Repetition-based Structure Analysis of Music Recordings

Wiederholungsbasierte Strukturanalyse von Musikaufnahmen

Dissertation

Der Technischen Fakultät

der Friedrich-Alexander-Universität Erlangen-Nürnberg

zur

Erlangung des Doktorgrades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt von

Nanzhu Jiang

aus

Jilin, China

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung:	11 Feb. 2015
Vorsitzende des Promotionsorgans:	Prof. Dr.-Ing. habil. Marion Merklein
1. Gutachter:	Prof. Dr. Meinard Müller
2. Gutachter:	apl. Prof. Dr. Frank Kurth

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.

Erlangen, Dezember 1, 2014

Nanzhu Jiang

Acknowledgements

I am thankful to many people who supported me during the whole PhD studies.

First of all, I would like to express many thanks to my husband, Shuyan Liu, for the warm-hearted support during all these years. From the bottom of my heart, I would like to give all my best thanks to him.

I would like to express thousands of gratitude to my supervisor, Meinard Müller, for the excellent supervision and many of the in-depth scientific discussions. It is him who opens the opportunities for my research, in both Max-Planck-Institute Informatik in Saarbrücken and Friedrich-Alexander-University of Erlangen-Nuremberg. I believe that in my life long time I will benefit from his guidance and philosophy not only towards scientific work but also towards life.

I also would like to thank many of my friends and colleges who support me during the writing of my thesis. Here I sincerely thank my colleagues Jonathan Driedger, Thomas Prätzlich, Christian Dittmar, Peter Grosche, Verena Konz, Thomas Helten, and Andreas Baak. All of you have helped me in various during my PhD studies. I am sincerely thankful for that. Besides that, I would also thank Stefan Balke and Patricio Lopez-Serrano as well as other colleagues, who helped me with proof-reading of this thesis. Also I am grateful for my friends Shujie Li, Shujie Guo, Min Ye, Qian Ma, Yuxin Gao, Zhe Zuo, Lichao Li, Yecheng Gu, Zhihu Chen, and Qi Gao. All of you have brought happiness to my life in Germany.

I am thankful to the very friendly secretaries: Elke Weiland and Tracy Harris in Erlangen as well as Sabine Budde and Ellen Fries in Saarbrücken. Thank you for helping me in all kinds of administrative circumstances. Thousands of thanks to Elke Weiland and Vlora Arifi-Müller, who gave me many useful guidance for the life in Germany and advised me in various aspects about female career and life.

Finally, I would like to appreciate my parents. It is their love and care to support me all the time, which bring me the braveness to face difficulties and challenges in life.

Abstract

Music Information Retrieval (MIR) is a current area of research which aims at providing techniques and tools for searching, organizing, processing and interacting with music data. In order to extract musically meaningful information from audio recordings, one requires methods from various fields such as digital signal processing, music theory, human perception, and information retrieval. One central research topic within MIR is referred to as music structure analysis, where an important goal is to divide a music recording into temporal segments and to group these segments into musically meaningful categories. The extracted structural information can be used for a variety of other MIR tasks including music navigation, audio thumbnailing, audio summarization, and chord recognition.

The structure of a music recording depends on various principles such as temporal order, repetition, contrast, variation and homogeneity. Based on these principles, many approaches for music structure analysis have been proposed in the literature. However, it remains difficult to perform music structure analysis in a fully automated fashion. One reason is that music structure can be considered on different temporal levels so that even music experts may disagree on how to structure a given piece of music. Furthermore, the task of music structure analysis is complex when analyzing audio recordings due to possible acoustic variations across different musical sections.

In this thesis, we focus on repetition-based approaches for music structure analysis. As one main contribution, we introduce a novel fitness-based method that extracts repetitive structures from audio recordings. First, using signal processing techniques, the given audio recording is converted into a feature sequence that captures harmonic and melodic aspects. Next, using the concept of similarity matrices, the feature sequence is analyzed with respect to recurring patterns. In particular, we discuss various enhancement techniques to cope with musical variations such as tempo differences and transpositions. Using alignment techniques related to Dynamic Time Warping (DTW), we introduce a novel fitness measure that assigns a fitness value to each segment. Each fitness value expresses how much and how well the respective segment explains the repetitive structure of the entire recording. This fitness measure serves as the main basis for several other contributions made in this thesis.

First of all, we deal with a subproblem of music structure analysis called audio thumbnailing with the goal to determine the audio segment that best represents a given music recording. We show that our fitness measure is useful in detecting suitable audio thumbnails by considering segments of high fitness. Then, we present a novel scape plot representation that makes it possible to visualize repetitive structures of the entire music recording in a hierarchical, compact, and intuitive way. This visualization does not only indicate the benefits and limitations of our methods, but also yields interesting musical insights into the data. As an application within musicology, we show how our techniques can be applied for analyzing and segmenting music recordings in sonata form. To this end, we adapted our repetition-based approach for detecting the coarse structure of a sonata (exposition, development, recapitulation) and introduced a rule-based approach measuring local harmonic relations for analyzing finer substructures. Furthermore, we discuss how the fitness-based structure analysis can be extended for deriving more general musical structures that consist of several groups of repeating segments. As a further technical

contribution, we show how the computational efficiency of our structure analysis approach can be improved significantly by using multi-resolution strategies.

Zusammenfassung

Das aktuelle Forschungsgebiet des *Music Information Retrieval* (MIR) befasst sich mit der Bereitstellung von Techniken und Werkzeugen zum Suchen, Organisieren, Verarbeiten sowie zur Interaktion mit Musikdaten. Um musikalisch sinnvolle Informationen aus Audioaufnahmen zu extrahieren, werden Methoden aus vielen Bereichen, darunter digitale Signalverarbeitung, Musiktheorie, menschliche Wahrnehmung und dem *Information Retrieval* eingesetzt. Ein zentrales Forschungsgebiet im MIR ist die Musikstrukturanalyse, bei der ein Musikstück in zeitliche Segmente zerlegt wird und diese anschließend in musikalisch sinnvolle Kategorien gruppiert werden. Die extrahierte Strukturinformation kann für eine Vielzahl anderer MIR-Aufgabenstellungen wie Musiknavigation, *Audio Thumbnailing*, Audiozusammenfassung und Akkorderkennung verwendet werden.

Die Struktur einer Musikaufnahme hängt von verschiedenen Aspekte wie der zeitlichen Reihenfolge, von Wiederholungen, Kontrasten, Variationen und Homogenität ab. In der Literatur finden sich viele Ansätze zur Musikstrukturanalyse, die auf diese Prinzipien aufbauen. Allerdings erweist es sich als schwierig, eine Musikstrukturanalyse vollständig automatisiert durchzuführen. Ein Grund dafür ist, dass Musikstruktur auf verschiedenen zeitlichen Stufen betrachtet werden kann, sodass selbst Musikexperten darüber streiten mögen, wie ein konkretes Musikstück zu strukturieren sei. Weiterhin ist die Musikstrukturanalyse von Audioaufnahmen eine schwierige Aufgabe aufgrund möglicher akustischer Unterschiede über verschiedene musikalische Passagen hinweg.

In dieser Arbeit konzentrieren wir uns auf wiederholungsbasierte Ansätze zur Musikstrukturanalyse. Als Hauptbeitrag stellen wir eine neuartige, *fitness*-basierte Methode vor, welche geeignet ist, Wiederholungsstrukturen in einer Audioaufnahme zu detektieren. Hierzu wird die Aufnahme zuerst mittels Methoden der Signalverarbeitung in eine Merkmalsdarstellung überführt, die harmonische und melodische Eigenschaften abbildet. Durch die Verwendung von Selbstähnlichkeitsmatrizen wird diese Merkmalsdarstellung bezüglich wiederholt auftretender Muster analysiert. Insbesondere diskutieren wir verschiedene Verbesserungsstrategien, um musikalische Variationen wie Tempoänderungen und Transpositionen abzudecken. Durch Alinierungsmethoden ähnlich dem *Dynamic Time Warping* (DTW) führen wir ein neuartiges *Fitnessmaß* ein, welches jedem Segment einen sogenannten Eignungswert zuordnet. Jeder dieser Werte gibt an, wie gut und zu welchem Anteil das jeweilige Segment die Wiederholungsstruktur der kompletten Aufnahme erklärt. Dieses Maß dient als Ausgangspunkt für einige weitere Beiträge dieser Arbeit.

Zuerst beschäftigen wir uns mit einem Teilproblem der Musikstrukturanalyse namens *Audio Thumbnailing*, welches zum Ziel hat, das Audiosegment zu bestimmen, welches ein Musikstück am besten beschreibt. Wir zeigen, dass unser *Fitnessmaß* zur Bestimmung von sinnvollen *Audio Thumbnails* durch Betrachtung von Segmenten mit hoher *Fitness* geeignet ist. Anschließend präsentieren wir eine neue *Scape-Plot*-Darstellung, welche die Visualisierung von Wiederholungsstrukturen des gesamten Musikstückes auf eine hierarchische, kompakte und intuitive Weise ermöglicht. Diese Visualisierung zeigt nicht nur die Möglichkeiten und Grenzen unserer Methoden auf, sondern führt auch zu interessanten musikalischen Einblicken in die Daten. Als Anwendung in der Musikwissenschaft zeigen wir, wie unsere Techniken zur Analyse und Segmentierungen von Audioaufnahmen in der Sonatenhauptsatzform verwendet werden können. Hierzu wird unser wiederholungsbasier-

ter Ansatz zur Ermittlung der Grobstruktur einer Sonate (Exposition, Durchführung, Reprise) angepasst und ein regelbasierter Ansatz zum Messen lokaler harmonischer Beziehungen eingeführt. Weiterhin diskutieren wir, wie die *fitness*-basierte Strukturanalyse erweitert werden kann, um allgemeinere musikalische Strukturen bestehend aus mehreren Gruppen wiederholter Segmente aufzufinden. Als einen weiteren technischen Beitrag zeigen wir, wie die Rechenzeit für unseren Strukturanalyse-Ansatz durch die hierarchische Betrachtung mehrerer Auflösungsstufen signifikant verringert werden kann.

Contents

Chapter 1

Introduction

Music is a pleasure that people often enjoy in daily life. Everywhere we go, we listen to different kinds of music according to the situation: gentle music played at restaurants for relaxation; dance music at pubs and discotheques provides an exciting atmosphere; pop music on the car radio makes driving more pleasant; and classical music at theaters and concerts for pure enjoyment. With its diverse compositional styles, music brings us various listening experiences and affects our mood by evoking emotions such as happiness, excitement or inspiration. We kind of need music in many of the situations to elicit particular emotions and feelings, therefore music has already become a substantial part of our lives.

As modern technologies develop rapidly, music has become much easier to access from computers, portable players, and mobile devices. This has also changed the way that people acquire music. Nowadays, thousands of music recordings are uploaded daily to the Internet, propagating rapidly to a mass of listeners all around the world. Given that millions of music tracks are available on websites, assisting tools for dealing with these music tracks are in high demand. These tools comprise, for example: software for personal music collection organization and management [74, 84, 134]; music identification apps to directly retrieve a song based on its musical content or title [3, 4, 16, 17, 87, 135]; music recommendation systems [8, 9] to assist users in finding interesting tracks (which fit their own preferences) from vast collections of albums that are updated daily; music navigation systems that help users quickly jump to a certain point or sub-clip within a long track. Up to now, most music websites still use single-colored playback bars which only indicate the current time position. Although it is possible to jump inside a track by clicking or typing a time point, such jumping function still does not satisfy the users' need for quickly locating positions of certain musical content. This is especially true for long tracks—such as symphonies and concerts—where providing navigation information like structure sections alongside the playback bar can be very helpful for users. However, due to the huge amount of music data, generating this type of information manually is often a dull task, involving onerous labor. It is therefore necessary for information technologies to supply automated tools which assist users in dealing with music easily and conveniently.

Music Information Retrieval (MIR) is a research field which aims at providing methodological solutions to musical content-based searches [18], seeking to fulfill the demands of

users while searching, processing and interacting with music. The ultimate goal of this field is to develop automated methods for extracting musically meaningful information which is highly related to music content [49]. People working in this field have conducted many researches and experiments to derive such information. According to the specific task, some examples are: pitch information representing which notes are played; beat information that indicates how the rhythm is performed; timbre information that describes what instruments are used in a recording. In order to extract such musically meaningful information, a combination of techniques are needed from other fields which involve digital signal processing, music theory, physics, human perception, information retrieval, and machine learning. Therefore, music information retrieval is an interdisciplinary field which usually involve researchers coming from diverse fields to collaborate together to tackle the challenging tasks in this field.

Although much research has been done in recent years, automated music processing still remains quite challenging because of the complex and variable nature of music signals. For instance, a typical music recording is usually polyphonic (a mixture of sounds generated by several instruments or human voices). When various instruments are played together, the sound signals that they generate interact and interfere with each other, resulting in acoustic properties that are very different from the simple sum of individual sound. Performing automated music analysis on such recordings is, in general, rather difficult. One needs to consider many substantial problems which might lead to a method's failure. Furthermore, human perception of music signal may differ substantially from the sound that the signal actually presents. For example, if a note is played on a piano, most people can perceive the sound of that note, but not the sound of its harmonics. When several notes are played together, we might only hear some of the strong notes and neglect the weaker notes because the sound of them are masked by the strong ones. Moreover, human judgments about music can contrast starkly with those of machines. As an illustration, take a popular song with two refrain sections, one refrain is played by four instruments and the other is played by five. A human listener might consider that the two refrain sections sounds somehow the same [1], but machine algorithms might judge that the two sections have timbre difference. All these problems need to be considered when attempting to develop automated methods for extracting musically meaningful information. More specifically, one needs to design algorithms adapted to human perception, which are therefore necessary to be tolerant for some variations.

An automated music analysis procedure typically consists of two main steps. Firstly, according to the task specification (or user demand), an audio file is converted into a set of musically meaningful features by applying knowledge from physics and music theory, as well as techniques mainly from digital signal processing. Then, these features are fed into a machine learning and information retrieval pipeline to further compute the required information of the task. In the case of a chord recognition task, users usually want to know the underlying chords played in an audio file. For this scenario, an automated procedure would first convert the audio into some features which reflect its harmony and then use classification modules to generate chord labels. Another example is the beat tracking task: note onset features (describing time positions that notes are played) are first extracted

[1] According to [118, 129], most people can distinguish up to four instruments in polyphonic music even if the music is played by more instruments.

and then analyzed for periodicity. Based on these features, beat positions and tempo information can be further derived.

Among all the research topics in MIR, in this thesis we deal with a central research topic: automated structure analysis of music recordings. One major goal of structure analysis is to divide a music recording into temporal segments and then group these segments into musically meaningful categories [108]. Typical structural information could be, for instance, the intro, verse, chorus or bridge sections of a popular song; or the first theme, the second theme, and transition sections of a classical work; or different stanza sections of a folksong. These are all semantic sections of a particular type of music. According to the purpose, structure information could also be derived from other principles such as solo or non-solo sections for choir music, speech or singing sections for opera, or purely instrumental or vocal sections for many kinds of music. Sometimes, there could also be the case that no semantic labels can be given. In this case, sections within a musical piece can also be labeled using symbols. For instance, A_1 and A_2 describe similar (yet not identical) sections, B and C represent dissimilar sections, whereas a_1 and a_2 identify subsections within a larger A section. As stated in [57], having such structural information allows easily integrating intra-piece navigation into a music playback bar, generating representative short clips for the entire recording, or even using this information as prior knowledge to aid further music processing tasks.

Although structure information for musical pieces with simple sections and no strong variations can be readily derived with existing techniques, structure analysis in general remains a difficult task. Firstly, one needs to keep in mind that all structural sections (such as those mentioned above) are subjective decisions by human annotators. Thus, a general agreement may not be reached among several annotators to ensure a *correct* or *perfect* decision. Various musicians may have differing opinions about the structure of a given musical piece —- even in the case of classical works that have been extensively studied. As an example, for a number of Beethoven's sonatas, different musical analysis books document different decisions on the existence and positions of some transition sections. Secondly, when dealing only with audio recordings (i.e., without sheet music), structure information is hard to derive—especially if the audio segments contain pronounced musical variations. For example, for a popular song, a repeated verse section may share the same melody as the other verse sections, but performed only purely instrumental whereas other verse sections are with singing voice. Such kind of timbre change brings many challenges to the music structure analysis. Thirdly, many of the music pieces often constitute hierarchical structures. This is particularly true for large-scale music compositions such as symphonies. It is still a question how coarsely or how finely the structure of a piece should be outlined. There exist many possible levels: movements, passages, themes, motifs, measures or chords—it is therefore challenging to decide an appropriate degree to reveal all these sections of different hierarchies.

Despite the difficulties described above, the ultimate goal of this thesis is to to conquer these challenges and develop automated methods to derive structural information from music recordings. As stated by [108], musical structure depends on various principles such as temporal order, repetition, contrast, variation, and homogeneity. Therefore, many approaches to music structure analysis have been developed, which can be roughly grouped into three classes according to [108]. First, repetition-based methods, which are employed

to identify recurring patterns; Second, novelty-based methods, which are used to detect transitions between contrasting parts; Third, homogeneity-based methods, that determine which passages are consistent with respect to a given musical property. In all three cases, one has to account for different musical dimensions such as melody, harmony, rhythm, or timbre [95]. In this thesis, we contribute to repetition-based music structure analysis where we particularly aimed for finding repetitive structures in a given audio recording. We derive meaningful structure information by considering the quality and quantity of these repetitions. It should be noted that we perform structure analysis exclusively upon audio recordings, without using symbolic data such as MIDI or scores as prior information.

The structure analysis techniques introduced in this thesis can be briefly described as follows. Starting with a music audio recording, we use signal processing techniques to extract musically meaningful features that correlate to aspects of harmony and melody. Next, we compute the similarity matrix which can reveal similar audio feature sequences (similar audio segments). We carefully enhance the similarity matrices in order to suppress the musical variations such as tempo change and transposition. After that, by adapting classical sequence comparison methods such as Dynamic Time Warping (DTW), we designed a novel procedure that allows for extracting all repeated segments of a given segment within the recording. Having this repetition-extraction as basis, we further extend and improve several other structure-related tasks such as efficient estimation of the most representative segment of a recording; displaying structure hierarchies as a compact visualization, revealing structure for special music recordings in particular music forms. More detailed contributions can be found in the next section.

1.1 Contributions

In this section, we present main contributions of this thesis in a chapter-wise fashion, where each paragraph corresponds to the contributions condensed for each chapter.

We first introduce in Chapter 2 a basic analysis tool for music structure analysis, the similarity matrix. It was firstly used by Foote [36] in the music domain. Within the research field of music processing, the concept of similarity matrices (SMs) has been used for a multitude of analysis and retrieval tasks, including structure analysis and version identification (for example, [11, 12, 19, 20, 30, 35, 45, 52, 61, 71, 78, 95, 107, 108, 119, 121]). For such tasks, improving the structural properties of the SMs at an early stage in the processing pipeline has turned out to be of crucial importance [108]. In this chapter, we take several existing enhancement strategies for similarity matrices as introduced in [89, 97, 121], and implement essential enhancing techniques which assist for structure analysis. Such techniques allow for identifying similar music segments even under strong musical variations in harmony, melody or tempo. After the enhancements, similar segments could be revealed by the path-like (or stripe-like) structures in a similarity matrix. Therefore, by extracting these path-like structures, we can derive repetition relationships between the segments. This is a crucial step in music structure analysis. As a main contribution, we provide MATLAB implementations for SM computation as well as the enhancement techniques in the form of a toolbox—the SM Toolbox. Using this toolbox, one can process similarity matrices in various ways.

Having these enhanced similarity matrices, we can then use them to detect repetitive structures of an audio recording by identifying paths in similarity matrices. In Chapter 3, as the main contribution, we introduce an automated procedure that can extract all repetitions of a given segment. Our idea is to design a novel fitness measure that assigns a fitness value to a segment which expresses how much and how well a segment "explains" the repetitive structure of the entire recording. Having this fitness measure, we can then deal with an important sub-task of audio structure analysis: audio thumbnailing. The goal of this task is to determine one audio segment that best represents a given music recording. Typically, such a segment has many (approximate) repetitions covering large parts of the piece. Therefore, to derive the thumbnail, one needs to detect the most repetitive segment within the recording. In our thumbnailing procedure, Since we have no prior knowledge of the thumbnail's length or location, we compute fitness values for all possible segments and then defined the thumbnail to be the fitness-maximizing segment. In the design of the fitness measure, we introduce an optimization scheme that jointly performs path structure extraction and grouping—two error-prone steps that are usually performed successively. As a result, our approach is capable of coping with strong musical and acoustic variations that may occur within and across related segments. As second contribution, we introduce the concept of fitness scape plots that shows the fitness values over all possible audio segments. A visualization of this fitness scape plot yields a compact high-level view on the structural properties of the entire music recording. Finally, to show the robustness and practicability of our thumbnailing approach, we present various experiments based on different audio collections comprising popular and classical music. As a side remark, the MATLAB code for our thumbnailing procedure is also provided.

From some experiments conducted in this thesis we know that our proposed thumbnailing procedure can usually detect the thumbnail segment successfully for a given music recording. However, because we perform brute-force computing for all possible segments about their repetitiveness, one main drawback of the procedure is the long computation time. For a given recording of three minutes, the procedure need roughly one minute to derive the thumbnail. In Chapter 4, we show how the computational efficiency of a our state-of-the-art thumbnailing approach can be improved significantly. As a first acceleration strategy, we propose an efficient multi-level sampling strategy to reduce the number of segments the fitness has to be computed for. Second, we obtain further accelerations by suitably adjusting the resolution used in the fitness computation depending on the level of the segment. As a third acceleration strategy, we exploit an intrinsic property of the fitness computation that allows us to estimate the fitness for certain segments without any further computation. Our experimental results show that combining these three strategies leads to accelerations by a factor of 20 to 200 depending on the duration of the song while keeping the overall accuracy for the thumbnail estimation.

The thumbnailing procedure we proposed can reveal the repetitive properties of a given segment throughout the whole recording. However, analyzing each single segment separately is not sufficient to provide the overall structure of the whole recording. Moreover, the potential hierarchical relationships between longer and shorter segments are missing. For example, one repeating long segment might consists of two short sub-segments that are also repeated. Focusing only on segments of certain length is not enough to reflect the hierarchical relations between the segments. Therefore, it remains an unsolved prob-

lem that how to reveal the relationships for all segments of a given piece of music. In Chapter 5, we present a novel scape plot representation that allows for visualizing repetitive structures of the entire music recording in a hierarchical, compact, and intuitive way. Within a scape plot, each point corresponds to an audio segment identified by its center and length. As our main contribution, we assign to each point a color value so that two segment properties become apparent. Firstly, we use the lightness component of the color to indicate the repetitiveness of the encoded segment, where we revert to a recently introduced fitness measure. Secondly, we use the hue component of the color to reveal the relations between different segments. To this end, we introduce a novel grouping procedure that automatically maps related segments to similar hue values. By discussing a number of popular and classical music examples, we illustrate the potential and visual appeal of our representation and also indicate limitations.

After introducing the repetition detection procedure in previous chapters, Chapter 6 discusses how it can be applied to analyze music with hierarchical structures. We investigate music pieces in sonata form, as it is one of the most important large-scale musical structures, posing a challenge on account of its sections at different hierarchical levels. Typically, the first movements of symphonies and sonatas follow the sonata form, which (in its most basic form) starts with an exposition and a repetition thereof, continues with a development, and closes with a recapitulation. The recapitulation can be regarded as an altered repeat of the exposition, where certain substructures (first and second subject groups) appear in musically modified forms. As the first contribution of this chapter, we introduce automated methods for analyzing music recordings in sonata form, where we proceed in two steps. In the first step, we derive the coarse structure by exploiting the fact that the recapitulation is a (modified) repetition of the exposition. This requires audio structure analysis tools that are invariant under local modulations. In the second step, we identify finer substructures by capturing relative modulations between the subject groups in exposition and recapitulation. We evaluate and discuss our results by means of the Beethoven piano sonatas. As the second contribution, we introduce a novel visualization that not only indicates the benefits and limitations of our methods, but also yields some interesting musical insights into the data.

In Chapter 7, we discuss full structure analysis of music recordings. As our main technical contributions, we propose two repetition-based approaches, both extending the previously introduced thumbnailing procedure. In the first approach, which is straightforward, we simply apply the original thumbnailing procedure in an iterative fashion, with each iteration deriving a group of repetitive segments correspond to one musical part. In the second approach, we introduce a novel method for jointly estimating the two most repetitive segments within one optimization procedure. By extracting the repetitions of the two thumbnail segments, we can also identify large portions of the repetitive structures of the music recording. Finally, we report on experimental results demonstrating the performances of the two approaches and also point out the strengths and limitations of our approaches.

1.2 Included Publications

The main contributions of this thesis are based on the following publications, which appear in conference proceedings and journal articles in the field of music signal processing and music information retrieval.

[93] Meinard Müller, Nanzhu Jiang, and Harald Grohganz. SM Toolbox: MATLAB implementations for computing and enhancing similiarty matrices. In *Proceedings of the AES Conference on Semantic Audio*, London, GB, 2014.

[95] Meinard Müller, Nanzhu Jiang, and Peter Grosche. A robust fitness measure for capturing repetitions in music recordings with applications to audio thumbnailing. *IEEE Transactions on Audio, Speech & Language Processing*, 21(3):531–543, 2013.

[91] 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.

[57] Nanzhu Jiang and Meinard Müller. Towards efficient audio thumbnailing. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, 2014.

[92] Meinard Müller and Nanzhu Jiang. A scape plot representation for visualizing repetitive structures of music recordings. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, pages 97–102, Porto, Portugal, 2012.

[56] Nanzhu Jiang and Meinard Müller. Automated methods for analyzing music recordings in sonata form. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, pages 595–600, Curitiba, Brazil, 2013.

[58] Nanzhu Jiang and Meinard Müller. Estimating double thumbnails for music recordings. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015.

1.3 Supplemental Publications

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

[48] Harald Grohganz, Michael Clausen, Nanzhu Jiang, and Meinard Müller. Converting path structures into block structures using eigenvalue decompositions of self-similarity matrices. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, pages 209–214, Curitiba, Brazil, 2013.

[94] Meinard Müller, Nanzhu Jiang, Harald Grohganz, and Michael Clausen. Strukturanalyse für Musiksignale. In *Proceedings of 43th GI Jahrestagung*, pages 2943–2957, Koblenz, Germany, 2013.

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

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

[130] Balaji Thoshkahna, Meinard Müller, Venkatesh Kulkarni, and Nanzhu Jiang. Novel audio features for capturing tempo salience in music recordings. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015.

1.4 Related Work

In this section, we summarize the related work and available structure approaches. Until now, there are already some summary articles or book chapters which discuss music structure analysis. Such literature include: the lecture notes by Peeters [110], the overview article of Paulus et al. [108], the article by Dannenberg and Goto [30], book chapter by Müller [88] and book chapter in [62].

As detailed in [108], musical structure depends on various principles such as temporal order, repetition, contrast, variation, and homogeneity. Therefore, a large number of different approaches to music structure analysis have been developed, whereas one can roughly distinguish between three different classes of methods. First, repetition-based methods are employed to identify recurring patterns. Second, novelty-based methods are used to detect transitions between contrasting parts. Third, homogeneity-based methods are used to determine passages that are consistent with respect to some musical property. As stated by Peeters [109–111], according to the techniques used in the approaches, repetition-based structure analysis methods can be considered as *sequence approaches*, whereas the other two classes are considered as *state approaches*.

1.4.1 Repetition-based Approaches

Since we focus on repetition-based structure analysis in this thesis, here we mainly summarize the available approaches which based on detecting repetitions or repeated patterns.

Most repetition-based approaches usually treat the music signal as repetition of sequences of sound events [110]. Here, the order of the sequence is important to form the music material such as melody and chord progression [108]. Most approaches aimed to first detect repeated sequences, and then use them as a cue to segment the musical signal.

Since melody and harmony are important musical aspects for detecting repetitions, many of the approaches use chroma features, which are more robust for small variations, for structure analysis. Chroma features have been first introduced to music structure analysis by Bartsch and Wakefield [10], where they proposed an audio thumbnailing procedure for popular music. Later, Dannerberg and Hu also proposed the usage of chroma features for discovering music patterns [31]. Also Goto used chroma features in his work [43] to detect chorus sections for popular songs.

One common basic tool used in repetition-based approaches is the similarity matrix (or distance matrix). We discuss the similarity matrix in detail in Chapter 2. Foote first used

the similarity matrix to visualize the music structure in [36]. As the very early contribution, Cooper and Foote report the use of similarity matrices to summarize structure for popular music [29]. Instead of using similarity matrices, some approaches use time-lag matrices (for example, Goto used time-lag matrices in [43]), which can be considered as a kind of rotated version of similarity matrices.

As we will discussed in Chapter 2, two similar feature sequence, which represent two similar music sections, will result in high consecutive similarity values which look like the shape of "stripe" or 'path'. If the two sections of music are played in roughly the same tempo, the stripes are parallel to the diagonal direction. Many of the approaches aim at detecting such stripes to derive similar feature sequences, or in other words, repeated audio material. Such stripes may be easy for human eyes to detect, however, because of the small gaps inside the stripes and some local noise around the stripes, it is difficult for automated algorithms to directly extract these stripes without any pre-processing. Therefore, for many approaches, one important step is to enhance the stripes in the similarity matrix. One idea for stripe enhancement is to apply some low-pass filters along the diagonal direction to smooth the gaps in the stripe and also blur the local noises [108]. For example, Bartsch and Wakefield process the similarity matrix in this way for their proposed thumbaniling procedure [11]. Another similar idea is to use two filters, where one first apply a low-pass averaging filter on the diagonal direction to reinforce diagonal stripes and then a high-pass filter on the anti-diagonal direction to remove non-diagonal noises. This strategy is proposed by Aucouturier and Sandler in their work [6]. Also Peeters use this strategy in [109]. In addition, a third kind of idea is proposed by Marolt in [79] to find melodic patterns. He perform the stripe enhancement and noise reduction in the similarity matrix by first calculating several similarity matrices with different window lengths of involved features and then multiplying these matrices element-wise. Furthermore, several approaches enhance the stripes by image processing techniques. For example, Lu et al. [75] apply erosion and dilation filters along the diagonal direction to close small gaps in significant repetition stripes, and remove short line fragments which may cause by noise. Similar technique is applied in the approach proposed by Ong et al. [104], where the very short line segments are removed by application of the opening operation of a morphological filter. Note that most of the above mentioned approaches assume that the repeated sections are played in the same tempo. This is the main reason that the smoothing are performed in the diagonal direction. However, when repeated sections are played in different tempo, the stripes are not straight lines but more like curves which bended in some other directions. Müller and Kurth proposed a modified smoothing method to handle such situation with the idea to incorporate contextual information into the local similarity measure [97].

After enhancing the stripes in the similarity matrix, the next step is to extract the stripes from the similarity matrix, and to further group them into different classes for deriving repetitions correspond to different parts of the music. Dannerberg and Hu described in [31] several path extraction and cluster techniques such as dynamic programming and iterative greedy algorithms. Goto developed a so called *RefraiD* method [43] to extract all chorus sections for a popular song, which is based on thresholding. It uses a time-lag version of the similarity matrix where lags represent possible repetitions, and then selects high peaks above an automatically determined threshold to search the line segments. Shiu et al. [123] interpret similarity entries as the probabilities generated by certain states at certain time,

then he performed the Viterbi algorithm to detect the path with largest cumulative probability. Müller and Kurth also proposed in [98] a path extraction algorithm, where the main idea is to iteratively locate the path at the highest similarity value and construct the path with some heuristics. Besides of the above mentioned approaches, according to [105], there are also approaches based on Hidden Markov Models (HMM), we refer to the related literature [6, 73, 115].

1.4.2 Novelty-based Approaches

As many of the listening experience has shown, strong changes in music often imply start or end of a new structural section. Therefore, novel-based structure analysis approaches aim at detecting such changes of an audio recording and use them to segment the recording. The basic idea of these methods is to compute a novelty value that tells the music differences between a certain time position and its previous time position. Such music difference may come from various musical aspects, for example, dynamics, harmony, timbre or rhythm. Foote [37] proposed a method to find segment boundaries by deriving a novelty curve. This curve is computed by correlating a 2D checkerboard kernel along the diagonal of the similarity matrix. In the end, the high peaks in the novelty curve are selected as segment boundaries.

There are also other novelty-based approaches. For example, Tzanetakis and Cook [133] proposed to a segmentation approach where they first extract several kinds of features from music signal. Such features include spectral centroid, spectral Flux, and zero crossings. Next, the differences between successive feature frames are computed using Mahalanobis distance measure. After that, the derivative of the distance is taken and peaks (segment boundaries) are picked using simple heuristics. In this approach, the peaks roughly correspond to texture change in music. Besides this, Jensen [53] proposed to segment the music according to rhythmic aspect of music. He proposed to use rhythm-based features to calculate similarity matrices and also a smoothed novelty measure which calculated on a small square on the diagonal of an SSM. In this way, he minimized computaion cost and derived an efficient boundary estimation method. Moreover, Jensen proposed in his later work [54] an segmentation pipeline where he used features based on rhythm, timbre and harmony to calculate SSMs and and tried to find block structures along the main diagonal of SSMs. The segmentation problem was then formulated as an optimization problem and he proposed a cost function to be minimize the average similarity within the blocks while keeping the number of segments small [108]. Turnbull et al. [132] proposed a boundary detection method using different features that capture music changes in timbre, harmony, or rhythm aspects. The boundaries are further derived by peak picking or by a supervised classifier. Also Kaiser and Peeters [59] proposed a multi-scale novelty approach that allows to capture segment boundaries of different hierarchical levels. Serra et al. [120] proposed a novel segmentation procedure where they generate robust boundary candidates using time lag matrix and processed by statistical methods, then the pairwise segment similarity are computed to derive labels for the segments.

1.4.3 Homogeneity-based Approaches

As the third class of structure analysis methods, homogeneity-based approaches analyze the music signal with respect to consistency in musical property. As stated by [108], after the novel-based approaches which derive the segment boundaries, the content of the segments are analyzed and the segments are classified by homogenous clusters. Coope and Foote [29] proposed an approach where they first estimate segment boundaries using a novelty-based method, and then compare pairwise segment distance using Kullback-Leibler divergence. They presented the pairwise distances in a so called 'segment-indexed similarity matrix' and then applied Singular Value Decomposition to perform clustering of segments. Besides this, there are also similar approaches, see [42, 73]. Furthermore, as stated in [108], some of homogeneity-based methods involve Hidden Markov Models (HMMs), where each musical part can be considered as a state in a HMM. For detail, we refer to the literature [1, 2, 5, 7, 38, 71, 72, 112].

1.4.4 Other approaches

In this section, we describe several approaches that cannot be classified as the three above mentioned classes.

Some approaches combine different principles together to perform the structure analysis. For example, Paulus and Klapuri [106, 107] proposed a cost function that combine various segmentation principles for structural descriptions. Then for a given recording, the cost function is minimized over all possible descriptions to generate the segmentation and labels. Besides this, Kaiser and Peeters [60] proposed a method to the state and sequence segmentation in one scheme.

There are also different approaches using different tools to perform structure analysis. Kaiser and Sikora [61] proposed a method using non-negative matrix factorization (NMF) [68] to segment regions of acoustically similar frames in a self-similarity matrix. Their method proceeds as follows. Firstly, a self-similarity matrix is computed based on timbre related features. Next, the SSM is segmented by a novelty-based method to find segment boundary candidates. After that, the SSM is decomposed by NMF. The resulting decomposed matrices closely related with structural parts of the music recordings. Finally, clustering techniques are applied to group segments belonging to the same musical parts. Besides this, Chen [23] proposed to a model consists of two parts for structure analysis. The first part is to use a two-level clustering algorithm which estimate segments according to harmonic or timbre aspect of music. The second part performed labeling by combining the segments coming from the two aspects into one score matrix and then use NMF to determine the segments types. There are also other approaches based on NMF. We refer to the literature [103, 137]

Different with the thumbnailing method that select only one representative segment, Nieto et al. proposed in [102] a criterion for audio summarization where the goal is to select a set of segments that best represent the overall music recording. There, they use a measure of compression which describes loss of information as well as a measure of disjoint information which describe the non-overlap extent between chosen segments. In the end, the

summarization criterion is to defined as the harmonic mean of the two measures. The aim of the criterion is to loss as little information as possible, while avoiding overlap between chosen segments [102]. Besides this, Nieto and Bello [100] proposed music segmentation method where they use 2D fourier magnitude coefficients (FMC) as features. There, they first derive the beat-Synchronous chroma representations, and then compute 2D FMC features which can achieve transposition invariance, phase shift invariance and local tempo invariance. Next, the 2D FMC features are divided into a sequence of fixed-size patches, and a k-means clustering is applied to group the patches. Moreover, Nieto et al. recently proposed another method to identify polyphonic patterns from audio recordings. For detail, we refer to [101].

There are also many other approaches which compare the automated algorithms with human annotations for example, Levy et al. [70] use some music segmentation algorithms based on timbre and some based on harmonic features with human annotations. They concluded that no algorithm is clear superior than others but a combination of the two types algorithms is better than either of them. Besides this, Smith and Chew [126] also analyze several structure segmentation algorithms to study the relationships between the performances of the algorithms and the human annotations. In addition, Smith and Chew [127] anaylzed feature relevance for structure analysis by analyzing multiple SSMs that derived from human annotations as well as from musical features. Furthermore, Smith et al. studied the human perception of segment boundaries with respect to audio properties in [125]. They compare the human annotated boundaries with peaks in novelty curves (these novelty curves are computed by different features related with various aspects of music), and find that nearly all annotated boundaries correspond to peaks in novelty curves.

Another recently proposed approach is by Grohganz et al. [48] where repetition-based structure analysis problem is converted into a homogeneity-based problem. There, a novel procedure is introduced to converting path structures into block structures by applying an eigenvalue decomposition of the SSM in combination with suitable clustering techniques. Such conversion may open up novel ways for handling both principles within a unified structure analysis framework. Besides this, McFee and Ellis [83] also performed similar conversion using spectral graph theory.

1.4.5 Evaluation

The Music Information Retrieval Evaluation eXchange (MIREX 2014) is an annual competition of automated music analysis and retrieval methods [32, 33]. One important task is the structural segmentation task [2] which aimed at comparing different structure analysis methods. Here, three kinds of different evaluation measures, namely the boundary retrieval measure, pairwise frame measure, and normalised conditional entropies, are fixed as standard measures. These measures are described by Lukashevich [76] in detail. After some pilot tests, in this thesis, we decided to use pairwise frame measure as proposed by [71], as well as the boundary retrieval (hit rate) measure as proposed by [132] to check the performance of our structure analsis approaches in Chapter 6 and Chapter 7. Note

[2] http://www.music-ir.org/mirex/wiki/2014:Structural_Segmentation. The structure segmentation competition is organized since 2008.

that we use a different F-measure only for evaluating the performance of thumbnailing procedure in Chapter 3 and Chapter 4. As for the data used in the MIREX structure segmentation task, so far we have known that music recordings from the RWC database [44], the SALAMI database [128], and recordings from the Beatles [82] are included as test data. In this thesis, we also use the Beatles and RWC data to evaluate our proposed methods. We describe these data in detail in Section 3.7.1.

As a side remark, as many automated methods only aimed at detecting repeated patterns in a piece of music [24–27, 51, 63, 66, 67, 85], starting in 2013, there is another side-task in MIREX so called "discovery of repeated themes and sections" with the aim to find repeating patterns for music pieces. This task considers not only music recordings but also some symbolic music.

Chapter 2

Similarity Matrix

In this chapter, we discuss a widely used tool for music structure analysis—the similarity matrix. The content of this chapter closely follows the publication [93].

The fundamental concept of similarity matrices is central for the analysis of many kinds of time series. In the field of music information retrieval, similarity matrices are widely used by researchers to compare two different time series extracted from a given music recording. The relationships between different time series can be revealed by certain visual characteristics of a similarity matrix. For example, if a continuous sequence of high similarity values is present in the comparison of two time series, this usually indicates that the underlying two time series are similar to each other. In addition, if these two time series are from the same music recording, this may even suggest that the two time series are repetitions of each other. In this chapter, discuss the basic knowledge and present several enhancements for similarity matrices, which are needed for music structure analysis.

To compute a similarity matrix, generally, one starts with a feature space $\mathcal{F}$ containing the elements of the time series under consideration as well as a similarity measure $\mathbf{s} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ that allows for comparing these elements. Then given two time series $X = (x_1, x_2, \ldots, x_N)$ and $Y = (y_1, y_2, \ldots, y_M)$, the *similarity matrix* $\mathcal{S} \in \mathbb{R}^{N \times M}$ is defined by

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

where $x_n, y_m \in \mathcal{F}$, $n \in [1\!:\!N] = \{1, 2, \ldots, N\}$ and $m \in [1\!:\!M]$. Typically, the value $\mathcal{S}(n, m)$ is high (dark color in Figure 2.1) if the two elements x_n and y_m are similar, otherwise $\mathcal{S}(n, m)$ is low (light color). Instead of a similarity measure, one often uses a cost or distance measure which then results in a so-called *cost matrix* or *distance matrix*. Since such matrices can easily be converted into similarity matrices (e. g. by taking one minus cost values), we only consider in the following the case of similarity matrices.

In the case that the sequences X and Y coincide, the resulting matrix is often referred to as *self-similarity matrix* (SSM). Such matrices have been used under the name *recurrence plot* for the analysis of chaotic systems [34]. Later, Foote [36] introduced self-similarity matrices to the music domain in order to visualize the time structure of a given audio recording. Since then, similarity matrices and their relatives have been widely used for various music analysis and retrieval tasks including audio structure analysis [30, 108], structure-based

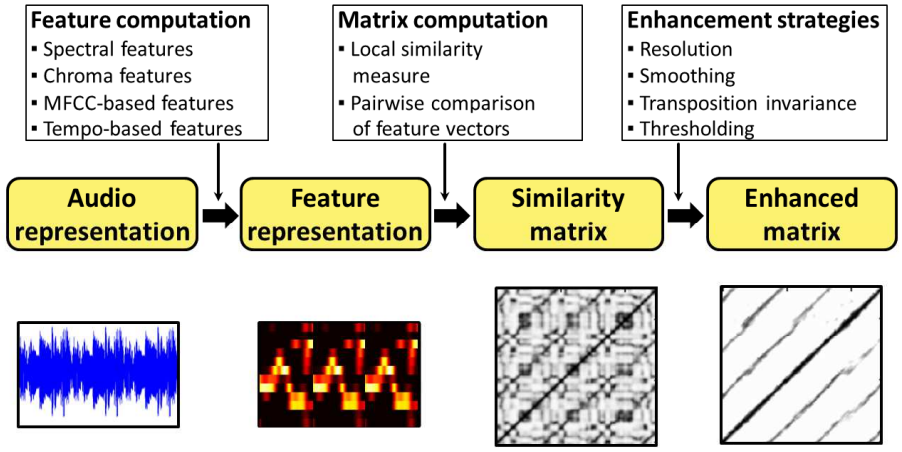

Figure 2.1: Overview of the similarity matrix computation.

audio retrieval [12], audio thumbnailing [11,45,95], music synchronization [35] and version or cover song identification [119, 121].

In the music context, the first step for computing a similarity matrix is to convert the given audio representations into suitable feature representations, which emphasize different musical aspects such as harmony, tempo, or timbre. The properties of the resulting similarity matrix crucially depend on the respective feature type as well as on the underlying similarity measure used to compare the features. Furthermore, many different smoothing, thresholding, and other strategies have been proposed for enhancing certain structural properties of a similarity matrix while suppressing unwanted, noise-like artifacts [108]. This leads to a large number of variants of similarity matrices, which may show quite different behaviors in the context of a specific music analysis task. See Figure 2.1 for an overview and Figure 2.3 for examples.

In this chapter, we first describe the fundamentals for computing a similarity matrix, and then address several enhancement strategies which are essential for music structure analysis. As the main technical contribution, we developed a toolbox, which contains MATLAB code for computing and modifying certain properties of similarity matrices. Also several enhancement strategies designed for structure analysis are clearly illustrated. We name this toolbox "the SM toolbox"[1]. In particular, the SM toolbox contains functions for enhancing path-like structures that are important in repetition-based music structure analysis. Furthermore, we also provide a number of additional tools for parsing, navigation, and visualization synchronized with audio playback in our toolbox[2].

The remainder of this chapter is organized as follows. In Section 2.1, we briefly introduce the feature representations which similarity matrices are computed from. We use chroma features, which capture harmonic content of an audio file, as the input features for computing similarity matrices. In Section 2.2, we give a summary on the various en-

[1] the SM toolbox is freely available at http://www.audiolabs-erlangen.de/resources/MIR/SMtoolbox/. It is released under a GNU-GPL license.

[2] This is the joint work with Harald Grohganz in Bonn University, for technical detail of additional tools please check [93]

hancement strategies for similarity matrices and discuss the role of the most important parameters that can be used to modify the matrices' characteristics. Then, in Section 2.3, we describe the functions of the SM toolbox. We focus on some specific techniques that are general enough to illustrate the importance of structural enhancements. Finally, in Section 2.4, we conclude this chapter with some general remarks on the suitability of the various enhancement steps depending on the specific application context.

2.1 Feature Representation

In the first step of computing similarity matrices, the waveform-based audio recordings are transformed into suitable feature representations, which capture specific acoustic and musical properties. As detailed in [108], the suitability of a feature type largely depends on the respective application. For example, MFCC[3]-based and related spectral-based features may be suitable to capture aspects such as instrumentation and timbre. Other features based on onset and novelty curves or tempograms are used to capture beat, tempo, and rhythmic information. Finally, chroma-based audio features, which relate to harmonic and melodic properties, have turned out to be a powerful tool for many music analysis and retrieval tasks. Each 12-dimensional chroma vector describes a signal's local energy distribution over an analysis window (frame) across the 12 pitch classes of the equal tempered scale (ignoring octave information). Hence, the resulting feature space is $\mathcal{F} = \mathbb{R}^{12}$. As an example, we use in the following a chroma variant referred to as `CENS` (Chroma Energy Normalized Statistics) features [88, Section 3.3][4], which come along with two parameters: a length parameter $w \in \mathbb{N}$ controlling the size of the analysis window (frame length) and a downsampling parameter d controlling the feature rate. We denote the resulting feature as $\texttt{CENS}(w, d)$, see [90] for details.

In order to have a better illustration for `CENS` features and similarity matrices, we first generate an audio file to serve as a running example. Later, we will discuss different `CENS` features and similarity matrices computed from this audio file. To this aim, first we extract some sections from a piano audio recording and synthesize them such that the result audio file consists of four structural parts $A_1A_2BA_3$, with the total length of 142 seconds. In this audio file, A_2 is a modulation of A_1 transposed by one semitone upwards, whereas A_3 is a repetition of A_1, however in a much faster tempo. The B part is a totally different part compared to the three A parts.

Next, in Figure 2.2 we illustrate several `CENS` features computed using different parameter settings. The `CENS` features in this figure are all extracted from the synthesized example audio recording mentioned above. In the default setting of `CENS` features, each feature frame covers 0.1 second of audio information, and therefore the original feature resolution is 10 Hz, meaning 10 features per second. By setting the parameters $w = 1$ and $d = 1$, Figure 2.2a shows the basic $\texttt{CENS}(1, 1)$ features with analysis window length of 1, and downsampling factor of 1. This means the features considers only 1 frame in each analysis

[3] MFCC stands for Mel-Frequency Cepstral Coefficients. MFCC features are well-known features that closely related to the timbre aspect of audio files.

[4] A MATLAB implementation of CENS features is part of the Chroma Toolbox, which is freely available at http://www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/

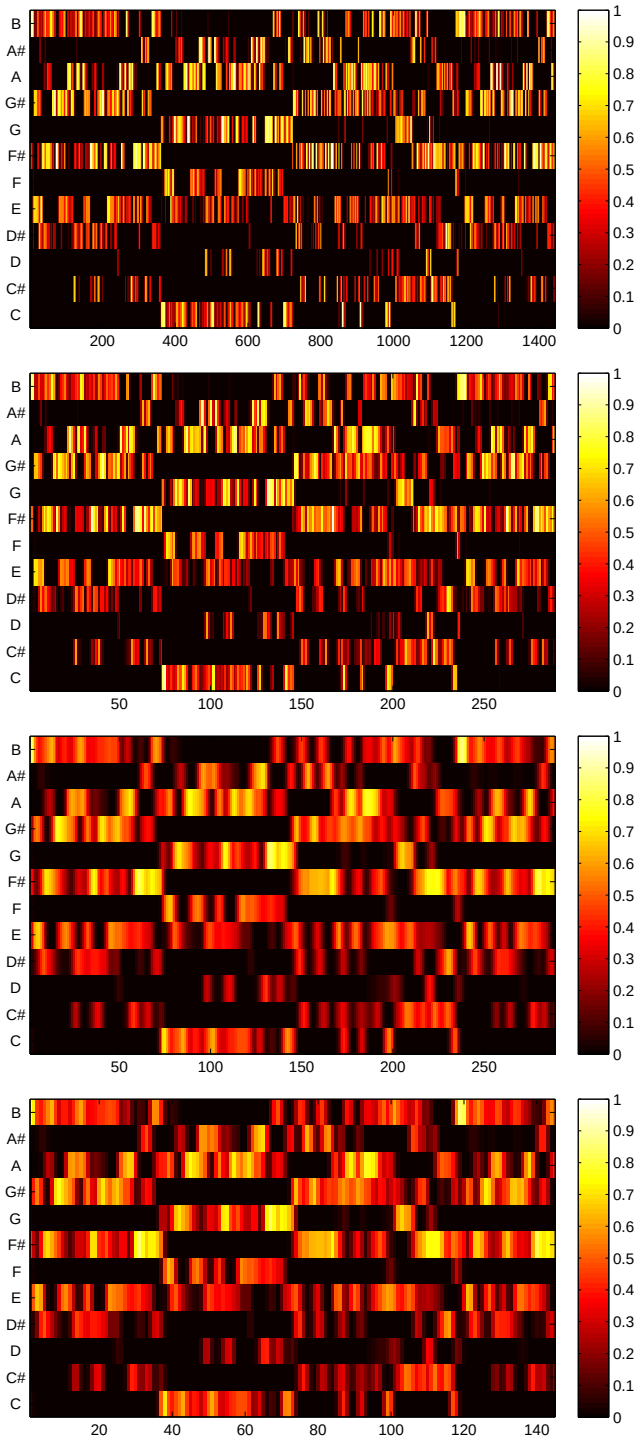

Figure 2.2: CENS features computed from different parameter settings for the same audio recording. **(a)** `CENS`(1,1), which stands for `CENS` features computed using the window length parameter $w = 1$, and downsampling parameter $d = 1$. We use similar notation of parameters also for the following figures. **(b)** `CENS`(11,5) **(c)** `CENS`(41,5) **(d)** `CENS`(41,10)

window and no downsampling is performed. In this way, the CENS$(1,1)$ features which are extracted from the example audio recording will result in 1420 feature frames, with each frame corresponds to 0.1 second of the audio recording. This is clearly illustrated in Figure 2.2a. Since each feature with such fine resolution focuses on too short audio information, it is not suitable to use them to reflect properties of long structural section. Therefore, in the following figures, we adjust the parameter settings to make the features suitable for music structure analysis. In Figure 2.2b, by setting $w = 11$, each feature takes 11 frames in its analysis window and therefore corresponds to 1.1 second of audio information. By setting $d = 5$ we downsample the features to a factor of 5 and get the resulting feature resolution as 2 Hz. This is also the fixed setting of CENS features for computing the similarity matrices in Figure 2.3. In addition, for comparison purpose, we also illustrate CENS(41,5) in Figure 2.2c, and CENS(41,10) in Figure 2.2d. Although we do not use them in the similarity matrix computation of the example audio recording, these settings are rather suitable for recordings of longer duration, e. g., symphonies which have a duration of one hour or even longer.

2.2 Matrix Enhancement

In this section, we give an overview of the various enhancement strategies or similarity matrices. Even though all enhancement strategies are implemented for general similarity matrices, we consider in the following only the case of self-similarity matrices for the sake of simplicity. As illustration, Figure 2.3 shows various variants of self-similarity matrices for the audio recording we mentioned in Section 2.1. In the next sub-sections, we will introduce how these self-similarity matrices are computed and also emphasize the enhancement strategies that are illustrated by these figures.

2.2.1 Similarity Measure

As mentioned in the introduction, one requires a notion of similarity (or dissimilarity, distance, cost) for a quantitative comparison of two elements $x, y \in \mathcal{F}$. In the case of $\mathcal{F} = \mathbb{R}^D$ being a Euclidean space of dimension D, typical measures are based on the ℓ^p-norm defined by $\|x\|_p = \big(\sum_{i=1}^{D} |x(i)|^p\big)^{1/p}$ for a vector $x = (x(1), x(2), \ldots, x(D))^T$. Then, for example, a similarity measure $\mathbf{s}$ may be obtained by setting $\mathbf{s}(x, y) = a - \|x - y\|_p^b$ for constants $a \in \mathbb{R}$ and $b \in \mathbb{N}$. In the following, we only consider the case $p = 2$ and assume that x and y are normalized with respect to this norm. Then, using $a = 2$ and $b = 2$, the measure $\mathbf{s}$ boils down to the inner product $\langle x|y\rangle$ (up to a factor of two), which measures the cosine of the angle between x and y. Figure 2.3a shows the self similarity matrix computed using cosine similarity measure for the feature sequence CENS$(11, 5)$ (see Figure 2.2b) of the example audio recording.

2.2.2 Smoothing

One important property of similarity matrices is the appearance of paths of high similarity that are parallel to the main diagonal [108, 112]. Each such path encodes the similarity

of two segments that are obtained by projecting the path onto the horizontal and vertical axes, respectively. The identification and extraction of such paths is the main step in many music analysis applications. However, due to musical and acoustic variations, the path structure is often very noisy and hard to extract.

To some extent, such noise can be reduced simply by using longer analysis windows in the feature computation step and adjusting the feature rate. To further enhance the path structure, one general strategy is to apply some kind of smoothing filter along the direction of the main diagonal, resulting in an emphasis of diagonal information in $\mathcal{S}$ and a denoising of other structures, see [11, 97, 109, 121] and Figure 2.3b. Such a filtering process is closely related to the concept of *time-delay embedding*, which has been widely used for the analysis of dynamical systems [80]. A simple filtering along the main diagonal only works well if there are no relative tempo differences between the segments to be compared. However, this assumption is often violated for music, where a part may be repeated with a faster or slower tempo. To deal with such tempo difference, a multiple filtering approach has been suggested in [97], where a similarity matrix is filtered along various directions that lie in a neighborhood of the direction defined by the main diagonal. Each such direction corresponds to a tempo difference and results in a separate filtered similarity matrix. The final similarity matrix is obtained by taking the cell-wise maximum over all these matrices. In this way, the path structure is also enhanced in the presence of local tempo variations as illustrated in Figure 2.3c.

In our implementation, we have simulated the multiple filtering approach by an efficient procedure that is based on a combination of feature and matrix resampling steps and simple diagonal smoothing. All operations can be expressed by full matrix operations, which are efficiently realized in MATLAB. Two main parameters are provided for controlling the smoothing quality: a smoothing length parameter ℓ and discrete set Θ of relative tempo differences, see Section 2.3 for more explanations.

The implemented smoothing filter is realized to smooth in the forward direction, which results in a fading out of the paths in particular when using a large length parameter. To avoid this fading out, one can use a forward-backward option, which applies the filter also in backward direction. The final similarity matrix is then obtained by taking the cell-wise maximum over the forward-smoothed and backward-smoothed matrices, see Figure 2.3d.

2.2.3 Transposition Invariance

It is often the case that certain musical parts are repeated in a transposed form as the part A_2 in our example. Such transpositions can be simulated by cyclically shifting chroma vectors [45]. In [89] this idea was used to construct *transposition-invariant* similarity matrices. To this end, one chroma feature sequence is left unaltered whereas the other chroma feature sequence is cyclically shifted along the chroma dimension in the twelve possible ways. Then, for each shifted version, a similarity matrix is computed, and the final similarity matrix is obtained by taking the cell-wise maximum over the twelve matrices. In this way, the repetitive structure is revealed even in the presence of key transpositions (Figure 2.3e). Furthermore, storing the maximizing shift index for each cell results in another matrix referred to as *transposition index matrix*, which displays the harmonic

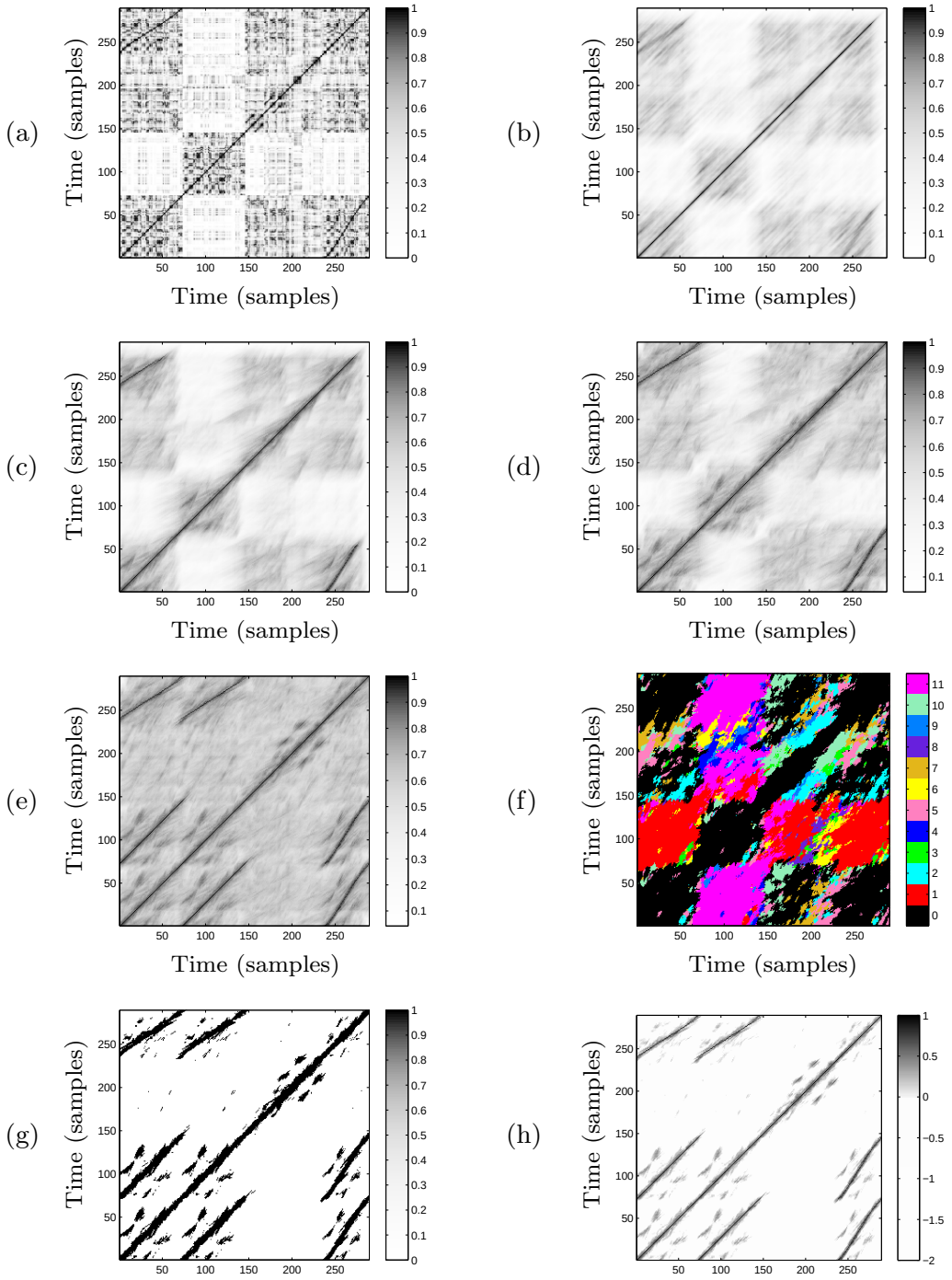

Figure 2.3: Variants of similarity matrices for the same audio recording. The figures are generated using the code shown in Table 2.2. **(a)** Original SSM using `CENS` features of 2 Hz resolution. **(b)** SSM after applying diagonal smoothing. **(c)** SSM after applying tempo-invariant smoothing. **(d)** SSM after appyling forward-backward smoothing. **(e)** Transposition-invariant SSM. **(f)** Transposition index matrix. **(g)** SSM after thresholding and binarization. **(h)** SSM after thresholding, scaling, and applying a penalty parameter.

relations within the music recording (Figure 2.3f). For example, this matrix reveals that the A_2-segment is indeed transposed by one semitone upwards relative to the A_1-segment.

In our implementation, we have provided a parameter Γ for specifying the chroma indices to be considered in the cyclic shifts. For example, $\Gamma = [0]$ leads to the original similarity matrix, whereas $\Gamma = [0:11]$ leads to the transposition-invariant version. At this point, we want to note that introducing transposition-invariance by cell-wise maximization over several matrices may increase the noise-level in the resulting similarity matrix. Therefore, the transposition-invariant matrix should be computed on the basis of smoothed matrices, since the smoothing typically goes along with a suppression of unwanted noise.

2.2.4 Thresholding

In many music analysis applications, similarity matrices are further processed by suppressing all values that fall below a given threshold. On the one hand, such a step often leads to a substantial reduction of the noise while leaving only the most significant structures. On the other hand, weaker but still relevant information may be lost. Actually, the thresholding strategy may have a significant impact on the final results and has to be carefully chosen in the context of the considered application.

In its simplest form, one can apply a global thresholding strategy. In our implementation, providing a threshold parameter $\tau > 0$, all values $\mathcal{S}(n,m)$ of a given similarity matrix $\mathcal{S}$ below τ are set to zero. Also binarization of the similarity matrix can be applied by setting all values above the threshold to one and all others to zero, see Figure 2.3g. Instead of binarization, one may perform a scaling where the range $[\tau : \mu]$ is linearly scaled to $[0:1]$ in the case that $\mu = \max_{n,m}\{\mathcal{S}(n,m)\} > \tau$, otherwise all entries are set to zero. Sometimes it may be beneficial to introduce an additional penalty parameter $\delta \leq 0$ and setting all original values below the threshold to the value δ. The global threshold τ can be chosen also in a relative fashion using the parameter ρ by keeping $\rho \cdot 100\%$ of the cells having the highest value, see (Figure 2.3h). Finally, as described in [121], thresholding can also be performed using a more local strategy by thresholding in a column- and row-wise fashion. To this end, for each cell (n,m) the value $\mathcal{S}(n,m)$ is kept if it is among the $\rho \cdot 100\%$ of the largest cells in row n and at the same time among the $\rho \cdot 100\%$ of the largest cells in column m, all other values are set to zero.

2.3 Toolbox

In this section, we describe the functions of our SM toolbox. The matrix enhancement components as described in Section 2.2 form the core of the SM toolbox, which is freely available at the website [124] under a GNU-GPL license. Table 2.1 gives an overview of the main MATLAB functions along with the most important parameters. Note that there are many more parameters and additional functions not discussed in this chapter.

To demonstrate how our toolbox can be applied, we now discuss the code example shown in Table 2.2, which is also contained in the toolbox as function demoSMtoolbox.m. Our example starts in lines 1–8 with computing a suitable chroma-based feature representation

Filename	Main parameters	Description
wav_to_audio.m	–	Import of WAV files and conversion to expected audio format.
audio_to_pitch_via_FB.m	winLenSTMSP	Extraction of pitch features from audio data.
pitch_to_CENS.m	winLenSmooth $\widehat{=}$ w, downsampSmooth $\widehat{=}$ d	Derivation of `CENS` features from `Pitch` features.
features_to_SM	smoothLenSM $\widehat{=}$ ℓ, forwardBackward	Smoothing of SM.
	(tempoRelMin, tempoRelMax, tempoNum) $\widehat{=}$ Θ	Application of tempo invariance.
	circShift $\widehat{=}$ Γ	Application of transposition invariance.
threshSM	threshTechnique, threshValue $\widehat{=}$ τ or $\widehat{=}$ ρ	Application of different thresholding techniques.
	penality $\widehat{=}$ δ, applyBinarize, applyScale	Application of binarization or scaling.
visualizeSM	colormapPreset, print, figureName, imageRange	Visualiation of similarity matrix.
visualizeTransIndex	colormapPreset, print, figureName	Visualization of transposition index matrix.
makePlotPlayable	featureRate, fs	Synchronized playback of audio file along with a plotted figure.

Table 2.1: Overview of the main MATLAB functions contained in SM toolbox [124] and the most important parameters. The first three functions for feature extraction are contained in the Chroma Toolbox [90].

for the given audio recording. The used functions are part of the Chroma Toolbox [90]. Note that these features only serve as an example and any other feature representation may be used equally well in the following steps. The call to the function wav_to_audio, which is a simple wrapper around MATLAB's wavread.m, converts the input WAV file into a mono version at a sampling rate of 22050 Hz. Next, `Pitch` and `CENS` features are computed, where the struct paramPitch and the struct paramCENS are used to pass optional parameters to the feature extraction function. If some parameters or the whole struct are not set manually, then meaningful default settings are used. This is a general principle, which applies for the chroma toolbox as well as for the SM toolbox. In the current settings, the resulting `CENS` features have a feature resolution of 2 Hz, see [90] for details.

In lines 10–49, the various self-similarity matrices as shown in Figure 2.3 are computed, where different parameter settings that are encoded by the struct paramSM are used. First, in line 10 a self-similarity matrix S is computed by comparing f_CENS with itself. Note that one may also input two different feature sequences resulting in a more general similarity matrix. Furthermore, note that no parameters are specified in the function call features_to_SM. As a result, the function-internal default settings are used, where a simple inner product is used as similarity measure and no matrix enhancement is applied. The matrix is visualized by the function visualizeSM (line 12) using a colormap that is specified by the parameter paramVis.colormapPreset (line 11), see Figure 2.3a.

Next, various enhancement strategies are activated by setting the corresponding parameter values. In line 14, the smoothing length parameter ℓ is set to 20 (given in feature samples), which corresponds to 10 s of the original audio when using a feature rate of 2 Hz, see Figure 2.3b. In lines 18–20, the discrete set Θ used for tempo-invariant smoothing is defined by three different parameters: tempoRelMin specifies the minimal relative tempo difference contained in Θ, tempoRelMax the maximal relative tempo difference contained

```
clear;close all;
filename='Test_AABA.wav';
f_audio=wav_to_audio('','data_music/',filename);
paramPitch.winLenSTMSP=4410;
f_pitch=audio_to_pitch_via_FB(f_audio,paramPitch);
paramCENS.winLenSmooth=11;
paramCENS.downsampSmooth=5;
f_CENS=pitch_to_CENS(f_pitch,paramCENS);

S=features_to_SM(f_CENS,f_CENS);
paramVis.colormapPreset=2;
visualizeSM(S,paramVis);

paramSM.smoothLenSM=20;
S=features_to_SM(f_CENS,f_CENS,paramSM);
visualize_SM(S,paramVis);

paramSM.tempoRelMin=0.5;
paramSM.tempoRelMax=2;
paramSM.tempoNum=7;
S=features_to_SM(f_CENS,f_CENS,paramSM);
visualizeSM(S,paramVis);

paramSM.forwardBackward=1;
S=features_to_SM(f_CENS,f_CENS,paramSM);
visualizeSM(S,paramVis);

paramSM.circShift=[0:11];
[S,I]=features_to_SM(f_CENS,f_CENS,paramSM);
visualizeSM(S,paramVis);
visualizeTransIndex(I);

paramThres.threshTechnique=1;
paramThres.threshValue=0.75;
paramThres.applyBinarize=1;
S_thres=threshSM(S,paramThres);
visualizeSM(S_thres,paramVis);

paramThres.threshTechnique=2;
paramThres.threshValue=0.15;
paramThres.applyBinarize=0;
paramThres.applyScale=1;
paramThres.penalty=-2;
S_final=threshSM(S,paramThres);
paramVis.imageRange=[-2,1];
paramVis.colormapPreset=3;
paramVis.print=1;
paramVis.figureName='SM_final';
handleFigure=visualizeSM(S_final,paramVis);

parameterMPP.fs=22050;
parameterMPP.featureRate=2;
makePlotPlayable(f_audio,handleFigure,parameterMPP);
```

Table 2.2: Code example generating matrices shown in Figure 2.3.

in Θ, and tempoNum the actual number of elements of Θ (using a logarithmic sampling between tempoRelMin and tempoRelMax for the intermediate values). The resulting matrix is again visualized by line 22, see Figure 2.3c. In line 24, the forward-backward smoothing is activated, see Figure 2.3d. Then, in line 28, the transposition-invariance using all twelve possible cyclic chroma shifts is activated. In line 25, the self-similarity matrix S as well as the transposition index matrix I are returned and visualized in the next two lines, see Figure 2.3e/f. Finally, the similarity matrix is further processed by applying the thresholding function threshSM. Various thresholding strategies specified by the parameter threshTechnique are available. In lines 33–36, a simple global thresholding (threshTechnique=1) using an absolute threshold $\tau = 0.75$ (threshValue=0.75) and binarization (applyBinarize=1) is applied, see Figure 2.3g. Similarly, in lines 39–44, relative thresholding (threshTechnique= 2) with $\rho = 0.15$ (threshValue=0.15) as well as with scaling and penalty is applied, see Figure 2.3h. Note that depending on the thresholding technique, the parameter threshValue is interpreted either as absolute threshold or as relative threshold. Finally, note that further parameters can be specified for the visualization function including a print option to save the generated figure as .eps file, see lines 45–49.

Besides these main functions, our toolbox also offered code for audio thumbnailing application, which we we detailed introduce in Section 3.8. Furthermore, the toolbox contains additional demo files for more complex audio recordings.

2.4 Further Notes

As we discussed before, many variants of similarity matrices based on different features, similarity measures, and enhancements have been suggested in the MIR literature for analyzing, comparing, structuring, and retrieving audio material. At this point, we want to emphasize that there is no single variant that works best in all situations and the requirements of the used similarity matrix very much depends on the specific application in mind. For example, for many tasks related to cover song identification [119] or audio structure analysis [30, 45, 108], audio-based chroma features at a feature resolution of roughly 2 Hz have turned out to be a meaningful choice. Obviously, such resolutions are much too coarse when considering tasks such as high-resolution music synchronization [35]. When considering segmentation and classification tasks based on, e. g., timbre rather than harmony, one needs to use different features such as MFCCs [36]. Also the smoothing variant very much depends on the application. As noted in [108], similarity matrices typically contain path-like structures (accounting for repetition-based properties) and block-like structures (accounting for homogeneity-based properties). The smoothing variants discussed in this chapter enhance path-like structures, but destroy block-like structures. This is not always wanted. For example, when performing homogeneity-based structure analysis [61, 71], one requires different smoothing techniques that enhance the block structure. Generally speaking, smoothing decreases the noise level in similarity matrices, thus introducing additional robustness to the overall procedure. On the downside, valuable structural information may be smoothed out and lost for the subsequent analysis. Another quite obvious but important remark is that one should only apply enhancement strategies only when they are actually needed. For example, when performing structure analysis for music with constant tempo (which is often the case for popular music) a simple diagonal smoothing

may perform already well and tempo-invariance is not needed. Actually, applying tempo-invariance in this situation may even introduce unwanted artifacts. Similarly, if one does not expect any modulations, transposition-invariance should not be used. The reason is that achieving invariance also comes at some cost. For example, computing tempo- and transposition-invariant similarity matrices as done in our SM toolbox, the noise of the individual matrices may penetrate to the final matrix obtained by maximization, which may increase the overall noise level. Thus, such strategies need to be applied with care and also the order in which enhancement strategies are applied may have a significant impact on the final results.

As a general goal of this chapter, we want to raise the awareness of such issues. Also, providing cleaned-up example code, we hope that our toolbox may inspire future research in music information retrieval and may serve as illustrative material in education.

Chapter 3

Fitness Measure for Capturing Repetitions

In this chapter, we introduce how we capture repetitions in a given music recording. The content of this chapter forms the basis of this thesis and the material closely follows the publications [95] and [91].

Music is highly structured and based on different principles such as repetition, contrast, variation, and homogeneity to create certain relationships between notes, melodies, chords, harmonies, or rhythms. The automated detection of such relations, a task closely related to audio structure analysis, constitutes a fundamental research topic within the area of music information retrieval. One major goal of structure analysis is to divide a music recording into temporal segments and to group these segments into musically meaningful categories [108]. Such segments may refer to chorus sections of a piece of popular music, to stanzas of a folk song, or to the first theme of a symphony. Such musical parts are often characterized by the fact that they are repeated several times throughout the piece. Actually, finding the repetitive structure of a music recording has been a well-studied problem over the last years, see, e. g., [11, 29, 31, 45, 77, 88, 98, 107, 109, 114, 115, 122, 137] and the overview articles [30, 108].

Most of these approaches work well for music in which the repetitions largely agree. However, in music performances, musical parts are rarely repeated in precisely the same way. For example, the repeated verses of a popular song typically share the same melody but differ in terms of the underlying lyrics. Furthermore, a verse may be repeated instrumentally, with the soloist deviating from the original verse by freely improvising the melody. Also, a verse may be repeated in a transposed form, with all notes being shifted, for example, one semitone upwards. The situation becomes even more complex for amateur recordings where non-professional singers often have severe intonation problems and deviate significantly from the expected pitches. For genres such as classical music, the main melody may be repeated by different instruments with changing accompaniment and in different keys. Also, repeated parts may show significant differences in tempo. In summary, audio segments that are considered as repetitions may differ significantly in such aspects as dynamics, instrumentation, articulation, and tempo, not to speak of pronounced musical variations. In such cases, structure analysis becomes a hard and ill-posed task with many

problems still remaining unsolved.

In this chapter, we focus on capturing repetitive segments from a given recording. In particular, we concentrate on a problem which aimed at finding the most representative and repetitive segment of a given music recording, a task often referred to as *audio thumbnailing*, see, e. g., [11, 19, 28, 72]. Typically, such a segment should have many (approximate) repetitions, and these repetitions should cover large parts of the recording. As the main technical contribution of this chapter, we introduce a fitness measure that assigns a fitness value to each audio segment that simultaneously captures two aspects. First, it indicates *how well* the given segment explains other related segments and, second, it indicates *how much* of the overall music recording is covered by all these related segments. Similar to [28, 109], the audio thumbnail is then defined to be the segment having maximal fitness. In the computation of the fitness measure, one important conceptual idea of our approach is to avoid hard decisions and error-prone steps in an early stage of the algorithmic pipeline. In particular, as opposed to previous approaches, we introduce an optimization scheme that jointly performs path structure extraction and grouping—two error-prone steps that are usually performed successively. As a result, we obtain a robust procedure that can detect repetitive elements even in the presence of strong musical variations. We also describe an efficient algorithm based on dynamic programming for computing the fitness measure. As a further contribution of this chapter, we introduce the concept of a scape plot representation that shows the fitness values over all possible audio segments. A visualization of this fitness scape plot yields a compact high-level view on the structural properties of the entire music recording. Finally, we present experiments based on different audio collections comprising popular music as well as classical music. In combination with enhanced feature representations, we show that our fitness measure can even cope with strong variations in tempo, instrumentation, and transpositions that may occur within and across related segments. By discussing several explicit examples, we indicate the strengths as well as the limitations of our approach.

The remainder of this chapter is organized as follows. In Section 3.1, we discuss basic knowledge of repetition-based structure analysis and specify some notation. Then, in Section 3.2, we briefly summarize the general concept of self-similarity matrices and emphasize various enhancement strategies that are necessary. Section 3.3 contains the main contributions of this chapter, where we give a detailed technical description of the fitness measure. Then, in Section 3.4 we explain how we solve the audio thumbnailing problem by applying our fitness measure. As a second contribution, we introduce a scape plot visualization that indicates the fitness values of all possible audio segments in Section 3.5. Next, we discuss basic properties of our fitness concept and offer a number of illustrative examples in Section 3.6. Our experiments are then described in Section 3.7. Finally, we provide the implementation of fitness measure and audio thumbnailing in Section 3.8, and further conclude remarks and an outlook on future in Section 3.9.

3.1 Background

In this chapter, we contribute to repetition-based music structure analysis using chroma-based audio features that correlate to aspects of harmony and melody. In particular, we

extend and improve on a sequence approach as introduced in [109]. In the remainder of this section, we summarize the main principles of repetition-based audio structure analysis, introduce some general notation, and discuss in more detail the relation of our approach to previous work.

In the following, we distinguish between a piece of music (in an abstract sense) and a particular audio recording (an actual performance) of the piece. The term *part* is used in the context of the abstract music domain, whereas the term *segment* is used for the audio domain [108]. Musical parts are often denoted by the letters $A, B, C, \ldots$ in the order of their first occurrence, where indices are used to indicate repetitions. For example, the sequence $A_1A_2B_1B_2CA_3B_3B_4$ describes the *musical form* of the Hungarian Dance No. 5 by Johannes Brahms. The structure of this piece, as given as a recording by Ormandy and serving as our running example, is shown in Figure 3.1. The musical form consists of three repeating A-parts, four repeating B-parts, as well as a C-part. Hence, given the Ormandy recording, the goal of the structure analysis problem considered in this chapter is to find the segments within the recording that correspond to the A-part or to the B-part.

Most approaches to repetition-based structure analysis proceed as follows. First, the music recording is transformed into a sequence $X := (x_1, x_2, \ldots, x_N)$ of suitable feature vectors $x_n \in \mathcal{F}$, $n \in [1:N] := \{1, 2, \ldots, N\}$, where $\mathcal{F}$ denotes a suitable feature space (see Chapter 2, Section 2.1). In the following, a *segment* α is understood to be a subset $\alpha = [s:t] \subseteq [1:N]$ specified by its start point s and its end point t. Based on a similarity measure $\mathbf{s} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$, one obtains a self-similarity matrix (SSM) denoted by $\mathcal{S} \in \mathbb{R}^{N \times N}$ and defined by $\mathcal{S}(n,m) := \mathbf{s}(x_n, x_m)$, $1 \leq n, m \leq N$. In the following, a tuple $(n,m) \in [1:N]^2$ is called a *cell* of $\mathcal{S}$, and the value $\mathcal{S}(n,m)$ is referred to as the *score* of the cell (n,m). The score value $\mathcal{S}(n,m)$ is high (dark color in Figure 3.1) if the two feature vectors x_n and x_m are similar, otherwise $\mathcal{S}(n,m)$ is low (light color).

As we discussed in the previous related work section, the crucial observation is that repeating patterns in the feature sequence X appear as parallel "stripes" in $\mathcal{S}$, see [29, 108]. More precisely, these stripes are encoded by paths of cells of high score running roughly in parallel to the main diagonal. Each of the paths encodes the similarity of a pair of segments, which are given by the two projections of the path onto the vertical and horizontal axis of $\mathcal{S}$, respectively. This is also illustrated by Figure 3.1, where two such paths are highlighted within $\mathcal{S}$. One of the paths encodes the similarity between the audio segments corresponding to A_1 and A_2, and the other path encodes the similarity between the segments corresponding to B_1 and B_2. Note that the first path is exactly parallel to the main diagonal, indicating that the parts A_1 and A_2 are played in the same tempo, whereas the second path is curved, indicating that the parts B_1 and B_2 are played in different tempi. In fact, in the Ormandy interpretation, the B_2-part is played much faster than the B_1-part. This fact is also revealed by the gradient of the path, which encodes the relative tempo difference.

To identify repetitions, most approaches extract the path structure from an SSM and apply a clustering step to the pairwise relations obtained from the paths in order to derive entire groups of mutually similar segments. For example, one group contains all A-part segments, another all B-part segments. This step can be considered as forming some kind of transitive closure of the path relations [30, 31, 88]. However, note that strong musical

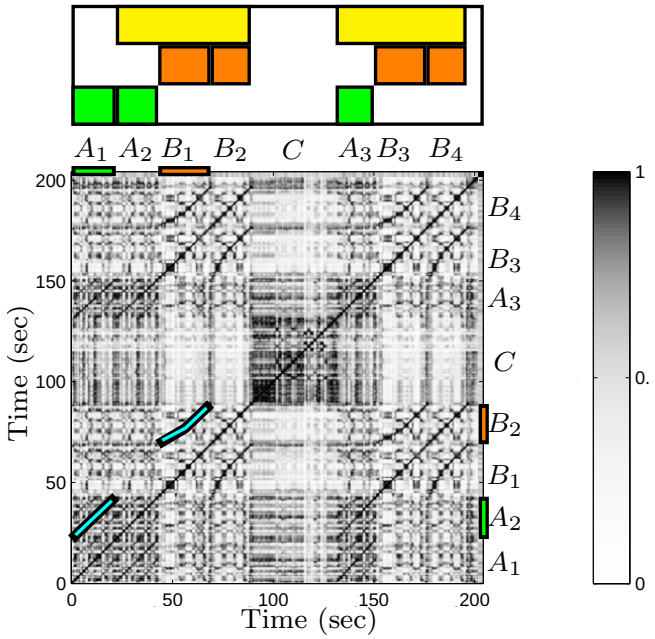

Figure 3.1: Musical form and self-similarity matrix of a recording by Ormandy of Brahms' Hungarian Dance No. 5.

and acoustic variations may lead to rather noisy and fragmented path structures, which makes both steps—path extraction as well as grouping—error-prone and fragile. In [45], a grouping process is described that balances out inconsistencies in the path relations by exploiting a constant tempo assumption. However, when dealing with varying tempo, the grouping process constitutes a challenging research problem [31, 98].

As opposed to previous approaches, the idea of our approach is to jointly perform the path extracting and grouping step. We realize this idea by assigning a fitness value to a given segment in such a way that all existing relations within the entire recording are simultaneously accounted for. In other words, instead of extracting individual paths, we extract entire groups of paths, whereby consistency properties within a group are automatically enforced by our construction.

The general idea of assigning some kind of fitness value for each segment of the audio recording is not new and has already been formulated by Cooper and Foote [28]. In this early work, the authors calculate the fitness of a given segment as the normalized sum of the self-similarity between the segment and the entire recording, which can be thought of as some sort of "summary score." The thumbnail is then defined to be the fitness-maximizing segment. Also, a visualization of the fitness over all possible segments has been indicated in [28]. As one main limitation, the fitness measure does not take any path relations into account, thus yielding only limited information on the repetitiveness of a segment.

To capture the repetitive structure, Peeters [109] has introduced a fitness measure referred to as "likelihood." To compute this measure, a binary-valued diagonal path structure is

extracted from an SSM. Then, for a given segment (called the "candidate mother segment"), the likelihood is defined as the sum of the lengths of all segments explained by diagonal paths over the candidate mother segment, where overlaps between repeating instances are prevented by suitable constraints. Finally, the thumbnail (referred to as the "mother segment") is defined as the candidate mother segment of maximal likelihood.

Our fitness measure builds upon and extends the pioneering work described in [28, 109] in various ways. First, instead of an explicit extraction of the binary-valued path structure, we avoid such a hard decision by working on a real-valued SSM (even though we also apply some thresholding for denoising purposes). Second, using a variant of dynamic time warping instead of looking for diagonal paths, our approach allows for handling tempo differences between repeating segments. Third, we combine a coverage criterion (which is similar to the likelihood in [109]) with a score criterion to define a fitness measure that balances out two contradicting principles (large coverage versus high average score). Fourth, introducing suitable normalization steps, where we disregard trivial self-explanations similar to [81], our fitness measure is well-suited to compare repetition properties of segments of different lengths. Finally, we introduce a unifying mathematical framework and an optimization scheme based on dynamic programming to efficiently compute the fitness measure.

We would like to point out that our work has also been inspired by Paulus and Klapuri [107], even though the task and concepts of this chapter are fundamentally different from [107]. The fitness measure introduced in [107] expresses properties of an *entire structure*, whereas our fitness measure expresses properties of a *single segment*.

3.2 Self-Similarity Matrices

The degree of similarity between two repeating segments crucially depends on the feature type, the similarity measure, and on how the self-similarity matrix $\mathcal{S}$ is defined and post-processed. In this section, we describe the structure-enhanced and transposition-invariant SSM used in our experiments. Since the construction of the SSM is not the focus of this chapter, we only give a brief description and revert to existing literature, see also Figure 3.2 for an overview. For the detail implementation and more comprehensive examples, we refer to Chapter 2. From a technical point of view, our fitness measure is generic in the sense that it works with general self-similarity matrices that fulfill the normalization properties $\mathcal{S}(n,m) \leq 1$ for $n,m \in [1:N]$ and $\mathcal{S}(n,n) = 1$ for $n \in [1:N]$. Note that so far we only set the upper bound of similarity values to be one, but do not set the lower bound values. This is because the lower bound are further decided by one of the matrix enhancement strategies where we include proper negative penalty values in the $\mathcal{S}$.

First of all, we convert the audio signal into twelve-dimensional chroma-based audio features, which closely correlate to the aspect of harmony and have become a widely used tool in processing and analyzing music data [11, 39, 88, 90, 97, 109, 119]. In our experiments, we use a chroma variant referred to as `CENS` [88, Section 3.3] with a feature rate of 2 Hz (two features per second). This resolution has turned out to be suitable not only for audio structure analysis [98], but also for related tasks such as cover song identification [119] and audio matching [64]. Normalizing the features, we use the inner product as a similarity

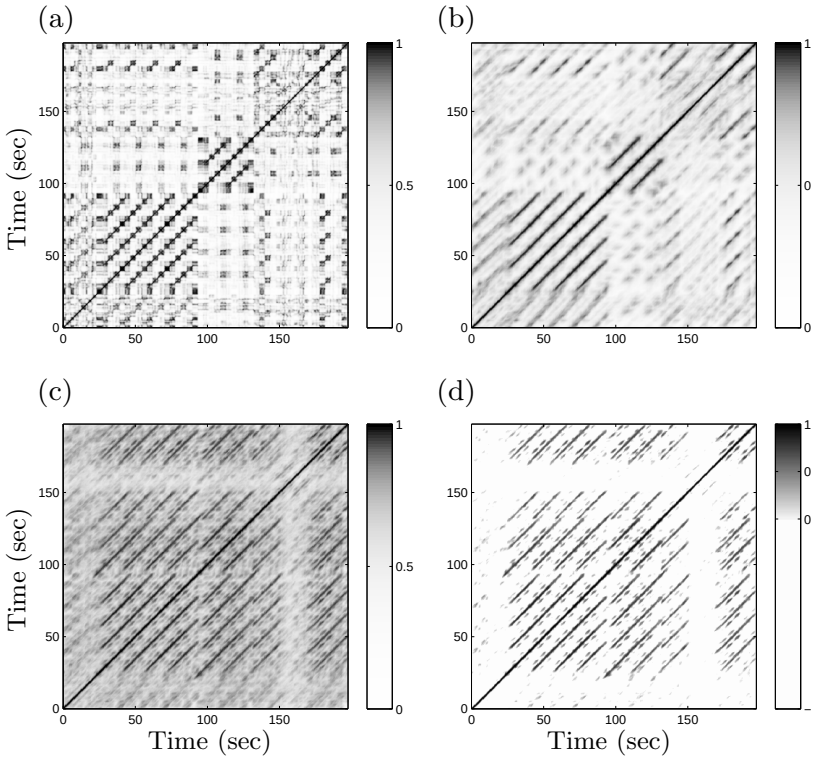

Figure 3.2: Self-similarity matrices for the song "In the year 2525" by Zager and Evans, which has several transposed repetitions in its second half. **(a)** Initial self-similarity matrix. **(b)** Path-enhanced matrix. **(c)** Transposition-invariant matrix. **(d)** Thresholded matrix with $\delta = -2$.

measure to compute a self-similarity matrix $\mathcal{S}$, see Figure 3.2a.

To further enhance the path structure of $\mathcal{S}$[1], one typical procedure is to apply some kind of smoothing filter along the direction of the main diagonal, resulting in an emphasis of diagonal information in $\mathcal{S}$ and a denoising of other structures, see [11, 97, 109, 121] and Figure 3.2b for an illustration. In our implementation, we use a smoothing variant as described in [97], which can deal with local tempo variations. Furthermore, to account for transpositions between related segments, we adopt the concept of *transposition-invariant* SSMs as introduced in [89]. The idea is to first compute the similarity between the original feature sequence and each of the twelve cyclically shifted versions of the chroma feature sequence [45], resulting in twelve similarity matrices. Then, the transposition-invariant SSM is calculated by taking the point-wise maximum over these twelve matrices, see Figure 3.2c.

In view of the subsequent application, we further process the SSM by suppressing all values that fall below a given threshold. Using a suitable threshold parameter $\tau > 0$ and a penalty parameter $\delta \leq 0$, we first set the score values of all cells with a score below τ to the value δ and then linearly scale the range $[\tau : 1]$ to $[0 : 1]$, see Figure 3.2d. The

[1] All the enhancement strategies of a similarity matrix mentioned in this chapter can be found at http://www.audiolabs-erlangen.de/resources/MIR/SMtoolbox

thresholding introduces some kind of denoising, whereas the parameter δ imposes some penalty on all cells of low score. Intuitively, we want the relevant path structure to lie in the positive part of the resulting SSM, whereas all other cells are given a negative score. Finally, to enforce the normalization properties needed in our fitness construction, we set $\mathcal{S}(n,n) = 1$ for $n \in [1:N]$ (this property may have been lost in the smoothing process due to boundary effects). Note that different methods can be used for thresholding [121]. In the following, we choose the threshold in a relative fashion by keeping $\rho \cdot 100\%$ of the cells having the highest score and set $\delta = -2$. The role of ρ will be further investigated in Section 3.7.

3.3 Fitness Measure

In this section, we introduce and discuss our novel fitness measure. In assigning a fitness value to a given segment α, our idea is to simultaneously account for its relations to other segments of the audio recording. Extending the approach [109] based on diagonal paths, we introduce the concept of a path family over α, which allows for expressing repetitive relations even in the presence of tempo differences (Section 3.3.1). We then look for an optimal path family over α from among many possible path families. To compute such an optimal path family, we introduce an efficient algorithm based on dynamic programming (Section 3.3.2). Next, in Section 3.3.3, we explain how to assign a coverage value as well as an average score value to a given path family. The fitness of the segment α is then defined by the harmonic mean of coverage and score of the optimal path family over α.

3.3.1 Path Family

Let $X = (x_1, x_2, \ldots, x_N)$ be a feature sequence and let $\mathcal{S}$ be a self-similarity matrix as introduced in Section 3.2. Furthermore, let a segment be

$$\alpha = [s:t] \subseteq [1:N] \tag{3.1}$$

with $|\alpha| := t - s + 1$ denoting its length. For later usage, we define a *segment family* of size K to be a set

$$\mathcal{A} := \{\alpha_1, \alpha_2, \ldots, \alpha_K\} \tag{3.2}$$

of pairwise disjoint segments: $\alpha_k \cap \alpha_j = \emptyset$ for $k, j \in [1:K]$, $k \neq j$. Let $\gamma(\mathcal{A}) := \sum_{k=1}^{K} |\alpha_k|$ be the *coverage* of $\mathcal{A}$.

Next, a *path* over α of length L is a sequence

$$p = ((n_1, m_1), \ldots, (n_L, m_L)) \tag{3.3}$$

of cells $(n_\ell, m_\ell) \in [1:N]^2$, $\ell \in [1:L]$, satisfying $m_1 = s$ and $m_L = t$ (boundary condition) and $(n_{\ell+1}, m_{\ell+1}) - (n_\ell, m_\ell) \in \Omega$ (step size condition), where Ω denotes a set of admissible step sizes. In our setting, we use

$$\Omega = \{(1,2), (2,1), (1,1)\}, \tag{3.4}$$

which constrains the slope of a path within the bounds of $1/2$ and 2, see [88, Chapter 4].

For a path p, we associate two segments defined by the projections respectively as:

$$\pi_1(p) \quad := [n_1 : n_L] \tag{3.5}$$
$$\pi_2(p) \quad := [m_1 : m_L] \tag{3.6}$$

Note that the boundary condition enforces $\pi_2(p) = \alpha$. The other segment $\pi_1(p)$ is referred to as an *induced segment*, see Figure 3.3 for examples.

The *score* $\sigma(p)$ of p is defined as

$$\sigma(p) = \sum_{\ell=1}^{L} \mathcal{S}(n_\ell, m_\ell)\,. \tag{3.7}$$

Note that each path over the segment α encodes a relation between α and an induced segment, where the score $\sigma(p)$ yields a kind of quality measure for this relation.

Next, we introduce the concept of a *path family* over α, which is defined to be a set

$$\mathcal{P} := \{p_1, p_2, \ldots, p_K\} \tag{3.8}$$

of size K, consisting of paths p_k over α, $k \in [1:K]$.

Furthermore, as an additional condition, we require the induced segments to be pairwise disjoint or, in other words, the set $\{\pi_1(p_1), \ldots, \pi_1(p_K)\}$ to be a segment family. This condition ensures that there are no overlaps between related segments as also required in the approach by Peeters [109], see also Figure 3.3 for an illustration of this condition.

Next, extending the definition in Eq. (3.7), the *score* $\sigma(\mathcal{P})$ of the path family $\mathcal{P}$ is defined as

$$\sigma(\mathcal{P}) := \sum_{k=1}^{K} \sigma(p_k). \tag{3.9}$$

As is also indicated by Figure 3.3, there are in general a large number of possible path families over α. Among these path families, let

$$\mathcal{P}^* := \operatorname*{argmax}_{\mathcal{P}} \ \sigma(\mathcal{P}) \tag{3.10}$$

denote an optimal path family of maximal score. In the following, the family consisting of the segments induced by the paths of $\mathcal{P}^*$ will be referred to as the *induced segment family* (of $\mathcal{P}^*$ or of α). As will be shown in Section 3.3.2, the optimal path family $\mathcal{P}^*$ can be computed efficiently using dynamic programming. In Section 3.3.3, we explain how our fitness measure is derived from the score $\sigma(\mathcal{P}^*)$ and the induced segment family of $\mathcal{P}^*$.

3.3.2 Optimization Scheme

We now describe an efficient algorithm for computing an optimal path family for a given segment in a running time that is linear in the product of the length of the feature sequence

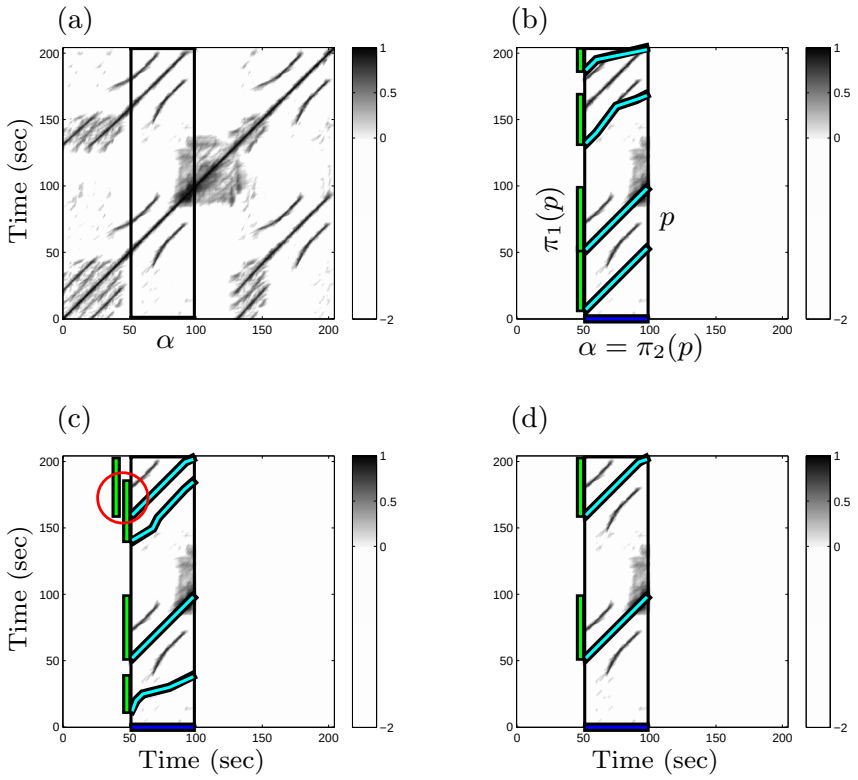

Figure 3.3: SSM of our Brahms example with various paths over the segment $\alpha = [50:100]$. The induced segments are indicated on the vertical axis. **(a)** SSM. **(b)** Paths forming a path family. **(c)** Paths not forming a path family (induced segments overlap). **(d)** Paths forming an optimal path family.

and the length of the segment. Our algorithm is based on a modification of classical dynamic time warping (DTW) as originally developed for speech processing [113] and also extensively used in music processing [52,88]. Given two sequences, say $X := (x_1, x_2, \ldots, x_N)$ and $Y := (y_1, y_2, \ldots, y_M)$, the objective of classical DTW is to compute an optimal path that *globally* aligns X and Y, where the first elements as well as the last elements of the two sequences are to be aligned. The step size condition as specified by the set Ω constraints the slope of the path. Furthermore, using $\Omega = \{(1,2),(2,1),(1,1)\}$, as in our case, each element of X is aligned to at most one element of Y (and vice versa).

Now, when computing an optimal path family over a given segment $\alpha = [s:t] \subseteq [1:N]$, $M := |\alpha|$, the role of Y is taken over by α, and the conditions change compared to classical DTW. In particular, α can be simultaneously aligned to several (non-overlapping) subsequences of X. However, for each such subsequence, the entire segment α is to be aligned. Furthermore, certain sections of X may be left completely unconsidered in the alignment. To account for these new constraints, we introduce additional steps that allow us to skip certain sections of X and to jump from the end to the beginning of the given segment α, see also Figure 3.4 for an illustration. First, we define the $N \times M$ submatrix

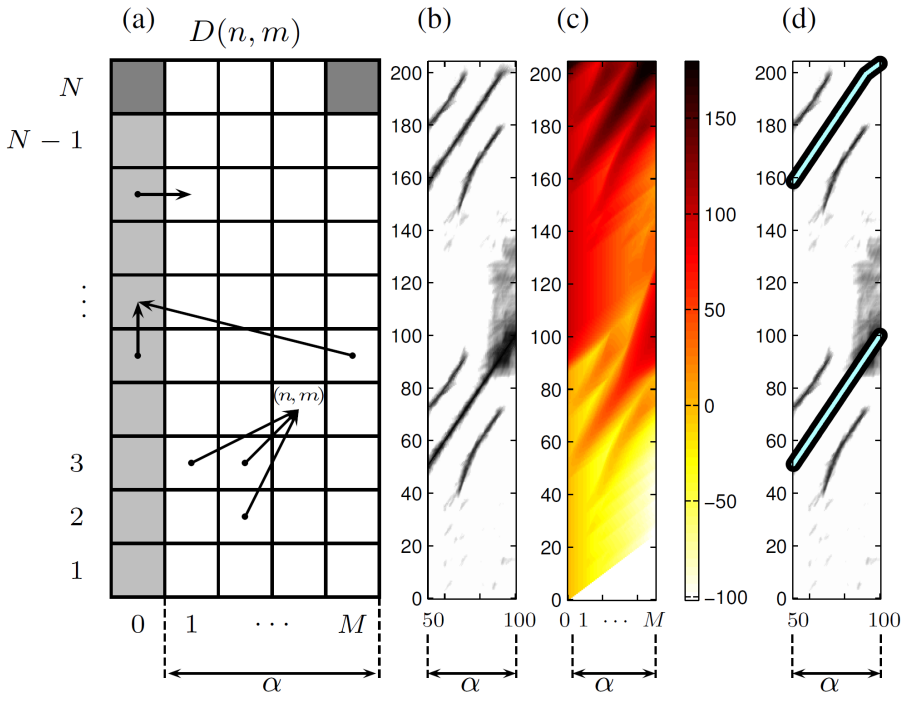

Figure 3.4: **(a)** Illustration of the various predecessors in computing the accumulated score matrix. **(b)** Submatrix $\mathcal{S}^\alpha$ with $\alpha = [50:100]$ of the SSM shown in Figure 3.3. **(c)** Accumulated score matrix D. **(d)** Optimal path family.

$\mathcal{S}^\alpha \in \mathbb{R}^{N,M}$ by taking the columns s to t of $\mathcal{S}$:

$$\mathcal{S}^\alpha(n,m) = \mathcal{S}(n, m+s-1) \tag{3.11}$$

for $n \in [1:N]$ and $m \in [1:M]$. Next, we define an accumulated score matrix $D \in \mathbb{R}^{N,M+1}$ (with rows indexed by $[1:N]$ and columns indexed by $[0:M]$), by the following recursion:

$$D(n,m) = \mathcal{S}^\alpha(n,m) + \max\{D(i,j) \mid (i,j) \in \Phi(n,m)\} \tag{3.12}$$

for $n \in [2:N]$ and $m \in [2:M]$, where $\Phi(n,m) = \{(n-i, m-j) \mid (i,j) \in \Omega\} \cap [1:N] \times [1:M]$ denotes the set of possible predecessors. So far, this is similar to the classical DTW algorithm. The constraint conditions and additional jump steps are realized by the definition of the values of D for the remaining index pairs (n,m) with $n = 1$ and $m \in \{0,1\}$. The first column of D indexed by $m = 0$, which plays a special role, is recursively defined by $D(1,0) = 0$ and

$$D(n,0) = \max\{D(n-1,0), D(n-1,M)\} \tag{3.13}$$

for $n \in [2:N]$. First, the term $D(n-1,0)$ enables the algorithm to move upwards without accumulating any (possibly negative) score, thus realizing the condition that sections of X may be skipped without penalty (negative score). Second, the term $D(n-1,M)$ closes up a path (ensuring that the entire segment α is aligned to a subsequence of X), while ensuring that the next possible segment does not overlap with the previous segment. The

second column of D indexed by $m = 1$ is defined by

$$D(n, 1) = D(n, 0) + \mathcal{S}^{\alpha}(n, 1) \tag{3.14}$$

for $n \in [1 : N]$, which starts a new path. Finally, to complete the initialization, we set $D(1, m) = -\infty$ for $m \in [2 : M]$, which forces the first path to start with the first element of α. The score of an optimal path family is then given by

$$\sigma(\mathcal{P}^*) = \max\{D(N, 0), D(N, M)\}. \tag{3.15}$$

The first term $D(N, 0)$ reflects the case that the final section of X may be skipped, and the second term $D(N, M)$ ensures that in the other case the entire segment α is aligned to a suffix of X. The associated optimal path family $\mathcal{P}^*$ can be constructed from D using a back-tracking algorithm as in classical DTW, see [88, Chapter 2] for details. As the only modification, the cells of $\mathcal{S}^{\alpha}$ that belong to the first auxiliary column (indexed by $m = 0$) are to be omitted to obtain the final path family. Obviously, the presented algorithm has a complexity (in terms of memory requirements and running time) of $O(MN)$.

3.3.3 Definition of Fitness Measure

We now give a formal definition of our fitness measure. At this point, we only assume that the given SSM $\mathcal{S} \in \mathbb{R}^{N \times N}$ has the property that $\mathcal{S}(n, m) \leq 1$ for all cells $(n, m) \in [1 : N]^2$, and $\mathcal{S}(n, n) = 1$ for $n \in [1 : N]$.

For the segment α, let $\mathcal{P}^* = \{p_1, \ldots, p_K\}$ be an optimal path family. In view of our fitness measure, the score $\sigma(\mathcal{P}^*)$ does not yet have the desired properties, since it also depends on the lengths of the paths and captures trivial self-explanations. For example, the segment $\alpha = [1 : N]$ explains the entire sequence X perfectly, which is a trivial fact. More generally, each segment α explains itself perfectly (this information is encoded by the main diagonal of a self-similarity matrix). Therefore, to disregard such trivial self-explanations, we simply subtract the length $|\alpha|$ from the score $\sigma(\mathcal{P}^*)$. Furthermore, we normalize the score with regard to the lengths L_k of the paths p_k (see Section 3.3.1) contained in the optimal path family $\mathcal{P}^*$. This yields the *normalized score* $\bar{\sigma}(\alpha)$ defined by

$$\bar{\sigma}(\alpha) := \frac{\sigma(\mathcal{P}^*) - |\alpha|}{\sum_{k=1}^{K} L_k}. \tag{3.16}$$

From the assumption $\mathcal{S}(n, n) = 1$, we obtain $\bar{\sigma}(\alpha) \geq 0$. Furthermore, note that, when using $\Omega = \{(1, 2), (2, 1), (1, 1)\}$, we get $\sum_k L_k \leq N$. This together with $\mathcal{S}(n, m) \leq 1$ implies the property $\bar{\sigma}(\alpha) \leq 1 - |\alpha|/N$. Intuitively, the value $\bar{\sigma}(\alpha)$ expresses the *average score* of the optimal path family $\mathcal{P}^*$ (minus a proportion for the self-explanations).

Next, we define some kind of *coverage* measure for α. To this end, let $\mathcal{A}^* := \{\pi_1(p_1), \ldots, \pi_1(p_K)\}$ be the segment family induced by $\mathcal{P}^*$, and let $\gamma(\mathcal{A}^*)$ be its coverage as defined in Section 3.3.1. Similar to the normalized score, we define the *normalized coverage* $\bar{\gamma}(\alpha)$ by

$$\bar{\gamma}(\alpha) := \frac{\gamma(\mathcal{A}^*) - |\alpha|}{N}. \tag{3.17}$$

As above, the length $|\alpha|$ is subtracted to compensate for trivial coverage. Obviously, one has $\bar{\gamma}(\alpha) \leq 1 - |\alpha|/N$.

To combine the coverage and the score measure, we define the *fitness* $\varphi(\alpha)$ of the segment α to be the harmonic mean

$$\varphi(\alpha) := 2 \cdot \frac{\bar{\sigma}(\alpha) \cdot \bar{\gamma}(\alpha)}{\bar{\gamma}(\alpha) + \bar{\sigma}(\alpha)}. \tag{3.18}$$

In doing so, the fitness integrates the normalized score and coverage into a single measure, inheriting the property $\varphi(\alpha) \leq 1 - |\alpha|/N$ from $\bar{\sigma}(\alpha)$ and $\bar{\gamma}(\alpha)$.

In conclusion, note that our normalization neglects trivial self-explanation and allows for comparing segments of different length while slightly favoring shorter segments. To illustrate the last property, suppose that a piece has the musical form $A_1 A_2 \ldots A_6$. Then $\varphi(\alpha) = 5/6$ when α corresponds to A_1, $\varphi(\alpha) = 2/3$ when α corresponds to $A_1 A_2$, $\varphi(\alpha) = 1/2$ when α corresponds to $A_1 A_2 A_3$, and $\varphi(\alpha) = 0$ when α corresponds to the entire piece.

3.4 Audio Thumbnailing

The fitness measure can be directly applied to audio thumbnailing. Similar to [28, 109], the basic idea is to define the thumbnail to be the segment of maximal fitness:

$$\alpha^* := \operatorname*{argmax}_{\alpha} \ \varphi(\alpha). \tag{3.19}$$

Note that the induced segment family of α^* reveals the repetitive structure of the thumbnail. To account for prior knowledge and to remove spurious estimates, one can impose additional requirements on the thumbnail solution. In particular, as also shown by our experiments in Section 3.7, introducing a lower bound θ for the minimal possible thumbnail length allows us to reduce the effect of noise scattered in the underlying self-similarity matrix. Extending the above definition, we define

$$\alpha^*_\theta := \operatorname*{argmax}_{\alpha, |\alpha| \geq \theta} \ \varphi(\alpha). \tag{3.20}$$

In this section we only presented how we select the thumbnail using our fitness measure. We will continue to discuss and illustrate the properties of the fitness measure and the thumbnailing procedure in detail in Section 3.6. In the next section, we introduce a visualization which allows us to check fitness values for all possible segments.

3.5 Fitness Scape Plot

In this section, we introduce a compact fitness representation for the entire music recording, showing the fitness $\varphi(\alpha)$ for all possible segments α. Note that each segment $\alpha = [s:t] \subseteq [1:N]$ can be represented by its center $c(\alpha) := (s+t)/2$ and its length $|\alpha|$. Using the center to parameterize a horizontal axis and the length to parametrize the height, each segment corresponds to a point in some triangular representation, also referred to as a *scape plot*.

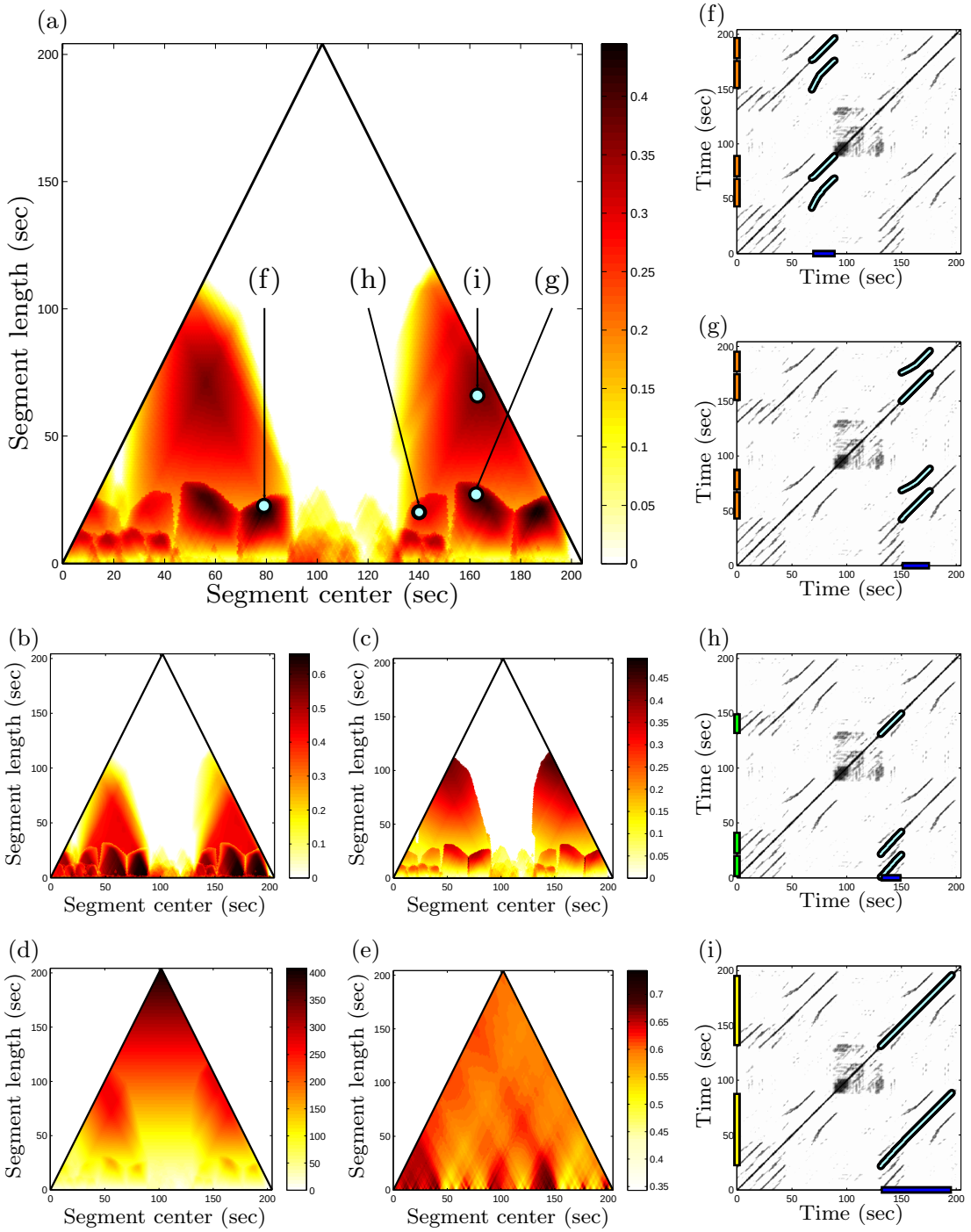

Figure 3.5: Various scape plot representations as well as different optimal path families and induced segment families over different segments α for our Brahms example. **(a)** Fitness measure (harmonic mean). **(b)** Normalized score. **(c)** Normalized coverage. **(d)** Score. **(e)** Average measure as suggested in [29]. **(f)** $\alpha = \alpha^* = [68:89]$ (thumbnail, maximal fitness, corresponding to B_2). **(g)** $\alpha = [150:176]$ (corresponding to B_3). **(h)** $\alpha = [131:150]$ (corresponding to A_3). **(i)** $\alpha = [131:196]$ (corresponding to $A_3B_3B_4$).

Such scape plots were originally introduced by Sapp [116, 117] to represent harmony in musical scores in a hierarchical way. In our context, we define a scape plot Φ by setting $\Phi(c(\alpha), |\alpha|) := \varphi(\alpha)$ for segment α. Figure 3.5a shows a color-coded representation of the scape plot for our Brahms example. Note that the maximal entry of Φ corresponds to the maximal fitness value, thus defining the thumbnail α^*. Furthermore, the segments with $|\alpha| \geq \theta$ correspond to all points in Φ that lie above a horizontal line with its height specified by θ.

3.6 Properties of Fitness Measure

We now discuss some explicit examples to illustrate the properties and potential of our fitness measure, the scape plot, and the derived segmentation.

We first continue with our Brahms example with the musical form $A_1A_2B_1B_2CA_3B_3B_4$, see Figure 3.1. The fitness scape plot of the Ormandy recording of this piece, as shown in Figure 3.5a, reflects this structure in a hierarchical way. The fitness-maximizing segment is $\alpha^* = [68:89]$ ($c(\alpha) = 78.5$, $|\alpha| = 22$) and corresponds to B_2. Furthermore, the induced segment family consists of the four B-part segments, see Figure 3.5f. Note that all four B-part segments have almost the same fitness and lead to more or less the same segment family. For example, Figure 3.5g shows the induced segment family, when considering the segment corresponding to B_3. This reflects the fact that each of the B-part segments may serve equally well as the thumbnail. Recall that our fitness measure slightly favors shorter segments. Therefore, since in this recording the B_2-part is played faster than the B_3-part, our fitness measure favors the B_2-part segment to the B_3-part segment. The scape plot also reveals other local maxima of musical relevance. For example, the local maximum corresponding to segment $\alpha = [131:150]$ ($c(\alpha) = 140.5$, $|\alpha| = 20$) reflects part A_3, and the induced segment family reveals the three A-parts, see Figure 3.5h. Furthermore, the local maximum corresponding to segment $\alpha = [131:196]$ ($c(\alpha) = 163.5$, $|\alpha| = 66$) reflects $A_3B_3B_4$, which is a repetition of $A_2B_1B_2$, see Figure 3.5i. Again, note that, because of the normalization, the fitness of $\alpha = [131:196]$ is well below the one of, e. g., the thumbnail $\alpha^* = [68:89]$.

Next, we illustrate that in the definition of the fitness measure (see Eq. (3.18)) the combination of the normalized score and coverage is of crucial importance. Figure 3.5b shows the scape plot with only the normalized score. Since this measure only expresses the average score of a path family without expressing how much of the audio material is actually captured, many of the small segments have a relatively high score. Using such a measure would typically result in false positive segments of short length. In contrast, using only the normalized coverage would typically favor longer segments, see Figure 3.5c. Now, by combining score and coverage, our fitness measure balances out the two conflicting principles of having firm repetitions (high score) and of explaining possibly large portions of the recording (high coverage). Next, Figure 3.5d illustrates the importance of normalization and subtraction of self-explanations. We obtain this scape plot by simply using the score $\sigma(\mathcal{P}^*)$ of the optimizing path family $\mathcal{P}^*$ over α. As a result, longer segments typically dominate the shorter segments, with the entire recording having maximal score. Finally, Figure 3.5e shows a scape plot with the average score measure as proposed by [29], where

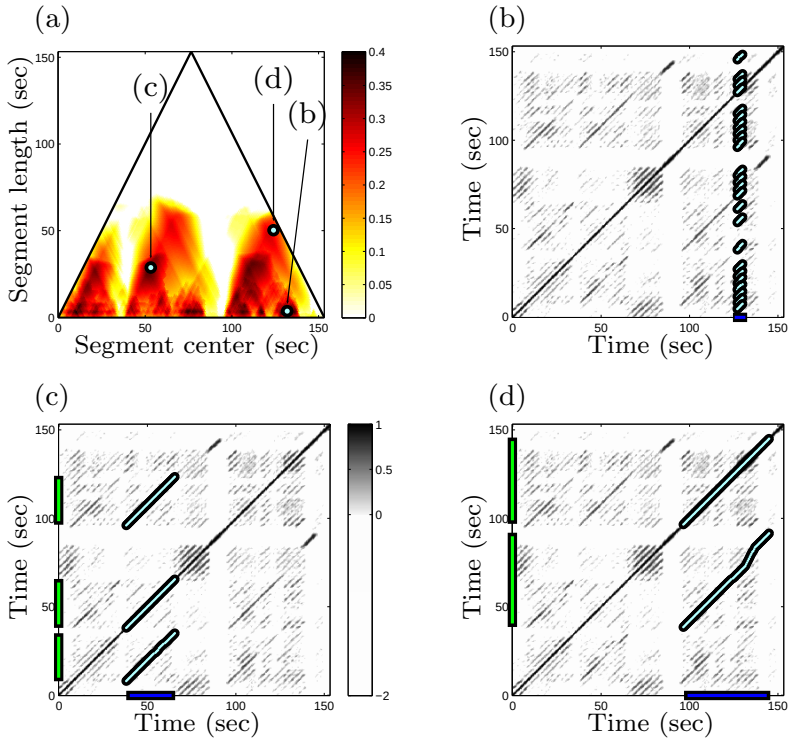

Figure 3.6: Various optimal path families and induced segment families over different segments α for the Beatles song "Twist and Shout" having the musical form $IV_1V_2B_1V_3B_2O$. **(a)** Fitness scape plot. **(b)** $\alpha = \alpha^* = [127:130]$. **(c)** $\alpha = \alpha^*_\theta = [38:65]$ using $\theta = 10$ (corresponding to V_2). **(d)** $\alpha = \alpha^*_\theta = [97:145]$ using $\theta = 40$ (corresponding to V_3B_2).

each segment α is assigned the average score of all cells of the submatrix $\mathcal{S}^\alpha$ (the matrix $\mathcal{S}^\alpha$ is defined in Section 3.3.2).[2] This average score measure captures relatively coarse homogeneity properties rather than repetitive structures.[3] As a result, the scape plot is less structured and only reveals high values for the A-parts that show a high degree of homogeneity with regard to the harmonic content (as captured by chroma-based audio features).

As a second example, Figure 3.6 shows the scape plot and various induced segment families for the Beatles song "Twist and Shout." This song has the rough musical form $IV_1V_2B_1V_3B_2O$ consisting of a short intro (I-part), three verses (V-part), two bridges (B-part) and an outro (O-part). Interestingly, the fitness maximizing segment $\alpha^* = [127:130]$ is very short and leads to a large number of spurious induced segments, see Figure 3.6b. The reason is that the song contains a short harmonic phrase that is repeated over and over again. As a consequence, the self-similarity matrix contains many repeated spurious path

[2] In the computation of the average score measure, we use the initial SSM without thresholding ($\rho = 1$, $\delta = 0$) as shown in Figure 3.2a. Actually, using enhanced matrices as shown in Figure 3.2c yields similar results.

[3] The average score measure [29] was originally applied to timbre-related audio features, which tend to form homogeneity regions in an SSM.

fragments which, as a whole family, lead to a high score as well as to a high coverage value. To circumvent such problems, one can consider the segment α_θ^* as defined in Eq. (3.20) to enforce a minimal length for the thumbnail. For example, setting $\theta = 10$ (given in seconds) we obtain the segment $\alpha_\theta^* = [38:65]$ for our Beatles song, which corresponds to the verse V_2, see Figure 3.6c. This indeed yields a musically meaningful thumbnail. By further increasing the lower bound, we obtain superordinate repeating parts such as $\alpha_\theta^* = [97:145]$ corresponding to V_3B_2 (when using $\theta = 40$), see Figure 3.6d.

3.7 Experiments

In this section, we report on various experiments, which highlight the applicability and the performance of our fitness measure. In particular, we give a quantitative evaluation and discuss the benefits as well as the limitations of our approach in the context of an audio thumbnailing application for classical and popular music.

In the pipeline of our experiments, we apply our fitness measure to determine the audio thumbnail. In the following, we describe the datasets (Section 3.7.1), introduce the evaluation measure (Section 3.7.2), investigate the role of various parameters (Section 3.7.3), compare our thumbnailing procedure to other approaches (Section 3.7.4), and finally discuss possible error sources (Section 3.7.5). In our evaluation, we rely on manually generated ground-truth (GT) annotations, which serve as references to compare against. Note that the term "ground-truth" as well as the usage of such annotations is problematic in the sense that different experts may disagree on how to segment and label the data. Therefore, it would be desirable to compare against annotations obtained by an entire panel of experts and to see if the results obtained by automated methods exhibit the same variability as the ones by the experts. Nevertheless, using annotations obtained by only one expert, the subsequent evaluation still indicates certain tendencies and illustrates the overall behavior of our procedure.

3.7.1 Datasets

We now describe three datasets consisting of popular and classical music recordings. The first dataset consists recordings from the 12 studio albums by the British band "The Beatles". This dataset is a subset of the *Isophonics* dataset [4] which is provided by Center for Digital Music at Queen Mary University of London. The Isophonics dataset contains 301 recordings of various artists (The Beatles, Michael Jackson, Queen, Carole King and Zweieck) with chord annotations, key annotations and structure annotations as well [82]. We take the 180 Beatles recordings and their respective structure annotations from the Isophonics dataset and form the first dataset for our experiments, we denote it as `BEATLES`. Actually, for five of these songs[5], no clear repetitions were present in the annotations. Leaving out these songs in our experiments, `BEATLES` contains 175 recordings (instead of the original 180 songs).

[4] http://www.isophonics.net/datasets

[5] 'HappinessIsAWarmGun', 'HerMajesty', 'Revolution9', 'TheEnd', and 'YouNeverGiveMeYourMoney'

The second dataset consists recordings taken from the RWC (Real World Computing) Music Database [6]. The RWC database is generated by the Japanese National Institute of Advanced Industrial Science and Technolody (AIST). It is a copyright-cleared music database which is available for researchers to work as a common foundation. It contains music recordings together with manual annotations such as beat, chord, structure annotations. Here, we take the RWC popular music database [46] which consists of 100 recordings and the respective structure annotations [44] together to form our second dataset, which is denoted as `RWC-POP`.

The third dataset consists of classical music, which is the 49 Mazurkas composed by Frédéric Chopin. The music recordings are taken from the Mazurka Project [7] provided by the AHRC Research Centre for the History and Analysis of Recorded Music (CHARM). The Mazurka Project contains 2792 piano recordings played by various artists. For each piece of the Mazurkas, we first manually generated the structure annotation for the corresponding MIDI file and then transfer the structure annotations to the respective recordings automatically by a synchronization procedure in our group [35, 99]. We take the complete recordings of the 49 Mazurkas played by the three pianists Rubinstein (1966), Cohen (1997), and Ezaki (2006), respectively. The resulting 147 recordings and their structure annotations form the third dataset in our experiments, which we denoted as `MAZURKA`.

Table 3.1 gives an overview of the two datasets, indicating the number of recordings, the average duration of the recordings, and the total duration. Furthermore, as explained in more detail in Section 3.7.2, the average number of GT thumbnails per song, the average duration (given in seconds and in percent) of GT thumbnails, and the average coverage (given in percent) of the induced segment families are specified.

	Num	Av. Dur.	Total Dur.	$\lvert\mathcal{A}^{\mathrm{GT}}\rvert$	$\lvert\alpha^{\mathrm{GT}}\rvert$	$\lvert\alpha^{\mathrm{GT}}\rvert$	$\gamma(\mathcal{A}^{\mathrm{GT}})$
Dataset	[#]	[sec]	[hh:mm]	[#]	[sec]	[%]	[%]
`BEATLES`	175	162.1	7:52	4.0	22.1	14.4	54.1
`RWC-POP`	100	244.7	6:47	6.3	16.5	6.8	40.0
`MAZURKA`	147	171.2	6:59	4.3	18.9	11.9	45.9

Table 3.1: Overview of the two evaluation datasets showing the respective number (Num), average duration (Av. Dur.) and total duration (Total Dur.) of the recordings. The remaining numbers are specified in the text.

3.7.2 Evaluation Measures

For both datasets, there are structural annotations that consist of segmentations of the music recordings and a labeling of the segments by letters such as $A, B, C, \ldots$ representing musical parts. For each label, one can associate a segment family that consists of all segments marked with this label. In the thumbnailing scenario, we do not need the entire structure annotations, but only the label and associated segment family that represents the most repetitive musical part. This segment family then serves as the *GT thumbnail*

[6] https://staff.aist.go.jp/m.goto/RWC-MDB/
[7] http://www.mazurka.org.uk/

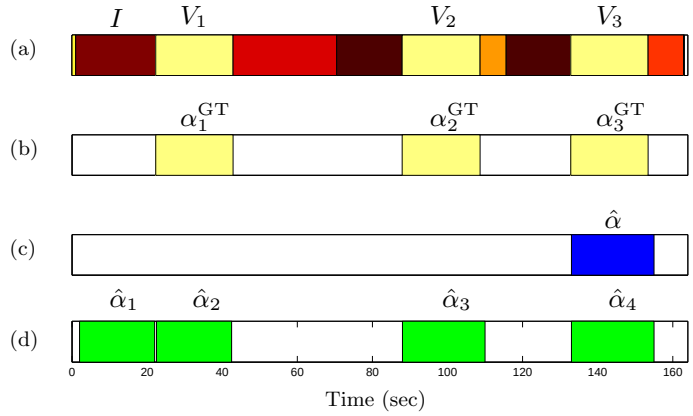

Figure 3.7: Different segmentations for the Beatles song "Birthday." **(a)** GT structure annotation. **(b)** Induced segment family $\mathcal{A}^{\mathrm{GT}}$ of the GT thumbnail. **(c)** Estimated thumbnail $\hat{\alpha}$. **(d)** Induced segment family $\hat{\mathcal{A}}$ of $\hat{\alpha}$.

family. To derive such a family, we compute for each labeled segment the normalized coverage for the associated segment family as in Eq. (3.17). Then, we take the labeled segment that maximizes the normalized coverage as the GT thumbail and the associated segment family as the GT thumbnail family. For example, in Figure 3.7, the GT thumbnail family corresponds to the three verse segments labeled as V_1, V_2, and V_3. Note that each of these segments may serve equally well as the GT thumbnail. In general, let $\mathcal{A}^{\mathrm{GT}} := \{\alpha_1^{\mathrm{GT}}, \ldots, \alpha_K^{\mathrm{GT}}\}$ denote the GT thumbnail family representing the various possible GT thumbnails, see also Figure 3.7b.

Furthermore, let $\hat{\alpha}$ denote an estimated segment obtained from a given thumbnailing procedure. To measure how well the estimated thumbnail $\hat{\alpha}$ corresponds to the GT thumbnails, we compute the precision $\mathrm{P}_k^{\alpha} = (|\hat{\alpha} \cap \alpha_k^{\mathrm{GT}}|)/|\hat{\alpha}|$, the recall $\mathrm{R}_k^{\alpha} = (|\hat{\alpha} \cap \alpha_k^{\mathrm{GT}}|)/|\alpha_k^{\mathrm{GT}}|$, and the F-measure $\mathrm{F}_k^{\alpha} = 2\mathrm{P}_k^{\alpha}\mathrm{R}_k^{\alpha}/(\mathrm{P}_k^{\alpha} + \mathrm{R}_k^{\alpha})$ for each $k \in [1\!:\!K]$ and then define the *thumbnail F-measure* by

$$\mathrm{F}^{\alpha} = \max_{k \in [1:K]} \mathrm{F}_k^{\alpha}. \tag{3.21}$$

In other words, the thumbnail F-measure expresses to what extent $\hat{\alpha}$ agrees with one of the GT thumbnails contained in $\mathcal{A}^{\mathrm{GT}}$. As an example, Figure 3.7 shows the case where the estimated thumbnail $\hat{\alpha}$ best agrees with α_3^{GT} (corresponding to V_3), yielding $\mathrm{F}^{\alpha} = 0.96$. Finally, let $\hat{\mathcal{A}} = \{\hat{\alpha}_1, \ldots, \hat{\alpha}_M\}$ be the induced segment family of $\hat{\alpha}$. For example, as shown by Figure 3.7, the Beatles song "Birthday" has three segments annotated as verse, whereas the induced family of the estimated thumbnail consists of four segments. It turns out that the intro (segment labeled as I) is harmonically very similar to the verse segments. Actually, the intro is an instrumental version of the verse. So, it is hard to say whether the estimated result or the ground-truth should be questioned.

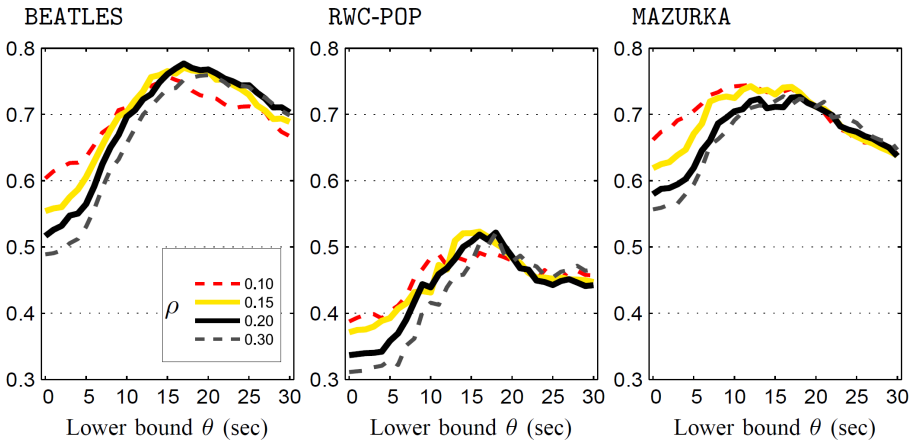

Figure 3.8: Thumbnail F-measure values F^{α} for three different datasets in dependency of different parameters. The horizontal axis specifies the lower bound parameter θ (given in seconds) and the different colors correspond to different values for the relative threshold parameters ρ.

3.7.3 Dependency on parameters

We now examine the overall performance of our thumbnailing procedure. Note that it is not our intention to optimize and to advocate specific parameter settings. Instead, our main goal is to investigate the role and interdependencies of certain parameters as well as to indicate the conceptual benefits introduced by our fitness measure. In the following, we use the transposition-invariant SSM as introduced in Section 3.2 and Section 2.2.4. To suppress small values, we use the relative threshold parameter ρ in the SSM together with the penalty parameter $\delta = -2$ (see Section 3.2). Furthermore, the parameter θ (see Section 3.4) is used to specify a minimum length for the chosen thumbnail.

Figure 3.8 shows the thumbnail F-measure F^{α} for various choices of $\rho \in \{0.1, 0.15, 0.2, 0.3\}$ and different estimated thumbnails $\hat{\alpha} = \alpha^*_{\theta}$ using $\theta \in [0 : 30]$. Note that the results first improve with increasing θ (up to $\theta = 15$) and then deteriorate again when further increasing the lower bound. For example, in case of the dataset `BEATLES`, the F^{α}-value is roughly 0.75 (or slightly above) when using $\theta = 17$, independent of the specific choice of ρ. As also illustrated by Figure 3.6, using the lower bound θ allows for disregarding path families that consist of a large number of short spurious path fragments. This also explains why the role of ρ becomes more important for smaller θ: using smaller values for ρ removes more of the noise-like artifacts in the SSM that typically lead to spurious path fragments. While the thumbnail F-measure shows a similar behavior for the datasets `BEATLES` and `MAZURKA`, the results are significantly worse for `RWC-POP`. A manual investigation reveals that the annotations for `RWC-POP` are rather inconsistent. Often, related segments are marked by different labels and vice versa. Such an example is shown by Figure 3.9a, where a manual inspection showed that the bridges B_1 and B_2 are closely related (with regard to harmonic progression) to the four chorus sections (C) and the intro (I), whereas the bridge B_3 is rather different from the other bridges. Furthermore, the five segments labeled as V are far from being repetitions. Actually, V_1 and V_3 form a family, whereas V_2, V_4 and V_5 form a different family. So the poor results for `RWC-POP` are often due to such ground-truth problems rather than due to algorithmic problems. Nevertheless, the

	BEATLES	RWC-POP	MAZURKA
Fitness φ	0.761	0.508	0.711
Normalized score $\bar{\sigma}$	0.631	0.460	0.659
Normalized coverage $\bar{\gamma}$	0.618	0.423	0.565
Average score measure [28]	0.476	0.320	0.436
Baseline (entire song)	0.263	0.138	0.240
Baseline (second sixth)	0.620	0.498	0.526
Average score measure [28] (GT length)	0.555	0.361	0.516
Fitness (GT length)	0.775	0.501	0.836

Table 3.2: Thumbnail F-measure F^{α} for various settings (using $\rho = 0.2$ and $\theta = 15$). The last four settings serve as baseline, where the segment length is specified using prior knowledge.

tendencies for F^{α} (in dependency of ρ and θ) are similar for all three datasets. Finally note that an optimal choice for θ also crucially depends on the statistics of the respective dataset, in particular on the average length of the expected thumbnails (see column $|\alpha^{\mathrm{GT}}|$ of Table 3.1).

3.7.4 Comparison of thumbnailing procedures

For the subsequent experiments, we exemplary choose $\rho = 0.2$ in combination with $\theta = 15$. By comparing results obtained from different procedures, we now demonstrate that our fitness measure outperforms other measures in the thumbnailing context. Table 3.2 shows the thumbnail F-measure for various settings. In the first four settings, we apply the same thumbnail selection strategy (using $\hat{\alpha} = \alpha^{*}_{\theta}$ as described in Section 3.4) based on four different measures: the fitness measure φ, the normalized score $\bar{\sigma}$, the normalized coverage $\bar{\gamma}$, and the average score measure suggested in [28], see also Figure 3.5. For example, for BEATLES, we obtain $F^{\alpha} = 0.761$ when using φ and $F^{\alpha} = 0.476$ when using the average score measure. For both datasets, we obtain the best results when using φ, whereas the average score measure does not work well in this context. Also, using $\bar{\sigma}$ and $\bar{\gamma}$ separately does not yield the same quality as with their combined usage in φ.

For comparison, we also conducted a number of additional baseline experiments. First, we computed F^{α} when using the entire song as the estimated thumbnail. Second, splitting up each recording into six segments of equal length, we used the second segment as the estimated thumbnail ("second sixth").[8] Third, using the actual length of the GT thumbnail for each recording, we applied the thumbnailing procedure restricted to the respective GT length, once using the average score measure and then using our fitness measure. The last four rows of Table 3.2 show the resulting thumbnail F-measures for each of these four baseline procedures. Again, our fitness measure yielded better results than the average score measure. Furthermore, using the second sixth of a song yielded seemingly good results, e. g., $F^{\alpha} = 0.620$ for BEATLES. This relatively high number was

[8] The motivation for using six segments is that this results in a segment length close to the average ground-truth thumbnail length, see Table 3.1. Furthermore, using the second segment for each recording is motivated by the fact that many songs or Mazurkas start with an intro and then continue with a verse or first theme corresponding to the thumbnail.

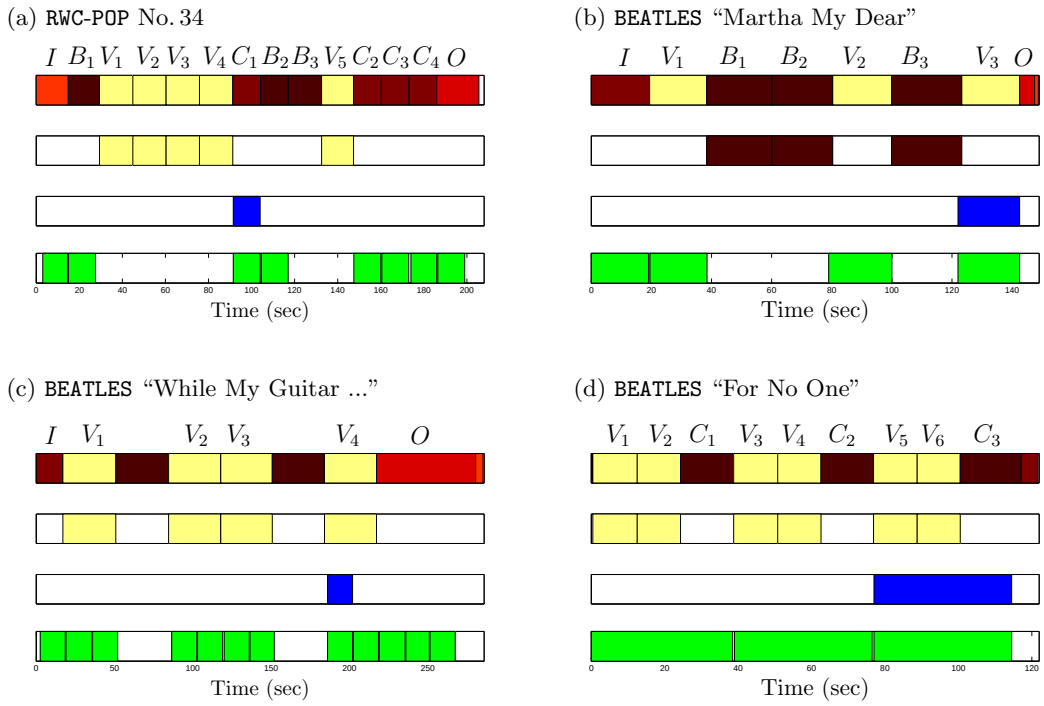

Figure 3.9: GT segmentation and thumbnailing results as in Figure 3.7 for different music recordings illustrating various error sources. **(a)** Poor annotation problem. **(b)** Confusion problem. **(c)** Subordinate structure problem. **(d)** Superordinate structure problem.

not only a consequence of the datasets' statistics, but also a consequence of the rather "soft" nature of our evaluation measure.

3.7.5 Error sources

As a first example, let us consider the Beatles song "Martha My Dear" (Figure 3.9b), where the annotated bridge segments were chosen as GT thumbnail family, whereas the estimated thumbnail derived from our procedure corresponds to a verse segment (which is actually quite similar to the intro). For this song, the V-part and B-part segments have roughly the same duration. As a result, it is hard to decide whether to use label V or B for defining the GT segment family. A second GT problem occurs when the thumbnail has a subordinate structure. For example, in the Beatles song "While My Guitar Gently Weeps" (Figure 3.9c) the verse has a subordinate structure basically consisting of two repeating subparts. Therefore, a segment that corresponds to the first or second half of the V-part may also serve as a meaningful GT thumbnail. Actually, such a segment was chosen by our procedure as the estimated thumbnail. Finally, superordinate repeating parts may also define meaningful thumbnails (in terms of being the element maximizing the normalized coverage) as illustrated by the Beatles song "For No One" (Figure 3.9d). Here, our procedure identified the superordinate structure VVC (consisting of two verses and a chorus) as thumbnail.

	A	B	$A/2$	$B/2$	Sup	Total
Fitness φ	0.55	0.01	0.06	0.01	0.19	0.83
Normalized score $\bar{\sigma}$	0.34	0.10	0.25	0.02	0.01	0.72
Normalized coverage $\bar{\gamma}$	0.15	0.02	0.04	0.00	0.12	0.33
Average score measure [28]	0.18	0.06	0.18	0.01	0.01	0.45
Baseline (entire song)	0.00	0.00	0.00	0.00	0.00	0.00
Baseline (second sixth)	0.17	0.05	0.03	0.00	0.09	0.34

Table 3.3: Accuracy rates for five different cases (GT thumbnail families) and different settings for dataset `BEATLES` using $\rho = 0.2$, $\theta = 15$, and $\mathrm{F}^{\alpha} \geq 0.8$. The last column shows the total accuracy rates summed over the five cases.

Using the dataset `BEATLES`,[9] we now give a quantitative evaluation that not only shows how often these phenomena actually occur, but also sheds a different light on the results presented in Table 3.2. To this end, we introduce a different, "harder" evaluation method. Given a recording and a ground-truth family, we say that an estimated thumbnail $\hat{\alpha}$ is *correct* if the resulting thumbnail F-measure lies above a certain threshold, otherwise it is *incorrect*. In our experiments, we use the criterion $\mathrm{F}^{\alpha} \geq 0.8$, basically saying that the estimated thumbnail must agree with a GT thumbnail by at least 80 %. Then, as for the evaluation, we simply count the songs with correctly estimated thumbnails and divide this number by the total number of songs, leading to a proportion we refer to as *accuracy rate*.[10] We compute the accuracy rate using different segment families to serve as the ground-truth: the segment family maximizing the normalized coverage (this case was also considered before and is denoted as case A), the segment family having the second highest normalized coverage (case B), the segment families corresponding to the halves of the parts (subordinate structures, case $A/2$ and case $B/2$), and segment families corresponding to superordinate structures using suitable N-grams of labels (case Sup).

Table 3.3 shows the accuracy rates for the five different cases and for different settings. For example, using the thumbnailing procedure based on our fitness measure, the GT thumbnail A was identified in 55 % of the songs, whereas a superordinate structure was identified in 19 % of the songs. In total, the procedure delivered a thumbnail corresponding to one of the five cases in 83 % of the songs. For the other settings, the results get significantly worse. For example, the average score measure leads to a success rate of only 45 %, when admitting all five cases. As mentioned above, using the normalized score tends to favor shorter segments (note the accuracy rate of 25 % for the case $A/2$), whereas using the normalized coverage tends to favor longer segments (note the accuracy rate of 12 % for the case Sup). Finally, in the last two rows of Table 3.3 one finds the accuracy rates for the two baseline methods as used in Table 3.2. By no surprise, using the "harder" counting measure (instead of a "softer" thumbnail F-measure F^{α}), the accuracy rate is 0 when using the entire song as the thumbnail. Also, using the "second sixth" of a recording as the thumbnail leads to a rather poor accuracy rate of only 34 %. In conclusion, Table 3.3 again shows that the fitness measure constitutes a valuable tool for audio thumbnailing.

[9] For `MAZURKA` the numbers are quite similar.

[10] Naturally, using a stricter (softer) criterion such as $\mathrm{F}^{\alpha} \geq 0.85$ ($\mathrm{F}^{\alpha} \geq 0.75$) leads to lower (higher) accuracy rates. The overall tendencies of the accuracy rates are similar for different choices.

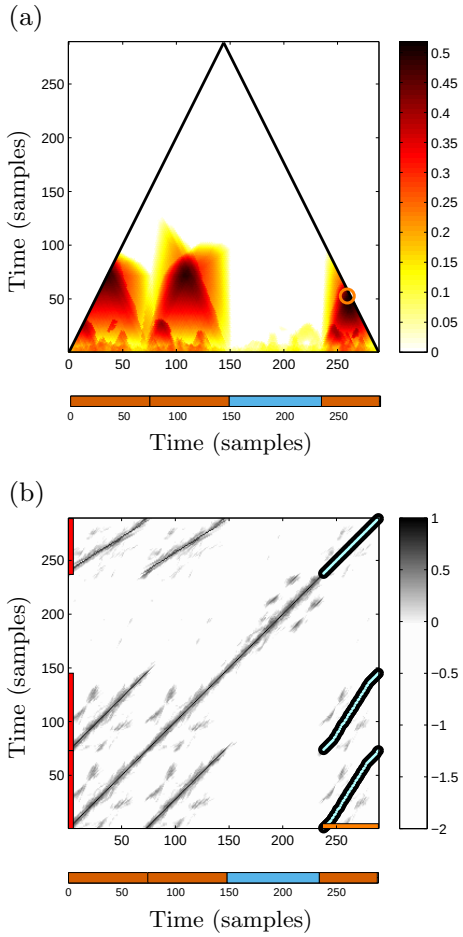

Figure 3.10: Thumbnailing application. **(a)** Fitness scape plot and ground-truth segmentation. **(b)** SSM with thumbnail (shown on horizontal axis), path family (cyan), induced segment family (shown on vertical axis) and ground-truth segmentation.

3.8 Implementation

The SM toolbox, which is presented in Section 2.3, also includes the MATLAB implementation for our audio thumbnailing procedure. In this section, we introduce some related functionality of audio thumbnailing as well as some assisting visualization tools, which are another main contribution of the SM toolbox.

Table 3.4 is an extension of Table 2.1. It shows the main functions implementing the thumbnailing procedure. After several enhancement processing as introduced in Section 2.2, we can get a self-similarity matrix with transposition invariance, tempo invariance and proper smoothing (as computed in line 44 of Table 2.2). The function SSM_to_scapePlotFitness is used to derive a fitness scape plot from a similarity matrix. Here, various step size and weighting parameters can be used to adjust the procedure. We

Filename	Main parameters	Description
SSM_to_scapePlotFitness	stepSize, stepWeight	Computation of fitness scape plot from self-similarity matrix.
scapePlotFitness_to_thumbnail	lowerBound	Computation of thumbnail segment from fitness scape plot.
thumbnailSSM_to_pathFamily		Computation of induced segment family from thumbnail.
visualizeScapePlot	featureRate, print, figureName	Visualization of fitness scape plot.
visualizePathFamilySSM	featureRate, showSegInduced, showThumbnail	Visualization of similarity matrix and path family.
visualizeSegFamily	print, figureName	Visualization of segment family.

Table 3.4: Overview of the main MATLAB functions contained in SM toolbox [124] for the thumbaniling application. The most important parameters are shown in the middle column. This Table is an extension of Table 2.1.

again use the audio file which is introduced in Section 2.1 as an illustrative example. The computed fitness values can be visualized in a fitness scape plot, and the visualization can be done by function visualizeScapePlot, see Figure 3.10a. Furthermore, using the fitness scape plot as input, the function scapePlotFitness_to_thumbnail outputs the thumbnail. Then, together with the SSM, the function thumbnailSSM_to_pathFamily computes the corresponding path family and the induced segment family, which can be visualized by the function visualizePathFamilySSM, see Figure 3.10b. Finally, the function visualizeSegFamily can be used for visualizing any segment family, e. g., the ground-truth segmentation as shown beneath the scape plot and SSM.

3.9 Conclusions and Further Notes

In this chapter, we have made several contributions to the field of music and audio structure analysis. First of all, we have introduced a novel fitness measure that expresses how representative a given segment is in terms of its repetitiveness. Our experiments have shown that the fitness-maximizing segment generally yields good estimates for musically meaningful thumbnails—even in the presence of acoustic and musical variations across repeated segments. The main idea was to jointly perform path extraction and grouping within a unifying optimization scheme, which yields a trade-off between quantity and length of paths (coverage) and quality of paths (score). Furthermore, we have shown how optimal path families (underlying the fitness measure) can be computed efficiently by suitably modifying the classical DTW algorithm. As a further contribution, we have introduced a scape plot representation that yields a compact and (so we think) aesthetically appealing visualization of the global repetitive structure of a given music recording.

Our techniques may be applicable not only to audio thumbnailing, but also to general structure analysis and segmentation tasks. For example, our fitness measure may be helpful for detecting and analyzing long-term repetitive structures that arise in large-scale works such as symphonies or sonatas. We discuss the application on such data in Chapter 6. Further challenges regard efficiency issues when computing and analyzing the scape plot representation. For example, similar to [107], one may use additional cues such as points of novelty in order to restrict start and end points of thumbnail candidates.

Then, the scape plot may only be evaluated on a grid or certain points. Furthermore, the scape plot computation may be accelerated by using multiscale approaches based on different feature resolutions. Particularly for regions within the scape plot corresponding to long segments or segments of poor fitness, a coarse feature resolution may suffice. Subsequently, only the regions of potentially high fitness need to be refined using a higher feature resolution. We further discuss such efficiency issues in Chapter 4.

Chapter 4

Towards Efficient Audio Thumbnailing

In this chapter we introduce several strategies to accelerate the computation of our thumbnailing procedure, the content of this chapter closely follows the publication [57].

In the previous chapter we have discussed about the audio thumbnailing task which aims to automatically determine the most representative section of a given music recording. [11, 19, 28, 45, 72, 95]. Typically, good candidates of audio thumbnails can be, for example, a chorus section of a pop song or a main theme of a classical work. Such musical parts are typically repeated several times throughout the recording. Therefore, to determine a thumbnail automatically, most procedures try to identify a section that has on the one hand side a certain appropriate duration and on the other side many (approximate) repetitions.

We have also built our own thumbnailing procedure that captures repetitiveness as well as coverage, as described in Chapter 3. Even though the procedure yields promising thumbnailing results representing the state-of-the-art, it has the drawback of being computationally expensive because of the two following reasons. First, the fitness is computed for all possible segments, the number of which is quadratic with the duration of the song. Second, in computing the fitness of a single segment, the segment is brought into relation to other repeating segments, a process that again requires a quadratic running time. Altogether, this yields a complexity that is proportional to the fourth power in the duration of the song.

As main contribution of this chapter, we introduce three different strategies that lead to significant accelerations of the original procedure. As a first strategy, we introduce a hierarchical multi-level approach, where the fitness is first computed on a coarse grid of scape plot points (the points corresponding to audio segments). Then, suitable neighborhoods of only those grid points that have the highest fitness are selected and iteratively refined, which significantly reduces the overall number of fitness computations. The second strategy is to accelerate the actual fitness computation by adjusting the resolution used in deriving the mutual repetition relations of the segments, where the resolution is coupled to the level of the previously described grid sampling approach. As the third strategy, we

exploit the mutual relations that are detected in the fitness computation. These relations express repetitiveness within certain segment families and allow us to estimate the fitness for all these segments in one step without any further computation. Our experiments on two different datasets (Beatles Songs, Chopin Mazurkas) show that each of these strategies lead to significant accelerations that are independent from each other. Using a combined approach, we obtain accelerations by a factor of roughly 20 to 200 (depending on the duration of the song) while keeping the overall accuracy of the thumbnailing procedure.

The remainder of this chapter is organized as follows. We describe the three acceleration strategies in Section 4.1. Then, in Section 4.2, we report on our systematic experiments and draw some conclusions.

4.1 Acceleration Strategies

Before we describe the acceleration strategies, here we briefly summarize the thumbnailing procedure and the scape plot visualization which are necessary for understanding these strategies. Also we introduce some notations following Chapter 3. We assume that the given music recording is represented by a feature sequence with a sampled time axis indexed by $[1 : N] = \{1, 2, \dots N\}$. (In our experiments we use a feature resolution of 2 Hz.) A segment is then understood to be a subset $\alpha = [s : t] \subseteq [1 : N]$ specified by its starting point s and its end point t with $|\alpha| := t - s + 1$ denoting its length. In Chapter 3, a fitness measure of a segment is defined to capture its repetitiveness as well as coverage in a given audio recording. In detail, we compute for all possible segments of a given recording about their fitness values. Mathematically, we denote fitness value as $\varphi(\alpha)$ for a segment α. Then, similar to [28, 109], the audio thumbnail is defined to be the segment having maximal fitness. Mathematically, we denote the thumbnail to be $\alpha^* := \text{argmax}_{\alpha}(\varphi(\alpha))$.

Representing each audio segment by means of its center and length, the fitness values of all segments can be visualized by a triangular *scape plot*, which reveals the repetitive structure of the entire music recording in a hierarchical and compact way [92, 116, 117]. Each point of the scape plot corresponds to a segment $\alpha = [s : t]$, where the horizontal coordinate encodes the center $c(\alpha) := (s + t)/2$ of the segment and the vertical coordinate its length $|\alpha|$. The fitness value $\varphi(\alpha)$ is then visualized in some color-coded form. The fitness scape plot represents the repetitive structure of the music recording in some hierarchical way, see Chapter 3 Section 3.5 for details. An example is shown in Figure 4.1, which shows a scape plot for the song "Act Naturally" by the Beatles. The fitness maximizing point, indicated by the circle, corresponds to the verse section, which appears four times in the song.

4.1.1 Acceleration by Multi-Level Sampling

The first of our acceleration strategies is rather straightforward. Instead of computing the fitness for all possible segments (at the given resolution of the feature sequence indexed by $[1 : N]$), we apply an iterative multi-level approach, see Figure 4.2 for an overview. To this end, we consider a *regular grid* of points in the two-dimensional scape plot representation,

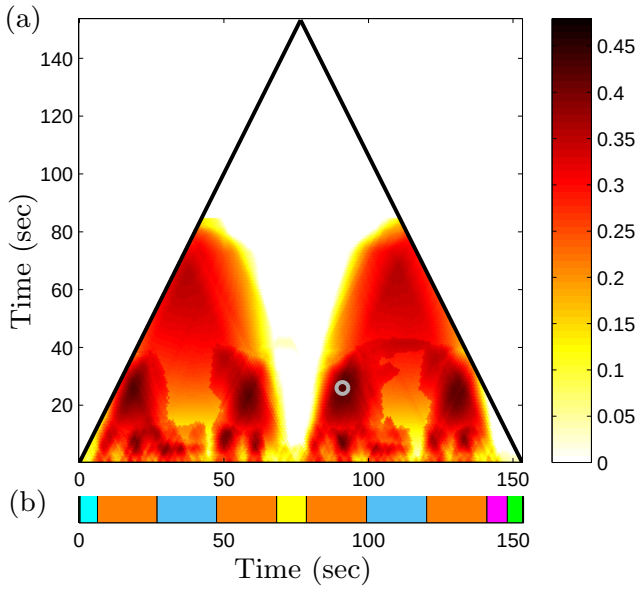

Figure 4.1: Thumbnailing procedure for Beatles song "Act Naturally". **(a)** Fitness scape plot with thumbnail segment indicated by the circle point. **(b)** Ground-truth structure annotation.

where neighboring grid points are d samples apart either in horizontal or in vertical direction, see Figure 4.2a. The fitness of these grid points are then computed, see Figure 4.2b. The parameter $d \in \mathbb{N}$ determines the density of the grid with $d = 1$ yielding the scape plot in full resolution. In the first step of the multi-level approach, we use a parameter $d = d_1 > 1$ and compute the fitness only for those points that lie on the resulting scape plot grid.

One crucial observation in the scape plot is all points that lie in a neighborhood of a scape plot point of high fitness also typically have large fitness values. Therefore, it is reasonable to assume that the thumbnail segment lies in the neighborhood of one of the grid points of high fitness. This observation justifies the next steps of our procedure. Fixing a parameter $M \in \mathbb{N}$, we select among all grid points the M points that have the top M largest fitness values (or less points if the grid contains less than M points). These M points are also referred to as *anchor points*. Using a parameter $d_2 \in \mathbb{N}$ with $1 \leq d_2 < d_1$ (in our experiments we use $d_2 = d_1/2$ assuming that d_1 is a power of two), we consider all points on the refined grid (based on d_2) that are direct neighbors of one of the M anchor points. For all the resulting additional points, we then compute the fitness, see Figure 4.2c.

This last step can be iterated by selecting again the M points of highest fitness (among all previously considered points in the first two steps), using a finer grid based on some $d_3 \in \mathbb{N}$ with $1 \leq d_3 < d_2$, and again considering the neighbors, see Figure 4.2d. This process is repeated until one reaches the finest resolution. Finally, we define the scape

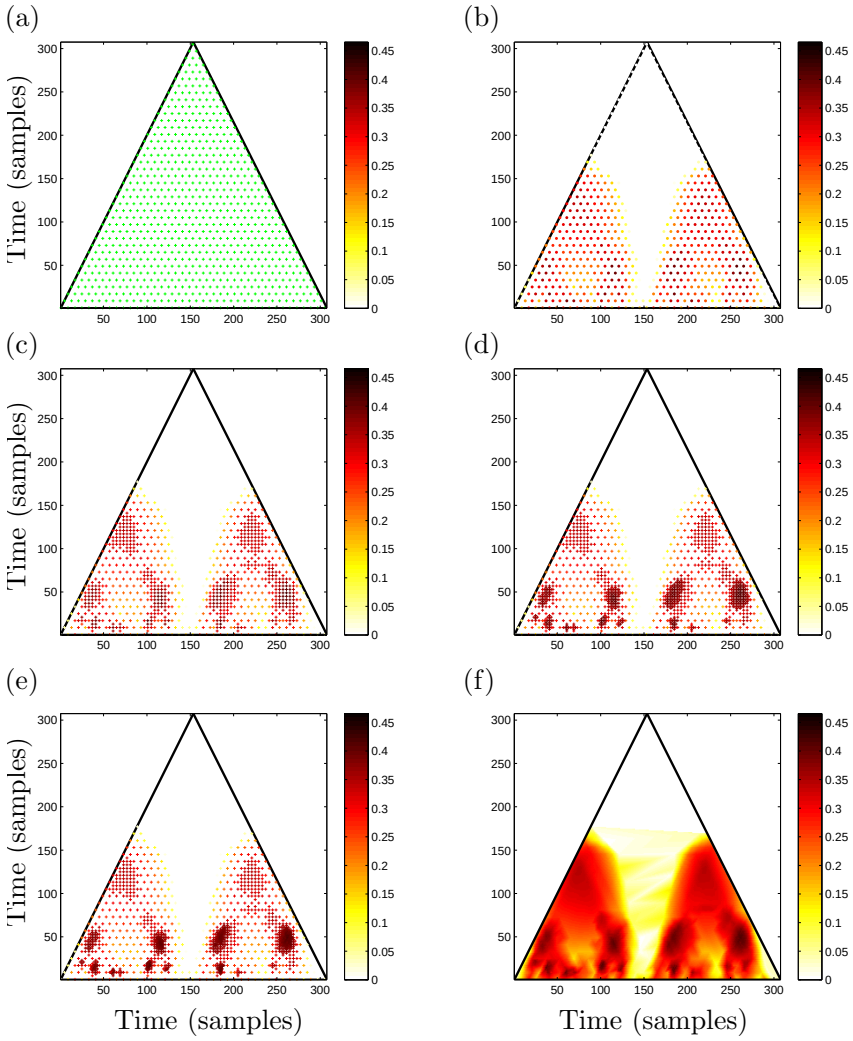

Figure 4.2: Illustration of the multi-level grid sampling approach for the Beatles song "Act Naturally," see Figure 4.1. The used parameters are $I = 4$ with $(d_1, d_2, d_3, d_4) = (8, 4, 2, 1)$ and $M = 100$. **(a)** Grid of the first step (using $d_1 = 8$). **(b)** Fitness computed for the grid points in (a). **(c)** Refinement after second step (using $d_2 = 4$). **(d)** Refinement after third step (using $d_3 = 2$). **(e)** Refinement after forth step (using $d_4 = 1$). **(f)** Scape plot obtained from (e) by interpolation, compare with Figure 4.1a.

plot point α_1^* to be the one of maximal fitness over all grid points considered in the entire procedure.

We say that our procedure has been *successful* if α_1^* coincides or, at least, is close to the actual thumbnail α^*. Here, as we will discuss later, we mean by "close" that the segments α^* and α_1^* induce the same repetitive structure. Besides finding the original thumbnail, also the visualization of the scape plot may be of interest. To this end, we generate a visualization on the finest possible resolution using simple interpolation techniques applied

to all grid points considered in the multi-level approach, see Figure 4.2f [1].

To conclude the description of the first acceleration strategy, note that the parameters should be chosen in such a way that the procedure is successful, the running time is reduced as much as possible, and the visual impression of the interpolated scape plot is close to the original one. In our experiments, as we will present in Section 4.2, the specific setting turned out not to be crucial within a wide range of parameters leading to similar results. In particular, using $I = 4$ with $(d_1, d_2, d_3, d_4) = (8, 4, 2, 1)$ and $M \in [10 : 100]$ has turned out to be a reasonable choice.

4.1.2 Acceleration by Multi-Resolution Fitness Computation

In the first strategy, we have reduced the number of segments the fitness measure has to be evaluated for. We now describe a second acceleration strategy which speeds up the actual fitness computation. To this end, we first need to summarize how the fitness measure is defined, see Section 3.3 for details. In the computation of the fitness measure, an enhanced self-similarity matrix (SSM) is computed on the basis of chroma features extracted from the music recording, see Figure 4.3a. Then for each segment α, an *optimal path family* that simultaneously reveals the relations between α and all other similar segments is computed. By projecting such an optimal path family to the vertical axis, one obtains an induced segment family, where each element of this family defines a segment similar to α. As an illustration, Figure 4.3b shows such an optimal path family for the segment $\alpha = [158 : 209]$ (horizontal axis) as well as the induced segments $\alpha_1, \alpha_2, \alpha_3$, and α_4 (vertical axis). Note that these four segments are exactly the four repeating verse sections of the song.

The computation of an optimal path family over a given segment α can be done using dynamic programming in $O(|\alpha| \cdot N)$ operations, see Chapter 2. The algorithm is similar to the one used for dynamic time warping, see, e. g., [88, Chapter 4]. Obviously, the running time can be reduced when reducing the resolution in the underlying SSM. For example, in theory, reducing the resolution by a factor of two yields a speed up of the dynamic programming step by a factor of four. However, reducing the resolution too much may also lead to a deterioration of the similarity matrix, where important structural properties may get lost [97] and may lead to inaccuracies in the fitness computation. As a result, certain relations to be captured by the optimal path family may be missed as illustrated by Figure 4.3d. Therefore, applying this strategy needs to be done with care.

In a pilot experiment, we accelerated the fitness computation by simply reducing the SSM resolution from 2 Hz to 1 Hz. This led to a substantial reduction in running time with only a small decrease in the overall accuracy of the thumbnail estimation. Next, we further reduced the SSM resolution to 0.5 Hz. While further speeding up the computation, this resolution resulted in a severe deterioration of the thumbnail estimation, in particular in the case of thumbnails of short duration. In other words, 0.5 Hz is too low a resolution to reveal the desired structures. Therefore, we apply the strategy of reducing

[1] Comparing to the original fitness scape plot in Figure 4.1, this interpolated version has interpolation artifacts that it exhibits some "smearing" effect between the two big "mountains". This is because we used the MATLAB function 'TriScatteredInterp' for interpolating scattered data by linear interpolation. One can improve this by using other interpolation techniques.

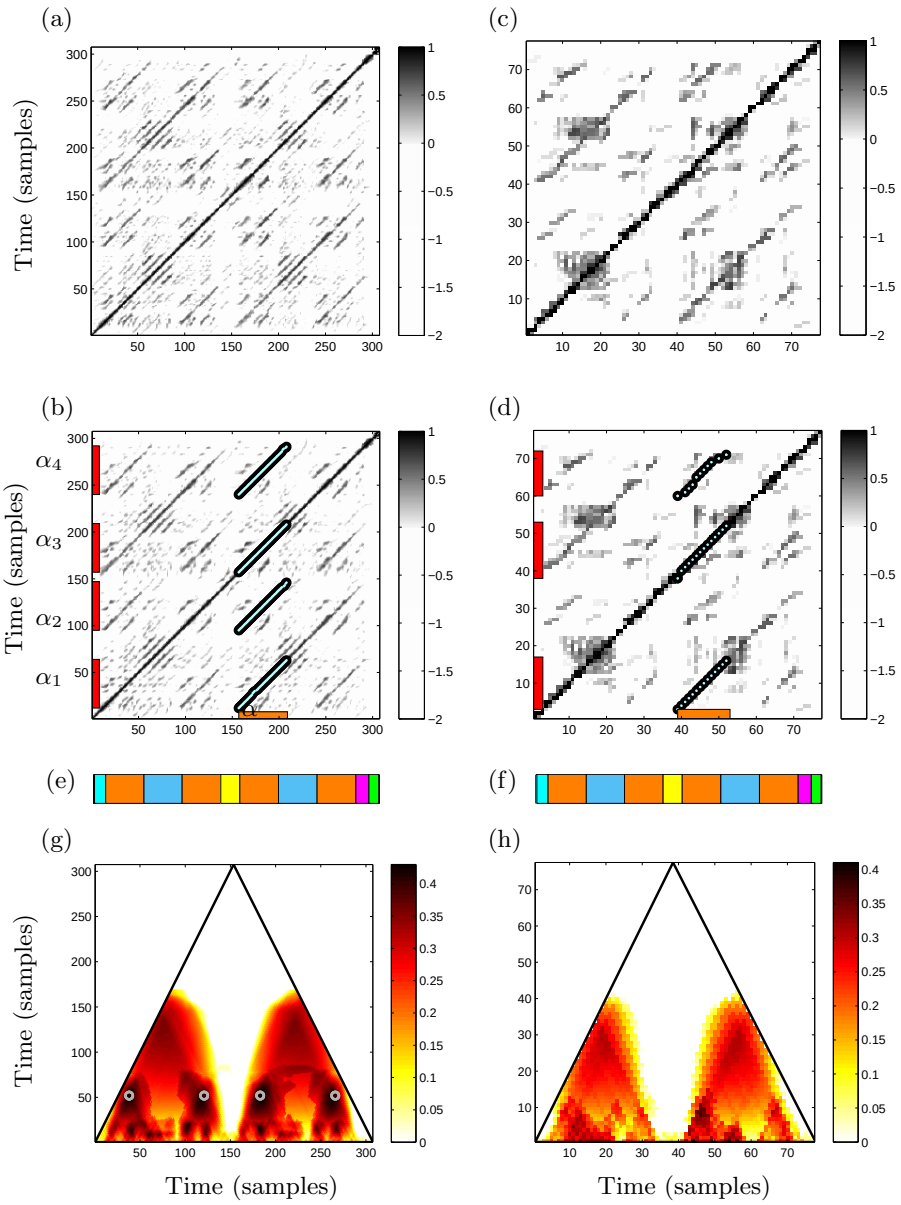

Figure 4.3: Illustration of the fitness computation and possible risks of the reduction of the SSM resolution. **(a)** SSM with 2 Hz resolution. **(b)** Optimal path family for the segment $\alpha = [158 : 209]$ using the SSM from (a). **(c)** SSM with 0.5 Hz resolution. **(d)** Optimal path family for the segment $\alpha = [158 : 209]$ using the SSM from (c). **(e)/(f)** Ground-truth segmentation. **(g)** Fitness scape plot obtained from (a). The segment $\alpha = [158 : 209]$ and all induced segments are indicated by circles. **(h)** Fitness scape plot obtained from (c). The resolution of the fitness scape plot is too low to reveal the thumbnail segment accurately.

the SSM resolution in a level-dependent way not going beyond a 1 Hz resolution. Using $(d_1, d_2, d_3, d_4) = (8, 4, 2, 1)$ as explained in Section 4.1.1, we use the finest resolution of 2 Hz only for the last step ($d_4 = 1$). For all previous steps we use a reduced SSM resolution of 1 Hz. As for practical computations, as we will discuss in Section 4.2 in more detail, the

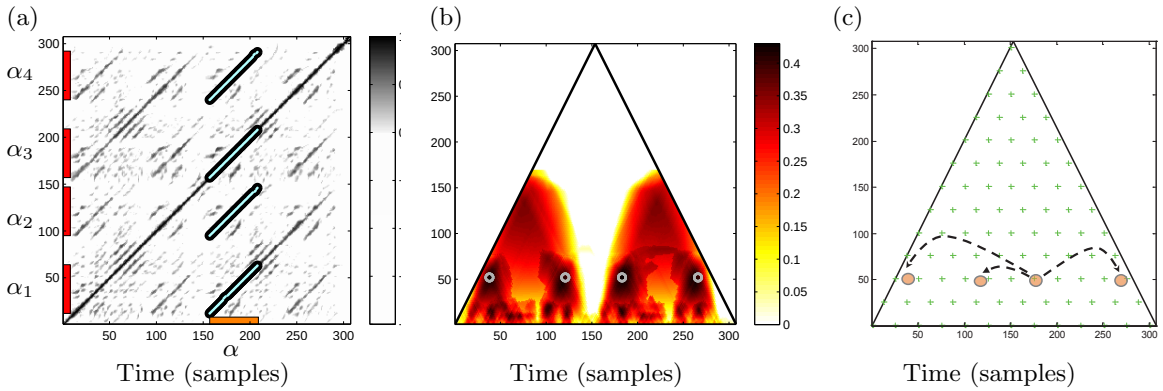

Figure 4.4: Illustration of the third acceleration strategy, fitness reuse. **(a)** Optimal path family and induced segments for the segment $\alpha = [158 : 209]$ using the SSM from Figure 4.3(a). **(b)** Fitness scape plot obtained from (a). The segment $\alpha = [158 : 209]$ and all induced segments are indicated by circles. **(c)** Exploiting path relations for fitness estimation of the induced segments.

resolution reduction becomes particularly important for the first step, where the fitness measure is evaluated for all points on the coarse scape plot grid. Here using a 1 Hz instead of a 2 Hz resolution yields in our experiments a speed up of roughly a factor of four (or even more) without any significant loss in the overall accuracy.

4.1.3 Acceleration by Fitness Reuse

We describe a third acceleration strategy, where we exploit the intrinsic properties of the fitness computation. Recall that in the computation of the fitness value $\varphi(\alpha)$ of a segment α, an optimal path family over α is determined, and the segments induced by this path family are the (approximate) repetitions of α (see Figure 4.4a). Now, the crucial observation is that each of the induced segments (being similar to α) also has more or less the same repetition relations as the segment α. As a result, the fitness of each of the induced segments is also close to the one of α. For example, in the case of $\alpha = [158 : 209]$ shown in Figure 4.4b, we obtain $\varphi(\alpha) \approx \varphi(\alpha_i)$, where α_i, $i \in [1 : 4]$, denote the four induced segments as defined in Section 4.1.2.

Based on this observation, we proceed as follows. When computing the fitness $\varphi(\alpha)$ for a segment α in the overall procedure, we reuse the value $\varphi(\alpha)$ as estimate for the fitness of all segments induced by α, as illustrated by Figure 4.4c. This information is stored in a suitable data structure. In this way, when we need to compute the fitness of another segment β at a later stage, we first check if there is already a segment β' in its suitable neighborhood, whose fitness value is available (either computed or estimated at a previous stage). If yes, we skip the fitness computation of β. Instead of β, we then use β' and its known fitness value for the subsequent steps. In our experiments, the above mentioned neighborhood is chosen to be two seconds.

(a) BEATLES

	OR	ML	ML+MR	ML+FR	ML+MR+FR
TimeOverall	61.67	2.49	0.69	1.41	0.55
SpeedUp	-	24.7	88.8	43.6	112.5
TimeLevel1	-	2.01	0.35	0.94	0.21
TimeLevel2	-	0.13	0.06	0.13	0.06
TimeLevel3	-	0.13	0.06	0.12	0.06
TimeLevel4	-	0.13	0.13	0.10	0.11
EvalSame	-	0.50	0.31	0.70	0.40
EvalSameTol	-	0.95	0.83	0.92	0.82
EvalThumbF	0.77	0.77	0.75	0.76	0.76

(b) MAZURKA

	OR	ML	ML+MR	ML+FR	ML+MR+FR
TimeOverall	143.67	5.68	1.12	2.88	0.77
SpeedUp	-	25.3	128.7	50.0	187.1
TimeLevel1	-	5.10	0.71	2.34	0.38
TimeLevel2	-	0.16	0.06	0.15	0.06
TimeLevel3	-	0.15	0.07	0.14	0.06
TimeLevel4	-	0.15	0.15	0.10	0.12
EvalSame	-	0.60	0.37	0.69	0.35
EvalSameTol	-	0.88	0.78	0.88	0.73
EvalThumbF	0.71	0.71	0.73	0.71	0.71

Table 4.1: Experimental results for the running time behavior and the accuracy of various acceleration strategies for audio thumbnailing, see text for explanation. The times indicate average running times per song given in seconds.

4.2 Experiments

We now describe our systematic experiments and investigate the effect of our acceleration strategies. Note that it is not in the scope of this chapter to discuss the specific parameter settings and to evaluate the actual thumbnailing procedure—this has been done in Chapter 3. Instead our goal is to illustrate to which extent the original procedure can be accelerated while obtaining the same thumbnail accuracy and visual impression of the scape plot as described in Chapter 3.

We have conducted our experiments on the basis of two datasets BEATLES and MAZURKA that have also been used in Chapter 3. The detail description of the two datasets can be found in Section 3.7.1. For both datasets, we derived thumbnail annotations from existing structure annotations, where a thumbnail annotation consists of an entire family of repeating segments with each segment serving equally well as a thumbnail.

In the original thumbnail procedure in Chapter 3, which we denoted by OR, an SSM resolution of 2 Hz is used. As for the multi-level acceleration procedure (ML) from Section 4.1.1, we use $I = 4$ with $(d_1, d_2, d_3, d_4) = (8, 4, 2, 1)$ and $M = 100$. In the multi-resolution fitness approach (MR), we use the setting as described in Section 4.1.2. Finally, for the fitness reuse strategy (FR), we use a neighborhood of two seconds as described in Section 4.1.3. Note that all acceleration procedures can be used in a combined fashion. As said before,

the specific settings are not crucial at this point and are chosen in a more conservative way to yield similar thumbnail results as the original procedure (and a similar visual impression of the interpolated scape plot to the original one, see Figure 4.3f). To demonstrate this, we consider three measures. The first two measure `EvalSame` and `EvalSameTol` indicate if the accelerated procedure yields exactly the same or nearly the same (with a tolerance of two seconds) thumbnail segment as the original approach. Note that these measure are not really suitable to measure the success since there is an entire family of valid thumbnail segments. Therefore, as a third measure, we use the same thumbnail F-measure `EvalThumbF` as described in Chapter 3 to show if the thumbnail obtained by accelerated procedures has the same quality as the thumbnail computed by the original approach.

Table 4.1 summarizes the experimental results. The algorithms have been implemented in Matlab R2012 (using C/C++ for the dynamic programming component), and tests were run on a computer with Intel Core i5-3470, 3.20 GHz CPU, 8 GByte RAM, under 64-bit Windows 7. In the following, we assume that the SSM on the finest level as used in Chapter 3 has been pre-computed and is given to all procedures as input. Then the times shown in Table 4.1 indicate the average running times per song given in seconds to derive the scape plot and the thumbnail from the SSM. For example, in the original procedure `OR`, it took in average 61.67 s to compute the thumbnail for the songs of `BEATLES` resulting in an overall thumbnail F-measure of `EvalThumbF` $= 0.77$. Applying the multi-level procedure `ML` results in an average running time of 2.49 s per song, which is a speed up of a factor of 24.7. The individual running times for the four levels are indicated in the next four rows of Table 4.1a. The full grid computation at the first level (using $d_1 = 8$) is `TimeLevel1` $= 2.01$ s and takes much more time then the subsequent refinement steps. As for the accuracy, the value `EvalSame` $= 0.50$ shows that the acceleration procedure yields exactly the same thumbnail as `OR` in only half of the cases. However, as indicated by `EvalSame` $= 0.95$, in most cases one only has a small shift in the computed segments. In the other cases, the acceleration procedure may yield a different thumbnail segment, which is in the same segment family. This is shown by the fact the that thumbnail F-measure for `ML` (and also for the other procedures) is basically the same as for `OR`. From an application point of view, such a segment is equally suited as thumbnail.

Now, let us have a look at the other acceleration procedures and their combinations. Using `MR` on top of `ML` increases the overall running time by an additional factor of roughly four. In particular, this speed up mainly results from the usage of a coarser resolution at the full grid computation at the first level. Similarly, `FR` on top of `ML` increases the overall running time by an additional factor of roughly two. As a main result of this chapter, our experiments show that one obtains by far the largest speed up when combining all three strategies without a substantial loss in the thumbnail accuracy. For example, in case of `BEATLES`, the combined approach needs an average running time of 0.55 s per song compared to 61.67 s of the original procedure, which is a speed up of a factor of 112.5. As shown in Table 4.1b, similar findings hold for the independent dataset `MAZURKA`. The reason for the slightly higher running times is that `MAZURKA` tends to comprise longer recordings than `BEATLES`. Altogether, this also demonstrates that our methods scale well to other types of music beyond popular music.

4.3 Further Notes

We have introduced three conceptually different acceleration strategies that lead to substantial speed-ups for a recent state-of-the-art thumbnailing procedure. These accelerations are important steps towards computing a thumbnail on-the-fly, which paves the way to applications in real-time services.

Here we need to point out that it is not always necessary to compute a full scape plot for a given audio recording in order to derive the thumbnail. If any prior information about the thumbnail could be given before the computation, we can use them to limit the computation in a certain range. For instance, if the appropriate length of the thumbnail is previously known, say 10 to 20 seconds for a popular song, the procedure can then be restricted to compute segments only in that length range. This will largely increase the computation efficiency and estimation accuracy for deriving the thumbnail. Therefore, in real time thumbnail estimation for large dataset, one can first restrict a proper range length before thumbnail searching, and then let the procedure compute the thumbnail only in the specified range of the scape plot.

Another possible improvement which remains for future work could be, for example, changing the main code from MATLAB to C code, since there is still MATLAB internal processing time which is not under users' full control. Therefore, reimplementing and optimizing the whole experimental pipeline in C code will further increase the efficiency.

Chapter 5

Visualization of Music Structure

In this chapter we present a visualization technique which shows all repetitive structures of a given audio recording. The content of this chapter closely follows the publication [92].

Finding the repetitive structure of a music recording has been a central and well-studied task within the wide area of music structure analysis, see, e. g., [29, 45, 72, 77, 98, 107] and the overview articles [30, 108]. Even though most of these approaches work well when repetitions are mutually similar, structure analysis becomes a hard and even ill-posed task when audio segments that refer to the same musical part exhibit pronounced musical variations. One way to circumvent such problems is to only visualize structural elements and their relations without explicitly extracting them. For example, in [36] self similarity matrices are used to visualize overall structural patterns or, in [138], repeating and related elements are indicated by arc diagrams.

In our previous chapters, we have introduced a procedure which can extract repetitive structures of a given music recording. In this chapter, we contribute to structure visualization by introducing a novel representation that can reveal hierarchical repetitive structures of a given music recording. Inspired by the work of Sapp [116], we use the concept of a 2D scape plot, where each point represents an audio segment by means of its center and length(see also Section 3.5). As our main contribution, we describe an automated procedure for assigning a color value to each point such that the repetitive structure of the music recording becomes apparent. On the one hand, we use the lightness component of the color to indicate the repetitiveness of the respective segment. This repetitiveness is expressed in terms of a fitness measure as introduced in Chapter 3. On the other hand, we use the hue component of the color to reveal the relations across different segments, where we introduce a function that maps related segments to similar hue values. As a result, one obtains a hierarchical structure visualization of the underlying music recording referred to *structure scape plot*, see Figure 5.4g for an example. We hope that this representation not only visually appeals to the reader, but also brings valuable and even surprising insights into the structural properties of a recording.

The remainder of this chapter is organized as follows. We first introduce some previous proposed visualization techniques for music structure in Section 5.1. Then, in Section 5.2, we review our underlying fitness measure and describe the corresponding fitness scape

plot. Then, in Section 5.3, we introduce our structure scape plot representation which is based on a novel distance measure to compare different segments as well as on an efficient grouping and coloring procedure. Based on a number of illustrative examples, we discuss benefits and limitations of our structure visualization in Section 5.4 and conclude with Section 5.6 by indicating directions for future work.

5.1 Background of Music Visualization

In the recent years, the availability of digital music data has grown dramatically on the internet. Thousands of new music tracks are uploaded to websites every day. Most of these tracks are provided with the descriptive metadata in textual form, such as song title, composer, etc. Usually such text descriptions cannot give users straight forward impressions about the music content. Therefore, many of the tools and algorithms in the field of music information retrieval are developed to search and navigate based on music content [138]. Among others, music visualization techniques can provide vivid graphical description revealing the relationship of different parts of music content, and therefore help users to acquire a better understanding about the music.

In this section, we briefly summarize some existing work on visualization techniques which focus on music content. A first technique called Arc Diagrams introduced by Wattenberg in [136] aims at visualizing repetitions in music. It reveals all repeated sections of a given recording by connecting them using arc bridges. The hierarchy of repeated sections can also be shown, since it use bigger arc identify longer repeated sections and smaller arc correspond to shorter sections which repeat inside the long section. This esthetically illustrates the repetitive structure of a music recording. However, when too may arcs are presented, the visualization became messy and thus difficult to understand. Therefore, one needs to control the trade-off between the detail of the repetitions shown and the complexity of the visualization. Based on the idea of the Arc Diagrams, some derived versions of visualizations focus not on the section level of music, but on much shorter levels e. g., beats or chords. In [65], Lamere visualize a music recording by connecting similar beats and chords using arc bridges. In this way, repeated sections can also be easily identified by the visualization, they appear mostly in the dense region of arcs since repeated sections usually consists of similar beats and chords. In addition, Wu and Bello also proposed a visualization method for music structure analysis in [138], which based on Arc Diagrams. Unlike other methods, this method does not make decisions about segment boundaries. Instead, it controls the granularity of the Arc Diagrams by a parameter which related with the length of the recurrence analysis window. There expriments show that such visualization technique can help users to identify basic musical forms.

A second technique is the scape plot introduced by Sapp in [117] and [116]. There, he introduced a triangle visualization of the local key information. The horizontal axis of the triangle indicates the time position of analysis window center, whereas the vertical axis indicates the analysis window length. Each point in this triangle corresponds a certain time section in the music, specified by the coordinate of the point. The color of the point tells the local key information for that section. In this way, the lowest level in the triangle is the chord of a single time frame, and the highest level represents the global key of the

whole piece. In our work of this chapter, we closely follow Sapp's visualization in [117]. We use the point position to indicate a section in music, and corresponding value indicates the repetitive structure.

Besides of the above mentioned techniques, there are many techniques capturing other aspects of music and partially reflect music structure. For example, Gómez and Bonada proposed several views [40] which aim at analyzing the tonal content of a piece of music such as chords and keys. One can partially visualize the music structure by their key grams and the key scape views. Their method is based on key estimation and tonality model in [39, 41]. Bergstrom et al. developed a visualization technique called Isochords that highlights the consonant intervals between notes and common chords in music [14]. It visualizes the note relationship, chord structure and progression by nodes and triangles. Chan et al. designed various ways to reveal the semantic structure for orchestral work in [22]. In addition, there are also websites provide visualization services of music data. For example, Songle website [1] which developed by AIST (National Institute of Advanced Industrial Science and Technology) Japan provide vivid visualization and navigation tools for understanding music structural segment, beat, melody line and chords [47]. For other visualization techniques, we refer to the overview report by Chan [21].

5.2 Fitness Scape Plot

In this section, we briefly summarize our fitness measure and fitness scape plot which is previously introduced in Chapter 3. Let $[1:N] = \{1, 2, \ldots N\}$ denote the (sampled) time axis of a given music recording. Then a segment is a subset $\alpha = [s:t] \subseteq [1:N]$ specified by its starting point s and its end point t. Let $|\alpha| := t - s + 1$ denote the length of segment α. A fitness measure has been introduced that assigns to each audio segment α. a fitness value $\varphi(\alpha) \in \mathbb{R}$ which simultaneously captures two aspects. First, it indicates how well the given segment explains other related segments. Second, it indicates how much of the overall music recording is covered by all these related segments.

In the computation of the fitness measure, an enhanced self-similarity matrix (SSM) is computed from the music recording based on chroma-based audio features (see also Chapter 2). It is well known that each path of the SSM (a stripe of high score running in parallel to the main diagonal) reveals the similarity of two segments (given by the two projections of the path onto the vertical axis and horizontal axis) [108]. Our main idea presented in Chapter 3 is to compute for each audio segment α a so-called *optimal path family* over α that simultaneously reveals the relations between α and all other similar segments. By projecting such optimal path family to the vertical axis, we get the corresponding induced segment family, where each element of this family defines a segment similar to α. The induced family of segments defines a segmentation of the audio recording.

Here we reuse the example in Chapter 3 Section 3.6 to illustrate the idea. We take the recording of Hungarian Dance No. 5 by Johannes Brahms, which has the musical form $A_1A_2B_1B_2CA_3B_3B_4$, see Figure 5.1a. Note that in this recording the second occurrence of the B-part (denoted by B_2) is played faster then the first occurrence (denoted by B_1).

[1] http://songle.jp/

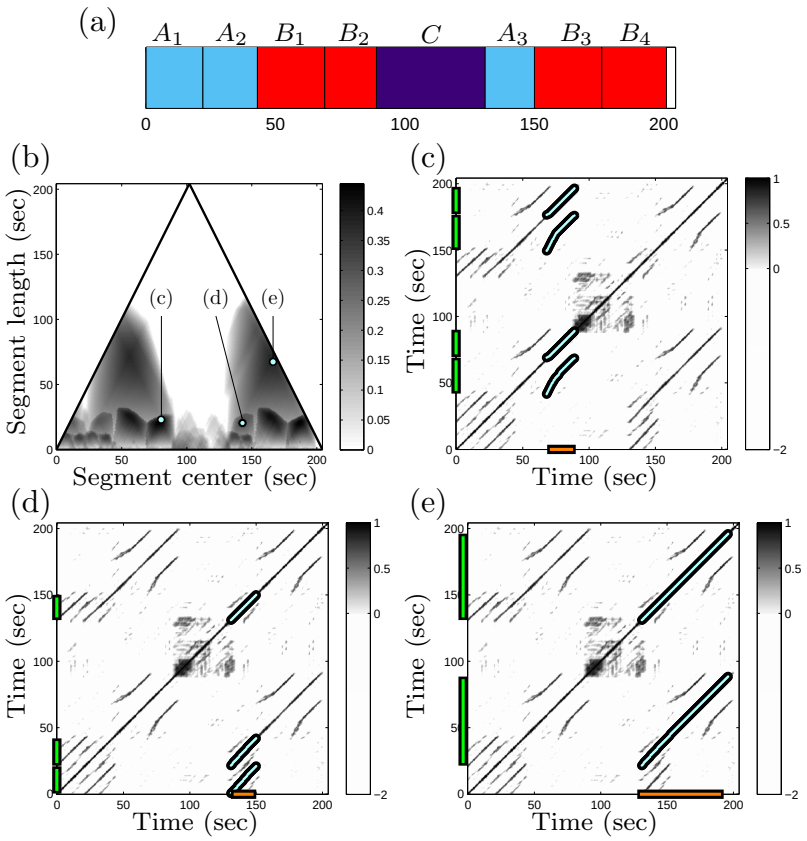

Figure 5.1: Various representations for an Ormandy recording of Brahms' Hungarian Dance No. 5. **(a)** Musical form $A_1A_2B_1B_2CA_3B_3B_4$. **(b)** Fitness scape plot. The remaining subfigures show the SSM with optimal path families for various segments α (horizontal axis) and induced segment families (vertical axis). **(c)** $\alpha = [68:89]$ (thumbnail, maximal fitness, corresponding to B_2). **(d)** $\alpha = [131:150]$ (corresponding to A_3). **(e)** $\alpha = [131:196]$ (corresponding to $A_3B_3B_4$).

This explains why the segment corresponding to B_2 is shorter than the one corresponding to B_1. Figure 5.1c shows an optimal path family (cyan stripes) for the B_2-segment $\alpha = [68:89]$ (horizontal axis) as well as the induced segment family (vertical axis). The induced segmentation consists of four segments corresponding to the four occurrences of the B-part in this recording. Similarly, Figure 5.1d shows the optimal path family for the segment $\alpha = [131:150]$ (corresponding to the A_3-part) and the induced segmentation (consisting of the three A-part segments). Finally, Figure 5.1e reveals that, for the long segment $\alpha = [131:196]$ (corresponding to $A_3B_3B_4$), there exists one similar segment (corresponding to $A_2B_1B_2$).

The fitness value of a given segment is derived from the corresponding optimal path family and the values of the underlying SSM. Intuitively, one considers the overall score accumulated by the path family and the total length covered by the induced segmentation. After a suitable normalization, the fitness is defined as the harmonic mean of of coverage and score. For further details, we refer to Chapter 3.

Our fitness scape plot, which is introduced in Chapter 3 Section 3.5, is a compact rep-

resentation for the entire music recording showing fitness $\varphi(\alpha)$ for each possible segment α. Using the center as horizontal coordinate and the length as vertical coordinate, each segment can be represented as a point in some triangular representation also referred to as *scape plot*. Such scape plots were original introduced by Sapp [116] to represent harmony in musical scores in a hierarchical way. In our context, we define a scape plot Φ by setting $\Phi(c(\alpha), |\alpha|) := \varphi(\alpha)$ for segment α. Figure 5.1b shows a visualization of the fitness scape plot for our Brahms example, where the fitness is represented by a lightness grayscale ranging from white (fitness is zero) to black (fitness is high). The points corresponding to the three segments discussed above are marked within the scape plot by small circles. For example, the segment $\alpha = [68:89]$ (corresponding to B_2) has the scape plot coordinates $c(\alpha) = 78.5$ (horizontal axis) and $|\alpha| = 22$ (vertical axis). Actually, this segment has the highest fitness among all possible segments and is also referred to as *thumbnail* [91].

The fitness scape plot represents the repetitiveness of each segment in a compact and hierarchical form. For example, in our Brahms example, the repeating segments corresponding to the A-parts and B-parts are reflected by local maxima in the scape plot. Also the repetitions of the superordinate segments corresponding to ABB are captured by the plot. However, so far, the visualization of the fitness scape plot does not reveal the relations *across* different segments. In other words, nothing is said about groups of pairwise similar segment corresponding to the various musical parts.

5.3 Structure Scape Plot

Actually, the grouping information is implicitly encoded by the optimal path families underlying the fitness measure. To make these relations more explicit, we now extend the grayscale of the fitness scape plot by a color component that reflects the cross-segment relations. Based on the induced segmentations, we first introduce a distance measure that allows for comparing two arbitrary segments (Section 5.3.1). Then the objective is to map similar segments to similar colors and dissimilar segments to distinct colors. In the following, we proceed in several steps including a color mapping step (Section 5.3.2), a point sampling and interpolation step (Section 5.3.3), and a color combination step (Section 5.3.4). The overall pipeline of our procedure is also illustrated by Figure 5.4.

5.3.1 Segment Distance Measure

Recall from Section 5.2 that for a given segment α there is an optimal path family along with an induced segment family, where each segment of this family is similar to α. Let $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ denote the induced segment family of α, then the segments α_k, $k \in [1 : K]$, can be thought of as the (approximate) repetitions of α. Note that, by definition, overlaps between repetitions are not allowed.

Now, let α and β be two arbitrary segments. Intuitively, we consider these two segments to be close if they are approximately repetitions of each other (or at least if some repetitions of α and β have a substantial overlap), otherwise α and β are considered to be far apart. More precisely, let $\mathcal{A} = \{\alpha_1, \ldots, \alpha_K\}$ and $\mathcal{B} = \{\beta_1, \ldots, \beta_L\}$ be the respective induced

segment families. Then, we define the distance $\delta(\alpha, \beta)$ between α and β to be

$$\delta(\alpha, \beta) := 1 - \max_{k \in [1:K], \ell \in [1:L]} \frac{|\alpha_k \cap \beta_\ell|}{|\alpha_k \cup \beta_\ell|}, \tag{5.1}$$

see also Figure 5.2 for an illustration. In other words, the distance is obtained by subtracting the maximal overlap (relative to the union) over all repetitions of α and β from the value 1. For example, the B_1-segment and B_2-segment for the Brahms recording have a small distance (close to zero) since the induced segment families more or less coincide (consisting of the four B-part segments). In contrast the B_1-segment and the A_1-segment have a large distance (close to one) since none of their repetitions have a substantial overlap.

5.3.2 Color Mapping

Based on the distance measure δ, we now introduce a procedure for mapping the scape plot points (segments) to color values in such a way that distance relations are preserved. To this end, we first need to specify a suitable color space. Because of its perceptual relevance, we revert to the HSL model, which is a cylindric parametrization of the RGB color space [50]. Here the angle coordinate $H \in [0, 360]$ (given in degrees) refers to the hue, the coordinate $S \in [0, 1]$ to the saturation, and the coordinate $L \in [0, 1]$ (with 0 being black and 1 being white) to the lightness of the color. To obtain "full" saturated colors, we fix the parameter $S = 1$. Figure 5.3 shows the color space for $S = 1$ spanned by the coordinates H and L. Note that the hue angle coordinates $H = 0$ and $H = 360$ encode the same color (by definition this is the color "red"). In the following, we reserve the lightness coordinate to represent the fitness value and only use the hue coordinate to represent the cross-segment relationships.

The problem of mapping the scape plot points to the hue color coordinates (which topologically corresponds to the unit circle) in a distance preserving way can be seen as an

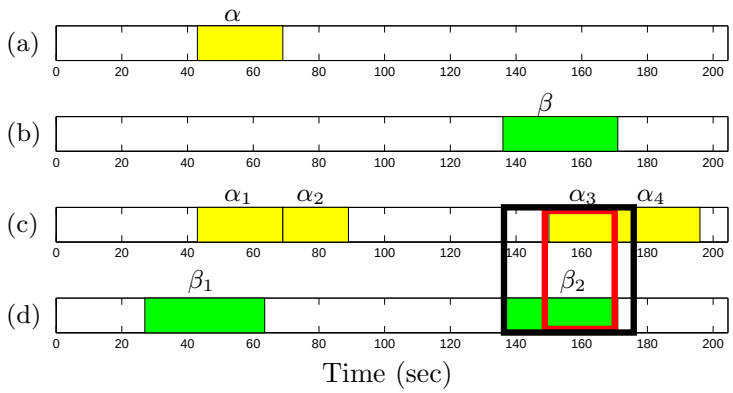

Figure 5.2: Illustration of the computation of the distance measures $\delta(\alpha, \beta)$ used to compare two segments α (shown in **(a)**) and β (shown in **(b)**). The respective induced segment families are shown in **(c)** and **(d)**, respectively. The black box indicates the union and the red box the overlap of the two segments which are used to compute distance value $\delta(\alpha, \beta)$.

instance of *multidimensional scaling* (MDS), see [15]. Generally, MDS refers to a family of related techniques which allow for mapping a set of points with pairwise distance values onto a low-dimensional Euclidean space (often dimension 2 or 3 for visualization purposes) such that the distances between the original points are approximated by the Euclidean distances of the mapped points.

In the following, we use basic MDS techniques to map the scape plot points onto the unit circle (representing the hue color space). Let M denote the number of scape plot points to be considered in the mapping, see Figure 5.4b. First, we compute an $M \times M$-distance matrix Δ by comparing the M points in a pairwise fashion using δ. Next, we perform a principal component analysis (PCA) of Δ and consider the two eigenvectors corresponding to the two largest eigenvalues. The columns of Δ (which are indexed by the M scape plot points) are then projected onto the two-dimensional Euclidean space defined by these two eigenvectors, see Figure 5.4c. Using PCA, the variance across the mapped column vectors is maximized. Therefore, scape plot points that have a distinct distance distribution to the other points (encoded by its respective column vectors) are likely to be mapped to different regions in the 2D space, see [15] for details. Furthermore, as shown in Figure 5.4c, the projected points are usually distributed in a circular fashion (even though this is not guaranteed and crucially depends on the distance distributions of the original points). Finally, we normalize the projected points with respect to the Euclidean norm to obtain points on the unit circle, which yields angle parameters that are associated to hue values, see Figure 5.4d. Figure 5.4e shows the original scape plot points colored with the derived hue values.

5.3.3 Sampling and Interpolation

Using all scape plot points in the described color mapping procedure may be problematic because of two reasons. Firstly, using a large number M of points would not only make the computation of the $M \times M$ distance matrix Δ but also of the subsequent PCA rather expensive. Therefore, the number M of used points should be kept small. Secondly, using all scape plot points may over-represent segments of short lengths that are located in the lower part of the triangular scape plot. As a result, the distance relations of the short segments may dominate the selection of the eigenvectors obtained in the PCA step.

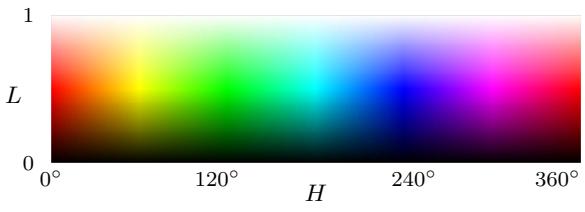

Figure 5.3: Cylindric HSL (hue, saturation, lightness) color representation. The figure shows only the outside surface of the cylinder corresponding to the saturation $S = 1$.

Therefore, we only choose a suitable subset of scape plot points, also referred to as *anchor points*, and then transfer the obtained hue color information to the other points using interpolation techniques [2].

Note that scape plot points of higher fitness are structurally more relevant than scape plot points of lower fitness. Therefore, in the anchor point selection step, we sample the scape plot by taking the fitness into account. To this end, we use a greedy procedure that consists of two steps. Firstly, we select the scape plot point of maximal fitness as an anchor point. Secondly, around this anchor point, we specify a neighborhood of size $\xi > 0$, and set the fitness values of all points in this neighborhood to zero excluding them for the subsequent procedure. The role of the neighborhood is to avoid a sampling that is locally too dense. This procedure is repeated until either all of the remaining scape plot points have a fitness of zero, or until a specified maximal number of points M_0 is reached, see also Figure 5.4b.

Sometimes the fitness values of short segments are rather "noisy." This may also have musical reasons since such segments often correspond to highly repetitive fragments like a short riff or a single chord of dominant harmony. Therefore, it is often beneficial to exclude such short segments in the anchor point selection by only considering scape plot points whose length coordinate lies above a certain lower bound $\theta > 0$. The influence of the parameters M_0, ξ, and θ on the resulting number of anchor points M is discussed in Section 5.4.

The color mapping as described in Section 5.3.2 is now applied only to the anchor points. In the next step, the color information is transferred to arbitrary scape plot points by simply interpolating color values of the nearest neighborhood anchor points. However, since the hue values live on a unit circle (rather than in the two-dimensional Euclidean space), one needs to use spherical interpolation instead of linear interpolation. Figure 5.4f shows the interpolation result obtained from the anchor points of Figure 5.4e.

5.3.4 Color Combination

So far, we have derived two scape plot visualizations: one indicating the repetitive properties (fitness value represented by lightness, see Figure 5.4a) and the other indicating the cross-segment relations (represented by hue colors, see Figure 5.4f). We now combine this information within a single scape plot representation, which we also refer to as *structure scape plot*. To this end, we first linearly map the fitness values onto the lightness parameter space $[0, 1]$ of the HSL model such that $L = 1$ (white) corresponds to the fitness value 0 and $L = 0$ (black) to the maximal fitness value occurring in the fitness scape plot. Furthermore, by rotating the hue parameter space (unit circle) we normalize the color assignment such that the thumbnail (fitness-maximizing scape plot point) is mapped to the color "red" (angle $H = 0$). Finally, for each scape plot point we use the saturation $S = 1$, the computed lightness L, and the normalized hue angle H to obtain a single color value.

Figure 5.4g shows the final result of the structure scape plot for our Brahms example. Note

[2] Note that the technique introduced in this section is different from the sampling and interpolation technique introduced for acceleration strategy (Section 4.1.1)

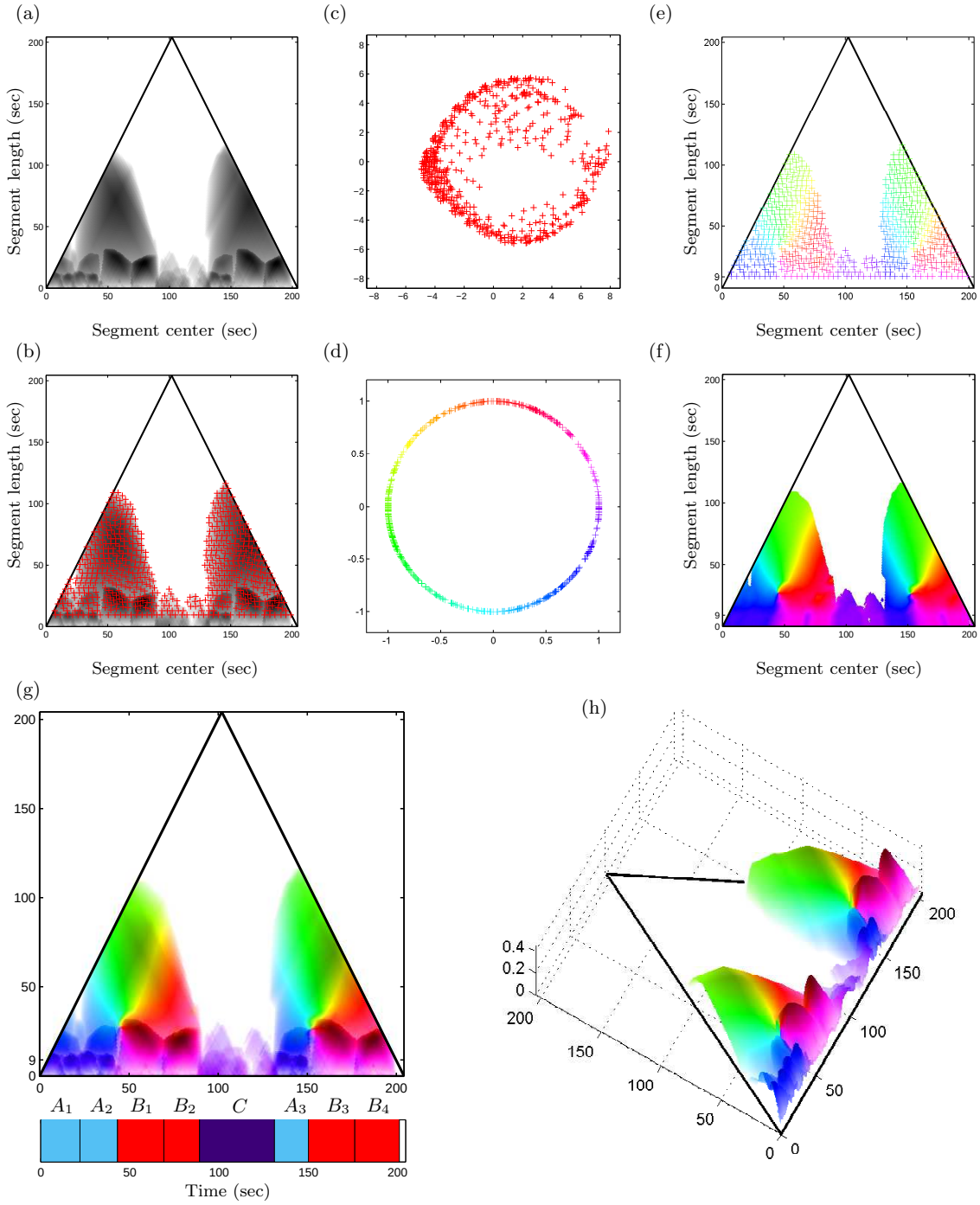

Figure 5.4: Illustration of the pipeline for computing the structure scape plot for Brahms. **(a)** Fitness scape plot. **(b)** Fitness scape plot with sampled anchor points. **(c)** Anchor points projected onto the first two principal components. **(d)** Anchor points projected to the unit circle colored with the resulting hue value. **(e)** Hue-colored anchor points. **(f)** Hue-colored scape plot using interpolation techniques. **(g)** Structure scape plot combining fitness (lightness) and cross-segment relation (hue) information. **(h)** 3D version of (g).

that the four B-part segments (repetitions of the B_2-thumbnail) are represented by red, the three A-part segments by blue, and the superordinate two ABB-part segments by green. Furthermore, the visualization reveals some substructures of the A-parts, each actually consisting of two (approximate) repetitions. Finally, note that smaller segments within the C-part are assigned to the color violet. Since the C-part contains many fragments sharing the same harmony, our procedure has captured some repetitiveness also in this middle part.

Besides the 2D plots mentioned above, we also computed the 3D version of structure scape plot as shown in Figure 5.4h. Based on the 2D version shown in Figure 5.4g, we additionally assign the height of each point by its fitness value. In this way, the repetitive structures can be more clearly visualized, since the quality of repetition is now not only revealed by the color but also by the height of points. Furthermore, by comparing the shape of those colored mountains, one can clearly see the subtle difference across the repetitions.

5.4 Examples and Discussion

In this section, we indicate the potential and some limitations of our visualization procedure by discussing representative examples. In our experiments, we used audio recordings considering popular music as well as classical music. On the one hand, we employed the dataset consisting of recordings of the 12 studio albums by "The Beatles" using the structure annotations as described by [82]. On the other hand, we used the complete Rubinstein (1966) recordings of the 49 Mazurkas composed by Frédéric Chopin, where we manually generated some structure annotations for each piece. Note that these annotations are not needed to generate the structure scape plots, but are only used to compare our visualizations with some sort of ground truth. As mentioned in the introduction, the purpose of the scape plot visualizations is to yield a compact and intuitive representation without the necessity of explicitly extracting the structure.

As for the parameter settings, we choose the sampling parameter M_0, the neighborhood parameter ξ, and the length lower bound parameter θ in a relative fashion depending on the duration of the respective music recording. In particular, we determine the upper bound M_0 and the neighborhood parameter ξ to result in a number M of anchor points ranging between 200 and 250 for each recording. Furthermore, the lower bound θ was set to correspond to 5-7 % of the recording's total duration. Figure 5.5 and Figure 5.6 show structure scape plots for some representative music recordings. For example, Figure 5.5(a,b) show the scape plot for a Rubinstein performance of Chopin's Mazurka Op. 17 No. 3. The five A-part segments, which also comprise the thumbnail, are represented by red. Furthermore, the three B-part segments are indicated by a lighter orange color, and the superordinate ABA-part segments are represented by green. Also substructures of the A-part segments are visible: indeed each A-part consists of two similar subparts. In the 3D visualization in Figure 5.5b, the substructures of A part can be seen much clearly, as the red peaks which correspond to the sub-parts exhibit the largest height (according to their fitness values). Interestingly, the segments corresponding to the C- and the two D-parts are all represented by pink. Actually this is musically meaningful, since each of

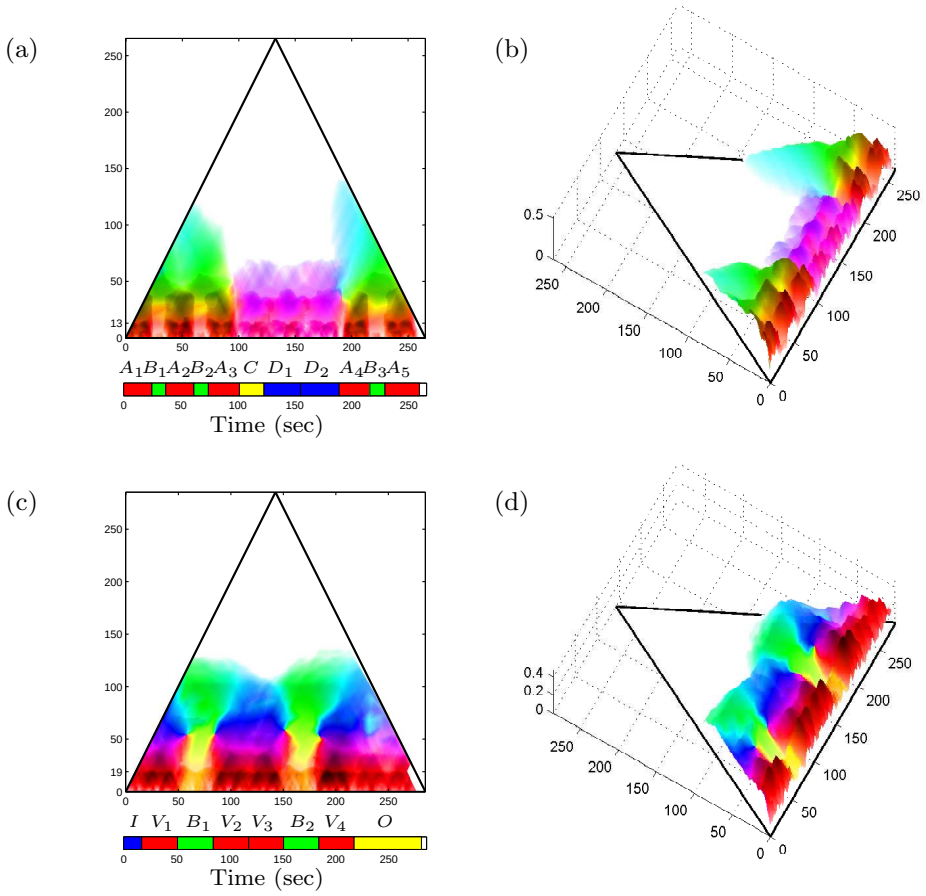

Figure 5.5: Structure scape plots in 2D and 3D versions as well as structure annotations for recordings of various pieces. **(a)/(b)** Chopin Mazurka Op. 17 No. 3. **(c)/(d)** Beatles song "While My Guitar Gently Weeps."

the two repeating D-parts is only a slight extension of the C-part. Again, we can observe the similar phenomena in the 3D version.

Figure 5.5(c,d) visualize the structure scape plot for the Beatles song "While My Guitar Gently Weeps." Also in this example, the structure scape plot nicely reflects the overall musical form. Each of the four verse segments (V-part) consists of two (approximately) repeating subparts, say $V = WW$. Actually, the intro also corresponds to such a subpart ($I = W$) and the outro corresponds to three of these subparts ($O = WWW$), which also explains the red coloring of these segments. Furthermore, the color blue corresponds to WWW-segments and the color green to VBV-segments. Again, the 3D version of the scape plot in Figure 5.5d present the same structural information, but the substructures of V and O parts can be visualized more clearly (as shown by those red peaks).

The structure scape plot of a recording of the Mazurka Op. 33 No. 3 is shown in Figure 5.6(e,f) which indicate a number of substructures not reflected in the structure annotation (see both A parts). Finally, Figure 5.6(g,h) correctly reproduce the overall structure

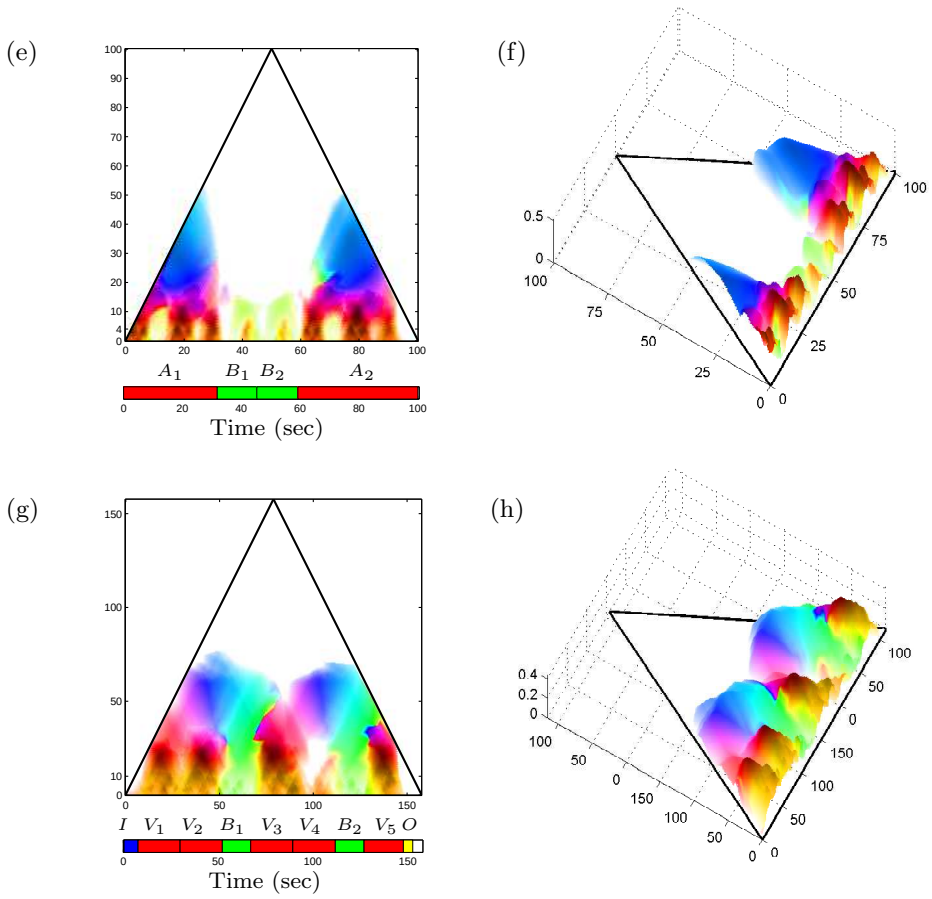

Figure 5.6: Structure scape plots in 2D and 3D versions as well as structure annotations for recordings of various pieces. **(e)/(f)** Chopin Mazurka Op. 33 No. 3. **(g)/(h)** Beatles song "You Can't Do That."

of the Beatles song "You Can't Do That." Only the V_4-segment has not been captured well. Actually, V_4 corresponds to an instrumental section with some vocal interjections, which make the V_4-segment spectrally quite different to the other four V-part segments.

5.5 Problem Discussion

In this section, we discuss some limitations and problems that may occur in our visualization approach. As an illustrating example, we consider the Beatles song "Hello Goodbye." Figure 5.7b shows the structure scape plot using our standard parameter setting as described above. The red color corresponds to the four VR-part segments, which also comprise the thumbail. However, the individual V-part and R-part segments are all represented by green and are not distinguishable. The reason for this is that the lower bound for the anchor points was set to $\theta = 14$ seconds, which is too high to capture the finer structures. By decreasing this parameter to $\theta = 10$ seconds, V-part and R-part segments

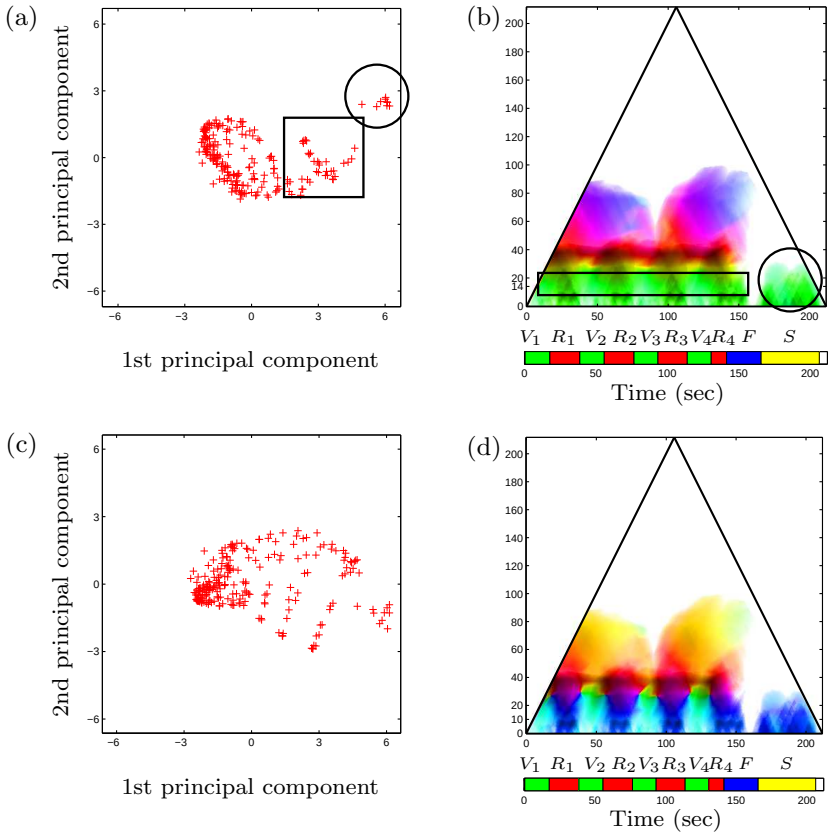

Figure 5.7: Anchor points projected onto the first two principal components (left) and resulting structure scape plot (right) for the Beatles song "Hello Goodbye." **(a)/(b)** Using $\theta = 14$ seconds. **(c)/(d)** Using $\theta = 10$ seconds.

are separated, see Figure 5.7d. As this example shows, the choice of the parameter θ may have a significant impact on the final visualization. The Beatles example also indicates a second problem that may arise in our color mapping procedure. Usually, the anchor points projected to the two principal components are homogeneously distributed along the unit circle as in our Brahms example, see Figure 5.4c. Therefore, projecting these points to the unit circle (to yield the desired hue values) does not destroy too much of the neighborhood relations. However, in the Beatles example, the projected anchor points are rather scattered in the two-dimensional Euclidean space with some outliers as indicated by the boxed and circled points shown in Figure 5.7a. Therefore, projecting these points onto the unit circle may result in the same hue value for anchor points that are actually far apart. This explains, why the substructures within the S-part are mapped to the same color as substructures of the VR-part, see Figure 5.7b.

5.6 Further Notes

The limitations and problems mentioned in previous section actually point out some future research directions. Possible improvements of the color mapping step may be achieved by applying more involved generalized multidimensional scaling techniques which directly map the anchor points to a smooth manifold (in our case the unit circle). Also, the one-dimensional hue color space may not suffice to suitable capture more intricate cross-segment relations. Here, a more flexible usage of the color space may help to better represent more complex structures. So far, we have only given a qualitative evaluation to demonstrate the potential of our techniques. In this context, user studies may be necessary to better understand the actual user needs and the applicability of our concepts. Besides introducing a novel segment distance function as well as a grouping and coloring procedure, the main contribution of this chapter was to introduce the concept of a structure scape plot for visualizing repetitive structures of music recordings. We hope that our visualization is not only aesthetically appealing, but also may allow a user to explore and browse musical structures in novel ways.

Chapter 6

Analyzing Music Recordings in Sonata Form

In previous chapters, we have introduced our automated procedure that analyzes and extracts repetitive structures of a given music recording. So far, the music pieces analyzed in previous chapters are in small-scale structures such as popular music or piano Mazurkas. In this chapter, we perform automated structure analysis on music pieces in *sonata form*, which exhibit large-scale and hierarchical structures. The material of this chapter closely follows publication [56].

The sonata form has been one of the most important large-scale musical structures used since the early Classical period. The first movements of symphonies and sonatas follow the sonata form. Typically, the most basic sonata form consists of an *exposition* (E), and a repetition of exposition, a *development* (D), and a *recapitulation* (R). Sometimes, one can find an additional introduction (I) and a closing coda (C), thus yielding the form IE_1E_2DRC [1] . In particular, the exposition and the recapitulation stand in close relation to each other both containing two subsequent contrasting subject groups (often simply referred to as first and second theme) connected by some transition. However, in the recapitulation, these elements are musically altered compared to their occurrence in the exposition. In particular, the second subject group appears in a modulated form, see [69] for details. In the field of musicology, many methodical analysis of music pieces written in sonata form have already been performed (see for example, the analysis of Beethoven piano sonatas by Tovey in [131]). However, due to the hierarchical nature and complex harmonic organization presented in sonata form, it remains a challenging task for automated procedures to analyze music pieces in sonata form.

In this chapter, we present automated methods for analyzing and deriving the structure for a given audio recording of a piece of music in sonata form. This task is a specific case of audio structure analysis. In most previous work, the considered structural parts for pop music or small-scale classical music are often assumed to have a duration between 10 and 60 seconds, for example a chorus section in popular music often has a duration less than 30

[1] To describe a musical from, one often uses the capital letters to refer to musical parts, where repeating parts are denoted by the same letter. The subscripts indicate the order of repeated occurrences.

seconds. Such assumption about structural parts usually results in some kind of medium-grained analysis. Also, repeating parts are often assumed to be quite similar in tempo and harmony, where only differences in timbre and instrumentation are allowed. Furthermore, *global* modulations can be handled well by cyclic shifts of chroma-based audio features [45]. When dealing with the sonata form, certain aspects become more complex and therefore challenging for automated analysis. First, the duration of musical parts is much longer, often exceeding two minutes. Even though the recapitulation can be considered as some kind of repetition of the exposition, significant local differences that may last for a couple of seconds or even 20 seconds may exist between these parts. Furthermore, there may be additional or missing sub-structures as well as relative tempo differences between the exposition and recapitulation. Finally, these two parts reveal differences in form of *local* modulations that cannot be handled by a global cyclic chroma shift.

In this chapter, we show how our automated structure analysis procedure deals with challenges mentioned above. The remainder of this chapter is organized as follows. Firstly, in Section 6.1 we summarize the related music theory that is needed for understanding the harmonic relationship and general structure of the sonata form. Then, we introduce our automated analysis approch which proceed in two steps. In the first step, we describe how we adapted our repetition detection procedure to identify the large-scale sections of exposition and the recapitulation(Section 6.2). To deal with local modulations, we use the concept of transposition-invariant self-similarity matrices [89]. In the second step, we reveal finer substructures in exposition and recapitulation by capturing relative modulation differences between the first and the second subject groups (Section 6.3). As for the evaluation of the two steps, we consider the first movements in sonata form of the piano sonatas by Ludwig van Beethoven, which constitutes a challenging and musically outstanding collection of works [131]. Besides some quantitative evaluation, we also contribute with a novel visualization that not only indicates the benefits and limitations of our methods, but also yields some interesting musical insights into the data.

6.1 Sonata Form

In this section, we summarize some basic music theory which is needed for understanding the sonata form. The content below are mainly gathered from the following music theory and analysis books [13, 69, 131].

The musical form refers to the overall structure of a piece of music by its repeating and contrasting parts, which stand in certain relations to each other [86]. The sonata form (also named as sonata-allegro form or first movement form) is a musical structure which has been widely used since the early Classical period. The sonata form gives a composition a specific identity and has been widely used for the first movements in symphonies, sonatas, concertos, string quartets, and so on. The formal definition focuses on the thematic and harmonic content which is present in different sections. We will discuss each section in detail in the following paragraphs.

Figure 6.1 gives an overview on the typical structure of the sonata form. We first introduce the large scale structure. Generally, the most basic sonata form consists of three large sections including the exposition, the development, and the recapitulation. Usually the

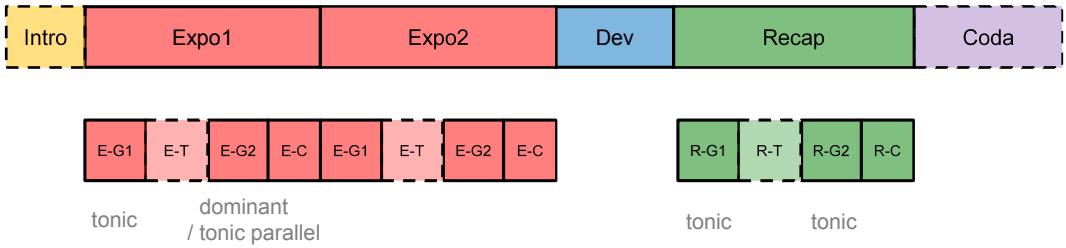

Figure 6.1: Overview of the structural organization of the sonata form. The upper part shows the coarse structure, the lower part shows the fine structure of the exposition and recapitulation. The gray colored text at the bottom describes the tonal properties of the fine structure. The sections surrounded by dashed boarders are optional.

exposition section repeats itself one time right after it reaches its end. Sometimes, the sonata form may additionally includes an introduction section at the beginning of the piece or a coda section at the end of the piece (see top part of Figure 6.1). From a functionality point of view, the themes and harmonic organization of the tonal material are first presented in the exposition, then get elaborated and contrasted in the development, and in the end get resolved in the recapitulation. The optional introduction section may or may not contain material related with the exposition, whereas the optional coda section usually contains material from the movement. Such main sections of sonata form are large scale structures that may have rather long passages, and they can be further divided into several smaller sections as substructures. In this chapter, we consider the large scale sections as the coarse structure of the sonata form. In addition, the subsections of these large sections are considered to be the fine structures of sonata form.

Now we look closer into the exposition section and recapitulation section to explain their internal substructures. The exposition typically consist of four small sections which have different semantic meaning. The primary musical content is presented in two contrasting subject groups. Here, in the first subject group ($G1$) the music is in the *tonic* (the home key) of the movement, whereas in the second subject group ($G2$) it is in the *dominant* (for major sonatas) or in the *tonic parallel* (for minor sonatas). To make the harmonic content transfer smoothly from the first group to the second group, there is usually a *modulating transition* (T) between them. At the end of the exposition there is often an additional closing theme or *codetta* (C). The recapitulation contains similar sub-parts as the exposition, however it includes some important harmonic changes. In the following discussion, we denote the four sub-parts in the exposition by *E-G*1, *E-T*, *E-G*2, and *E-C*. The sub-parts in the recapitulation are denoted by *R-G*1, *R-T*, *R-G*2, and *R-C* accordingly (see bottom part of Figure 6.1 for illustration). The first subject groups *E-G*1 and *R-G*1 are typically repeated in more or less the same way and are both appearing in the tonic. However, in contrast to *E-G*2 appearing in the dominant or tonic parallel, the second subject group *R-G*2 appears in the tonic. Furthermore, compared to *E-T*, the transition *R-T* is often extended, sometimes even presenting new material and local modulations, see [69] for details. Note that the described structure indicates a tendency rather then being a strict rule. For example, in many of the Beethoven sonatas, the transition between the two subject groups are omitted. There are also many other

exceptions and modifications as the following examples demonstrate.

As for the development, it does not have subject groups as the exposition and the recapitulation. Generally, it presents one or more themes from the exposition, but alters these themes and contrasts them. Sometimes, new themes or new material are also included in the development. It typically starts in the same key as the end of the exposition. Then it moves to different keys through the whole section. In this thesis, we do not analyze the fine structure of the development because its harmonic content is very complex. We consider the development as a whole section in the coarse structure of the sonata form without analyzing its internal fine structure. The fine structure analysis is performed on the exposition and the recapitulation.

In the following, we introduce the automated method of analyzing and deriving the structure for music pieces in sonata form. We proceed our structure analysis in two steps. We derive the main sections as the coarse structure of sonata form in the first step. Then we further analyze the fine structures for the exposition and the recapitulation in the second step.

6.2 Coarse Structure

In the first step, our goal is to split up a given music recording into segments that correspond to the large-scale musical structure of the sonata form. On this coarse level, we assume that the recapitulation is basically a repetition of the exposition, where the local deviations are to be neglected. Thus, the sonata form IE_1E_2DRC is dominated by the three repeating parts E_1, E_2, and R.

To find the most repetitive segment of a music recording, we apply and adjust our repetition detection procedure (thumbnailing procedure) proposed in Chapter 3. To this end, the music recording is first converted into a sequence of chroma-based audio features[2], which relate to harmonic and melodic properties [90]. From this sequence, a suitably enhanced self-similarity matrix (SSM) is derived. In our case, we apply in the SSM calculation a relatively long smoothing filter of 12 seconds, which allows us to better bridge local differences in repeating segments. Furthermore, to deal with local modulations, we use a *transposition-invariant* version of the SSM, see [89]. To compute such a matrix, one compares the chroma feature sequence with cyclically shifted versions of itself, see [45]. For each of the twelve possible chroma shifts, one obtains a similarity matrix. The transposition-invariant matrix is then obtained by taking the entry-wise maximum over the twelve matrices (see also Chapter 2, Section 2.2.3). Furthermore, storing the shift index which yields the maximum similarity for each entry results in another matrix referred to as *transposition index matrix*, which will be used in Section 6.3. Based on such transposition-invariant SSM, we apply our thumbnailing procedure to compute for each audio segment a fitness value that expresses how well the given segment explains other related segments (also called induced segments) in the music recording. These relations

[2]In our scenario, we use a chroma variant referred to as `CENS` features, which are part of the Chroma Toolbox http://www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/. Using a long smoothing window of four seconds and a coarse feature resolution of 1 Hz, we obtain features that show a high degree of robustness to smaller deviations, see [90] for details.

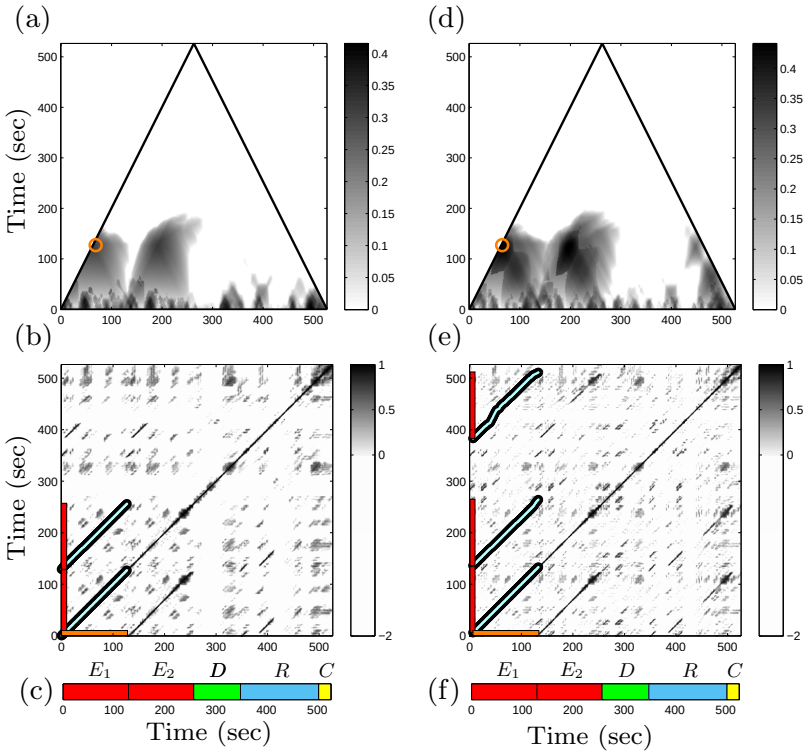

Figure 6.2: Thumbnailing procedure for `Op031No2-01` ("Tempest"). **(a)/(d)** Scape plot representation using an SSM without/with transposition invariance. **(b)/(e)** SSM without/with transposition invariance along with the optimizing path family (cyan), the thumbnail segment (indicated on the horizontal axis) and induced segments (indicated on the vertical axis). **(c)/(f)** Ground-truth segmentation.

are expressed by a so-called path family over the given segment. The thumbnail is then defined as the segment that maximizes the fitness. Furthermore, a triangular scape plot representation is computed, which shows the fitness of all segments and yields a compact high-level view on the structural properties of the entire audio recording.

We expect that the thumbnail segment, at least on the coarse level, should correspond to the exposition (E_1), while the induced segments should correspond to the repeating exposition (E_2) and the recapitulation (R). To illustrate this, we consider as our running example a Barenboim recording of the first movement of Beethoven's piano sonata Op. 31, No. 2 ("Tempest"), see Figure 6.2. In the following, we also use the identifier `Op031No2-01` to refer to this movement. Being in the sonata form, the coarse musical form of this movement is E_1E_2DRC. Even though R is some kind of repetition of E_1, there are significant musical differences. For example, the first subject group in R is modified and extended by an additional section not present in E_1, and the second subject group in R is transposed five semitones upwards (and later transposed seven semitones downwards) relative to the second subject group in E_1. In Figure 6.2, the scape plot representation (top) and SSM along with the ground truth segmentation (bottom) are shown for our example, where on the left an SSM without and on the right an SSM with transposition invariance has been used. In both cases, the thumbnail segment corresponds to part E_1. However, without using transposition-invariance, the recapitulation is not among the

No.	Piece ID	GT Musical Form	P	R	F
1	Op002No1-01	E_1E_2DR	0.99	0.90	0.90
2	Op002No2-01	E_1E_2DR	0.99	0.96	0.96
3	Op002No3-01	E_1E_2DRC	0.95	0.97	0.97
4	Op007-01	E_1E_2DRC	1.00	0.99	0.99
5	Op010No1-01	E_1E_2DR	0.99	0.93	0.93
6	Op010No2-01	E_1E_2DR	0.95	0.86	0.86
7	Op010No3-01	E_1E_2DRC	0.93	0.94	0.94
8	Op013-01	IE_1E_2DRC	0.96	0.95	0.95
9	Op014No1-01	E_1E_2DRC	0.97	0.97	0.97
10	Op014No2-01	E_1E_2DRC	0.94	0.96	0.96
11	Op022-01	E_1E_2DR	1.00	0.97	0.97
12	Op026-01	-	-	-	-
13	Op027No1-01	-	-	-	-
14	Op027No2-01	-	-	-	-
15	Op028-01	E_1E_2DRC	1.00	0.99	0.99
16	Op031No1-01	E_1E_2DRC	0.83	0.74	0.74
17	Op031No2-01	E_1E_2DRC	0.90	0.85	0.85
18	Op031No3-01	E_1E_2DRC	0.99	0.98	0.98
19	Op049No1-01	E_1E_2DRC	0.96	0.91	0.91
20	Op049No2-01	E_1E_2DR	1.00	0.96	0.96
21	Op053-01	E_1E_2DRC	0.99	0.97	0.97
22	Op054-01	-	-	-	-
23	Op057-01	$EDRC$	0.92	0.78	0.78
24	Op078-01	$IE_1E_2D_1R_1D_2R_2$	0.98	0.84	0.84
25	Op079-01	$E_1E_2D_1R_1D_2R_2C$	0.50	0.55	0.55
26	Op081a-01	IE_1E_2DRC	0.86	0.88	0.88
27	Op090-01	$EDRC$	0.76	0.85	0.85
28	Op101-01	$EDRC$	0.97	0.89	0.89
29	Op106-01	E_1E_2DRC	0.99	0.98	0.98
30	Op109-01	$EDRC$	0.92	0.86	0.86
31	Op110-01	$EDRC$	0.91	0.81	0.81
32	Op111-01	IE_1E_2DRC	0.65	0.64	0.64
	Average		0.92	0.86	0.89

Table 6.1: Ground truth annotation and evaluation results (pairwise P/R/F values) for the thumbnailing procedure using Barenboim recordings for the first movements in sonata form of the Beethoven piano sonatas.

induced segments, thus not representing the complete sonata form, see Figure 6.2b. In contrast, using transposition-invariance, also the R-segment is identified by the procedure as a repetition of the E_1-segment, see Figure 6.2e.

At this point, we want to emphasize that only the usage of various smoothing and enhancement strategies in combination with a robust thumbnailing procedure makes it possible to identify the recapitulation. The procedure described in Chapter 3 is suitably adjusted by using smoothed chroma features having a low resolution as well as applying a long smoothing length and transposition-invariance in the SSM computation. Additionally, when deriving the thumbnail, we apply a lower bound constraint for the minimal possible segment length of the thumbnail. This lower bound is set to one sixth of the duration of the music recording, where we make the musically informed assumption that the exposition typically covers at least one sixth of the entire movement.

To evaluate our procedure, we use the complete Barenboim recordings of the 32 piano sonatas by Ludwig van Beethoven. Among the first movements, we only consider the 28 movements that are actually composed in sonata form. For each of these recording, we

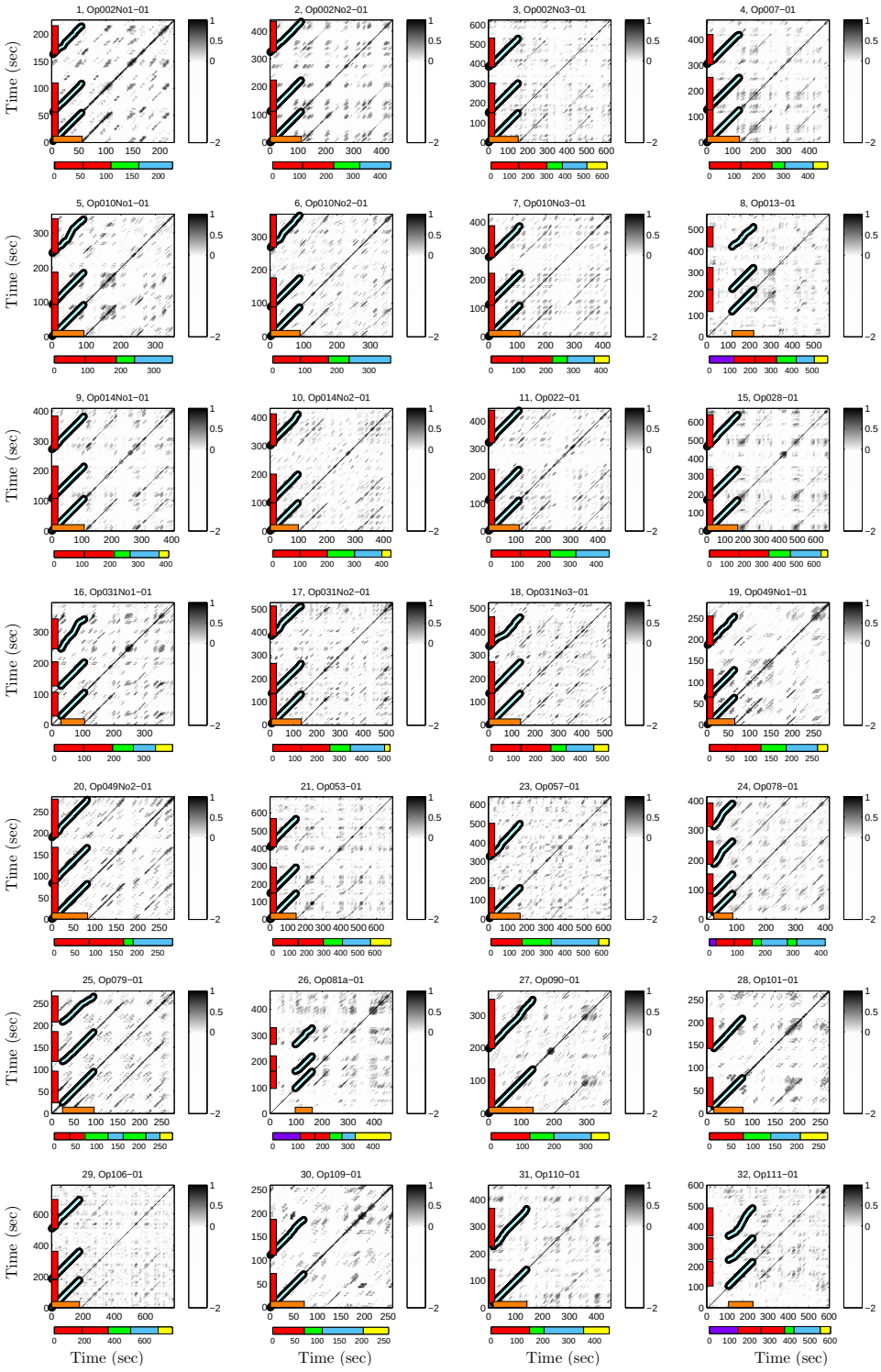

Figure 6.3: Results of the thumbnailing procedure for the 28 first movements in sonata form. The figure shows for each recording the underlying SSM along with the optimizing path family (cyan), the thumbnail segment (indicated on horizontal axis) and the induced segments (indicated on vertical axis). Furthermore, the corresponding GT segmentation is indicated below each SSM.

manually annotated the large-scale musical structure also referred to as ground-truth (GT) segmentation, see Table 6.1 for an overview. Then, using our thumbnailing approach, we computed the thumbnail and the induced segmentation (resulting in two to four segments) for each of the 28 recordings. We use the pairwise frame clustering evaluation measure, which is the standard evaluation measures used in music structure analysis [71, 108], and report the Precision (P), Recall (R), and F-measure (F) values. We compared the computed segments with the E- and R-segments specified by the GT annotation, see Table 6.1. As can be seen, one obtains high P/R/F-values for most recordings, thus indicating a good performance of the procedure. This is also reflected by Figure 6.3, which shows the SSMs along with the path families and ground truth segmentation for all 28 recordings. However, there are also a number of exceptional cases where our procedure seems to fail. For example, for `Op079-01` (No. 25), one obtains an F-measure of only 0.55. Actually, it turns out that for this recording the D-part as well as R-part are also repeated resulting in the form $E_1E_2D_1R_1D_2R_2C$. As a result, our minimum length assumption that the exposition covers at least one sixth of the entire movement is violated. However, by reducing the bound to one eighth, one obtains for this recording the correct thumbnail and an F-measure of 0.85. In particular, for the later Beethoven sonatas, the results tend to become poorer compared to the earlier sonatas. From a musical point of view, this is not surprising since the later sonatas are characterized by the release of common rules for musical structures and the increase of compositional complexity [131]. For example, for some of the sonatas, the exposition is no longer repeated, while the coda takes over the role of a part of equal importance.

6.3 Fine Structure

In the second step, our goal is to find substructures within the exposition and recapitulation by exploiting the relative harmonic relations that typically exist between these two parts. Recall in Section 6.1 that the first subject group of the exposition (E-$G1$) and the first subject group of recapitulation (R-$G1$) contain more or less the same material and both appear in the tonic key. In contrast to this, the second subject group of the exposition (E-$G2$) appears in the dominant or tonic parallel key whereas the second subject group of recapitulation (R-$G2$) remains in the tonic key. Furthermore, compared to the transition of the exposition (E-T), the transition of the recapitulation (R-T) might have some extensions or including new material. In this section, we consider all these musical clues in our automated procedure to help analyzing the fine structure of the sonata form.

To illustrate the harmonic relations between the subject groups, let us assume that the movement is written in C major. Then, in the exposition, E-$G1$ would also be in C major, and E-$G2$ would be in G major. In the recapitulation, however, both R-$G1$ and R-$G2$ would be in C major. Therefore, while E-$G1$ and R-$G1$ are in the same key, R-$G2$ is a modulated version of E-$G2$, shifted five semitones upwards (or seven semitones downwards). In terms of the maximizing shift index as introduced in Section 6.2, one can expect this index to be $i = 5$ in the transposition index matrix when comparing E-$G2$ with R-$G2$.[3] Similarly, for minor sonatas, this index is typically $i = 9$, which corresponds

[3] We assume that the index encodes shifts in upwards direction. Note that the shifts are cyclic, so that

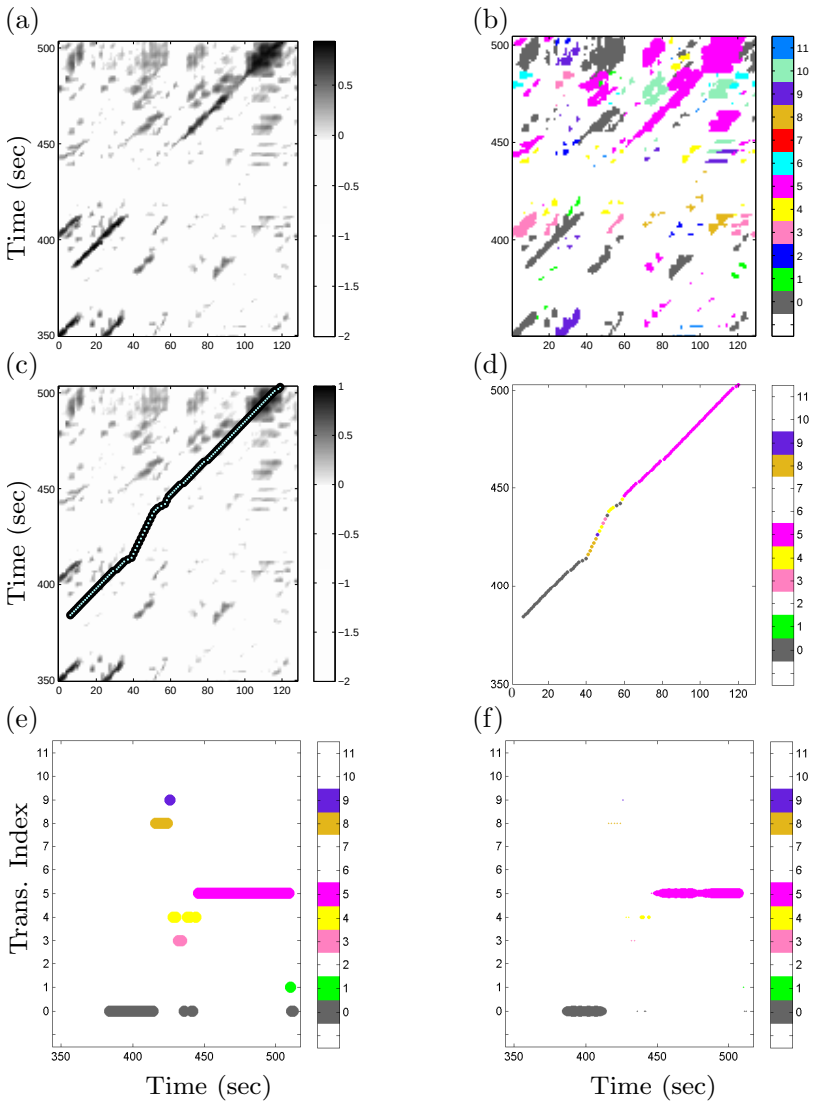

Figure 6.4: Illustration for deriving the WRTI (weighted relative transposition index) representation using `Op031No2-01` as example. **(a)** Enlarged part of the SSM shown in Figure 6.2e, where the horizontal axis corresponds to the E_1-segment and the vertical axis to the R-segment. **(b)** Corresponding part of the transposition index matrix. The white regions in this matrix are the places where we do not consider the transposition index (since their corresponding similarity values in the SSM in figure (a) is less than or equal to zero.) **(c)** Path component of the optimizing path family as shown in Figure 6.2e. **(d)** Transposition index restricted to the path component. **(e)** Transposition index plotted over time axis of R-segment. **(f)** Final WRTI representation.

to shifting three semitones downwards from the tonic parallel to the tonic.

Based on this observation, we now describe a procedure for detecting and measuring the relative differences in harmony between the exposition and the recapitulation. To illustrate this procedure, we continue our example `Op031No2-01` from Section 6.2, where

shifting five semitones upwards is the same as shifting seven semitones downwards.

we have already identified the coarse sonata form segmentation, see Figure 6.2e. Recall that when computing the transposition-invariant SSM, one also obtains the *transposition index matrix*, which indicates the maximizing chroma shift index [89]. Figure 6.4a shows an enlarged part of the enhanced and thresholded SSM as used in the thumbnailing procedure, where the horizontal axis corresponds to the exposition E_1 and the vertical axis to the recapitulation R. Figure 6.4b shows the corresponding part of the transposition index matrix, where the chroma shift indices are displayed in a color-coded form.[4] As revealed by Figure 6.4b, the shift indices corresponding to E-$G1$ and R-$G1$ are zero (gray color), whereas the shift indices corresponding to E-$G2$ and R-$G2$ are five (pink color). To further emphasize these relations, we focus on the path that encodes the similarity between E_1 and R, see Figure 6.4c. This path is a component of the optimizing path family computed in the thumbnailing procedure, see Figure 6.2e. We then consider only the shift indices that lie on this path, see Figure 6.4d. Next, we convert the vertical time axis of Figure 6.4d, which corresponds to the R-segment, into a horizontal time axis. Over this horizontal axis, we plot the corresponding shift index, where the index value determines the position on the vertical index axis, see Figure 6.4e. In this way, one obtains a function that expresses for each position in the recapitulation the harmonic difference (in terms of chroma shifts) relative to musically corresponding positions in the exposition. We refine this representation by weighting the shift indices according to the SSM values underlying the path component. In the visualization of Figure 6.4f, these weights are represented by the thickness of the plotted dots. In the following, for short, we refer to this representation as the WRTI (weighted relative transposition index) representation of the recapitulation.

Figure 6.5 shows the WRTI representations for the 28 recordings discussed in Section 6.2. Closely following [131], we manually annotated the segments corresponding to $G1$, T, $G2$, and C within the expositions and recapitulations of these recordings[5], see Table 6.2. In Figure 6.5, the segment corresponding to R-T is indicated by a blue vertical line (end of R-$G1$) and a red vertical line (beginning of R-$G2$). Note that for some sonatas (e. g., `Op002No3-01` or `Op007-01`) there is no such transition, so that only the red vertical line is visible. For many of the 28 recordings, as the theory suggests, the WRTI representation indeed indicates the location of the transition segment by a switch from the shift index $i = 0$ to the shift index $i = 5$ (for sonatas in major) or to $i = 9$ (for sonatas in minor). For example, for the movement `Op002No1-01` (No. 1) in F minor, the switch from $i = 0$ to $i = 9$ occurs in the transition segment. Or for our running example `Op031No2-01` (No. 17), there is a clearly visible switch from $i = 0$ to $i = 5$ with some further local modulations in between. Actually, this sonata already constitutes an interesting exception, since the shift of the second subject group is from the dominant (exposition) to the tonic (recapitulation) even though the sonata is in minor (D minor). Another more complex example is `Op013-01` (No. 8, "Pathétique") in C minor, where E-$G1$ starts in $E^\flat$ minor, whereas R-$G1$ starts in F minor (shift index $i = 2$) before it reaches the tonic C minor (shift index $i = 9$). Actually, our WRTI representation reveals these harmonic relations.

To obtain a more quantitative evaluation, we located the transition segment R-T by determining the time position (or region) where the shift index $i = 0$ (typically corresponding

[4] For the sake of clarity, only those shift indices are shown that correspond to the relevant entries (having a value above zero) of the SSM shown in Figure 6.4a.

[5] As far as this is possible due to many deviations and variations in the actual musical forms.

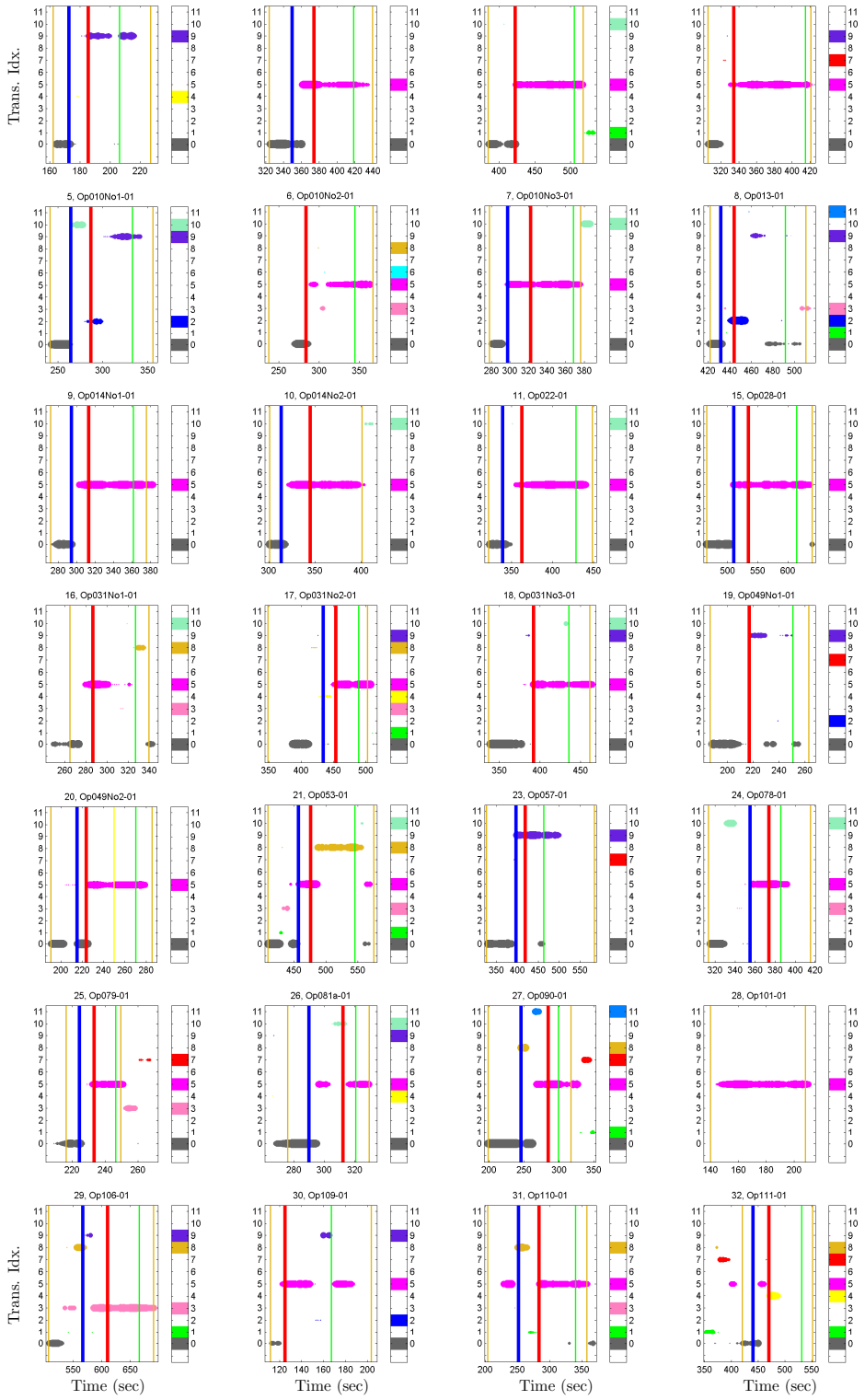

Figure 6.5: WRTI representations for all 28 recordings. The manual annotations of the segment boundaries between *R-G*1, *R-T*, *R-G*2, and *R-C* are indicated by vertical lines. In particular, the blue line indicates the end of *R-G*1 and the red line as the beginning of *R-G*2.

No.	Piece ID	$G1$	T	$G2$	C	$\Delta(G1)$	In(T)	$\Delta(G2)$
1	Op002No1-01	10.6	12.6	20.8	20.4		y	
2	Op002No2-01	26.0	24.4	44.2	21.1		y	
3	Op002No3-01	37.9	-	82.9	12.3	-0.6	n	
4	Op007-01	29.0	-	80.7	5.7	-11.5	n	
5	Op010No1-01	23.2	22.4	45.9	22.4		y	
6	Op010No2-01	46.2	-	60.3	22.2		n	2.0
7	Op010No3-01	20.1	24.7	46.2	7.5	-5.6	n	
8	Op013-01	10.1	12.1	47.2	18.8		y	
9	Op014No1-01	22.8	18.6	48.4	13.9		y	
10	Op014No2-01	13.0	31.4	55.7	-		y	
11	Op022-01	17.5	23.5	65.7	19.8		y	
12	Op026-01	-	-	-	-	-	-	-
13	Op027No1-01	-	-	-	-	-	-	-
14	Op027No2-01	-	-	-	-	-	-	-
15	Op028-01	45.2	24.7	80.3	25.4	-4.0	n	
16	Op031No1-01	21.6	-	40.2	12.6	-12.5	n	
17	Op031No2-01	85.7	19.6	34.9	13.6	-5.4	n	
18	Op031No3-01	55.4	-	42.9	25.7	-10.3	n	
19	Op049No1-01	30.5	-	33.5	12.5	-6.0	n	
20	Op049No2-01	24.6	8.6	26.2	15.2		n	8.9
21	Op053-01	47.6	19.3	69.2	29.1		y	
22	Op054-01	-	-	-	-	-	-	-
23	Op057-01	70.3	22.7	43.7	120.8	-7.3	n	
24	Op078-01	41.7	18.9	11.7	29.5	-15.9	n	
25	Op079-01	8.0	8.9	13.2	2.9		y	
26	Op081a-01	13.9	22.3	8.3	8.8		y	
27	Op090-01	47.1	38.9	14.1	18.2		y	
28	Op101-01	-	-	-	-	-	-	-
29	Op106-01	60.0	43.4	55.5	24.9	-36.7	n	
30	Op109-01	13.7	-	41.9	36.6	-6.1	n	
31	Op110-01	47.8	32.0	56.0	17.3	-26.0	n	
32	Op111-01	20.3	29.9	61.0	20.4		y	

Table 6.2: Ground truth annotation and evaluation results for finer-grained structure. The columns indicate the number of the sonata (No.), the identifier, as well as the duration (in seconds) of the annotated segments corresponding to *R*-$G1$, *R*-T, *R*-$G2$, and *R*-C. The last three columns indicate the position of the computed transition center (CTC), see text for explanations.

to *R*-$G1$) changes to the most prominent non-zero shift index within the *R*-segment (typically corresponding to *R*-$G2$ and usually having a transition index of $i = 5$ or $i = 9$), where we neglect all other shift indices. This position (or region) was computed by a simple sweep algorithm to find the optimal position that separates the weighted zero-indices (which should be on the left side of the optimal sweep line) and the weighted indices of the prominent index (which should be on the right side of the optimal sweep line). In the case that there is an entire region of optimal sweep line positions, we took the center of this region. In the following, we call this time position the *computed transition center* (CTC). In our evaluation, we then investigated whether the CTC lies within the annotated transition *R*-T or not. In the case that the CTC is not in *R*-T, it may be located in *R*-$G1$ or in *R*-$G2$. In the first case, we computed a negative number indicating the directed distance given in seconds between the CTC and the end of *R*-$G1$, and in the second case a positive number indicating the directed distance between the CTC and the beginning of *R*-$G2$. Table 6.2 shows the results of this evaluation, which demonstrates that for most recordings the CTC is a good indicator for *R*-T. The poorer values are in most case due to the deviations in the composition from the music theory. Often, the modulation differences between exposition and recapitulation already start within the final section of the first subject group, which explains many of the negative numbers in Table 6.2. As for the late sonatas such as `Op106-01` (No. 29) or `Op110-01` (No. 31), Beethoven has already radically broken with conventions, so that our automated approach (being naive from a musical point of view) is deemed to fail for locating the transition.

6.4 Further Notes

In this chapter, we have introduced automated methods for analyzing and segmenting music recordings in sonata form. We adapted a thumbnailing approach for detecting the coarse structure and introduced a rule-based approach measuring local harmonic relations for analyzing the finer substructure. As our experiments showed, we achieved meaningful results for sonatas that roughly follow the musical conventions. However, automated methods reach their limits in the case of complex movements, where composition rules have been broken up. We hope that even for such complex cases, automatically computed visualizations such as our introduced WRTI (weighted relative transposition index) representation may still yield some musically interesting and intuitive insights into the data, which may be helpful for musicological studies.

Chapter 7

Repetition-based Structure Analysis

In this chapter, we address the problem of full structure analysis of audio recordings. Here, the goal is to automatically segment the recordings into temporal segments and to group them into music meaningful categories [108]. Typical structural parts of music pieces could be, for example, intro, verse or refrain of a piece of popular music; or first theme, transition, or second theme of a piece of classical work; or sometimes simply annotated parts such as "A", "B", "C" indicating distinct sections for certain pieces of music.

Although the general music structure analysis has been studied and discussed in many research publications in recent years, it is still quite challenging to make this analysis totally automatic. This is mainly due to the four following reasons. Firstly, the identification of repeated sections that correspond to the same musical part is difficult. In music recordings, repeated sections often contain strong variations, which include tempo differences, local key shifts, local changes of melody, and differences in instruments. For example, one verse section of a popular song may contains only singing voice whereas other verse sections consists of singing voice and accompanying instruments. This leads to a strong acoustic deviation when comparing such a section to other verse sections. Human listeners may tolerate this acoustic deviation and consider this section to be one of the repeated verse sections. However it is difficult for automated algorithms to decide whether this section is a repetition of other sections when prominent variations exist. Secondly, the structure parts of some music pieces are hierarchical, where large parts consist of several small parts. For example, in some large classical music work, there are different levels of sections in different lengths. The very long movement sections often consist of several short passages, and short passages further consist of some even shorter themes. It is very difficult for an automated procedure to estimate all such hierarchical sections in one framework. Thirdly, there are many aspects of music to be considered when analyzing the structure of a piece of music. For example, from the functionality aspect, the entire structure could be divided into sections such as intro, verse and refrain. But from the instrumental aspect, it could be divided into sections as solo, chorus or instrumental sections. Finally, the last reason that the automated structure analysis is challenging is that different human listeners may have very different opinions about the structure sections for the same piece

of music. Even for some well-studied classical work, various musicians may not agree on a universal structure decision. In this way, even if a structure analysis result generated by an automated algorithm differs from the one generated by a musician, we might consider it to be musically meaningful after a manual inspection. Such a result may reflect another possible segmentation of the certain piece of music.

In previous chapters, we have already developed some automated methods towards the full structure analysis of music recordings. For example, as a first trial, in Chapter 5 we introduced a compact visualization technique that allows for identifying similar sections by similar color, and for distinguishing different sections by using different colors. In this way, the overall repetitive structures of a music recording can be visualized in a colored scape plot. As a second trial, in Chapter 6 we proposed an automated structure analysis method for recordings in sonata form, where the aim is to derive both the large-scale coarse sections as well as the small-scale fine sections of this special musical form. Although both methods perform well in our experiments, they have their own limitations. The first visualization technique is actually an indirect way of structure analysis, since it avoids to explicitly specify the structure sections by means of segment boundaries. For analysis tasks where the goal is to detect the exact boundaries of a music section, this method is not appropriate to apply. In addition, the second method is designed to analyze only music pieces with a certain kind of structure (sonata form). For music pieces that do not have this kind of structure, the approach is not suitable.

In this chapter, we offer a more general solution towards full structure analysis of music recordings. Since important sections in music pieces are often repeated several times, we use repetitive sections of music recordings as a cue to divide a piece of music into temporal segments and further derive its structure. The main idea is to first extract repetitive sections which correspond to the same musical part, and then assign the same label to these segments. Sections which correspond to different parts will be assigned to different labels. In addition, for the remaining sections which are not repetitive in the piece of music, we treat each of them as a distinct part in the structure.

To realize this idea, we adapt our repetition extraction (thumbnailing) procedure which is introduced in Chapter 3. Based on that, we propose two novel approaches which aim at full structure analysis of music pieces. In the first approach, we iteratively identify one repetitive musical part in each iteration by estimating all repetitive segments corresponding to that part. To this end, in each iteration, we first apply our repetition detection procedure to estimate the current most repetitive segment as well as its repetitions, and then exclude them from the next round of computation. Such iterations are performed until no repetitive segments can be extracted, or until the remaining segments are too short to be of structural importance. We name this method "the iterative approach". The second method is also based on our repetition extraction procedure but works different. We modify our previous repetition extraction procedure such that instead of computing only one most repetitive segment, the most two repetitive segments are estimated within one optimization procedure that tries to jointly maximize the score and coverage of two different segments. By doing this, the lengths and boundaries of the repetitive segments from different groups are computed simultaneously in an optimized way. We name this method "the joint approach". In the following sections, we will introduce both the methods, and analyze the strengths and limits of both approaches in detail.

The remainder of this chapter is organized as follows. We first briefly review the previous proposed thumbnailing procedure in Section 7.1. Then, as main contributions of this chapter, we introduce the iterative approach in Section 7.2. and present the joint approach in Section 7.3. Next, we report on the systematic experiments and evaluations in Section 7.4. Finally, we conclude in Section 7.5 together with an outlook on future research.

7.1 Repetition Detection

To understand how we capture repeated segments of a music recording, we first briefly review the original thumbnailing procedure proposed in Chapter 3. The main idea is to compute a fitness measure that captures repetitiveness as well as coverage for each possible segment of a given audio recording. Following Chapter 3, we assume that the given music recording is represented by a feature sequence with a sampled time axis indexed by $[1:N] = \{1, 2, \ldots N\}$. A segment is then understood to be a subset $\alpha = [s:t] \subseteq [1:N]$ specified by its starting point s and its end point t with $|\alpha| := t - s + 1$ denoting its length. In our experiments we use chroma features with a feature resolution of 2 Hz.

In the computation of the fitness measure, first an enhanced Self-Similarity Matrix (SSM) is computed on the basis of chroma features [90] extracted from the music recording. To deal with local tempo differences and local key changes between the repetitions, we enhanced the SSM to achieve a higher degree of transposition invariance and tempo invariance [93]. Next, for each segment, an optimal path family that simultaneously reveals the relations between the segment and all other similar segments is computed. By projecting such an optimal path family to the vertical axis, one obtains an induced segment family, where each element in this family is similar to the given segment. Note that by our imposed constraints, these induced segments can not overlap with each other. The fitness measure of a segment is associated with some kind of score and coverage of the optimal path family. After that, we compute fitness values for all possible segments of an audio recording and select the segment with the maximum fitness as the thumbnail.

The fitness values can be visualized by means of a triangular scape plot [92, 116, 117]. Each point of the scape plot corresponds to a segment $\alpha = [s:t]$, where the horizontal coordinate encodes the center $c(\alpha) := (s+t)/2$ of the segment and the vertical coordinate encodes its length $|\alpha|$. The fitness value $\varphi(\alpha)$ is then visualized in some color-coded form. The fitness scape plot represents the repetitive structure of the music recording in some hierarchical way.

As an illustration, Figure 7.1a shows the scape plot computed for all possible segments of the Beatles recording "Birthday". The green circle in the scape plot indicates the point of maximal fitness, which corresponds to the thumbnail segment $\alpha_0 = [264:316]$. Comparing this segment to the ground truth annotation (indicated by colored rectangles), we see that the computed thumbnail exactly corresponds to the third V (verse) part, which is the most repetitive part according to the annotation. Figure 7.1b shows the computed optimal path family (cyan paths) for the thumbnail segment α_0 (horizontal axis). The induced segment family is shown on the vertical axis. It consists of the induced segments which are derived from the optimal path family. Each of such induced segments represents a segment that similar to α_0.

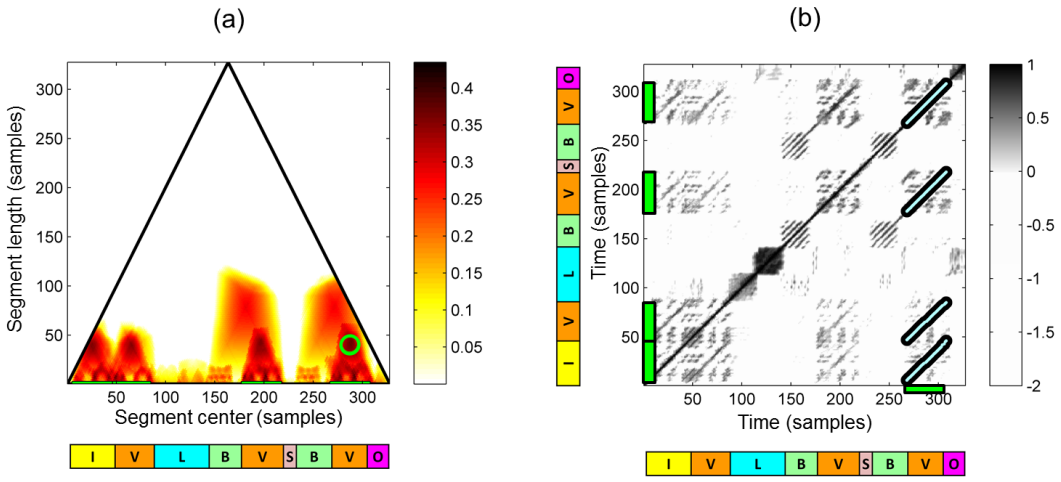

Figure 7.1: Illustration of the thumbnailing procedure applied to the Beatles song "Birthday". **(a)** The fitness scape plot, with the fitness maximizing point indicated by the green circle. **(b)** The enhanced SSM, the computed thumbnail segment (vertical axis), the optimal path family (cyan paths), and the derived induced segments (vertical axis). Note that one of the induced segment correspond to the thumbnail segment itself.

By a comparison to the ground truth annotation, we see that three of the four induced segments exactly reveal the three V parts. Although the first induced segment (the bottom one on the vertical axis) corresponds to the I (Intro) part, by listening inspection we found this intro is actually an instrumental version of the verse parts and can be considered as a special "verse". In this way, the induced segments, which were derived from our thumbnail segment, successfully reveal all "verse" parts of the audio recording.

7.2 The Iterative Approach

In the example shown in Figure 7.1, we see that an induced segment family can reveal all sections of a repetitive music part. According to the ground truth annotation, the most repetitive part is the V part with three sections, and the second repetitive part is the B (bridge) part with two sections. All other parts are single parts, with each of them having only one section. One crucial observation for this example is that all these single parts (I, L, S and O parts), are separated by sections of repetitive parts (V and B parts). As a result, we assume that if we correctly estimate the repetitive parts for this recording, other non-repetitive parts of the recording can also be derived by considering each of the remaining separated regions as a distinct part. In this way, by estimating the repetitive parts and those distinct single parts, the overall structure of the recording can be derived.

7.2.1 Main Idea

Having the above mentioned observation as the basis, we come up with an intuitive idea which is to iteratively estimate repetitive segments that correspond to one music part.

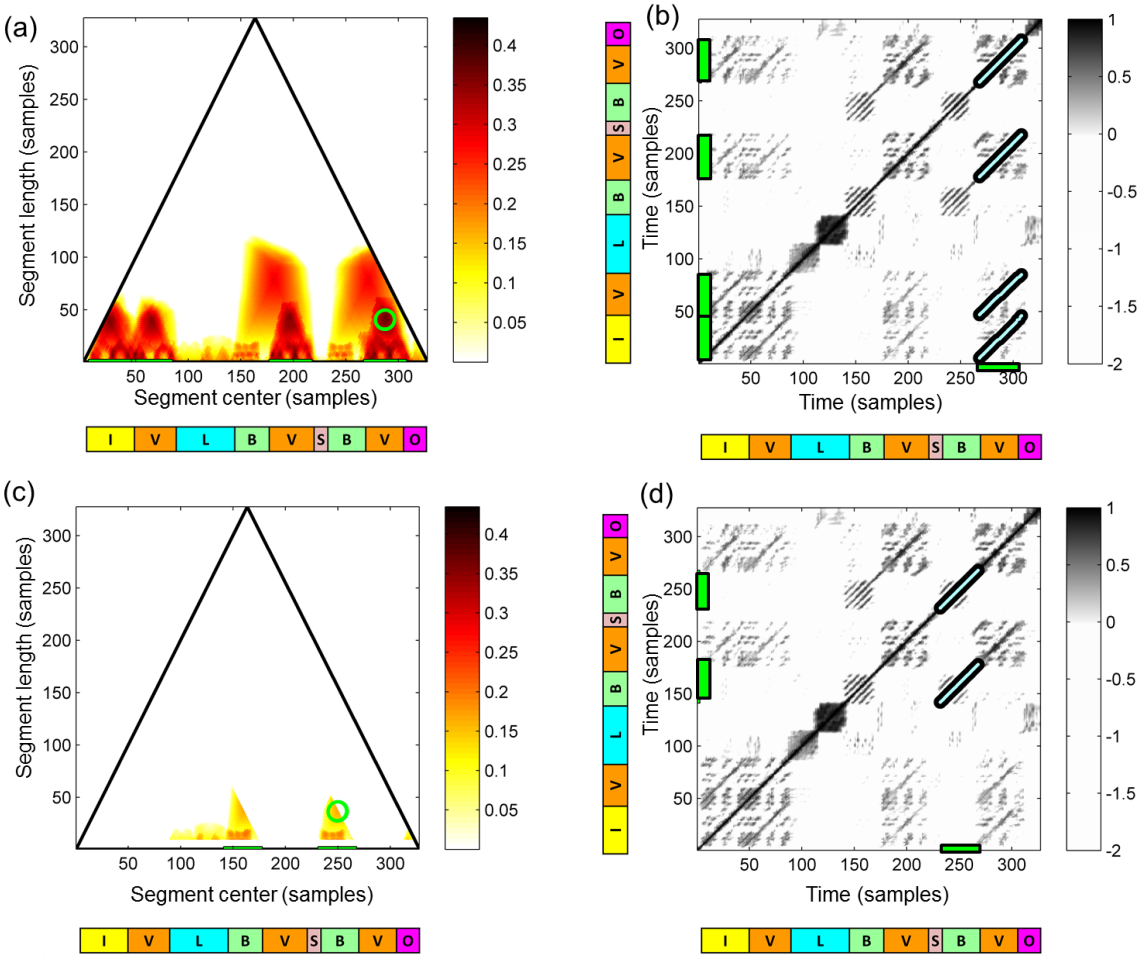

Figure 7.2: Illustration of the computation steps of the iterative approach applied on the Beatles song "Birthday". We present the scape plot with the fitness maximizing point in the left figures, and present the enhanced self-similarity matrix (SSM), the thumbnail segment (horizontal axis), its optimal path family (cyan colored), and the induced segments (vertical axis) in the right figures. The colored rectangles indicate the ground truth structure annotation. In order to help for understanding the segments removal step in the scape plot figure, the induced segments are also plotted on the horizontal axis. **(a)/(b)** The first computation round (the original thumbnailing procedure). **(c)/(d)** The second computation round.

To this aim, in each iteration, we derive a group of repetitive segments and then exclude them from the audio recording. We now illustrate our idea using the same Beatles song "Birthday" as an example. In the first computation round, we apply the thumbnailing procedure and estimate all repetitive segments that correspond to the V sections, see Figure 7.2(a) and (b). Then we remove these segments such that the procedure does not consider them in the next computation round. We introduce the detail about the segment removal in the next section. In the second computation round, we apply the thumbnailing procedure again and estimate the second most repetitive segments. In this way, the segments that correspond to B sections are also derived, see Figure 7.2(c) and (d). Also, the newly detected segments are removed. The approach continues to proceed such iterations until either no repetitive segments can be detected, or until the remaining

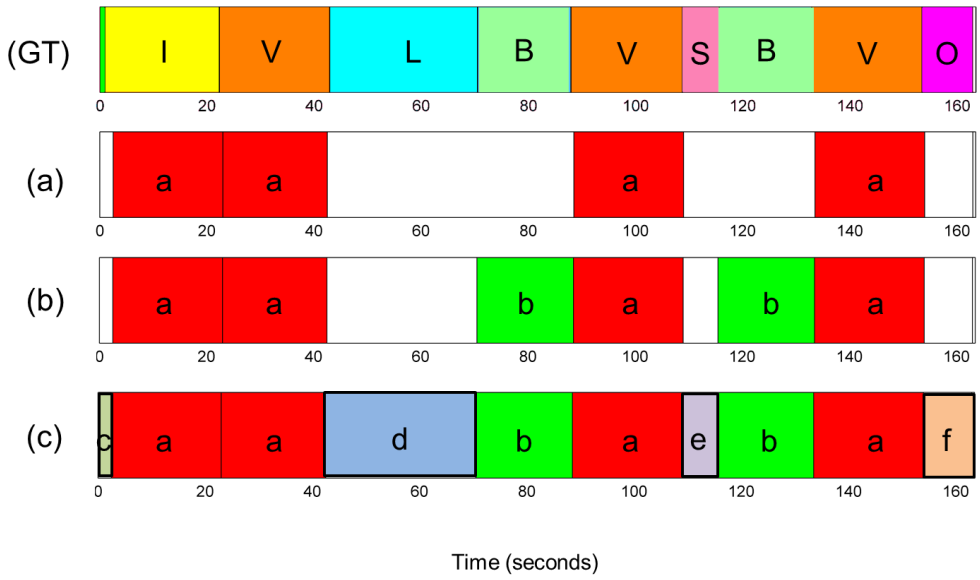

Figure 7.3: Structure estimation using the iterative approach for the Beatles song "Birthday". **(GT)** The ground truth segmentation annotation **(a)** Repetitive segments estimated after the first computation round. **(b)** Repetitive segments estimated after the second computation round. **(c)** Final structure estimation result.

segments are too short to be a structural element. In this example, after two rounds of computation, the procedure cannot detect any repetitive segments and therefore stopped here.

We gather the estimated segments of each computation round and present them in the structure estimation visualization as shown in Figure 7.3. After the first computation round, we label the estimated segments as "a" segments, see Figure 7.3a. These segments reveal the sections of the V part in the ground truth. After the second round computation, we label the segments detected in this round as "b" segments, see Figure 7.3b. Again, these segments reveal the two sections of the B part in the annotations. After the procedure stops, we identify all remainder segments in the audio recording as distinct parts and assign each of them a different label such as "c", "d","e" and "f". This results in the final structure estimation as shown in Figure 7.3c. Comparing to the ground truth annotation in Figure 7.3(GT), we see that the computed results successfully reveal the overall structure of this song.

7.2.2 Segment Removal

One important step in the iterative approach is the segment removal. After we estimated a group of repetitive segments (the induced segments of the thumbnail) in one computation round, we need to exclude them such that the next computation round do not consider the repetitiveness of these segments but focus on repetitiveness of other segments. Furthermore, since the final structure estimation result cannot contain any overlapped segments, those segments that overlap with the already estimated segments are not allowed to be

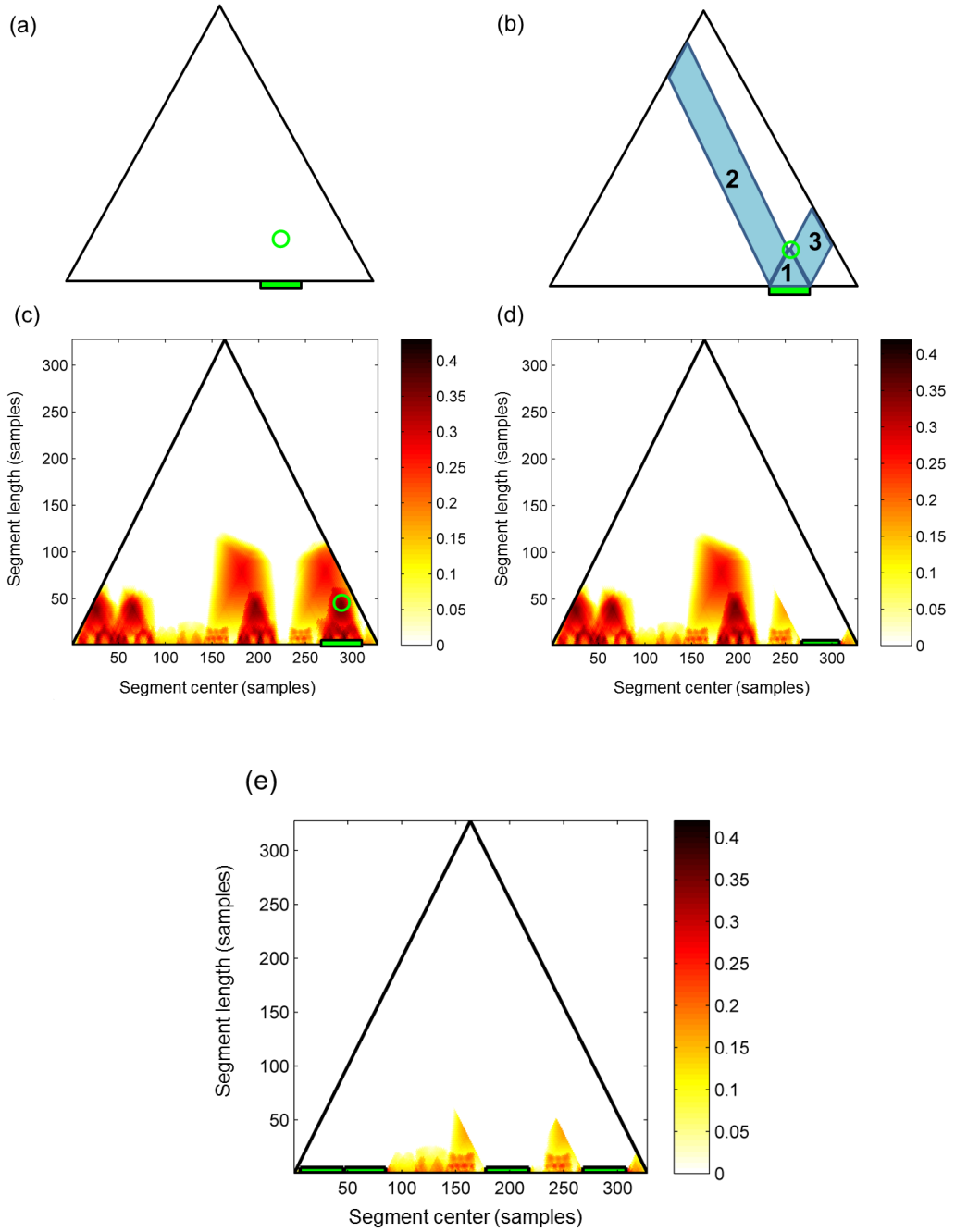

Figure 7.4: Illustration of the segment removal step of the iterative approach. **(a)** The green circle indicates a point in the fitness scape plot that corresponds to one of the estimated repetitive segments to be removed (segment α_r). **(b)** The segment removal step for α_r. The points in the three regions represent those segments that overlap with α_r, which should also be removed. **(c)** The fitness scape plot of Beatles song "Birthday". The green circle indicates one of the four induced segments. **(d)** The resulting fitness scape plot after performing segment removal for the induced segment in (c). **(e)** The resulting fitness scape plot after performing segment removal for all the four induced segments. The four induced segments are plotted on the horizontal axis.

structural segment candidates. Therefore, such segments should be removed as well.

Since the fitness scape plot represents all segments by means of points in a compact view, we perform the segment removal in the fitness scape plot so that one can clearly visualize which segments should be removed. Our idea is to first select the points that represent the estimated segments, and then select those points whose corresponding segments overlapped with the estimated segments. We set fitness values of all the selected points to zero, so that the corresponding segments of these points cannot be picked by the iterative approach as structural segment candidate any more.

Now we use Figure 7.4 as an illustrative example for the segment removal step. In Figure 7.4a, the green circle indicates one of the estimated repetitive segments. We denote this segment as α_r, meaning a segment to be removed. In Figure 7.4b, we indicate all points whose corresponding segments overlapped with α_r. The points in region 1 represent the segments which locate inside α_r. The points in region 2 represent those segments which overlapped with α_r from their right side, and region 3 represent those segments which overlapped with α_r from their left side. We remove all points in these three regions from the scape plot. After that, all the remaining points in the scape plot do not overlap with α_r. Now we use the same Beatles "Birthday" as a real example to show the result of segment removal. In the first round computation, we estimated four induced segments as shown in Figure 7.2b. Suppose we perform the segment removal step for the last induced segment. We first locate the fitness point of this induced segment by the green circle in the scape plot as shown in Figure 7.4c. Then we remove all segments that overlapped with this segment by setting the fitness values of the corresponding points to zero in the fitness scape plot (These points are selected as shown in Figure 7.4b). The resulting fitness scape plot can be seen in Figure 7.4d. After that, we perform the segment removal for the other three induced segments as well. The final fitness scape plot can be seen in Figure 7.4e.

There still remains one problem for the above described segment removal method. After performing the segment removal, the thumbnail segment computed in the next round cannot overlap with the removed segments. However, it is still possible that one of its induced segments, which derived from the thumbnail, overlap with the removed segments. Figure 7.5 illustrates this phenomenon using the Beatles song "Devil In Her Heart". Figure 7.5b shows the thumbnail segment and the four induced segments estimated in the first round computation. After performing the segment removal step for the induced segments, the fitness scape plot is shown in Figure 7.5c. However, in the second round computation, except of the induced segment that corresponds to the thumbnail segment, all other induced segments overlap with the segments estimated in the first round (compare green segments on vertical axes of Figure 7.5(d) and (b)). In this situation, we use the following strategy: for any segment that overlap with the previous estimated segments in certain portions, we throw away the overlapped portion of that segment and treat the non-overlapped portion a single segment. If a segment totally overlaps with the previous estimated segments, it will be totally discarded from the final estimation result. If it is only partially overlaps with the previous estimated segments, the non-overlapped portion of that segment will be kept into the final structure estimation result.

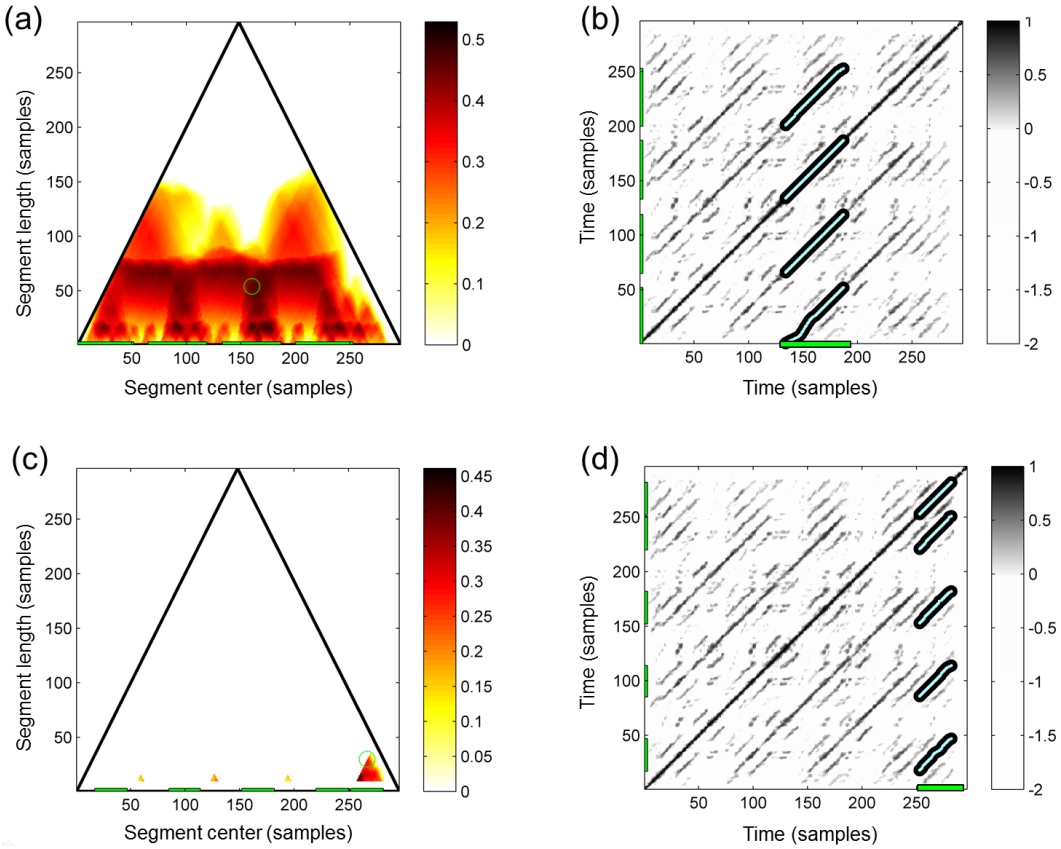

Figure 7.5: Illustration for the problem of the segment removal step in the iterative approach. We use Beatles song "Devil In Her Heart" as the example. We present the scape plot with the fitness maximizing point in the left figures, and present the enhanced self-similarity matrix (SSM), the thumbnail segment (horizontal axis), and the induced segments (vertical axis) in the right figures. **(a)/(b)** The first computation round. Note that the four induced segments and all segments overlap with them are removed after this computation round. **(c)/(d)** The second computation round. In (d), we find the problem that although the thumbnail segment does not overlap with the previous removed segments, but some of its induced segments overlap with the removed segments.

7.2.3 Problem Analysis

The above illustration of the iterations shown in Figure 7.5 also reveals the main problem of the iterative approach, that is, the segments estimated in the later round are dependent on the segments estimated in the previous round. If the segments in the first round were estimated too long such that they "eaten up" regions that belongs to other musical part, this will lead to the incorrect estimation of segments correspond to other musical part in later iterations. The reason of this problem is that the original thumbnailing procedure, which is used in each iteration, is designed to capture *one* thumbnail only. It selects the thumbnail in a greedy way without considering the influence for other segments.

To illustrate this problem, we convert the induced segments of each iteration of Figure 7.5 into the structure estimation result as shown in Figure 7.6. The top sub-figure shows the ground truth annotation. The middle sub figure and the bottom one show the result

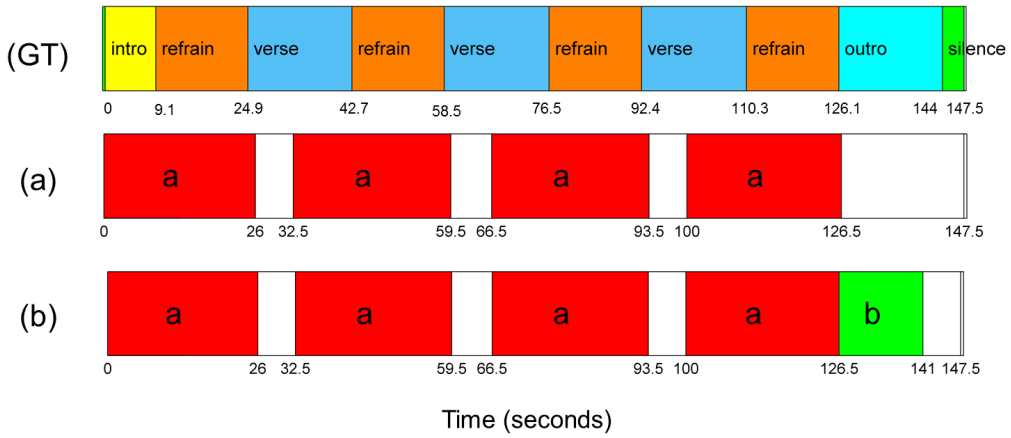

Figure 7.6: Illustration of the structure estimation result for Beatles song "Devil In Her Heart" (related to Figure 7.5). **(GT)** The ground truth segmentation annotation. **(a)** Structure estimation after the first round computation. **(b)** Structure estimation after the second round computation.

segmentation after the first round computation and the second round computation, respectively. We see from Figure 7.6a that after the first round computation, all estimated repetitive segments (red colored segments with label "a") are passed into the final structure estimation result. However, these segments are estimated much longer compared to the most repetitive refrain part, as annotated in the ground truth. Each of these segments corresponds to not only the refrain part but also a fraction of the verse part. As a consequence, after the second round computation shown in Figure 7.6b, the approach does not derive the second thumbnail segment that corresponds to the verse part. Instead, it estimates the thumbnail segment (the green colored segment) which corresponds to the outro part in the ground truth. However, for this thumbnail segment, all its induced segments (except of itself) are totally overlapped with the previous identified segments in the first computation round. Therefore they are discarded from the final structure segmentation. After a listening inspection, we find that the outro part is a slight altered version of the refrain part can also be considered as a special refrain. Actually, according to Figure 7.5d, the iterative approach correctly estimated the thumbnail and induced segments that correspond to all "refrain" sections. However, even if the segments estimated in the second computation round are correct, they cannot contribute to the final structure estimation. Because their places are "eaten up" by the problematic segments estimated in the first round computation. We call this problem as "A eaten up B" problem.

Another illustration of this "A eaten up B" problem can be seen in Figure 7.7, where we perform the structure estimation for the Beatles recording "Act Naturally". The repetitive segments estimated in the first round (see Figure 7.7a) correspond to the verse sections in the annotation. But each of these computed segments is estimated a bit longer than the actual length of the verse part in the ground truth, and they occupy some portions which should belongs to the bridge sections. As a consequence, the repetitive segments estimated in the second computation round cannot be accurate any more, since the estimated segments in the first round computation have "eaten up" some of regions

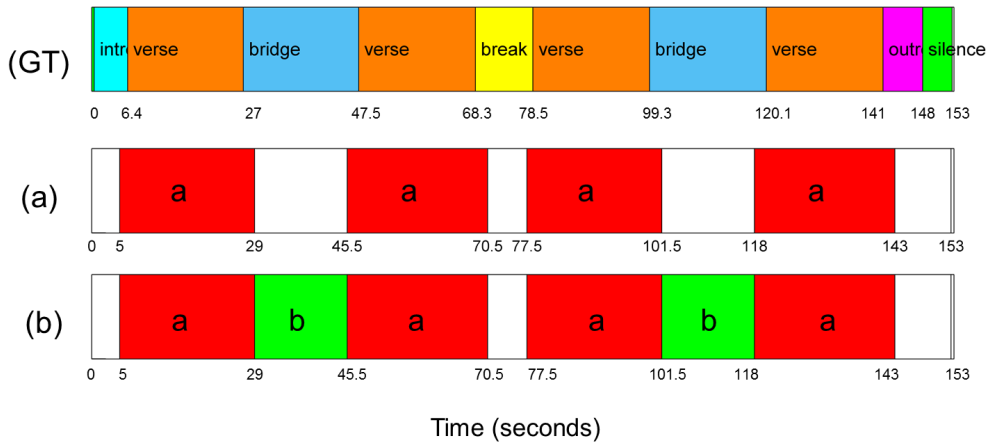

Figure 7.7: Illustration of the structure estimation result for the Beatles song "Act Naturally". **(GT)** The ground truth segmentation annotation **(a)** Structure estimation after the first round iteration. **(b)** Structure estimation after the second round iteration.

which should belong to the segments in the second round. This is illustrated in Figure 7.7b, that each of the 'b' segments is estimated a bit shorter compared to their corresponding bridge sections in the ground truth.

7.3 The Joint Approach

Although the iterative approach aimed at identifying repetitive segments that correspond to one musical part in each iteration, it may be problematic for the estimation of segment lengths and boundaries. In particular, if the segments estimated in the first round are longer than they actually should be, the segments in the remaining iterations are also influenced due to the error prolongation from the the first round. As we already discussed in Section 7.2.3, this is because in each iteration, the approach focus on picking the current most repetitive segment in a greedy fashion, without considering the balance between itself and other segments. In order to avoid this problem, we introduce a second repetition-based method in this section, which works in a different way and serves as a compensate of the the first approach.

In the second approach, we modify our thumbnailing procedure so that instead of picking one most repetitive segment, the procedure will jointly estimate two most repetitive segments in one optimization scheme. In this way, the lengths and positions of the two segments are jointly optimized and thus the errors at the segment boundaries are minimized. Idealy, the structure information derived by the joint approach should be more exact compared to the iterative approach. We present the technical details of this approach in Section 7.3.1. Then, considering the actual implementation, we introduce an acceleration procedure for this approach in Section 7.3.3. This joint approach also has its own limitations as it is only suitable for audio recordings with two main repetitive parts in the structure. In order to evaluate its performance and compare it to the iterative approach, we analyze both the strengths and drawbacks of the two approaches by qualitative

examples and quantitative evaluations in Section 7.4.

7.3.1 The Joint Fitness Measure for Two Segments

In Section 3.3, we have introduced our definition of the fitness measure that captures the repetitiveness of one segment. We now extend it and propose a joint fitness measure to capture the repetitiveness of a pair of segments. We closely follow the notations which were originally introduced in Section 3.3 and we also extend some definitions [1].

7.3.1.1 Joint Path Family

Following the notations in Section 3.3, let $X = (x_1, x_2, \ldots, x_N)$ be a feature sequence and $\mathcal{S} \in \mathbb{R}^{N \times N}$ an enhanced self-similarity matrix. Following the definition of a segment in equation (3.1), we denote two disjoint segment as:

$$\begin{aligned} \alpha &= [s_1 : t_1] \subseteq [1 : N] \\ \beta &= [s_2 : t_2] \subseteq [1 : N] \end{aligned} \tag{7.1}$$

where $s_1 \leq t_1 < s_2 \leq t_2$. Let $|\alpha| := t_1 - s_1 + 1$ and $|\beta| := t_2 - s_2 + 1$ denote their lengths, respectively.

Next, we keep the definition of a path as defined by Equation (3.3), and denote a path over α as p^α and a path over β as p^β. We keep the step size condition Ω which constrains the slope of a path as defined in Equation 3.4. In this way, the slope of a path is still within the bounds of 1/2 and 2.

We also keep the definition for the two projections of a path as defined by Equation (3.6) and Equation (3.5). In this way, $\pi_1(p^\alpha)$ denotes an induced segment of a path p^α, and $\pi_1(p^\beta)$ an induced segment of a path p^β.

Joint path family:

Now, based on the path family definition in Equation (3.8), we newly define: a *joint path family* over α and β, which is a set

$$\mathcal{P}^{\alpha\beta} := \{p_1^\alpha, p_2^\alpha, \ldots, p_U^\alpha, p_1^\beta, p_2^\beta, \ldots, p_V^\beta\} \tag{7.2}$$

of size $U + V$, consisting of paths p_u^α over α and paths p_v^β over β, where $u \in [1 : U]$ and $v \in [1 : V]$. In addition, similar as in Section 3.3, we also impose that the induced segments derived from this joint path family are pairwise disjoint. In other words, the set $\{\pi_1(p_1^\alpha), \ldots, \pi_1(p_U^\alpha), \pi_1(p_1^\beta), \ldots, \pi_1(p_V^\beta)\}$ consists of pairwise disjoint segments.

Score of a joint path family:

We keep the same definition $\sigma(p)$ for the score of a path as defined in Equation (3.7), and based on the definition $\sigma(\mathcal{P})$ for the score of a path family as defined in Equation (3.9).

[1] To better understand this approach, one can first refer to Figure 7.11 and Figure 7.9, which illustrate some of the definitions.

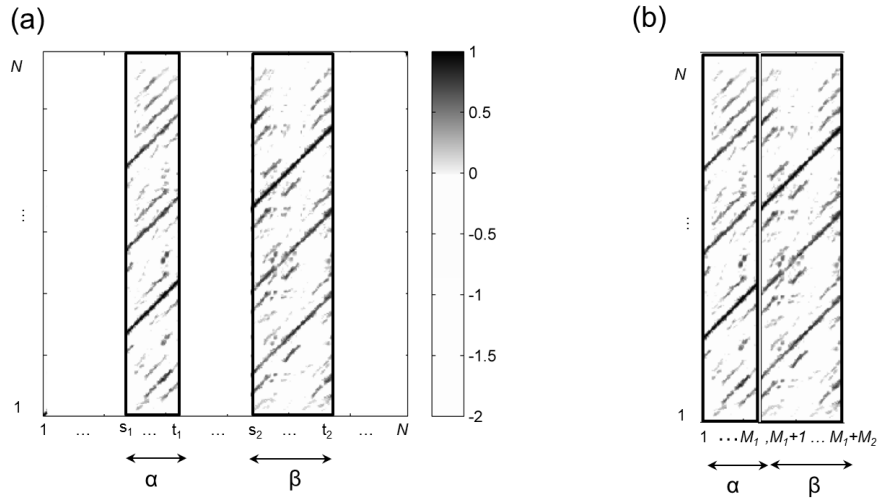

Figure 7.8: Illustration of the generation of $S^{\alpha\beta}$. **(a)** The selected columns that correspond to α and β in an similarity matrix $\mathcal{S}$. **(b)** Those columns from (a) are combined to form the sub-matrix $S^{\alpha\beta}$. Note that the indices on the horizontal axis of $S^{\alpha\beta}$ have changed.

we define the *score of a joint path family* as:

$$\sigma(\mathcal{P}^{\alpha\beta}) := \sum_{u=1}^{U} \sigma(p_u^{\beta}) + \sum_{v=1}^{V} \sigma(p_v^{\beta}). \tag{7.3}$$

Similar as before, there are in general many possible joint path families over α and β. Among these path families, there exists an optimal path family of maximum score, defined as

$$\mathcal{P}^*_{\alpha\beta} := \underset{\mathcal{P}^{\alpha\beta}}{\operatorname{argmax}}\ \sigma(\mathcal{P}^{\alpha\beta}). \tag{7.4}$$

7.3.1.2 Optimization Scheme

In Section 3.3.2, we have introduced an optimization scheme for computing an optimal path family over a segment. Now in this section, we aim to compute an optimal joint path family over a pair of segments. This family should contain some paths over the first segment as well as paths over the second segment. The requirement is that, the induced segments of these paths do not overlap. Based on the optimization scheme introduced in Section 3.3.2, we now describe a modified algorithm that can efficiently compute an optimal joint path family.

To account for a second segment, we adapt the dynamic time warping (DTW) procedure described in Section 3.3.2 and extend it for the second segment. Following Section 3.3.2, Let $X = (x_1, x_2, \ldots, x_N)$ be the feature sequence of the entire audio recording, $Y := (x_{s_1}, \ldots, x_{t_1})$ and $Z := (x_{s_2}, \ldots, x_{t_2})$ the feature sequences corresponding to α and β, respectively. The goal of the new optimization scheme is to simultaneously align paths between Y (or Z) and some sub-sequences of X, with the constraint that no overlaps

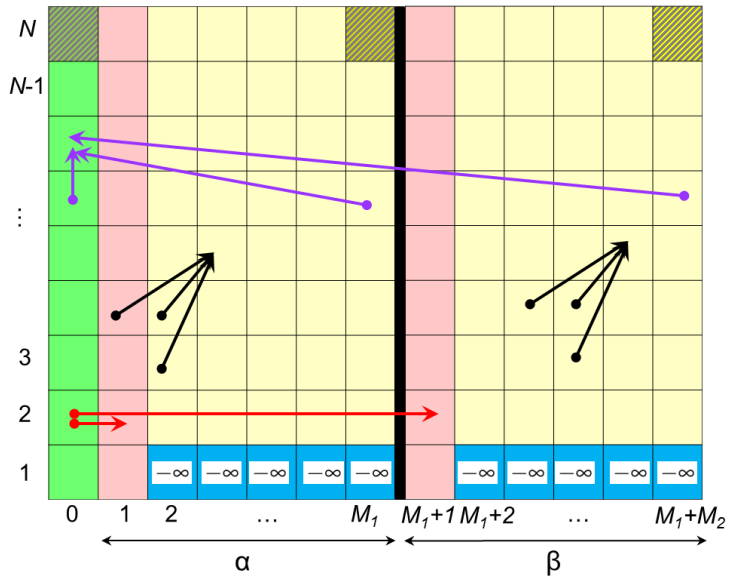

Figure 7.9: Illustration of the optimization scheme in computing the accumulated score matrix D. The colored regions and the arrows indicate some step conditions which are imposed to control path starting (Equation (7.8) and (7.9)), path alignment (Equation (3.4), (7.5) and (7.6)), and path closing together with possible section skipping (Equation (7.7)). The final optimal score can be derived from the score of the three shadowed cells in the top row using Equation (7.10).

between these sub-sequences of X are allowed. The goal is to determine the optimal alignment that defines the optimal joint path family of maximal score. Note that we impose the entire segments of α and β to be aligned with sub-sequences of X. Furthermore, in order to skip some sub-sequences of X which are neither similar to α nor to β, certain sections of X can be left completely unconsidered in the alignment.

To account for these constraints, similarly as before, we introduce some steps that allow us to skip certain sections of X and to jump from the end to the beginning of the given segment α (or β). The step conditions mainly changed for the involvement of a second segment β.

Score accumulation matrix and path alignment:

First, we define an $N \times (M_1 + M_2)$ submatrix $\mathcal{S}^{\alpha\beta}$ by taking the columns s_1 to t_1 and s_2 to t_2 of $\mathcal{S}$, see Figure 7.8. Next, by extending the score matrix definition in Equation (3.12), we introduce a new accumulated score matrix D'. By setting different step conditions for different regions in D', we realize the above mentioned constraints. To this end, we define $D' \in \mathbb{R}^{N,(1+M_1+M_2)}$ (with rows indexed by $[1:N]$ and columns indexed by $[0:(M_1+M_2)]$), by the following recursion:

$$D'(n,m) = \mathcal{S}^{\alpha\beta}(n,m) + \max\{D'(i,j) \mid (i,j) \in \Phi'(n,m)\} \tag{7.5}$$

for $n \in [2:N]$ and $m \in [2:M_1] \cup [(M_1+2):(M_1+M_2)]$ (the yellow regions in Figure 7.9), where

$$\begin{aligned}\Phi'(n,m) =& \{(n-i, m-j) \mid (i,j) \in \Omega\} \quad \cap \\ & \{[1:N] \times ([1:(M_1-1)] \cup [(M_1+1):(M_1+M_2-1)])\}\end{aligned} \tag{7.6}$$

denotes the set of possible predecessors (see the black arrows in Figure 7.9). Note that we keep the step size condition Ω as defined in Equation 3.4. So far, these definitions are used for computing the accumulated score during path alignments. Comparing to Figure 3.4 in Section 3.3.2 which illustrates the score accumulation and path alignment for only one segment α, here in this section we include an extra accumulation region that corresponds to β, thus enables the path alignment for both segment α and β with sequence X.

Skipping sections of X and path closure:

Then, we need to modify the constraint conditions that allow for skipping some sections in X. This is realized by extending the step conditions in Equation (3.13). The first column of D' indexed by $m = 0$ plays a special role, and it is recursively defined as:

$$D'(n,0) = \max\{D'(n-1,0), D'(n-1,M_1), D'(n-1,M_1+M_2)\} \tag{7.7}$$

for $n \in [2\!:\!N]$ and initialized by $D'(1,0) = 0$ (see the green region and the purple arrows in Figure 7.9). The term $D'(n-1,0)$ enables the algorithm to move upwards without accumulating any (possibly negative) score, thus allows for skipping some sections of X without penalty (negative score). Note that the term $D'(n-1,M_1)$ closes up a path over α, and the term $D'(n-1,M_1+M_2)$ closes up a path over β. The later two terms ensure that the entire segment α or β is aligned to the sub-sequence of X, and the next possible sub-sequence of X to be aligned does not overlap with the previous aligned sub-sequence.

Starting a path:

After introducing path alignment and closure, we now present how to start a new path. This is realized by controlling the column for $m = 1$ and $m = M_1 + 1$ in D' which correspond to the beginning of α and β, respectively. Extending Equation (3.14), we define the new constraints as:

$$D'(n,1) = D'(n,0) + \mathcal{S}^{\alpha\beta}(n,1) \tag{7.8}$$

$$D'(n,M_1+1) = D'(n,0) + \mathcal{S}^{\alpha\beta}(n,M_1+1) \tag{7.9}$$

for $n \in [1:N]$ (see the pink regions and the red arrows in Figure 7.9).

Initialization:

Finally, in order to initialize the D' matrix, we set $D'(1,m) = -\infty$ for $m \in [2:M_1] \cup [(M_1+2):(M_1+M_2)]$ (see the blue region in Figure 7.9), which forces the first path to start either with the first element of α or the first element of β. Based on these definitions, similar as in Equation (3.15), the score of an optimal joint path family is then given by

$$\sigma(\mathcal{P}^*_{\alpha\beta}) = \max\{D'(N,0), D'(N,M_1), D'(N,(M_1+M_2))\} \tag{7.10}$$

(see the shaded cells in the top row in Figure 7.9). The first term $D'(N,0)$ reflects the situation that the optimal path family may skip the alignment with the final section of X, and the later two terms $D'(N,M_1)$ and $D'(N,(M_1+M_2))$ ensure that for the other cases, the last path is either aligned with the entire segment α or with the entire segment of β. The associated optimal joint path family $\mathcal{P}^*_{\alpha\beta}$ can be derived from D by using a back-tracking algorithm as in classical DTW (see [88, Chapter 2]).

7.3.1.3 Joint Fitness Measure

We now define the new joint fitness measure. Similar as in Section 3.3.3, we associate the joint fitness measure for a pair of segments with their optimal joint path family. We consider two properties of the joint path family, which are the score and the coverage. Compared to the previous fitness measure, we need to add the contributions of the paths over the second segment into the joint fitness measure. In addition, the contribution of a segment itself to the score and the coverage need to be excluded, otherwise the segment representing the entire audio file will get the maximum score and coverage, which is undesirable.

First, we consider the score measurement. Let $\mathcal{P}^*_{\alpha\beta} = \{p^\alpha_1, \ldots, p^\alpha_U, p^\beta_1, \ldots, p^\beta_V\}$ be an optimal path family for a pair of segments α and β. By extending Equation (3.16), the *normalized score* $\bar{\sigma}'(\alpha, \beta)$ is defined as:

$$\bar{\sigma}'(\alpha, \beta) := \frac{\sigma(\mathcal{P}^*_{\alpha\beta}) - |\alpha| - |\beta|}{\sum_{u=1}^{U} L^\alpha_u + \sum_{v=1}^{V} L^\beta_v} \tag{7.11}$$

where L^α_u and L^β_v are the lengths of the respective paths p^α_u and p^β_v from the optimal joint path family. Second, we consider some kind of coverage measure for α and β . Let $\mathcal{A}^*_{\alpha\beta} := \{\pi_1(p^\alpha_1), \ldots, \pi_1(p^\alpha_U), \pi_1(p^\beta_1), \ldots, \pi_1(p^\beta_V)\}$ be the segment family induced by $\mathcal{P}^*_{\alpha\beta}$, and let $\gamma(\mathcal{A}^*_{\alpha\beta})$ be the coverage of this induced segment family, which is defined as

$$\gamma'(\mathcal{A}^*_{\alpha\beta}) = \sum_{u=1}^{U} |\pi_1(p^\alpha_u)| + \sum_{v=1}^{V} |\pi_1(p^\beta_v)|. \tag{7.12}$$

Then, extending Equation (3.17), we define the *normalized coverage* $\bar{\gamma}'(\alpha, \beta)$ as:

$$\bar{\gamma}'(\alpha, \beta) := \frac{\gamma'(\mathcal{A}^*_{\alpha\beta}) - |\alpha| - |\beta|}{N}. \tag{7.13}$$

Finally, combining the normalized score and the normalized coverage, we define the *joint fitness measure* for α and β to be their harmonic mean:

$$\varphi'(\alpha, \beta) := 2 \cdot \frac{\bar{\sigma}'(\alpha, \beta) \cdot \bar{\gamma}'(\alpha, \beta)}{\bar{\sigma}'(\alpha, \beta) + \bar{\gamma}'(\alpha, \beta)}. \tag{7.14}$$

7.3.2 Joint Thumbnail

Similarly as in Section 3.4, among all possible pairs of segments of an audio recording, we can define the joint thumbnails as the pair of segments having maximal joint fitness:

$$(\alpha, \beta)^* := \underset{\alpha,\beta}{\operatorname{argmax}}\ \varphi'(\alpha, \beta). \tag{7.15}$$

This is the extension of Equation (3.19).

In addition, to account for prior knowledge such as length information of a thumbnail, we also impose two lower bounds of segment lengths θ_1 and θ_2 for the joint thumbnail.

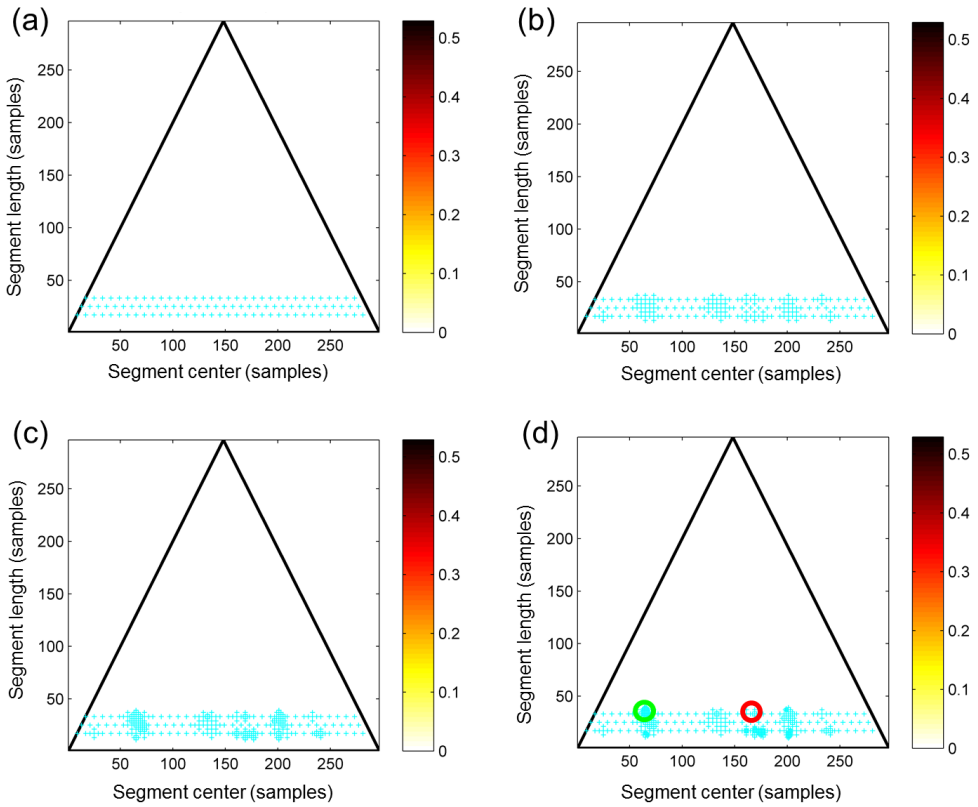

Figure 7.10: Illustration of the sampling procedure for computing an optimal pair of segments. Using Beatles "Devil In Her Heart" as example. **(a)** The grid sampling in the fitness scape plot with grid parameter $d_1 = 8$. **(b)**First time refinement for top M pairs of points, with $M = 50$ and refinement parameter $d_2 = 4$. **(c)** Second time refinement, with $M = 50$ and $d_3 = 2$. **(d)** Third time refinement, with $M = 50$ and $d_4 = 1$. The procedure stops at this stage and the point pair of maximal joint fitness is indicated by the green and red circles.

Extending Equation (3.20), we define the joint thumbnail with the length lower bounds as:

$$(\alpha, \beta)^*_{(\theta_1,\theta_2)} := \underset{(\alpha,\beta),|\alpha|\geq\theta_1,|\beta|\geq\theta_2}{\operatorname{argmax}} \varphi'(\alpha, \beta). \tag{7.16}$$

7.3.3 Practical Computation

The core idea of the joint approach is to jointly estimate the two most repetitive segments and their corresponding induced segments. In order to pick out the most repetitive segments, we need to compute joint fitness values for all possible pairs of segments, and then select the segment pair of maximum joint fitness. Theoretically, this is not a big issue to compute. However, when we deal with real audio recordings, computing joint fitness values for all possible segment pairs is nearly an infeasible task. To proof this, suppose we have feature sequence indexed by $[1 : N]$. Considering the shortest segment having

only one feature and the longest segment having N features, the number of all possible segments in the recording is $M_s = N \cdot (N+1)/2$. We can see that M_s goes quadratic as N increases. Furthermore, suppose P_s is the number of segment pairs [2], then we have $P_s = \binom{M_s}{2} = \frac{M_s \cdot (M_s - 1)}{2}$. We can also find that P_s goes quadratic as M_s increases. Since we have two times quadratic relation, P_s goes to the power of four as N increases. Therefore, the computation is very expensive when N is large. Even for a small value of N, the computation is still very costly. Therefore, in order to counter this problem, instead of computing all possible pairs of segments, we exploit an acceleration strategy to compute only a certain amount of segment pairs, and select the maximum from them. The idea of this acceleration is to sample some points in the fitness scape plot and refine the neighbors for those points having high joint fitness values. This is very similar to the strategy we used for increasing the computation speed of our thumbnailing procedure as introduced in Section 4.1.1. The underlying assumption is that, for segment pairs having high joint fitness values, their neighbor points also have high fitness. By iteratively refining neighbor points around those pairs of points with high joint fitness, we can finally approximately get to the point pair with maximal joint fitness.

To illustrate our acceleration strategy, we take again the Beatles song "Devil In Her Heart" as an example. This song was also used in Figure 7.5 and Figure 7.6. First, to start the acceleration procedure, we consider a regular grid sampling as shown in Figure 7.10a. The distance between neighboring points in either vertical or horizontal direction is controlled by a parameter of d_1 in samples. In this example, we set $d_1 = 8$ samples. Since our aim is to search for the optimal pair of segments, we do not need to consider those segments that can never be in the optimal pair. Therefore, we focus on segments having an appropriate length corresponding to a musical part. Such segments may have a length of 10 to 20 seconds corresponding to a verse or a chorus section of a pop song. To realize this, we introduce two length parameters to control the amount of segments: $\theta_l \in \mathbb{N}$ as the lower bound of length and $\theta_u \in \mathbb{N}$ as the upper bound of length. In Figure 7.10a, we set $\theta_l = 20$ and $\theta_u = 40$. Since we use a feature resolution of 2 Hz, these values corresponds to 10 and 20 seconds, respectively. The joint fitness of each possible pair of grid points are then computed. These points are the first points that we include in the point collection.

Next, we perform our refinement strategy which include some new points into the point collection. By iteratively including new points with high joint fitness in the neighbor of previously selected points, we can finally estimate the optimal pair of points approximately in the scape plot [3]. To this aim, we perform the refinement in an iterative fashion with each iteration consists of the following three steps: firstly, we select $M \in \mathbb{N}$ pairs of points having the top M joint fitness values among all pairs of points in each iteration. Such points are named anchor points. Secondly, we refine the neighbors around these anchor points. Using a fine grid parameter $d_2 \in \mathbb{N}$ with $1 \leq d_2 < d_1$ (in our scenario, we use $d_2 = d_1/2$, assuming d_1 is a power of two), we newly include the four (up, down, left and right) neighbors of the anchor points, which are d_2 samples apart from them (these neighbors form a finer grid). These newly added points, together with the already selected points, form the point collection of the current iteration. Finally, we compute the joint

[2] In order to estimate computation complexity, here we do not consider whether the two segments in the segment pair overlapped with each other or not.

[3] This is similar to the point selection strategy introduced in Section 4.1.1.

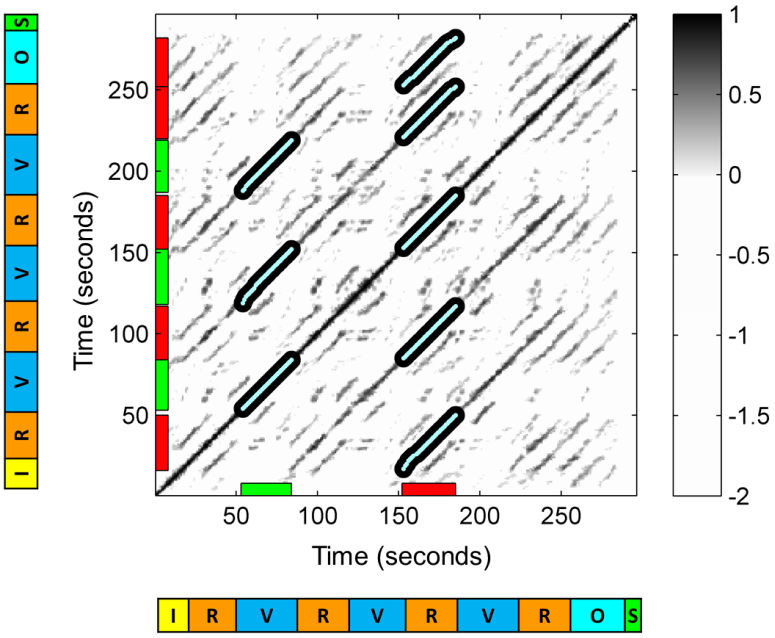

Figure 7.11: Illustration of the joint thumbnails computed for the Beatles song "Devil In Her Heart" using the joint approach. The joint thumbnails (horizontal axis) exactly correspond to a V (verse) and a R (refrain) section in the ground truth segmentation (indicated by the colored rectangles). We also present the computed optimal joint path family (cyan paths), and its induced segments (vertical axis) which correspond to the two main repetitive verse and refrain parts in the ground truth. Note that the O (outro) part annotated in the ground truth is actually a fading version of the refrain part.

fitness for all possible pairs of points in the collection.

We then iterate all these three steps until the fine grid parameter reach a previously settled minimum value d_{stop}. In our scenario, we set $d_{stop} = 1$. Figure 7.10a illustrate the grid sampling with a grid parameter to be $d_1 = 8$. The three refinement iterations are shown in Figure 7.10(b), (c) and (d) with the pairs of points selection parameter $M = 50$ and a refinement distance $d_2 = 4, d_3 = 2$, and $d_4 = 1$ in each iteration respectively.

Figure 7.10d shows the resulting points selected by sampling and refinement after the procedure stops. We then compute joint fitness values for all possible combination of segment pairs in the resulting points. Here, the segment pairs which do not fulfilled the disjoint condition are excluded from the computation. The optimal segment pair having maximum joint fitness is then identified, as shown by the red and green circles in Figure 7.10d. In addition, we compute the optimal path family for this pair of points and illustrate them in Figure 7.11. The induced segments on the vertical axis clearly reflect the repetitiveness of the pair of segments which derived by the selected pair of points in the scape plot. These induced segments are then considered as two groups of segments. We then assign labels to these two groups of segments with "A" and "B" and present them as the final structure estimation result, as can be seen in Figure 7.12b, which will

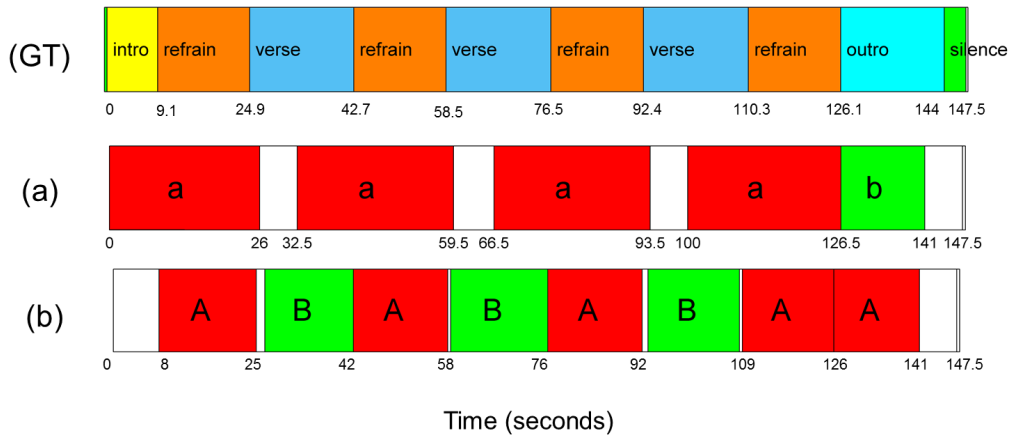

Figure 7.12: Illustration of the structure annotation and structure estimation result for the Beatles song "Devil In Her Heart". **(GT)** Ground truth annotation. **(a)** Estimation result by the iterative approach. **(b)** Estimation result by the joint approach.

be discussed in the next section.

7.4 Evaluation

In this section, we investigate the behavior of our iterative approach as well as the joint approach using several qualitative illustrative examples as well as some quantitative evaluation results.

Note that in the following result visualization figures, we use lowercase letters such as "a" and "b" to indicate segments derived from the iterative approach, whereas the uppercase letters such as "A" and "B" are used to indicate segments derived from the joint approach. In addition, all the result visualization figures in this section are presented in the same style: the sub-figure at the top shows the ground truth structure annotation. The middle one shows the structure estimation derived from the iterative approach as introduced in Section 7.2. The sub-figure at the bottom shows the result from the joint approach as introduced in Section 7.3. Such arrangement of figures allows us to easily compare the segmentation results yielded by the two approaches with the ground truth annotation. All the test data used in this section are Beatles recordings as introduced in Section 3.7.1.

7.4.1 Qualitative Examples

Example 1: "Devil In Her Heart".

We start with the structure estimation result of our running example: the Beatles song "Devil In Her Heart", as shown in Figure 7.12. By visual comparison of the two structure estimation results in Figure 7.12, we find that the joint approach performs much better than the iterative approach. This is mainly because the joint approach balances the

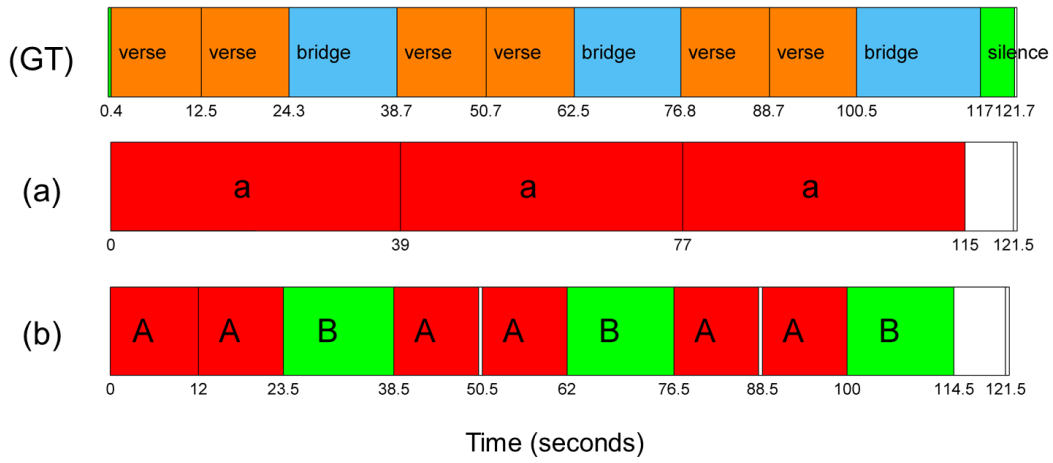

Figure 7.13: Illustration of the structure annotation and structure estimation result for Beatles song "For No One". **(GT)** Ground truth annotation. **(a)** Structure estimation result of the iterative approach. **(b)** Structure estimation result of the joint approach.

contributions from the two groups of repetitive segments, whereas the iterative approach only focuses on one repetitive group of segments without considering its influence for others.

After a careful inspection of the audio recording and the ground truth annotation, we found that the annotated intro part actually shares the same harmony as the verse part, but the melody of the intro is a shortened version of the melody in the verse part. In addition, the annotated outro part is an altered version of the refrain. In the iterative approach, since we use chroma features which capture harmonic aspects of a music recording, the thumbnailing procedure, which is computed based on chroma features, greedily selects segments whose underlying harmonic repetitions get high fitness score, and cover a large portion of the song. Therefore, the iterative approach picks up the longest well-repeated harmony of this song in the first round computation, which is a segment that corresponds to an entire refrain section and a portion of a verse section, see "a" segments in Figure 7.12a. Furthermore, after the first round computation, the iterative approach continues to the second round and estimates the second thumbnail, see the "b" segment that corresponds to "outro" in Figure 7.12a. However, because all induced segments of this second thumbnail overlap with those "a" segments, they are excluded from the final result. This is why we only see one "b" segment in the result visualization.

As we can see the structure estimation result by the joint approach in Figure 7.12b, the "A" segments correspond to all refrain sections and the outro section, and the "B" segments correspond to all verse sections in the annotation. In this example, the joint fitness measure used in this joint approach gets an optimal balance between the two repetitive refrain and verse sections. Therefore it successfully estimates the appropriate lengths of the two repetitive segments.

Example 2: "For No One"

As the second example, we compare the results of the two approaches computed for the

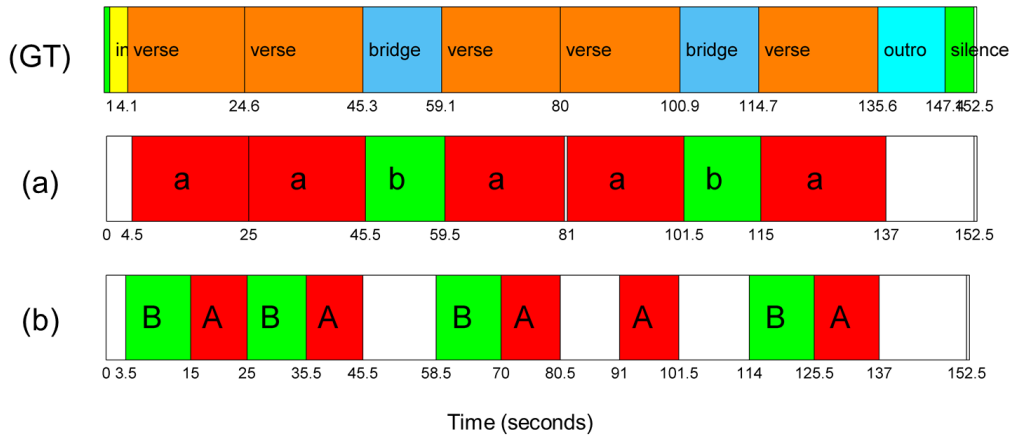

Figure 7.14: Illustration of the structure annotation and structure estimation result for Beatles song "A Hard Day's Night". **(GT)** Ground truth annotation. **(a)** Structure estimation result of the iterative approach. **(b)** Structure estimation result of the joint approach.

Beatles song "For No One", see Figure 7.13. According to the ground truth annotation which is shown at the top of this figure, this song mainly consists of two repeating parts, the verse part and the bridge part. However, in this song, the combination of two consecutive verse sections and one refrain section also forms a longer repetitive part on a higher hierarchical level.

This is exactly the result derived from the iterative approach. Instead of estimating segments that correspond to verse or bridge sections, the iterative approach estimated three long segments as "a" segments in the first round of computation. By comparison to the ground truth, we see that these "a" segments correspond to the combination of two verse sections and one bridge section. Next, the approach goes into the second computation round. Since all "a" segments are removed and there are not many regions left in the audio, the approach cannot find any repetitive segments and thus terminates. In this example, the result of the iterative approach shows an obvious under-segmentation problem, which is caused by false merge of segments [76]. This indicates that some of the computed segments should be further separated (see "a" segments).

We now look at the result derived by the joint approach. Different from the iterative approach, the joint approach successfully estimates six "A" segments that correspond to the six "verse" sections in the ground truth, and the three "B" segments that correspond to the three "verse" sections as well. Apparently, for this song, the joint approach outperforms the iterative approach. This is because the joint fitness measure balances the repetitiveness contributions between different kinds of segments.

Example 3: "A Hard Day's Night"

As the third example, we use the Beatles song "A Hard Day's Night" to inspect the behavior of the two approaches. As can be seen from Figure 7.14, the iterative approach outperforms the joint approach for this song.

In the iterative approach, the estimated "a" segments and "b" segments correspond to

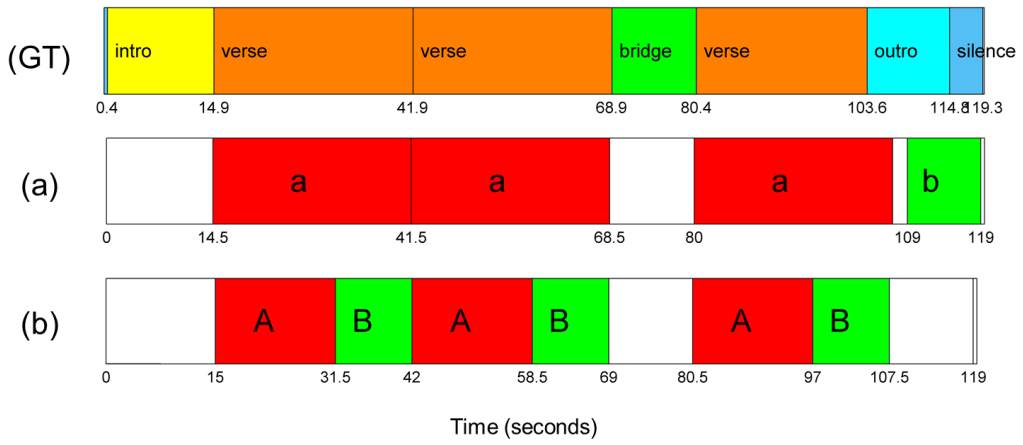

Figure 7.15: Illustration of the structure annotation and structure estimation result for Beatles song "Do You Want To Know A Secret". **(GT)** Ground truth annotation. **(a)** Structure estimation result of the iterative approach. **(b)** Structure estimation result of the joint approach.

verse sections and bridge sections of the ground truth, respectively. However, in the joint approach, the computed "A" and "B" segments only correspond to a portion of annotated sections. In particular, "A" segments correspond to the second half of verse sections while "B" segments correspond to the first half of verse sections. This is a typical over-segmentation problem, which is caused by false fragmentation [76]. In addition, the bridge sections in the ground truth are neglected by the joint approach. This is due to the limitation of the joint approach, that it currently only estimates two most repetitive segments. Therefore, although the bridge sections are also repetitive, the joint approach cannot detect them because it already picked out two repetitive segments. We further discuss this issue in Section 7.5.

Example 4: "Do You Want To Know A Secret"

As the fourth example, we use the Beatles song "Do You Want To Know A Secret" to illustrate the situation that the iterative approach yields a result that is closer to the ground truth compared to the result of the joint approach.

As Figure 7.15 shows, the three estimated "a" segments by the iterative approach correspond to the three verse sections of the ground truth. The first and second "a" segments correctly reflect the lengths of the first two verse sections, however, the third "a" segment is longer than the corresponding verse section in the annotation. Note that there is only one "b" segment estimated by the iterative approach. Theoretically, such segment without repetitions cannot be detected by a repetition-based approach. By inspecting the path family of the "b" segment, we found that there is actually one repetition of the "b" segment from 36 to 44 seconds. However, since this repetition overlaps with one of the previous detected "a" segments, it is discarded by the approach. Therefore, although the "b" segment is repetitive, the final result only contains one "b" segment.

Besides of the identified repetitive segments, we now consider the remaining segments in the audio recording. The first unannotated segment (see the white region at the beginning

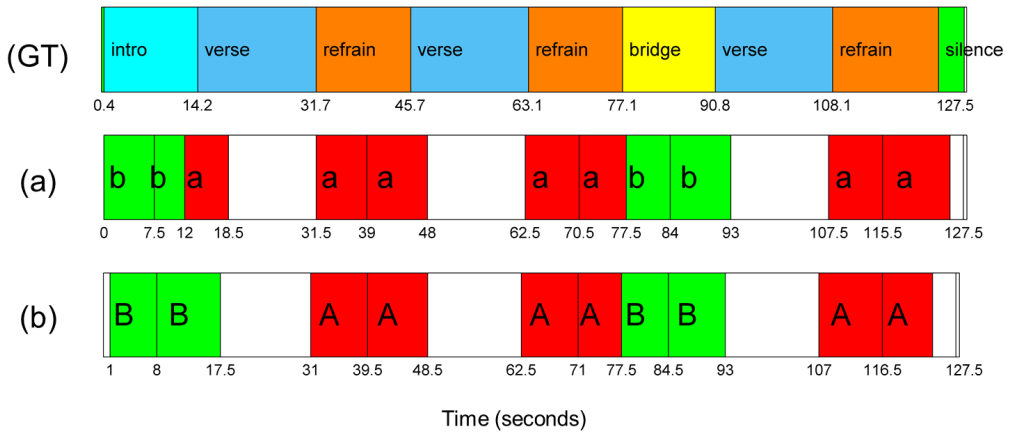

Figure 7.16: Illustration of the structure annotation and structure estimation result for Beatles song "Eleanor Rigby". **(GT)** Ground truth annotation. **(a)** Structure estimation result of the iterative approach. **(b)** Structure estimation result of the joint approach.

of Figure 7.15a) correctly reflects the "intro" part in the ground truth. Also, the second unannotated segment reflects the "bridge" part in the annotation. In this case, the structure estimation result yielded by the iterative approach is very close to the ground truth, except of the last outro part which is estimated with a time delay.

Next, we analyze the result by the joint approach. As can be seen from Figure 7.15b, the "A" and "B" segments derived from the joint approach together reflect the verse sections in the ground truth, but each of them only partially captures some regions of verse sections. This is mainly again due to the nature of the joint approach, that it must take two most repetitive segments as result. Since no other repetitions can be detected in this song, the joint approach just separates a long repeating section into two segments. Besides of the Example 3, this song is another evidence showing that the joint approach may yield an over-segmentation problem. Also, this clearly reveals the main limitations of the joint approach, that it must jointly detect two repetitive segments. If the structure of the audio recording contains three or more repetitive parts, the current joint approach needs to be modified to handle such situations. We discuss this issue in Section 7.5.

Example 5: "Eleanor Rigby"

As the fifth example, the result computed on the Beatles song "Eleanor Rigby" illustrates the situation that the iterative approach and the joint approach sometimes lead to roughly the same results, see Figure 7.16.

In the result of the iterative approach, most "a" segments correspond to sub-sections of the refrain in the ground truth, and the "b" segments correspond to sub-sections of the bridge and the intro. After a listening inspection, we found that although the intro and the bridge are annotated differently, the underlying music content in these two parts is identical. Therefore the "b" segments reflect meaningful repetitive sections.

Both the iterative and the joint approach suffer from the over-segmentation problem. This is due to musical reasons. There are actually shorter phrases repeating inside the refrain

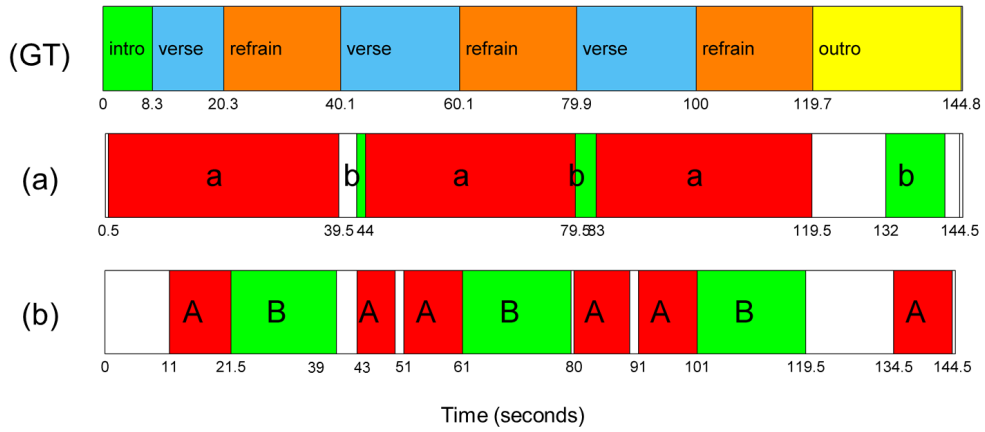

Figure 7.17: Illustration of the structure annotation and structure estimation result for Beatles song "Everybodys Got Something To Hide Except Me And My Monkey". **(GT)** Ground truth annotation. **(a)** Structure estimation result of the iterative approach. **(b)** Structure estimation result of the joint approach.

parts and bridge parts. The refrain part actually consists of two phrases which are in the same melody. The same situation applies for the bridge part. Therefore, both the iterative approach and joint approach capture such short phrases, which are obviously more repetitive than the annotated sections.

To our surprise, the iterative approach did not identify the three "verse" sections of the ground truth. Although these sections are annotated with the same verse label, the music content in these sections contain strong acoustic differences.

Example 6, "Everybodys Got Something To Hide Except Me And My Monkey"

As the final example, we use the Beatles song "Everybodys Got Something To Hide Except Me And My Monkey" to illustrate the situation that neither of the two approaches works well. As shown in Figure 7.17, the iterative approach again has the "A eaten B" problem where the estimated "a" segments in the first round are too long that leads to the incorrect segments estimated in the second round. Also, the joint approach does not work well. Although the "B" segments estimated in the joint approach correspond to three "refrain" sections in the ground truth, the "A" segments are problematic. The first "A" segment roughly corresponds to the first verse section, but all other "A" segments correspond to sub-sections of verse. Actually, after a listening inspection to the audio file, we found that the first verse section is shorter than the other two verse sections because it lacks some musical content. From the result of this song we can see that, when strong musical variations such as lacking of content are present in the recording, it is not enough to use only repetition-based approaches to perform music structure analysis. A possible improvement could be to combine boundary detection methods with the repetition-based approaches.

	P	R	F	diff (F)	max (F)
Iterative	0.71	0.70	0.69	-0.01	0.74
Joint	0.77	0.64	0.68		

Table 7.1: Structure evaluation results using the pairwise frame P/R/F evaluation measure averaged on 178 Beatles recordings.

	P	R	F	diff(F)	max (F)
Iterative	0.53	0.65	0.56	0.02	0.64
Joint	0.52	0.70	0.58		

Table 7.2: Structure evaluation results using the boundary retrieval P/R/F evaluation measure averaged on 178 Beatles recordings.

7.4.2 Quantitative Evaluation

In this section, we describe our systematic evaluations for both the iterative approach and the joint approach to check their general performance. We use the Beatles dataset as described in Section 3.7.1, where we take all 180 recordings of "The Beatles" band and the corresponding structure annotations [82].

For each of the recording, we apply the iterative approach as well as the joint approach and get two kinds of structure estimations. Then, we evaluate both of the structure estimations using the pairwise frame evaluation measure which presented in Precision (P), Recall (R) and F-measure (F) values [71, 76]. In addition, we use the boundary retrieval hit rate P/R/F-values to evaluate the segment boundaries of structure estimation results. We consider a computed segment boundary to be correct when it is within 3 seconds from a real segment boundary in the ground truth. These two evaluation measures are standard measures used in MIREX (Music Information Retrieval Evaluation eXchange) competition for the music structure segmentation task.

Now we describe our parameter settings for the experiments. We use a feature resolution of 2 Hz for all the experiments. For both of the iterative and the joint approach, we set the segment length lower bound as 10 seconds, and the segment length upper bound as 30 seconds. After completing a pilot test, we found that two recordings, "Her Majesty" (duration 24 seconds) and "Maggie Mae" (duration 40 seconds), are too short for our approaches to yield meaningful structure estimation result. Neither of the approaches can find a repeating segment with the constraint of 10 seconds length lower bound. In this case, we exclude these two recordings in our experiments and compute the average structure evaluation results over the remaining 178 recordings.

Table 7.1 shows the result for the averaged pairwise frame evaluation (We present the result for individual recordings in the big Table 7.3). We present result by means of P/R/F values for the iterative approach in the first row, then for the joint approach in the second row. We can find that for the averaged F-measure evaluation, the iterative approach gets 0.69, and the joint approach gets 0.68. Next, the F-measure difference between the two approaches averaged all over the dataset is shown in the third column. The small value -0.01 indicates that the overall performances of both approaches are

roughly similar for the Beatles dataset. After that, in order to see what we can best achieve, we select for each song the better result of the two approaches, and average over all songs to generate the max possible result we can get. This yields an F-measure of 0.74 shown in the last column, which is roughly 0.05 higher compared to either of the two approaches. Such difference indicates that the two approaches actually behave differently on various songs. By individual inspection, we found that the joint approach outperforms the iterative approach far better for some of the songs, but works worse for some other songs, which is mainly due to the over-segmentation problem. Also, it indicates that if we select the appropriate approach for different songs, we can further improve our structure estimation.

Similarly, Table 7.2 shows the evaluation result for the averaged boundary retrieval. The iterative approach gets 0.56 at the averaged F-measure boundary evaluation,, and the joint approach gets 0.58. The averaged difference of -0.01 again reinforce our conclusion that the overall performance of the two approaches averaged on the Beatles data are roughly the same. The averaged maximum F-measure gets 0.74, which is also higher than the result yield by either approach. Such similar behavior again indicates that there could be further improvement for structure estimation if we can select the appropriate approach for different songs.

We see from the evaluation result that our pure repetition-based structure analysis achieve at the Beatles dataset for F-meausre of 0.74 for pairwise frame clustering, and F-meausre of 0.64 for boundary retrieval hit rate. One of the state-of-the-art approaches proposed by Serra et.al [120] is evaluated on the same Beatles dataset and gets 0.77 for pairwise frame F-measure and 0.71 for boundary F-measure. Comparing to their approach, we get only 0.03 lower at the pairwise frame measure, and 0.07 lower at the boundary hit rate.

7.5 Further Notes

Usually, popular music have comparably simple structure with only verse and chorus parts repeating. Other parts, such as intro, outro, or bridge, are often single parts which are not repeated. In such cases, our joint approach, which jointly estimate two most repetitive segments and their repetitions, fits well for this scenario. However, if a music recording has more than two repeating parts, the joint approach cannot estimate all of them. One possible solution is to extend the current joint approach to a more generalized approach. Theoretically, if we could know the number of the repeating parts in the music, we could modify the approach to jointly consider the repetitiveness of the corresponding number of most repetitive segments (multiple thumbnails). This remains a challenging future work, since there is still some practical computation efficiency issue to be solved.

In this section, we have proposed two repetition-based approaches for music structure analysis. These two approaches focus on repeated harmonic content of the music. Other aspects of music, such as melody, timbre or rhythm, is not much considered in the proposed approaches. It would be beneficial to include other aspects of music signal to assist our repetition-based approaches. For example, involving boundary detection algorithms which can restrict the start and end of repetitive segments might improve the performance of our approaches. We have already seen from the qualitative evaluation examples that

one main problem of the approach is the inaccurate segment boundaries. Also some over-segmentation or under-segmentation problems could be to some extent avoided when giving some hints of the segment boundaries.

It is necessary to study how humans segment the a piece of music. For some cases, two segments may be repetitions decided by automated methods, but humans might consider not. One typical example is that in the pop song scenario. The "intro" section might just be an version of "verse" without singing voice, where the automated algorithm might judge them to be repetitions but humans say they are different. Actually, they can be considered as the same section from the harmonic aspect, but from the functional aspect they are not. For such cases, some prior information could be given to bridge the gap between human analysis and machine analysis. For example, when preparing manual annotated structure annotations, it would be better to ask human annotators to generate several annotations each focusing different aspect of music (such as the annotations in the SALAMI dataset [128]). In this way, we could take the annotations which focus on harmonic aspect, to evaluate the algorithms which based on this aspect.

		pairwise frame measure								boundary retrieval measure							
		Iterative			Joint			diff(F)	max(F)	Iterative			Joint			diff(F)	max(F)
		F	P	R	F	P	R			F	P	R	F	P	R		
1	AcrossTheUniverse	0.60	0.55	0.65	0.71	0.77	0.66	0.11	0.71	0.58	0.56	0.60	0.59	0.53	0.67	0.01	0.59
2	ActNaturally	0.81	0.74	0.89	0.67	0.70	0.63	-0.14	0.81	0.78	0.69	0.90	0.43	0.38	0.50	-0.35	0.78
3	ADayInTheLife	0.39	0.37	0.41	0.56	0.55	0.58	0.18	0.56	0.44	0.29	0.92	0.45	0.31	0.83	0.01	0.45
4	AHardDaysNight	0.96	0.95	0.97	0.57	0.91	0.42	-0.39	0.96	0.84	0.89	0.80	0.70	0.62	0.80	-0.15	0.84
5	AllIveGotToDo	0.76	0.77	0.75	0.72	0.80	0.66	-0.04	0.76	0.25	0.22	0.29	0.67	0.63	0.71	0.42	0.67
6	AllMyLoving	0.92	0.88	0.96	0.93	0.90	0.96	0.01	0.93	0.84	0.73	1.00	0.89	0.80	1.00	0.05	0.89
7	AllYouNeedIsLove	0.72	0.60	0.92	0.76	0.77	0.75	0.04	0.76	0.71	0.59	0.91	0.38	0.33	0.45	-0.33	0.71
8	AndILoveHer	0.87	0.86	0.88	0.55	0.75	0.44	-0.32	0.87	0.30	0.27	0.33	0.08	0.06	0.11	-0.22	0.30
9	AndYourBirdCanSing	0.61	0.78	0.51	0.56	0.76	0.44	-0.06	0.61	0.80	0.80	0.80	0.64	0.58	0.70	-0.16	0.80
10	AnnaGoToHim	0.72	0.75	0.68	0.83	0.80	0.86	0.11	0.83	0.60	0.60	0.60	0.60	0.60	0.60	0.00	0.60
11	AnotherGirl	0.95	0.95	0.95	0.93	0.98	0.88	-0.02	0.95	0.88	1.00	0.78	0.60	0.55	0.67	-0.28	0.88
12	AnyTimeAtAll	0.73	0.68	0.80	0.87	0.86	0.88	0.13	0.87	0.50	0.45	0.56	0.67	0.58	0.78	0.17	0.67
13	AskMeWhy	0.60	0.89	0.46	0.55	0.91	0.39	-0.06	0.60	0.73	0.67	0.80	0.64	0.58	0.70	-0.09	0.73
14	ATasteOfHoney	0.47	0.57	0.40	0.50	0.67	0.40	0.03	0.50	0.12	0.09	0.17	0.56	0.42	0.83	0.44	0.56
15	BabyItsYou	0.54	0.83	0.40	0.49	0.87	0.34	-0.05	0.54	0.00	0.00	0.00	0.47	0.33	0.80	0.47	0.47
16	BabysInBlack	0.62	0.61	0.63	0.63	0.67	0.60	0.01	0.63	0.54	0.54	0.54	0.64	0.67	0.62	0.10	0.64
17	BabyYoureARichMan	0.68	0.76	0.62	0.62	0.69	0.56	-0.06	0.68	0.38	0.28	0.63	0.32	0.24	0.50	-0.06	0.38
18	BackInTheUSSR	0.52	0.43	0.67	0.56	0.57	0.55	0.03	0.56	0.26	0.30	0.23	0.48	0.50	0.46	0.22	0.48
19	Because	0.46	0.32	0.78	0.36	0.28	0.50	-0.10	0.46	0.11	0.13	0.10	0.38	0.36	0.40	0.27	0.38
20	BeingForTheBenefitOf	0.55	0.53	0.57	0.39	0.53	0.31	-0.16	0.55	0.67	0.55	0.86	0.45	0.33	0.71	-0.21	0.67
21	Birthday	0.80	0.69	0.96	0.77	0.69	0.86	-0.04	0.80	0.95	1.00	0.90	0.82	0.75	0.90	-0.13	0.95
22	BlackBird	0.78	0.71	0.87	0.53	0.61	0.47	-0.25	0.78	0.55	0.50	0.60	0.44	0.35	0.60	-0.10	0.55
23	BlueJayWay	0.42	0.51	0.36	0.41	0.52	0.34	-0.01	0.42	0.46	0.32	0.82	0.39	0.28	0.64	-0.07	0.46
24	Boys	0.62	0.77	0.53	0.71	0.80	0.64	0.08	0.71	0.58	0.47	0.78	0.74	0.70	0.78	0.15	0.74
25	CantBuyMeLove	0.72	0.84	0.63	0.57	0.67	0.50	-0.15	0.72	0.80	0.80	0.80	0.45	0.42	0.50	-0.35	0.80
26	CarryThatWeight	0.79	0.95	0.68	0.75	0.96	0.61	-0.05	0.79	0.57	0.40	1.00	0.53	0.36	1.00	-0.04	0.57
27	Chains	0.91	0.97	0.86	0.60	0.82	0.47	-0.31	0.91	0.70	0.64	0.78	0.61	0.50	0.78	-0.09	0.70
28	ComeTogether	0.48	0.66	0.38	0.45	0.65	0.34	-0.03	0.48	0.33	0.22	0.70	0.32	0.21	0.60	-0.02	0.33
29	CryBabyCry	0.75	0.78	0.72	0.70	0.78	0.63	-0.05	0.75	0.81	0.73	0.92	0.72	0.69	0.75	-0.09	0.81
30	DearPrudence	0.44	0.72	0.32	0.43	0.81	0.30	-0.00	0.44	0.19	0.13	0.43	0.32	0.21	0.71	0.13	0.32
31	DevilInHerHeart	0.58	0.47	0.76	0.82	0.74	0.92	0.24	0.82	0.42	0.44	0.40	0.70	0.62	0.80	0.27	0.70
32	DigAPony	0.91	0.88	0.94	0.91	0.91	0.92	-0.00	0.91	0.81	0.79	0.85	0.73	0.65	0.85	-0.08	0.81
33	DigIt	0.60	0.98	0.43	0.60	0.99	0.43	0.00	0.60	0.44	0.40	0.50	0.20	0.17	0.25	-0.24	0.44
34	DizzyMissLizzy	0.52	0.83	0.37	0.53	0.82	0.39	0.01	0.53	0.73	0.62	0.89	0.70	0.57	0.89	-0.03	0.73
35	DoctorRobert	0.47	0.57	0.41	0.54	0.71	0.44	0.07	0.54	0.33	0.30	0.38	0.32	0.27	0.38	-0.02	0.33
36	DontBotherMe	0.86	0.84	0.89	0.85	0.84	0.86	-0.01	0.86	0.89	1.00	0.80	0.89	1.00	0.80	0.00	0.89
37	DontPassMeBy	0.39	0.46	0.34	0.62	0.71	0.56	0.23	0.62	0.14	0.10	0.22	0.38	0.29	0.56	0.25	0.38
38	DoYouWantToKnowASecr	0.93	0.89	0.97	0.70	0.92	0.56	-0.23	0.93	0.71	0.71	0.71	0.53	0.50	0.57	-0.18	0.71
39	DriveMyCar	0.65	0.50	0.91	0.82	0.84	0.80	0.17	0.82	0.57	0.60	0.55	0.72	0.64	0.82	0.15	0.72
40	EightDaysAWeek	0.93	0.90	0.97	0.91	0.94	0.88	-0.03	0.93	0.86	0.75	1.00	0.76	0.67	0.89	-0.10	0.86
41	EleanorRigby	0.61	0.64	0.58	0.64	0.72	0.58	0.03	0.64	0.70	0.57	0.89	0.78	0.64	1.00	0.09	0.78
42	EverybodysGotSomethi	0.63	0.55	0.74	0.79	0.81	0.77	0.16	0.79	0.56	0.50	0.63	0.70	0.53	1.00	0.14	0.70
43	EverybodysTryingToBe	0.48	1.00	0.32	0.44	1.00	0.28	-0.04	0.48	0.41	0.30	0.67	0.56	0.44	0.78	0.15	0.56
44	EveryLittleThing	0.85	0.84	0.86	0.78	0.86	0.72	-0.07	0.85	0.75	0.69	0.82	0.72	0.64	0.82	-0.03	0.75
45	FixingAHole	0.71	0.86	0.60	0.70	0.91	0.56	-0.01	0.71	0.53	0.40	0.80	0.52	0.41	0.70	-0.01	0.53
46	Flying	0.68	0.75	0.62	0.49	0.60	0.41	-0.19	0.68	0.30	0.20	0.60	0.40	0.27	0.80	0.10	0.40
47	ForNoOne	0.55	0.51	0.61	0.92	0.94	0.90	0.37	0.92	0.37	0.50	0.30	0.86	0.82	0.90	0.48	0.86
48	ForYouBlue	0.64	0.69	0.60	0.46	0.65	0.36	-0.18	0.64	0.64	0.50	0.88	0.48	0.35	0.75	-0.16	0.64
49	GetBack	0.63	0.56	0.74	0.69	0.64	0.74	0.05	0.69	0.65	0.53	0.83	0.67	0.56	0.83	0.02	0.67
50	GettingBetter	0.63	0.61	0.65	0.62	0.61	0.63	-0.01	0.63	0.53	0.42	0.73	0.58	0.54	0.64	0.05	0.58
51	Girl	0.70	0.56	0.91	0.83	0.82	0.85	0.14	0.83	0.64	0.64	0.64	0.61	0.44	1.00	-0.03	0.64
52	GlassOnion	0.68	0.53	0.94	0.66	0.53	0.88	-0.02	0.68	0.71	0.75	0.67	0.53	0.50	0.56	-0.18	0.71
53	GoldenSlumbers	0.48	0.69	0.37	0.49	0.70	0.37	0.01	0.49	0.25	0.17	0.50	0.27	0.18	0.50	0.02	0.27
54	GoodDaySunshine	0.56	0.47	0.70	0.89	0.96	0.83	0.33	0.89	0.20	0.22	0.18	0.75	0.69	0.82	0.55	0.75
55	GoodMorningGoodMorni	0.61	0.76	0.51	0.59	0.78	0.47	-0.02	0.61	0.57	0.40	1.00	0.62	0.44	1.00	0.04	0.62
56	GoodNight	0.83	0.75	0.94	0.86	0.81	0.92	0.02	0.86	0.64	0.58	0.70	0.73	0.67	0.80	0.09	0.73
57	GotToGetYouIntoMyLif	0.71	0.68	0.75	0.72	0.69	0.76	0.01	0.72	0.45	0.38	0.56	0.60	0.55	0.67	0.15	0.60
58	HappinessIsAWarmGun	0.76	0.78	0.75	0.72	0.77	0.68	-0.04	0.76	0.11	0.07	0.25	0.10	0.06	0.25	-0.01	0.11
59	TheContinuingStoryOf	0.76	0.64	0.96	0.79	0.70	0.90	0.02	0.79	0.91	0.83	1.00	0.82	0.75	0.90	-0.09	0.91
60	HelloGoodbye	0.61	0.54	0.71	0.61	0.53	0.73	0.00	0.61	0.42	0.38	0.45	0.38	0.33	0.45	-0.03	0.42

Table 7.3: Structure evaluation result using the iterative approach as well as the joint approach for the 180 Beatles songs. The table is continued in the next pages.

		pairwise frame measure								boundary retrieval measure							
		Iterative			Joint			diff(F)	max(F)	Iterative			Joint			diff(F)	max(F)
		F	P	R	F	P	R			F	P	R	F	P	R		
61	Help	0.63	0.55	0.72	0.87	0.86	0.87	0.24	0.87	0.23	0.18	0.33	0.64	0.54	0.78	0.41	0.64
62	HelterSkelter	0.46	0.58	0.38	0.33	0.45	0.26	-0.13	0.46	0.35	0.25	0.60	0.36	0.26	0.60	0.01	0.36
63	HereComesTheSun	0.60	0.45	0.91	0.71	0.67	0.75	0.11	0.71	0.56	0.71	0.45	0.67	0.56	0.82	0.11	0.67
64	HereThereAndEverywhe	0.80	0.70	0.94	0.81	0.70	0.95	0.01	0.81	0.84	0.80	0.89	0.94	1.00	0.89	0.10	0.94
65	HerMajesty	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
66	HoldMeTight	0.93	0.91	0.96	0.97	0.95	0.99	0.04	0.97	0.82	0.88	0.78	0.70	0.64	0.78	-0.12	0.82
67	HoneyDont	0.60	0.50	0.74	0.73	0.85	0.64	0.13	0.73	0.43	0.50	0.38	0.61	0.50	0.77	0.17	0.61
68	HoneyPie	0.84	0.78	0.91	0.86	0.79	0.94	0.02	0.86	0.78	0.75	0.82	0.86	0.90	0.82	0.07	0.86
69	IAmTheWalrus	0.54	0.51	0.58	0.57	0.56	0.59	0.03	0.57	0.39	0.35	0.43	0.34	0.33	0.36	-0.04	0.39
70	IDontWantToSpoilTheP	0.53	0.59	0.48	0.53	0.61	0.46	-0.00	0.53	0.67	0.64	0.70	0.80	0.80	0.80	0.13	0.80
71	IflFell	0.67	0.95	0.52	0.67	0.95	0.52	0.00	0.67	0.42	0.29	0.71	0.38	0.29	0.57	-0.04	0.42
72	IfINeededSomeone	0.82	0.93	0.74	0.84	0.96	0.74	0.01	0.84	0.90	1.00	0.82	0.90	1.00	0.82	0.00	0.90
73	IllBeBack	0.91	0.93	0.89	0.90	0.97	0.84	-0.02	0.91	0.70	0.67	0.73	0.59	0.50	0.73	-0.10	0.70
74	IllCryInstead	0.95	0.95	0.94	0.94	0.95	0.93	-0.01	0.95	0.93	1.00	0.88	0.93	1.00	0.88	0.00	0.93
75	IllFollowTheSun	0.91	0.92	0.89	0.88	0.92	0.83	-0.03	0.91	0.80	0.73	0.89	0.64	0.54	0.78	-0.16	0.80
76	ImALoser	0.69	0.60	0.82	0.85	0.79	0.91	0.16	0.85	0.64	0.58	0.70	0.82	0.75	0.90	0.18	0.82
77	IMeMine	0.91	0.99	0.85	0.94	0.99	0.89	0.02	0.94	0.59	0.50	0.71	0.53	0.42	0.71	-0.06	0.59
78	ImHappyJustToDanceWi	0.77	0.67	0.90	0.86	0.79	0.94	0.09	0.86	0.59	0.63	0.56	0.94	1.00	0.89	0.35	0.94
79	ImLookingThroughYou	0.92	0.95	0.89	0.92	0.97	0.88	0.01	0.92	0.80	0.86	0.75	0.71	0.67	0.75	-0.09	0.80
80	ImOnlySleeping	0.53	0.42	0.69	0.59	0.60	0.57	0.06	0.59	0.42	0.50	0.36	0.71	0.59	0.91	0.29	0.71
81	ImSoTired	0.64	0.76	0.55	0.62	0.78	0.52	-0.01	0.64	0.48	0.36	0.71	0.57	0.43	0.86	0.10	0.57
82	INeedYou	0.94	0.95	0.92	0.93	0.95	0.90	-0.01	0.94	0.70	0.64	0.78	0.78	0.78	0.78	0.08	0.78
83	InMyLife	0.62	0.52	0.75	0.92	0.89	0.96	0.30	0.92	0.44	0.50	0.40	0.76	0.73	0.80	0.32	0.76
84	ISawHerStandingThere	0.58	0.50	0.70	0.64	0.56	0.75	0.06	0.64	0.55	0.67	0.46	0.58	0.64	0.54	0.04	0.58
85	IShouldHaveKnownBett	0.77	0.80	0.74	0.81	0.81	0.81	0.04	0.81	0.43	0.38	0.50	0.27	0.25	0.30	-0.16	0.43
86	ItsOnlyLove	0.78	0.79	0.77	0.65	0.72	0.60	-0.12	0.78	0.32	0.25	0.43	0.21	0.17	0.29	-0.11	0.32
87	ItWontBeLong	0.65	0.54	0.81	0.74	0.69	0.80	0.09	0.74	0.76	0.80	0.73	0.95	1.00	0.91	0.19	0.95
88	IveGotAFeeling	0.41	0.62	0.30	0.40	0.62	0.30	-0.01	0.41	0.19	0.14	0.30	0.19	0.14	0.30	0.00	0.19
89	IveJustSeenAFace	0.57	0.52	0.63	0.82	0.83	0.81	0.25	0.82	0.55	0.75	0.43	0.89	0.92	0.86	0.34	0.89
90	IWannaBeYourMan	0.73	0.76	0.70	0.50	0.62	0.42	-0.23	0.73	0.55	0.46	0.67	0.21	0.20	0.22	-0.33	0.55
91	IWantToTellYou	0.85	0.81	0.90	0.83	0.82	0.85	-0.02	0.85	0.60	0.55	0.67	0.80	0.73	0.89	0.20	0.80
92	IWantYou	0.74	0.69	0.80	0.67	0.67	0.68	-0.06	0.74	0.33	0.21	0.78	0.33	0.21	0.78	-0.01	0.33
93	IWill	0.86	0.85	0.86	0.67	0.79	0.59	-0.18	0.86	0.63	0.50	0.83	0.63	0.50	0.83	0.00	0.63
94	Julia	0.84	0.83	0.85	0.57	0.71	0.48	-0.27	0.84	0.57	0.50	0.67	0.64	0.50	0.89	0.07	0.64
95	KansasCityHeyHeyHeyH	0.44	0.86	0.30	0.40	0.89	0.26	-0.04	0.44	0.46	0.35	0.67	0.36	0.31	0.44	-0.10	0.46
96	LetItBe	0.43	0.39	0.49	0.59	0.62	0.57	0.16	0.59	0.21	0.16	0.31	0.71	0.61	0.85	0.50	0.71
97	LittleChild	0.57	0.53	0.62	0.92	0.93	0.91	0.35	0.92	0.67	0.57	0.80	0.84	0.89	0.80	0.18	0.84
98	LongLongLong	0.54	0.52	0.56	0.50	0.55	0.46	-0.04	0.54	0.44	0.32	0.70	0.25	0.21	0.30	-0.19	0.44
99	LovelyRita	0.38	0.35	0.42	0.59	0.66	0.54	0.21	0.59	0.40	0.30	0.60	0.67	0.57	0.80	0.27	0.67
100	LoveMeDo	0.81	0.88	0.76	0.55	0.82	0.41	-0.26	0.81	0.75	0.75	0.75	0.53	0.45	0.63	-0.22	0.75
101	LoveYouTo	0.50	0.37	0.77	0.49	0.43	0.56	-0.01	0.50	0.18	0.17	0.20	0.08	0.07	0.10	-0.10	0.18
102	LucyInTheSkyWithDiam	0.85	0.86	0.84	0.90	0.92	0.88	0.05	0.90	0.71	0.63	0.83	0.71	0.63	0.83	0.00	0.71
103	MaggieMae	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
104	MagicalMysteryTour	0.94	0.92	0.96	0.88	0.90	0.87	-0.06	0.94	0.75	0.69	0.82	0.72	0.64	0.82	-0.03	0.75
105	MarthaMyDear	0.66	0.66	0.67	0.67	0.66	0.68	0.00	0.67	0.67	0.60	0.75	0.74	0.64	0.88	0.07	0.74
106	MaxwellsSilverHammer	0.57	0.62	0.53	0.82	0.99	0.70	0.25	0.82	0.57	0.43	0.83	0.71	0.58	0.92	0.14	0.71
107	MeanMrMustard	0.84	1.00	0.72	0.53	0.99	0.36	-0.31	0.84	0.33	0.25	0.50	0.22	0.14	0.50	-0.11	0.33
108	Michelle	0.50	0.42	0.61	0.89	0.87	0.91	0.40	0.89	0.31	0.29	0.33	0.86	1.00	0.75	0.55	0.86
109	Misery	0.77	0.68	0.89	0.83	0.80	0.86	0.06	0.83	0.63	0.71	0.56	0.67	0.58	0.78	0.04	0.67
110	Money	0.55	0.57	0.54	0.51	0.59	0.44	-0.05	0.55	0.52	0.43	0.67	0.70	0.57	0.89	0.17	0.70
111	MotherNaturesSon	0.80	0.75	0.86	0.82	0.76	0.88	0.01	0.82	0.70	0.64	0.78	0.70	0.64	0.78	0.00	0.70
112	MrMoonlight	0.57	0.68	0.50	0.48	0.76	0.35	-0.10	0.57	0.67	0.55	0.86	0.50	0.35	0.86	-0.17	0.67
113	NoReply	0.89	0.92	0.86	0.93	0.93	0.92	0.04	0.93	0.63	0.50	0.83	0.71	0.63	0.83	0.09	0.71
114	NorwegianWoodThisBir	0.75	0.76	0.74	0.76	0.80	0.72	0.01	0.76	0.64	0.54	0.78	0.70	0.57	0.89	0.06	0.70
115	NotASecondTime	0.91	0.90	0.92	0.97	0.98	0.97	0.07	0.97	0.89	0.89	0.89	0.89	0.89	0.89	0.00	0.89
116	NowhereMan	0.65	0.68	0.63	0.72	0.80	0.66	0.07	0.72	0.54	0.47	0.64	0.69	0.56	0.91	0.15	0.69
117	ObLaDiObLaDa	0.62	0.51	0.78	0.71	0.78	0.65	0.09	0.71	0.61	0.70	0.54	0.65	0.52	0.85	0.04	0.65
118	OctopussGarden	0.84	0.94	0.76	0.64	0.92	0.49	-0.20	0.84	0.31	0.21	0.57	0.44	0.30	0.86	0.14	0.44
119	OhDarling	0.82	0.99	0.70	0.84	0.95	0.75	0.02	0.84	0.57	0.46	0.75	0.52	0.40	0.75	-0.05	0.57
120	OneAfter909	0.72	0.87	0.61	0.54	0.76	0.43	-0.17	0.72	0.84	0.89	0.80	0.70	0.62	0.80	-0.15	0.84

Table 7.4: Continuation of Table 7.3.

		pairwise frame measure								boundary retrieval measure							
		Iterative			Joint			diff(F)	max(F)	Iterative			Joint			diff(F)	max(F)
		F	P	R	F	P	R			F	P	R	F	P	R		
121	PennyLane	0.84	0.81	0.87	0.84	0.83	0.85	0.00	0.84	0.80	0.71	0.91	0.74	0.63	0.91	-0.06	0.80
122	Piggies	0.77	0.66	0.94	0.63	0.72	0.55	-0.15	0.77	0.63	0.67	0.60	0.45	0.42	0.50	-0.18	0.63
123	PleaseMisterPostman	0.63	0.73	0.56	0.57	0.62	0.53	-0.06	0.63	0.55	0.47	0.67	0.73	0.80	0.67	0.18	0.73
124	PleasePleaseMe	0.92	0.89	0.94	0.66	0.86	0.54	-0.25	0.92	0.83	1.00	0.71	0.63	0.56	0.71	-0.21	0.83
125	PolythenePam	0.46	0.48	0.43	0.68	0.68	0.69	0.22	0.68	0.31	0.25	0.40	0.40	0.40	0.40	0.09	0.40
126	PSILoveYou	0.74	0.63	0.90	0.57	0.59	0.55	-0.17	0.74	0.50	0.57	0.44	0.20	0.18	0.22	-0.30	0.50
127	Revolution1	0.44	0.48	0.40	0.44	0.47	0.41	-0.00	0.44	0.21	0.15	0.33	0.20	0.14	0.33	-0.01	0.21
128	Revolution9	0.45	0.67	0.34	0.43	0.68	0.31	-0.02	0.45	0.04	0.02	0.33	0.04	0.02	0.33	0.00	0.04
129	RockAndRollMusic	0.63	0.51	0.82	0.73	0.63	0.86	0.09	0.73	0.90	1.00	0.82	0.91	0.91	0.91	0.01	0.91
130	RockyRaccoon	0.55	0.52	0.60	0.54	0.53	0.55	-0.01	0.55	0.58	0.45	0.82	0.62	0.50	0.82	0.04	0.62
131	RollOverBeethoven	0.61	0.60	0.62	0.58	0.67	0.51	-0.02	0.61	0.10	0.09	0.10	0.09	0.08	0.10	-0.00	0.10
132	RunForYourLife	0.84	0.76	0.94	0.57	0.69	0.49	-0.27	0.84	0.70	0.70	0.70	0.38	0.36	0.40	-0.32	0.70
133	SavoyTruffle	0.70	0.58	0.90	0.71	0.59	0.89	0.00	0.71	0.67	0.73	0.62	0.67	0.73	0.62	0.00	0.67
134	SexySadie	0.63	0.54	0.77	0.49	0.52	0.47	-0.14	0.63	0.60	0.50	0.75	0.58	0.44	0.88	-0.02	0.60
135	SgtPeppersLonelyHear	0.49	0.42	0.59	0.48	0.41	0.58	-0.01	0.49	0.67	0.50	1.00	0.67	0.50	1.00	0.00	0.67
136	SgtPeppersLonelyHear	0.41	0.82	0.28	0.48	0.86	0.33	0.06	0.48	0.22	0.14	0.50	0.00	0.00	0.00	-0.22	0.22
137	SheCameInThroughTheB	0.85	0.85	0.86	0.84	0.86	0.82	-0.02	0.85	0.67	0.67	0.67	0.57	0.50	0.67	-0.10	0.67
138	SheSaidSheSaid	0.93	0.89	0.97	0.93	0.89	0.97	0.00	0.93	0.88	1.00	0.78	0.88	1.00	0.78	0.00	0.88
139	ShesLeavingHome	0.95	0.96	0.93	0.95	0.97	0.93	0.00	0.95	0.90	1.00	0.82	0.86	0.90	0.82	-0.04	0.90
140	Something	0.91	0.97	0.85	0.67	0.86	0.55	-0.24	0.91	0.84	0.73	1.00	0.40	0.33	0.50	-0.44	0.84
141	StrawberryFieldsFore	0.42	0.43	0.40	0.47	0.49	0.46	0.05	0.47	0.25	0.21	0.30	0.55	0.50	0.60	0.30	0.55
142	SunKing	0.67	0.72	0.63	0.67	0.69	0.64	-0.00	0.67	0.20	0.14	0.33	0.22	0.17	0.33	0.02	0.22
143	Taxman	0.58	0.51	0.67	0.59	0.53	0.65	0.01	0.59	0.55	0.75	0.43	0.69	0.75	0.64	0.15	0.69
144	TellMeWhatYouSee	0.90	0.86	0.95	0.92	0.90	0.94	0.01	0.92	0.89	0.89	0.89	0.89	0.89	0.89	0.00	0.89
145	TellMeWhy	0.82	0.84	0.80	0.82	0.84	0.80	0.00	0.82	0.61	0.50	0.78	0.61	0.50	0.78	0.00	0.61
146	TheEnd	0.87	0.84	0.91	0.85	0.84	0.86	-0.02	0.87	0.27	0.17	0.67	0.21	0.13	0.67	-0.06	0.27
147	TheFoolOnTheHill	0.55	0.48	0.65	0.56	0.56	0.55	0.01	0.56	0.57	0.60	0.55	0.29	0.24	0.36	-0.29	0.57
148	TheLongAndWindingRoa	0.68	0.89	0.55	0.63	0.91	0.48	-0.05	0.68	0.21	0.18	0.25	0.76	0.62	1.00	0.55	0.76
149	TheNightBefore	0.92	0.88	0.97	0.54	0.74	0.42	-0.39	0.92	0.95	1.00	0.90	0.72	0.60	0.90	-0.23	0.95
150	TheresAPlace	0.88	0.81	0.96	0.61	0.75	0.52	-0.26	0.88	0.57	0.57	0.57	0.71	0.60	0.86	0.13	0.71
151	TheWord	0.67	0.74	0.61	0.67	0.85	0.56	0.00	0.67	0.73	0.80	0.67	0.80	1.00	0.67	0.07	0.80
152	ThingsWeSaidToday	0.58	0.75	0.47	0.49	0.77	0.36	-0.09	0.58	0.54	0.41	0.78	0.56	0.44	0.78	0.02	0.56
153	ThinkForYourself	0.81	0.79	0.84	0.83	0.82	0.84	0.02	0.83	0.57	0.55	0.60	0.76	0.73	0.80	0.19	0.76
154	TicketToRide	0.94	0.98	0.90	0.96	0.97	0.95	0.02	0.96	0.76	0.67	0.89	0.67	0.58	0.78	-0.10	0.76
155	TillThereWasYou	0.62	0.69	0.57	0.74	0.88	0.64	0.12	0.74	0.67	0.64	0.70	0.76	0.73	0.80	0.10	0.76
156	TomorrowNeverKnows	0.67	0.91	0.53	0.41	0.66	0.30	-0.25	0.67	0.76	0.80	0.73	0.58	0.45	0.82	-0.18	0.76
157	TwistAndShout	0.77	0.90	0.68	0.79	0.92	0.69	0.01	0.79	0.47	0.40	0.57	0.56	0.45	0.71	0.08	0.56
158	TwoOfUs	0.91	0.94	0.88	0.64	0.80	0.53	-0.27	0.91	0.72	0.60	0.90	0.38	0.31	0.50	-0.34	0.72
159	Wait	0.57	0.54	0.62	0.77	0.89	0.69	0.20	0.77	0.67	0.78	0.58	0.92	0.92	0.92	0.25	0.92
160	WhatGoesOn	0.60	0.68	0.55	0.79	0.95	0.68	0.18	0.79	0.30	0.24	0.40	0.84	0.89	0.80	0.55	0.84
161	WhatYoureDoing	0.70	0.74	0.66	0.82	0.91	0.74	0.12	0.82	0.60	0.60	0.60	0.89	1.00	0.80	0.29	0.89
162	WhenIGetHome	0.58	0.49	0.72	0.68	0.68	0.68	0.10	0.68	0.44	0.50	0.40	0.27	0.25	0.30	-0.17	0.44
163	WhenImSixtyFour	0.90	0.86	0.94	0.99	0.99	0.98	0.09	0.99	0.71	0.83	0.63	0.86	1.00	0.75	0.14	0.86
164	WhileMyGuitarGentlyW	0.66	0.53	0.88	0.69	0.55	0.91	0.03	0.69	0.53	0.36	1.00	0.52	0.37	0.88	-0.01	0.53
165	WhyDontWeDoItInTheRo	0.41	0.97	0.26	0.37	0.95	0.23	-0.04	0.41	0.35	0.23	0.75	0.38	0.25	0.75	0.02	0.38
166	WildHoneyPie	0.75	0.70	0.82	0.75	0.71	0.80	-0.00	0.75	0.77	1.00	0.63	0.80	0.86	0.75	0.03	0.80
167	WithALittleHelpFromM	0.53	0.44	0.67	0.84	0.89	0.80	0.32	0.84	0.48	0.50	0.45	0.80	0.71	0.91	0.32	0.80
168	WithinYouWithoutYou	0.39	0.74	0.27	0.36	0.69	0.24	-0.03	0.39	0.29	0.19	0.67	0.24	0.16	0.44	-0.06	0.29
169	WordsOfLove	0.68	0.94	0.53	0.57	0.81	0.44	-0.11	0.68	0.82	0.78	0.88	0.64	0.50	0.88	-0.19	0.82
170	YellowSubmarine	0.70	0.71	0.69	0.54	0.65	0.46	-0.16	0.70	0.57	0.42	0.89	0.38	0.29	0.56	-0.19	0.57
171	YerBlues	0.52	0.70	0.41	0.44	0.74	0.32	-0.08	0.52	0.70	0.57	0.89	0.08	0.06	0.11	-0.62	0.70
172	Yesterday	0.84	0.77	0.92	0.85	0.83	0.87	0.01	0.85	0.60	0.55	0.67	0.78	0.78	0.78	0.18	0.78
173	YouCantDoThat	0.91	0.98	0.85	0.56	0.95	0.39	-0.36	0.91	0.76	0.73	0.80	0.56	0.47	0.70	-0.20	0.76
174	YouLikeMeTooMuch	0.87	0.89	0.85	0.59	0.74	0.49	-0.28	0.87	0.73	0.67	0.80	0.67	0.57	0.80	-0.06	0.73
175	YouNeverGiveMeYourMo	0.57	0.63	0.52	0.64	0.78	0.54	0.07	0.64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
176	YouReallyGotAHoldOnM	0.50	0.42	0.61	0.79	0.74	0.85	0.30	0.79	0.76	0.69	0.85	0.88	0.92	0.85	0.12	0.88
177	YoureGoingToLoseThat	0.72	0.59	0.92	0.72	0.61	0.88	0.01	0.72	0.74	0.70	0.78	0.56	0.56	0.56	-0.18	0.74
178	YourMotherShouldKnow	0.63	0.69	0.57	0.57	0.73	0.47	-0.06	0.63	0.61	0.50	0.78	0.67	0.53	0.89	0.06	0.67
179	YouveGotToHideYourLo	0.73	0.80	0.66	0.85	0.96	0.77	0.13	0.85	0.36	0.31	0.44	0.70	0.57	0.89	0.33	0.70
180	YouWontSeeMe	0.79	0.97	0.66	0.64	0.89	0.49	-0.15	0.79	0.73	0.62	0.89	0.62	0.47	0.89	-0.11	0.73
	average	0.69	0.71	0.70	0.68	0.77	0.64	-0.01	0.74	0.56	0.53	0.65	0.58	0.52	0.70	0.02	0.64

Table 7.5: Continuation of Table 7.4.

Chapter 8

Conclusion of the Thesis

In this paper, we have made several contributions to the field of music structure analysis. In particular, we focused on repetition-based approaches which aims at estimating repetitive structures of a music recording.

In Chapter 2, we discussed the similarity matrix, which is a widely used tool for music structure analysis. As the main contribution of Chapter 2, we gathered some existed enhancing techniques for the similarity matrix and presented cleaned MATLAB implementation in a toolbox, called the SM toolbox, which allows for modifying certain properties of similarity matrices. Such enhancing techniques include smoothing, thresholding, as well as strategies which deal with music variations such as tempo change and local transposition. By our illustrative examples, we showed that our enhancing functions significantly improved path-like structures in similarity matrices. This served as a good basis for identifying similar segments which encoded by the path-like structure.

After enhancing the similarity matrix, the next step is to estimate repeated segments in the given music recording based on the paths in the similarity matrix. As the main contribution of Chapter 3, we have introduced a novel fitness measure that assigns a fitness value to each segment that expresses how much and how well the segment "explains" the repetitive structure of the entire recording. Opposed to other approaches which estimate repetitions by performing the path extraction step and grouping step successively, we performed the two steps in an unifying optimization scheme, which yields a trade-off between quantity and length of paths (coverage) and quality of paths (score). By doing so, we avoided making a hard decision on which paths to take when esteeming repetitions for a segment. Instead, we let the optimization scheme to make different but appropriate decisions on which paths to take when estimating repetitions for different segments. Moreover, by computing the fitness values for all possible segments, we selected the audio thumbnail segment (the most representative segment of the audio recording) to be the one having maximum fitness. In the following, we name this fitness-based procedure the thumbnailing procedure. Our experiments have shown that our thumbnailing procedure generally yields good estimates for musically meaningful thumbnails—even in the presence of acoustic and musical variations across repeated segments. In addition, as the second contribution of Chapter 3, we have introduced a scape plot representation that yields a compact visualization of the repetitiveness of all segments of a given music recording.

Using our illustrative example, we showed that our fitness scape plot provided the global repetitive structures of a recording. Also, the hierarchical relationship of long repetitive segments and their sub-segments can also be clearly visualized in our fitness scape plot.

Even though our thumbnailing procedure yields promising thumbnailing results representing the state-of-the-art, it has the drawback of being computationally expensive. As the contributions of Chapter 3, we introduced three acceleration strategies that significantly improve the computational efficiency of our thumbaniling procedure. The main ideas of the three strategies are: multi-level sampling of the scape plot; multi-resolution computing of the SSM; and using already computed fitness values to approximate fitness of similar segments. Our experimental results have shown that, using a combined approach of all three strategies, we obtained significant accelerations while keeping the overall accuracy of the thumbnailing procedure. These substantial accelerations are important steps towards computing a thumbnail on-the-fly, which paves the way to perform thumbaniling in real-time services. Furthermore, our illustrative example showed that the interpolated fitness scape plot computed by the acceleration strategies looks close to the original fitness scape plot. This allowed us to perform fast approximations of the overall repetitive structures for audio recordings. This is especially useful when estimating structures for long recordings (such as symphonies). When we are not interested in the very accurate structure information but rather rough or coarse structure information, such approximation of fitness scape plot will serve as a good overview which can be fast computed.

The fitness scape plot presented in previous chapters only shows the repetitive structures of each single segments without explicitly telling the relationships between these segments. In Chapter 5 we extended the fitness scape plot, and presented a novel structure scape plot that using color to reveal relationship between segments. In this structure scape plot, each point represented an audio segment identified by its length and center. We assigned to each point a color value so that the lightness of the color indicated the repetitiveness of encoded segment, and the hue of the color revealed the relations between different segments. Our qualitative examples demonstrated that the structure scape plots were able to reflect the overall musical form by presenting different parts in different colors for both popular music such as Beatles and short classical pieces like Chopin Mazurkas. Moreover, the repeating subparts of the main parts in the structure could also be reflected. Besides of the successful examples, we also investigated problematic examples that indicated the problems of our proposed visualization technique. For example, this technique is very sensitive to parameter settings. A small change in parameters might result in meaningless visualization. Besides that, the most error-prone step arise from the PCA projection of the distance matrix, which are used for revealing segment relationships. Since we project the matrix from some high dimension space into two dimension space, the distances of segments are not accurate in the lower space, which brings much errors. One may consider using other techniques (such as Multi-dimensional Scaling) to perform this step. Altogether, these problems indicate possible future work to further improve this visualization.

The main contribution of Chapter 6 is that we combined music knowledge with our repetition-detection procedure and analyzed the structures of music pieces composed in sonata form. The sonata form exhibit hierarchical structures, which consists of both large-scale structural parts (coarse structures) and small-scale subparts (fine structures) composed in certain musical rules. Our experiments have shown that the coarse structure

of most Beethoven sonatas could be accurately estimated. Here, the challenges of deriving the coarse structure arose from the long time differences of music content between the large repeating parts. The crucial point was to use chroma features having a low resolution as well as applying a long smoothing length and transposition-invariance in the similarity matrix. As for the estimation of fine structures, we introduced an automated method which measures local harmonic relations. These relations were highly related with the finer substructure. As our experiments showed, we achieved meaningful results for sonatas that roughly follow the musical conventions. However, automated methods reach their limits in the case of complex movements, where composition rules have been broken up. Music pieces in such situations are also difficult for music experts. We hope that even for such complex cases, our proposed representation, which visualize relative transpositions for fine structures, may still yield some musically interesting and intuitive insights into the data, which may be helpful for musicological studies.

In Chapter 7, we introduced automated methods for full structure analysis of music recordings. As the main contributions, we proposed two repetition-based approaches that are extended versions of our thumbnailing procedure: the iterative approach and the joint approach. Here, the iterative approach was to apply our thumbnailing procedure in an iterative fashion in order to derive repetitive sections that correspond to the same musical parts iteratively. In the joint approach, we computed the two most repetitive segments (two thumbnails) within one optimization procedure that tries to jointly maximize the score and coverage of two different disjoint segments. By doing this, we estimated the repetitive sections that correspond to two most repetitive musical parts. Our experiments have shown that these two approaches yielded meaningful structure segmentation results for the case if popular music. The main problem of the iterative approach is that it extracted the repetitive segments in a greedy manner that it did not consider influences for other segments. Therefore, sometimes the segments estimated in the first round computation were too long, which result in the too short segments in the second round computation. The joint approach to some extent compensate this issue by jointly considering the repetitiveness of two groups of repetitive segments. However, sometimes the joint approach suffered from the over-segmentation problem, that it extracted two repetitive sub-segments of one repeating long segment. For such cases, it could be beneficial to include other aspects from the music signal to assist our repetition-based approaches. For example, involving boundary detection algorithms (such as [120]) which can restrict the start and end boundaries of segments may improve the performance of the repetition-based approaches. Also some over-segmentation or under-segmentation problems can be avoided when giving some hints of the segment boundaries.

All methods we presented in this thesis are based on repetitions in the music structure. These methods mainly focus on the harmony aspect of music recordings. This means that other aspects of music, such as melody, timbre, and dynamics, are not so much considered in these methods. However, these aspects are also important cues for segmenting a music recording. One possible future work would be, similar like in [107] to combine other methods, such as novelty-based or homogeneity-based methods, with our repetition-based approaches, to form a late-fusion approach that use various cues to decide the music structure.

Appendix A

Structure Analysis with Boundary Constraints

In the last chapter of the thesis, we have introduced two different repetition-based procedures for the full structure analysis of audio recordings. Although both of the approaches detect meaningful repetitive segments from audio recordings, the estimated boundaries of such segments are sometimes problematic. For the iterative approach, this is because the procedure estimates repetitive segments in a greedy way without considering influences for other segments. For the joint approach, the algorithm suffers mainly because of over-segmentation. The resulting segments of the joint approach might correspond to sub-sections in the ground truth. In this way, some boundaries of these segments are obviously problematic.

In this appendix, as a indication towards extensions of our repetition-based approaches, we consider the problem of integrating segment boundary information into repetition-based procedures. Segment boundaries are often caused by sudden change of melody, tempo, timbre, etc. Such changes are important cues for segmenting music recordings. So far, these changes are not captured by repetition-based procedures. Therefore in this appendix, we include boundary information into our iterative approach and the joint approach. Besides of this, integrating boundary information to our approaches may significantly reduce the number of segments to be considered. Therefore, there are also large benifits in view of reducing the overall running time.

A.1 Different Methods of Boundary Integration

In this section, we discuss different ways of integrating boundary information into our repetition-based approaches. Our main idea is to use these annotated boundaries to restrict the start or end time of the computed segments. We realize this idea by selecting a subset of segments whose boundaries are constrained by annotated boundaries. We visualize these selected segments by means of points in the fitness scape plot. Then we run the repetition-based approaches to decide the relationship of these segments and further derive the structure segmentation.

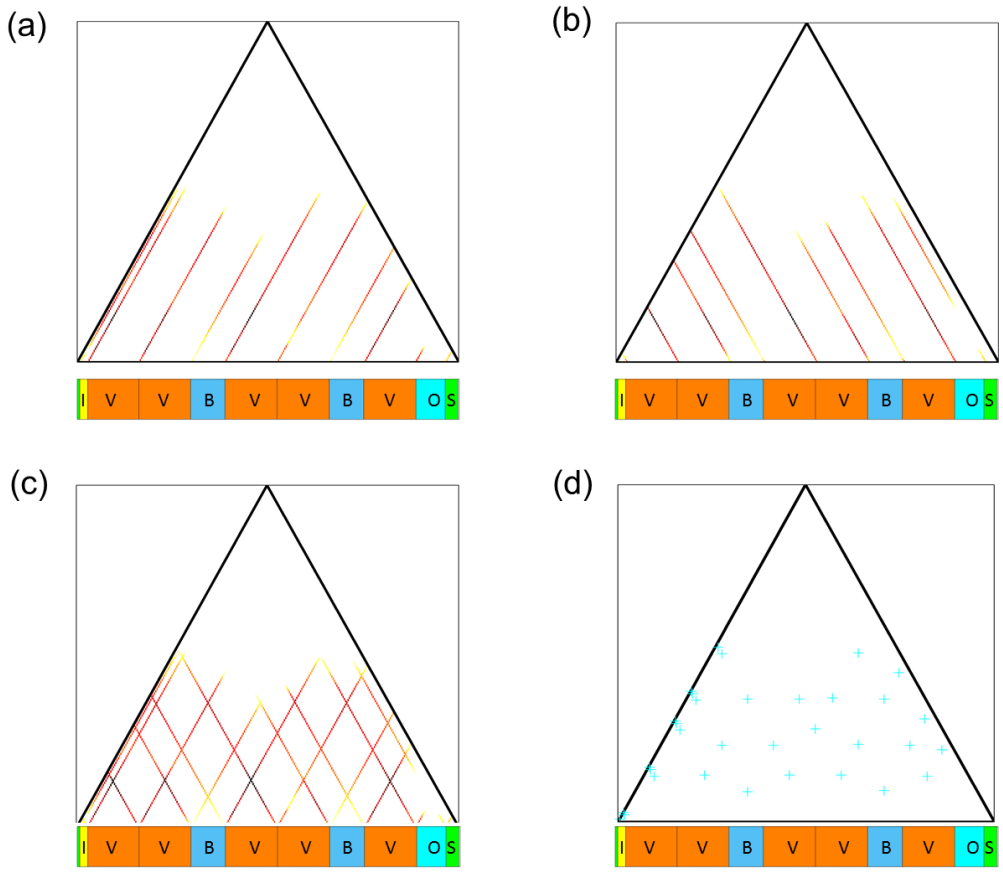

Figure A.1: Illustration of the four different strategies of integrating boundary information into repetition-based approaches. The colored rectangles show the ground truth annotation of the Beatles song "A Hard Day's Night". The lines and points in the scape plots represent the selected segments by the corresponding strategies. **(a)** Segments whose start boundaries are constrained. **(b)** Segments whose end boundaries are constrained. **(c)** Segments whose either start or end boundaries are constrained. **(d)** Segments whose both start and end boundaries are constrained.

In our scenario, we have four strategies of selecting the segments. By using different boundary constraint strategies, different subset of segments are selected, and then pass to the repetition-based approaches to derive the final structure segmentation. These four strategies give the restrictions from loose degree to strict degree. By doing this, we can further test the performance of our repetition-based approaches with different extent of prior knowledge.

In the first strategy, we only constraint start boundaries of segments to be considered. In this case, any segment computed from the repetition-based approaches must start with the given boundaries derived from the ground truth annotation. Since we do not restrict the length of a segment, theoretically any segment which starts at the specified time of the boundary could be a candidate. Since there are many possible candidate segments, instead of visualizing them one by one, we visualize all these segments in the corresponding fitness scape plot as shown in Figure A.1a. The lines in the fitness scape plot are formed by the fitness points whose corresponding segments start at the annotated boundaries.

In order to ignore the segments whose fitness is too low, we visualize the points in their corresponding fitness values.

In the second strategy, we constraint end boundaries of computed segments. In this way, computed segments must end at the given time points derived from the annotations. Similarly as above, since we do not specify the length of segments, such segments can start at any time. The restricted segment candidates are visualized in Figure A.1b.

The above presented strategies consider only one side boundary of segments. Next we discuss strategies that restrict two side boundaries of segments. In the third strategy, the given boundaries can be considered either as start boundaries or end boundaries of segments. Segments selected by this strategy are just the combination of the segments from the two previous strategies. We visualize these selected segments in Figure A.1c.

Finally, in the fourth strategy, we restrict both start and end boundaries of segments. In other words, candidate segments must have one of the annotated boundary as its start boundary, and another one as its end boundary. This is the most strict constraint compared to the previous three strategies. As above, the segment candidates derived from this strategy can be seen in Figure A.1d. Since this strategy select only several segments, opposed to previous strategies, in this figure we use cyan colored "+" points in the fitness scape plot to represent those possible segments.

A.1.1 Comparison of Different Strategies

After discussing the four different strategies of combining the ground truth boundaries with the repetition-based structure analysis, we check the actual performance of the above mentioned strategies in this section.

Recall that we present on page 112 in Figure 7.14 which illustrate the repetition-based structure analysis result for the same song "A Hard Days Night" without any boundary information as prior knowledge. This serves as a baseline for comparing the four strategies. As we can see in Figure 7.14, the structure analysis result of the iterative approach is already good whereas that of the joint approach is problematic. Therefore we focus only on the boundary integration strategies with the joint approach for this song.

Figure A.2 presents the structure analysis result using the joint approach with different boundary integration strategies. As before, the top sub-figure shows the ground truth annotation. Next, we present the pure joint approach result (the same as in Figure 7.14b) in the second sub-figure. This result has an over-segmentation problem, where the extracted "A" segments correspond to only half of the annotated "verse" part in the ground truth. Next, we combine the joint approach with the ground truth boundaries using the four different strategies and present them in other sub-figures of Figure A.2.

As we see from the figure, restricting only the start boundaries of segments (Figure A.2c) or only end boundaries of segments (Figure A.2d), the joint approach yield more or less the same correct structure segmentation. In other words, having such start boundaries or end boundaries as prior knowledge, the joint approaches is able to derive repetitive segments more accurately. Only a slight "shift" happens at the first "B" segment in Figure A.2c. Next, using the third strategy which restrict either start boundaries or end boundaries,

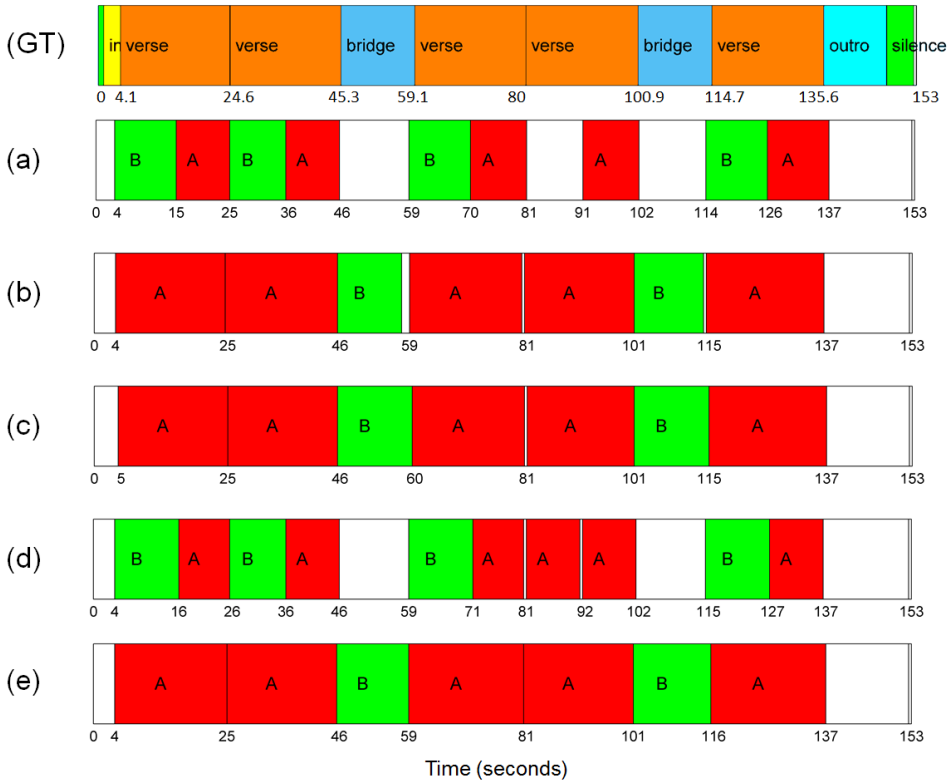

Figure A.2: Structure analysis result of the joint approach with different boundary restriction strategy, using Beatles song "A Hard Days Night". **(GT)** The ground truth segmentation annotation. **(a)** No boundary restriction. **(b)** Restriction on start boundaries of the computed segments. **(c)** Restriction on end boundaries of the computed segments. **(d)** Restriction on either start or end boundaries of the computed segments. **(e)** Restriction on both start and end boundaries of the computed segments.

the joint approach yields again the over-segmentation problem, see Figure A.2d. This is because that the joint approach takes one of the end boundary candidates as the boundary of "A" segment, and one of the start boundary candidates as the boundary of "B" segment, which results in the wrong segmentation. Actually both that start and end candidate should be for the "A" segment. Such phenomena indicates that, although the boundary information can be integrated into the repetition-based approaches, it is still quite challenging for the algorithms to decide the concrete length of segments. Finally, we present the result for the most strict integration strategy that restrict both the start and end boundaries of segments in Figure A.2e. We see that all segments are estimated correctly. Different from the previous ones, using this strategy, the structure analysis approach cannot take other time points as boundaries, thus the over-segmentation problem is avoided. What still might happen is the under-segmentation problem, where a segment choose two boundaries which are not neighbor boundaries, and form a longer, higher-hierarchical segment.

A.1.2 Evaluation of Different Integration Strategies

After discussing the different strategies by means of the example, in this section we test the general performance of these strategies by a quantitative evaluation. By doing so, we check the general behavior of these strategies tested on various songs.

As a pilot experiment, we first select 36 Beatles songs to form a small dataset, where each song is checked that it does contain repetitive sections in the ground truth structure annotation. Then we conduct our experiments testing the different boundary integration strategies as mentioned above. The important parameter settings for both the iterative approach and the joint approach are kept as the same as the last chapter (see Section 7.4.2). Finally, the structure segmentation result, which yielded by both approaches, are evaluated using the same evaluation measure as in the last chapter. In other words, we still present the evaluation numbers by means of the pairwise frame clustering and the boundary retrieval hit rate.

We first use the pairwise frame clustering evaluation to test the performance of both the iterative approach and the joint approach with boundary information integration. Table A.1 shows the evaluation result. We first present the evaluation results of the both approaches without boundary constraints consider the results as the baselines. With no hint of segment boundaries, the iterative approach gets an F-measure of 0.70 whereas the joint approach gets 0.67. Next, using the first strategy which restrict the start boundaries of the computed segments, both of the the approaches show significant improvements: the iterative approach gets 0.74 and the joint approach gets 0.79. Such improvements indicate that the boundary information helps the repetition-based approaches to decide the position of segments, thus to some extent avoid the segment shifting problem. This is especially true for the joint approach, which gets an F-measure increase of 0.12 compared to the result derived from the baseline. Next, using the second strategy which restrict the end boundaries of segments, both of the approaches get higher result compared to the baseline. However, the joint approach does not improve so significant as using the first strategy. It gets an F-measure of 0.75 using this strategy whereas 0.79 using the first strategy. Moving on to the fourth column in the table, where we use the third strategy to restrict either start or end boundaries of segments, the performance of the two approaches decrease slightly. This is mainly because offering more boundary candidates also bring difficulty to the algorithms to decide position and length of segments. Using this strategy, the algorithms may yield more incorrect segments comparing to the previous two strategies. Finally, in the last column we apply the strictest strategy, which is to restrict both the start and the end boundaries to exactly specify where the candidate segments could be. We reach the highest performance: the iterative approach gets 0.76 and the joint approach gets 0.82. Note that this is the theoretical upper limit performance of the repetition-based approaches apllied on this dataset. It also means that roughly 80% correctness seems to be the highest repetition-based structure analysis result using Beatles dataset.

As a second evaluation, we use the boundary retrieval evaluation measure to check the structure analysis performance of these strategies. Table A.2 shows the result. Using purely repetition-based structure analysis, the iterative and the joint approach get only 0.59 and 0.60 in F-measure, respectively. These two numbers again serve as baseline results for the comparison with the next four strategies. Next, by giving constraints to

F pair	baseline	start only	end only	start or end	start and end
Iterative	0.70	0.74	0.74	0.73	0.76
Joint	0.67	0.79	0.75	0.72	0.82

Table A.1: Results of repetition-based structure analysis under different constraint strategies of ground truth boundaries. Here we use the pairwise frame P/R/F evaluation measure as in the last chapter.

F bound	baseline	start only	end only	start or end	start and end
Iterative	0.59	0.70	0.72	0.68	0.80
Joint	0.60	0.72	0.73	0.69	0.85

Table A.2: Results of repetition-based structure analysis under different constraint strategies of ground truth boundaries. Here we use the boundary retrieval P/R/F evaluation measure as in the last chapter.

the start boundaries of segments, we get 0.70 for the iterative approach and 0.72 for the joint approach. Both of them are significantly higher than the baseline results. This also again support the conclusion from the previous pairwise frame evaluation that restricting start boundaries can greatly improve the structure analysis performance. After that, using segment end boundaries as constraints, we get 0.72 for the iterative approach and 0.73 for the joint approach. These results show that the second strategy yields a similar improvement on the approches as the first strategy. Moving to the fourth column, which restrict either start or end boundaries, again the result decrease to only 0.68 for the iterative approach and 0.69 for the joint approach. Such deterioration also happens in the pairwise evaluation for this strategy. Therefore, we conclude that this strategy yields the worst performance among the four different boundary integration strategies. In the end, specifying the start and end boundaries of segments, we get a highest result of 0.80 for the iterative approach and 0.85 for the joint approach. This also gives us the upper limit of the repetition-based approaches for boundary evaluation. Since we only extract repetitive segments, the boundaries of those non-repetitive segments can not be estimated, which is the reason why we cannot get the perfect 1.00 F-measure.

Bibliography

[1] Samer Abdallah, Katy Nolad, Mark Sandler, Michael A. Casey, and Christopher Rhodes. Theory and evaluation of a Bayesian music structure extractor. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 420–425, London, GB, 2005.

[2] Samer Abdallah, Mark Sandler, Christopher Rhodes, and Michael A. Casey. Using duration models to reduce fragmentation in audio segmentation. *Machine Learning*, 65(2–3):485–515, 2006.

[3] Eric Allamanche, Jürgen Herre, Oliver Hellmuth, Bernhard Fröba, Thorsten Kastner, and Markus Cremer. Content-based identification of audio material using MPEG-7 low level description. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2001.

[4] Andreas Arzt, Sebastian Böck, and Gerhard Widmer. Fast identification of piece and score position via symbolic fingerprinting. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, Porto, Portugal, 2012.

[5] Jean-Julien Aucouturier and Mark Sandler. Segmentation of musical signals using hidden Markov models. In *Proceedings of the 110th AES Convention*, Amsterdam, NL, 2001.

[6] Jean-Julien Aucouturier and Mark Sandler. Finding repeating patterns in acoustic musical signals: Applications for audio thumbnailing. In *Proceedings of the Audio Engineering Society 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, pages 412–421, Espoo, Finland, 2002.

[7] Luke Barrington, Antoni B. Chan, and Gert Lanckriet. Modeling music as a dynamic texture. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):602–612, 2010.

[8] Mathieu Barthet, György Fazekas, and Mark Sandler. Multidisciplinary perspectives on music emotion recognition: Recommendations for content-and context-based models. In *Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval (CMMR)*, London, UK, 2012.

[9] Mathieu Barthet, David Marston, Chris Baume, György Fazekas, and Mark Sandler. Design and evaluation of semantic mood models for music recommendation. In

Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR), Curitiba, Brazil, 2013.

[10] Mark A. Bartsch and Gregory H. Wakefield. To catch a chorus: Using chroma-based representations for audio thumbnailing. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 15–18, New Paltz, NY, USA, 2001.

[11] 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.

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

[13] Bruce Benward and Marilyn Saker. *Music in Theory and Practice*. McGraw-Hill Humanities/Social Sciences/Languages, 8th edition, 2008.

[14] Tony Bergstrom, Karrie Karahalios, and John C. Hart. Isochords: Visualizing structure in music. In *Proceedings of Graphics Interface (GI)*, New York, USA, 2007.

[15] Ingwer Borg and Patrick J. F. Groenen. *Modern Multidimensional Scaling Theory and Applications*. Springer, 2005.

[16] Pedro Cano, Eloi Batlle, Emilia Gómez, Leandro de C. T. Gomes, and Madeleine Bonnet. Audio fingerprinting: Concepts and applications. In Saman K. Halgamuge and Lipo Wang, editors, *Computational Intelligence for Modelling and Prediction*, volume 2 of *Studies in Computational Intelligence*, pages 233–245. Springer, 2005.

[17] Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma. A review of audio fingerprinting. *J. VLSI Signal Process. Syst.*, 41(3):271–284, November 2005.

[18] 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.

[19] 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.

[20] Wei Chai and Barry Vercoe. Semantic segmentation and summarization of music. *IEEE Signal Processing Magazine, Special Issue on Semantic Retrieval of Multimedia*, 23(2), 2006.

[21] Wing-Yi Chan. A report on musical structure visualization. http://www.cse.ust.hk/~wallacem/winchan/research.html, 2010.

[22] Wing-Yi Chan, Huamin Qu, and Wai-Ho Mak. Visualizing the semantic structure in classical music works. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):161–173, Jan.-Feb. 2010.

[23] Ruofeng Chen and Ming Li. Music structural segmentation by combining harmonic and timbral information. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Miami, USA, 2011.

[24] Shih-Chuan Chiu, Man-Kwan Shan, Jiun-Long Huang, and Hua-Fu Li. Mining polyphonic repeating patterns from music data using bit-string based approaches. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2009.

[25] Tom Collins, Robin Laney, Alistair Willis, and Paul H Garthwaite. Modeling pattern importance in chopin's mazurkas. *Music Perception*, 2011.

[26] Darrell Conklin. Discovery of distinctive patterns in music. *Intell. Data Anal.*, 14(5):547–554, 2010.

[27] Darrell Conklin. Distinctive patterns in the first movement of brahms' string quartet in c minor. *Journal of Mathematics and Music*, 4(2):85–92, 2010.

[28] Matthew Cooper and Jonathan Foote. Automatic music summarization via similarity analysis. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 81–85, Paris, France, 2002.

[29] Matthew Cooper and Jonathan Foote. Summarizing popular music via structural similarity analysis. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 127–130, New Paltz, NY, USA, 2003.

[30] Roger B. Dannenberg and Masataka Goto. Music structure analysis from acoustic signals. In David Havelock, Sonoko Kuwano, and Michael Vorländer, editors, *Handbook of Signal Processing in Acoustics*, volume 1, pages 305–331. Springer, New York, NY, USA, 2008.

[31] Roger B. Dannenberg and Ning Hu. Pattern discovery techniques for music audio. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 63–70, Paris, France, 2002.

[32] 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.

[33] J.Stephen Downie, AndreasF. Ehmann, Mert Bay, and M.Cameron Jones. The music information retrieval evaluation exchange: Some observations and insights. *Advances in Music Information Retrieval*, 2010.

[34] Jean-Pierre Eckmann, S. Oliffson Kamphorst, and David Ruelle. Recurrence plots of dynamical systems. *Europhysics Letters*, 4(9):973–977, 1987.

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

[36] 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.

[37] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 452–455, New York, NY, USA, 2000.

[38] Sheng Gao, Namunu Chinthaka Maddage, and Chin-Hui Lee. A hidden Markov model based approach to music segmentation and identification. In *Proceedings of the 4th Pacific Rim Conference on Multimedia (PCM)*, pages 1576–1580, Singapore, 2003.

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

[40] Emilia Gómez and Jordi Bonada. Tonality visualization of polyphonic audio. In *Proceedings of the International Computer Music Conference (ICMC)*, Barcelona, Spain, 2005.

[41] Emilia Gómez and Perfecto Herrera. Estimating the tonality of polyphonic audio files: Cognitive versus machine learning modelling strategies. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004.

[42] Michael M. Goodwin and Jean Laroche. A dynamic programming approach to audio segmentation and music / speech discrimination. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 309–312, Montreal, QC, Canada, 2004.

[43] 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.

[44] Masataka Goto. Aist annotation for the rwc music database. In *roceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 359–360, 2006.

[45] 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.

[46] 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.

[47] Masataka Goto, Jun Ogata, Kazuyoshi Yoshii, Hiromasa Fujihara, Matthias Mauch, and Tomoyasu Nakano. Podcastle and songle: Crowdsourcing-based web services for retrieval and browsing of speech and music content. In *Proceedings of the 2012 ACM Workshop on Crowdsourcing for Multimedia (CrowdMM)*, Lyon, France, 2012.

[48] Harald Grohganz, Michael Clausen, Nanzhu Jiang, and Meinard Müller. Converting path structures into block structures using eigenvalue decompositions of self-similarity matrices. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, pages 209–214, Curitiba, Brazil, 2013.

[49] Peter Grosche. *Signal Processing Methods for Beat Tracking, Music Segmentation, and Audio Retrieval*. PhD thesis, Saarland University and MPI Informatik, 2012.

[50] Allan Hanbury. Constructing cylindrical coordinate colour spaces. *Pattern Recognition Letters*, 29:494–500, 2008.

[51] Jia-Lien Hsu, Chih-Chin Liu, and Arbee LP Chen. Discovering nontrivial repeating patterns in music data. *Multimedia, IEEE Transactions on*, 3(3):311–325, 2001.

[52] 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.

[53] Kristoffer Jensen. A causal rhythm grouping. In *Computer Music Modeling and Retrieval*, volume 3310 of *Lecture Notes in Computer Science*, pages 83–95. Springer Berlin / Heidelberg, 2004.

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

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

[56] Nanzhu Jiang and Meinard Müller. Automated methods for analyzing music recordings in sonata form. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, pages 595–600, Curitiba, Brazil, 2013.

[57] Nanzhu Jiang and Meinard Müller. Towards efficient audio thumbnailing. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, 2014.

[58] Nanzhu Jiang and Meinard Müller. Estimating double thumbnails for music recordings. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015.

[59] Florian Kaiser and Geoffroy Peeters. Multiple hypotheses at multiple scales for audio novelty computation within music. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[60] Florian Kaiser and Geoffroy Peeters. A simple fusion method of state and sequence segmentation for music structure discovery. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, Curitiba, Brazil, 2013.

[61] Florian Kaiser and Thomas Sikora. Music structure discovery in popular music using non-negative matrix factorization. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 429–434, Utrecht, The Netherlands, 2010.

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

[63] Ian Knopke and Frauke Jürgensen. A system for identifying common melodic phrases in the masses of palestrina. *Journal of New Music Research*, 38(2):171–181, 2009.

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

[65] Paul Lamere. http://musicmachinery.com/2012/11/19/visualizing-the-structure-of-pop-music/, Retrieved 04.06.2014.

[66] Olivier Lartillot. Efficient extraction of closed motivic patterns in multi-dimensional symbolic representations of music. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 229–235. IEEE, 2005.

[67] Olivier Lartillot and Petri Toiviainen. Motivic matching strategies for automated pattern extraction. *Musicae Scientiae*, 11(1 suppl):281–314, 2007.

[68] H. Sebastian Lee, Daniel D. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[69] Hugo Leichtentritt. *Musikalische Formenlehre*. Breitkopf und Härtel, 12. Auflage, Wiesbaden, Germany, 1987.

[70] Mark Levy, Katy Noland, and Mark Sandler. A comparison of timbral and harmonic music segmentation algorithms. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1433–1436, Honolulu, Hawaii, USA, 2007.

[71] 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.

[72] 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.

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

[74] Beth Logan. Contentbased playlist generation: Exploratory experiments. Technical report, HewlettPackard Labs One Cambridge Center, 2002.

[75] Lie Lu, Muyuan Wang, and Hong-Jiang Zhang. Repeating pattern discovery and structure analysis from acoustic music data. In *Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval*, New York, NY, USA, 2004.

[76] 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.

[77] Namunu C. Maddage. Automatic structure detection for popular music. *IEEE Multimedia*, 13(1):65–77, 2006.

[78] Namunu Chinthaka Maddage, Mohan Kankanhalli, and Haizhou Li. Effectiveness of signal segmentation for music content representation. In Shinichi Satoh, Frank Nack, and Minoru Etoh, editors, *Advances in Multimedia Modeling*, volume 4903 of *Lecture Notes in Computer Science*, pages 477–486. Springer Berlin / Heidelberg, 2008.

[79] Matija Marolt. A mid-level melody-based representation for calculating audio similarity. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 280–285, Victoria, Canada, 2006.

[80] Norbert Marwan, M. Carmen Romano, Marco Thiel, and Jürgen Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 438(5-6):237–329, 2007.

[81] Matthias Mauch. *Automatic Chord Transcription from Audio Using Computational Models of Musical Context*. PhD thesis, Queen Mary University of London, 2010.

[82] 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.

[83] Brian McFee and MDaniel P.W. Ellis. Analyzing song structure with spectral clustering. In *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014.

[84] Brian McFee and Gert Lanckriet. The natural language of playlists. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, Miami, USA, 2011.

[85] David Meredith. Point-set algorithms for pattern discovery and pattern matching in music. *Content-Based Retrieval*, (06171), 2006.

[86] Richard Middleton. Form. In Bruce Horner and Thomas Swiss, editors, *Key terms in popular music and culture*, pages 141–155. Wiley-Blackwell, 1999.

[87] Riccardo Miotto and Nicola Orio. A music identification systems based on chroma indexing and statistical modeling. In *ISMIR*, pages 301–306, 2008.

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

[89] 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.

[90] 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.

[91] 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.

[92] Meinard Müller and Nanzhu Jiang. A scape plot representation for visualizing repetitive structures of music recordings. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, pages 97–102, Porto, Portugal, 2012.

[93] Meinard Müller, Nanzhu Jiang, and Harald Grohganz. SM Toolbox: MATLAB implementations for computing and enhancing similiarty matrices. In *Proceedings of the AES Conference on Semantic Audio*, London, GB, 2014.

[94] Meinard Müller, Nanzhu Jiang, Harald Grohganz, and Michael Clausen. Strukturanalyse für Musiksignale. In *Proceedings of 43th GI Jahrestagung*, pages 2943–2957, Koblenz, Germany, 2013.

[95] Meinard Müller, Nanzhu Jiang, and Peter Grosche. A robust fitness measure for capturing repetitions in music recordings with applications to audio thumbnailing. *IEEE Transactions on Audio, Speech & Language Processing*, 21(3):531–543, 2013.

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

[97] 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.

[98] Meinard Müller and Frank Kurth. Towards structural analysis of audio recordings in the presence of musical variations. *EURASIP Journal on Advances in Signal Processing*, 2007(1), 2007.

[99] Meinard Müller, Henning Mattes, and Frank Kurth. An efficient multiscale approach to audio synchronization. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 192–197, Victoria, Canada, 2006.

[100] Oriol Nieto and Juan Pablo Bello. Music segment similarity using 2d-fourier magnitude coefficients. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[101] Oriol Nieto and Morwaread M. Farbood. Dentifying polyphonic patterns from audio recordings using music segmentation techniques. In *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014.

[102] Oriol Nieto, Eric J. Humphrey, and Juan Pablo Bello. Compressing music recordings into audio summaries. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, 2012.

[103] Oriol Nieto and Tristan Jehan. Convex non-negative matrix factorization for automatic music structure identification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[104] Bee Suan Ong, Emilia Gómez, and Sebastian Streich. Automatic extraction of musical structure using pitch class distribution features. In *In Proceedings of the Workshop on Learning the Semantics of Audio Signals (LSAS)*, pages 53–56, 2006.

[105] Jouni Paulus. *Signal Processing Methods for Drum Transcription and Music Structure Analysis*. PhD thesis, Tampere University of Technology, Tempere, Finland, 2009.

[106] Jouni Paulus and Anssi P. Klapuri. Music structure analysis by finding repeated parts. In *Proceedings of the 1st ACM Audio and Music Computing Multimedia Workshop*, pages 59–68, Santa Barbara, CA, USA, 2006.

[107] 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.

[108] 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.

[109] 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.

[110] Geoffroy Peeters. Deriving musical structure from signal analysis for music audio summary generation: "sequence" and "state" approach. In *Computer Music Modeling and Retrieval*, volume 2771 of *Lecture Notes in Computer Science*, pages 143–166. Springer Berlin / Heidelberg, 2004.

[111] Geoffroy Peeters. Music structure dicovery: Measurig the "state-ness" of times. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR): Late Breaking session*, 2011.

[112] Geoffroy Peeters, Amaury La Burthe, and Xavier Rodet. Toward automatic music audio summary generation from signal analysis. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Paris, France, 2002.

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

[114] Christopher Raphael. Automatic segmentation of acoustic musical signals using hidden markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:360–370, 1998.

[115] Christophe Rhodes and Michael A. Casey. Algorithms for determining and labelling approximate hierarchical self-similarity. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 41–46, Vienna, Austria, 2007.

[116] Craig Stuart Sapp. Harmonic visualizations of tonal music. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 423–430, La Habana, Cuba, 2001.

[117] Craig Stuart Sapp. Visual hierarchical key analysis. *ACM Computers in Entertainment*, 3(4):1–19, 2005.

[118] Michael Schoeffler, Fabian-Robert Stöter, Harald Bayerlein, Bernd Edler, and Jürgen Herre. An Experiment about Estimating the Number of Instruments in Polyphonic Music: A Comparison Between Internet and Laboratory Results. In *Proceedings of 14th International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, Brazil, 2013.

[119] 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.

[120] Joan Serrà, Meinard Müller, Peter Grosche, and Josep Ll. Arcos. Unsupervised music structure annotation by time series structure features and segment similarity. *IEEE Transactions on Multimedia*, 16(5):1229–1240, 2014.

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

[122] Xi Shao, Namunu C. Maddage, Changsheng Xu, and Mohan S. Kankanhalli. Automatic music summarization based on music structure analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Pennsylvania, USA, 2005.

[123] Yu Shiu, Hong Jeong, and C-C Jay Kuo. Similar segment detection for music structure analysis via viterbi algorithm. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 789–792, July 2006.

[124] SM Toolbox. http://www.audiolabs-erlangen.de/resources/MIR/SMtoolbox/, Retrieved 30.08.2014.

[125] J.B.L. Smith, Ching-Hua Chuan, and E. Chew. Audio properties of perceived boundaries in music. *IEEE Transactions on Multimedia*, 16(5):1219–1228, Aug 2014.

[126] Jordan B. L. Smith and Elaine Chew. A meta-analysis of the mirex structure segmentation task. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, Curitiba, Brazil, 2013.

[127] Jordan B. L. Smith and Elaine Chew. Using quadratic programming to estimate feature relevance in structural analyses of music. In *Proceedings of the ACM International Conference on Multimedia*, pages 113–122, 2013.

[128] Jordan Bennett Louis Smith, John Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J. Stephen Downie. Design and creation of a large-scale database of structural annotations. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 555–560, Miami, FL, USA, 2011.

[129] Fabian-Robert Stöter, Michael Schoeffler, Bernd Edler, and Jürgen Herre. Human ability of counting the number of instruments in polyphonic music. In *Proceedings of Meetings on Acoustics Vol. 19*, Montreal, Canada, 2013.

[130] Balaji Thoshkahna, Meinard Müller, Venkatesh Kulkarni, and Nanzhu Jiang. Novel audio features for capturing tempo salience in music recordings. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015.

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

[132] Douglas Turnbull, Gert Lanckriet, Elias Pampalk, and Masataka Goto. A supervised approach for detecting boundaries in music using difference features and boosting. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 51–54, Vienna, Austria, 2007.

[133] George Tzanetakis and Perry Cook. Multifeature audio segmentation for browsing and annotation. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 103–106, New Platz, NY, USA, 1999.

[134] Rob van Gulik, Fabio Vignoli, and Huub van de Wetering. Mapping music in the palm of your hand, explore and discover your collection. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, 2004.

[135] Avery Wang. The Shazam music recognition service. *Communications of the ACM*, 49(8):44–48, 2006.

[136] Martin Wattenberg. Arc diagrams: visualizing structure in strings. *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 110–116, 2002.

[137] R. J. Weiss and J. P. Bello. Unsupervised discovery of temporal structure in music. *IEEE Journal of Selected Topics in Signal Processing*, 5:1240–1251, 2011.

[138] Ho-Hsiang Wu and Juan P. Bello. Audio-based music visualization for music structure analysis. In *Proceedings of Sound and Music Computing Conference (SMC)*, Barcelona, Spain, 2010.