Continuously going up with the voice, the singer finishes the song with the last stanza only $\tau = 4.5$ semitones below the transcription, thus differing by more than three semitones from the beginning. Note that in our processing pipeline, we compute tuning parameters on the stanza level. In other words, significant shifts in tuning within a stanza cannot yet be captured by our methods. This may be one unwanted reason

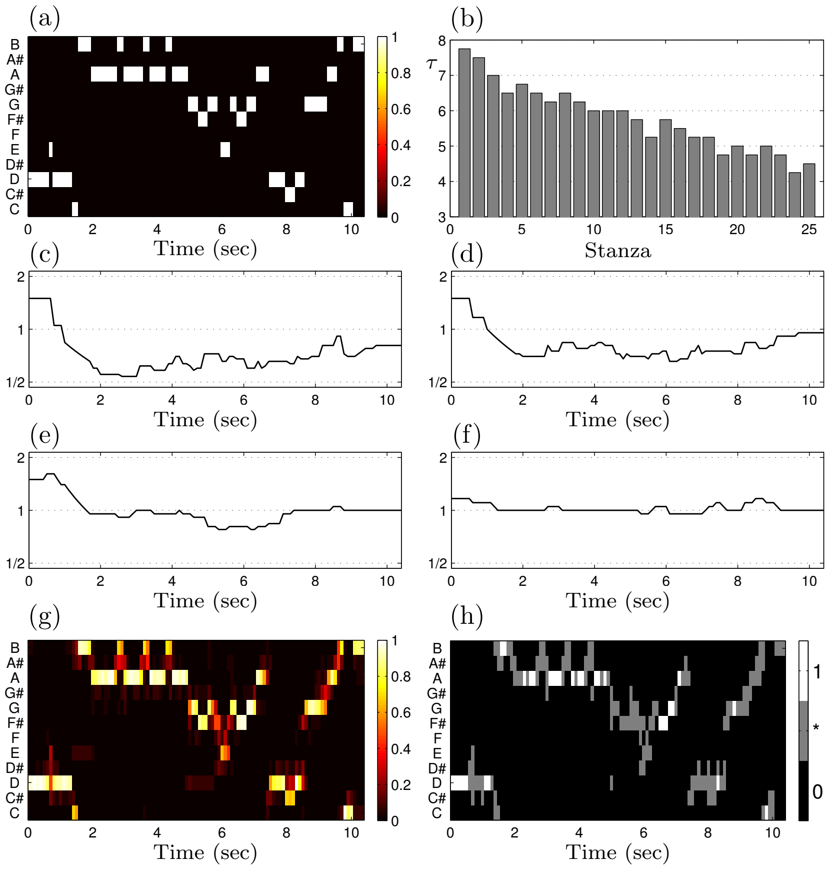


Figure 7.4: Various performance aspects for a field recording of NLB72246 comprising 25 stanzas. **(a)** Reference chromagram. **(b)** Tuning parameter τ for each stanza. **(c) - (f)** Tempo curves for the stanzas 1, 7, 19, and 25. **(g)** Average chromagram. **(h)** Chroma template.

when obtaining many inconsistencies in our chroma templates. For the future, we think of methods on how to handle such detuning artifacts within stanzas.

After compensating for tuning differences, we apply DTW-based warping techniques in order to compensate for temporal differences between the recorded stanzas, see Section 7.1. Actually, an optimal warping path p^* encodes the relative tempo difference between the two sequences to be aligned. In our case, one sequence corresponds to one of the performed stanzas of the field recording and the other sequence corresponds to the idealized transcription, which was converted into a MIDI representation using a constant tempo of 120 BPM. Now, by aligning the performed stanza with the reference stanza (on the level of chromagram representations), one can derive the relative tempo deviations between these two versions [133]. These tempo deviations can be described through a tempo curve that, for each position of the reference, indicates the relative tempo difference between the performance and the reference. In Figure 7.4c-f, the tempo curves for four recorded stanzas of NLB72246 are shown. The horizontal axis encodes the time axis of the MIDI reference (rendered at 120 BPM), whereas the vertical encodes the relative tempo difference in form of a factor. For example, a value of 1 indicates that the performance has the same tempo as the reference (in our case 120 BPM). Furthermore, the value 1/2 indicates half the

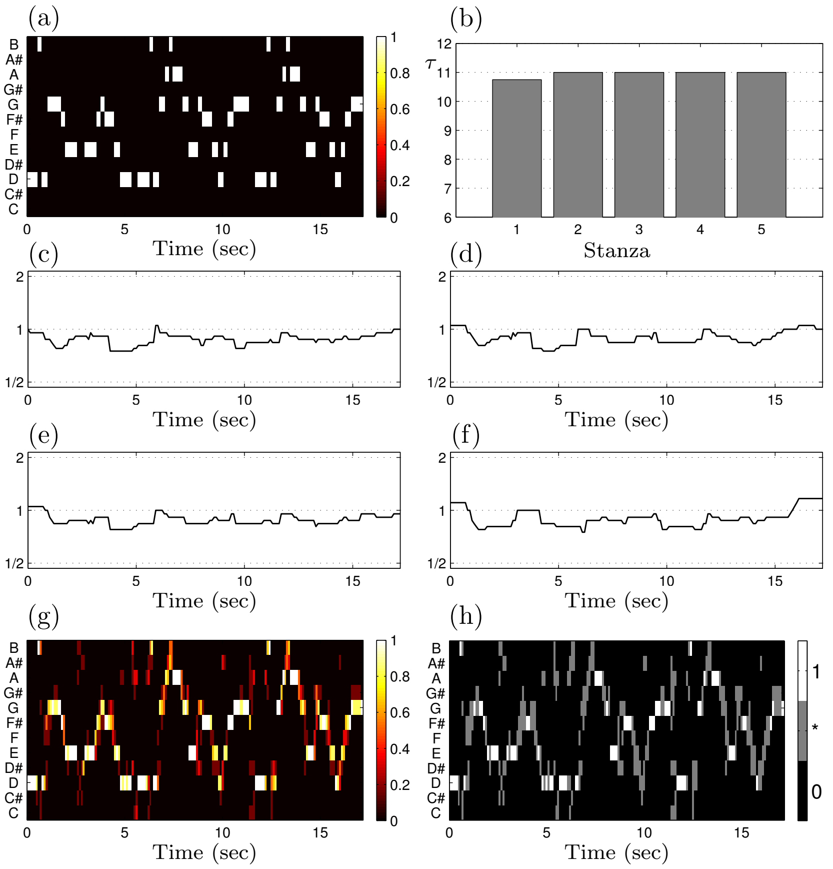


Figure 7.5: Various performance aspects for a field recording of NLB73626 comprising 5 stanzas. **(a)** Reference chromagram. **(b)** Tuning parameter τ for each stanza. **(c) - (f)** Tempo curves for the first 4 stanzas. **(g)** Average chromagram. **(h)** Chroma template.

tempo (in our case 60 BPM) and the value 2 indicates twice the tempo relative to the reference (in our case 240 BPM). As can be seen from Figure 7.4c, the singer performs the first stanza at an average tempo of roughly 85 BPM (factor 0.7). However, the tempo is not constant throughout the stanza. Actually, the singer starts with a fast tempo, then slows down significantly, and accelerates again towards the end of the stanza. Similar tendencies can be observed in the performances of the other stanzas. As an interesting observation, the average tempo of the stanzas continuously increases throughout the performance. Starting with an average tempo of roughly 85 BPM in the first stanza, the tempo averages to 99 BPM in stanza 7, 120 BPM in stanza 19, and reaches 124 BPM in stanza 25. Also, in contrast to stanzas at the beginning of the performance, the tempo is nearly constant for the stanzas towards the end of the recording. This may be an indicator that the singer becomes more confident in her singing capabilities as well as in her capabilities of remembering the song.

Finally, after tuning and temporally warping the audio chromagrams, we compute an average chromagram and a chroma template. In the quantization step, we use a threshold δ. In our experiments, we set $\delta = 0.1$, thus disregarding inconsistencies that occur in less than 10% of the stanzas. This introduces some robustness towards outliers. The average

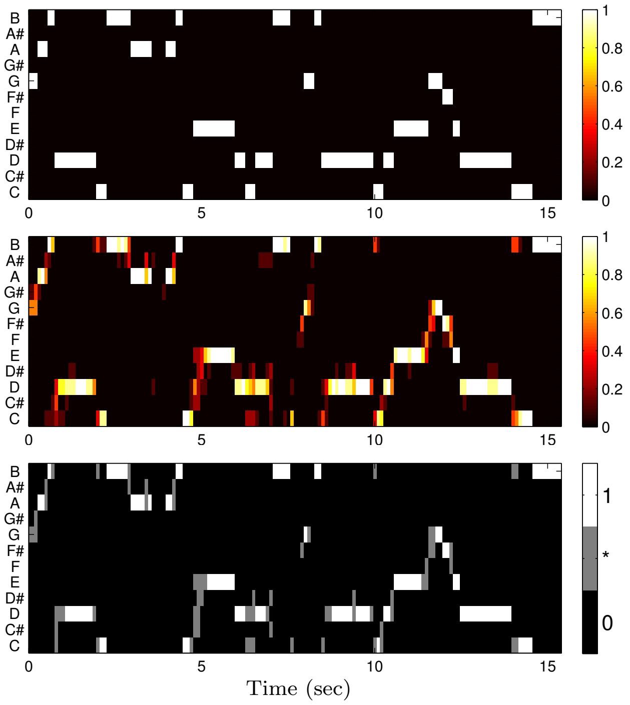


Figure 7.6: Reference chromagram (top), average chromagram (middle) and chroma template (bottom) for the folk song recording NLB74437 comprising 8 stanzas.

chromagram and a chroma template for NLB72246 are shown of Figure 7.4g and Figure 7.4h, respectively. Here, in contrast to Figure 7.3, all 25 stanzas of the field recording were considered in the averaging process. As explained above, the wildcard character $*$ (gray color) of a chroma template indicates inconsistent performance aspects across the various stanzas of the field recording. Since we already compensated for tuning and tempo differences before averaging, the inconsistencies indicated by the chroma templates tend to reflect local melodic inconsistencies and inaccuracies. We illustrate this by our running example, where the inconsistencies particularly occur in the third phrase of the stanza (starting with the fifth second of the MIDI reference). One possible explanation for these inconsistencies may be as follows. In the first two phrases of the stanza, the melody is relatively simple in the sense that neighboring notes differ only either by a unison interval or by a second interval. Also the repeating note A4 plays the role of a stabilizing anchor within the melody. In contrast, the third phrase of the stanza is more involved. Here, the melody contains several larger intervals as well as a meter change. Therefore, because of the higher complexity, the singer may have problems in accurately and consistently performing the third phrase of the stanza.

As a second example, we consider the folk song NLB73626, see Figure 7.5. The corresponding field recording comprises 5 stanzas, which are sung in a relatively clean and consistent way. Firstly, the singer keeps the pitch more or less on the same level throughout the performance. This is also indicated by Figure 7.5b, where one has a tuning parameter of $\tau = 4$ for all, except for the first stanza where one has $\tau = 3.75$. Secondly, as shown

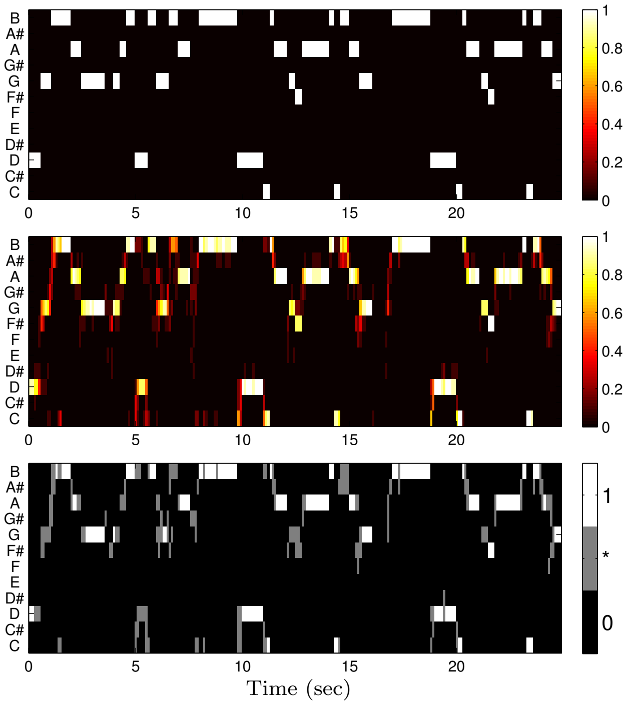


Figure 7.7: Reference chromagram (top), average chromagram (middle) and chroma template (bottom) for the folk song recording NLB73287 comprising 11 stanzas.

by Figure 7.5c-f, the average tempo is consistent over all stanzas. Also, the shapes of all the tempo curves are highly correlated. This temporal consistency may be an indicator that the local tempo deviations are a sign of artistic intention rather than a random and unwanted imprecision. Thirdly, the chroma template shown in Figure 7.5h exhibits many white regions, thus indicating that many notes of the melody have been performed in a consistent way. The gray areas, in turn, which correspond to the inconsistencies, appear mostly in transition periods between consecutive notes. Furthermore, they tend to have an ascending or descending course while smoothly combining the pitches of consecutive notes. Here, one reason is that the singer tends to slide between two consecutive pitches, which has the effect of some kind of portamento. All of these performance aspects indicate that the singer seems to be quite familiar with the song and confident in her singing capabilities.

We close our discussion on performance analysis by having a look at the chroma templates of another three representative examples. Figure 7.6 shows the chroma template of the folk song NLB74437. The template shows that the performance is very consistent, with almost all notes remaining unmasked. Actually, this is rather surprising since NLB74437 is one of the few recordings, where several singers perform together. Even though, in comparison to other recordings, the performers do not seem to be particularly good singers and even differ in tuning and melody, singing together seems to mutually stabilize the singers thus resulting in a rather consistent overall performance. Also the chroma template shown in Figure 7.7 is relatively consistent. Similarly to the example shown in Figure 7.5, there

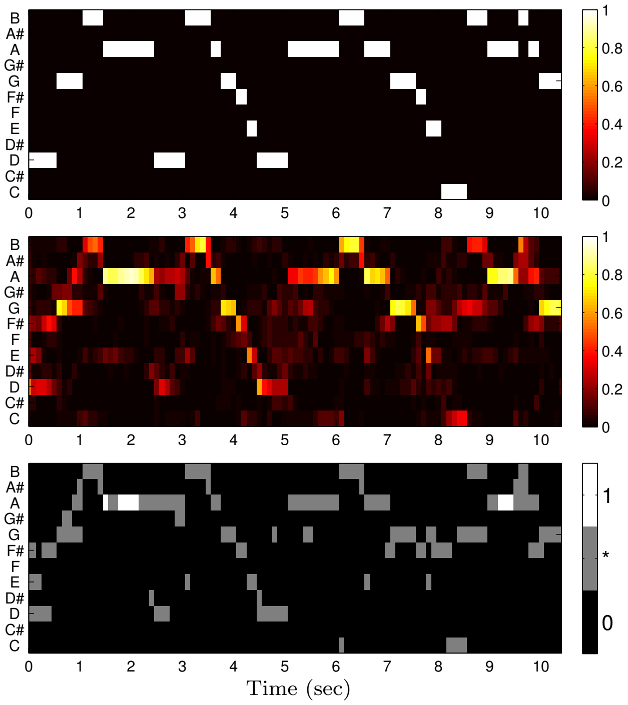


Figure 7.8: Reference chromagram (top), average chromagram (middle) and chroma template (bottom) for the folk song recording NLB72395 comprising 12 stanzas.

are inconsistencies that are caused by portamento effects. As a last example, we consider the chroma template of the folk song NLB72395, where nearly all notes have been marked as inconsistent, see Figure 7.8. This is a kind of negative result, which indicates the limitations of our concept. A manual inspection showed that some of the stanzas of the field recording exhibit significant structural differences, which are neither reflected by the transcription nor in accordance with most of the other stanzas. For example, in at least two recorded stanzas one entire phrase is omitted by the singer. In such cases, using a global approach for aligning the stanzas inevitably leads to poor and semantically meaningless alignments that cause many inconsistencies. The handling of such structural differences constitutes an interesting research problem.

7.3 A User Interface for Folk Song Navigation

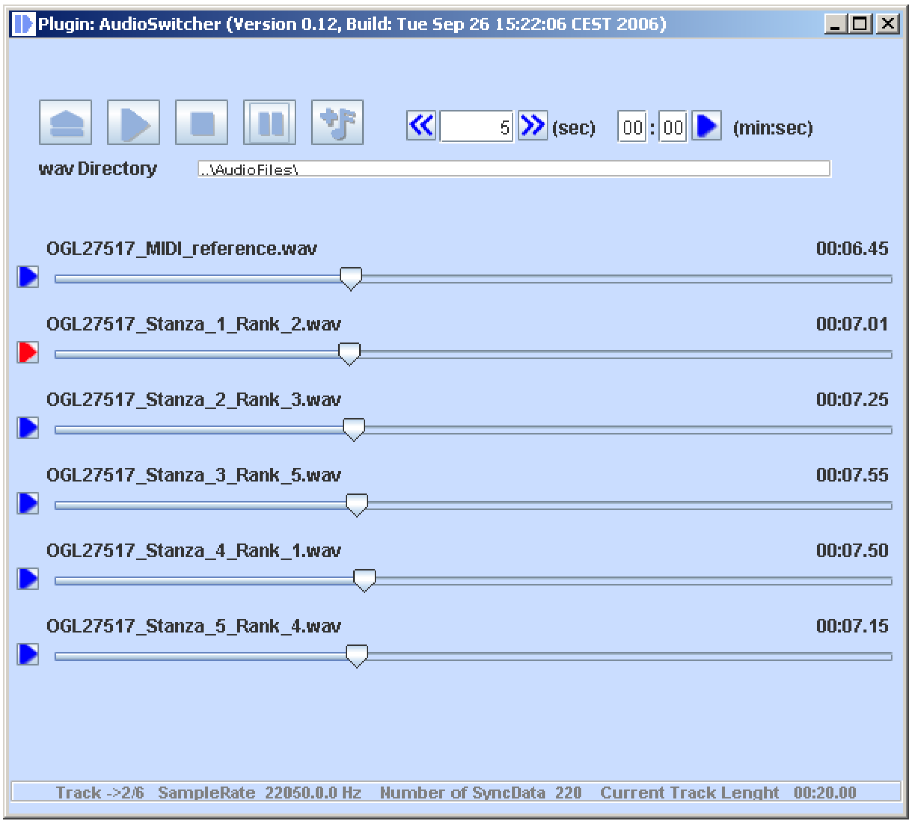
Chroma templates capture performance aspects and variations in the various stanzas of a folk song. In particular, the chroma templates give a *visual* impression of the variations occurring. We now present a user interface that allows for analyzing such variations by means of listening to and, in particular, comparing the different stanzas of an audio recording in a convenient way.

Once having segmented the audio recording into stanzas and computed alignment paths between the MIDI reference and all audio stanzas, one can then derive the temporal correspondences between the MIDI and the audio representation with the objective to associate note events given by the MIDI file with their physical occurrences in the audio recording, see [123] for details. The result can be regarded as an automated annotation of the entire audio recording with available MIDI events. Such annotations facilitate multimodal browsing and retrieval of MIDI and audio data, thus opening new ways of experiencing and researching music. For example, most successful algorithms for melody-based retrieval work in the domain of symbolic or MIDI music. On the other hand, retrieval results may be most naturally presented by playing back the original recording of the melody, while a musical score or a piano-roll representation may be the most appropriate form for visually displaying the query results. For a description of such functionalities, we refer to [26].

Furthermore, aligning each stanza of the audio recording to the MIDI reference yields a multi-alignment between all stanzas. Exploiting this alignment, one can implement interfaces that allow a user to seamlessly switch between the various stanzas of the recording thus facilitating a direct access and comparison of the audio material [123]. The *Audio Switcher* [57] constitutes such a user interface, which allows the user to open in parallel a synthesized version of the MIDI reference as well as all stanzas of the folk song recording, see Figure 7.9. Each of the stanzas is represented by a slider bar indicating the current playback position with respect to the stanza's particular time scale. The stanza that is currently used for audio playback, in the following referred to as active stanza, is indicated by a red marker located to the left of the slider bar. The slider knob of the active stanza moves at constant speed while the slider knobs of the other stanzas move accordingly to the relative tempo variations with respect to the active stanza. The active stanza may be changed at any time simply by clicking on the respective playback symbol located to the left of each slider bar. The playback of the new active stanza then starts at the time position that musically corresponds to the last playback position of the former active stanza. This has the effect of seamlessly crossfading from one stanza to another while preserving the current playback position in a musical sense. One can also jump to any position within any of the stanzas by directly selecting a position of the respective slider. Such functionalities assists the user in detecting and analyzing the differences between several recorded stanzas of a single folk song. The Audio Switcher is realized as plug-in of the SyncPlayer system [106; 57], which is an an advanced software audio player with a plug-in interface for MIR applications and provides tools for navigating within audio recordings and browsing in music collections. For further details and functionalities, we refer to the literature.

7.4 Conclusion

In this chapter, we presented a multimodal approach for extracting performance parameters from folk song recordings by comparing the audio material with symbolically given reference transcriptions. As the main contribution, we introduced the concept of chroma templates that reveal the consistent and inconsistent melodic aspects across the various stanzas of a given recording. In computing these templates, we used tuning and time warping strategies to deal with local variations in melody, tuning, and tempo.



Plugin: AudioSwitcher (Version 0.12, Build: Tue Sep 26 15:22:06 CEST 2006)

5 (sec) 00 : 00 (min:sec)

wav Directory ..\AudioFiles\

OGL27517_MIDI_reference.wav 00:06.45

OGL27517_Stanza_1_Rank_2.wav 00:07.01

OGL27517_Stanza_2_Rank_3.wav 00:07.25

OGL27517_Stanza_3_Rank_5.wav 00:07.55

OGL27517_Stanza_4_Rank_1.wav 00:07.50

OGL27517_Stanza_5_Rank_4.wav 00:07.15

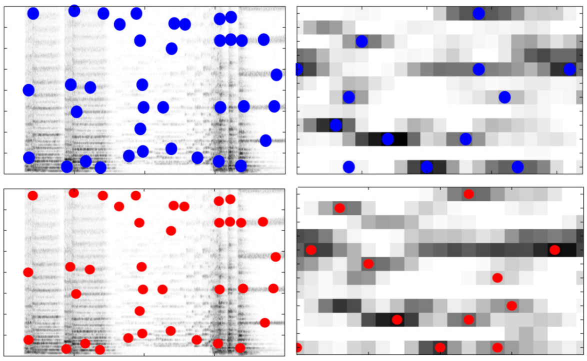
Track ->2/6 SampleRate 22050.0.0 Hz Number of SyncData 220 Current Track Lenght 00:20.00

Figure 7.9: Instance of the Audio Switcher plug-in of the SyncPlayer showing the synthesized version of the MIDI reference and the five different stanzas of the audio recording of OGL27517.

The variabilities across the various stanzas of a given recording revealed and observed in this chapter may have various causes, which need to be further explored in future research. Often these causes are related to questions in the area of music cognition. A first hypothesis is that stable notes are structurally more important than variable notes. The stable notes may be the ones that form part of the singer's mental model of the song, whereas the variable ones are added to the model at performance time. Variations may also be caused by problems in remembering the song. It has been observed that often melodies stabilize after the singer performed a few iterations. Such effects may offer insight in the working of the musical memory. Furthermore, melodic variabilities caused by ornamentations can also be interpreted as a creative aspect of performance. Such variations may be motivated by musical reasons, but also by the lyrics of a song. Sometimes song lines have an irregular length, necessitating the insertion or deletion of notes. Variations may also be introduced by the singer to emphasize key words in the text or, more general, to express the meaning of the song. Finally one may study details on tempo, timing, pitch, and loudness in relation to performance, as a way of characterizing performance styles of individuals or regions.

Part III

Audio Retrieval



Chapter 8

A Review of Content-Based Music Retrieval

The way music is stored, accessed, distributed, and consumed underwent a radical change in the last decades. Nowadays, large collections containing millions of digital music documents are accessible from anywhere around the world. Such a tremendous amount of readily available music requires retrieval strategies that allow users to explore large music collections in a convenient and enjoyable way. Most audio search engines rely on metadata and textual annotations of the actual audio content [19]. Editorial metadata typically include descriptions of the artist, title, or other release information. The drawback of a retrieval solely based on editorial metadata is that the user needs to have a relatively clear idea of what he or she is looking for. Typical query terms may be a title such as "Act naturally" when searching the song by The Beatles or a composer's name such as "Beethoven" (see Figure 8.1a).[1] In other words, traditional editorial metadata only allow to search for already known content. To overcome these limitations, editorial metadata has been more and more complemented by general and expressive annotations (so called *tags*) of the actual musical content [10; 97; 173]. Typically, tags give descriptions of the musical style or genre of a recording, but may also include information about the mood, the musical key, or the tempo [110; 172]. In particular, tags form the basis for music recommendation and navigation systems that make the audio content accessible even when users are not looking for a specific song or artist but for music that exhibits certain musical properties [173]. The generation of such annotations of audio content, however, is typically a labor intensive and time-consuming process [19; 172]. Furthermore, often musical expert knowledge is required for creating reliable, consistent, and musically meaningful annotations. To avoid this tedious process, recent attempts aim at substituting expert-generated tags by user-generated tags [172]. However, such tags tend to be less accurate, subjective, and rather noisy. In other words, they exhibit a high degree of variability between users. Crowd (or social) tagging, one popular strategy in this context, employs voting and filtering strategies based on large social networks of users for "cleaning" the tags [110]. Relying on the "wisdom of the crowd" rather than the "power of the few" [99], tags assigned by many users are considered more reliable than tags assigned

[1] www.google.com (accessed Dec. 18, 2011)

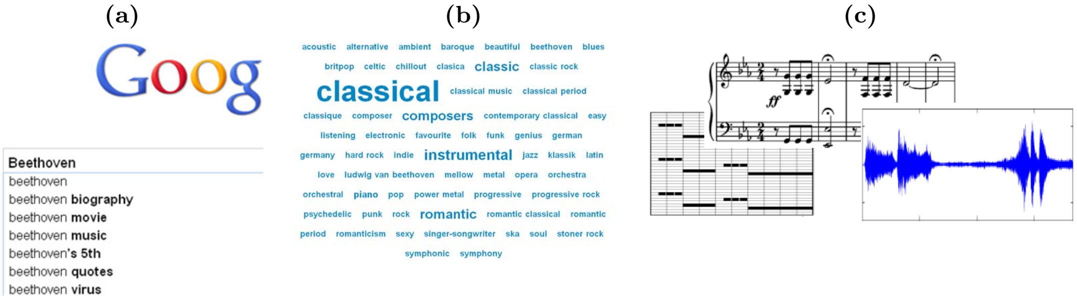

(a)
(b)
(c)
Goog
Beethoven
beethoven
beethoven biography
beethoven movie
beethoven music
beethoven's 5th
beethoven quotes
beethoven virus
acoustic alternative ambient baroque beautiful beethoven blues
britpop celtic chillout clasica classic classic rock
classical classical music classical period
classique composer composers contemporary classical easy
listening electronic favourite folk funk genius german
germany hard rock indie instrumental jazz klassik latin
love ludwig van beethoven mellow metal opera orchestra
orchestral piano pop power metal progressive progressive rock
psychedelic punk rock romantic romantic classical romantic
period romanticism sexy singer-songwriter ska soul stoner rock
symphonic symphony
ff

Figure 8.1: Illustration of retrieval concepts. **(a)** Traditional retrieval using textual metadata (e. g., artist, title) and a web search engine. **(b)** Retrieval based on rich and expressive metadata given by tags. **(c)** Content-based retrieval using audio, MIDI, or score information.

by only a few users. Figure 8.1b shows the Last.fm[2] *tag cloud* for “Beethoven”. Here, the font size reflects the frequency of the individual tags. One major drawback of this approach is that it relies on a large crowd of users for creating reliable annotations [110]. While mainstream pop/rock music is typically covered by such annotations, less popular genres are often scarcely tagged. This phenomenon is also known as the “long-tail” problem [20; 172]. To overcome these problems, *content-based retrieval* strategies have great potential as they do not rely on any manually created metadata but are exclusively based on the audio content and cover the entire audio material in an objective and reproducible way [19]. One possible approach is to employ automated procedures for tagging music, such as automatic genre recognition, mood recognition, or tempo estimation [9; 173]. The major drawback of these learning-based strategies is the requirement of large corpora of tagged music examples as training material and the limitation to queries in textual form. Furthermore, the quality of the tags generated by state-of-the-art procedures does not reach the quality of human generated tags [173].

In this chapter, we present and discuss various retrieval strategies based on audio content that follow the *query-by-example* paradigm: given an audio recording or a fragment of it (used as query or example), the task is to automatically retrieve documents from a given music collection containing parts or aspects that are similar to it. As a result, retrieval systems following this paradigm do not require any textual descriptions. However, the notion of similarity used to compare different audio recordings (or fragments) is of crucial importance and largely depends on the respective application as well as the user requirements. Such strategies can be loosely classified according to their specificity, which refers to the degree of similarity between the query and the database documents.

The remainder of this chapter is organized as follows. In Section 8.1, we first give an overview on the various audio retrieval tasks following the query-by-example paradigm. In particular, we extend the concept of specificity by introducing a second aspect: the granularity of a retrieval task referring to the temporal scope. Then, we discuss representative state-of-the-art approaches to audio identification (Section 8.2), audio matching (Section 8.3), and version identification (Section 8.4). In Section 8.5, we discuss open problems in the field of content-based retrieval and give an outlook on future directions.

[2] www.last.fm (accessed Dec. 18, 2011)

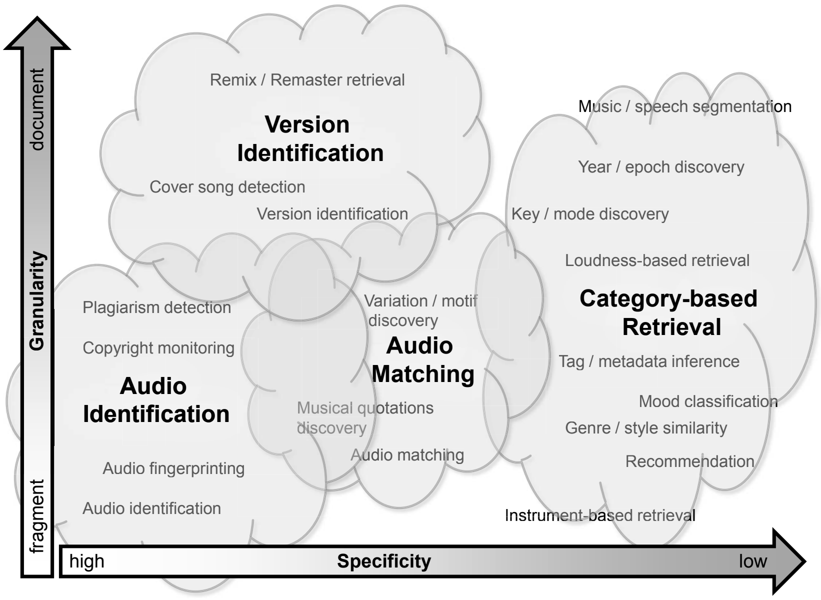


Figure 8.2: Specificity/granularity pane showing the various facets of content-based music retrieval.

8.1 Audio-Based Query-By-Example

Many different audio content-based retrieval systems have been proposed, following different strategies and aiming at different application scenarios. Generally, such retrieval systems can be characterized by various aspects such as the notion of similarity, the underlying matching principles, or the query format. Following and extending the concept introduced in [19], we consider the following two aspects: *specificity* and *granularity*, see Figure 8.2. The *specificity* of a retrieval system refers to the degree of similarity between the query and the database documents to be retrieved. High-specific retrieval systems return exact copies of the query (in other words, they *identify* the query or occurrences of the query within database documents), whereas low-specific retrieval systems return vague matches that are similar with respect to some musical properties. As in [19], different content-based music retrieval scenarios can be arranged along a specificity axis as shown in Figure 8.2 (horizontally). We extend this classification scheme by introducing a second aspect, the *granularity* (or temporal scope) of a retrieval scenario. In *fragment-level* retrieval scenarios, the query consists of a short fragment of an audio recording, and the goal is to retrieve all musically related fragments that are contained in the documents of a given music collection. Typically, such fragments may cover only a few seconds of audio content or may correspond to a motif, a theme, or a musical part of a recording. In contrast, in *document-level* retrieval, the query reflects characteristics of an entire document and is compared with entire documents of the database. Here, the notion of similarity typically is rather coarse and the features capture global statistics of an entire recording. In this context, one has to distinguish between some kind of internal and some kind of external granularity of the retrieval tasks. In our classification scheme, we use the term fragment-level when a fragment-based similarity measure is used to compare fragments of audio recordings (internal), even though entire documents are returned as matches (external). Using such a classification allows for extending the specificity axis to a specificity/granularity pane as shown in Figure 8.2. In particular, we have identified four different groups of retrieval scenarios corresponding to the four clouds in Figure 8.2. Each of the clouds, in turn, encloses a number of different retrieval scenarios. Obviously, the clouds are not strictly separated but blend into each other. Even though this taxonomy is rather vague and sometimes questionable, it gives an intuitive overview of the various retrieval paradigms while illustrating their subtle but crucial differences.

An example of a high-specific fragment-level retrieval task is *audio identification* (sometimes also referred to as *audio fingerprinting* [16]). Given a small audio fragment as query, the task of audio identification consists in identifying the particular audio recording that is the source of the fragment [1]. Nowadays, audio identification is widely used in commercial systems such as Shazam.[3] Typically, the query fragment is exposed to signal distortions on the transmission channel [16; 107]. Recent identification algorithms exhibit a high degree of robustness against noise, MP3 compression artifacts, uniform temporal distortions, or interferences of multiple signals [56; 82]. The high specificity of this retrieval task goes along with a notion of similarity that is very close to the identity. To make this point clearer, we distinguish between a piece of music (in an abstract sense) and a specific performance of this piece. In particular for Western classical music, there typically exist a large number of different recordings of the same piece of music performed by different musicians. Given a query fragment, e. g., taken from a Bernstein recording of Beethoven's Symphony No. 5, audio fingerprinting systems are not capable of retrieving, e. g., a Karajan recording of the same piece. Likewise, given a query fragment from a live performance of "Act naturally" by The Beatles, the original studio recording of this song may not be found. The reason for this is that existing fingerprinting algorithms are not designed to deal with strong non-linear temporal distortions or with other musically motivated variations that affect, for example, the tempo or the instrumentation.

At a lower specificity level, the goal of fragment-based *audio matching* is to retrieve all audio fragments that musically correspond to a query fragment from all audio documents contained in a given database [105; 135]. In this scenario, one explicitly allows semantically motivated variations as they typically occur in different performances and arrangements of a piece of music. These variations include significant non-linear global and local differences in tempo, articulation, and phrasing as well as differences in executing note groups such as grace notes, trills, or arpeggios. Furthermore, one has to deal with considerable dynamical and spectral variations, which result from differences in instrumentation and loudness.

One instance of document-level retrieval at a similar specificity level as audio matching is the task of *version identification*. Here, the goal is to identify different versions of the same piece of music within a database [161]. In this scenario, one not only deals with changes in instrumentation, tempo, and tonality, but also with more extreme variations concerning the musical structure, key, or melody, as typically occurring in remixes and cover songs. This requires document-level similarity measures to globally compare entire documents.

[3] www.shazam.com (accessed Dec. 18, 2011)

Finally, there are a number of even less specific document-level retrieval tasks which can be grouped under the term *category-based retrieval.* This term encompasses retrieval of documents whose relationship can be described by cultural or musicological categories. Typical categories are genre [174], rhythm styles [65; 157], or mood and emotions [98; 171; 182] and can be used in fragment as well as document-level retrieval tasks. Music recommendation or general music similarity assessments [14; 183] can be seen as further document-level retrieval tasks of low specificity.

In the following, we elaborate the aspects of specificity and granularity by means of representative state-of-the-art content-based retrieval approaches. In particular, we highlight characteristics and differences in requirements when designing and implementing systems for audio identification, audio matching, and version identification. Furthermore, we address efficiency and scalability issues. We start with discussing high-specific audio fingerprinting (Section 8.2), continue with mid-specific audio matching (Section 8.3), and then discuss version identification (Section 8.4).

8.2 Audio Identification

Of all content-based music retrieval tasks, audio identification has received most interest and is now widely used in commercial applications. In the identification process, the audio material is compared by means of so-called *audio fingerprints*, which are compact content-based signatures of audio recordings [16]. In real-world applications, these fingerprints need to fulfill certain requirements. First of all, the fingerprints should capture highly specific characteristics so that a short audio fragment suffices to reliably identify the corresponding recording and distinguish it from millions of other songs. However, in real-world scenarios, audio signals are exposed to distortions on the transmission channel. In particular, the signal is likely to be affected by noise, artifacts from lossy audio compression, pitch shifting, time scaling, equalization, or dynamics compression. For a reliable identification, fingerprints have to show a significant degree of robustness against such distortions. Furthermore, scalability is an important issue for all content-based retrieval applications. A reliable audio identification system needs to capture the entire digital music catalog, which is further growing every day. In addition, to minimize storage requirements and transmission delays, fingerprints should be compact and efficiently computable [16]. Most importantly, this also requires efficient retrieval strategies to facilitate very fast database look-ups. These requirements are crucial for the design of large-scale audio identification systems. To satisfy all these requirements, however, one typically has to face a trade-off between contradicting principles.

There are various ways to design and compute fingerprints. One group of fingerprints consist of short sequences of frame-based feature vectors such as Mel-Frequency Cepstral Coefficients (MFCC) [17], Bark-scale spectrograms [82; 83], or a set of low-level descriptors [1]. For such representations, vector quantization [1] or thresholding [82] techniques, or temporal statistics [150] are needed for obtaining the required robustness. Another group of fingerprints consist of a sparse set of characteristic points such as spectral peaks [52; 181] or characteristic wavelet coefficients [96]. As an example, we now describe the peak-based fingerprints suggested by Wang [181], which are now commercially used in the

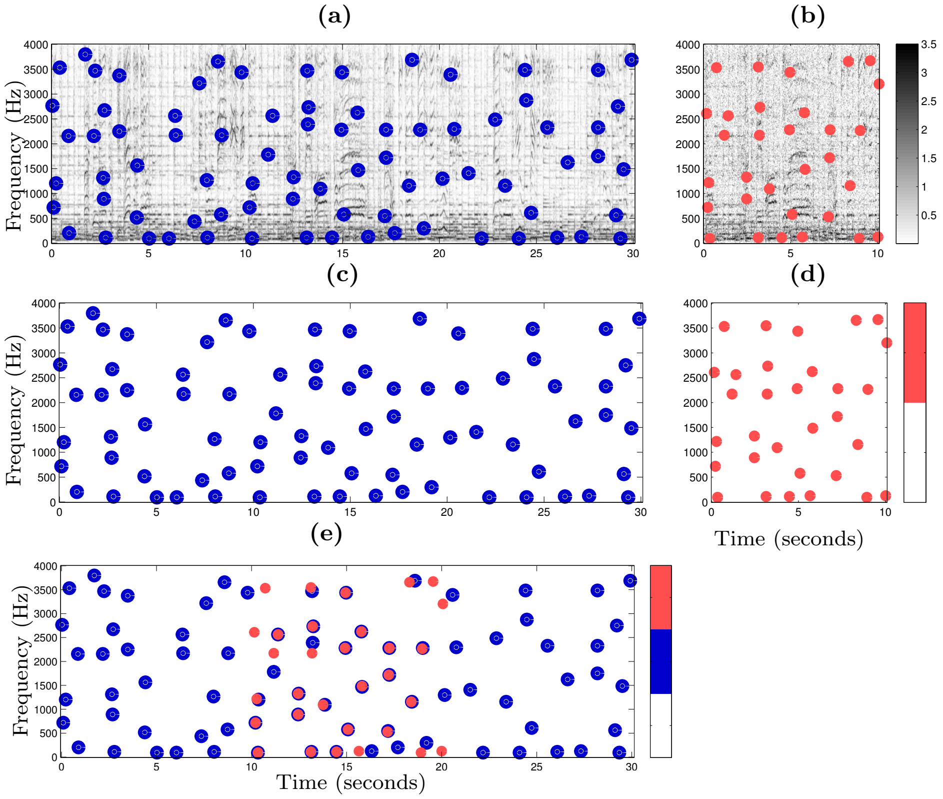


Figure 8.3: Illustration of the Shazam audio identification system using a recording of "Act naturally" by The Beatles as example. **(a)** Database document with extracted peak fingerprints. **(b)** Query fragment (10 seconds) with extracted peak fingerprints. **(c)** Constellation map of database document. **(d)** Constellation map of query document. **(e)** Superposition of the database fingerprints and time-shifted query fingerprints.

Shazam music identification service[4].

The Shazam system provides a smartphone application that allows users to record a short audio fragment of an unknown song using the built-in microphone. The application then derives the audio fingerprints which are sent to a server that performs the database lookup. The retrieval result is returned to the application and presented to the user together with additional information about the identified song. In this approach, one first computes a spectrogram from an audio recording using a short-time Fourier transform. Then, one applies a peak-picking strategy that extracts local maxima in the magnitude spectrogram: time-frequency points that are locally predominant. Figure 8.3 illustrates the basic retrieval concept of the Shazam system using a recording of "Act naturally" by The Beatles. Figure 8.3a and Figure 8.3b show the spectrogram for an example database document (30 seconds of the recording) and a query fragment (10 seconds), respectively. The extracted peaks are superimposed to the spectrograms. The peak-picking step reduces the complex spectrogram to a "constellation map", a low-dimensional sparse representation of the original signal by means of a small set of time-frequency points, see Figure 8.3c and Figure 8.3d. According to [181], the peaks are highly characteristic, reproducible,

[4] www.shazam.com (accessed Dec. 18, 2011)

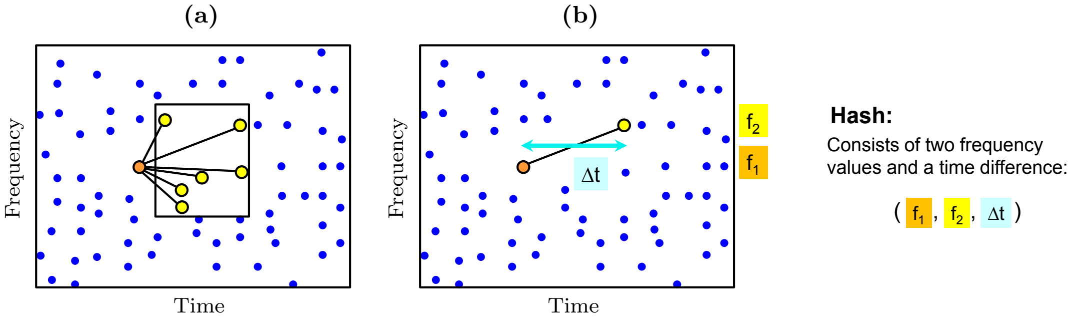


Figure 8.4: Illustration of the peak pairing strategy of the Shazam algorithm. **(a)** Anchor peak and assigned target zone. **(b)** Pairing of anchor peak and target peaks to form hash values.

and robust against many, even significant distortions of the signal. Note that a peak is only defined by its time and frequency values, whereas magnitude values are no longer considered.

The general database look-up strategy works as follows. Given the constellation maps for a query fragment and all database documents, one locally compares the query fragment to all database fragments of the same size. More precisely, one counts matching peaks, i. e., peaks that occur in both constellation maps. A high count indicates that the corresponding database fragment is likely to be a correct hit. This procedure is illustrated in Figure 8.3e, showing the superposition of the database fingerprints and time-shifted query fingerprints. Both constellation maps show a high consistency (many red and blue points coincide) at a fragment of the database document starting at time position 10 seconds, which indicates a hit. However, note that not all query and database peaks coincide. This is because the query was exposed to signal distortions on the transmission channel (in this example additive white noise). Even under severe distortions of the query, there still is a high number of coinciding peaks thus showing the robustness of these fingerprints.

Obviously, such an exhaustive search strategy is not feasible for a large database as the run-time linearly depends on the number and sizes of the documents. For the constellation maps, as proposed in [107], one tries to efficiently reduce the retrieval time using indexing techniques—very fast operations with a sub-linear run-time. However, directly using the peaks as hash values is not possible as the temporal component is not translation-invariant and the frequency component alone does not have the required specificity. In [181], a strategy is proposed, where one considers pairs of peaks. Here, one first fixes a peak to serve as “anchor peak” and then assigns a “target zone” as indicated in Figure 8.4a. Then, pairs are formed of the anchor and each peak in the target zone, and a hash value is obtained for each pair of peaks as a combination of both frequency values and the time difference between the peaks as indicated in Figure 8.4b. Using every peak as anchor peak, the number of items to be indexed increases by a factor that depends on the number of peaks in the target zone. This combinatorial hashing strategy has three advantages. Firstly, the resulting fingerprints show a higher specificity than single peaks, leading to an acceleration of the retrieval as fewer exact hits are found. Secondly, the fingerprints are translation-invariant as no absolute timing information is captured. Thirdly, the combinatorial multiplication of the number of fingerprints introduced by considering pairs of peaks as well as the local nature of the peak pairs increases the robustness to signal degradations.

The Shazam audio identification system facilitates a high identification rate, while scaling to large databases. One weakness of this algorithm is that it can not handle time scale modifications of the audio as frequently occurring in the context of broadcasting monitoring. The reason for this is that time scale modifications (also leading to frequency shifts) of the query fragment completely change the hash values. Extensions of the original algorithms dealing with this issue are presented in [52; 176].

8.3 Audio Matching

The problem of audio identification can be regarded as largely solved even for large scale music collections. Less specific retrieval tasks, however, are still mostly unsolved. In this section, we highlight the difference between high-specific audio identification and mid-specific audio matching while presenting strategies to cope with musically motivated variations. In particular, we introduce chroma-based audio features [4; 59; 123] and sketch distance measures that can deal with local tempo distortions. Finally, we indicate how the matching procedure may be extended using indexing methods to scale to large datasets [18; 105].

For the audio matching task, suitable descriptors are required to capture characteristics of the underlying piece of music, while being invariant to properties of a particular recording. Chroma-based audio features [4; 123], sometimes also referred to as pitch class profiles [59], are a well-established tool for analyzing Western tonal music and have turned out to be a suitable mid-level representation in the retrieval context [18; 105; 135; 123]. Assuming the equal-tempered scale, the chroma attributes 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. Capturing energy distributions in the twelve pitch classes, chroma-based audio features closely correlate to the harmonic progression of the underlying piece of music. This is the reason why basically every matching procedure relies on some type of chroma feature, see Section 5.2.

There are many ways for computing chroma features. For example, the decomposition of an audio signal into a chroma representation (or chromagram) may be performed either by using short-time Fourier transforms in combination with binning strategies [59] or by employing suitable multirate filter banks [123; 127]. Figure 8.5 illustrates the computation of chroma features for a recording of the first five measures of Beethoven's Symphony No. 5 in a Bernstein interpretation. The main idea is that the fine-grained (and highly specific) signal representation as given by a spectrogram (Figure 8.5c) is coarsened in a musically meaningful way. Here, one adapts the frequency axis to represent the semitones of the equal tempered scale (Figure 8.5d). The resulting representation captures musically relevant pitch information of the underlying music piece, while being significantly more robust against spectral distortions than the original spectrogram. To obtain chroma features, pitches differing by octaves are summed up to yield a single value for each pitch class, see Figure 8.5e. The resulting chroma features show increased robustness against

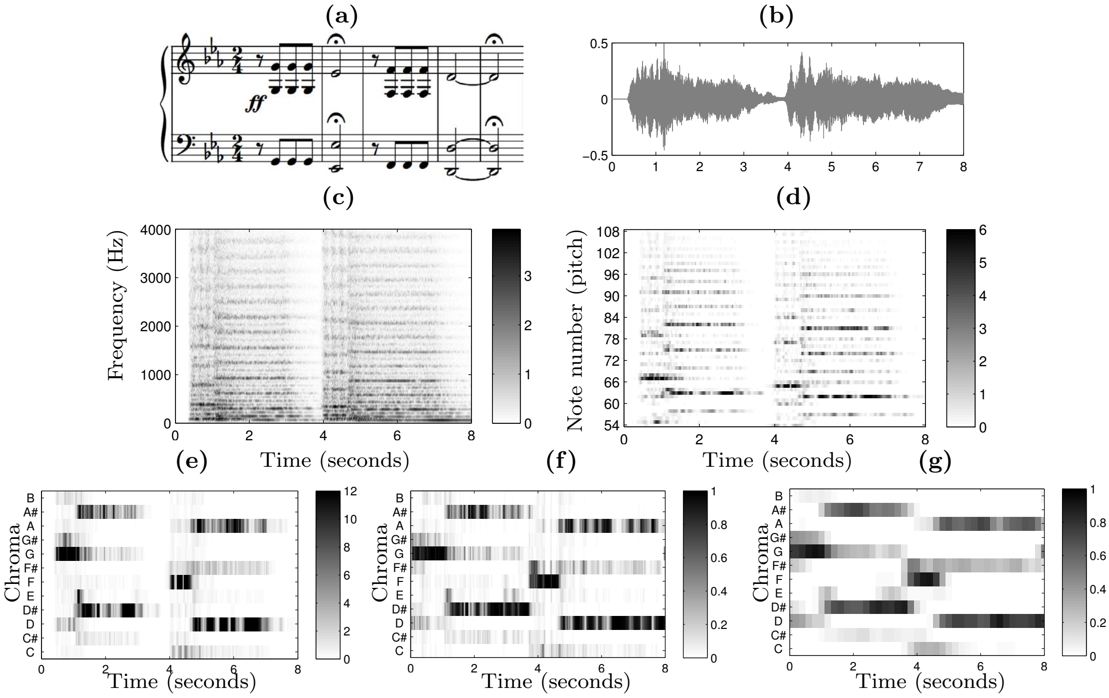


Figure 8.5: Illustration of various feature representations for the beginning of Beethoven's Opus 67 (Symphony No. 5) in a Bernstein interpretation. **(a)** Score of the excerpt. **(b)** Waveform. **(c)** Spectrogram with linear frequency axis. **(d)** Spectrogram with frequency axis corresponding to musical pitches. **(e)** Chroma features. **(f)** Normalized chroma features. **(g)** Smoothed version of chroma features.

changes in timbre, as typically resulting from different instrumentations.

The degree of robustness of the chroma features against musically motivated variations can be further increased by using suitable post-processing steps. See [127] for some chroma variants.[5] For example, normalizing the chroma vectors (Figure 8.5f) makes the features invariant to changes in loudness or dynamics. Furthermore, applying a temporal smoothing and downsampling step (see Figure 8.5g) may significantly increase robustness against local temporal variations that typically occur as a result of local tempo changes or differences in phrasing and articulation, see also [127] There are many more variants of chroma features comprising various processing steps. For example, applying logarithmic compression or whitening procedures enhances small yet perceptually relevant spectral components and the robustness to timbre [120; 126]. A peak picking of spectrum's local maxima can enhance harmonics while suppressing noise-like components [59; 46]. Furthermore, generalized chroma representations with 24 or 36 bins (instead of the usual 12 bins) allow for dealing with differences in tuning [59]. Such variations in the feature extraction pipeline have a large influence and the resulting chroma features can behave quite differently in the subsequent analysis task.

Figure 8.6 shows spectrograms and chroma features for two different interpretations (by

[5] MATLAB implementations for some chroma variants are supplied by the Chroma Toolbox: www.mpi-inf.mpg.de/resources/MIR/chromatoolbox (accessed Dec. 18, 2011)

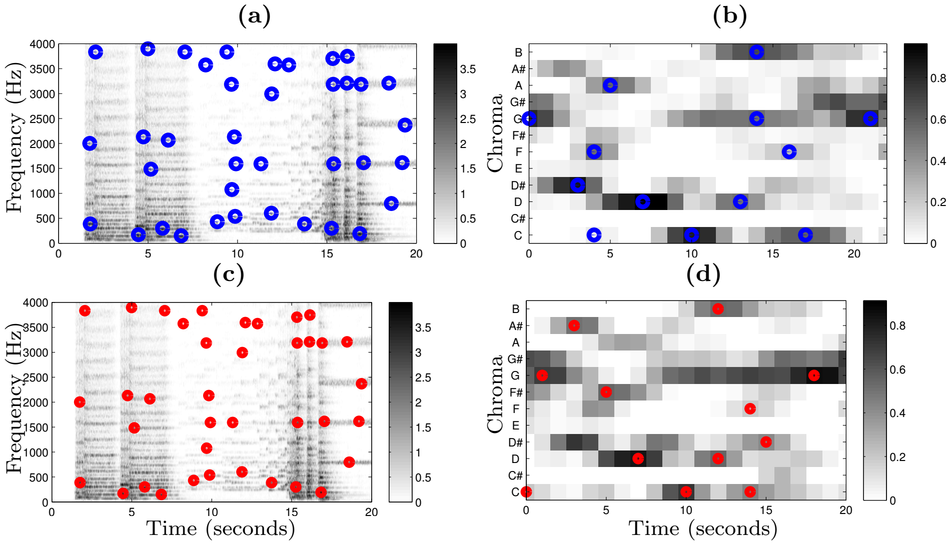


Figure 8.6: Different representations and peak fingerprints extracted for recordings of the first 21 measures of Beethoven's Symphony No. 5. **(a)** Spectrogram-based peaks for a Bernstein recording. **(b)** Chromagram-based peaks for a Bernstein recording. **(c)** Spectrogram-based peaks for a Karajan recording. **(d)** Chromagram-based peaks for a Karajan recording.

Bernstein and Karajan) of Beethoven's Symphony No. 5. Obviously, the chroma features exhibit a much higher similarity than the spectrograms, revealing the increased robustness against musical variations. The fine-grained spectrograms, however, reveal characteristics of the individual interpretations. To further illustrate this, Figure 8.6 also shows fingerprint peaks for all representations. As expected, the spectrogram peaks are very inconsistent for the different interpretations. The chromagram peaks, however, show at least some consistencies, indicating that fingerprinting techniques could also be applicable for audio matching [12]. This strategy is further analyzed in Chapter 9.

Instead of using sparse peak representations, one typically employs a subsequence search, which is directly performed on the chroma features. Here, a query chromagram is compared with all subsequences of database chromagrams. As a result one obtains a matching curve as shown in Figure 8.7, where a small value indicates that the subsequence of the database starting at this position is similar to the query sequence. Then the best match is the minimum of the matching curve. In this context, one typically applies distance measures that can deal with tempo differences between the versions, such as edit distances [5], dynamic time warping (DTW) [123; 135], or the Smith-Waterman algorithm [162]. An alternative approach is to linearly scale the query to simulate different tempi and then to minimize over the distances obtained for all scaled variants [105]. Figure 8.7 shows three different matching curves which are obtained using strict subsequence matching, DTW, and a multiple query strategy.

To speed up such exhaustive matching procedures, one requires methods that allow for efficiently detecting *near* neighbors rather than exact matches. A first approach in this direction uses inverted file indexing [105] and depends on a suitable codebook consisting of a finite set of characteristic chroma vectors. Such a codebook can be obtained in an

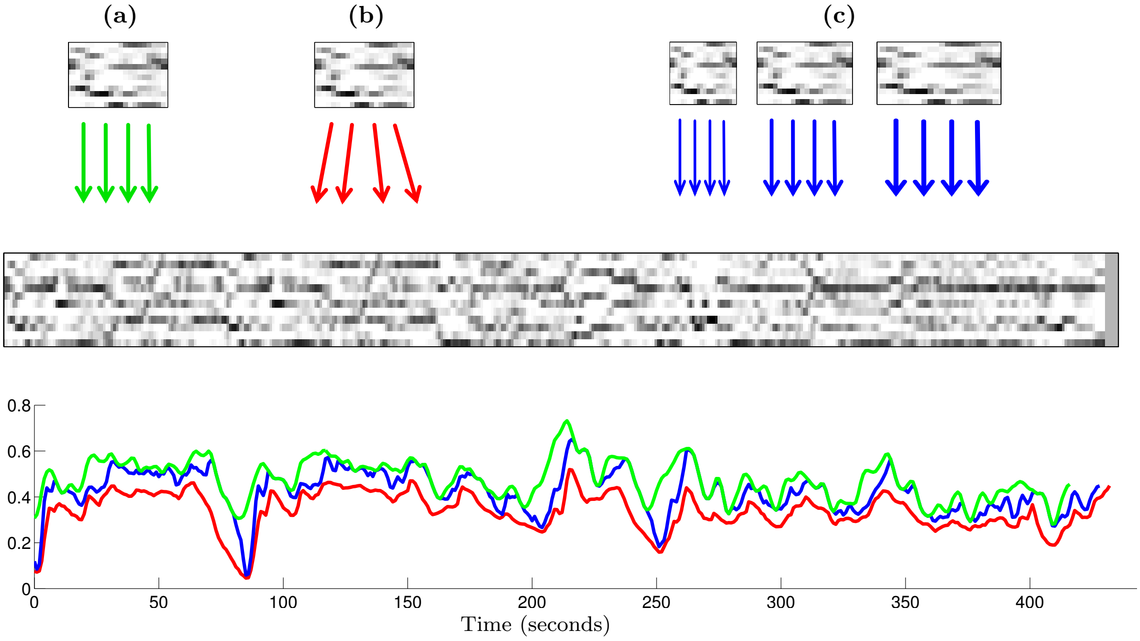


Figure 8.7: Illustration of the the audio matching procedure for the beginning of Beethoven's Opus 67 (Symphony No. 5) using a query fragment corresponding to the first 22 seconds (measures 1-21) of a Bernstein interpretation and a database consisting of an entire recording of a Karajan interpretation. Three different strategies are shown leading to three different matching curves. **(a)** Strict subsequence matching. **(b)** DTW-based matching. **(c)** Multiple query scaling strategy.

unsupervised way using vector quantization or in a supervised way exploiting musical knowledge about chords. The codebook then allows for classifying the chroma vectors of the database and to index the vectors according to the assigned codebook vector. This results in an inverted list for each codebook vector. Then, an exact search can be performed efficiently by intersecting suitable inverted lists. However, the performance of the exact search using quantized chroma vectors greatly depends on the codebook. This requires fault-tolerance mechanisms which partly eliminate the speed-up obtained by this method. Consequently, this approach is only applicable for databases of medium size [105]. An approach presented in [18] uses an index-based near neighbor strategy based on locality sensitive hashing (LSH). Instead of considering long feature sequences, the audio material is split up into small overlapping *shingles* that consist of short chroma feature subsequences. The shingles are then indexed using locality sensitive hashing which allows for scaling this approach to larger datasets. However, to cope with temporal variations, each shingle covers only a small portion of the audio material and queries need to consist of a large number of shingles. The high number of table look-ups induced by this strategy may become problematic for very large datasets where the index is stored on a secondary storage device. In Chapter 10, we present an investigation with the goal to reduce the number of table look-ups by representing each query (consisting of 15-25 seconds of the audio) with only a single shingle. To handle temporal variations, a combination of local feature smoothing and global query scaling is proposed.

In summary, mid-specific audio matching using a combination of highly robust chroma features and sequence-based similarity measures that account for different tempi results in a good retrieval quality. However, the low specificity of this task makes indexing much harder than in the case of audio identification. This task becomes even more challenging when dealing with relatively short fragments on the query and database side.

8.4 Version Identification

In the previous tasks, a musical fragment is used as query and similar fragments or documents are retrieved according to a given degree of specificity. The degree of specificity was very high for audio identification and more relaxed for audio matching. If we allow for even less specificity, we are facing the problem of version identification [161]. In this scenario, a user wants to retrieve not only exact or near-duplicates of a given query, but also any existing re-interpretation of it, no matter how radical such a re-interpretation might be. In general, a version may differ from the original recording in many ways, possibly including significant changes in timbre, instrumentation, tempo, main tonality, harmony, melody, and lyrics. For example, in addition to the aforementioned Karajan's rendition of Beethoven's Symphony No. 5, one could be also interested in a live performance of it, played by a punk-metal band who changes the tempo in a non-uniform way, transposes the piece to another key, and skips many notes as well as most parts of the original structure. These types of documents where, despite numerous and important variations, one can still unequivocally glimpse the original composition are the ones that motivate version identification.

Version identification is usually interpreted as a document-level retrieval task, where a single similarity measure is considered to globally compare entire documents [5; 46; 170]. However, successful methods perform this global comparison on a local basis. Here, the final similarity measure is inferred from locally comparing only parts of the documents—a strategy that allows for dealing with non-trivial structural changes. This way, comparisons are performed either on some representative part of the piece [60], on short, randomly chosen subsequences of it [117], or on the best possible longest matching subsequence [162; 164].

A common approach to version identification starts from the previously introduced chroma features; also more general representations of the tonal content such as chords or tonal templates have been used [161]. Furthermore, melody-based approaches have been suggested, although recent findings suggest that this representation may be suboptimal [55]. Once a tonal representation is extracted from the audio, changes in the main tonality need to be tackled, either in the extraction phase itself, or when performing pairwise comparisons of such representations.

Tempo and timing deviations have a strong effect in the chroma feature sequences, hence making their direct pairwise comparison problematic. An intuitive way to deal with global tempo variations is to use beat-synchronous chroma representations [12; 46]. However, the required beat tracking step is often error-prone for certain types of music and therefore may negatively affect the final retrieval result. Again, as for the audio matching task, dynamic programming algorithms are a standard choice for dealing with tempo variations [123], this time applied in a local fashion to identify longest matching subsequences or local

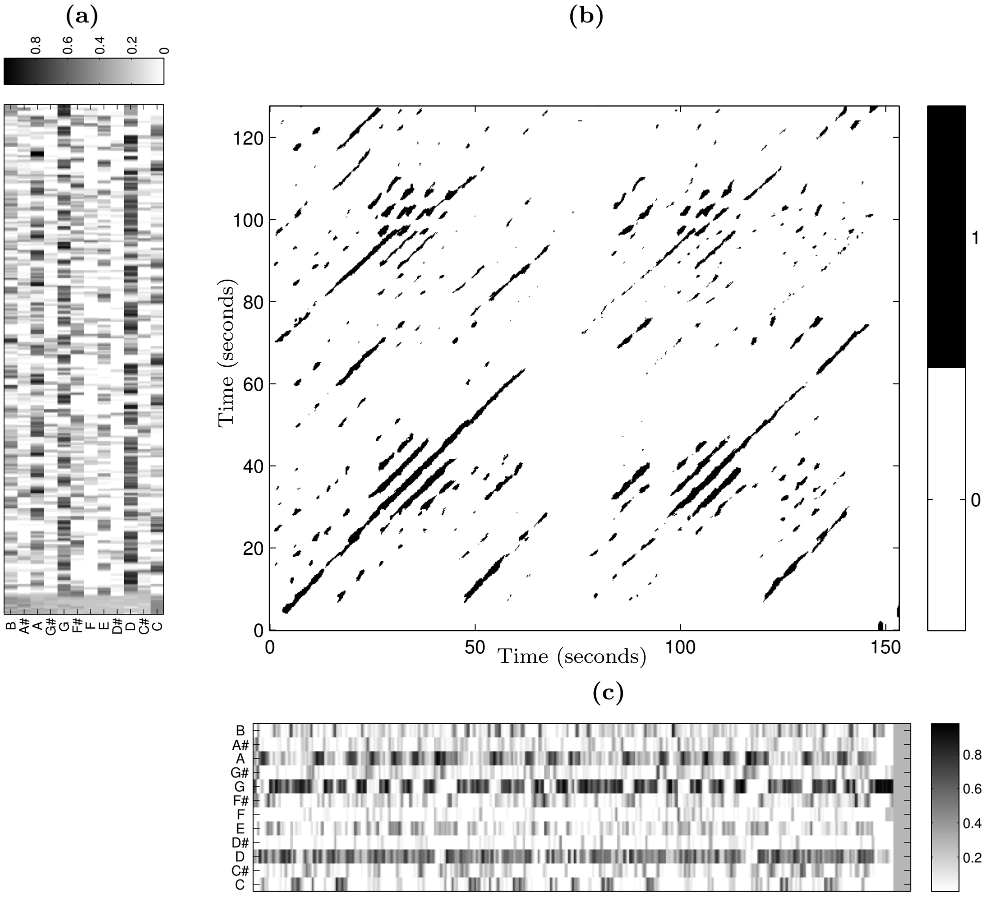


Figure 8.8: Similarity matrix for "Act naturally" by The Beatles, which is actually a cover version of a song by Buck Owens. **(a)** Chroma features of the version by The Beatles. **(b)** Score matrix. **(c)** Chroma features of the version by Buck Owens.

alignments [162; 164].

An example of such an alignment procedure is depicted in Figure 8.8 for our "Act naturally" example by The Beatles. The chroma features of this version are shown in Figure 8.8c. Actually, this song is originally not written by The Beatles but a cover version of a Buck Owens song of the same name. The chroma features of the original version are shown in Figure 8.8a. Alignment algorithms rely on some sort of scores (and penalties) for matching (mismatching) individual chroma sequence elements. Such scores can be real-valued or binary. Figure 8.8b shows a binary score matrix encoding pair-wise similarities between chroma vectors of the two sequences. The binarization of score values provides some additional robustness against small spectral and tonal differences. Correspondences between versions are revealed by the score matrix in the form of diagonal paths of high score. For example, in Figure 8.8, one observes a diagonal path indicating that the first 60 seconds of the two versions exhibit a high similarity.

For detecting such path structures, dynamic programming strategies make use of an accumulated score matrix. In their local alignment version, where one is searching for sub-

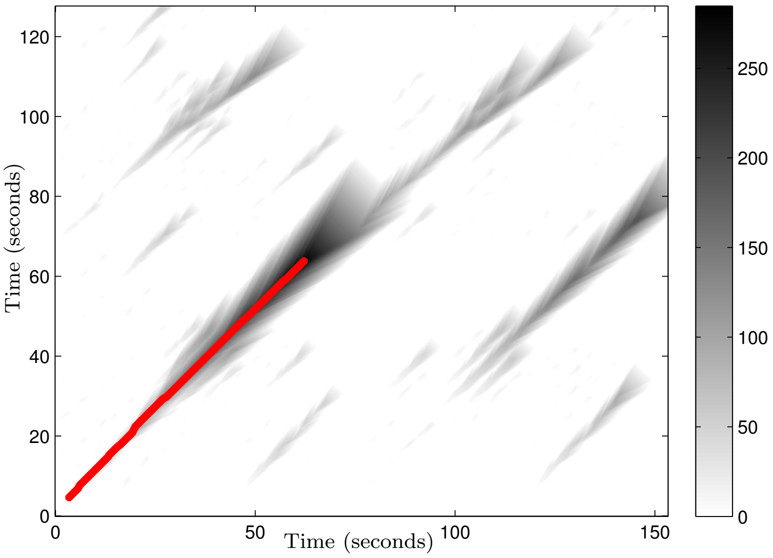


Figure 8.9: Accumulated score matrix with optimal alignment path for the "Act naturally" example (as shown in Figure 8.8).

sequence correspondences, this matrix reflects the lengths and quality of such matching subsequences. Each element (consisting of a pair of indices) of the accumulated score matrix corresponds to the end of a subsequence and its value encodes the score accumulated over all elements of the subsequence. Figure 8.9 shows an example of the accumulated score matrix obtained for the score matrix in Figure 8.8. The highest-valued element of the accumulated score matrix corresponds to the end of the most similar matching subsequence. Typically, this value is chosen as the final score for the document-level comparison of the two pieces. Furthermore, the specific alignment path can be easily obtained by backtracking from this highest element [123]. The alignment path is indicated by the red line in Figure 8.9. Additional penalties account for the importance of insertions/deletions in the subsequences. In fact, the way of deriving these scores and penalties is usually an important part of the version identification algorithms and different variants have been proposed [5; 162; 164]. The aforementioned final score is directly used for ranking candidate documents to a given query. It has recently been shown that such rankings can be improved by combining different scores obtained by different methods [151], and by exploiting the fact that alternative renditions of the same piece naturally cluster together [109; 165].

The task of version identification allows for these and many other new avenues for research [161]. However, one of the most challenging problems that remains to be solved is to achieve high accuracy and scalability at the same time, allowing low-specific retrieval in large music collections [12]. Unfortunately, the accuracies achieved with today's non-scalable approaches have not yet been reached by the scalable ones, the latter remaining far behind the former.

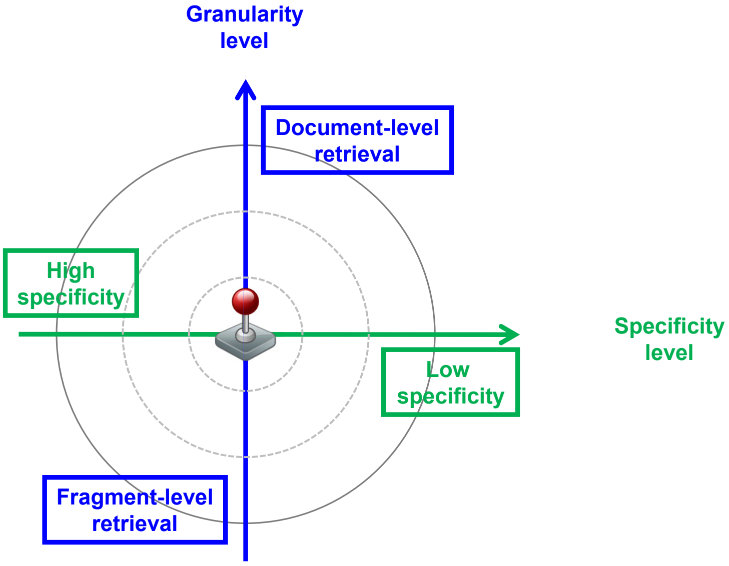


Figure 8.10: Joystick-like user interface for continuously adjusting the specificity and granularity levels used in the retrieval process.

8.5 Further Notes

The retrieval strategies presented in this chapter allow for discovering and accessing music even in cases where the user does not explicitly know what he or she is actually looking for. For designing a retrieval system that increases the user experience, however, one also has to better account for user requirements in content-based retrieval systems. For example, one may think of a comprehensive framework that allows a user to adjust the specificity level at any stage of the search process. Here, the system should be able to seamlessly change the retrieval paradigm from high-specific audio identification, over mid-specific audio matching and version identification to low-specific genre identification. Similarly, the user should be able to flexibly adapt the granularity level to be considered in the search. Furthermore, the retrieval framework should comprise control mechanisms for adjusting the musical properties of the employed similarity measure to facilitate searches according to rhythm, melody, or harmony or any combination of these aspects.

Figure 8.10 illustrates a possible user interface for such an integrated content-based retrieval framework, where a joystick allows a user to continuously and instantly adjust the retrieval specificity and granularity. For example, a user may listen to a recording of Beethoven's Symphony No. 5, which is first identified to be a Bernstein recording using an audio identification strategy (moving the joystick to the leftmost position). Then, being interested in different versions of this piece, the user moves the joystick upwards (document-level) and to the right (mid-specific), which triggers a version identification. Subsequently, shifting towards a more detailed analysis of the piece, the user selects the famous fate motif as query and moves the joystick downwards to perform some mid-specific fragment-based audio matching. Then, the system returns the positions of all occurrences of the motif in all available interpretations. Finally, moving the joystick to the rightmost position, the user may discover recordings of pieces that exhibit some general similarity like style or mood. In combination with immediate visualization, navigation, and feedback mechanisms, the user is able to successively refine and adjust the query formulation as well as the retrieval strategy, thus leading to novel strategies for exploring, browsing, and interacting with large collections of audio content.

Chapter 9

Musically-Motivated Audio Fingerprints

As introduced in the last chapter, the high-specific task of audio identification constitutes an important research topic and is of commercial relevance [1; 15; 181]. In particular, audio identification systems are highly efficient and can deal with music collections comprising millions of songs. However, audio identification systems are not capable of retrieving different performances of the same piece of music. The reason for this is that existing audio fingerprinting algorithms are not designed for dealing with musical variations such as strong non-linear temporal distortions, variations that concern the articulation, instrumentation, or ornamentation. Dealing with such variations, the scalability of mid-specific audio matching and version identification is still problematic.

In this chapter, we investigate to which extent well-established audio fingerprints originally proposed in [181] and introduced in Section 8.2 can be modified to allow for retrieving musically related recordings while retaining the efficiency of index-based approaches. To this end, we replace the traditional fingerprints based on spectral peaks by fingerprints based on peaks of more musically oriented feature representations including log-frequency and chroma representations. Our motivation for adopting this approach is that such peak structures, according to [181], are temporally localized, reproducible, and robust against many, even significant distortions of the signal. Furthermore, the spectral peaks allow for applying efficient hash-based indexing techniques. The main contribution of this chapter is to systematically analyze the resulting peak structures in view of robustness and discriminative power. Finding a good trade-off between these two principles is a non-trivial task. On the one hand, using fine-grained feature representations (such as a spectrogram) results in fingerprints that are too specific, thus not facilitating cross-version retrieval. On the other hand, using coarse feature representations (such as a chromagram) results in peak fingerprints that are too unspecific and noisy, thus not having the required discriminative power.

In our investigation, we proceed in four steps, see Figure 9.1 for an overview. For a piece of music (indicated by the excerpt of the score in Figure 9.1a) we assume to have multiple performances given in the form of audio recordings, see Figure 9.1b. in the first step, for each of the performances, we derive a feature representation (Figure 9.1c). In our

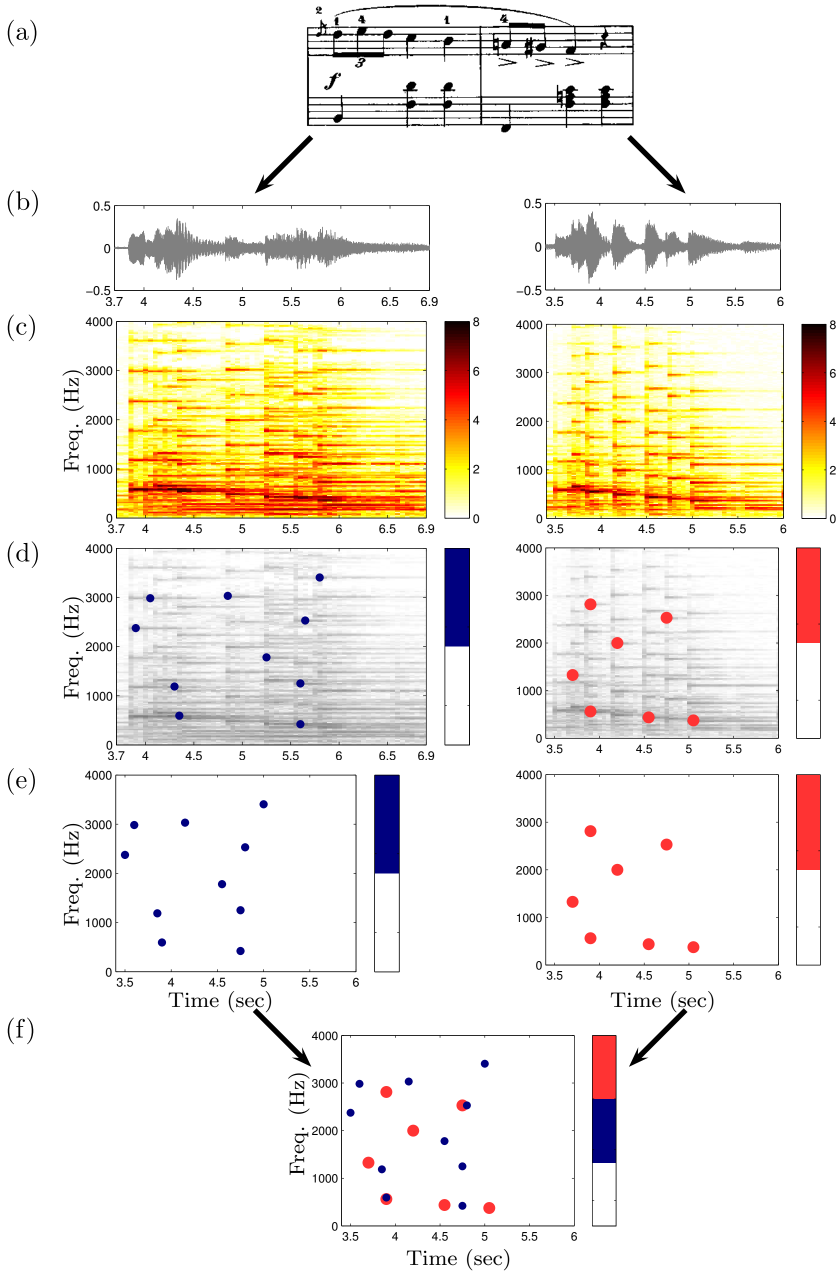


Figure 9.1: Overview of the framework used in our investigation. **(a)** Score of a piece. **(b)** Waveforms of two different recorded performances. **(c)** Feature representations for the performances. **(d)** Peak fingerprints extracted from the feature representations. **(e)** Temporally warped fingerprints based on a common time line. **(f)** Overlayed peak representations indicating peak consistencies.

experiments we actually investigate five different time-frequency representations derived from the originally used spectrogram. In the second step, we derive peak fingerprints from the different feature representations similar to [181], see Figure 9.1d. Next, we investigate which and how many of the peaks consistently appear across different performances. To compensate for temporal differences between performances, we use in the third step music synchronization techniques [49] to warp the peak fingerprints onto a common time line (Figure 9.1e). Finally, in the fourth step, the fingerprints are analyzed with respect to peak consistency across the different performances (Figure 9.1f). Our experimental results in the context of a music retrieval scenario indicate that, using suitably modified peak fingerprints, one can transfer traditional audio fingerprinting techniques to other tasks such as audio matching and cover song identification as introduced in Section 8.3 and Section 8.4, respectively.

The remainder of the chapter is organized as follows. In Section 9.1 we introduce various peak fingerprints based on different feature representations. In Section 9.2, as our main contribution, we systematically investigate the trade-off between robustness and discriminative power of the various audio fingerprints. Finally, discussions and an outlook on a modified audio fingerprinting system can be found in Section 9.3.

9.1 Modified Peak Fingerprints

Our approach is based on the concept of spectral peaks originally introduced by Wang [181] and explained in Section 8.2. In this approach, characteristic time-frequency peaks extracted from a spectrogram are used as fingerprints, thus reducing a complex spectrogram to a sparse peak representation of high robustness against signal distortions. Such peak representations allows for applying efficient hash-based indexing techniques. We transfer this approach to a more flexible retrieval scenario by considering various feature representations that are obtained by partitioning the frequency axis of the original spectrogram, while the temporal axis of all representations is fixed to yield a feature rate of 20 Hz (20 feature per second), see Figure 9.2 for an illustration of the different feature representations.

The first feature representation is a magnitude spectrogram as employed in the original approach. Following [181], the audio signal is sampled at $f_s = 8000$ Hz and discrete Fourier transforms are calculated over windows of 1024 samples. In the following, the resulting feature representation is referred to as `SPEC`, see Figure 9.2b. The second feature representation is a log-frequency spectrogram [15]. Using a suitable binning strategy, we group the Fourier coefficients of the original spectrogram into 33 non-overlapping frequency bands covering the frequency range from 300 Hz to 2000 Hz. Exhibiting a logarithmic spacing, the bands roughly represent the Bark scale. In the following, this feature representation is referred to as `LOGF`, see Figure 9.2c. As third feature representation, we consider a constant-Q transform where the frequency bins are logarithmically spaced and the ratios of the center frequencies to bandwidths of all bins are equal (Q factor). In our investigation, we employ the efficient implementation provided by the *Constant-Q Transform Toolbox for Music Processing*[1], see [155]. Here, we set the number of frequency bins per octave to 12 (each bin corresponds to one semitone of the equal-tempered scale) and consider the frequency range from 80 Hz to 4000 Hz. In the following, this feature is referred to as `CONSTQ`, see Figure 9.2d. To obtain the fourth feature representation, we decom-

[1] http://www.elec.qmul.ac.uk/people/anssik/cqt/

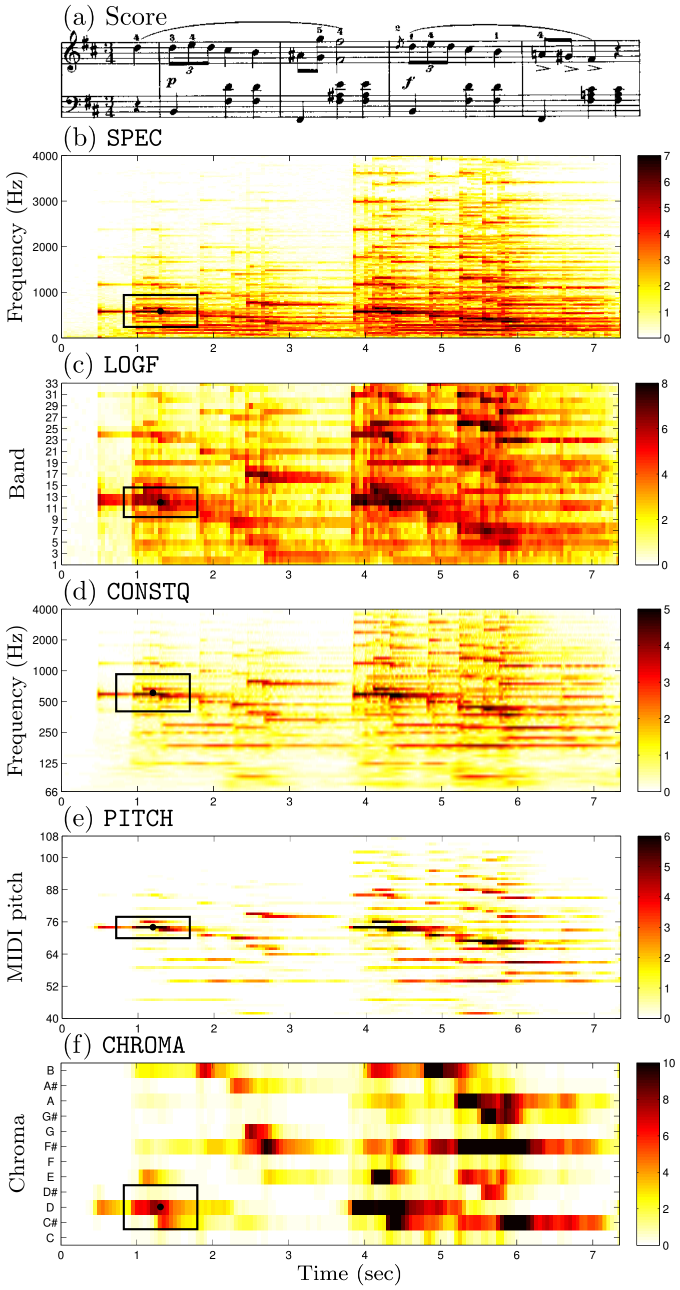


Figure 9.2: Score and various feature representations for the first 7.35 seconds of a Hatto (2006) performance of the first 5 bars of Chopin's Mazurka Op. 30 No. 2. One peak and the corresponding neighborhood is shown for each of the feature representations.

pose the audio signal into 88 frequency bands with center frequencies corresponding to the pitches of the equal-tempered scale and compute the short-time energy in windows of length 100 ms. For deriving this decomposition, we use a multirate filter bank as described in [123] and denote the resulting feature as PITCH, see Figure 9.2e. The fifth feature representation is a chroma representation which is obtained from PITCH by adding up the corresponding values that belong to the same chroma. In the following, this feature is referred to as CHROMA, see Figure 9.2e. Implementations for PITCH and CHROMA are provided by the *Chroma Toolbox*[2], see [127].

In the second step, we employ a similar strategy as proposed in [181] to extract characteristic peaks from the various feature representations. Given a feature representation $\mathcal{F} \in \mathbb{R}^{T \times K}$ where $\mathcal{F}(t,k)$ denotes the feature value at frame $t \in [1:T] := \{1,2,\ldots,T\}$ for some $T \in \mathbb{N}$ and frequency bin $k \in [1:K]$ for some $K \in \mathbb{N}$, we select a point (t_0,k_0) as a peak if $\mathcal{F}(t_0,k_0) \geq \mathcal{F}(t,k)$ for all $(t,k) \in \left[t - \Delta^{\mathrm{time}} : t + \Delta^{\mathrm{time}}\right] \times \left[k - \Delta^{\mathrm{freq}} : k + \Delta^{\mathrm{freq}}\right]$ in a local neighborhood defined by Δ^{time} and Δ^{freq}. The size of this neighborhood allows for adjusting the peak density. In our implementation, we use an additional absolute threshold on the values $\mathcal{F}(t_0,k_0)$ to prevent the selection of more or less random peaks in regions of very low dynamics. The selected peaks are represented in the form of a binary matrix $\mathcal{P} \in \{0,1\}^{T \times K}$ by setting $\mathcal{P}(t_0,k_0) = 1$ for (t_0,k_0) being a peak and zero elsewhere. This peak selection strategy reduces a complex time-frequency representation $\mathcal{F}$ to a sparse set $\mathcal{P}$ of time-frequency points. Note that the values of $\mathcal{F}(t,k)$ are no longer considered in the fingerprints.

In our experiments, we fix $\Delta^{\mathrm{time}} = 20$ corresponding to one second for all five feature representations. The range of the frequency neighborhood Δ^{freq}, however, was experimentally determined for each feature representation. For SPEC we set $\Delta^{\mathrm{freq}} = 25$ (corresponding to 200 Hz), for LOGF we set $\Delta^{\mathrm{freq}} = 2$, for CONSTQ we set $\Delta^{\mathrm{freq}} = 3$, for PITCH we set $\Delta^{\mathrm{freq}} = 3$, and for CHROMA we set $\Delta^{\mathrm{freq}} = 1$, see Figure 9.2 for an illustration of the neighborhood for each of the feature representations.

9.2 Experiments

We now investigate the musical expressiveness of the various peak fingerprints. In Section 9.2.1, we start with introducing the datasets used in our experiments. Then, in Section 9.2.2, we sketch how the peaks of different performances are warped to a common time line. In Section 9.2.3, we discuss an experiment that indicates the degree of peak consistency across different performances depending on the underlying feature representation. Finally, in Section 9.2.4, we describe a document-based retrieval experiment.

9.2.1 Dataset

For our subsequent experiments, we use three different groups of audio recordings corresponding to pieces of classical music by three different composers, see Table 9.1. The first group **Chop** consists of 298 piano recordings of five Mazurkas by Frédéric Chopin collected in the Mazurka Project.[3] The second group **Beet** consists of ten recorded performances of Beethoven's *Symphony No. 5*. This collection contains orchestral as well as piano performances. The third group **Viva** contains seven orchestral performances of the *Summer* from Vivaldi's *Four Seasons*. Table 9.1 lists the number of performances as well as the total duration of each movement/piece. The union of all groups is referred to as

[2] http://www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/

[3] http://mazurka.org.uk/

Groups	Composer	Piece	Description	#(Perf.)	Dur. (min)
Chop	Chopin	Op. 17, No. 4	Mazurka	62	269
	Chopin	Op. 24, No. 2	Mazurka	64	147
	Chopin	Op. 30, No. 2	Mazurka	34	48
	Chopin	Op. 63, No. 3	Mazurka	88	189
	Chopin	Op. 68, No. 3	Mazurka	50	84
Beet	Beethoven	Op. 67, 1. Mov.	Fifth	10	75
	Beethoven	Op. 67, 2. Mov.	Fifth	10	98
	Beethoven	Op. 67, 3. Mov.	Fifth	10	52
	Beethoven	Op. 67, 4. Mov.	Fifth	10	105
Viva	Vivaldi	RV 315, 1. Mov.	Summer	7	38
	Vivaldi	RV 315, 2. Mov.	Summer	7	17
	Vivaldi	RV 315, 3. Mov.	Summer	7	20
All				359	1145

Table 9.1: The groups of audio recordings used in our experiments. The last two columns denote the number of different performances and the overall duration in minutes.

All and contains 359 recordings with an overall length of 19 hours. In view of extracting peak fingerprints, these three groups are of increasing complexity. While for the piano recordings of **Chop**, one expects relatively clear peak structures, peak picking becomes much more problematic for general orchestral music (group **Beet**) and music dominated by strings (group **Viva**).

9.2.2 Synchronization of Fingerprints

In our retrieval scenario, there typically are tempo differences between the different interpretations of a piece. In our initial experiments, we do not want to deal with this issue and compensate for tempo differences in the performances by temporally warping the peak representations onto a common time line. To this end, we, in a preprocessing step, use a music synchronization technique [49] to temporally align the different performances of a given piece of music. More precisely, suppose we are given N different performances of the same piece yielding the peak representations $\mathcal{P}_n$, $n \in [1:N]$. Then, we take the first performance as reference and compute alignments between the reference and the remaining $N-1$ performances. The alignments are then used to temporally warp the peak representations $\mathcal{P}_n$ for $n \in [2:N]$ onto the time axis of the peak representation $\mathcal{P}_1$. The resulting warped peak fingerprints are denoted by $\tilde{\mathcal{P}}_n$ and we set $\tilde{\mathcal{P}}_1 = \mathcal{P}_1$, see Figure 9.1e for an illustration.

9.2.3 Experiment: Peak Consistency

In a first experiment, we investigate to which extent the various peak fingerprints coincide across different performances of a piece. Here, the degree of peak consistency serves as an indicator for the robustness of the respective feature representation towards musical variations. We express the consistency of the fingerprints of two performances in terms of pairwise precision P, recall R, and F-measure F. More precisely, given two performances $n, m \in [1:N]$ of a piece, we obtain the aligned peak fingerprints $\tilde{\mathcal{P}}_n$ and $\tilde{\mathcal{P}}_m$ as explained

Groups	SPEC	LOGF	CONSTQ	PITCH	CHROMA
Chop	0.081	0.205	0.157	0.185	0.375
Beet	0.051	0.139	0.126	0.137	0.288
Viva	0.059	0.143	0.124	0.132	0.262
All	0.080	0.203	0.156	0.184	0.373

Table 9.2: Mean of pairwise F-measure values expressing peak consistencies for the different groups.

in Section 9.2.2. Then, a peak (t_0, k_0) of $\tilde{\mathcal{P}}_m$ is called *consistent* relative to $\tilde{\mathcal{P}}_n$ if there is a peak (t, k_0) of $\tilde{\mathcal{P}}_n$ with $t \in [t_0 - \tau : t_0 + \tau]$. Here, the parameter $\tau \geq 0$ specifies a small temporal tolerance window. Otherwise, the peak is called *non-consistent*. The number of consistent fingerprints is denoted by $\#(\tilde{\mathcal{P}}_n \cap \tilde{\mathcal{P}}_m)$, the overall number of peaks in $\tilde{\mathcal{P}}_n$ and $\tilde{\mathcal{P}}_m$ is denoted $\#(\tilde{\mathcal{P}}_n)$ and $\#(\tilde{\mathcal{P}}_m)$, respectively. Then, pairwise precision $\mathrm{P}_{n,m}$, recall $\mathrm{R}_{n,m}$, and F-measure $\mathrm{F}_{n,m}$ are defined as

$$\mathrm{P}_{n,m} = \frac{\#(\tilde{\mathcal{P}}_n \cap \tilde{\mathcal{P}}_m)}{\#(\tilde{\mathcal{P}}_m)}\ , \quad \mathrm{R}_{n,m} = \frac{\#(\tilde{\mathcal{P}}_n \cap \tilde{\mathcal{P}}_m)}{\#(\tilde{\mathcal{P}}_n)}\ , \quad \mathrm{F}_{n,m} = \frac{2 \cdot \mathrm{P}_{n,m} \cdot \mathrm{R}_{n,m}}{\mathrm{P}_{n,m} + \mathrm{R}_{n,m}}\ . \tag{9.1}$$

Note, that $\mathrm{P}_{n,m} = \mathrm{R}_{m,n}$, $\mathrm{R}_{n,m} = \mathrm{P}_{m,n}$, and therefore $\mathrm{F}_{n,m} = \mathrm{F}_{m,n}$. F-measure values are computed for all N performances of a group yielding an $(N \times N)$-matrix of pairwise F values. Mean values for the groups are obtained by averaging over the respective F-measures. Here, as $\mathrm{F}_{n,n} = 1$ and $\mathrm{F}_{n,m} = \mathrm{F}_{m,n}$, we only consider the values of the upper triangular part of the matrix excluding the main diagonal.

Table 9.2 shows the mean of pairwise F-measure values for the different groups of our dataset. In this experiment, we use the tolerance parameter $\tau = 1$ (corresponding to ±50 ms), which turned out to be a suitable threshold for compensating inaccuracies introduced by the synchronization procedure, see [49]. First note that the originally used spectrogram peaks do not work well across different performances. For example, in the case of **Chop**, one obtains F = 0.081 for SPEC indicating that only few of the peak fingerprints consistently occur across different performances. The peaks extracted from the other four feature representations show a higher degree of consistency across performances e.g., in the case of **Chop**, F = 0.205 for LOGF, F = 0.157 for CONSTQ, F = 0.185 for PITCH, and F = 0.375 for CHROMA. This improvement is achieved by the coarser and musically more meaningful partition of the frequency axis. Furthermore, our results show a dependency on the characteristics of the audio material. In particular, the peaks are more consistent for **Chop** (e.g. F = 0.375 for CHROMA) than for **Beet** (F = 0.288) and **Viva** (F = 0.262). The reason for this effect is twofold. Firstly, the piano pieces as contained in **Chop** exhibit pronounced note onsets leading to consistent peaks. For complex orchestral and string music as in **Beet** and **Viva**, however, the peaks are less dominant leading to a lower consistency. Secondly, the consistency results are also influenced by the accuracy of the peak synchronization as introduced in Section 9.2.2. Typically, the synchronization technique [49] yields very precise alignments for piano music as contained in **Chop**. For orchestral and string pieces as in **Beet** and **Viva**, however, there are more synchronization inaccuracies leading to lower F-measure values.

This effect is further addressed by Figure 9.3 showing mean F-measure values as a function of the tolerance parameter τ for the different groups. In general, the tolerance parameter

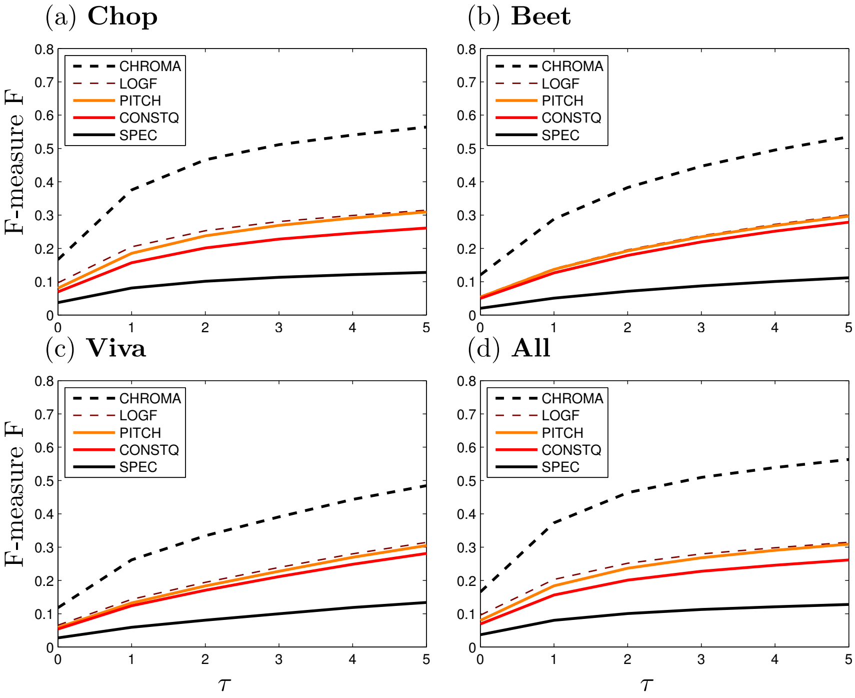


Figure 9.3: Dependency of the mean of pairwise F-measure values on the tolerance parameter τ for **(a)** group **Chop**, **(b)** group **Beet**, **(c)** group **Viva**, and **(d)** the union of all groups **All**.

has a large influence on the evaluation results. In particular, one observes a large increase in F-measure values when introducing a tolerance of $\tau = 1$ (in comparison with $\tau = 0$) regardless of feature type and group. Note that for **Chop** (Figure 9.3a), further increasing τ has a smaller effect than for **Beet** (Figure 9.3b) and **Viva** (Figure 9.3c). In the following experiment, we set $\tau = 1$.

9.2.4 Experiment: Document-Based Retrieval

In the second experiment, we investigate the identification rate of the modified peak fingerprints in a document-based retrieval scenario. Given a short query extracted from one performance, the goal is to correctly retrieve all performances of the same piece from a larger dataset. Exploiting the warped peak fingerprints $\tilde{\mathcal{P}}$ (see Section 9.2.2), we define a query Q and a database collection $\mathcal{D}$. The database consists of $|\mathcal{D}|$ performances (documents) of different groups. For a query Q and a document $D \in \mathcal{D}$, we compute the peak consistency F-measure as in Eq. (9.1) between Q and all subsegments of D having the same length as Q. High F-values indicate high degrees of peak consistency between Q and subsegments of D. Considering document-level retrieval, the similarity between Q and D is defined as the maximum F-measure over all subsegments of D.

In the evaluation, given the set $\mathcal{D}_Q \subset \mathcal{D}$ of documents that are relevant to the query Q (i. e., interpretations of the piece underlying the query), we follow [164] and express the retrieval accuracy using the *mean of average precision* (MAP) measure denoted as

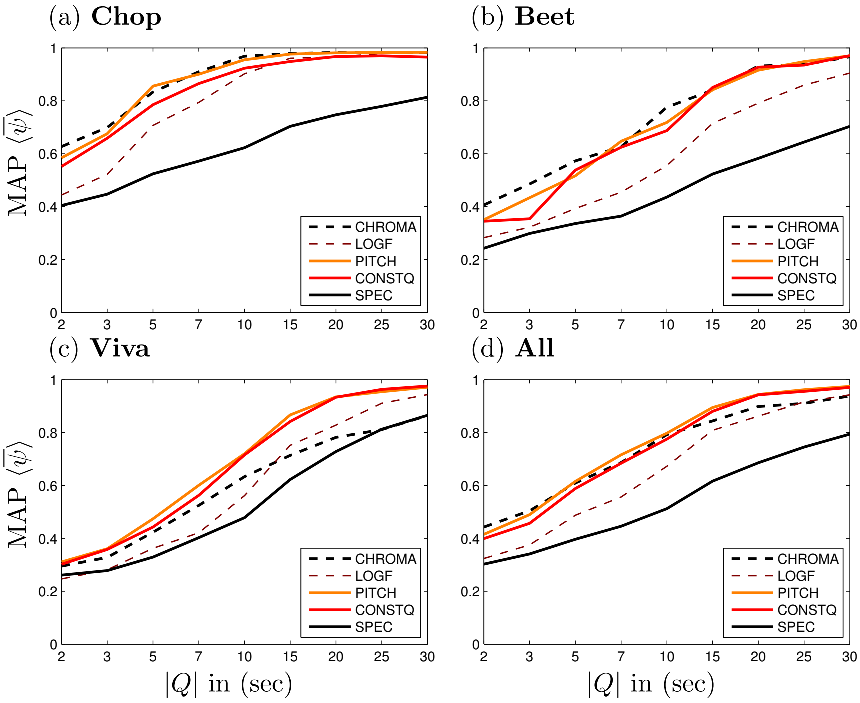

(a) **Chop**
(b) **Beet**
(c) **Viva**
(d) **All**

Figure 9.4: Results for the retrieval experiment showing the dependency of MAP values $\langle\overline{\psi}\rangle$ on the query length $|Q|$ using queries from **(a) Chop** ($\langle\overline{\psi}\rangle_{\text{null}} = 0.190$), **(b) Beet** ($\langle\overline{\psi}\rangle_{\text{null}} = 0.040$), **(c) Viva** ($\langle\overline{\psi}\rangle_{\text{null}} = 0.032$), and **(d)** average over all queries.

$\langle\overline{\psi}\rangle$.[4] To this end, we sort the documents $D \in \mathcal{D}$ in descending order with respect to the similarity between D and Q and obtain the precision ψ_Q at rank $r \in [1 : |\mathcal{D}|]$ as

$$\psi_Q(r) = \frac{1}{r}\sum_{i=1}^{r}\Gamma_Q(i) \ , \tag{9.2}$$

where $\Gamma_Q(r) \in \{0, 1\}$ indicates whether a document at rank r is contained in $\mathcal{D}_Q$. Then, the average precision $\overline{\psi}_Q$ is defined as

$$\overline{\psi}_Q = \frac{1}{|\mathcal{D}_Q|}\sum_{r=1}^{|\mathcal{D}|}\psi_Q(r)\Gamma_Q(r) \ . \tag{9.3}$$

Finally, given C different queries we compute $\overline{\psi}_Q$ for each Q and average over all C values to obtain the mean of average precision measure $\langle\overline{\psi}\rangle$. In our experiments, for a fixed query length $|Q|$, we randomly select $C = 100$ queries from each group. Additionally, we estimate the accuracy level $\langle\overline{\psi}\rangle_{\text{null}}$ expected under the null hypothesis of a randomly created sorted list, see [164] for details.

Figure 9.4 shows the resulting MAP $\langle\overline{\psi}\rangle$ values as a function of the query length $|Q|$ for the five features. The queries are taken from the different groups, the database $\mathcal{D}$ contains

[4] The same measure is used in the MIREX Cover Song Identification, see www.music-ir.org/mirex/wiki/2010:Audio_Cover_Song_Identification

all performances of **All**. As the results show, the retrieval accuracy using modified peak fingerprints is much higher than using the original spectrogram peaks. In particular, using the musically motivated features PITCH, CONSTQ, and CHROMA results in the highest MAP $\langle\overline{\psi}\rangle$ for **All**, see Figure 9.4d. For **Viva** (Figure 9.4c), the retrieval accuracy for PITCH and CONSTQ is significantly higher than for CHROMA. Here, a manual inspection revealed that the peaks of CHROMA, although more consistent across performances than peaks of PITCH and CONSTQ (see Section 9.2.3), exhibit less discriminative power. In the case of the less pronounced peaks of **Viva**, this frequently results in undesired high consistency for unrelated fingerprints when using CHROMA. Contrary, the higher discriminative power of peaks from PITCH and CONSTQ (although of lower overall consistency) results in higher retrieval accuracies.

Furthermore, the results show a great dependency of the retrieval accuracy on the query length $|Q|$. Surprisingly, in the case of **Chop** (Figure 9.4a), even $|Q| = 2$ sec leads to already relatively high MAP values. Increasing the query length, the MAP values increase for all feature representations and groups of audio recordings. For all groups, using a query length of 20 sec in combination with peak fingerprints extracted from PITCH or CONSTQ results in MAP values $\langle\overline{\psi}\rangle > 0.9$. In particular for the more complex data contained in **Beet** (Figure 9.4b) and **Viva** (Figure 9.4c) using longer queries further improves the identification rate across performances.

9.3 Further Notes

For cross-version retrieval scenarios, one needs retrieval systems that can handle variations with regard to musical properties such as tempo, articulation, timbre or instrumentation. Dealing with a much lower specificity level as in the fingerprinting scenario, the development of efficient cross-version retrieval systems that scale to huge data collections still faces challenging problems to be researched.

In this chapter, we studied the robustness and discriminative power of modified audio fingerprints by considering peak consistencies across different versions of the same piece of music. As our experiments reveal, peak fingerprints based on musically motivated time-pitch or time-chroma representations allow for an identification of different performances of the same piece of music. In contrast to 3-5 sec long queries considered for traditional audio fingerprinting, 15-25 sec are necessary for obtaining a robust and accurate cross-performance identification procedure. Our results indicate that, using more musical feature representations, it is possible to employ similar techniques as used by Shazam for other music retrieval tasks such as audio matching or cover song retrieval.

In our investigation, temporal differences between performances were not considered but compensated in a preprocessing step using an offline music synchronization technique. In particular, for designing an efficient and scalable system, indexing techniques based on robust and discriminative hashes that can cope with temporal differences between performances need to be researched.

In [12], a peak-based strategy based on “chroma landmarks” is proposed and applied to cover song detection on the Million Song Dataset [11]. The authors address the problem of temporal differences between the cover versions by computing beat-synchronized chroma features. In the resulting feature sequence, each chroma vector corresponds to exactly one beat interval of the music recording. As a result, the extracted fingerprint peaks are (in theory) invariant against temporal distortions. As discussed in Part I of this thesis, however, automatic beat tracking is a challenging task even for pop and rock music. For classical music, as considered in our scenario, automatic beat tracking becomes even more difficult and often results in a large number of beat tracking errors.

Chapter 10

Characteristic Audio Shingles

In this chapter, we address the fundamental issue on how cross-version retrieval can be accelerated by employing index structures that are based on suitably designed elementary building blocks. Our approach is built upon ideas of two recently proposed retrieval systems [18; 105] introduced in Section 8.3. In [105], a matching procedure is described that allows for a fragment-based retrieval of all audio excerpts musically related to a given query audio fragment. To this end, the query and all database documents are converted to sequences of chroma-based audio features. To cope with temporal variations, global scaling techniques are employed to derive multiple queries that simulate different tempi. Finally, feature quantization techniques in combination with inverted file indexing is applied to speed up the retrieval process. The authors report on speed-up factors of 10-20 for medium sized data collections. However, using a codebook of fixed size, this approach does not scale well to collections of millions of songs. In [18], a different approach is described. Instead of considering long feature sequences, the audio material is split up into small overlapping *shingles* that consist of short chroma feature subsequences. These shingles are indexed using locality sensitive hashing. While being very efficient (the authors report on a speed-up factor of 100) and scalable to even large data collections, the proposed shingling approach has one major drawback. To cope with temporal variations in the versions, each shingle covers only a small portion of the audio material (three seconds in the proposed system). As a result, an individual shingle is too short to characterize well a given piece of music. Therefore, to obtain a meaningful retrieval result, one needs to combine the information retrieved for a large number of query shingles. As a consequence, many hash-table lookups are required in the retrieval process. This becomes particularly problematic, when the index structure is stored on a secondary storage device.

Based on ideas of these two approaches, we systematically investigate how one can significantly reduce the number of hash-table lookups. Our main idea is to use a shingling approach, where an individual shingle covers a relatively large portion of the audio material (between 10 and 30 seconds). Compared to short shingles, such large shingles have a higher musical relevance so that a much lower number of shingles suffices to characterize a given piece of music. However, increasing the size of a shingle comes at the cost of increasing the dimensionality and possibly loosing robustness to temporal variations. Building on well-known existing techniques, the main contribution is to systematically investigate the delicate trade-off between the query length, feature parameters, shingle dimension, and index settings. In particular, we experimentally determine a setting that allows for retrieving most versions of a piece of music when using only a single 120-dimensional shingle covering roughly 20 seconds of the audio material. For dealing with temporal variations, we investigate two conceptually different matching strategies. Furthermore, we show that such large shingles can still be indexed using locality sensitive hashing with only a small degradation in retrieval quality.

The remainder of this chapter is organized as follows. First, we introduce the overall retrieval approach (Section 10.1) and the two matching strategies (Section 10.2). Then, in Section 10.3, we report on our systematic experiments. Conclusions are given in Section 10.4.

10.1 Cross-Version Retrieval Strategy

In our cross-version retrieval scenario, given a short fragment of a music recording as query, the goal is to retrieve all music recordings (documents) that contain a passage similar to the query from a large dataset. The retrieval result for a query is given as a ranked list of document identifiers. To this end, we proceed in three steps. Given a query Q and a document D to be compared, the first step consists in converting Q and D into sequences of feature vectors $X = (x_1, \dots, x_M)$ and $Y = (y_1, \dots, y_N)$, respectively. In our system, as in [105; 18], we use 12-dimensional chroma-based audio features, which are a powerful mid-level representation for capturing harmonic content in music recordings, while being robust to other musical aspects. See Section 8.3 for a more detailed introduction to chroma features in the audio matching context. More precisely, as in Section 5.2, we use a chroma variant referred to as CENS[1] features [123], which involve a temporal smoothing by averaging chroma vectors over a window of length w and downsampling by a factor of d. In our experiments, we use a feature rate of 10 Hz for the basic chroma vectors. Then, for example, setting $d = 10$ and $w = 41$ results in one feature vector per second (a feature resolution of 1 Hz), where each vector is obtained by averaging over 41 consecutive frames, corresponding to roughly 4 sec of the audio. The resulting features $\texttt{CENS}(w, d)$ show an increased robustness to local tempo changes and allow for flexibly adjusting the temporal resolution, see Figure 8.5.

In the second step, the sequence X is compared with subsequences $Y_t^M := (y_t, \dots, y_{t+M-1})$ of length M for $t \in [1 : N - M + 1]$. Here, we adopt the idea of audio shingles [18] and reorganize the sequences of feature vectors into shingle vectors. In our system, we represent each query Q as a single shingle of dimension $M \times 12$. Then, we use the cosine measure to obtain a similarity value between X and all subsequences of Y of length M defined as

$$s(X, Y_t^M) = \frac{\langle X | Y_t^M \rangle}{\|X\| \cdot \|Y_t^M\|}, \tag{10.1}$$

where $\|\cdot\|$ denotes the Euclidean norm. In the third step, we then express the document-

[1] *Chroma Energy Normalized Statistics* features, provided by the Chroma Toolbox www.mpi-inf.mpg.de/resources/MIR/chromatoolbox

wise similarity of Q and D as

$$S(Q, D) = \max_{t \in [1:N-M+1]} \left(s(X, Y_t^M) \right) \ . \tag{10.2}$$

Given Q and a dataset $\mathcal{D}$ containing $|\mathcal{D}|$ documents, we compute S between Q and all $D \in \mathcal{D}$ and rank the result by descending $S(Q, D)$. In practice, however, such an exhaustive search strategy is not needed to find the relevant documents. Instead, one tries to efficiently cut down the set of candidate subsequences using index-based strategies such as locality sensitive hashing (LSH) and computes S in Eq. (10.2) using only the retrieved shingles (setting $s(X, Y_t^M) = 0$ for non-retrieved shingles Y_t^M).

Given the set $\mathcal{D}_Q \subset \mathcal{D}$ of documents that are relevant to the query Q, we follow [164] and express the retrieval accuracy in terms of the *mean of average precision* (MAP) measure as introduced in Section 9.2.4. Using several queries, we compute $\overline{\psi}_Q$ (see Eq. (9.3)) for each Q and average over all values to obtain the MAP value $\langle\overline{\psi}\rangle$. Furthermore, we determine $\langle\overline{\psi}\rangle_{\text{null}}$ expected under the null hypothesis of a randomly created sorted list as in [164].

10.2 Tempo-Invariant Matching Strategies

Typically there are tempo differences in the versions considered in our retrieval scenario. As a result, a musical passage represented by a query can be realized in another version with significant temporal differences. In that case, our choice of representing each query as a single shingle would require a comparison of shingles representing feature sequences of differing length. One approach to this problem is to use similarity measures based on dynamic time warping (DTW) or Smith-Waterman [162]. However, regarding computationally efficiency and an application in the indexing context, such procedures are problematic. Instead, we employ the *query scaling strategy* as proposed in [105]. Here, tempo differences are handled by creating R scaled variants of the query $Q^{(1)}, \ldots, Q^{(R)}$, each simulating a global change in the tempo of the query. The similarity value between D and Q is then defined as

$$S(Q, D) = \max_{r \in [1:R]} \left(S(Q^{(r)}, D) \right) \ . \tag{10.3}$$

Furthermore, as a baseline strategy, we handle tempo difference between Q and D using an offline *DTW-based procedure* [49] that ensures that corresponding feature sequences coincide in all versions. This idealized procedure serves as reference in our experiments as it provides an optimal estimate of $S(Q, D)$ even in the case of strong non-linear temporal distortions.

10.3 Experiments

In this section, we describe our systematic experiments to investigate the influence certain parameters have on the trade-off between efficiency and shingle characteristic. First, in Section 10.3.1, we introduce our dataset. Then, in Section 10.3.2, we report on a first experiment investigating how long a query Q needs to be to accurately characterize all versions

	Composer	Piece	Description	#	Dur. (min)
Chop	Chopin	Op. 17, No. 4	Mazurka	62	269
	Chopin	Op. 24, No. 2	Mazurka	64	147
	Chopin	Op. 30, No. 2	Mazurka	34	48
	Chopin	Op. 63, No. 3	Mazurka	88	189
	Chopin	Op. 68, No. 3	Mazurka	50	84
Beet	Beethoven	Op. 67, 1. Mov.	Fifth	10	75
	Beethoven	Op. 67, 2. Mov.	Fifth	10	98
	Beethoven	Op. 67, 3. Mov.	Fifth	10	52
	Beethoven	Op. 67, 4. Mov.	Fifth	10	105
Viva	Vivaldi	RV 315, 1. Mov.	Summer	7	38
	Vivaldi	RV 315, 2. Mov.	Summer	7	17
	Vivaldi	RV 315, 3. Mov.	Summer	7	20
$\mathcal{D}_{\text{Queries}}$				359	1145
$\mathcal{D}$				2484	9725

Table 10.1: The music collection used in our experiments. The last two columns denote the number of different performances and the duration in minutes.

of the underlying piece and what a suitable feature resolution is. In Section 10.3.3, we analyze how well tempo differences between the versions can be handled by the query scaling approach (avoiding computationally problematic warping procedures). In Section 10.3.4, we work out whether the shingle dimension can be further reduced using principal component analysis (PCA). Finally, in Section 10.3.5, we analyze how cross-version retrieval can be accelerated by indexing the resulting shingles using locality sensitive hashing (LSH) and how much accuracy is lost in this step.

10.3.1 Dataset

In our experiments, we use a dataset $\mathcal{D}$ of 2484 audio recordings with an overall runtime of 162 hours, see Table 10.1. A subset (denoted $\mathcal{D}_{\text{Queries}}$) of 359 recordings is used for obtaining queries. This part of the dataset corresponds to the dataset introduced in Section 9.2.1. These recordings correspond to classical music pieces by three different composers. For each piece, there are 7 to 88 different recorded versions available. More precisely, the first part **Chop** consists of 298 piano recordings of five Mazurkas by Frédéric Chopin.[2] The second part **Beet** consists of ten recorded performances of Beethoven's *Symphony No. 5* in orchestral as well as piano interpretations. The third part **Viva** contains seven orchestral performances of the *Summer* from Vivaldi's *Four Seasons*. Additionally, we add 2125 recordings of various genre to enlarge the dataset. In our experiments, we randomly select 100 queries from each of the three parts of $\mathcal{D}_{\text{Queries}}$ and average the results over the resulting 300 queries.

10.3.2 Evaluation of Query Length and Feature Resolution

In a first experiment, we investigate how much of a recording needs to be captured by the query Q to robustly characterize all versions of the underlying piece. Furthermore, we

[2] This data is provided by the Mazurka Project http://mazurka.org.uk/

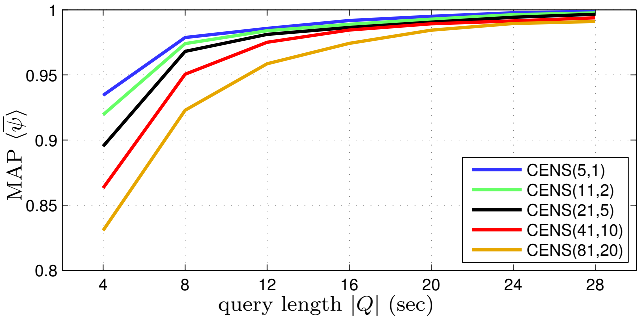


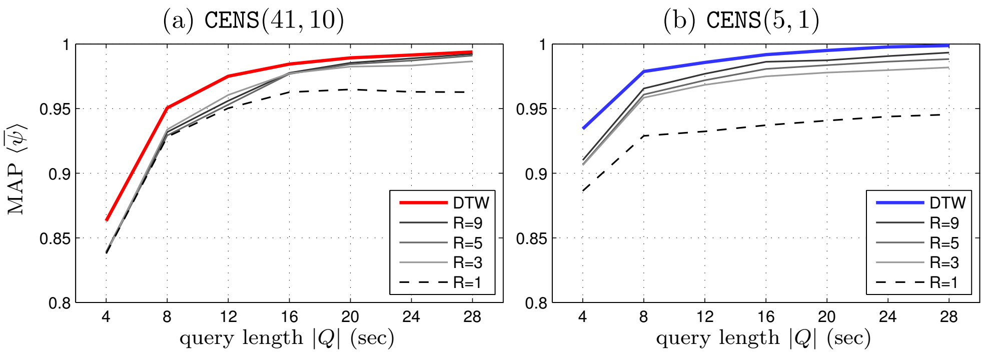
Figure 10.1: MAP values as a function of query length $|Q|$ using $\texttt{CENS}(w, d)$ in different feature resolutions. Null hypothesis $\langle\overline{\psi}\rangle_{\text{null}} = 0.015$.

analyze to what extent the temporal resolution of the features can be reduced without negatively affecting the retrieval quality. Here, we exploit the downsampling and smoothing parameters d and w of the $\texttt{CENS}(w, d)$ features. The goal is to reduce the overall dimensionality of the query while retaining as much of the retrieval accuracy as possible. For the moment, we use the DTW-based procedure to account for tempo differences between the versions.

Figure 10.1 shows MAP values obtained using $\texttt{CENS}(w, d)$ features with seven different query lengths $|Q|$ and five different feature resolutions. Obviously, the longer $|Q|$ the higher the retrieval quality. For example, for $|Q| = 28$ sec, one obtains MAP values of $\langle\overline{\psi}\rangle \approx 0.99$, regardless of the feature resolution. Short queries, however, can not accurately capture the characteristics of a piece, leading to significantly lower MAP values. Reducing the feature resolution, one observes lower MAP values, too, in particular in combination with short queries. For example, using $|Q| = 4$ sec, one obtains $\langle\overline{\psi}\rangle \approx 0.94$ for $\texttt{CENS}(5, 1)$ (10 Hz resolution) and $\langle\overline{\psi}\rangle \approx 0.83$ for $\texttt{CENS}(81, 20)$ (0.5 Hz resolution). Increasing the query length, however, this effect vanishes. In particular for $|Q| \geq 20$ sec one obtains similar MAP values, independent of the feature resolution. Using $d = 10$ (1 Hz) as in $\texttt{CENS}(41, 10)$ with $|Q| = 20$ sec constitutes a good trade-off between query dimensionality and query characteristic. This setting results in shingles with a dimensionality of 240.

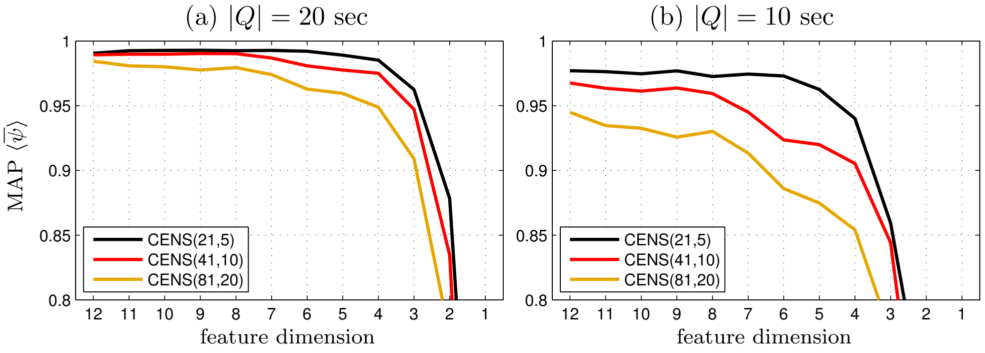
10.3.3 Evaluation of Matching Strategies

In this experiment, we investigate how much of retrieval accuracy is lost when using the query scaling approach for handling tempo differences instead of the idealized DTW-based technique. Figure 10.2a shows the retrieval quality using $\texttt{CENS}(41, 10)$ for different query scaling settings. Here, we use R variants of the query with scaling factors specified by the set T. $R = 1$ means that only the original query is used. Furthermore, we use $R = 3$ with $T = \{0.8, 1, 1.25\}$, meaning that the query is also stretched by a factor of 0.8 and 1.25 (thus simulating tempo changes of roughly $\pm 25\%$). Similarily, we use $R = 5$ with $T = \{0.66, 0.8, 1, 1.25, 1.5\}$ and $R = 9$ with $T = \{0.66, 0.73, 0.8, 0.9, 1, 1.1, 1.25, 1.35, 1.5\}$. The red line indicates the DTW-based result as shown in Figure 10.1. From these results, we draw two conclusions. Firstly, the scaling strategy ($R > 1$) significantly increases the retrieval quality in comparison to only using the original query ($R = 1$). The actual



(a) CENS$(41,10)$ (b) CENS$(5,1)$

Figure 10.2: MAP values obtained for four query scaling strategies and the DTW-based strategy using **(a)** CENS$(41,10)$ and **(b)** CENS$(5,1)$.



(a) $|Q| = 20$ sec (b) $|Q| = 10$ sec

Figure 10.3: MAP values as a function of feature dimension obtained by PCA-based dimension reduction of CENS(w,d).

choice of parameters does not seem to be crucial. In the case of our dataset, already a small number of additional queries ($R = 3$) seems to be sufficient. Secondly, the scaling strategy leads to very similar results as the computationally expensive DTW-based strategy, in particular when using a large smoothing window (e.g., $w = 41$ in CENS$(41,10)$). In the case of the smaller smoothing window $w = 5$ in CENS$(5,1)$ (see Figure 10.2b), the difference is more significant. In summary, a local feature smoothing in combination with a global scaling strategy yields a robust yet computational simple alternative to warping procedures.

10.3.4 Evaluation of Dimensionality Reduction

In a third experiment, we investigate in how far statistical data reduction based on Principal Component Analysis (PCA) can be applied to CENS features to further reduce the dimensionality of the query.

PCA estimates the principal components, i.e., the directions with maximum variance in the feature space and facilitates a projection of the original data points onto a new coordinate system spanned by a subset of these principal components. Ordering the components

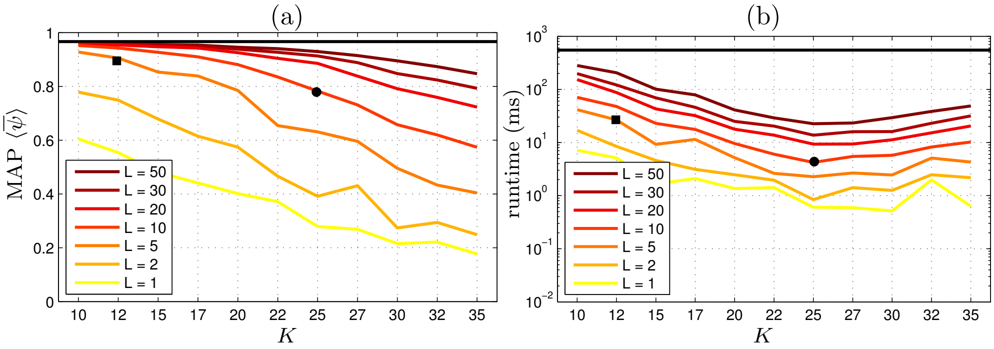


Figure 10.4: Illustration of the MAP values and runtimes obtained using `CENS(41, 10)-6` with different parameter settings in the LSH-based retrieval experiment. **(a)** Retrieval quality MAP. **(b)** Overall runtime per query including index lookup time and document ranking time. Horizontal black lines indicate values obtained by the exhaustive search.

in the order of decreasing variance guarantees an optimal representation of the data in a feature space with reduced dimensionality. We estimate the principal components using all non-query documents of our dataset and project all feature sequences onto the most dominating components. Figure 10.3 shows MAP values obtained for PCA-reduced variants of CENS features with 1-12 remaining dimensions. For a query length $|Q| = 20$ sec (Figure 10.3a), MAP values are nearly unaffected when reducing the number of dimensions from 12 to 4, in particular for higher feature resolutions. However, in combination with shorter queries of $|Q| = 10$ (Figure 10.3b), the retrieval quality is more affected by a dimensionality reduction.

In the following, we use the first 6 components of `CENS`(41, 10) features, denoted as `CENS(41, 10)-6`.[3] Using $|Q| = 20$, this results in 120-dimensional shingles, which constitutes a reasonable trade-off between shingles dimensionality and shingle characteristic.

10.3.5 Indexed-Based Retrieval by Locality Sensitive Hashing

We now investigate whether it is possible to index shingles of this size using locality sensitive hashing (LSH) for accelerating the retrieval. LSH is a hash-based approach for finding approximate nearest neighbors based on the principal that similar shingles are indexed with the same hash value. In our experiment, we use an implementation of the Exact Euclidean LSH (E^{2}LSH) algorithm [28]. We index all shingles of the entire dataset $\mathcal{D}$ using L parallel indices and K hash functions. For a query shingle Q we retrieve all shingles from the index with the same hash value as the query. Given this (typically small) set of candidate shingles, we derive the ranked list of documents and compute MAP values as described in Section 10.1.

Figure 10.4 shows MAP values (Figure 10.4a) and runtime per query in milliseconds[4]

[3] Further experiments revealed that `CENS(41, 10)-6` is very similar to the musically motivated 6-dimensional *tonal centroid* proposed in [84]. This is also related to computing a Fourier transform of chroma features as proposed in [175].

[4] obtained on a Xeon X5560 CPU with 72GB of RAM

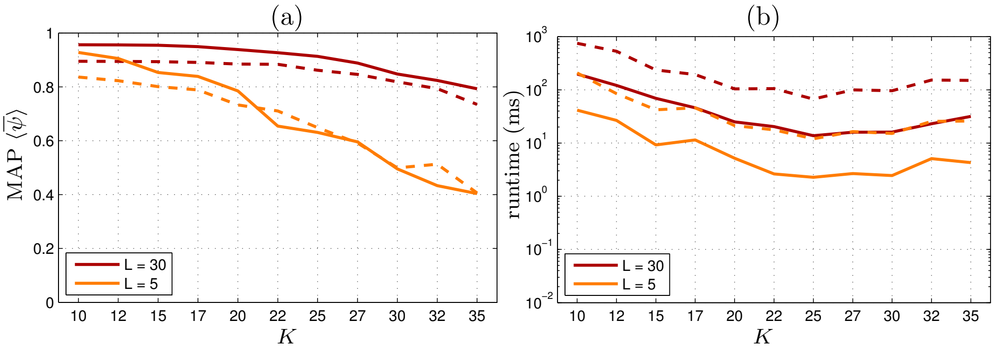


Figure 10.5: Comparison of two LSH-based retrieval strategies. Warping strategy (solid line) and query scaling strategy ($R = 5$, $T = \{0.66, 0.8, 1, 1.25, 1.5\}$) (dashed line) using `CENS(41, 10)-6`. **(a)** MAP values. **(b)** Overall runtime per query.

(Figure 10.4b) as a function of K for different L.[5] These are crucial parameters having a tremendous influence on the trade-off between retrieval quality and runtime. For example, setting $K = 12$ and $L = 5$ results in a MAP $\langle\overline{\psi}\rangle \approx 0.90$, see black square in Figure 10.4a. This is only slightly lower than the MAP value one obtains for the exhaustive search (horizontal black line). However, the runtime for this setting is significantly (by a factor of 25) faster than for the exhaustive search, see black square in Figure 10.4b. K and L allow for controlling the trade-off between speed and quality of the results. Setting $K = 25$ and $L = 10$, the MAP drops to $\langle\overline{\psi}\rangle \approx 0.80$ (black circle). However, this goes along with a decrease of query runtime to 5 ms, a speed-up of 100 in comparison to the exhaustive search.

The results shown in Figure 10.4 are again obtained using the ideal DTW-based procedure for handling tempo differences. Figure 10.5 now shows the comparison of the warping (solid line) with the query scaling approach (dashed line) for $L = 5$ and $L = 30$. Similar as for the exhaustive search discussed in Section 10.3.3, using $R = 5$, one observes only a small drop in retrieval quality (see Figure 10.4a). Using this strategy, the runtime per query linearly increases with the number of scaled queries R (see Figure 10.4b).

10.4 Conclusion

Concluding the experiments, one can say that even when using large shingles (covering roughly 20 seconds of audio material), LSH-based indexing techniques can be applied for obtaining a significant speed-up of the retrieval process (up to factor of 100). At the same time, most of the accuracy of an exhaustive search can be retained. To facilitate this, we determined suitable parameter settings with regard to query length, feature resolution and smoothing, as well as shingle dimension. The advantage of using shingles that represent a large audio fragment is that most versions of a given piece can be characterized and retrieved by using a single shingle. A combination of local feature smoothing and global query scaling is used to avoid any kind of complex warping operation. In future work,

[5] The quantization parameter denoted r in [28] is found as proposed in [18].

this can be exploited to significantly reduce the number of hash-table lookups needed for performing cross-version retrieval. The number of lookups becomes a crucial bottleneck when the index structure is stored on secondary storage devices, which is unavoidable when dealing with collections of millions of songs. Furthermore, using different hash functions may lead to improvements of retrieval quality and run-time [140]. In particular, spherical hash functions as proposed in [168] may be well suited for the characteristics of chroma shingles.

Chapter 11

Conclusion of the Thesis

In this thesis, we discussed various applications of signal processing methods to music. In particular, we focused on three central tasks in the field of music information retrieval: beat tracking and tempo estimation, music segmentation, and content-based retrieval. For all three tasks, we exploited musical knowledge about the signals' properties to derive feature representations that show a significant degree of robustness against musical variations but still exhibit a high musical expressiveness.

In Part I of the thesis, we dealt with the extraction of local tempo and beat information. Opposed to previous approaches that assume constant tempo throughout a recording, our analysis particularly focused on music with temporal variations. As one major contribution, we introduced a novel concept for deriving musically meaningful local pulse information from possibly noisy onset information. Exploiting the local quasi-periodicity of music signals, the main benefit of our PLP mid-level representation is that it can locally adjust to changes in tempo. Even for classical music with soft note onsets, we were able to extract meaningful local tempo and beat information. However, for highly expressive interpretations of romantic music, the assumption of local quasi-periodicity is often violated leading to poor results. In such cases, our PLP concept at least yields a confidence measure to reveal problematic passages.

The understanding of physical and musical properties that make beat tracking difficult is of essential importance for improving the performance of automated approaches. As second contribution of Part I, we introduced a novel evaluation framework for beat tracking algorithms where multiple performances of a piece of music are considered simultaneously. This approach yields a better understanding not only of the algorithms' behavior but also of the underlying music material. As third contribution of Part I, we introduced various tempogram representations that reveal local tempo and rhythm information while being robust to extraction errors. Furthermore, exploiting musical knowledge about the different pulse levels, we introduced a class of robust mid-level features that reveal local tempo information while being invariant to pulse level confusions. Being the tempo-based counterpart of the harmony-based chromagrams, the cyclic tempograms are suitable for music analysis and retrieval tasks where harmonic or timbral properties are not relevant.

In Part II of this thesis, we introduced various signal processing methods with the goal to make folk song field recordings more easily accessible for research and analysis purposes. As folk songs are part of oral culture, it seems plausible that by looking at the original audio recordings one may derive new insights that can not be derived by simply looking at the transcribed melodies. As one main contribution of Part II, we presented two procedures for segmenting a given folk song recording into its individual stanzas by exploiting knowledge about the strophic structure of folk songs. In particular, we introduced a combination of various enhancement strategies to account for the intonation fluctuations, temporal variations, and poor recording conditions. Our experiments indicate that robust segmentation results can be obtained even in the presence of strong temporal and spectral variations without relying on any reference transcription. Limitations of our segmentation procedures remain in the case of structural differences across the stanzas of a song. Increasing the robustness of the segmentation procedure to handle such variations remains a challenging future problem.

Furthermore, we presented a multimodal approach for extracting performance parameters by comparing the audio material with a suitable reference transcription. As main contribution, we introduced the concept of chroma templates that reveal the consistent and inconsistent melodic aspects across the various stanzas of a given recording. In computing these templates, we used tuning and time warping strategies to deal with local variations in melody, tuning, and tempo.

The segmentation and analysis techniques introduced in Part II of the thesis constitute only a first step towards making field recordings more accessible to performance analysis and folk song research. Only by using automated methods, one can deal with vast amounts of audio material, which would be infeasible otherwise. Our techniques can be considered as a kind of preprocessing to automatically screen a large number of field recordings with the goal to detect and locate interesting and surprising passages worth being examined in more detail by domain experts. This may open up new challenging and interdisciplinary research directions not only for folk song research but also for music information retrieval and music cognition [178].

In Part III of this thesis, we discussed various content-based retrieval strategies based on the query-by-example paradigm. Such strategies can be loosely classified according to their specificity, which refers to the considered degree of similarity between the query and the database documents. As one contribution of Part III, a second classification principle based on granularity was introduced. The resulting specificity/granularity scheme gives a compact overview of the various retrieval paradigms while illustrating their subtle but crucial differences. The specificity has a significant impact on the efficiency of the retrieval system. Search tasks of high specificity typically can be realized efficiently using indexing techniques. In contrast, search tasks of low specificity need more flexible and cost-intensive mechanisms for dealing with musical variations.

As further contribution of Part III, we presented two investigations with the goal to scale low specificity cross-version retrieval to large datasets. Firstly, we studied the robustness and discriminative power of modified audio fingerprints. As our experiments reveal, modified peak fingerprints based on musically motivated time-pitch or time-chroma representations can handle a high degree of spectral variations and allow for an identification of different performances of the same piece of music. However, the issue on how the temporal variations between performances can be considered in this approach is still unclear and should be subject to future research. Our second approach to efficient cross-version retrieval is based on audio shingling, where each query is represented by a single shingle that covers a long segment of the audio recording. In this approach, a combination of strategies is used to derive compact yet highly characteristic and robust audio shingles. Robustness to spectral variations is obtained using suitable variants of chroma features, whereas temporal variations are handled by using a combination of local feature smoothing and global query scaling strategies. Using the resulting low-dimensional shingles, LSH-based indexing techniques can be applied for significantly speeding up the retrieval process.

Aside from efficiency and scalability issues, another major challenge in content-based music retrieval refers to cross-modal retrieval scenarios, where the query as well as the retrieved documents can be of different modalities. For example, one might use a small fragment of a musical score to query an audio database for recordings that are related to this fragment. Or a short audio fragment might be used to query a database containing MIDI files. In the future, comprehensive retrieval frameworks need to be developed that offer multi-faceted search functionalities in heterogeneous and distributed music collections containing all sorts of music-related documents.

All feature representations presented in this thesis show a significant degree of robustness against musical variations while still exhibiting a high musical expressiveness. The increased robustness is achieved by exploiting model assumptions about the analyzed music signals. These model assumptions, however, go along with a reduced generalizability. For example, in our PLP concept, we assume local quasi-periodicity, which allows us to obtain meaningful results even in the presence of weak note onsets and continuous tempo changes. In the case of local tempo distortions as found in the Chopin Mazurkas, however, this assumption is violated and the limits of our approach are reached. For such kind of signals, a different approach (e. g., based on an explicit determination of note onset positions and an evaluation of inter-onset-intervals) may lead to better results [40]. Similarly, in our folk song analysis, we assume a strophic form and obtain robust segmentation results even in the presence of significant spectral and temporal variations. The limits of this repetition-based approach, however, are reached when structural variations within the stanzas occur, i. e., when the assumption of a strophic form is violated. In the case of such variations, novelty-based approaches for detecting segment boundaries may be less fragile [143].

As these examples show, one grand challenge for music signal processing is related to the question on how the developed techniques and methods can be made more general and applicable to cover a wide range of music signals. In the years of MIR research, solutions have been presented that can cope with isolated facets of music signals in restricted and well-defined scenarios. In future research, more efforts need to be put into developing approaches that are capable of dealing with and adopting to arbitrary music signals to better reflect the various facets of music.

Appendix A

Tempogram Toolbox

The tempo and pulse-related audio features described in Part I of this thesis are released as MATLAB implementations in a *Tempogram Toolbox*, which is provided under a GNU-GPL license at www.mpi-inf.mpg.de/resources/MIR/tempogramtoolbox. The functionality provided by our tempogram toolbox is illustrated in Figure A.1, where an audio recording of Claude Debussy's Sonata for Violin and Piano in G minor (L 140) serves as an example. The audio recording is available from Saarland Music Data (SMD) http://www.mpi-inf.mpg.de/resources/SMD/. Analyzing this recording with respect to tempo is challenging as it contains weak note onsets played by a violin as well as a number of prominent tempo changes.

Given an audio recording (Figure A.1a), we first derive a novelty curve as described in Section 2.3 (Figure A.1b). Given such a (possibly very noisy) onset representation the toolbox allows for deriving a predominant local pulse (PLP) curve as introduced in Section 2.6 (Figure A.1c). As second main part, our tempogram toolbox facilitates various tempogram representations as introduced in Chapter 4 that reveal local tempo characteristics even for expressive music exhibiting tempo-changes. To obtain such a representation, the novelty curve is analyzed with respect to local periodic patterns. Here, the toolbox provides Fourier-based methods (Figure A.1d,f) as well as autocorrelation-based methods (Figure A.1e,g), see Section 4.1. For both concepts, representations as time/tempo (Figure A.1d,e) as well as time/time-lag tempogram (Figure A.1f,g) are available. Furthermore, resampling and interpolation functions allow for switching between tempo and time-lag axes as desired. The third main part of our toolbox provides functionality for deriving cyclic tempograms from the tempogram representations as introduced in Section 4.2. The cyclic tempo features constitute a robust mid-level representation revealing local tempo characteristics of music signals while being invariant to changes in the pulse level (Figure A.1h,i). Being the tempo-based counterpart of the chromagrams, cyclic tempograms are suitable for music analysis and retrieval tasks. Finally, the tempogram toolbox contains a variety of functions for the visualization and sonification of extracted tempo and pulse information.

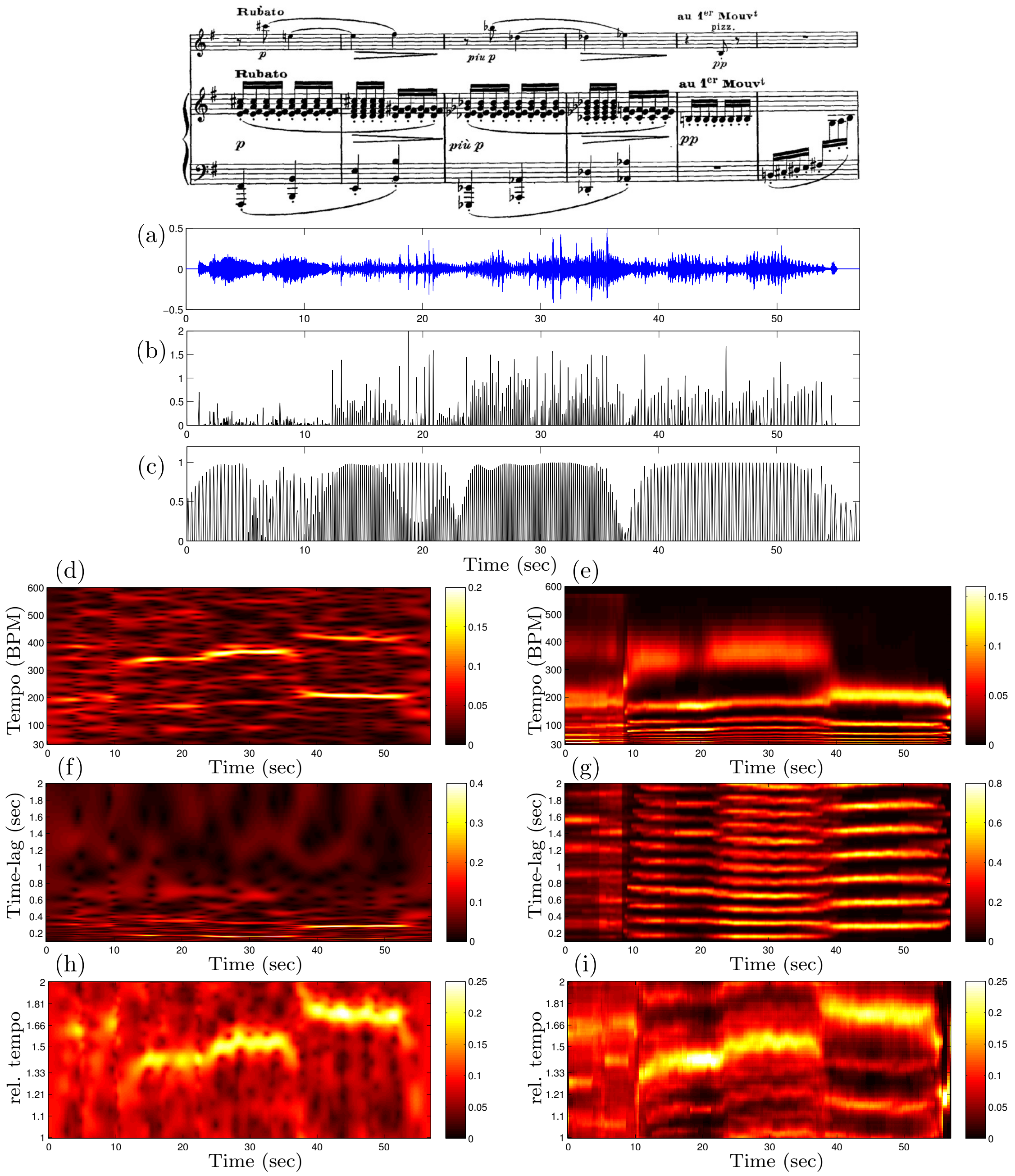


Figure A.1: Illustration of the functionality of the tempogram toolbox using an excerpt (second movement, bars 79 – 107) of an audio recording of Claude Debussy's Sonata for Violin and Piano in G minor (L 140). **(a)** Waveform of the excerpt. **(b)** Novelty curve extracted from the audio recording indicating note onset candidates. **(c)** PLP curve revealing the predominant local pulse. **(d)** Fourier-based tempogram. **(e)** Autocorrelation-based tempogram. **(f)** Fourier-based tempogram with time-lag axis. **(g)** Autocorrelation-based tempogram with time-lag axis. **(h)** Fourier-based cyclic tempogram. **(i)** Autocorrelation-based cyclic tempogram.

Bibliography

[1] Eric Allamanche, Jürgen Herre, Oliver Hellmuth, Bernhard Fröba, and Markus Cremer. AudioID: Towards content-based identification of audio material. In *Proc. 110th AES Convention*, Amsterdam, NL, 2001.

[2] Miguel Alonso, Bertrand David, and Gaël Richard. Tempo and beat estimation of musical signals. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004.

[3] Miguel Alonso, Gaël Richard, and Bertrand David. Accurate tempo estimation based on harmonic+noise decomposition. *EURASIP Journal on Advances in Signal Processing*, 2007:Article ID 82795, 14 pages, 2007.

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

[5] Juan Pablo Bello. Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 239–244, Vienna, Austria, 2007.

[6] Juan Pablo Bello. Measuring structural similarity in music. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2013–2025, 2011.

[7] 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.

[8] Juan Pablo Bello, Chris Duxbury, Mike Davies, and Mark Sandler. On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, 11(6):553–556, 2004.

[9] Thierry Bertin-Mahieux, Douglas Eck, Francois Maillet, and Paul Lamere. Autotagger: A model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research*, 37(2):115–135, 2008.

[10] Thierry Bertin-Mahieux, Douglas Eck, and Michael I. Mandel. Automatic tagging of audio: The state-of-the-art. In Wenwu Wang, editor, *Machine Audition: Principles, Algorithms and Systems*, chapter 14, pages 334–352. IGI Publishing, 2010.

[11] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 591–596, Miami, USA, 2011.

[12] Thierry Bertin-Mahieux and Daniel P.W. Ellis. Large-scale cover song recognition using hashed chroma landmarks. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 117–120, New Platz, NY, 2011.

[13] Jeff A. Bilmes. Techniques to foster drum machine expressivity. In *International Computer Music Conference*, Tokyo, Japan, 1993.

[14] Dmitry Bogdanov, Joan Serrà, Nicolas Wack, and Perfecto Herrera. Unifying low-level and high-level music similarity measures. *IEEE Transactions on Multimedia*, 13(4):687 –701, 2011.

[15] Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma. A review of algorithms for audio fingerprinting. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 169–173, St. Thomas, Virgin Islands, USA, 2002.

[16] Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma. A review of audio fingerprinting. *Journal of VLSI Signal Processing Systems*, 41(3):271–284, 2005.

[17] Pedro Cano, Eloi Batlle, Harald Mayer, and Helmut Neuschmied. Robust sound modeling for song detection in broadcast audio. In *Proceedings of the 112th AES Convention*, pages 1–7, 2002.

[18] Michael A. Casey, Christophe Rhodes, and Malcolm Slaney. Analysis of minimum distances in high-dimensional musical spaces. *IEEE Transactions on Audio, Speech & Language Processing*, 16(5), 2008.

[19] Michael A. Casey, Remco Veltkap, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[20] Òscar Celma. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, 1st edition, 2010.

[21] Ali Taylan Cemgil, Bert Kappen, Peter Desain, and Henkjan Honing. On tempo tracking: Tempogram representation and kalman filtering. *Journal of New Music Research*, 28(4):259–273, 2001.

[22] Wei Chai and Barry Vercoe. Music thumbnailing via structural analysis. In *Proceedings of the ACM International Conference on Multimedia*, pages 223–226, Berkeley, CA, USA, 2003.

[23] Nick Collins. A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions. In *AES Convention 118*, Barcelona, Spain, 2005.

[24] Nick Collins. Using a pitch detector for onset detection. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 100–106, London, UK, 2005.

[25] Olmo Cornelis, Micheline Lesaffre, Dirk Moelants, and Marc Leman. Access to ethnic music: advances and perspectives in content-based music information retrieval. *Signal Processing*, In Press, 2009.

[26] 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, 2008.

[27] Roger B. Dannenberg. Toward automated holistic beat tracking, music analysis and understanding. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 366–373, London, UK, 2005.

[28] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the 20th ACM Symposium on Computational Geometry*, pages 253–262, Brooklyn, New York, USA, 2004.

[29] Matthew E.P. Davies, Norberto Degara, and Mark D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Technical Report C4DM-TR-09-06, Queen Mary University, Centre for Digital Music, 2009.

[30] Matthew E.P. Davies, Norberto Degara, and Mark D. Plumbley. Measuring the performance of beat tracking algorithms using a beat error histogram. *IEEE Signal Processing Letters*, 18(3):157–160, 2011.

[31] Matthew E.P. Davies and Mark D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1009–1020, 2007.

[32] Alain de Cheveigné and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America (JASA)*, 111(4):1917–1930, 2002.

[33] Noberto Degara, Matthew E.P. Davies, Antonio Pena, and Mark D. Plumbley. Onset event decoding exploiting the rhythmic structure of polyphonic music. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1228–1239, 2011.

[34] Norberto Degara, Antonio Pena, Matthew E.P. Davies, and Mark D. Plumbley. Note onset detection using rhythmic structure. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5526–5529, Dallas, TX, USA, 2010.

[35] Norberto Degara, Antonio Pena, and Soledad Torres-Guijarro. A comparison of score-level fusion rules for onset detection in music signals. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 117–122, Kobe, Japan, 2009.

[36] Norberto Degara, Enrique Argones Rúa, Antonio Pena, Soledad Torres-Guijarro, Matthew E. P. Davies, and Mark D. Plumbley. Reliability-informed beat tracking of musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):290–301, 2012.

[37] Simon Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58, 2001.

[38] Simon Dixon. An empirical comparison of tempo trackers. In *Proceedings of the Brazilian Symposium on Computer Music (SBCM)*, pages 832–840, Fortaleza, CE, 2001.

[39] Simon Dixon. Onset detection revisited. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 133–137, Montreal, Quebec, Canada, 18–20 2006.

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

[41] Simon Dixon and Werner Goebl. Pinpointing the beat: Tapping to expressive performances. In *Proceedings of the International Conference on Music Perception and Cognition (ICMPC)*, pages 617–620, Sydney, Australia, 2002.

[42] Simon Dixon, Werner Goebl, and Emilios Cambouropoulos. Perceptual smoothness of tempo in expressively performed music. *Music Perception*, 23(3):195–214, 2006.

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

[44] Daniel P.W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.

[45] Daniel P.W. Ellis, Courtenay V. Cotton, and Michael I. Mandel. Cross-correlation of beat-synchronous representations for music similarity. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 57–60, Taipei, Taiwan, 2008.

[46] 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, 2007.

[47] Antti J. Eronen and Anssi P. Klapuri. Music tempo estimation with k-NN regression. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(1):50–57, 2010.

[48] Sebastian Ewert and Meinard Müller. Using score-informed constraints for NMF-based source separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 129–132, Kyoto, Japan, 2012.

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

[50] Florian Eyben, Sebastian Böck, Björn Schuller, and Alex Graves. Universal onset detection with bidirectional long short-term memory neural networks. In *Proceedings of the International Society for Music Information Retrieval Conference ISMIR*, pages 589–594, Utrecht, The Netherlands, 2010.

[51] Hugo Fastl and Eberhard Zwicker. *Psychoacoustics, Facts and Models.* Springer, 2007 (3rd Edition).

[52] Sébastien Fenet, Gaël Richard, and Yves Grenier. A scalable audio fingerprint method with robustness to pitch-shifting. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 121–126, Miami, FL, USA, 2011.

[53] Jonathan Foote. Visualizing music and audio using self-similarity. In *Proceedings of the ACM International Conference on Multimedia*, pages 77–80, Orlando, FL, USA, 1999.

[54] Jonathan Foote and Shingo Uchihashi. The beat spectrum: A new approach to rhythm analysis. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*, Los Alamitos, CA, USA, 2001.

[55] Rémi Foucard, Jean-Louis Durrieu, Mathieu Lagrange, , and Gaël Richard. Multimodal similarity between musical streams for cover version detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, USA, 2010.

[56] Dimitrios Fragoulis, George Rousopoulos, Thanasis Panagopoulos, Constantin Alexiou, and Constantin Papaodysseus. On the automated recognition of seriously distorted musical recordings. *IEEE Transactions on Signal Processing*, 49(4):898–908, 2001.

[57] 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, 2007.

[58] 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.

[59] Emilia Gómez. *Tonal Description of Music Audio Signals.* PhD thesis, UPF Barcelona, 2006.

[60] Emilia Gómez, Bee Suan Ong, and Perfecto Herrera. Automatic tonal analysis from music summaries for version identification. In *Proceedings of the 121st AES Convention*, San Francisco, CA, USA, 2006.

[61] Masataka Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, 2001.

[62] 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.

[63] 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.

[64] 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.

[65] Fabien Gouyon. *A computational approach to rhythm description: audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing.* PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2005.

[66] Fabien Gouyon and Simon Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29:34–54, 2005.

[67] Fabien Gouyon, Simon Dixon, and Gerhard Widmer. Evaluating low-level features for beat classification and tracking. In *Proceedings of the 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, 2007.

[68] Fabien Gouyon and Perfecto Herrera. Pulse-dependent analysis of percussive music. In *Proceedings of the AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, Espoo, Finland, 2002.

[69] Fabien Gouyon, Anssi P. Klapuri, Simon Dixon, Miguel Alonso, George Tzanetakis, Christian Uhle, and Pedro Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.

[70] Louis Peter Grijp and Herman Roodenburg. *Blues en Balladen. Alan Lomax en Ate Doornbosch, twee muzikale veldwerkers.* AUP, Amsterdam, 2005.

[71] Peter Grosche and Meinard Müller. Computing predominant local periodicity information in music recordings. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 33–36, New Paltz, NY, USA, 2009.

[72] Peter Grosche and Meinard Müller. A mid-level representation for capturing dominant tempo and pulse information in music recordings. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 189–194, Kobe, Japan, 2009.

[73] Peter Grosche and Meinard Müller. Extracting predominant local pulse information from music recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1688–1701, 2011.

[74] Peter Grosche and Meinard Müller. Tempogram toolbox: Matlab implementations for tempo and pulse analysis of music recordings. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, Miami, FL, USA, 2011, late-breaking contribution.

[75] Peter Grosche and Meinard Müller. Toward characteristic audio shingles for efficient cross-version music retrieval. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 473–476, Kyoto, Japan, 2012.

[76] Peter Grosche and Meinard Müller. Toward musically-motivated audio fingerprints. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 93–96, Kyoto, Japan, 2012.

[77] Peter Grosche, Meinard Müller, and Frank Kurth. Cyclic tempogram – a mid-level tempo representation for music signals. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5522–5525, Dallas, Texas, USA, 2010.

[78] Peter Grosche, Meinard Müller, and Frank Kurth. Tempobasierte Segmentierung von Musikaufnahmen. In *Proceedings of the 36th Deutsche Jahrestagung für Akustik (DAGA)*, Berlin, Germany, 2010.

[79] 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.

[80] Peter Grosche, Meinard Müller, and Joan Serrà. Audio content-based music retrieval. In Meinard Müller, Masataka Goto, and Markus Schedl, editors, *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups*, pages 157–174. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2012.

[81] Peter Grosche, Björn Schuller, Meinard Müller, and Gerhard Rigoll. Automatic transcription of recorded music. *Acta Acustica united with Acustica*, 98(2):199–215, 2012.

[82] Jaap Haitsma and Ton Kalker. A highly robust audio fingerprinting system. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 107–115, Paris, France, 2002.

[83] Jaap Haitsma and Ton Kalker. Speed-change resistant audio fingerprinting using autocorrelation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 728–731, 2003.

[84] Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. In *Proceedings of the ACM Workshop on Audio and Music Computing Multimedia*, pages 21–26, Santa Barbara, California, USA, 2006.

[85] André Holzapfel, Matthew E.P. Davies, Jose R. Zapata, João L. Oliveira, and Fabien Gouyon. On the automatic identification of difficult examples for beat tracking: towards building new evaluation datasets. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 89–92, Kyoto, Japan, 2012.

[86] André Holzapfel and Yannis Stylianou. Beat tracking using group delay based onset detection. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2008.

[87] André Holzapfel and Yannis Stylianou. Rhythmic similarity in traditional Turkish music. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 99–104, Kobe, Japan, 2009.

[88] André Holzapfel, Yannis Stylianou, Ali C. Gedik, and Bariş Bozkurt. Three dimensions of pitched instrument onset detection. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1517–1527, 2010.

[89] André Holzapfel, Gino Angelo Velasco, Nicki Holighaus, Monika Dörfler, and Arthur Flexer. Advantages of nonstationary Gabor transforms in beat tracking. In *Proceedings of the 1st international ACM workshop on Music Information Retrieval with User-centered and Multimodal Strategies (MIRUM)*, pages 45–50, Scottsdale, Arizona, USA, 2011.

[90] 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, USA, 2003.

[91] Kristoffer Jensen. Multiple scale music segmentation using rhythm, timbre, and harmony. *EURASIP Journal on Advances in Signal Processing*, 2007(1):11 pages, 2007.

[92] Kristoffer Jensen, Jieping Xu, and Martin Zachariasen. Rhythm-based segmentation of popular chinese music. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, UK, 2005.

[93] Nanzhu Jiang, Peter Grosche, Verena Konz, and Meinard Müller. Analyzing chroma feature types for automated chord recognition. In *Proceedings of the 42nd AES Conference on Semantic Audio*, Ilmenau, Germany, 2011.

[94] Zoltán Juhász. Motive identification in 22 folksong corpora using dynamic time warping and self organizing maps. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 171–176, Kobe, Japan, 2009.

[95] Zoltán Juhász. A systematic comparison of different European folk music traditions using self-organizing maps. *Journal of New Music Research*, 35:95–112(18), June 2006.

[96] Yan Ke, Derek Hoiem, and Rahul Sukthankar. Computer vision for music identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 597–604, San Diego, CA, USA, 2005.

[97] Hyoung-Gook Kim, Nicolas Moreau, and Thomas Sikora. *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval*, page 304. John Wiley & Sons, 2005. ISBN: 978-0-470-09334-4.

[98] Youngmoo E. Kim, Erik M. Schmidt, Raymond Migneco, Brandon C. Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A. Speck, and Douglas Turnbull. Music emotion recognition: A state of the art review. In *11th Intl. Society for Music Information Retrieval Conference (ISMIR)*, pages 255–266, Utrecht, The Netherlands, 2010.

[99] Aniket Kittur, Ed Chi, Bryan A. Pendleton, Bongwon Suh, and Todd Mytkowicz. Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. In *Computer/Human Interaction Conference (Alt.CHI)*, San Jose, CA, 2007.

[100] Anssi P. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3089–3092, Washington, DC, USA, 1999.

[101] Anssi P. Klapuri and Manuel Davy, editors. *Signal Processing Methods for Music Transcription.* Springer, New York, 2006.

[102] 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.

[103] Carol L. Krumhansl. *Cognitive foundations of musical pitch.* Oxford University Press, 1990.

[104] Frank Kurth, Thorsten Gehrmann, and Meinard Müller. The cyclic beat spectrum: Tempo-related audio features for time-scale invariant audio identification. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 35–40, Victoria, Canada, 2006.

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

[106] Frank Kurth, Meinard Müller, David Damm, Christian Fremerey, Andreas Ribbrock, and Michael Clausen. SyncPlayer - an advanced system for multimodal music access. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 381–388, London, UK, 2005.

[107] Frank Kurth, Andreas Ribbrock, and Michael Clausen. Identification of highly distorted audio material for querying large scale data bases. In *Proceedings of the 112th AES Convention*, 2002.

[108] Alexandre Lacoste and Douglas Eck. A supervised classification algorithm for note onset detection. *EURASIP Journal on Applied Signal Processing*, 2007:153165, 2007.

[109] Mathieu Lagrange and Joan Serrà. Unsupervised accuracy improvement for cover song detection using spectral connectivity network. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 595–600, Utrecht, The Netherlands, 2010.

[110] Paul Lamere. Social tagging and music information retrieval. *Journal of New Music Research*, 37(2):101–114, 2008.

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

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

[113] Pierre Leveau, Laurent Daudet, and Gaël Richard. Methodology and tools for the evaluation of automatic onset detection algorithms in music. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 72–77, Barcelona, Spain, 2004.

[114] Mark Levy and Mark Sandler. Structural segmentation of musical audio by constrained clustering. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):318–326, 2008.

[115] Mark Levy, Mark Sandler, and Michael A. Casey. Extraction of high-level musical structure from audio data and its application to thumbnail generation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 13–16, Toulouse, France, 2006.

[116] Hanna Lukashevich. Towards quantitative measures of evaluating song segmentation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 375–380, Philadelphia, USA, 2008.

[117] Matija Marolt. A mid-level representation for melody-based retrieval in audio collections. *IEEE Transactions on Multimedia*, 10(8):1617–1625, 2008.

[118] Paul Masri and Andrew Bateman. Improved modeling of attack transients in music analysis-resynthesis. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 100–103, Hong Kong, 1996.

[119] Matthias Mauch, Chris Cannam, Matthew E.P. Davies, Simon Dixon, Christopher Harte, Sefki Kolozali, Dan Tidhar, and Mark Sandler. OMRAS2 metadata project 2009. In *Late Breaking Demo of the International Conference on Music Information Retrieval (ISMIR)*, Kobe, Japan, 2009.

[120] Matthias Mauch and Simon Dixon. Approximate note transcription for the improved identification of difficult chords. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 135–140, Utrecht, The Netherlands, 2010.

[121] Martin F. McKinney, Dirk Moelants, Matthew E.P. Davies, and Anssi P. Klapuri. Evaluation of audio beat tracking and music tempo extraction algorithms. *Journal of New Music Research*, 36(1):1–16, 2007.

[122] D. Moelants, O. Cornelis, and M. Leman. Exploring African tone scales. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 489–494, Kobe, Japan, 2009.

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

[124] Meinard Müller and Michael Clausen. Transposition-invariant self-similarity matrices. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 47–50, Vienna, Austria, 2007.

[125] Meinard Müller, Daniel P. W. Ellis, Anssi Klapuri, and Gaël Richard. Signal processing for music analysis. *IEEE Journal on Selected Topics in Signal Processing*, 5(6):1088–1110, 2011.

[126] 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.

[127] 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)*, pages 215–220, Miami, FL, USA, 2011.

[128] Meinard Müller and Peter Grosche. Automated segmentation of folk song field recordings. In *Proceedings of the 10th ITG Conference on Speech Communication*, Braunschweig, Germany, 2012.

[129] Meinard Müller, Peter Grosche, and Nanzhu Jiang. A segment-based fitness measure for capturing repetitive structures of music recordings. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, pages 615–620, Miami, FL, USA, 2011.

[130] 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, 2009.

[131] Meinard Müller, Peter Grosche, and Frans Wiering. Towards automated processing of folk song recordings. In Eleanor Selfridge-Field, Frans Wiering, and Geraint A. Wiggins, editors, *Knowledge representation for intelligent music processing*, number 09051 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[132] Meinard Müller, Peter Grosche, and Frans Wiering. Automated analysis of performance variations in folk song recordings. In *Proceedings of the International Conference on Multimedia Information Retrieval (MIR)*, pages 247–256, Philadelphia, PA, USA, 2010.

[133] 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 International Society for Music Information Retrieval Conference (ISMIR)*, pages 69–74, Kobe, Japan, 2009.

[134] Meinard Müller and Frank Kurth. Enhancing similarity matrices for music audio analysis. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 437–440, Toulouse, France, 2006.

[135] 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.

[136] Meinard Müller and Tido Röder. Motion templates for automatic classification and retrieval of motion capture data. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pages 137–146, Vienna, Austria, 2006.

[137] Hélène Papadopoulos and Geoffroy Peeters. Simultaneous estimation of chord progression and downbeats from an audio file. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–124, 2008.

[138] 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.

[139] Richard Parncutt. A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception*, 11:409–464, 1994.

[140] Loïc Paulevé, Hervé Jégou, and Laurent Amsaleg. Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern Recognition Letters*, 31(11):1348–1358, 2010.

[141] Jouni Paulus and Anssi P. Klapuri. Measuring the similarity of rhythmic patterns. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 150–156, Paris, France, 2002.

[142] Jouni Paulus and Anssi P. Klapuri. Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1159–1170, 2009.

[143] Jouni Paulus, Meinard Müller, and Anssi P. 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.

[144] Geoffrey Peeters. Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 35–40, Vienna, Austria, 2007.

[145] Geoffroy Peeters. Time variable tempo detection and beat marking. In *Proceedings of the International Computer Music Conference (ICMC)*, Barcelona, Spain, 2005.

[146] Geoffroy Peeters. Template-based estimation of time-varying tempo. *EURASIP Journal on Advances in Signal Processing*, 2007(1):158–158, 2007.

[147] Geoffroy Peeters. "copy and scale" method for doing time-localized m.i.r. estimation:: application to beat-tracking. In *Proceedings of 3rd international workshop on Machine learning and music (MML)*, pages 1–4, Firenze, Italy, 2010.

[148] Geoffroy Peeters and Hélène Papadopoulos. Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1754–1769, 2011.

[149] Lawrence Rabiner and Bing-Hwang Juang. *Fundamentals of Speech Recognition.* Prentice Hall Signal Processing Series, 1993.

[150] Mathieu Ramona and Geoffroy Peeters. Audio identification based on spectral modeling of bark-bands energy and synchronization through onset detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 477–480, 2011.

[151] Suman Ravuri and Daniel P.W. Ellis. Cover song detection: from high scores to general classification. In *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, pages 65–68, Dallas, TX, 2010.

[152] 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.

[153] Craig Stuart Sapp. Hybrid numeric/rank similarity metrics. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 501–506, Philadelphia, USA, 2008.

[154] Eric D. Scheirer. Tempo and beat analysis of acoustical musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

[155] Christian Schörkhuber and Anssi P. Klapuri. Constant-Q transform toolbox for music processing. In *Sound and Music Computing Conference (SMC)*, Barcelona, 2010.

[156] Hendrik Schreiber, Peter Grosche, and Meinard Müller. A re-ordering strategy for accelerating index-based audio fingerprinting. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, pages 127–132, Miami, FL, USA, 2011.

[157] Björn Schuller, Florian Eyben, and Gerhard Rigoll. Tango or waltz?: Putting ballroom dance style into tempo detection. *EURASIP Journal on Audio, Speech, and Music Processing*, 2008:12, 2008.

[158] Eleanor Selfridge-Field, editor. *Beyond MIDI: the handbook of musical codes.* MIT Press, Cambridge, MA, USA, 1997.

[159] Jarno Seppänen. Tatum grid analysis of musical signals. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 131–134, 2001.

[160] Jarno Seppänen, Antti J. Eronen, and Jarmo Hiipakka. Joint beat & tatum tracking from music signals. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Victoria, Canada, 2006.

[161] Joan Serrà, Emilia Gómez, and Perfecto Herrera. Audio cover song identification and similarity: background, approaches, evaluation and beyond. In Z. W. Ras and A. A. Wieczorkowska, editors, *Advances in Music Information Retrieval*, volume 16 of *Studies in Computational Intelligence*, chapter 14, pages 307–332. Springer, Berlin, Germany, 2010.

[162] Joan Serrà, Emilia Gómez, Perfecto Herrera, and Xavier Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech and Language Processing*, 16:1138–1151, 2008.

[163] Joan Serrà, Meinard Müller, Peter Grosche, and Josep Lluis Arcos. Unsupervised detection of music boundaries by time series structure features. In *Proceedings of the AAAI International Conference on Artificial Intelligence*, Toronto, Ontario, Canada, 2012.

[164] Joan Serrà, Xavier Serra, and Ralph G. Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(9):093017, 2009.

[165] Joan Serrà, Massimiliano Zanin, Perfecto Herrera, and Xavier Serra. Characterization and exploitation of community structure in cover song networks. *Pattern Recognition Letters*, 2010. Submitted.

[166] William Arthur Sethares. *Rhythm and Transforms.* Springer, 2007.

[167] Dan Stowell and Mark Plumbley. Adaptive whitening for improved real-time audio onset detection. In *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, Denmark, 2007.

[168] Kengo Terasawa and Yuzuru Tanaka. Spherical LSH for approximate nearest neighbor search on unit hypersphere. In Frank Dehne, Jörg-Rüdiger Sack, and Norbert Zeh, editors, *Algorithms and Data Structures*, volume 4619 of *Lecture Notes in Computer Science*, pages 27–38. Springer Berlin / Heidelberg, 2007.

[169] Chee Chuan Toh, Bingjun Zhang, and Ye Wang. Multiple-feature fusion based onset detection for solo singing voice. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 515–520, Philadelphia, PA, USA, 2008.

[170] Wei-Ho Tsai, Hung-Ming Yu, and Hsin-Min Wang. Using the similarity of main melodies to identify cover versions of popular songs for music document retrieval. *Journal of Information Science and Engineering*, 24(6):1669–1687, 2008.

[171] Emiru Tsunoo, Taichi Akase, Nobutaka Ono, and Shigeki Sagayama. Musical mood classification by rhythm and bass-line unit pattern analysis. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.

[172] Douglas Turnbull, Luke Barrington, and Gert Lanckriet. Five approaches to collecting tags for music. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 225–230, Philadelphia, USA, 2008.

[173] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):467–476, 2008.

[174] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[175] 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.

[176] Jan Van Balen. Automatic recognition of samples in musical audio. Master's thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2011.

[177] Peter van Kranenburg, Joerg Garbers, Anja Volk, Frans Wiering, Louis P. Grijp, and Remco C. Veltkamp. Collaboration perspectives for folk song research and music information retrieval: The indispensable role of computational musicology. *Journal of Interdisciplinary Music Studies*, 4:17–43, 2010.

[178] Peter van Kranenburg, Jörg Garbers, Anja Volk, Frans Wiering, Louis Grijp, and Remco Veltkamp. Towards integration of MIR and folk song research. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 505–508, Vienna, AT, 2007.

[179] Peter van Kranenburg, Anja Volk, Frans Wiering, and Remco C. Veltkamp. Musical models for folk-song melody alignment. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 507–512, Kobe, Japan, 2009.

[180] Anja Volk, Peter van Kranenburg, Joerg Garbers, Frans Wiering, Remco C. Veltkamp, and Louis P. Grijp. The study of melodic similarity using manual annotation and melody feature sets. Technical Report UU-CS-2008-013, Department of Information and Computing Sciences, Utrecht University, 2008.

[181] Avery Wang. An industrial strength audio search algorithm. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 7–13, Baltimore, USA, 2003.

[182] Felix Weninger, Martin Wöllmer, and Björn Schuller. Automatic assessment of singer traits in popular music: Gender, age, height and race. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 37–42, Miami, FL, USA, 2011.

[183] Kris West and Paul Lamere. A model-based approach to constructing music similarity functions. *EURASIP Journal on Advances in Signal Processing*, 2007(1):024602, 2007.

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

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

[186] Frans Wiering, Louis P. Grijp, Remco C. Veltkamp, Jörg Garbers, Anja Volk, and Peter van Kranenburg. Modelling folksong melodies. *Interdiciplinary Science Reviews*, 34(2-3):154–171, 2009.

[187] Fu-Hai Frank Wu, Tsung-Chi Lee, Jyh-Shing Roger Jang, Kaichun K. Chang, Chun-Hung Lu, and Wen-Nan Wang. A two-fold dynamic programming approach to beat tracking for audio music with time-varying tempo. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 191–196, Miami, FL, USA, 2011.

[188] Jose R. Zapata and Emilia Gómez. Comparative evaluation and combination of audio tempo estimation approaches. In *42nd AES Conference on Semantic Audio*, Ilmenau, Germany, 2011.

[189] Ruohua Zhou, Marco Mattavelli, and Giorgio Zoia. Music onset detection based on resonator time frequency image. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(8):1685–1695, 2008.