Algorithmen zur strukturellen Analyse von Musikaufnahmen

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Harald G. Grohganz

aus

Frankfurt am Main

Bonn, Dezember 2014

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter:	Prof. Dr. Michael Clausen
2. Gutachter:	Prof. Dr. Meinard Müller
Tag der Promotion:	26. Februar 2015
Erscheinungsjahr:	2015

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Verzeichnis der Codebeispiele

1. Einleitung

Musik als eine Form der Kunst stellt eine komplexe Variante zwischenmenschlicher Interaktion und Kommunikation dar, bei der es oftmals schwierig ist, objektive Aussagen zu treffen. Manche musikalische Aspekte sind jedoch bis zu einem gewissen Grad beschreibbar, sodass wir beispielsweise einige Aussagen zu Harmonik und Rhythmik sowie zu klar erkennbaren Wiederholungsstrukturen treffen können – wobei wir uns bewusst sind, dass wir oftmals nur den »Regelfall« beschreiben. In dieser Arbeit werden wir uns stets darauf konzentrieren, Musik nach objektiven Gesichtspunkten zu *beschreiben* ohne ihren Inhalt zu *interpretieren*.

Das verhältnismäßig junge Forschungsgebiet des *Music Information Retrieval* (MIR) vereinigt ein weites Feld von Problemstellungen, Anwendungsgebieten und technischen Herangehensweisen. Die möglichen Themengebiete umfassen unter anderem: Identifikation und Klassifikation (etwa nach Stilrichtungen oder Stimmungen), Erkennung von Cover-Songs, Extraktion musikalischer Informationen aus Audioaufnahmen sowie Konzeption und Umsetzung von Benutzerschnittstellen, etwa zum Navigieren in großen Musikdatenbanken. Eine spezielle Problemstellung im MIR ist die strukturelle Segmentierung (engl. *structural segmentation*), bei der ein Musikstück nach semantischen Gesichtspunkten in zeitliche Abschnitte unterteilt wird, deren Benennung die Gemeinsamkeiten und Gegensätze einzelner Passagen verdeutlichen soll.

In diesem Kontext stellt sich bei der automatischen Analyse musikalischer Strukturen zunächst die Frage, was durch den Begriff des strukturellen Segments ausgedrückt werden soll. Nun weist Musik in ihrem Aufbau oftmals eine hierarchische Struktur auf, die bei einzelnen Noten oder Taktschlägen beginnt, und sich über Takte, Motive und Themen bis hin zur Grobunterteilung eines Stückes in wiederkehrende und neue Passagen erstreckt. Ein strukturelles Segment ist daher immer auf einer festen Hierarchiestufe angesiedelt, und die Bestimmung einer semantisch relevanten Hierarchieebene stellt ein wesentliches Problem dar. Dies wird durch die der Musik immanenten zeitlichen Komponente verursacht, die bei frühen Formen der Musiknotation von der Antike über die mittelalterlichen Neumen und Quadratnotationen bis hin zur »klassischen« Partitur ausschließlich in musikalisch-symbolischer Form vorliegt und entweder an die Geschwindigkeit der Sprache oder oftmals auch an Tanzbewegungen angepasst ist [170]. Bei einer solchen Repräsentation können musikalische Strukturinformationen wie etwa Motive verhältnismäßig leicht erkannt werden [89].

Seit der Erfindung des Phonographen durch Edison 1877 [71] ist es möglich, Schallereignisse direkt aufzuzeichnen und wiederzugeben, wodurch die musikalische Zeitunterteilung eine direkte *physikalische* Komponente erhält. Auch das Aufkommen von elektronischen

Instrumenten, welche beispielsweise die Abfolge der gedrückten Tasten speichern und wiedergeben können, ermöglichte eine weitere Art der Musikaufnahme, bei der zwar die Tonhöhen- und Lautstärkeinformationen symbolisch, die Zeitinformationen jedoch nur in physikalischer Form vorliegen. Die strukturelle Analyse dieser beiden Aufnahmetypen stellt somit eine spezielle Form der Rekonstruktion von musikalischen Zeitinformationen dar, die sowohl struktureller Natur (wie etwa Einsatzzeiten von Phrasen und Themen) als auch rhythmischer Natur (wie etwa Positionen von Schlägen und Takten) sein können.

Bei der strukturellen Segmentierung von Musikaufnahmen bewegen wir uns in einem Übergangsbereich zwischen Informatik und Musikwissenschaft, bei dem die Ansätze des *Information Retrieval* an ihre Grenzen stoßen. So ist eine musikwissenschaftliche Formanalyse in vielen Aspekten als eine interpretatorische Aufgabe anzusehen, bei der zwischen mehreren gleichermaßen als »richtig« anzusehenden Betrachtungsweisen abgewogen werden muss. Allein schon der Begriff der *Ähnlichkeit* zweier musikalischer Passagen ist nicht klar definiert und Gegenstand aktueller Forschung. Auf symbolischen Daten (Melodieverläufen) wurden beispielsweise bereits 1982 in [187] allein sieben Ähnlichkeitsmaße zur automatischen Klassifikation von Melodien vorgestellt. In neuerer Zeit werden verstärkt Segmentierungsprobleme auf Audiosignalen betrachtet [28, 147, 153], bei welchen die Noteninformationen nicht explizit vorliegen und die zusätzliche physikalische Komponenten wie Rauschen beinhalten, was für die Segmentierung eine zusätzliche Herausforderung darstellt. Auch die Bewertung der berechneten Ergebnisse ist eine nichttriviale Problemstellung. Durch den Vergleich mit manuell erstellten Referenz-Segmentierungen können zwar automatische Evaluationen durchgeführt werden, diese weisen allerdings eine Vielzahl methodischer und konzeptioneller Schwächen auf.

In dieser Arbeit beschäftigen wir uns schwerpunktmäßig mit der strukturellen Analyse[1] von Musikaufnahmen durch Ansätze, bei denen der Blick auf das gesamte Stück im Vordergrund steht. Hierunter fällt etwa die Segmentierung eines Musikstückes nach Wiederholungen und Homogenitätsbereichen, d. h. Passagen, bei denen ein oder mehrere musikalische Aspekte wie Tonart, Rhythmik oder Klangfarbe unverändert bleiben. Ein anderer dieser global wirkenden Ansätze behandelt die Bestimmung der zeitlichen Positionen von Schlagzeiten und Takten. Hierbei präsentieren wir einerseits methodische und algorithmische Beiträge wie die Umwandlung von Wiederholungen in Homogenitätsbereiche, andererseits neuartige Visualisierungen etwa des harmonischen Aufbaus eines Musikstückes sowie interaktive Benutzerschnittstellen. Die vorgestellten Verfahren wurden teils in MATLAB, teils in Java implementiert und anhand mehrerer Datensätze systematisch sowohl qualitativ als auch quantitativ ausgewertet. Mittels kritischer Diskussion sowohl der Methodik als auch der Ergebnisse leisten wir einen Beitrag zur differenzierten Betrachtung und Hinterfragung bestehender Systeme und Arbeitsweisen in der automatischen Musikstrukturanalyse. In einer Fallstudie verbinden wir die MIR-Ansätze

[1] Die strukturelle Analyse von Musikdaten stellt ein aktives Forschungsgebiet dar. In der MIR-Gemeinschaft entstehen parallel zu dieser Arbeit drei weitere Dissertationen von Nanzhu Jiang (Erlangen), Jordan Smith (London) und Oriol Nieto (New York), deren Schwerpunkt ebenfalls auf der Musik-Strukturanalyse liegt.

zur strukturellen Segmentierung mit den Erkenntnissen der Musiktheorie und tragen damit zu weitergehenden Forschungen im Bereich einer *musikalisch fundierten* Segmentierung bei.

Die vorgestellten Techniken und Algorithmen können eine Rolle bei einer Vielzahl von Anwendungen spielen. Hierzu zählen etwa Navigation und Interaktion in Musikdatenbanken, aber auch die gezielte Suche in großen Musikdatenbeständen, bei denen durch Verlinkung und Synchronisation verschiedenartiger Medien den Benutzern ein breit gefächerter Zugang zur Musik ermöglicht wird [26, 50]. Weiterhin stellt eine solche Segmentierung immer eine *Komplexitätsreduktion* dar, die beispielsweise im Rahmen einer vielschichtigen Darstellung von Musikstücken wie in [60] zur groben Orientierung verwendet wird. Als Teil der Signalanalyse und -verarbeitung kann eine semantisch relevante strukturelle Segmentierung auch ihren Beitrag zur Verbesserung von musikalischen Merkmalen leisten. Hierunter fallen beispielsweise adaptive Merkmalsmodelle, bei denen die Parameter zur Merkmalsberechnung auf verschiedenen Segmentklassen unterschiedlich gewählt werden können. Nicht zuletzt stehen die in der Audiosignalverarbeitung wie auch im *Music Information Retrieval* entwickelten Ansätze in gegenseitiger Wechselwirkung mit den Lösungen ähnlich gelagerte Probleme in angrenzenden Disziplinen, wie etwa bei der Bild- und Videoverarbeitung, in der Grundlagenforschung für autonome Systeme oder in der Bioinformatik.

1.1. Aufbau dieser Arbeit

Der Hauptteil der Arbeit ist in vier Kapitel unterteilt. In den ersten drei Kapiteln werden Problemstellungen aus dem Bereich der strukturellen Segmentierung eines Audiosignals thematisiert, wohingegen im letzten Kapitel die Schätzung rhythmischer Informationen einer synthetischen Musikaufnahme im Vordergrund steht.

In Kapitel 2 werden die Grundlagen der automatischen Musikstrukturanalyse vorgestellt. Nach einem Überblick über musikalische Aspekte und die daraus abgeleiteten technischen Merkmale folgt die Erläuterung der Selbstähnlichkeitsmatrix als eine vielseitige Möglichkeit zur Darstellung struktureller Informationen. Dieser Teil basiert auf [123]. Anschließend wird eine mathematische Modellierung der beiden Segmentierungsprinzipien Wiederholung und Homogenität entwickelt sowie ein gängiges Verfahren zur homogenitätsbasierten Segmentierung vorgestellt. Abschließend wird eine häufig zur Evaluation von Strukturanalyseverfahren verwendete Methode beschrieben und sowohl konzeptionell als auch anhand eines großen Datensatzes kritisch erörtert.

In Kapitel 3 wird ein neuartiges Verfahren zur Konvertierung von wiederholungsbasierten in homogenitätsbasierte Selbstähnlichkeitsmatrizen entwickelt. Nach einer Beschreibung des Algorithmus wird der theoretische Hintergrund erarbeitet und es werden einige auftretende Effekte analysiert und diskutiert. In einer qualitativen Analyse werden mehrere Beispiele ausführlich besprochen und die Möglichkeiten und Grenzen unserer Methode aufgezeigt. Weiterhin wird eine quantitative Evaluation auf mehreren Datensätzen beschrieben, in der un-

sere Ergebnisse mit anderen Verfahren verglichen werden. Das Kapitel stellt eine Fortführung von [62] dar.

In der in Kapitel 4 durchgeführten Fallstudie zur automatischen Musikstrukturanalyse wird Schuberts »Winterreise« als ein neuer Datensatz eingeführt sowie detaillierte Hintergrundinformationen zu den eigens angefertigten Referenz-Annotationen gegeben. Mit der Vorstellung von neuen Modifikationen von Tonarten- und Klangfarbenmerkmalen werden zwei für diese Fallstudie relevante musikalische Aspekte genauer untersucht sowie eine neuartige Visualisierung für harmonische Beziehungen eingeführt. Zum Abschluss wird in einem Ausblick eine mögliche verallgemeinerte strukturelle Segmentierung nach mehreren Aspekten umrissen und eine interaktive Online-Benutzeroberfläche präsentiert.

In Kapitel 5 wird ein neuartiges Verfahren zur nachträglichen Korrektur des Ergebnisses von Verfahren zur automatischen Rhythmus-Transkription von Musikaufnahmen mit sowohl symbolischen als auch physikalischen Komponenten beschrieben. Hierzu wird zuerst detailliert die Repräsentation rhythmischer Informationen im verbreiteten MIDI-Format erläutert. Anschließend beschreiben wir den Algorithmus des vorgestellten Verfahrens ausführlich, stellen einige Erweiterungen sowie die für dieses Verfahren implementierte Benutzeroberfläche vor und schildern ein praktisches Anwendungsbeispiel. Das Kapitel ist eine erweiterte Version von [63].

1.2. Beiträge

Insgesamt können die Hauptbeiträge dieser Arbeit wie folgt zusammengefasst werden:

- Mathematische Formulierung der Segmentierungsprinzipien Homogenität und Wiederholung sowie kritische Diskussion zur automatischen Evaluation der Ausgabe von maschinellen Verfahren zur Musikstrukturanalyse.
- Vorstellung einer neuartigen Methode zum Umwandeln der typischen Pfadmuster einer Selbstähnlichkeitsmatrix von Wiederholungsstrukturen in Homogenitätsbereiche, welche anschließend mittels sparse-NMF-Klassifizierung zur Strukturanalyse verwendet werden können.
- Erweiterung bestehender Visualisierungen von musikalischen Merkmalen und Selbstähnlichkeitsmatrizen sowie Entwicklung musiktheoretisch-orientierter Visualisierungen harmonischer Informationen.
- Erstellung einer interaktiven Anreicherung von graphischen Merkmalsdarstellungen um eine Möglichkeit zum synchronen Abspielen der zugrundeliegenden Audioinformationen für MATLAB und Internetseiten.
- Kommentierte Annotation von Schuberts »Winterreise« nach verschiedenen musikalischen Gesichtspunkten, Aufbereitung als Datensatz zur qualitativen Strukturanalyse sowie Bereitstellung auf einer interaktiven Website.

- Einige Modifikationen von Klangfarben- und Tonartenmerkmalen zur Erhöhung der Deskriptivität bei der automatischen Musikstrukturanalyse.
- Entwurf einer neuartigen Methode zur nachträglichen Korrektur der Ergebnisse von Verfahren zur automatischen Rhythmuserkennung sowie Implementierung eines Java-Programms zur automatischen semantischen Anreicherung von Musikaufnahmen im MIDI-Format.

1.3. Verwandte Publikationen

Einige Teile dieser Arbeit wurden bereits bei verschiedenen Konferenzen veröffentlicht. Im Folgenden sind die Publikationen des Autors mit Bezug zu den Inhalten dieser Arbeit in chronologischer Reihenfolge aufgeführt:

[123] Meinard Müller, Nanzhu Jiang, Harald Grohganz, and Michael Clausen. Strukturanalyse für Musiksignale. In *GI-Edition: Lecture Notes in Informatics*, pages 2943–2957, Koblenz, Germany, 2013.

[62] 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.

[122] 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.

[63] Harald Grohganz, Michael Clausen, and Meinard Müller. Estimating musical time information from performed MIDI files. In *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR)*, pages 35–40, Taipei, Taiwan, 2014.

Auf die folgenden beiden Publikationen wird im Laufe dieser Arbeit nicht näher eingegangen:

[27] David Damm, Harald Grohganz, Frank Kurth, Sebastian Ewert, and Michael Clausen. SyncTS: Automatic synchronization of speech and text documents. In *Proceedings of the AES International Conference Semantic Audio*, pages 98–107, Ilmenau, Germany, 2011.

[40] Jonathan Driedger, Harald Grohganz, Thomas Prätzlich, Sebastian Ewert, and Meinard Müller. Score-informed audio decomposition and applications. In *Proceedings of the ACM International Conference on Multimedia (ACM-MM)*, pages 541–544, Barcelona, Spain, 2013.

Danksagung

Diese Arbeit ist während meiner Tätigkeit am Institut für Informatik III der Rheinischen Friedrich-Wilhelms-Universität Bonn entstanden. Finanziert wurde meine Forschung weitgehend über das DFG-Projekt METRUM[2] (DFG CL 64/8-1, DFG MU 2686/5-1), das Nanzhu Jiang und mir die weitgehend unabhängige Anfertigung unserer Dissertationen zur strukturellen Segmentierung von Musiksignalen ermöglichte und für dessen Förderung ich der Deutschen Forschungsgemeinschaft sehr dankbar bin.

Ganz herzlich möchte ich mich bei meinen beiden Betreuern, Michael Clausen und Meinard Müller, für die persönliche und inspirierende Anleitung bedanken, die auch große Freiräume zum Umsetzen eigener Ideen und Ansätze sowie die anschließenden Hilfestellungen bei der Aufarbeitung der Ergebnisse beinhaltete. Weiterhin haben sie durch ihre motivierende und geduldige Art mein Verständnis für gute Wissenschaft geprägt und standen mir während der gesamten Promotionsphase als Diskussionspartner zur Seite.

Weiterhin gilt mein aufrichtiger Dank sowohl der (ehemaligen) Arbeitsgruppe Clausen an der Universität Bonn mit Sebastian Ewert, Verena Kriesel und David Damm als auch der Arbeitsgruppe Müller am Max-Planck-Institut für Informatik in Saarbrücken und später an den International Audio-Laboratories Erlangen mit Jonathan Driedger, Thomas Prätzlich, Nanzhu Jiang, Christian Dittmar, Verena Konz und Peter Grosche für die vielfältige Unterstützung, sei es durch Anregungen und Diskussionen, durch konkrete Erklärungen, durch gemeinsam aufgebaute und gepflegte Datenbestände und gemeinsam verwendeten Programmcode, oder einfach nur durch die von echter Freundschaft und Teamgeist geprägte familiäre Atmosphäre. Unserer wissenschaftlichen Hilfskraft Polina Gubaidullina möchte ich herzlich danken für die intensive Zusammenarbeit – insbesondere bei den Annotationen zur Winterreise – sowie die vielen Einblicke in die Sicht einer Musikwissenschaftlerin auf unsere Ideen, Ansätze und Vorgehensweisen.

Für viele anregende Diskussionen und Hinweise danke ich Christoph Weiß, Jordan Smith, Geoffroy Peeters, Masataka Goto und vielen anderen Mitgliedern unserer ISMIR-Gemeinschaft. Bei Susanne Gammer und Benjamin Hilger bedanke ich mich für zahlreiche Kommentare und Korrekturvorschläge. Nicht zuletzt möchte ich mich bei meiner Partnerin Melanie, meinen Eltern Corinna und Raphael und meinem Bruder Richard herzlich für die liebevolle Unterstützung bedanken und dafür, dass sie stets ein offenes Ohr für fachliche wie außerfachliche Angelegenheiten hatten.

[2] Mehrschichtige Analyse und Strukturierung von Musiksignalen

2. Grundlagen

Bei der automatisierten Verarbeitung von Musiksignalen steht man aufgrund der Vielfältigkeit von Musik in Form und Inhalt vor großen Herausforderungen. In diesem Kapitel stellen wir die Grundlagen der in dieser Arbeit präsentierten Problemstellungen und Lösungsmethoden vor. Hierzu geben wir einen Überblick über unterschiedliche Aspekte der Segmentierung und Strukturierung von Musiksignalen, bei dem wir zum einen auf verschiedene musikalische Aspekte wie Tempo, Rhythmus, Dynamik, Harmonik und Klangfarbe eingehen und zum anderen mehrere Segmentierungsprinzipien wie Wiederholung, Homogenität und Novelty vorstellen. Im Rahmen der Einführung einer mathematischen Modellierung dieses Problems beschreiben wir diese Prinzipien und deren Auswirkungen auf die erreichbaren Segmentierungen.

Um die Qualität der Resultate verschiedener Verfahren zu vergleichen, wird im Regelfall ein automatisiertes Auswertungsverfahren verwendet. Nach einer kurzen Vorstellung dieser Methode diskutieren wir im Anschluss die Möglichkeiten und Beschränkungen der dabei verwendeten Evaluationsmaße.

2.1. Einleitung

Segmentierung und Strukturierung sind für die automatisierte Verarbeitung von Musiksignalen von grundlegender Bedeutung[1]. Grob gesprochen geht es bei der *Segmentierung* um die Zerlegung eines Audiodatenstroms in inhaltlich sinnvolle Abschnitte und elementare Einheiten. Hierauf aufbauend werden bei der *Strukturierung* diese Abschnitte nach bestimmten Kriterien bezüglich ihrer Bedeutung oder Funktion semantischen Kategorien zugeordnet. Eine solche Strukturierung kann sich zum Beispiel auf die musikalische Form eines Musikstückes beziehen. Im Fall von Popmusik ist hierbei eine Audioaufnahme in Blöcke zu segmentieren, die der Intro (Einleitung), den Strophen, den Refrains und der Outro entsprechen. Oder im Fall der klassischen Sonatensatzform können sich die Blöcke auf Exposition, Durchführung, Reprise und Coda beziehen.

Musikalische Wiederholungsstrukturen werden häufig durch Abfolgen von (indizierten) Buchstaben wie zum Beispiel $A_1 B_1 C_1 C_2 A_2 B_2 C_3 D$ beschrieben, siehe auch Abbildung 2.1b. Hierbei beziehen sich gleiche Buchstaben auf sich wiederholende Blöcke und die Indizes auf die jeweiligen Positionen der Wiederholungen. Obige Abfolge besagt also, dass das zugrundeliegende

[1] Ein Großteil dieses Abschnitts ist aus [123] übernommen.

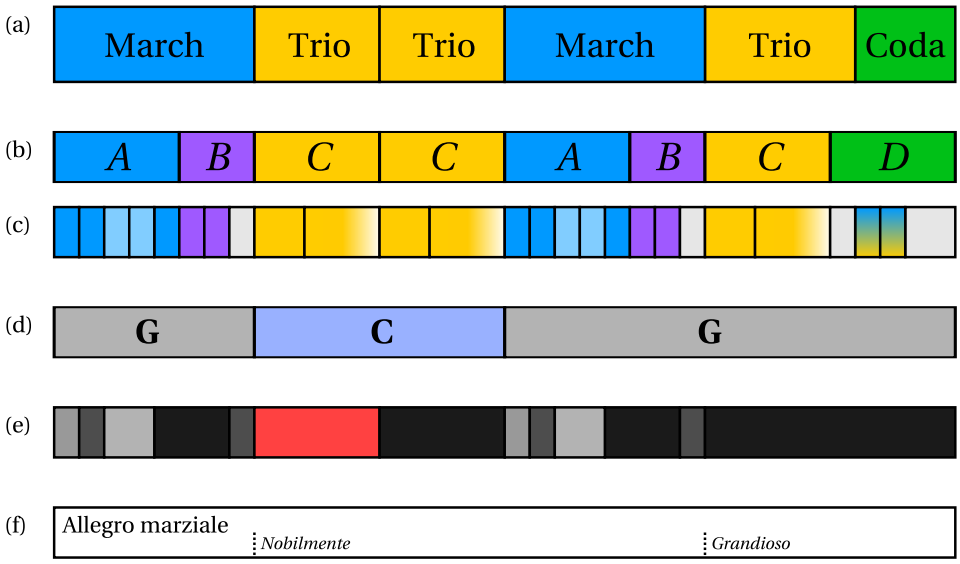


Abbildung 2.1.: Manuelle Strukturanalyse des Marsches *Pomp and Circumstance No. 4* von Edward Elgar: **(a)** Musikalische Form mit semantischen Bezeichnungen, **(b)** wiederholungsbasierte Grobstruktur, **(c)** wiederholungsbasierte Feinstruktur – in der Coda werden beide Themen verwoben, **(d)** Strukturierung nach Tonarten, **(e)** Strukturierung nach Instrumentierung – Schwarz/Grauabstufungen für Tutti, Rot für Streicher mit Holzbläsern, **(f)** Tempo- und Spielanweisungen.

Musikstück (hier: *Pomp and Circumstance March No. 4 in G major* von Edward Elgar[2]) aus zwei sich wiederholenden A-Teilen A_1 und A_2, zwei sich wiederholenden B-Teilen B_1 und B_2, drei sich wiederholenden C-Teilen C_1, C_2 und C_3 sowie einem abschließenden D-Segment besteht. Nach [82] ist hierbei zu bedenken, dass diese Buchstaben ohne weitere Erläuterungen keine große Aussagekraft haben.

Im Allgemeinen muss man bei Strukturierung von Musik ganz unterschiedliche zeitliche Skalen berücksichtigen, die oft hierarchisch angeordnet werden können. So können die Teile einer musikalischen Struktur häufig weiter untergliedert werden, indem man prägnante, sich wiederholende Ton- oder Akkordfolgen berücksichtigt. Diese können zum Beispiel ein Riff in Popmusik oder musikalische Themen und Motive im Fall klassischer Musik sein. Auf einer noch feineren zeitlichen Stufe können dann einzelne Akkorde, Töne, oder Noteneinsatzzeiten betrachtet werden.

Im Rahmen einer musikalischen Formanalyse des obigen Beispiels (siehe Abbildung 2.1a) ist die Interpretation der mit A und B bezeichneten Segmente als zwei Hälften eines zusam-

[2] Der Komponist schrieb die ersten vier Märsche in der Zeit zwischen 1901 und 1907, ein fünfter wurde 1930 fertiggestellt. Der Titel greift eine Zeile aus dem Stück *Othello* von William Shakespeare (3. Akt, 3. Szene) auf, wo es heißt: »*Pride, pomp, and circumstance of glorious war!*« [91].

mengehörenden thematischen Segments (»Marsch«) naheliegend, das in Kontrast zu dem melodisch eingängigen »Trio« bestehend aus den C-Teilen steht. Im abschließenden D-Teil werden schließlich die unterschiedlichen Themen von Marsch und Trio miteinander verwoben (siehe Abbildung 2.1c). Somit ist für dieses Beispiel die alternative Strukturannotation $AA'BBAA'B'C$ näher an der musikalischen Form, bildet allerdings die Wiederholungsstruktur nicht adäquat ab: Das mit A' bezeichnete Segment ist keine variierte Wiederholung von A, sondern stellt eine eigenständige, mit A lediglich thematisch verwandte Phrase dar. Wir bevorzugen daher im Folgenden die zuerst eingeführte wiederholungsbasierte Annotation. Auf einer feineren musikalischen Auflösung lassen sich etwa die als A bezeichneten Segmente in 5 Untersegmente unterteilen, wobei das 1., 2. und 5. Segment Wiederholungen desselben achttaktigen Themas sind, siehe auch Abbildung 2.1c. Für den Notentext dieses Beispiels siehe Abbildung 2.2.

Die Segmentierung und Strukturierung stellen oft den ersten Schritt für eine anschließende Weiterverarbeitung der Musiksignale dar, wie beispielsweise eine Klassifizierung, Annotation oder Indexierung. Diese Aufgaben sind zentrale Fragestellungen des noch relativ jungen Forschungsgebiets des *Music Information Retrieval* (MIR). Allgemeine Ziele dieses Gebiets liegen in der Entwicklung von Methoden und Systemen, die Benutzern große, in digitaler Form vorliegende Musikkollektionen in vielfältiger Weise zugänglich machen. MIR stellt ein interdisziplinäres Forschungsgebiet dar, das eine Vielzahl von Fachgebieten wie z. B. die Informatik (darunter die Signalverarbeitung), Musikwissenschaft und Bibliothekswissenschaft einschließt. Für einen allgemeinen Überblick über die beteiligten Disziplinen, über zentrale MIR-Fragestellungen und über bestehende MIR-Systeme verweisen wir auf die folgenden Überblicksartikel und Bücher [14, 37, 84, 116, 140, 195]. Eine umfangreiche Sammlung an Forschungsartikeln, die den aktuellen Stand der MIR-Forschung repräsentieren, stellen die Sammelbände[3] der jährlich stattfindende Konferenz der *International Society on Music Information Retrieval* (ISMIR) dar.

In diesem Kapitel stellen wir allgemeine musikalische Aspekte und Prinzipien vor, die bei der Segmentierung und Strukturierung von Musiksignalen von Bedeutung sind, siehe auch Abbildung 2.3. Nach einem kurzen Überblick über die verschiedenen Segmentierungsprinzipien (Abschnitt 2.2) und der Darstellung musikalischer Aspekte durch numerische Merkmale (Abschnitt 2.3) stellen wir mit den in Abschnitt 2.4 eingeführten Selbstähnlichkeitsmatrizen das zentrale Werkzeug für die Strukturanalyse vor. Diese Abschnitte stammen aus bzw. basieren zu großen Teilen auf [123]. In Abschnitt 2.5 stellen wir eine mathematische Modellierung des Strukturierungs- und Segmentierungsproblems vor und führen die in der restlichen Arbeit verwendete Notation ein. Der abschließende Abschnitt 2.7 stellt die verwendeten Verfahren zur automatischen Evaluation von Musikstrukturen vor und diskutiert ihre Grenzen.

[3] Die ISMIR-Sammelbände sind online auf der Webseite `http://www.ismir.net/` frei erhältlich.

2. Grundlagen

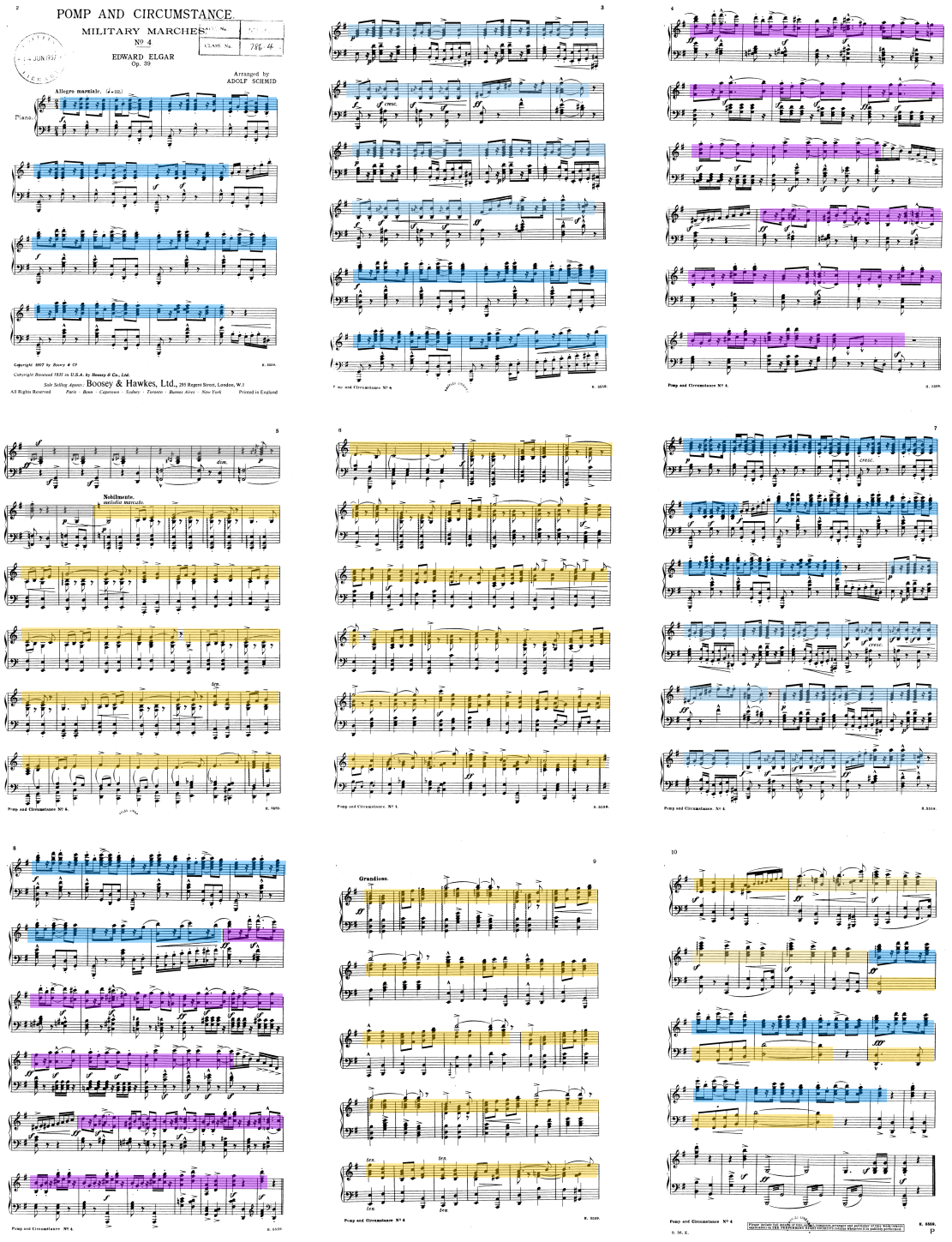


Abbildung 2.2.: Klavierauszug des *Pomp and Circumstance March No. 4* von Edward Elgar (Arrangement von Adolf Schmid, 1907) mit farblicher Hervorhebung der Feinstruktur.

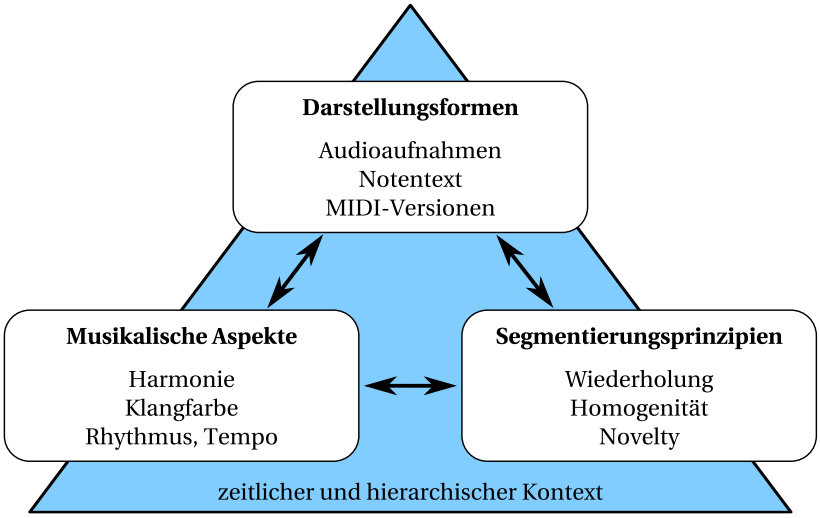


Abbildung 2.3.: Schematische Darstellungen einiger grundlegender Aspekte und Prinzipien für die Segmentierung und Strukturierung von Musiksignalen, nach [123].

2.2. Prinzipien der Segmentierung und Strukturierung

In der Literatur findet man zahlreiche Verfahren zur automatisierten Strukturanalyse, die auf ganz unterschiedlichen Annahmen basieren [28, 147, 153]. Im Folgenden wollen wir die wesentlichen zugrundeliegenden Prinzipien und deren musikalische Relevanz diskutieren. Hierbei folgen wir der in [147] vorgeschlagenen Terminologie. Dieser Abschnitt ist größtenteils aus [123] übernommen.

Bei *Novelty-basierten Verfahren* zur Segmentierung geht es um die Erkennung von Zeitpunkten oder Übergangsbereichen, in denen neuartige Signaleigenschaften auftreten [6, 49]. Dies können zum einen plötzliche energie- und spektralbasierte Signaländerungen sein, wie sie zum Beispiel in der Anschlagphase beim Anspielen einer Note auftreten [6]. Zum anderen können dies auch eher glattere Signalübergänge sein, die zum Beispiel in Form von Klangfarbenwechseln durch Änderungen in der Instrumentierung auftreten. Novelty-basierte Verfahren können als Spezialfall von *ereignisbasierten Verfahren* angesehen werden, bei denen es um die unüberwachte bzw. überwachte Detektion statistisch hervorstechender bzw. als Vorwissen spezifizierter Ereignisse im Audiodatenstrom geht. Eine weitere wichtige Klasse von Segmentierungsverfahren und sozusagen das Komplement stellen *homogenitätsbasierte Verfahren* dar, bei denen es um eine Unterteilung eines Audiodatenstroms in Abschnitte geht, die jeweils in sich bezüglich eines ausgezeichneten Aspekts homogen sind [100, 145]. Die Homogenität kann sich hierbei zum Beispiel auf die Klangfarbe, die Dynamik oder die Harmonik beziehen. Schließlich ist die Erkennung wiederkehrender Muster das Ziel von *wiederholungsbasierten*

Verfahren [18, 19, 59, 103, 105, 139, 152, 176], die eine zentrale Rolle bei der Segmentierung von Musiksignalen spielen.

Alle genannten Segmentierungsparadigmen sind im Musikbereich von zentraler Bedeutung und spiegeln sich zumindest mittelbar in musikalischen Gestaltungsprinzipien wider. Um dies zu erklären, wollen wir im Folgenden von der relativ stark vereinfachten Sichtweise ausgehen, dass sich die musikalische Struktur einer Musikaufnahme auf den groben zeitlichen Ablauf musikalisch sinnvoller Blöcke bezieht. Die musikalische Struktur ist sowohl für das Verständnis als auch die Erschließung von Musik von großer Bedeutung und steht über die musikalische Form (zum Formbegriff siehe [81, 82]) mit Gattung und Funktion eines Musikstückes in enger Beziehung [99].

Die in der Abfolge auftretenden Blöcke hängen semantisch oftmals zusammen, folgen dabei einem festen Schema und können sich damit ins Gedächtnis einprägen [81, 131]. Im wesentlichen folgen diese Beziehungen drei Gestaltungsprinzipien: Bei der *Wiederholung* werden Gedanken und Teile mehr oder weniger unverändert aufgegriffen. Die entsprechenden Blöcke gleichen sich dann bezüglich bestimmter Aspekte wie der Melodik, Harmonik oder Rhythmik. Über solche wiederkehrenden Muster wird ein zeitlicher Bezug innerhalb des Stücks hergestellt, der vom Hörer nachvollzogen werden und der das Gefühl der Vertrautheit und des musikalischen Verstehens hervorrufen kann. Als zweites Gestaltungsprinzip dient der *Kontrast*, bei dem zwei Blöcke unterschiedlichen Charakters aufeinandertreffen. Zum Beispiel folgt ein lauter Abschnitt einem leiseren, ein langsamer einem schnellen oder ein kammermusikalischer einem orchestralen [111]. Durch die bewusste Konstruktion eines Bruches wird eine kontrastierende Wirkung erzielt, die der Hörer quasi als Gegenstück zur Wiederholung als überraschendes Element erlebt und die dem Stück Farbigkeit verleiht. Das dritte Gestaltungsprinzip ist das der *Variation*. Hierbei werden Gedanken und Teile in abgewandelter Form aufgegriffen, wobei das Original immer noch durchscheint und wiedererkennbar bleibt. Die Abwandlungen können dabei unterschiedliche musikalische Aspekte wie Klangfarbe, Dynamik, Instrumentation, Harmonik oder Satzweise (Poly- und Homophonie) betreffen. Bei [82] wird noch die *Beziehungslosigkeit* als Ausdruck der Unkorreliertheit zweier Segmente aufgeführt. Diese steht im Gegensatz sowohl zu Wiederholung und Variation, die man als positive Korrelation zwischen den Segmenten beschreiben könnte, als auch zum Kontrast, der einer negativen Korrelation bezüglich eines speziellen Aspekts entspricht.

Das zentrale Ziel der Strukturanalyse von Musiksignalen besteht darin, mit automatisch agierenden Methoden die grobe musikalische Struktur direkt aus den Audioaufnahmen eines Musikstückes zu bestimmen. Um den unterschiedlichen Gestaltungsprinzipien gerecht zu werden, spielen alle Segmentierungsmethoden gleichermaßen eine zentraler Rolle. *Wiederholungsbasierte Verfahren* werden zur Identifikation wiederkehrender Muster wie Themen oder Phrasen sowie leichter Variationen benötigt. Durch *Novelty-basierte Verfahren* sind Übergänge zwischen kontrastierenden Blöcken aufzuspüren. *Homogenitätsbasierte Verfahren* dienen dazu, rhythmisch, harmonisch oder klanglich konsistente Bereiche zu erfassen, die in manchen Fällen als Varianten identifiziert werden können (z. B. melodische Variationen über ähnlichen Harmonieabfolgen). Allerdings stellt sich hierbei das Problem, dass es ohne weitere Informa-

tionen zum Kontext nicht möglich ist, die musikalische Relevanz der Homogenität eines oder mehrerer bestimmter musikalischer Aspekte zu bewerten. Bei einem Stück mag beispielsweise eine Änderung der Instrumentation eine große semantische Bedeutung haben und daher zur Strukturbildung beitragen, während bei einem anderen Stück die unterschiedlich vorkommenden Klangfarben keine strukturelle Bedeutung haben. Hierzu siehe auch [182] für eine Analyse der Korrelation zwischen einer manuell angefertigten Strukturierung und den homogenen Bereichen bezüglich verschiedener musikalischer Aspekte.

Bisher gibt es nur wenige Strukturierungsverfahren, welche die verschiedenen Segmentierungsprinzipien simultan berücksichtigten. Erste aktuelle in der Literatur beschriebene Methoden zeigen für den Fall von Popmusik, dass durch Kombination verschiedener Segmentierungsprinzipien erhebliche Verbesserungen in der Strukturanalyse erzielt werden können [146]. In Kapitel 3 diskutieren wir eine Prozedur, welche die Nutzung von Verfahren, die ursprünglich für die homogenitätsbasierte Segmentierung konzipiert worden sind, auch für die Segmentierung nach Wiederholungen erlaubt.

2.3. Musikalische Aspekte und Merkmale

Wie oben dargestellt können sich die Segmentierungsprinzipien auf ganz unterschiedliche musikalische Aspekte beziehen[4]. So kann ein Segment homogen bezüglich Klangfarbe, Dynamik oder Harmonik sein. Auf der anderen Seite können sich wiederholende Segmente durch ähnliche Harmonieverläufe auszeichnen, aber sich hinsichtlich der Instrumentierung oder Dynamik erheblich unterscheiden. Weiterhin sind Wiederholungsstrukturen auf einer zeitlich feingranularen Stufe oft mit rhythmischen Aspekten korreliert. Die unterschiedlichen musikalischen Aspekte spiegeln sich in den akustischen Eigenschaften eines als Wellenform gegebenen Musiksignals wider. Auch wenn der Mensch beim Hören einer Audioaufnahme oft sofort die relevanten Aspekte wahrnehmen kann, ist die automatische Extraktion musikalisch sinnvoller Merkmale aus Wellenformdaten eine im allgemeinen sehr schwierige Aufgabenstellung. Ein zentrales Ziel der Musiksignalverarbeitung besteht darin, ein gegebenes Musiksignal in geeignete Merkmalsdarstellungen zu transformieren, die zu unterschiedlichen musikalischen Aspekten korrelieren, siehe [118]. Im Folgenden wollen wir uns exemplarisch drei solchen Merkmalsdarstellungen zuwenden, siehe auch Abbildung 2.4. Als Hauptmusikbeispiel werden wir in dieser Arbeit auf eine Aufnahme von Edward Elgars *Pomp and Circumstance March No. 4* zurückgreifen[5].

Zum einen spielt bei der Strukturierung von Musiksignalen die *Instrumentierung* und *Klangfarbe* eine wichtige Rolle [11]. Hierbei steht die Klangfarbe häufig mit der Energieverteilung und deren zeitlichen Entwicklung in sogenannten kritischen Spektralbändern in Beziehung.

[4] Dieser Abschnitt wurde ebenfalls bereits in [123] veröffentlicht.

[5] Aufnahme des London Philharmonic Orchestra unter der Leitung von Sir Adrian Boult 1977, remastered 1986. Veröffentlicht von EMI Records Ltd. als Track 19 der Audio-CD »British Composers: Elgar, Enigma Variations, Pomp & Circumstance Marches Nos. 1–5« (ASIN B00000DO96), 1991.

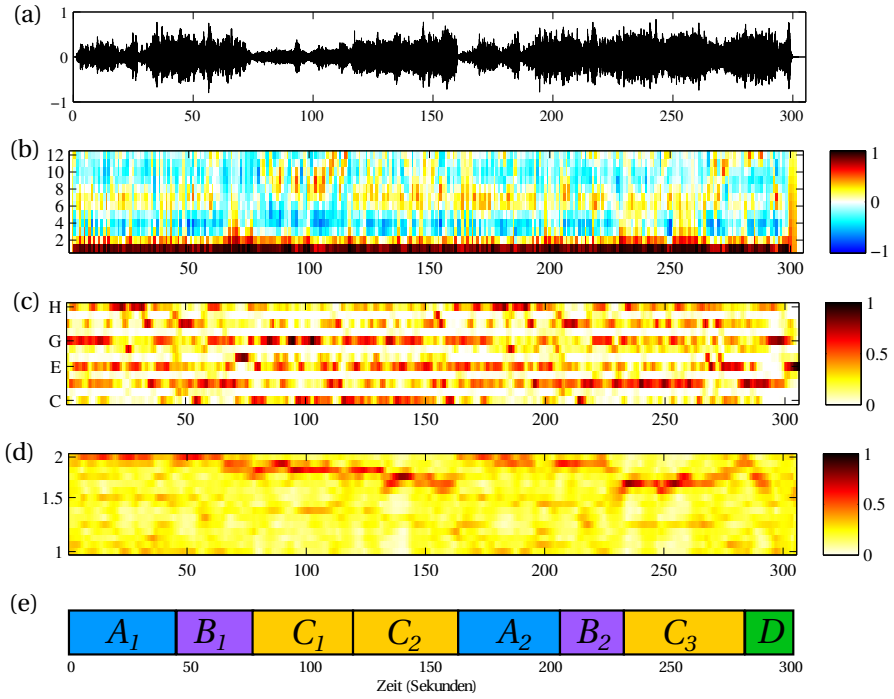


Abbildung 2.4.: Merkmalsdarstellungen für eine Aufnahme von Edward Elgars *Pomp and Circumstance March No. 4*: **(a)** Wellenform, **(b)** MFCC-basierte Merkmale, **(c)** Chroma-basierte Merkmale, **(d)** Tempo-basierte Merkmale. **(e)** Manuell annotierte wiederholungsbasierte Grobstruktur.

Bei der Analyse von Musiksignalen wird daher häufig auf als MFCCs (*Mel Frequency Cepstral Coefficients*) bekannte Merkmale zurückgegriffen, die ursprünglich für die Spracherkennung entwickelt wurden [31]. Nach Transformation des Musiksignals in eine Spektraldarstellung werden MFCC-basierte Merkmale durch Zusammenfassen geeigneter Frequenzbänder in perzeptuell motivierte Mel-Bänder und Anwendung einer dekorrelierenden Cosinustransformation gewonnen. Insbesondere die unteren MFCCs beschreiben die grobe Form der spektralen Einhüllenden, die wiederum zur Klangfarbe korreliert [192]. Abbildung 2.4b zeigt für unser Beispiel eine MFCC-basierte Merkmalsfolge (vgl. [213]) für die unteren 12 Koeffizienten. Die Merkmalsdarstellung spiegelt zum Beispiel wider, dass sich der erste C-Teil deutlich von allen anderen Teilen hinsichtlich der Klangfarbe unterscheidet.

Während sich MFCC-basierte Merkmale typischerweise für homogenitätsbasierte Segmentierungsaufgaben eignen, sind sie für wiederholungsbasierte Verfahren eher ungeeignet. Ein Grund hierfür ist die Tatsache, dass sich wiederholende Passagen melodisch deutlich weniger Unterschiede aufweisen als in Bezug auf die Klangfarbe, die selbst bei identischer Instrumentierung relativ starken Schwankungen unterworfen ist. Weiterhin wird in vielen Kompositionen bei Wiederholungen gerne die Instrumentierung variiert, wodurch die Unterschiede

in der Klangfarbe weiter zunehmen. Man denke hierbei zum Beispiel an eine Strophe mit Gesang, die später nochmals als reine Instrumentalversion erscheint. Grundlage der wiederholungsbasierten Strukturanalyse sind daher oft die in [196] eingeführten *Chroma-basierten Merkmalsdarstellungen*, die stark mit dem *Harmonieverlauf* des zugrundeliegenden Musikstückes korrelieren und ein hohes Maß an Invarianz bezüglich Änderungen in der Klangfarbe aufweisen [5, 55, 116, 128]. Diese harmoniebasierten Merkmale werden auch häufig für die Bestimmung lokal vorherrschender Tonarten sowie für die Akkordanalyse verwendet [69, 107, 166]. Ähnlich wie die MFCCs können Chroma-Merkmale aus einer Spektraldarstellung des Musiksignals abgeleitet werden. Hierbei werden allerdings die Frequenzbänder in musikalisch motivierte Tonhöhenbänder (gemäß der temperierten Stimmung entsprechend einer Klaviertastatur) zerlegt. Jede Tonhöhe kann eindeutig durch einen der zwölf Chromawerte $\mathrm{C}, \mathrm{C}^\sharp, \mathrm{D}, \ldots, \mathrm{H}$ und seine Oktavlage beschrieben werden. Im nächsten Schritt werden alle zum gleichen Chroma korrespondierenden Tonhöhenbänder zu einem Chromaband aufsummiert. Zum Beispiel werden die Energiewerte der Bänder zu den Tonhöhen A0, A1, …, A7 zu einem Energiewert zum Chroma A zusammengefasst. Nach einem anschließenden Normalisierungsschritt erhält man schließlich eine Folge von 12-dimensionalen Chromavektoren, wobei jeder dieser Vektoren die lokale Energieverteilung der im Audiosignal vorkommenden Frequenzen auf die 12 Chromabänder widerspiegelt. So zeigt zum Beispiel die *Chroma-basierte Merkmalsfolge* (auch *Chromagramm*[6] genannt) in Abbildung 2.4c, dass sich die Teile C_1 und C_2 vom Rest des Stückes harmonisch unterscheiden.

Neben Klangfarbe und Harmonik stellen *Tempo* (Schläge pro Minute) und *Rhythmus* , d. h. das relative zeitliche Verhältnis der Töne, weitere wichtige Aspekte der Musik dar. Zur Erfassung solcher Eigenschaften gehen die meisten Verfahren in zwei Schritten vor. Im ersten Schritt werden die Zeitpositionen möglicher Noteneinsätze (*onsets*) bestimmt. Hierbei nutzt man die Eigenschaft aus, dass das Anspielen eines Tons meist mit einer messbaren Änderung in der Signalenergie und der Frequenzzusammensetzung einhergeht [6]. Im zweiten Schritt werden die Noteneinsatzkandidaten dann hinsichtlich periodischer Muster untersucht, aus denen sich anschließend eine *Tempo-basierte Merkmalsdarstellung* (oder auch *Tempogramm*[7]) ableiten lässt [65, 154]. Ähnlich wie bei den zyklischen Chromagrammen, bei denen oktaväquivalente Tonhöhen identifiziert werden, können auch zyklische Tempogramme betrachtet werden, bei denen Zweierpotenz-äquivalente Tempi identifiziert werden [66]. Dies führt auf robuste Merkmalsdarstellungen, die lokale Tempounterschiede erfassen können. Das in Abbildung 2.4d dargestellte zyklische Tempogramm legt zum Beispiel die leichten Tempounterschiede zwischen den *A*- und *B*-Teilen auf der einen Seite und den *C*-Teilen andererseits offen. Tempogramme und daraus abgeleitete Schätzungen des rhythmischen Grundschlags werden in Kapitel 5 detaillierter besprochen.

[6] Unterschiedliche Varianten Chroma-basierter Merkmale sind Teil der unter `www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/` frei erhältlichen *Chroma Toolbox*, siehe auch [120].

[7] Tempo-basierter Merkmale sind Teil der unter `http://www.mpi-inf.mpg.de/resources/MIR/tempogramtoolbox/` frei erhältlichen *Tempogram Toolbox*.

2.4. Selbstähnlichkeitsmatrizen

Die Umwandlung eines Musiksignals in eine geeignete Merkmalsdarstellung stellt den ersten Schritt nahezu jedes Strukturanalyseverfahrens dar[8]. Hierbei hat der Typ der verwendeten Merkmalsdarstellung einen erheblichen Einfluss auf die erkennbaren Strukturen. Diese Tatsache spiegelt sich auch in sogenannten *Selbstähnlichkeitsmatrizen*[9] (engl. *self-similarity matrix*, SSM) wider, auf die wir im Folgenden näher eingehen wollen. Wie bereits erwähnt, wird in den meisten Ansätzen zur automatisierten Strukturanalyse das Musiksignal in eine Folge von Merkmalsvektoren transformiert. Unter Verwendung eines geeigneten Ähnlichkeitsmaßes wie etwa dem Standardskalarprodukt werden diese Vektoren dann paarweise verglichen. Die resultierenden Ähnlichkeitswerte können in einer quadratischen Selbstähnlichkeitsmatrix erfasst und durch eine geeignete Farbkodierung (z. B. dunkle Farbe für große und helle Farbe für kleine Werte) bildlich dargestellt werden. Abbildung 2.5 zeigt einige Selbstähnlichkeitsmatrizen, die auf den in Abbildung 2.4 dargestellten Merkmalsdarstellungen basieren.

Die strukturellen Bezüge innerhalb einer Merkmalsfolge werden in der resultierenden Selbstähnlichkeitsmatrix sichtbar. Grob kann man dabei zwischen zwei unterschiedlichen Mustern unterscheiden [28, 147]. Wenn sich zum einen die Elemente der Merkmalsfolge über einen gewissen Zeitraum nur wenig unterscheiden, so entspricht dies einem homogenen Segment. Ein paarweiser Vergleich dieser Elemente führt durchgängig zu großen Ähnlichkeitswerten, so dass in der resultierenden Selbstähnlichkeitsmatrix ein quadratischer *Block* sichtbar wird. Wenn zum anderen die Merkmalsfolge zwei sich wiederholende Teilfolgen enthält, dann sind die sich entsprechenden Elemente der beiden Teilfolgen ähnlich. Im allgemeinen sind aber die jeweiligen Teilfolgen in sich nicht notwendigerweise homogen. Anstelle eines Blocks wird damit in der Selbstähnlichkeitsmatrix ein *Pfad* sichtbar, dessen Projektionen auf die beiden Achsen den beiden Teilfolgen entsprechen.

Als Illustration betrachten wir wieder die Selbstähnlichkeitsmatrizen in Abbildung 2.5. Deutliche Blockstrukturen sind zum Beispiel im Fall der Tempo-Selbstähnlichkeitsmatrix (Abbildung 2.5b) in denjenigen Passagen erkennbar, wo das Tempo mehr oder weniger konstant bleibt. Bei der Chroma-Selbstähnlichkeitsmatrix (Abbildung 2.5c) sind bei den C-Teilen Blöcke erkennbar, was an der gleichbleibenden Harmonik in diesen Teilen liegt. Auf der anderen Seite zeigt diese Matrix auch deutliche Pfadstrukturen, welche sowohl die Wiederholungen der A- und B-Teile widerspiegeln als auch die wiederholungsbasierte Substruktur des A-Segmentes anzeigen. In Abbildung 2.5d wird illustriert, wie man durch geeignete Glättungs- und Schwellwertverfahren sowie Anwendung von Methoden aus der Bildverarbeitung die Pfadstruktur einer Selbstähnlichkeitsmatrix erheblich verbessern kann [126, 153, 173]. In Abschnitt 3.2 werden Anwendung und Auswirkung solcher Verfahren detaillierter besprochen.

Viele der in der Literatur beschriebenen Strukturanalyseverfahren basieren auf der Extraktion und Interpretation der Block- und Pfadmuster in geeignet definierten Selbstähnlichkeitsmatri-

[8] Dieser Abschnitt basiert auf [123].
[9] Häufig findet man in der Literatur auch den Begriff des *Rekurrenzplots*.

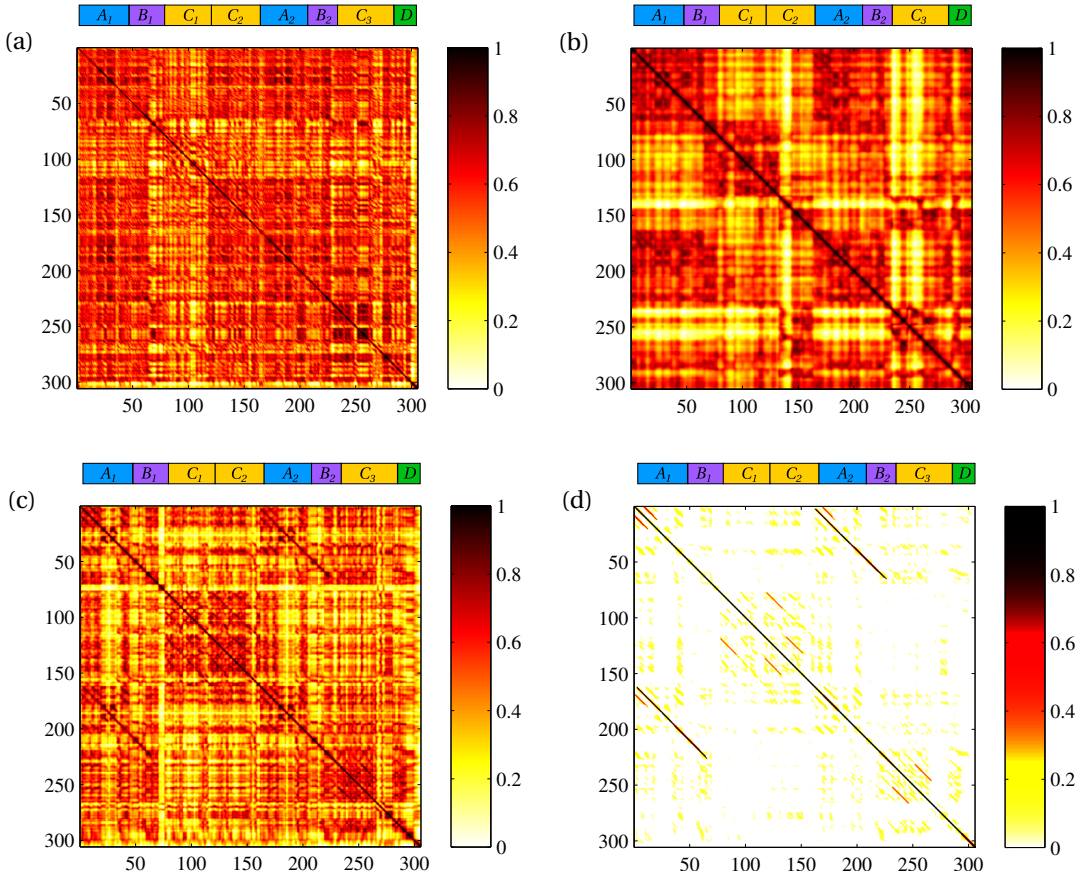


Abbildung 2.5.: Normalisierte Selbstähnlichkeitsmatrizen für die in Abbildung 2.4 dargestellten Merkmale: **(a)** MFCC-basierte Merkmale, **(b)** Tempo-basierte Merkmale, **(c)** Chroma-basierte Merkmale. **(d)** zeigt eine strukturell verbesserte Version von (c).

zen. Allerdings sind diese Muster oft stark verrauscht und fragmentiert. Weiterhin hängt die Deutlichkeit der Blöcke und Pfade nicht nur vom Typ der Merkmalsdarstellung ab, sondern vom verwendeten Ähnlichkeitsmaß und vor allem auch von der Fenstergröße und zeitlichen Auflösung bei der Merkmalsberechnung. Oft ist bei der Strukturanalyse ein Vergröberungs- oder Glättungsschritt nicht nur hinsichtlich des Rechenaufwands, sondern auch aus strukturellen Gründen von Vorteil, insbesondere bei der wiederholungsbasierten Segmentierung.

Hierfür sind nicht alle musikalischen Aspekte gleichermaßen geeignet, da bei Wiederholungen Melodie und Harmoniefolgen im Allgemeinen weniger bzw. in einem stärker festgelegten Rahmen variiert werden als die anderen Aspekte wie Klangfarbe und Tempo [82]. Daher werden für die wiederholungsbasierte Segmentierung oftmals Chroma-basierte Merkmale verwendet, siehe auch [116]. Eine dieser Variationen ist das Auftreten einer transponierten Wiederholung, d. h. ein Segment erscheint in verschiedenen Tonarten. Um diese Wiederholungen erkennen zu können, werden sogenannte transpositionsinvariante Selbstähnlichkeitsmatrizen betrach-

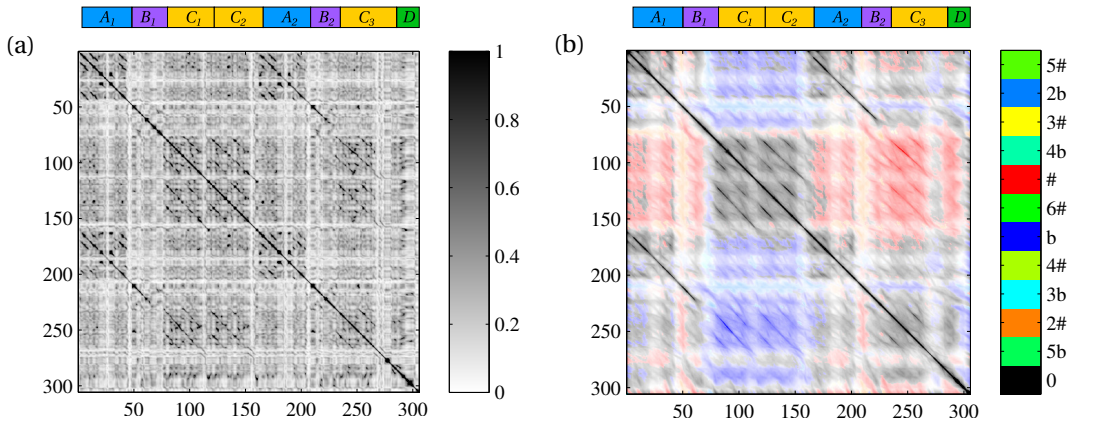


Abbildung 2.6.: Transpositionsinvariante Selbstähnlichkeitsmatrix mit dazugehörigem Transpositionsindex.

tet, wie sie in [117] vorgestellt wurden. Hierbei wird die Merkmalsfolge auch mit allen 12 möglichen Transpositionen verglichen, und in die Selbstähnlichkeitsmatrix wird für jede Position der höchste Ähnlichkeitswert eingetragen. Somit wird in Abbildung 2.6a deutlich, dass es sich in unserem Beispiel bei dem Segment C_3 tatsächlich um eine Wiederholung von C_1 und C_2 handelt – allerdings in einer anderen Tonart. Durch Eintragen der jeweiligen Transpositionsindizes in eine weitere Matrix kann weiterhin die harmonische Beziehung zwischen diesen Tonarten ermittelt werden. Bei diesem Stück stehen C_1 und C_2 in der Subdominanttonart des Stücks, wohingegen die zweite Wiederholung C_3 in der Tonika erscheint, woraus das dominantische Verhältnis zwischen diesen Segmenten entsteht.

Zur Verdeutlichung der harmonischen Beziehungen dieser Transpositionen verwenden wir die in Abbildung 2.7 dargestellte Farbskala. Diese weist der als Tonika erkannten Tonart den Farbton schwarz, der Dominante rot und der Subdominante blau zu. Bewegt man sich auf dem Quintenzirkel weiter in dominantische Richtung, so ändert sich die Farbe von rot über gelb bis hin zu grün, in Subdominantrichtung werden die Farben blau, cyan und türkis verwendet. Die Skala verwendet dabei deutlich unterscheidbare Farben für die verwandten Tonarten und weniger differenzierte Farben für die entfernten Tonarten. In Abschnitt 4.3.2 werden wir aufbauend auf dieser Farbskala eine Visualisierungsmethode für die verschiedenen lokalen Tonarten innerhalb eines Stückes vorstellen und diskutieren.

2.5. Modellierung des Strukturierungsproblems

In diesem Abschnitt stellen wir eine mathematische Modellierung der Segmentierung und Strukturierung von Musikstücken vor. Auf die hier eingeführten Begriffe und Bezeichnungen werden wir insbesondere in Abschnitt 3.4 zurückgreifen.

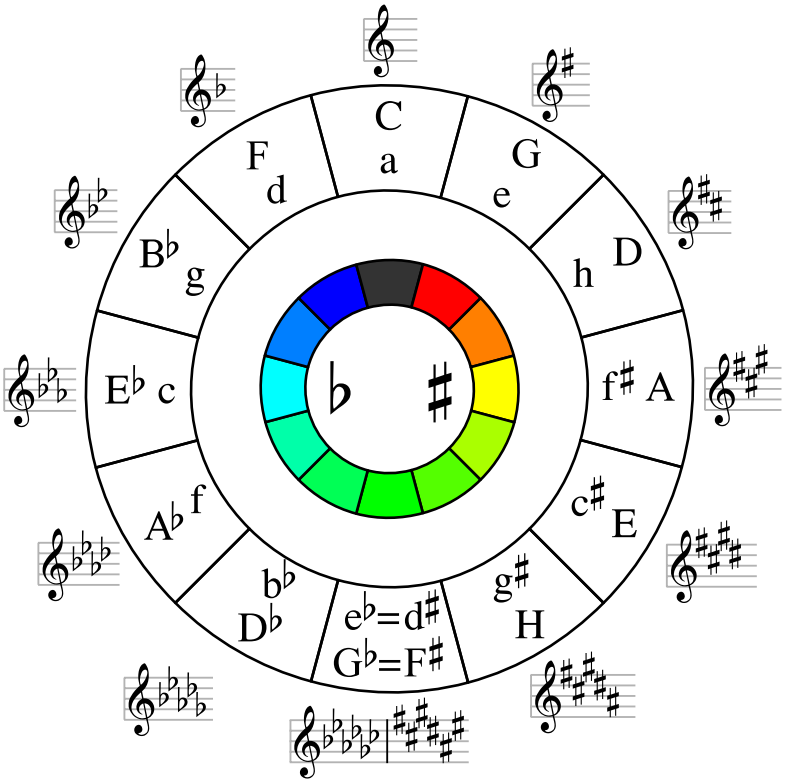


Abbildung 2.7.: Motivation unserer Farbskala zur Anzeige von Transpositionen aus dem Quintenzirkel. Man beachte, dass die Farben nicht absolut (schwarz=C-Dur), verwendet werden, sondern relativ bezüglich der (lokal) vorherrschenden Tonart (schwarz=Tonika).

Für die häufig vorkommenden Indexmengen verwenden wir angelehnt an MATLAB die Notation $[a:b] := [a,b] \cap \mathbb{Z}$, wobei $[a,b] := \{t \in \mathbb{R} \mid a \leq t \leq b\}$ für $a,b \in \mathbb{R}$. Für den Spezialfall $a,b \in \mathbb{N}, a < b$ entspricht dies der Menge $\{a, a+1, \ldots, b\}$.

Um die Ähnlichkeit von Segmenten eines Musikstückes beschreiben zu können, definieren wir zuerst für jedes Musikstück eine endliche M-elementige Menge $\mathcal{L}$ vorkommender *Segmentbezeichnungen* (engl. *segment labels*). Üblicherweise wählen wir für diese Bezeichnungen lateinische Großbuchstaben: $\mathcal{L} := \{A, B, C, \ldots\}$. Im Gegensatz zu den in Kapitel 2 vorgestellten Beispielen bei konkreten Musikstilen beinhalten diese Bezeichnungen üblicherweise keine semantische Bedeutung. Obwohl die Struktur vieler Musikstücke sich nicht eindeutig in klar disjunkte Segmentklassen zerlegen lässt (vgl. hierzu die Anmerkungen in [82] zur Problematik der Zuordnung einzelner Buchstaben zu den Segmenten) können wir durch Definition eines geeigneten Abstandsmaßes auf $\mathcal{L}$ die funktionale Nähe der verschiedenen Segmentklassen modellieren. Für die Praxis werden wir in den meisten Fällen jedoch auf die diskrete Metrik zurückgreifen, bei der alle Segmente mit verschiedenen Bezeichnungen ohne weitere Differenzierung als grundsätzlich verschieden angesehen werden. Zur besseren Übersicht ordnen wir

in diesem Fall die Buchstaben den Segmentklassen in der Reihenfolge ihres ersten Auftretens zu.

Die dem betrachteten Musikstück zugrundeliegende Zeitachse sei definiert als das Intervall $[1 : T]$, wobei $T \in \mathbb{N}$ die Länge des Stücks als Anzahl der verwendeten Merkmalsvektoren (engl. *feature vectors*) angibt. Diese Zahl berechnet sich aus der Länge des Stücks und der zeitlichen Auflösung der gewählten Merkmalsdarstellung. Aus diesem Grund werden wir bei der simultanen Verwendung mehrerer Merkmalsdarstellungen für alle Merkmale dieselbe zeitliche Auflösung wählen (bzw. diese mittels Interpolation herbeiführen). Für Details zu diesen Merkmalen siehe Abschnitt 2.3.

Mit $\mathcal{F}$ sei die Gesamtheit aller Vektoren eines speziellen Merkmals bezeichnet, also beispielsweise die Menge aller spektralen, MFCC-, Tempogramm- oder Chroma-Vektoren, und mit $\mathcal{F}^+ := \bigcup_{t \geq 1} \mathcal{F}^t$ die Menge aller möglichen Musikstücke in dieser Merkmalsdarstellung. Ein konkretes Stück der Länge $T \in \mathbb{N}$ wird durch eine Merkmalsfolge $f \in \mathcal{F}^T$ beschrieben. Für $1 \leq i < j \leq T$ bezeichnen wir die Teilfolge $(f_i, f_{i+1}, \ldots, f_j)$ von f mit $f_{i:j}$. Die Tatsache, dass $f_{i:j}$ eine Teilfolge von f ist, kürzen wir mit $f_{i:j} \sqsubset f$ ab.

Je nach betrachtetem Merkmal mag es zu restriktiv sein, zwei Merkmalsvektoren nur dann als gleichartig anzusehen, wenn sie tatsächlich gleich sind. Daher verwenden wir allgemeiner eine (nicht näher bestimmte) Äquivalenzrelation $\sim$, um eine starke Ähnlichkeit zwischen Elementen der Merkmalsmenge $\mathcal{F}$ auszudrücken. Diese setzen wir auf zwei verschiedene Weisen auf $\mathcal{F}^+$ fort: Im Falle einer *homogenitätsbasierten Strukturierung* sehen wir zwei Folgen $a, b \in \mathcal{F}^+$ als äquivalent an, wenn alle Glieder beider Folgen in einer einzigen $\sim$-Äquivalenzklasse liegen. Für die *wiederholungsbasierte Strukturierung* sind die beiden Folgen äquivalent, wenn sie gleichlang und komponentenweise $\sim$-äquivalent sind. Wir bezeichnen die beiden so definierten Äquivalenzrelationen auf $\mathcal{F}^+$ mit $\sim_H$ für den homogenitätsbasierten und mit $\sim_W$ für den wiederholungsbasierten Ansatz. Man beachte, dass diese Modellierung von idealen Homogenitätsbereichen ohne lokale »Ausreißer« sowie von exakten Wiederholungen insbesondere ohne Tempovariationen ausgeht. Bei der Strukturierung echter Musikaufnahmen treten eine Vielzahl von Abweichungen zum hier beschriebenen Idealfall auf, weswegen in der Praxis nichttriviale Glättungs- und Schwellwertverfahren zur Anwendung kommen. Aus Übersichtsgründen beschränken wir uns hier auf diesen Idealfall.

Abhängig von dem gewählten Strukturierungsprinzip erhalten wir eine Einteilung *aller* Teilfolgen von f in $\sim_H$- bzw. $\sim_W$-Äquivalenzklassen. Ein auf diese Sichtweise aufbauendes Verfahren zur Ermittlung harmonischer Informationen – was einer speziellen Form der Erkennung von Homogenitätsbereichen entspricht – stellt der in Abschnitt 4.3 vorgestellte Scape-Plot dar (siehe Abbildung 4.7). Eine ähnliche Darstellung für den wiederholungsbasierten Fall wird in [121] vorgestellt, bei der ein Abstandsmaß auf der Menge aller möglichen Teilfolgen berechnet wird, welches anschließend mittels einer kontinuierlichen Projektion auf den Farbkreis zur Illustration der Segmentähnlichkeiten verwendet wird.

Beim Versuch, aus diesen Informationen *eine* Segmentierung des Musikstückes abzuleiten, wird ein grundlegendes Problem der Strukturierungsaufgabe deutlich: Die Zuordnung jedes

Zeitpunktes des Musikstückes zu genau einer Segmentklasse führt im Falle hierarchischer Strukturen – wie sie für die Musik typisch sind und die in den Scape-Plots illustriert werden – zu mehrdeutigen Situationen, für die keine allgemein gültigen Lösungen formuliert werden können. Betrachten wir etwa das in Abbildung 2.1 illustrierte Beispiel des *Pomp and Circumstance March No. 4* von Edward Elgar: Das in der wiederholungsbasierten Grobstruktur (Teil b) mit A bezeichnete Segment ist in der als Feinstruktur bezeichneten Segmentierung (Teil c) in fünf Untersegmente zerlegt, die man etwa mit $a\,a\,b\,b\,a$ bezeichnen könnte. Andererseits treten die Grobsegmente A und B immer direkt hintereinander auf, sodass man diese auch zu einem gemeinsamen Segment zusammenfassen könnte, das in der musikalischen Form den Begriff »March« trägt (Abbildungsteil a).

Mathematisch entspricht die Herleitung einer Strukturierung dem Versuch, die musikalische Struktur durch eine Abbildung zu beschreiben, welche jedem Zeitpunkt des Musikstückes eine einzige, sinnvoll gewählte Segmentbezeichnung zuordnet. Für die Konstruktion dieser Abbildung mittels der $\sim_W$- bzw. $\sim_H$-Äquivalenz auf den Teilmengen der Merkmalsfolge f dürfen wir folglich statt aller Teilfolgen nur solche betrachten, die sich nicht überlappen. Dementsprechend suchen wir eine Partition[10] der Zeitachse $[1:T]$ des Musikstückes in disjunkte Teilintervalle $I_1,\ldots,I_M$, also eine Zerlegung $[1:T]=\bigsqcup_{m=1}^{M} I_m$. Hierbei gehen wir davon aus, dass die Intervalle in sortierter Reihenfolge vorliegen, d. h. wenn für zwei Zeitpunkte $t_1<t_2$ gilt: $t_1\in I_k, t_2\in I_m$, so ist $k\leq m$.

Die konkrete Wahl einer geeigneten Partition hängt dabei nicht nur vom ausgewählten Stück und von der gewählten Merkmalsdarstellung ab, sondern auch von weiteren aufgabenspezifischen Parametern, deren Schätzung einen wesentlichen Teil der Strukturierungsproblematik darstellt. So sind wir stets daran interessiert, dass die Segmente dieser Partition in möglichst wenig Äquivalenzklassen fallen, nicht zu kurz sind und die für uns relevanten Strukturen so gut wie möglich beschreiben. Im wiederholungsbasierten Fall werden einige dieser Bedingungen an sinnvolle Segmente etwa in [124] formuliert, um ein als *Thumbnail* bezeichnetes Segment zu finden, welches das im Laufe des Stückes am häufigsten vorkommende musikalische Material beinhaltet. Diese Abhängigkeit von der konkreten Problemstellung führt zu einer gewissen Beliebigkeit in der Auswahl dieser Partition, wodurch der Vergleich zweier Strukturierungen eines Stückes eine nichttriviale Aufgabenstellung darstellt, was wir auch im folgenden Abschnitt 2.7 näher untersuchen werden.

Für die folgenden Definitionen und Notationen nehmen wir an, wir hätten eine geeignete Partition $(I_m)_{m\in[1:M]}$ der Zeitachse $[1:T]$ gefunden, deren Teilintervalle in sortierter Reihenfolge vorliegen. Dies induziert eine Partition in Teilfolgen von f, die wir als *Segmentierung* von f oder f-Segmentierung $S=(\sigma_1,\ldots,\sigma_M)$ bezeichnen, wobei $\sigma_m := f_{I_m}$ das m-te Segment beschreibt. Die *Länge* dieses Segments bezeichnen wir mit $N_m := |I_m|$. Weiterhin können wir eine monoton steigende Folge der *Segmentgrenzen* $\mathcal{B}=(\mathcal{B}_1,\ldots,\mathcal{B}_{M+1})$ mittels $\mathcal{B}_m := I_m(1)-1$ für $m\in[1:M]$ und $\mathcal{B}_{M+1} := T$ angeben, wobei nach unserer Konvention $\mathcal{B}_m$ der letzten Zeit-

[10] Teilmengen $S_1,\ldots,S_M$ der Menge S bilden eine *Partition* von S, wenn $S=S_1\cup\cdots\cup S_M$ und die S_m paarweise disjunkt sind.

punkt *vor* dem m-ten Segment ist. Eine äquivalente Beschreibung direkt aus der Folge der Segmentlängen hat die Form

$$\mathcal{B}_m := \sum_{i=1}^{m-1} N_i \quad \text{für} \quad m \in [1 : M+1] .$$

Ausgehend von den Segmentgrenzen können wir das m-te Segment ($m \in [1 : M]$) auch durch $\sigma_m = f_{\mathcal{B}_m+1:\mathcal{B}_{m+1}}$ ausdrücken. Zu vorgegebener Mindestlänge $\theta \in \mathbb{N}$ heißt S eine θ-*Segmentierung*, wenn $N_m \geq \theta$ für alle $m \in [1 : M]$ gilt, d. h. alle Segmente weisen mindestens die Länge θ auf.

Eine *segmentweise Benennungsfunktion* oder kurz *Segmentbenennung* (engl. *segment labelling*) zu einer f-Segmentierung S ist eine Funktion $\bar{S} : [1 : M] \to \mathcal{L}$, die jedem Segmentindex eine Bezeichnung zuordnet. Mittels dieser Segmentbenennung und der Folge $\mathcal{B}$ der Segmentgrenzen können wir nun auch eine *punktweise Benennung* bzw. *punktweise Strukturierung* $\dot{S} : [1 : T] \to \mathcal{L}$ angeben, die jedem Zeitpunkt t des Musikstückes eine Segmentbezeichnung zuordnet. Diese Abbildung wird definiert als

$$\dot{S}(t) := \bar{S}(m) \quad \text{für } t \in [\mathcal{B}_m + 1 : \mathcal{B}_{m+1}] .$$

Bei dieser Konstruktion der punktweisen Benennungsfunktion wird deutlich, dass diese nur von den Segmentbenennungen und Segmentlängen abhängt. Daher können wir auf diese Weise auch ohne Vorliegen einer konkreten Merkmalsdarstellung über Strukturierungen und Segmente sprechen. Diese Herangehensweise werden wir in Abschnitt 3.4 zur Beschreibung von abstrakten Strukturierungen ohne ein zugrundeliegendes reales Musiksignal bzw. eine Merkmalsdarstellung desselben verwenden.

Umgekehrt können – zumindest im Fall der homogenitätsbasierten Strukturierung – aus der punktweisen Segmentierung auch Segmentgrenzen und -längen rekonstruiert werden. Hierzu betrachten wir die Menge der Zeitpunkte, an denen sich die punktweise Benennung ändert:

$$\mathcal{B} := \{0\} \cup \{t \in [1 : T-1] \mid \dot{S}(t) \neq \dot{S}(t+1)\} \cup \{T\}$$

Die Menge M der Segmente beträgt dann $|\mathcal{B}| - 1$. Die segmentweise Benennungsfunktion ergibt sich direkt als Zuordnung der Indizes der Segmente zu den Bezeichnungen eines der in ihnen enthaltenen Zeitpunkte: $\bar{S}(m) := \dot{S}(\mathcal{B}_{m+1})$.

Abschließend definieren wir zwei *Zulässigkeitsbedingungen* für Segmentierungen: Bei wiederholungsbasierten θ-Segmentierungen dürfen Teilfolgen mit Mindestlänge θ nicht in einem anderen Kontext vorkommen, ansonsten würden diese jeweils ein eigenständiges Segment bilden. Es gilt also für zwei Teilfolgen $\sigma_i, \sigma_j \in S$: Existieren kleinere Teilfolgen $\rho \sqsubset \sigma_i$ und $\rho' \sqsubset \sigma_j$ mit $\rho \sim_W \rho'$, so ist entweder bereits $\sigma_i \sim_W \sigma_j$ oder die Länge von ρ bzw. ρ' ist kleiner als θ.

Weiterhin stellen wir eine Maximalitätsforderung an die Strukturierung, dass es zu jedem

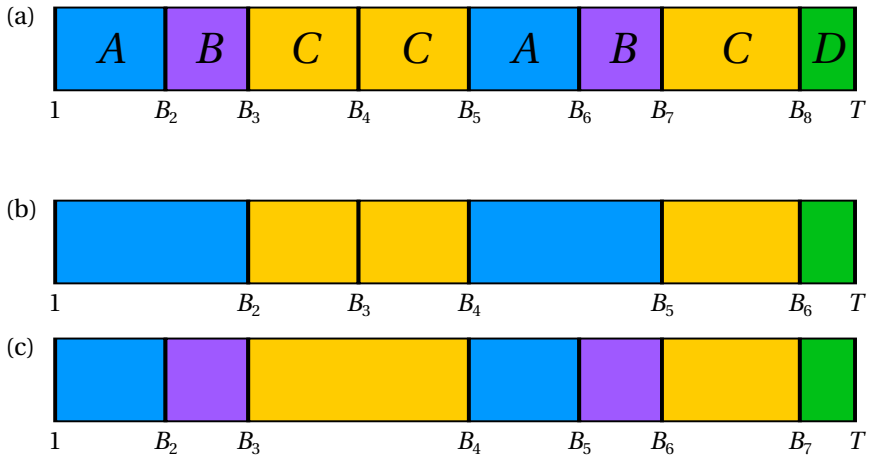


Abbildung 2.8.: Effekte der Zulässigkeitsbedingungen für Segmentierungen. Die Farbe der Segmente zeigt die Segmentklassen-Zugehörigkeit an. Hier gilt $T \approx 300\,\text{s}$, $\theta \approx 20\,\text{s}$. **(a)** Manuelle Strukturierung, vgl. Abbildung 2.1b. **(b)** Rein wiederholungsbasierte Segmentierung. **(c)** Rein homogenitätsbasierte Segmentierung für den Fall, dass die verschiedenen Segmentklassen hinreichend heterogen sind.

Segmentpaar (σ_i, σ_{i+1}) mit $\sigma_i \not\sim_W \sigma_{i+1}$ einen Index k gibt, sodass entweder $\sigma_i \sim_W \sigma_k$ und $\sigma_{i+1} \not\sim_W \sigma_{k+1}$ oder $\sigma_i \not\sim_W \sigma_k$ und $\sigma_{i+1} \sim_W \sigma_{k+1}$ gilt. Dies modelliert den Fall, dass zwei verschiedene, ausschließlich paarweise auftretende Segmente mittels wiederholungsbasierter Segmentierung nicht getrennt werden können. Für eine Illustration dieser Bedingung siehe Abbildung 2.8b.

Für homogenitätsbasierte Segmentierungen gilt die folgende Zulässigkeitsbedingung: Zwei benachbarte Segmente dürfen nicht äquivalent sein, ansonsten würden beide Segmente zu einem Segment zusammengefasst werden. Es gilt also für zwei Teilfolgen $\sigma_i, \sigma_j \in S$: Wenn $\sigma_i \sim_H \sigma_j$, so ist $|i - j| \neq 1$, vgl. Abbildung 2.8c.

Bei beiden Strukturierungsprinzipien sind wir nur an zulässigen Segmentierungen interessiert.

2.6. Homogenitätsbasierte Strukturierung

Mit dem Begriff der homogenitätsbasierten Strukturierung beschreiben wir die Unterteilung eines Musikstückes in homogene Segmente bezüglich eines speziellen musikalischen Aspekts. Musikalisch gesehen entspricht ein solcher homogener Bereich dem Gegenteil des überraschend eintretenden oder kontrastierenden Elements, er kann daher sowohl auf eine Wiederholung als auch auf eine leichte Variation des betrachteten musikalischen Aspekts hinweisen. Im Gegensatz zur wiederholungsbasierten Strukturierung steht der Homogenität allerdings kein musikalisches Gestaltungsmittel direkt gegenüber, weswegen eine rein homo-

genitätsbasierte Strukturierung nur in Ausnahmefällen geeignet ist, die musikalische Struktur eines Musikstückes adäquat zu beschreiben.

Stattdessen kann eine homogenitätsbasierte Strukturierung zur Gewinnung zusätzlicher Informationen etwa bei einer wiederholungsbasierten Segmentierung verwendet werden, zum Beispiel bei der Ermittlung von Variationen in der Instrumentierung bei einem mehrfach wiederholten Segment. So ist etwa bei Popsongs eine instrumentale Version einer Strophe ohne Gesang ein übliches Variationsmittel, welches bei einer rein wiederholungsbasierten Strukturierung nicht von den anderen Strophen unterschieden werden kann. Andererseits erlaubt eine rein homogenitätsbasierte Segmentierung die Unterscheidung zwischen Strophe und Refrain nicht, sofern diese ähnlich in Harmonik, Klangfarbe und Rhythmus sind oder insgesamt inhomogen bezüglich dieser Aspekte sind.

Wie bereits in Abschnitt 2.4 beschrieben, entspricht dieses Prinzip auf der technischen Seite dem Auffinden blockähnlicher Strukturen in einer Selbstähnlichkeitsmatrix eines Musikstückes. Damit diese Blöcke für eine automatische Analyse hinreichend stark ausgeprägt sind, ist die Verwendung von Merkmalen auf einer verhältnismäßig groben zeitlichen Auflösung nötig, da diese großen Einfluss auf die Struktur der betrachteten Selbstähnlichkeitsmatrizen hat.

In Abbildung 2.9 sind vier dieser Matrizen für verschiedene Auflösungsstufen von harmoniebasierten Chroma-Merkmalen abgebildet. Während bei einer feinen Auflösung von 1 Sekunde (a) nahezu ausschließlich Pfadstrukturen erkennbar sind, werden bei einer mittleren Auflösung von 4 Sekunden (b) allmählich erste Blockstrukturen erkennbar. Setzt man den Vergröberungsprozess der Auflösung fort, so sind bei 12 Sekunden die Blockstrukturen deutlich erkennbar (siehe etwa die ersten 60 Sekunden bei Abbildung 2.9c, die perfekt zur wiederholungsbasierten Feinstruktur korrespondieren) und bei einer sehr groben Auflösung von 22,5 Sekunden verbleiben nur noch große Blöcke, wohingegen viele kleinere Blöcke in großen pfadähnlichen Strukturen aufgegangen sind.

Neben diesen harmoniebasierten Merkmalen sind auch Klangfarben- sowie Tempomerkmale für eine homogenitätsbasierte Segmentierung geeignet, sofern diese musikalischen Aspekte bei dem jeweils betrachteten Stück in längeren Passagen unverändert bleiben. In [182] wird anhand einiger Stücke eine konkrete Analyse vorgenommen, welche Merkmale zur homogenitätsbasierten Beschreibung jedes einzelnen Segments geeignet sind. Hierbei hat sich nicht nur herausgestellt, dass innerhalb eines Stückes der relevante musikalische Aspekt für verschiedene Bereiche verschieden ist, sondern auch, dass für etliche Segmente kein Merkmal allein eine hinreichende Diskriminativität aufweist. Zur Analyse wird dabei ein Maß für die Blockähnlichkeit einer Passage verwendet, wie es auch in [158] vorgeschlagen wird.

In der bisherigen Forschung wurden zahlreiche Methoden für die Erkennung dieser Blockstrukturen vorgestellt. Darunter zählen Novelty-basierte Verfahren wie die Verwendung einer schachbrettartigen Schablone, die lokal mit der Diagonalen der Selbstähnlichkeitsmatrix verglichen wird, Clustering-Verfahren wie Hidden Markov Models oder dynamische Pro-

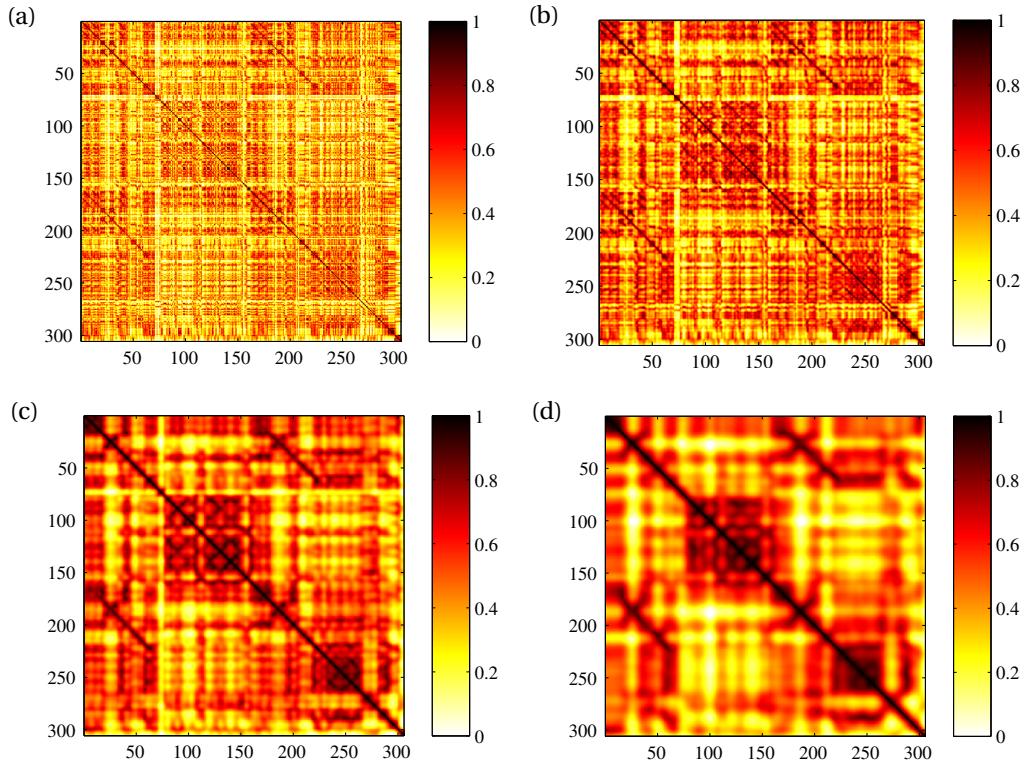


Abbildung 2.9.: Anwendung von Glättungsverfahren zur Verstärkung von Blockstrukturen in Chroma-Selbstähnlichkeitsmatrizen unseres Elgar-Beispiels bei verschiedenen Merkmalsauflösungen von **(a)** 1 Sekunde, **(b)** 4 Sekunden, **(c)** 12 Sekunden, **(d)** 22,5 Sekunden (vgl. [148]).

grammierung, aber auch rein homogenitätsbasierte Verfahren wie das später ausführlich vorgestellte Verfahren der nicht-negativen Matrixfaktorisierung.

Bei dem von *Jonathan Foote* in [48] vorgestellten Verfahren wird ein schachbrettartiges Muster einer gewissen Größe sukzessive über die Diagonale der Selbstähnlichkeitsmatrix bewegt und für jeden Punkt die Ähnlichkeit dieses als Kern bezeichneten Musters mit dem darunterliegenden Ausschnitt der Matrix verglichen. Die so berechneten Ähnlichkeitswerte bilden eine zeitabhängige Funktion (vgl. Abbildung 2.10b und d), bei der hohe Werte an denjenigen Zeitpunkten auftreten, bei denen – bei Vorliegen ausreichend diskriminativer Blockstrukturen – mit hoher Wahrscheinlichkeit ein Übergang von einem homogenen Segment zu einem anderen stattfindet. Somit wird mittels dieser Technik eine ursprünglich homogenitätsbasierte Problemstellung in eine Novelty-basierte Aufgabe überführt. Die Kandidaten für die Segmentgrenzen werden anschließend durch Verwendung von Verfahren zum Auffinden markanter lokaler Maximalstellen (*peak picking*) ermittelt.

Die Schwierigkeiten dieses Ansatzes liegen in dem hohen Einfluss der Größe des verwendeten

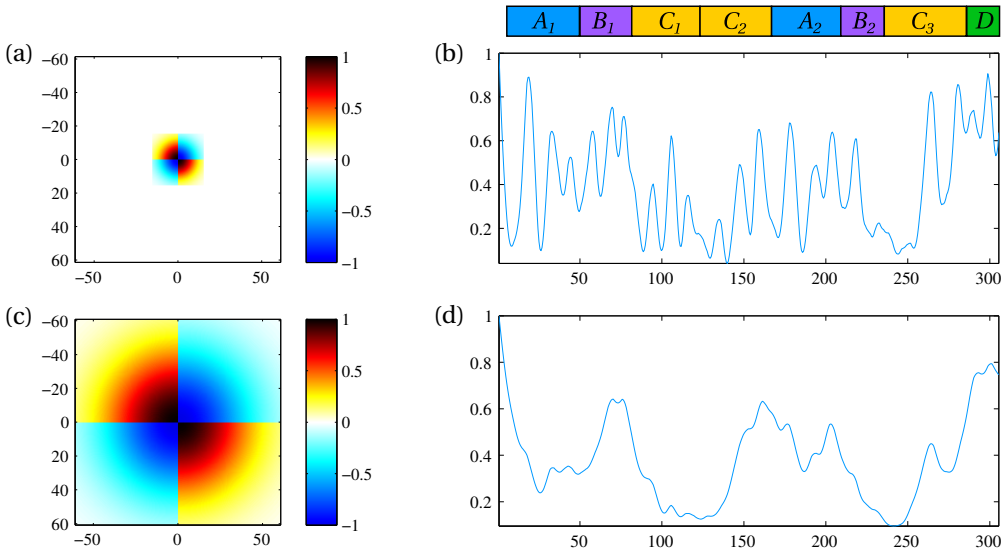


Abbildung 2.10.: Verwendung des Schachbrett-Kernel-Ansatzes zur Erkennung der Grenzen homogener Segmente einer Selbstähnlichkeitsmatrix mit Merkmalsauflösung von 12 s (dargestellt in Abbildung 2.9c). **(a)** und **(b)** Kernel mit einer Größe von 30 s mit daraus resultierender Novelty-Kurve für die Segmentgrenzen, **(c)** und **(d)** Kernel mit einer Größe von 120 s mit entsprechender Novelty-Kurve.

Schachbrett-Kerns auf die berechnete Noveltykurve. Auch die Modifikation des Kerns durch punktweise Multiplikation mit einer 2D-Gaußglocke wie bei den beiden in Abbildung 2.10 gezeigten Kernen kann diesen Effekt nicht wesentlich abmildern. Im Abbildungsteil d wird illustriert, dass auch bei einer sinnvoll gewählten Größe des Kerns leidlich einige Segmentgrenzen gefunden werden können. Weiterhin liefert diese Methode bei erfolgreicher Anwendung lediglich die Grenzen der Segmente und noch keine Zuordnung im Sinne einer Strukturierung. Daher wurde dieses Verfahren später mit zusätzlichen Clustering-Techniken versehen [49].

Der in [3] vorgestellte Ansatz basiert auf einem Hidden Markov Model (HMM), das Matrixbereiche mit gleichbleibenden statistischen Eigenschaften (sogenannte Texturen) erkennt. Es hat sich herausgestellt, dass diese zu musikalisch sinnvollen Strukturen wie Strophen- und Refrainpassagen korrespondieren. In [100] wird jedem Zeitpunkt ein (kontextloser) Zustand zugeordnet; diese werden anschließend mittels Anwendung eines HMM-Ansatzes zu Segmenten zusammengefasst. Im Gegensatz zur expliziten Bestimmung musikalisch motivierter Merkmale werden hier spektrale Hüllenmerkmale für die Bestimmung der Zustände verwendet und die Eigenschaften von musikalischen Segmenten werden von dem Verfahren durch maschinelles Training eigenständig ermittelt.

Bereits in [161] wurde ein anderes HMM-basiertes Verfahren zur Echtzeitsynchronisierung monophoner Audiodaten mit einer Notendarstellung vorgestellt. Die Kernidee dieses Verfah-

rens ist ebenfalls ein Segmentierungsansatz, bei dem die Segmente den einzelnen Noten bzw. Pausen entsprechen.

Eine andere Herangehensweise mittels dynamischer Programmierung wird beispielsweise in [57, 76] vorgestellt. Hierbei werden eine Selbstähnlichkeitsmatrix durchquerende Pfade betrachtet, deren Eigenschaften innerhalb einer Blockstruktur ein anderes Verhalten aufweisen als beim Übergang in eine andere Blockstruktur. Die Segmentierung wird anschließend aus diesen Übergangspositionen abgeleitet.

Ein weiterer Ansatz zur homogenitätsbasierten Strukturierung von Musikstücken stellt die nicht-negative Matrixfaktorisierung dar, welche ein Verfahren zur Komplexitätsreduktion von Matrizen mit blockähnlichen Strukturen darstellt. Diese Methode wollen wir im Folgenden näher diskutieren.

2.6.1. Nicht-negative Matrixfaktorisierung

Mit dem Begriff *Nicht-negative Matrixfaktorisierung* (engl. *Non-negative matrix factorization*, NMF) wird eine Reihe verwandter Verfahren zur Approximation einer Matrix mit nichtnegativen Einträgen durch das Produkt zweier kleinerer Matrizen beschrieben, die ebenfalls ausschließlich nichtnegative Einträge aufweisen [92]. Aufgrund seiner komplexitätsreduzierenden Eigenschaften wird es auf vielfältige Problemstellungen angewendet, um aus den Daten nicht direkt ersichtliche, sogenannte verborgene Komponenten (engl. *latent components*) zu ermitteln. Auch im *Music Information Retrieval* wird es erfolgreich zur Erkennung von musikalischen Strukturen in Audiodaten und zur Segmentierung eingesetzt [62, 79, 133].

Im Folgenden geben wir eine knappe Einführung. Sei $V \in \mathbb{R}_{\geq 0}^{m \times n}$ eine Matrix mit nichtnegativen Einträgen und $0 < k \ll \min(m, n)$ ein vorgegebener ganzzahliger Parameter. Gesucht werden Matrizen $W \in \mathbb{R}_{\geq 0}^{m \times k}$ und $H \in \mathbb{R}_{\geq 0}^{k \times n}$, sodass ihr Matrizenprodukt $W \cdot H$ eine möglichst gute Approximation an V darstellt, also eine gute Repräsentation einer im Allgemeinen hochdimensionalen Matrix durch ein Produkt zweier niederdimensionaler Matrizen, das dadurch einen vergleichsweise niedrigen Rang aufweist. Für das Maß dieser Approximation wird in vielen Fällen die Frobeniusnorm[11] $\| \bullet \|_2$ verwendet. Somit lässt sich das NMF-Problem auch als Minimierungsproblem formulieren:

$$\text{minimiere} \quad \|V - WH\|_2 \quad \text{mit} \quad W, H \geq 0\,. \tag{2.1}$$

Für die Berechnung einer optimalen lokalen Lösung werden hierfür Iterationsverfahren verwendet, bei denen die Matrizen W und H zunächst mit zufälligen oder mittels eines Vorverarbeitungsverfahrens berechneten Werten initialisiert und anschließend mittels multiplikativer Update-Regeln schrittweise optimiert werden, bis eine lokal optimale Lösung gefunden wird. Eine wesentliche Eigenschaft dieser Herangehensweise ist die Tatsache, dass Matrixeinträge,

[11] Für eine Matrix $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ ist die Frobeniusnorm definiert als $\|A\|_2 := \sqrt{\sum_{i,j} (a_{ij})^2}$.

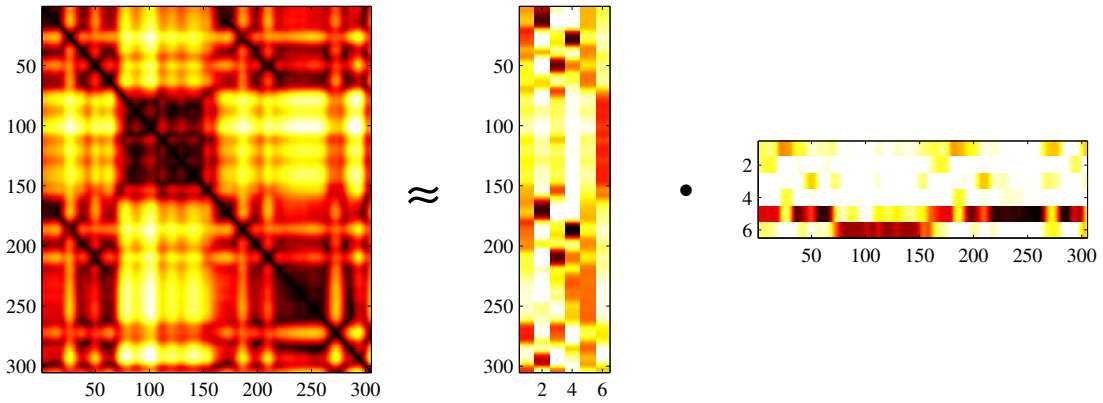


Abbildung 2.11.: Verwendung von *sparse NMF* zur Strukturerkennung einer Selbstähnlichkeitsmatrix mit Blockstrukturen. Der rechte Faktor kann direkt als Folge abstrakter Strukturmerkmale interpretiert werden.

die in einem beliebigen Iterationsschritt den Wert 0 angenommen haben, diesen in allen folgenden Schritten beibehalten. Die in [93] vorgestellten Update-Regeln lauten

$$H \leftarrow H \odot \frac{W^\top V}{W^\top W H},$$
$$W \leftarrow W \odot \frac{V H^\top}{W H H^\top},$$

wobei mit $\odot$ das punktweise Produkt (Hadamard-Produkt) zweier Matrizen bezeichnet wird. Eine knappe und übersichtliche Herleitung dieser Regeln ist in [186] zu finden.

Bereits in [93] wird eine semantische Interpretation dieser Matrizen vorgestellt: Geht man davon aus, dass die Spalten der Matrix V die Resultate eines mehrwertigen Messvorgangs beschreiben, so kann die Matrix W als eine Menge von Basisvektoren oder Schablonen (engl. *templates*) aufgefasst werden, deren Koeffizienten oder Aktivierungen (*activation*) durch die Matrix H angegeben werden. Diese Interpretation kann auf vielfältige Probleme angewendet werden, wodurch NMF zu einer weitverbreiteten Technik zur Erkennung verborgener Komponenten geworden ist.

Im Bereich des *Music Information Retrieval* wird NMF nicht nur für Segmentierungsaufgaben verwendet, sondern findet auch Anwendung beispielsweise bei Quellentrennungsverfahren von Audiodaten [40, 46, 169, 178, 197]. Auch im Bereich des »klassischen« text-basierten *Information Retrieval* ist NMF ein geeignetes Werkzeug zum Clustern von Dokumenten [211] und zum Auffinden verborgener Zusammenhänge [193]. In [24] werden eine Reihe von Erweiterungen und vielfältige Anwendungen dieses Verfahrens beschrieben.

Eine für uns relevante Weiterentwicklung dieses Verfahrens stellt eine zusätzliche Bedingung an die Dünnbesetztheit der Matrix H (engl. *sparsity constraint*) und wird daher auch als *sparse*

NMF (sNMF) bezeichnet [83], siehe auch Abbildung 2.11. Dabei wird das in Gleichung 2.1 formulierte Minimierungsproblem ersetzt durch

$$\text{minimiere} \quad \tfrac{1}{2}\|V - WH\|_2^2 + \alpha\|W\|_2^2 + \beta\left(\sum_{j=1}^{n}\|H_{\bullet,j}\|_1^2\right) \quad \text{mit} \quad W, H \geq 0\,. \tag{2.2}$$

wobei $\|\bullet\|_1$ die Summennorm[12] eines Vektors bezeichnet. Somit kann über die Parameter α und β eine Gewichtung zwischen dem Einfluss der Approximationsgüte, der Regularität (d. h. der Frobeniusnorm) von W und der Dünnbesetztheit von H vorgenommen werden. In der von den Autoren von [83] angebotenen MATLAB-Implementierung[13] dieses Verfahrens sind die Parameter α und β standardmäßig auf den Mittelwert aller Einträge der Matrix V gesetzt. Wie viele NMF-Algorithmen, darunter auch die einfachen Update-Regeln aus [93], basiert dieses Verfahren auf einem alternierenden nichtnegative kleinste Quadrate-Ansatz (*alternating nonnegative least squares*, ANLS), d. h. dem abwechselnden Optimieren der einzelnen Faktoren. Für diese Optimierung wird in [83] ein sogenannter *Block Principal Pivoting*-Algorithmus verwendet, der die Spalten der Matrix vor dem eigentlichen Minimierungsschritt nach Ähnlichkeit sortiert und in Clustern zusammenfasst, wodurch die benötige Rechenzeit für die Optimierungsschritte deutlich gesenkt wird.

Für die Anwendung auf die Segmentierung musikalischer Informationen stellt die Dünnbesetztheit der Aktivitätsmatrix H eine wesentliche Vereinfachung gegenüber dem in [79] vorgestellten Verfahren dar, welches das ursprüngliche NMF-Verfahren zur homogenitätsbasierten Segmentierung verwendet, wodurch ein zusätzlicher Gruppierungs- bzw. Clustering-Schritt notwendig wird. Im Gegensatz hierzu kann bei Verwendung von sNMF in vielen Fällen eine sinnvolle Segmentierung direkt aus der Matrix H abgelesen werden. Weiterhin wird bei sNMF auch sichergestellt, dass der Parameter k nur eine obere Schranke für die Anzahl der verwendeten Cluster darstellt und nicht etwa eine Aufteilung in k Cluster erzwingt. Die in Abbildung 2.11 gezeigte Aktivierungsmatrix weist beispielsweise in den Clustern 1 und 3 kaum Energie auf.

Eine sNMF-Variante wurde in [200] in einer modifizierten Form zur wiederholungsbasierten Strukturierung von Musikstücken ohne Notwendigkeit der Berechnung einer Selbstähnlichkeitsmatrix verwendet. Eine weitere Anwendung wurde in [75] zur Bestimmung lokaler Tonarten in einem Musikstück vorgestellt, was wir in Abschnitt 4.3 genauer diskutieren wollen.

2.7. Evaluation von Strukturierungsverfahren

Eine einfache Möglichkeit, die Qualität verschiedener Segmentierungsmethoden vergleichen zu können, stellt die *automatisierte Evaluation* auf standardisierten Datensätzen dar. Ein solcher Datensatz besteht dabei aus einer Menge von Audio-Dateien mit dazugehörigen, manuell erstellten Referenz-Annotationen.

[12] Für einen Vektor $v \in \mathbb{R}^m$ ist die Summennorm definiert als $\sum_{i=1}^{m}|v_i|$.
[13] `http://www.cc.gatech.edu/~hpark/nmfsoftware.php`

Der erste größere Datensatz, der explizit für die wissenschaftliche Analyse musikalischer Informationen zusammengestellt wurde, ist der *RWC*[14]-Datensatz des japanischen *National Institute of Advanced Industrial Science and Technology* (AIST). Dieser besteht aus 115 Pop-Songs, je 50 klassischen und Jazz-Stücken, 100 Stücken zur Bestimmung verschiedener Genres sowie Einzelaufnahmen von Solo-Instrumenten. Das AIST stellt ebenfalls Referenzannotationen zur Verfügung; diese umfassen Taktstrukturen, extrahierte Melodiestimmen, Positionen der Refrains sowie für die Pop-Songs synchronisierte MIDI-Dateien.

Für die 180 Songs der 12 Studio-Alben der britischen Rockband *The Beatles* wurden im Zeitraum zwischen 1989 bis 2001 vom Musikwissenschaftler *Alan W. Pollack* Strukturannotationen angefertigt[15]. Diese wurden anschließend von einer Arbeitsgruppe der *Universität Pompeu Fabra* in Barcelona mit Zeitinformationen versehen und später an der *Technischen Universität Tampere* (Finnland) einer Revision unterzogen und in [146] veröffentlicht. Für weitere Details zur diesem Datensatz siehe auch [144, S. 48].

Der in [108] vorgestellte *Isophonics*-Datensatz[16] des *Centre for Digital Music* der britischen *Queen Mary University of London* verfügt über 301 Stücke aus dem populärmusikalischen Bereich (die oben genannten 180 Songs der Beatles, 38 des Sängers *Michael Jackson*, 51 der Rockband *Queen*, 14 der Sängerin *Carole King* sowie 18 der deutschen Amateurband *Zweieck*) und stellt pro Stück neben anderen Analysen zu Akkorden und Tonarten auch eine strukturelle Referenzannotation zur Verfügung. Die Strukturbeschreibungen für die Beatles-Songs wurden dabei noch einmal überarbeitet.

Ein weiterer Datensatz von Strukturannotationen besteht ausschließlich aus Aufnahmen der 49 Mazurka-Tänze von Frédéric Chopin[17]. Diese Sammlung wurde vom *Mazurka Project*[18] am *Research Centre for the History and Analysis of Recorded Music (CHARM)* in London zusammengestellt und besteht aus insgesamt 2792 einzelnen Audioaufnahmen. Für jedes der 49 Stücke wurde die musikalische Struktur zuerst manuell beschrieben und anschließend mittels automatischer Synchronisation über eine MIDI-Repräsentation des Notentextes auf die Einzelaufnahmen übertragen [67, 174].

Im *SALAMI*[19]-Projekt [183] wurden mehrere Strukturannotationen für einen großen Datensatz von über 1600 Stücken aus verschiedenen musikalischen Genres von Barockmusik über Jazz, Rock und Pop bis hin zu traditioneller Musik aus außereuropäischen Kulturkreisen (*world music*) angefertigt. Davon sind die Hälfte der Annotationen öffentlich zugänglich, die andere

[14] *Real World Computing*, `https://staff.aist.go.jp/m.goto/RWC-MDB/`

[15] Diese Annotationen sind mit vielen Zusatzinformationen zu den einzelnen Stücken in der Rubrik »Alan W. Pollacks ›Notes On‹ series« unter `http://www.recmusicbeatles.com/` abrufbar.

[16] `http://www.isophonics.net/datasets`

[17] Opuszahlen 6, 7, 17, 24, 30, 33, 41, 50, 56, 59, 63, 67 und 68.

[18] `http://mazurka.org.uk`

[19] *Structural Analysis of Large Amounts of Music Information*, `http://ddmal.music.mcgill.ca/research/salami/`

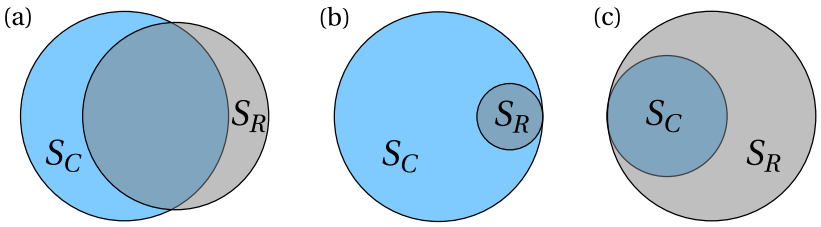


Abbildung 2.12.: Illustration der Evaluationsmaße Precision und Recall als Venn-Diagramm für **(a)** $|S_C| = 10$, $|S_R| = 8$, $|S_C \cap S_R| = 6$, **(b)** $|S_C| = 10$, $|S_R| = 1$, $|S_C \cap S_R| = 1$, **(c)** $|S_C| = 10$, $|S_R| = 30$, $|S_C \cap S_R| = 10$.

Hälfte wird zur automatischen Evaluation im Rahmen des MIREX[20]-Programms verwendet. Dieser Datensatz enthält pro Stück bis zu vier verschiedene Referenzannotationen, da zum einen dasselbe Stück von bis zu zwei Personen analysiert wurde und weiterhin jede Person das Stück einmal nach Grob- und einmal nach Feinstruktur unterteilte. In [44] wird eine Evaluation im Rahmen von MIREX von Algorithmen zur Strukturerkennung auf diesem Datensatz vorgestellt.

Die automatisierte Auswertung bedient sich der aus dem *Information Retrieval* bekannten Evaluationsmaße *Precision* $\boldsymbol{P}$, *Recall* $\boldsymbol{R}$ (auch Sensitivität genannt) und *F-measure* $\boldsymbol{F}$. Im »klassischen« *Information Retrieval* werden diese Maße zur Beschreibung der Güte von Suchalgorithmen verwendet, bei denen wir von einer Menge S_C korrekter Ergebnisse (*set of correct results*) und einer Menge S_R durch das Verfahren gefundener Ergebnisse (*set of retrieved results*) ausgehen. Precision beschreibt den Anteil korrekter Ergebnisse in der Menge der gefundenen Ergebnisse, Recall den Anteil gefundener Ergebnisse in der Menge der korrekten Ergebnisse, und als F-measure wird das harmonische Mittel der beiden Maße bezeichnet:

$$\boldsymbol{P} := \frac{|S_C \cap S_R|}{|S_R|}\,, \qquad \boldsymbol{R} := \frac{|S_C \cap S_R|}{|S_C|}\,, \qquad \boldsymbol{F} := \frac{2 \cdot \boldsymbol{P} \cdot \boldsymbol{R}}{\boldsymbol{P} + \boldsymbol{R}}\,. \tag{2.3}$$

Als Beispiel für diese Maße betrachten wir die Segmentgrenzen eines Musikstückes. Wir nehmen beispielsweise an, dass in einer Referenzannotation 10 Segmentgrenzen angegeben sind und dass ein Algorithmus 8 Grenzen findet, von denen 6 mit der Referenzannotation übereinstimmen[21]. Wir erhalten für die Evaluationsmaße die folgenden Werte: $\boldsymbol{P} = 6/8 = 0{,}75$; $\boldsymbol{R} = 6/10 = 0{,}6$; $\boldsymbol{F} = 2/3 \approx 0{,}67$, siehe Abbildung 2.12a. Die Wichtigkeit des F-measure für die Evaluation ergibt sich aus der Betrachtung der Extremfälle: Angenommen, ein Algorithmus gibt nur eine Segmentgrenze aus, dann nimmt die Precision zwar ihren Maximalwert von 1

[20] *Music Information Retrieval Evaluation eXchange* bezeichnet eine gemeinsame Evaluationsumgebung der MIR-Gemeinschaft zur standardisierten Auswertung von MIR-Systemen und Algorithmen betrieben von der US-amerikanischen *University of Illinois* [38, 39].

[21] In der Praxis erlauben wir hier eine gewisse Toleranz, die üblicherweise 3 oder 0,5 Sekunden beträgt. Man beachte, dass bei der Verwendung von Toleranzen die Abstände sowohl der annotierten als auch der detektierten Segmentgrenzen größer als die vorgegebene Toleranz sein müssen.

an, der Recall hingegen liegt nur bei 0,1 (Abbildung 2.12b). Wird umgekehrt alle 3 s eine Segmentgrenze ausgegeben, so ist zwar $\boldsymbol{R} = 1$, allerdings nimmt hier die Precision einen sehr niedrigen Wert an (Abbildung 2.12c). Das F-measure beträgt bei diesem Beispiel in beiden Fällen (eine gewisse Mindestlänge des Stückes vorausgesetzt) weniger als 0,2. Diese beiden Maße Precision und Recall wirken also entgegengesetzt, und das F-measure belohnt diejenigen Systeme, welche die beste Balance von Precision und Recall erreichen. Zu einer etwas ausführlicheren Diskussion eines ähnlichen Beispiels siehe [39].

Da wir bei der Auswertung von Strukturannotationen im Gegensatz zu Segmentgrenzen keine Menge »gefundener« und »korrekter Lösungen« vorliegen haben und im Allgemeinen auch die Bezeichnungen der Segmente nicht übereinstimmen müssen, wurde in [100] eine als »paarweise« (*pairwise*) bezeichnete Variante von Precision und Recall eingeführt. Hierbei vergleichen wir Paare von Zeitpunkten, die bei der automatisch erzeugten Strukturannotation mit derselben Segmentbezeichnung versehen worden sind, mit denjenigen Paaren, die in der manuell annotierten Referenzannotation dieselbe Bezeichnung tragen. Die Menge S_R der gefundenen Lösungen wird dabei ersetzt durch die Menge der vom auszuwertenden maschinellen Verfahren gleich benannten Paare von Zeitpunkten und die Menge S_C der korrekten Lösungen durch die in der Referenzannotation übereinstimmenden Paare. Die Berechnung der Maße erfolgt dann ebenfalls nach Gleichung 2.3.

Es gibt einige weitere populäre Evaluationsmaße wie etwa die in [104] vorgestellten Maße zur Ober- und Untersegmentierung, die allerdings stark entweder mit Precision, Recall oder dem F-Measure korrelieren [181].

Der einfachen Verwendbarkeit dieser automatischen Evaluation steht eine Reihe methodischer Schwächen gegenüber. Als erster Punkt ist sicherlich die dem SALAMI-Projekt zugrundeliegende Auffassung zu nennen, dass eine Referenzannotation nicht als angenommene absolute Wahrheit (engl. *ground truth*) angesehen werden kann. Vielmehr stellt der Vorgang einer musikalischen Segmentierung selbst eine interpretatorische und damit kreative Leistung dar, vgl. [81]. In Experimenten wurde weiterhin gezeigt, dass ein automatischer Vergleich der Annotationen zweier Personen im Schnitt nur ein F-measure von etwa 0,9 ergibt, siehe [146, 174]. Auf dem SALAMI-Datensatz haben wir die Strukturannotationen derjenigen 498 Stücke, für die Annotationen von zwei verschiedenen Personen vorliegen, ebenfalls gegeneinander ausgewertet. Dabei haben wir für das paarweise F-measure bei Betrachtung der Grobstruktur-Annotationen einen Durchschnittswert von 0,731 erhalten, das paarweise F-measure beträgt für die feingranulare Struktur sogar nur durchschnittlich 0,636. Für eine ausführliche Unterteilung dieser Ergebnisse nach musikalischen Stilrichtungen siehe Abschnitt A.1.

Jede Segmentierung ist eine Abwägung zwischen musikalischen Aspekten. Bei einem niedrigen Evaluationswert kann nicht unterschieden werden, ob sich die beiden Annotationen bezüglich des gewählten Aspekts unterscheiden oder die automatische Strukturierung zwar nach dem zutreffenden Aspekt erfolgt, aber technisch unzureichend ist. Somit bieten diese Evaluationsmaße für die Strukturanalyse wenig Anhaltspunkte zur Bewertung der Güte eines Algorithmus,

im Unterschied zu klarer definierten Aufgabenstellungen wie beispielsweise Tonartbestimmung (vgl. Abschnitt 4.3), Akkordanalysen oder Schätzung der Instrumentierung. In [79] erreicht auch die manuelle Segmentierung bezüglich der expliziten Merkmale Klangfarbe und Harmonik ausgewertet gegen eine vorgegebene Strukturannotation nur F-measure-Werte zwischen 0,76 und 0,8.

Weiterhin kann die durch den Vergleich der Segmentbezeichnungen berechnete Ähnlichkeit nur zwischen völliger Übereinstimmung und Abweichung unterscheiden. Sind in der Referenzannotation zwei Segmente beispielsweise mit AA' bezeichnet, so kann bei berechneten Lösungen AB oder AA nicht bestimmt werden, welche davon als korrekt anzusehen ist. Zur Problematik der Wahl solcher Segmentbezeichnungen siehe auch [9, 82].

Ebenfalls führt die Benennung von Segmenten nach ihren musikalischen Funktionen bei einer automatisierten Auswertung zu stark unterbestimmten Problemstellungen: Um eine funktionell annotierte Struktur wie im *Isophonics*-Datensatz gut rekonstruieren zu können, ist Zusatzwissen nötig; etwa dass das erste Segment oftmals als Einleitung (*intro*) und das letzte als Ausklang (*outro*) des Stückes bezeichnet wird, auch wenn es sich um Wiederholungen eines normalerweise anders bezeichneten Segments (z. B. Strophe bzw. Refrain) handeln. Werden nun die Evaluationsmaße zum Vergleich der Performance zweier wiederholungsbasierter Segmentierungsalgorithmen ohne funktionelles Vorwissen verwendet, so erzielt dasjenige Verfahren einen höheren Punktwert bei der Evaluation, welches beispielsweise das erste oder letzte Segment nicht als Wiederholung erkennt und damit die an es gerichteten Erwartungen (Auffinden aller Wiederholungen) schlechter erfüllt. Zur Analyse der verwendeten Bezeichnungen bei den zur automatischen Evaluation verwendeten Datensätzen siehe auch [180].

Für die Bewertung identischer *Segmentgrenzen* wird üblicherweise ein fester Toleranzbereich von 3 s verwendet, vgl. das Beispiel oben. Hier stellt sich die Frage nach der Willkürlichkeit, wenn eine Abweichung von 3 s noch als richtig, eine Abweichung von 3,1 s bereits als falsch gewertet wird, was auch in der MIR-Gemeinschaft kritisch diskutiert wird [134].

Abschließend kommen wir zu dem Ergebnis, dass ein hohes F-measure nicht unbedingt zu einer perzeptuell »sinnvollen« Segmentierung korrespondiert – und umgekehrt ein niedriger Wert nicht zwangsläufig zu einer unzureichend erkannten Struktur. Dadurch ist unserer Auffassung nach die Aussagekraft dieser und ähnlicher Evaluationsmaße zumindest als fragwürdig anzusehen. Dennoch werden wir mangels realistisch durchführbarer Alternativen[22] bei den im Laufe dieser Arbeit durchgeführten Auswertungen auf dieses Hilfsmittel zurückgreifen. Hierbei sind die oben aufgeführten grundsätzlichen Bedenken ob der Zulässigkeit und Aussagekraft der erzielten Werte stets zu berücksichtigen.

[22] Solche Alternativen wäre beispielsweise eine empirische Beurteilung von Segmentierungsresultaten durch Musiker, oder zumindest eine deutliche Vergrößerung der Menge an Referenzannotationen für ein Stück. Auch müsste die hierarchische Struktur eines Musikstückes deutlich stärker berücksichtigt werden, etwa durch Einführung eines Konfidenzmaßes für Segmentgrenzen oder Berücksichtigung alternativer Bezeichnungen für einzelne Segmente.

3. Konvertierung von Pfad- zu Blockstrukturen

Die musikalische Strukturanalyse basiert zu wesentlichen Teilen auf den beiden Prinzipien *Wiederholung* und *Homogenität* für die Zerlegung einer Audioaufnahme in musikalisch sinnvolle Strukturen. Bei der Konversion der Aufnahme in eine Selbstähnlichkeitsmatrix führen Wiederholungen typischerweise zu pfadähnlichen Strukturen, wohingegen homogene Bereiche durch Blockstrukturen repräsentiert werden.

In diesem Kapitel[1] führen wir eine neuartige Methode zur Konvertierung von Pfad- in Blockstrukturen ein, bei der eine Eigenwertzerlegung der Selbstähnlichkeitsmatrix mit sinnvollen Clustering-Verfahren kombiniert wird. Im Gegensatz zu vielen lokal operierenden Verfahren zur Wiederholungserkennung stellt dieses Methode einen globalen Ansatz zur Strukturerkennung dar. Die Effektivität dieses Konvertierungsansatzes zeigen wir durch die Möglichkeit, ursprünglich für homogenitätsbasierte Strukturanalyse entworfene Algorithmen nun auch auf wiederholungsbasierte Strukturen anwenden zu können. Somit eröffnet dieses Verfahren neue Wege zur Vereinigung beider Prinzipien in einem gemeinsamen Ansatz zur Strukturanalyse.

3.1. Einleitung

Die automatische Strukturanalyse – die Zerlegung einer gegebenen Audioaufnahme in zeitliche Segmente und die Gruppierung dieser Segmente in musikalisch sinnvolle Kategorien – stellt eines der zentralen Probleme des *Music Information Retrieval* dar [147], siehe auch Kapitel 2. Aufgrund verschiedener Strukturierungsprinzipien wie zeitliche Reihenfolge, Wiederholungen, kontrastierende Elemente, Variationen und Homogenität ist das Auffinden der musikalischen Struktur eine herausfordernde und oftmals unzureichend spezifizierte Problemstellung [183].

Insbesondere standen die beiden Segmentierungsprinzipien Wiederholung und Homogenität im Fokus bisheriger Forschungsansätze [147, 153]. Bei der wiederholungsbasierten Segmentierung werden wiederkehrende musikalische Passagen identifiziert, wohingegen bei der homogenitätsbasierten Segmentierung Stellen gesucht werden, in denen eine musikalischen Eigenschaft wie Tonart, Tempo oder Klangfarbe keine Veränderung erfährt.

In früheren Forschungen wurden zahlreiche Extraktions- und Clustering-Verfahren vorgestellt, welche die Handhabung entweder von Pfad- oder von Blockstrukturen ermöglichen, siehe u. a. [28, 79, 100, 127, 132, 146, 147, 153].

[1] Dieses Kapitel stellt eine erweiterte Version von [62] dar.

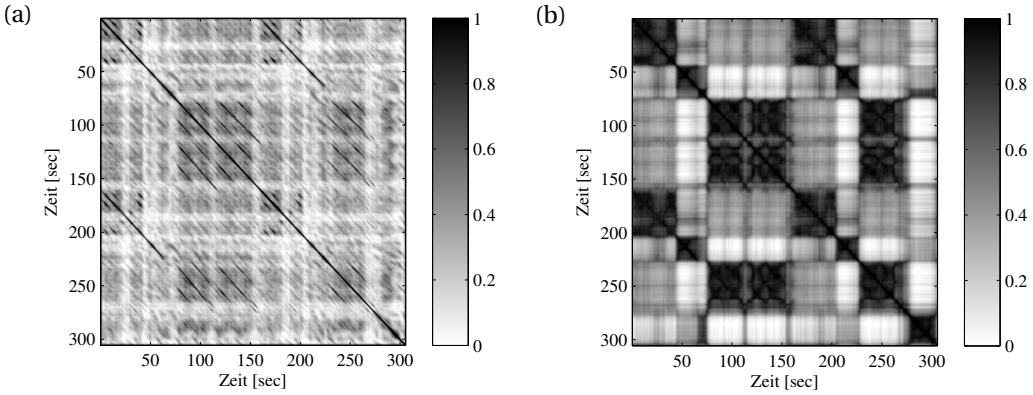


Abbildung 3.1.: Die Umwandlung einer **(a)** Selbstähnlichkeitsmatrix mit Pfadstrukturen in eine **(b)** Blockstrukturmatrix erlaubt die Anwendung homogenitätsbasierter Segmentierungsmethoden.

In [146] wird ein vereinheitlichtes Optimierungsverfahren vorgestellt, welches Pfad- und Blockstrukturen gleichzeitig erfasst. In [159, 174] werden strukturelle Veränderungen bezüglich Pfad- und Blockelementen zur Herleitung von Segmentgrenzen ermittelt. In [77, 79, 100] werden Ansätze zur homogenitätsbasieren Strukturanalyse eingeführt, wobei Glättungs- und Clustering-Techniken zur Verstärkung von Blockstrukturen als Vorverarbeitungsschritt angewendet werden. Eine sehr interessante Forschungsrichtung wird in [158] angedeutet, bei der eine Audioaufnahme lokal als pfad- oder block-ähnlich klassifiziert wird, um danach die verwendete Strategie zur Segmentierung zu wählen.

In diesem Kapitel beschäftigen wir uns mit der Aufgabe, ein wiederholungsbasiertes Strukturanalyse-Problem in ein homogenitätsbasiertes Problem umzuformen, um das in Abschnitt 2.6.1 vorgestellte Segmentierungsverfahren einsetzen zu können, siehe auch Abbildung 3.1. Die wesentlichen Vorteile dieser Methode sind ihre einfache Anwendbarkeit und die im Vergleich zu Pfadextraktions-Methoden höhere Robustheit gegenüber schwachen Pfadanfängen und -enden. Weiterhin entfällt die Notwendigkeit zur manuellen Rekonstruktion von Transitivitätsinformationen [163], siehe hierzu auch die Diskussion des letzten Beispiels in Abschnitt 3.5.1. Bei diesem Verfahren wird also im Gegensatz zu lokal agierenden Pfadextraktionsmethoden stets das ganze Stück berücksichtigt. Hingegen ist der diffizile Schritt der lokalen Pfadverstärkung weiterhin erforderlich. Ein weiterer systematischer Nachteil gegenüber rein wiederholungsbasierten Verfahren ist die Kombination der in Abschnitt 2.5 vorgestellten strukturellen Nachteile beider Segmentierungsprinzipien, somit können weder ausschließlich paarweise auftretende Segmente erkannt noch Grenzen zwischen Segmenten gleicher Bezeichnung ermittelt werden.

In diesem Kapitel stellen wir zunächst die von uns gewählte Methode zur Pfadverstärkung (Abschnitt 3.2) vor und erläutern insbesondere das *Image Opening*-Verfahren (Abschnitt 3.2.1). Danach folgt in Abschnitt 3.3 die Erläuterung des Konvertierungsalgorithmus von Pfad- zu Blockstrukturen. Ein tieferer Einblick in die Theorie hinter diesem Verfahren sowie eine Analy-

se einiger Eigenschaften wird in Abschnitt 3.4 gegeben. In Abschnitt 3.5 folgen sowohl eine qualitative als auch eine quantitative Auswertung des Verfahrens. Das Kapitel findet seinen Abschluss mit einer kurzen Diskussion und Zusammenfassung in Abschnitt 3.6.

3.2. Pfadverstärkung

Aus zeitlich fein aufgelösten Merkmalsfolgen berechnete Selbstähnlichkeitsmatrizen zeigen Wiederholungen musikalischen Materials als diagonale, pfadähnliche Strukturen an. In den meisten wiederholungsbasierten Ansätzen zur Segmentierung werden daher Methoden zur Pfadverstärkung genutzt, um diese Strukturen deutlicher herauszuarbeiten. Einige Aspekte der im diesem Abschnitt beschriebenen Methode sind generisch, andere speziell auf unsere Anwendung hin ausgerichtet. Für die Anforderungen unseres Konvertierungsverfahrens sind wir auf deutliche, zusammenhängende und rauscharme Pfade angewiesen. Hingegen sind die exakten Positionen der Start- und Endpunkte der Pfadstrukturen von nachrangiger Bedeutung. Zusätzlich ist unser Verfahren in der Lage, fehlende Pfadinformationen bis zu einem gewissen Grade zu rekonstruieren, wohingegen zusätzliche Pfadstrukturen, die keine strukturelle Bedeutung aufweisen, zu Schwierigkeiten führen können.

Bei den von uns verwendeten CRP-Merkmalen[2] (*Chroma DCT-Reduced Log Pitch*) handelt es sich um eine gegenüber Klangfarbenänderungen robuste Variante der harmoniebasierten Chroma-Merkmale [119]. In unseren Experimenten verwenden wir zuerst ungeglättete CRP-Merkmale mit einer Merkmalsauflösung von 10 Hz. Nach ℓ^2-Normalisierung dieser Merkmale wird eine Selbstähnlichkeitsmatrix(engl. *Self-similarity matrix*, SSM) durch paarweisen Vergleich der Elemente dieser Merkmalsfolge berechnet (Abbildung 3.2a), wobei wir das innere Produkt (Standard-Skalarprodukt) als Ähnlichkeitsmaß verwenden. Manchmal treten Wiederholungen auch als Transponierung in einer anderen Tonart auf, deren Auffinden in unserem Verfahren durch Betrachtung einer transpositionsinvarianten Selbstähnlichkeitsmatrix wie in [117] ermöglicht wird. Für eine ausführliche Beschreibung zur Berechnung dieser Selbstähnlichkeitsmatrizen sei auf Abschnitt 2.4 verwiesen. Aus Effizienzgründen skalieren wir die so berechnete Selbstähnlichkeitsmatrix auf eine fixe Größe von zumeist 1000×1000. Somit erhalten wir bei längeren Stücken eine gröbere Merkmalsauflösung als bei kürzeren Stücken.

Zu einer weiteren Verstärkung der Pfadstrukturen der Selbstähnlichkeitsmatrix $\mathcal{S}$ wird typischerweise eine Art Glättungsfilter in Richtung der Hauptdiagonalen angewendet, wodurch diagonal verlaufende Muster in $\mathcal{S}$ verstärkt und andere Strukturen abgeschwächt werden, siehe Abbildung 3.2b. In unserer Implementierung verwenden wir eine Glättungsvariante ähnlich zu der in [127] beschriebenen Methode, die auch lokale Tempoabweichungen berücksichtigen kann, siehe auch Codebeispiel 3.1.

Im nächsten Schritt werden üblicherweise Techniken zur Eliminierung kurzer und schwacher Pfadfragmente eingesetzt, siehe auch [174] für eine Beschreibung ähnlicher Ansätze. Wir

[2] Eine MATLAB-Implementierung dieser Merkmale kann unter `www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/` heruntergeladen werden.

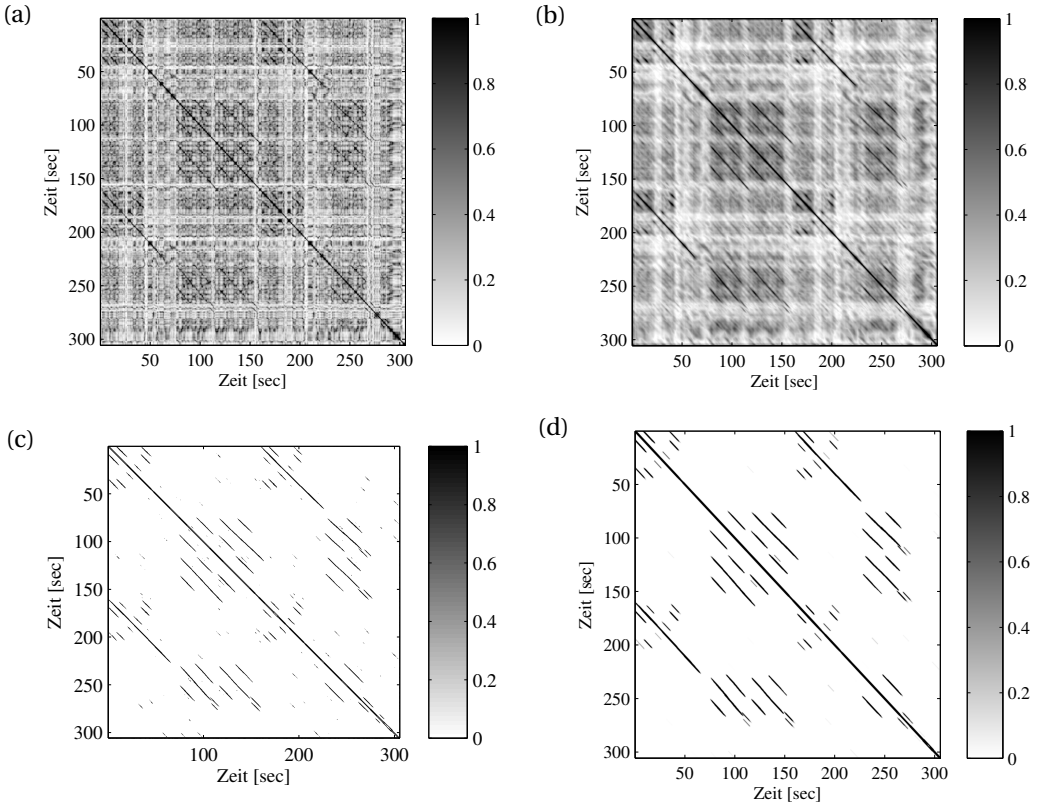


Abbildung 3.2.: Verstärkung von Pfadstrukturen einer Selbstähnlichkeitsmatrix: **(a)** Ausgangsmatrix, **(b)** 3 s-Pfadglättung, **(c)** Adaptiver Schwellwert, **(d)** *Image Opening*.

verwenden hier Methoden aus der Bildverarbeitung, indem wir die geglättete SSM wie ein Graustufenbild behandeln. Hohe Werte in der Selbstähnlichkeitsmatrix entsprechen dabei wie in den Illustrationen dunklen Farbwerten, niedrige Werte hellen. Zu diesen Methoden zählen zuerst die sogenannten *Schwellwert*-Verfahren (engl. *threshold*) zur Umwandlung eines Graustufenbildes in ein Binärbild. Hierbei handelt es sich um eine verallgemeinerte Rundungsregel, bei der alle Grauwerte größer oder gleich diesem Schwellwert durch schwarz (bzw. 1), alle kleineren durch weiß (0) ersetzt werden.

Für unser Verfahren verwenden wir statt diesem globalen Schwellwert für jede Bildposition einen lokalen Schwellwert, der nur von seiner unmittelbaren Umgebung abhängt. Diese Herangehensweise wird als *adaptives Schwellwertverfahren* (*adaptive thresholding*) bezeichnet und ist ein verbreitetes Verfahren in der automatischen Bildverarbeitung, welches unter anderem in der OpenCV-Bibliothek enthalten ist. Das adaptive Schwellwertverfahren stellt von seiner Funktionsweise her ein Hochpassfilter dar, bei dem der Grauwert jedes Bildpunktes mit den Werten seiner unmittelbaren Umgebung verglichen wird. Anschließend wird die Differenz aus dem betrachteten Wert und dem Mittelwert seiner Umgebung gebildet. Üblicherweise wird

```
% Input: featureSeq, winLength, (path) angles

% Compute transposition-invariant distance matrix
for t=1:12
   tmpDM = featureSeq_to_distMatrix(featureSeq, circshift(featureSeq, t-1));
   transpDM(:,:,t) = DM_smoothPaths(tmpDM, 0.5*winLength, angles);
end
distMatrix = min(transpDM, [], 3);

% An SSM is the inverse of a distance matrix
ssm = 1-distMatrix;
ssm = SM_threshold(ssm);

% Prepare structural elements and perform image opening
for a=1:length(angles)
   path = strel('line', winLength, -angles(a));
   pathMatrix(:,:,a) = imopen(ssm, path);
end
pathMatrix = max(pathMatrix, [], 3);
```

```
function matrix = DM_smoothPaths(matrix, winlength, angles)
   for a = 1:length(angles)
      % Creates a straight line having specific angle and length:
      myFilter = fspecial('motion', winlength, -angles(a));
      matrixArray(:,:,a) = conv2(matrix, myFilter, 'same');
   end
   matrix = min(matrixArray, [], 3);
end
```

Codebeispiel 3.1: MATLAB-Skript zur Berechnung einer Pfadmatrix `pathMatrix` aus einer gegebenen Merkmalsfolge `featureSeq` mittels Pfadglättung und Image Opening.

auf diese Differenzen dann ein globales Schwellwertverfahren mit Schwelle 0 angewendet, also genau diejenigen Punkte schwarz gefärbt, die dunkler als ihre Umgebung sind, wohingegen alle anderen weiß gefärbt werden. Für weitere Informationen zu diesem Verfahren siehe auch [138, 168].

Wir verwenden eine frei verfügbare Implementierung [210] dieses Verfahrens mit zusätzlichen Modifikationen: Nach Anwendung des Verfahrens werden (anstelle der Verwendung eines globales Schwellwertverfahrens) die negativen Werte durch den Wert 0 ersetzt und anschließend alle Werte quadriert, um den Einfluss kleinerer Differenzwerte zu reduzieren. Danach führen wir eine weitere Pfadglättung wie oben beschrieben durch und berechnen punktweise den Mittelwert zwischen der so bearbeiteten und der ungeglätteten Version. Dies führt dazu, dass auch schwache pfadähnliche Strukturen gefunden und verstärkt werden. Die so entstandene Matrix ist in Abbildung 3.2c illustriert und wird anschließend punktweise mit der ursprünglichen Selbstähnlichkeitsmatrix multipliziert.

Das so vorverarbeitete Bild enthält nun neben den Pfadstrukturen auch noch einige wenige Pfadfragmente und Artefakte, die für unser Verfahren noch entfernt werden müssen. Hierzu setzen wir ein weiteres Verfahren aus der Bildverarbeitung namens *Image Opening* ein, bei dem das Bild lokal mit Diagonallinien einer vorgegebenen Mindestlänge verglichen wird. Die

Richtungen dieser Linien stimmen mit den im vorangegangenen Glättungsschritt verwendeten Linien überein, ihre Mindestlänge ist allerdings doppelt so groß wie die Länge der zur Glättung gebrauchten Linien, um nicht künstlich entstandene Strukturen irrtümlicherweise als Pfade zu werten. Dieses Verfahren erlaubt somit das Aussondern von Rauscheffekten und erhält nur die markantesten Pfadstrukturen. In Abschnitt 3.2.1 werden wir eine kurze mathematische Beschreibung dieses Verfahrens geben. Eine Implementierung dieser Methode ist Bestandteil der *MATLAB Image Processing Toolbox* und wird durch die Funktion `imopen` aufgerufen. In Codebeispiel 3.1 ist eine gekürzte und leicht vereinfachte Version unserer Implementierung zur Berechnung einer Pfadmatrix dargestellt.

Im Anschluss daran führen wir einige heuristische Optimierungsschritte durch: Schwache Pfade können durch zusätzliche Skalierungs- und Schwellwertverfahren verstärkt werden. In manchen Fällen kann ein zusätzlicher Erosionsschritt (vgl. Abschnitt 3.2.1) zur Behebung der durch diagonale Glättung hervorgerufenen Pfadverlängerungen verwendet werden. Abschließend stellt eine leichte Glättung mit einem 3×3-Gaußfilter sicher, dass die Pfade eine Breite größer als 1 aufweisen. Diese Eigenschaft ist eine notwendige Voraussetzung für die Anwendbarkeit des Konvertierungsverfahrens von Pfaden zu Blöcken, siehe auch Abbildung 3.10.

Die so entstandene pfadverstärkte Selbstähnlichkeitsmatrix bezeichnen wir als *Pfadmatrix* , ein Beispiel ist in Abbildung 3.2d illustriert. Andere Verfahren wie [116, 152] verwenden eine solche Matrix direkt zur Herleitung der gesuchten Pfadstrukturen. Die *Extraktion* solcher Pfade ist allerdings oftmals anfällig für Fehler und zudem stark abhängig von Schwellwertparametern, was zu einem instabilen Verhalten führt, wobei insbesondere die Festlegung der Start- und Endpunkte der Pfade sich als schwierig erweist. Weiterhin ist die konkrete Klassifikation der Segmente auch hier ein nichttriviales Problem, siehe hierzu [116, 163].

Um diese Schwierigkeiten zu umgehen, haben wir ein Verfahren entwickelt, welches Pfadstrukturen einer Selbstähnlichkeitsmatrix in blockartige Strukturen überführt. Somit stellt die aus der Selbstähnlichkeitsmatrix erzeugte Pfadmatrix die Eingabe unseres Konvertierungsalgorithmus' dar. Man beachte, dass einige Details der Implementierung zur Berechnung der Pfadmatrix austauschbar sind. Unser Verfahren ist in gewissem Maße generisch und funktioniert mit allen Varianten von pfadverstärkten Selbstähnlichkeitsmatrizen, sofern diese eine dünnbesetzte Struktur aufweisen, die nur die relevanten Pfade beinhaltet.

Unsere Experimente (vgl. Abschnitt 3.5) haben gezeigt, dass sich die Herleitung einer Pfadmatrix im Allgemeinen ziemlich robust gegenüber Parameteränderungen verhält, allerdings mit einer Ausnahme betreffend die Länge der angestrebten Segmente. Dieser Wert wird sowohl während des lokalen Schwellwertverfahrens als auch von Image Opening verwendet und hat großen Einfluss auf die Menge und Gestalt der verbleibenden Pfade, siehe dazu auch Abbildung 3.13 in Abschnitt 3.5.1. Bei einer umfassenden Analyse der Performance verschiedener Segmentierungsverfahren hat sich eine signifikante Korrelation zwischen der Länge T eines Musikstücks in Sekunden und der durchschnittlichen Länge der manuell annotierten Segmente ergeben [181]. Daher wählen wir oftmals für diesen Parameter eine Länge von $\sqrt{T}$ Sekunden, ggf. mit einem konstanten Vorfaktor. In anderen Fällen ist eine konstante Länge

sinnvoll, hier haben sich 8 Sekunden bewährt. Für die vorangegangene Glättung nutzen wir zumeist unabhängig von dem Längenparameter für die Bildverarbeitungsschritte eine feste Länge von 3 Sekunden.

3.2.1. Image Opening

Image Opening ist eine sehr verbreitete Methode zur Rauschunterdrückung in der automatischen Bildverarbeitung, die auf lokalen Vergleichen eines Binär- (schwarz/weiß) oder Graustufenbildes mit einem *strukturierenden Element* basiert. In unserem Verfahren wird die Rolle des Bildes von der geglätteten Selbstähnlichkeitsmatrix eingenommen, die des strukturierenden Elements nacheinander von den verschiedenen Prototypen für die diagonal verlaufenden Linien.

In der mathematischen Morphologie wird mit dem Begriff »Image Opening« eine spezielle Operation beschrieben. Die folgende kurze Einführung basiert auf [172]. Üblicherweise betrachten wir ein Schwarz-Weiß-Bild als Binärmatrix, wobei schwarze Bildpunkte (engl. *pixel*) durch 1, weiße durch 0 dargestellt werden. Im Folgenden bezeichnen wir mit X die *Koordinaten* aller schwarzen und mit X^c die Koordinaten der weißen Bildpunkte. Die Menge $X + h$ beschreibe die *Translation* von X um einen Vektor $h \in \mathbb{Z}^2$, also $X + h := \{x + h \mid x \in X\}$. Weiterhin definieren wir die *transponierte Menge* $-X$ von X als die Menge aller Bildindizes mit $-x \in X$. Somit beschreibt $-X$ die Reflexion von X am Nullpunkt.

Nun können wir zwei grundlegende morphologische Operationen namens Erosion und Dilatation definieren. Beide Methoden sind für ein Bild X gemeinsam mit einem strukturierenden Element B definiert, welches ebenfalls eine Indexmenge im obigen Sinne ist. Die Operation Image Opening stellt die Hintereinanderausführung von Erosion und Dilatation mittels desselben Strukturelements dar. In Abbildung 3.3a wird die Pfadstruktur unseres Hauptbeispiels als Binärbild dargestellt, Teil (b) der Abbildung zeigt maßstabsgetreu eine Diagonallinie als strukturierendes Element.

Gegeben sei ein Bild X und ein strukturelles Element B. Die *Erosion* von X beinhaltet die Indizes x, für die $B + x$ vollständig in X enthalten ist. Sie ist folglich definiert als:

$$X \ominus B := \left\{x \in \mathbb{Z}^2 \mid B + x \subseteq X\right\} = \bigcap_{b \in B} X - b.$$

Abbildung 3.3c illustriert die Anwendung dieser Operation.

Die *Dilatation* von X bezeichnet die Menge derjenigen Indizes x, für welche die Menge $-B + x$ die Menge X schneidet, also

$$X \oplus B := \left\{x \in \mathbb{Z}^2 \mid (-B + x) \cap X \neq \emptyset\right\} = \bigcup_{b \in B} X + b.$$

In Abbildung 3.3d wird ein Beispiel für Dilatation dargestellt.

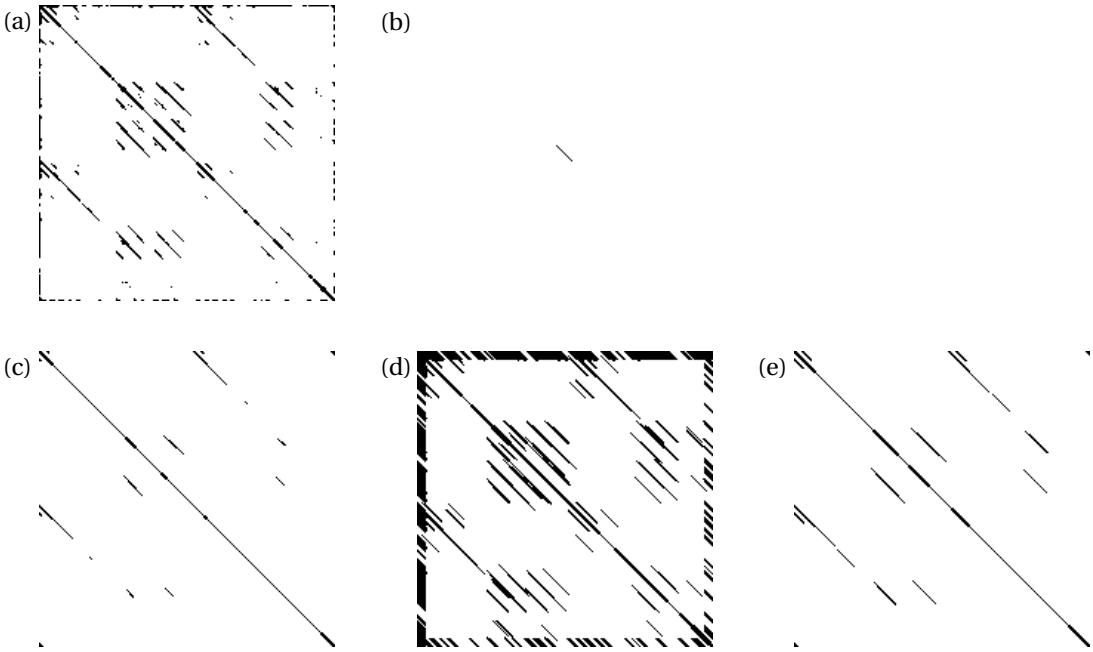


Abbildung 3.3.: Beispiele für morphologische Operationen eines Binärbildes – hier einer gröberen Version der Wiederholungsstruktur unseres Elgar-Beispiels: **(a)** Ausgangsbild X, **(b)** strukturelles Element B, **(c)** Erosion $X \ominus B$, **(d)** Dilatation $X \oplus B$, **(e)** Image Opening $X \circ B$.

Die Operation *Image Opening* angewendet auf X mit strukturellem Element B ist definiert als die Hintereinanderausführung von Erosion und Dilatation:

$$X \circ B := (X \ominus B) \oplus B \ .$$

Diese Menge aller Indizes x, für die $(-B + x)$ die erodierte Menge $X \ominus B$ schneidet, erlaubt eine geometrische Interpretation von $X \circ B$ als die Menge all derjenigen Indizes, die mindestens eine Translation von B vollständig überdecken. In [172, S. 52] wird dies beschrieben als »*The opening is the domain swept out by all the translates of B which are included in [...] X*«. Abbildung 3.3e illustriert die Anwendung dieser Operation.

Image Opening ist eine idempotente Operation, d. h. wird auf die Ausgabe dieser Operation noch einmal dieselbe Operation mit demselben strukturierenden Element angewendet, so verändert sie sich nicht mehr. Mit anderen Worten, es gilt für alle Bilder X und alle strukturierenden Elemente B die Gleichung $(X \circ B) \circ B = X \circ B$.

Bei Grauwertbildern bzw. entsprechenden strukturierenden Elementen wird üblicherweise die folgende Notation verwendet: x bzw. b stellen Abbildungen einer Indexmenge bzw. eines

Gitters $E \subseteq \mathbb{Z}^2$ auf die Menge $\mathbb{R} \cup \{-\infty, \infty\}$ dar [36]. In diesem Fall ist Erosion definiert durch

$$(x \ominus b)(i) = \inf_{j \in E}(x(j) - b(j - i)) ,$$

und Dilatation durch

$$(x \oplus b)(i) = \sup_{j \in E}(x(j) + b(i - j)) .$$

Auch hier wird Image Opening durch die Komposition dieser beiden Operatoren realisiert: $x \circ b := (x \ominus b) \oplus b$.

3.3. Konvertierungs-Algorithmus

Ausgehend von einer pfadverstärkten Selbstähnlichkeitsmatrix $\mathcal{S} \in [0,1]^{N \times N}$, die aus einer N-elementigen Merkmalsfolge generiert wurde, werden wir nun einen Algorithmus zur Berechnung einer neuen $N \times N$-Matrix aus Blockstrukturen vorstellen.

Die Matrix $\mathcal{S}$ beschreibt nun nicht mehr die Ähnlichkeit zwischen verschiedenen Zeitpunkten des Musikstückes, sondern zeigt durch die Pfadverstärkung die Ähnlichkeit von musikalischen Passagen an. Ist der Wert $\mathcal{S}(i, j)$ für zwei verschiedene Zeitpunkte i und j hoch, so befinden sich diese Zeitpunkte an derselben relativen Position eines wiederholten Segments. Zuerst modifizieren wir diese Pfadmatrix $\mathcal{S}$, um eine symmetrische Matrix zu erhalten. Zwar hat der paarweise Vergleich zwischen den Elementen einer Merkmalsfolge mittels eines symmetrischen Ähnlichkeitsmaßes grundsätzlich zu einer symmetrischen Selbstähnlichkeitsmatrix geführt, allerdings mag diese Eigenschaft durch die oben beschriebenen Pfadverstärkungen und Bildverarbeitungstechniken in einigen Details verloren gegangen sein. Diese Symmetrie kann durch Betrachtung von $\mathcal{S}^{\text{Pfad}} := \frac{1}{2}(\mathcal{S} + \mathcal{S}^\top)$ anstatt von $\mathcal{S}$ leicht wiederhergestellt werden, wobei $\mathcal{S}^\top$ die transponierte Matrix von $\mathcal{S}$ bezeichnet.

Als nächstes führen wir eine Eigenwertzerlegung der symmetrischen Matrix $\mathcal{S}^{\text{Pfad}}$ durch und untersuchen die Eigenschaften der resultierenden Eigenvektoren. Aus der Theorie der Hauptachsentransformation wissen wir, dass eine reellwertige Diagonalmatrix $D = \text{diag}(\lambda_1, \ldots, \lambda_N)$ und eine orthogonale Matrix E existieren, sodass $\mathcal{S}^{\text{Pfad}} = EDE^\top$ gilt, wobei die n-te Spalte $E(n)$ von E ein Eigenvektor von $\mathcal{S}^{\text{Pfad}}$ zum Eigenwert λ_n ist, d. h. es gilt $\mathcal{S}^{\text{Pfad}} E(n) = \lambda_n E(n)$. In unserer Implementierung multiplizieren wir die normierten Eigenvektoren mit ihren jeweiligen Eigenwerten und sortieren die so modifizierten Eigenvektoren bezüglich der Größe ihres Eigenwertes in absteigender Reihenfolge – die Eigenvektoren zum Eigenwert 0 sind hier irrelevant. Wie durch Abbildung 3.4b illustriert wird, transponieren wir die so gewichteten und sortierten Eigenvektoren und fassen sie als Zeilen einer $N \times N$-Matrix auf, die wir im Folgenden mit $\widetilde{E}$ bezeichnen.

Um eine bessere Intuition für diesen Algorithmus zu entwickeln, schauen wir uns drei synthetische Beispiele in Abbildung 3.5 an. Der Fall eines wiederholungsfreien Musikstücks ist mit seiner Eigenwertzerlegung in den Teilen (a) und (b) dargestellt, wohingegen der Fall zweier wie-

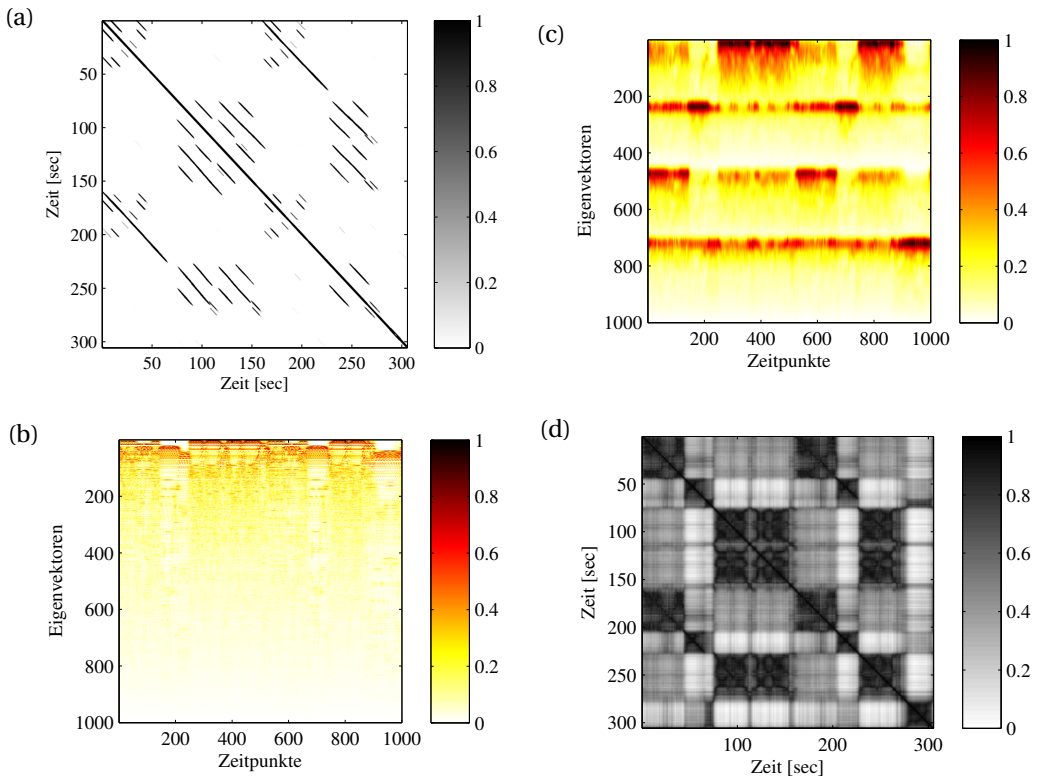


Abbildung 3.4.: Illustration des Algorithmus für die Konversion von Pfad- in Blockstrukturen: **(a)** Pfadverstärkte und symmetrisierte SSM, wie sie typischerweise für wiederholungsbasierte Strukturanalysen verwendet wird, **(b)** mit dem dazugehörigen Eigenwert gewichtete und sortierte Eigenvektoren (Zeilen) der SSM, **(c)** Eigenvektoren (Zeilen) nach Clustering und Nachverarbeitung, **(d)** aus (c) berechnete Selbstähnlichkeitsmatrix.

derholter Segmente in den Teilen (c) und (d) dargestellt ist. Ein drittes Beispiel entsprechend der Wiederholungsstruktur $A_1 B_1 B_2 B_3 A_2$ wird in den Teilen (e) und (f) gezeigt.

Wie wir im nächsten Abschnitt zeigen werden, spiegelt sich die Pfadstruktur der Selbstähnlichkeitsmatrix $\mathcal{S}^{\text{Pfad}}$ in den Trägereigenschaften (d. h. in den Positionen der Einträge ungleich Null) der Zeilen sowohl von $\mathcal{S}^{\text{Pfad}}$ als auch von $\widetilde{E}$ wider. In der Theorie unter Annahme einer idealen Pfadstruktur haben zwei Zeilen von $\widetilde{E}$ entweder denselben Träger (wenn sie zum selben Musiksegment oder einer Wiederholung gehören) oder die Träger sind disjunkt (wenn sie zu verschiedenen Segmenttypen gehören). Durch diese Beobachtung können wir die Spalten von $\widetilde{E}$ als Merkmalsvektoren betrachten. Dies führt zu einer Merkmalsfolge $\widetilde{E}(1),\ldots,\widetilde{E}(N)$, welche ebenfalls zur Definition einer Selbstähnlichkeitsmatrix $\text{SSM}(\widetilde{E})$ verwendet werden kann. Die Eigenschaften der Eigenvektoren implizieren, dass zwei Vektoren $\widetilde{E}(i)$ und $\widetilde{E}(j)$ weitgehend übereinstimmende Träger haben, wenn die Zeitpunkte i und j zu Wiederholungen desselben Segmenttyps gehören (oder zu Zeitpunkten eines nicht-wiederholten Segments, siehe dazu

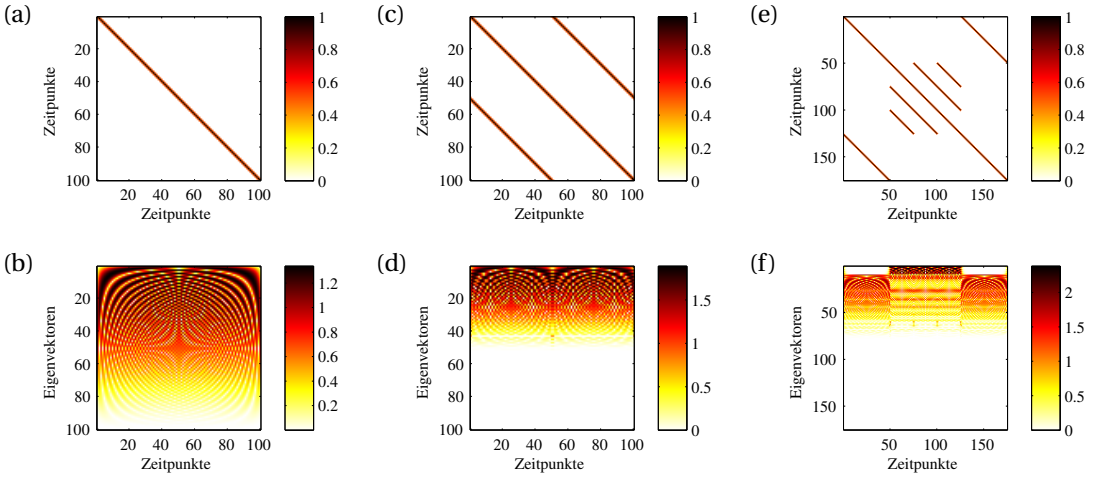


Abbildung 3.5.: Verschiedene Pfadmatrizen (obere Reihe) und dazugehörige Eigenvektoren (untere Reihe).

Abschnitt 3.4.5), und andernfalls sich deutlich unterscheiden. Als Folgerung daraus hat SSM($\widetilde{E}$) die gewünschte Blockstruktur.

Wenn nur eine rein wiederholungsbasierte Strukturierung berechnet werden soll, ist es nicht nötig, eine Selbstähnlichkeitsmatrix mit Blockstrukturen zu berechnen. In diesem Fall können auch die Eigenvektoren selbst segmentiert werden bzw. das Wissen aus dem Clustering-Prozess direkt verwendet werden. Die Stärke dieses Verfahrens liegt darin, wiederholungsbasierte Informationen in die Sprache des homogenitätsbasierten Strukturierungsprinzips übersetzen zu können, wodurch eine kombinierte Sichtweise auf beide Prinzipien im Rahmen eines gemeinsamen technischen Verfahrens möglich wird.

Um die Anwendung dieses Verfahrens auch auf realen Daten zu ermöglichen, wenden wir einige zusätzliche Verarbeitungsschritte auf die Matrix $\widetilde{E}$ an, bevor wir die neue Selbstähnlichkeitsmatrix berechnen. Hierzu ersetzen wir zuerst jeden Eintrag e von $\widetilde{E}$ durch $\log(20|e|+1)$, um so eine übermäßige Bevorzugung des am häufigsten wiederholten Segments zu verhindern. Dann wenden wir ein klassisches *k-means*-Clusteringverfahren[3] an, um die Eigenvektoren (Zeilen von $\widetilde{E}$) umzusortieren, sodass ähnliche Vektoren (d. h. Vektoren, die zu ähnlichen Strukturen korrespondieren) benachbart sind. Anschließend glätten wir die so umsortierte Matrix sowohl horizontal als auch vertikal, siehe Abbildung 3.4d. Hier sorgt die horizontale Glättung für ein Ausbalancieren der strikt positiven Einträge der Eigenvektoren, wohingegen die vertikale Glättung die Robustheit bezüglich lokaler Störungen ermöglicht[4]. Wir bezeichnen die so geglättete Matrix mit $\overline{E}$ und berechnen die dazugehörige Selbstähnlichkeitsmatrix

[3] In unserer Implementation wählen wir die Nummer der Cluster in Abhängigkeit von der Dauer des Musikstückes. Unsere Experimente haben gezeigt, dass jede Anzahl zwischen 5 und 20 zu vergleichbaren Ergebnissen führt.

[4] In unseren Experimenten verwenden wir ein Gaußfenster mit adaptiver Fensterlänge für die vertikale Glättung und einen festen Wert von 7 Zeitpunkten für die horizontale Glättung. Auch hier sind die konkreten Werte nicht ausschlaggebend.

```
pathMatrix = 0.5*(pathMatrix+pathMatrix');
N = size(pathMatrix,1);

% Perform eigenvalue decomposition
[EV, ED] = eig(pathMatrix);
% Weight eigenvectors with their eigenvalues, and transpose them
wEV = abs(ED) * EV;
wEV = wEV';

% Pointwise logarithmic compression
logEV = log(parameter.factorLogCompr*abs(wEV)+1);

% Cluster eigenvectors
score = pdist2(logEV, logEV, 'correlation');
scoreCl = kmeans(score, parameter.sortClusterNum);
[~, idxScore] = sort(scoreCl);
% Re-order eigenvectors due to clustering
logEV = logEV(idxScore, :);

% Smoothing and output
smoothWin = gausswin(ceil(0.25*N/parameter.sortClusterNum));
logEV = conv2(logEV, smoothWin, 'same');
blockMatrix = featureSeq_to_distMatrix(logEV);
```

Codebeispiel 3.2: MATLAB-Skript zur Umwandlung einer pfadverstärkten Selbstähnlichkeitsmatrix `pathMatrix` $= \mathcal{S}$ in eine Blockstrukturmatrix `blockMatrix` $= \mathcal{S}^{\text{Block}}$.

$\mathcal{S}^{\text{Block}} = \text{SSM}(\overline{E})$ wie oben. Diese Matrix stellt das Ergebnis unserer Konvertierungsprozedur dar, siehe Abbildung 3.4d.

Codebeispiel 3.2 zeigt einen Ausschnitt aus dem MATLAB-Code für diese Methode. Man beachte, dass die Implementierung für echte Audiodaten weitere Optimierungsschritte wie Skalierungen und Schwellwertverfahren beinhaltet, die der besseren Übersicht halber für dieses Codebeispiel entfernt wurden.

3.4. Theoretischer Hintergrund

Im folgenden Abschnitt diskutieren wir einige mathematische Hintergründe zur Funktionsweise des vorgestellten Algorithmus. Man beachte dabei, dass diese theoretische Modellierung mit den tatsächlichen Daten zwar im Allgemeinen nicht exakt übereinstimmt und somit die Ergebnisse leichte Unterschiede aufweisen können. Dennoch ist die Struktur der theoretischen Ergebnisse sehr ähnlich zu den in den Experimenten beobachteten Strukturen, wodurch der theoretische Ansatz als eine sinnvolle Erklärung gesehen werden kann. In praktischen Anwendungen erweisen sich die Pfadstrukturen oftmals als verrauscht und gestört, sodass die folgenden Eigenschaften der Eigenvektoren nicht vollständig erfüllt sind.

Zu Beginn betrachten wir eine künstliche Segmentierung und modellieren eine Segmentmatrix. Diese Matrix dient als ein Prototyp einer Selbstähnlichkeitsmatrix eines Musikstückes, welchem eine durch die künstliche Segmentierung vorgegebene Wiederholungsstruktur zu-

grunde liegt. Wir werden zeigen, dass mittels der Eigenvektoren dieser Segmentmatrix eine Selbstähnlichkeitsmatrix mit Blockstrukturen berechnet werden kann, die exakt mit den vorgegebenen Segmenten übereinstimmt. Weiterhin zeigen wir einige Unterschiede zwischen diesem Prototypen einer pfadverstärkten Selbstähnlichkeitsmatrix und den aus echten Audiomerkmalen berechneten Selbstähnlichkeitsmatrizen auf; insbesondere diskutieren wir den Einfluss dieser Unterschiede auf die berechneten Eigenvektoren.

3.4.1. Notationen

In diesem Abschnitt greifen wir auf die in Abschnitt 2.5 eingeführten Bezeichnungen zurück: Wir betrachten ein synthetisches Musikstück mit der Zeitachse $[1:N] := \{1,\ldots,N\}$. Dieses Stück sei in M Segmente unterteilt, wobei die Zeitachse des m-ten Segments durch das Intervall $[\mathcal{B}_m + 1 : \mathcal{B}_{m+1}]$ beschrieben werde, die Länge dieses Segments bezeichnen wir mit $\ell_m := \mathcal{B}_{m+1} - \mathcal{B}_m$. Durch eine segmentweise Benennungsfunktion $\bar{S} : [1:M] \to \mathcal{L} := \{A, B, C, \ldots\}$ werde jedem Segmentindex eine Bezeichnung zugeordnet. Die Menge der verwendeten Bezeichnungen identifizieren wir mit der Menge $[1:K]$, wobei $A \equiv 1$, $B \equiv 2$ usw. Die Menge der Segmente mit Benennung k nennen wir die k-te *Segmentklasse*. Wir nehmen für die theoretische Modellierung vereinfachend an, dass alle Segmente einer Klasse dieselbe Länge n_k aufweisen; die Anzahl dieser Segmente bezeichnen wir mit o_k.

Weiterhin gehen wir davon aus, dass die so vorgegebene Strukturierung *zulässig* im Sinne von Abschnitt 2.5 ist, insbesondere gibt es keine Segmente, die ausschließlich paarweise zusammen auftreten. So ist beispielsweise $\bar{S} \equiv (ABAB)$ nicht zulässig, da die Paare beginnend bei den Positionen 1 und 3 beide die Form (AB) haben, aber weder A noch B im Graph von $\bar{S}$ in einem anderen Kontext auftauchen. Aus wiederholungsbasierter Sicht würden diese beiden Segmente verbunden und als ein neues Segment C bezeichnet werden, sodass die Struktur dieses Beispiels (CC) wäre.

Beispiel 1 *Ein (synthetisches) Musikstück habe die Struktur* $A_1A_2B_1C_1A_3A_4B_2A_5$*, und die Folge* ℓ *der Segmentlängen betrage* $(10,10,30,20,10,10,30,10)$.

Dann gilt für dieses Stück: Die Anzahl M *der Segmente beträgt* 8*, die Zeitachse ist das Intervall* $[1:130]$*, und* (A,A,B,C,A,A,B,A) *ist die Folge der Werte der Benennungsfunktion* $\bar{S}$*. Weiterhin gilt* $n_1 = 10$*,* $n_2 = 30$*,* $n_3 = 20$ *sowie* $o_1 = 5$*,* $o_2 = 2$*,* $o_3 = 1$.

Für die k-te Segmentklasse nennen wir die Menge der dazugehörigen Zeitintervalle

$$T_k := \left\{ [\mathcal{B}_m + 1 : \mathcal{B}_{m+1}] \mid \bar{S}(m) = k \right\}.$$

Diese Elemente sortieren wir gemäß der Ordnung der $\mathcal{B}_m$, somit beschreibe $T_{k,i}$ $(i \in [1:o_k])$ das Zeitintervall des i-ten Segments der k-ten Segmentklasse. Weiterhin wird wie in Abschnitt 2.5 beschrieben eine punktweise Strukturierung $\dot{S}$ berechnet, welche jedem Zeitpunkt des Musikstücks eine Bezeichnung zuordnet.

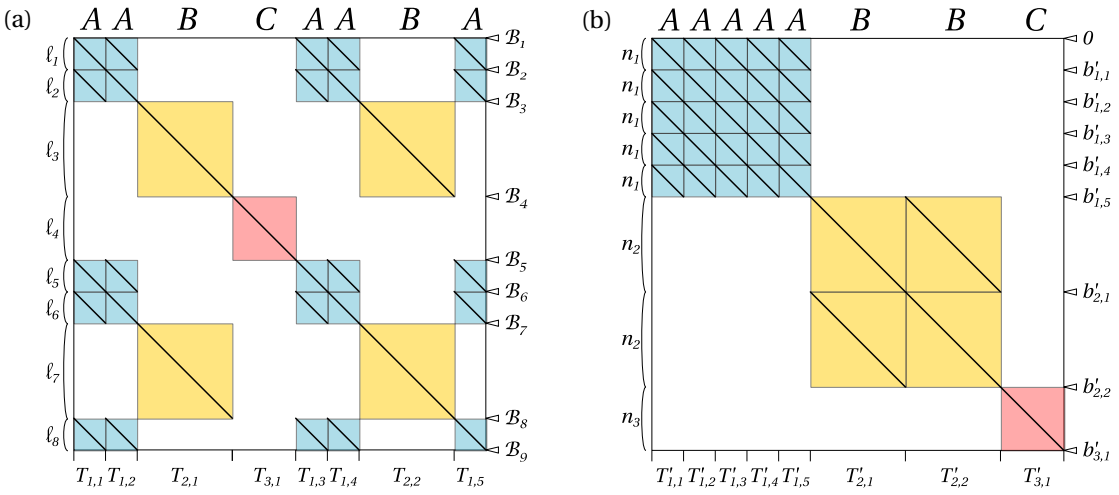


Abbildung 3.6.: Synthetische Selbstähnlichkeitsmatrizen zu Beispiel 1 mit der Wiederholungsstruktur $A_1A_2B_1C_1A_3A_4B_2A_5$. **(a)** Strukturmatrix A mit hervorgehobener Blockstruktur, **(b)** nach Segmentklassen sortierte Matrix A'.

3.4.2. Strukturmatrizen

Nun konstruieren wir für diese Segmente eine Darstellung in Matrixform, die uns später als Prototyp einer Selbstähnlichkeitsmatrix dient. Für ein $0 < \varepsilon \leq 1$ bezeichne dazu I_k^ε die ε-Tridiagonalmatrix der Größe $k \times k$, also

$$I_k^\varepsilon := \begin{pmatrix} 1 & \varepsilon & & & \\ \varepsilon & 1 & \varepsilon & & \\ & \ddots & \ddots & \ddots & \\ & & \varepsilon & 1 & \varepsilon \\ & & & \varepsilon & 1 \end{pmatrix} \in \{0, \varepsilon, 1\}^{k \times k}.$$

Für eine vorgegebene Strukturierung der Menge $[1 : N]$ bestehend aus einer Folge ℓ von Segmentlängen (bzw. in äquivalenter Darstellung aus einer Menge $\mathcal{B}$ von Segmentgrenzen) und einer segmentweisen Benennungsfunktion $\bar{S}$ definieren wir die $N \times N$-Blockmatrix

$$A := \left(\delta_{\bar{S}(i),\bar{S}(j)} \cdot I^\varepsilon_{\ell_i}\right)_{ij} \tag{3.1}$$

und bezeichnen sie als *Strukturmatrix*. Sie entspricht der Idealform der Selbstähnlichkeitsmatrix eines Stückes, welches die Struktur $(\mathcal{B}, \bar{S})$ aufweist, für eine Illustration siehe Abbildung 3.6a. Die Begründung für die Auswahl dieser Tridiagonalmatrix folgt später.

Für die Berechnung der Eigenwertzerlegung dieser Matrix führen wir eine Umsortierung der Segmente durch, sodass alle Elemente einer Segmentklasse direkt hintereinander stehen,

wobei die Segmentklassen alphabetisch nach ihren Bezeichnungen sortiert werden. Hierdurch erhalten wir eine *sortierte Strukturmatrix* A', eine neue Menge von Segmentgrenzen $\mathcal{B}'$ sowie eine entsprechende Benennungsfunktion $\bar{S}'$. Diese Matrix wird in Abbildung 3.6b illustriert.

Im Rest dieses Abschnittes beschreiben wir diesen Sortiervorgang: Die zur sortierten Strukturmatrix A' korrespondierenden Segmentgrenzen bezeichnen wir mit $\mathcal{B}' = \left\{0, b'_{1,1}, \ldots, b'_{1,o_1}, \ldots, b'_{K,1}, \ldots, b'_{K,o_K}\right\}$, wobei

$$b'_{k,j} = \sum_{i<k} o_i n_i + j n_k \,.$$

Somit können wir die Menge T' der korrespondierenden Zeitintervalle mittels

$$T'_{k,j} := \begin{cases} \left[b'_{k,j-1} + 1 : b'_{k,j}\right] & j > 1 \\ \left[b'_{k-1,o_{k-1}} + 1 : b'_{k,1}\right] & j = 1 \end{cases}$$

definieren, wobei $b'_{0,\bullet} := 0$.

Hintergrund: Mit $\oplus$ bezeichnen wir die *direkte Summe* von Matrizen, d. h. für zwei Matrizen X und Y gilt

$$X \oplus Y := \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix}.$$

Für zwei Matrizen X und Y ist das *Kroneckerprodukt* wie folgt definiert:

$$X \otimes Y := \begin{pmatrix} x_{11}Y & \ldots & x_{1n}Y \\ \vdots & & \vdots \\ x_{n1}Y & \ldots & x_{nn}Y \end{pmatrix} \quad \text{mit } X = (x_{ij}).$$

Mit $\mathbb{1}^{m\times n}$ bezeichnen wir die $m \times n$-Matrix, bei der jeder Eintrag identisch 1 ist. ◀

Berechnen wir nun die zu $(\mathcal{B}', \bar{S}')$ gehörige und in Abbildung 3.6b illustrierte Strukturmatrix A', so erhalten wir

$$A' = \bigoplus_{k=1}^{K} \left(\mathbb{1}^{o_k \times o_k} \otimes I^{\varepsilon}_{n_k}\right) =: \bigoplus_{k=1}^{K} A'_{kk} \,. \tag{3.2}$$

Da beide Segmentierungen bis auf die Sortierung identisch sind, gilt die Gleichung $A' = P^{-1} \cdot A \cdot P$, wobei P die Permutationsmatrix zur Permutation

$$\pi = \begin{pmatrix} T'_{1,1}, \ldots, T'_{1,o_1} & T'_{2,1}, \ldots, T'_{2,o_2} & \cdots & T'_{K,1}, \ldots, T'_{K,o_K} \\ T_{1,1}, \ldots, T_{1,o_1} & T_{2,1}, \ldots, T_{2,o_2} & \cdots & T_{K,1}, \ldots, T_{K,o_K} \end{pmatrix}$$

bezeichnet. Dies ist eine Schiebepermutation (engl. *shuffle permutation*), welche $(\mathcal{B}', \bar{S}')$ zurück in die Originalstrukturierung $(\mathcal{B}, \bar{S})$ überführt und dabei die zeitliche Reihenfolge innerhalb der Segmente unverändert lässt.

3.4.3. Eigenwertzerlegung

Da A' eine symmetrische Matrix ist, existiert nach dem Satz über die Hauptachsentransformation eine reellwertige Diagonalmatrix $D = \mathrm{diag}(\mu_1, \ldots, \mu_N)$ sowie eine orthogonale Matrix E, sodass $A' = EDE^\top$. Hierbei ist die j-te Spalte e_j von E ein Eigenvektor von A' zum Eigenwert μ_j, d. h. $A'e_j = \mu_j e_j$.

Multiplizieren wir jeden Eigenvektor aus E mit seinem zugehörigen Eigenwert, so erhalten wir die *gewichtete Eigenvektormatrix*

$$\tilde{E} = \begin{pmatrix} \tilde{e}_1 & \cdots & \tilde{e}_N \end{pmatrix} \quad \text{mit} \quad \tilde{e}_j := \mu_j e_j \,.$$

Der *Träger* eines Vektors x ist definiert durch $\mathrm{supp}(x) := \{i \mid x_i \neq 0\}$. In allen Abbildungen dieses Kapitels haben wir die Eigenvektoren immer zeilenweise gezeichnet, d. h. streng genommen werden dafür Linkseigenvektoren berechnet. Da hier die Linkseigenvektoren genau die transponierten Rechtseigenvektoren sind, greifen wir für die Berechnung im folgenden Abschnitt auf die geläufigere Notation als Spaltenvektoren zurück.

Zuerst zeigen wir, dass wir bei verschiedenen Diagonalblöcken disjunkte Träger der Spalten der gewichteten Eigenvektormatrix erreichen können:

> **Hintergrund:** Bildet man die direkte Summe $A \oplus B$ zweier beliebiger quadratischer Matrizen A und B, und ist λ ein Eigenwert von A oder von B, so gilt:
>
> $$\mathrm{EV}_\lambda(A) \oplus \mathrm{EV}_\lambda(B) = \mathrm{EV}_\lambda(A \oplus B) \,,$$
>
> wobei mit $\mathrm{EV}_\lambda(A) := \{v \mid Av = \lambda v\}$ der Eigenraum zum Eigenwert λ bezeichnet wird. Ist λ kein Eigenwert von A, so bezeichnet $\mathrm{EV}_\lambda(A)$ den Nullraum. Analoges gilt für B. Insbesondere sind die Eigenwerte von A und B bereits die Eigenwerte von $A \oplus B$. Weiterhin gilt: Ist X ein Eigenvektor von A und Y ein Eigenvektor von B, so sind wegen
>
> $$\begin{aligned} (A \oplus B)\begin{pmatrix} X \\ 0 \end{pmatrix} &= \begin{pmatrix} AX \\ B0 \end{pmatrix} = \begin{pmatrix} \lambda X \\ 0 \end{pmatrix} = \lambda \begin{pmatrix} X \\ 0 \end{pmatrix}, \\ (A \oplus B)\begin{pmatrix} 0 \\ Y \end{pmatrix} &= \begin{pmatrix} A0 \\ BY \end{pmatrix} = \begin{pmatrix} 0 \\ \lambda Y \end{pmatrix} = \lambda \begin{pmatrix} 0 \\ Y \end{pmatrix}. \end{aligned}$$
>
> die Vektoren $\binom{X}{0}$ und $\binom{0}{Y}$ Eigenvektoren von $A \oplus B$. Um bei einem gemeinsamen Eigenwert λ nicht auch Eigenvektoren der Form $\binom{X}{Y}$ berücksichtigen zu müssen, verwenden wir als Basis der Eigenräume möglichst dünnbesetzte Eigenvektoren, welche die Form $\binom{X}{0}$ und $\binom{0}{Y}$ statt etwa $\binom{X}{Y}$ und $\binom{X}{-Y}$ aufweisen.
>
> Daher berechnen wir die Eigenwerte mit zugehörigen Eigenvektoren von $A \oplus B$, indem wir für A und B separate Eigenwertzerlegungen durchführen und die Eigenvektoren anschließend nach obigem Schema mit Nullen auffüllen. Ist weiterhin $A = E_1 D_1 E_1^\top$ und $B = E_2 D_2 E_2^\top$ mit Diagonalmatrizen D_1 und D_2, so gilt:
>
> $$A \oplus B = (E_1 D_1 E_1^\top) \oplus (E_2 D_2 E_2^\top) = (E_1 \oplus E_2) \cdot (D_1 \oplus D_2) \cdot (E_1 \oplus E_2)^\top,$$

wobei $D_1 \oplus D_2$ wiederum eine Diagonalmatrix ist. Insbesondere ist $E_1 \oplus E_2$ ebenfalls eine orthogonale Blockdiagonalmatrix. ◀

Auf die umgeordnete Strukturmatrix A' bezogen genügt es folglich, die entsprechenden Eigenwertzerlegungen auf den isolierten Diagonalblöcken A'_{kk} separat durchzuführen. Die Indizes i dieses Diagonalblocks erfüllen alle die Bedingung $\dot{S}(\pi(i)) = k$, d. h. sie entsprechen genau denjenigen Indizes der ursprünglichen Strukturmatrix A, die mit der k-ten Segmentbezeichnung versehen sind. Folglich können die Einträge eines Eigenvektors nur bei denjenigen Indizes Werte ungleich 0 annehmen, die zu einer Segmentklasse korrespondieren. Weisen die Träger zweier Eigenvektoren einen nichtleeren Schnitt auf, so beschreiben diese denselben Diagonalblock und damit dieselbe Segmentklasse. Es gilt also:

$$\operatorname{supp}(\tilde{e}_i) \cap \operatorname{supp}(\tilde{e}_j) \neq \emptyset \quad \Rightarrow \quad \dot{S}(\pi(i)) = \dot{S}(\pi(j)) \,. \tag{3.3}$$

Im nächsten Schritt werden wir sehen, dass im hier modellierten Fall sogar Äquivalenz vorliegt.

Hintergrund: Das Kroneckerprodukt von Matrizen und die Matrixmultiplikation sind verträglich, denn für wohldefinierte Matrizenprodukte $A \cdot C$ und $B \cdot D$ gilt

$$(A \otimes B) \cdot (C \otimes D) = (A \cdot C) \otimes (B \cdot D). \tag{3.4}$$

Daraus folgt für quadratische Matrizen A und B: Sind $\lambda_1, \ldots, \lambda_m$ die Eigenwerte von A mit zugehörigen Eigenvektoren $X_1, \ldots, X_m$ und sind $\mu_1, \ldots, \mu_n$ die Eigenwerte von B mit zugehörigen Eigenvektoren $Y_1, \ldots, Y_n$, so ist $\lambda_i \cdot \mu_j$ Eigenwert von $A \otimes B$ zum Eigenvektor $X_i \otimes Y_j$, denn mit Gleichung 3.4 ist

$$(A \otimes B) \cdot (X_i \otimes Y_j) = (A \cdot X_i) \otimes (B \cdot Y_j) = (\lambda_i X_i) \otimes (\mu_j Y_j) = \lambda_i \mu_j (X_i \otimes Y_j).$$

Auf diese Weise erhält man alle Eigenwerte von $A \otimes B$. ◀

Der Diagonalblock A'_{kk} hat die Gestalt $\mathbb{1}^{o_k \times o_k} \otimes I^{\varepsilon}_{n_k}$. Also ist hier $A = \mathbb{1}^{o_k \times o_k}$ und $B = I^{\varepsilon}_{n_k}$ eine Tridiagonalmatrix. Nun ermitteln wir die Eigenwerte dieser speziellen Matrizen A und B nebst zugehörigen Eigenvektoren:

Die Matrix A hat den Eigenwert o_k mit Vielfachheit 1, der zugehörige Eigenvektor ist $\mathbb{1}^{o_k \times 1}$. Die weiteren Eigenwerte sind wegen der identischen Zeilen und Spalten alle gleich 0.

Bei der Eigenwertzerlegung von B schreiben wir abkürzend $N := n_k$, also ist $B = I^{\varepsilon}_N$. Die Matrix B stellt eine tridiagonale, symmetrische *Toeplitz-Matrix* dar. Deren Eigenwertzerlegung ist wohlbekannt, siehe beispielsweise [135, 179]. Genauer besitzt B die Eigenwerte

$$\mu_n = 1 + \varepsilon \cdot 2 \cos \tfrac{n\pi}{N+1} \,, \qquad 1 \leq n \leq N$$

mit den zugehörigen Eigenvektoren

$$V_n = \left(\sin \tfrac{n\pi}{N+1} \,,\, \sin \tfrac{2n\pi}{N+1} \;\ldots\,,\, \sin \tfrac{Nn\pi}{N+1}\right)^{\top} .$$

Der k-te Eintrag in V_n verschwindet genau dann, wenn $k \cdot n$ ein Vielfaches von $N+1$ ist. Falls n kein Teiler von $N+1$ ist, sind alle Einträge in V_n ungleich Null. Ebenso gilt: Falls k kein Teiler von $N+1$ ist, sind alle k-ten Einträge in sämtlichen V_n ungleich Null. Für jedes Paar dieser Eigenvektoren weisen insbesondere ihre Träger einen nichtleeren Schnitt auf.

In Verbindung mit Gleichung 3.3 folgt daraus die Äquivalenz zwischen der gleichen punktweisen Bezeichnung von Zeitpunkten des Musikstückes und der Struktur der Nulleinträge in der Matrix $\tilde{E}$ der gewichteten Eigenvektoren:

$$\operatorname{supp}(\tilde{e}_i) \cap \operatorname{supp}(\tilde{e}_j) \neq \emptyset \quad \Leftrightarrow \quad \dot{S}(\pi(i)) = \dot{S}(\pi(j)) \,. \tag{3.5}$$

Abschließend wenden wir diese Ergebnisse auf unsere ursprüngliche Strukturmatrix A an:

Da $A' = P^{-1}AP$, sind die Matrizen A und A' ähnlich und weisen somit dieselben Eigenwerte auf. Für die dazugehörigen Eigenvektoren gilt: Ist y Eigenvektor zu A', so ist Py Eigenvektor zu A. Wegen der Orthogonalität von P gilt $P^{-1} = P^\top$; und da P sogar eine Schiebepermutation ist, sind die Eigenvektoren von A einfach die segmentweise verschobenen Eigenvektoren von A'. Folglich gilt:

$$A = PA'P^\top = PE\,D\,E^\top P^\top = (PE)D(PE)^\top \,.$$

Die j-te Spalte der Matrix $P\tilde{E}$ sei mit pe_j bezeichnet. Somit erhalten wir folgende Variante von Gleichung 3.5:

$$\operatorname{supp}(pe_i) \cap \operatorname{supp}(pe_j) \neq \emptyset \quad \Leftrightarrow \quad \dot{S}(i) = \dot{S}(j) \,. \tag{3.6}$$

Dies zeigt, dass es in der Praxis nicht nötig ist, die Permutation π explizit zu berechnen. Die Eigenwertzerlegung einer pfadverstärkten Selbstähnlichkeitsmatrix liefert direkt die Matrix PE sowie die dazugehörigen Eigenwerte.

3.4.4. Begründung für die Modellierung mit Tridiagonalmatrizen

Bei unserem Ansatz haben wir die Selbstähnlichkeitsmatrix wiederholter Segmente durch Tridiagonalmatrizen der Form I^ε modelliert. Dies korrespondiert zu der Tatsache, dass bei realen Daten die auftretenden Pfadstrukturen oftmals geglättet sind und dadurch eine ähnliche Struktur aufweisen wie I^ε, siehe etwa Abbildung 3.4a.

Weiterhin zeigt eine kurze Überlegung, dass diese Glättung maßgeblich für das Funktionieren dieses Ansatzes ist. Verwenden wir zur Modellierung Einheitsmatrizen anstatt Tridiagonalmatrizen, so erhalten wir die analog zu Gleichung 3.2 definierte Matrix

$$\bigoplus_{k=1}^{K} \left(\mathbb{1}^{o_k \times o_k} \otimes I_{n_k}\right),$$

wobei I_n die Einheitsmatrix der Größe $n \times n$ bezeichnet.

Hier können die Eigenvektoren ebenfalls blockweise berechnet werden. Diese Blöcke bestehen

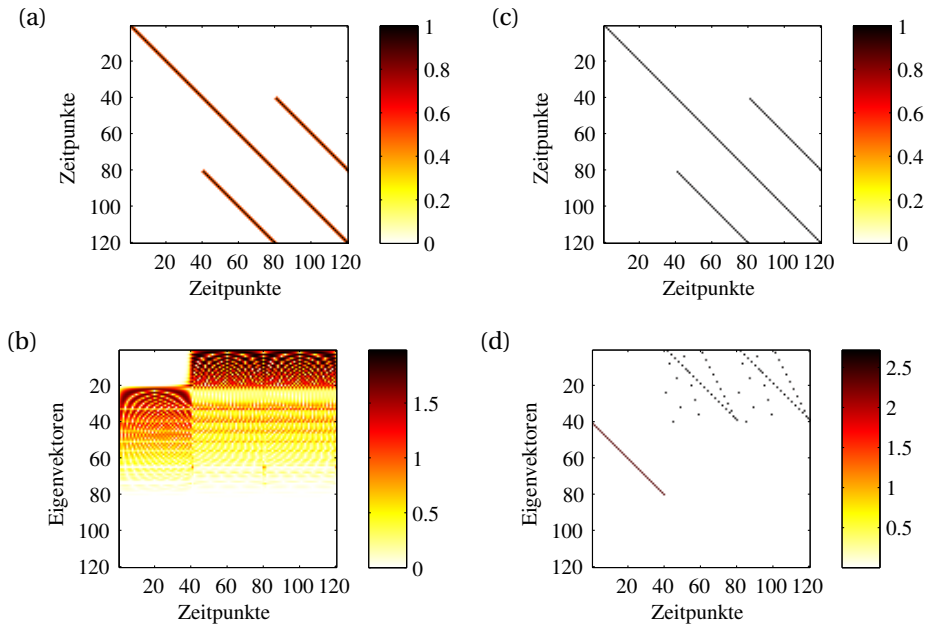


Abbildung 3.7.: Berechnung der Eigenvektoren für die Wiederholungsstruktur ABB. Links die bislang betrachtete Version mit den Tridiagonalmatrizen I^{ε}, rechts die Version mit Einheitsmatrizen.

nun aus gekachelten Einheitsmatrizen, deren Eigenvektor-Matrix die Einheitsmatrix selbst ist, da diese bereits in Diagonalform vorliegt. Somit geht allerdings die Eigenschaft der Eigenvektoren verloren, eine Indikatormatrix für gleiche Bezeichnungen zu bilden, d. h. die Äquivalenz in Gleichung 3.6 geht verloren, da die Eigenvektoren auch an Stellen Nullelemente beinhalten, die durch $\bar{S}$ auf die gleiche Bezeichnung abgebildet werden.

In Abbildung 3.7 wird dieser Effekt anhand der einfachen Segmentierung $\bar{S} \equiv (ABB)$ illustriert. Im Abbildungsteil (a) sehen wir die Modellierung mit den Tridiagonalmatrizen, was zu den bekannten Blockstrukturen in der Matrix der Eigenvektoren (b) führt. Auf der rechten Seite wurden die Tridiagonalmatrizen durch Einheitsmatrizen (c) ersetzt, wodurch in der in Abbildungsteil (d) illustrierten Eigenvektor-Matrix keine Blockstrukturen entstehen können. Somit können die einzelnen wiederholten Segmente nicht mehr erkannt werden und die Darstellung als Eigenvektoren genügt nicht mehr zur Schätzung zusätzlicher Informationen über die Struktur des zugrundeliegenden Musikstückes.

3.4.5. Weitere Eigenschaften

Grenzen der Modellierung

Bislang haben wir angenommen, dass wir eine pfadverstärkte Selbstähnlichkeitsmatrix eines Musikstücks durch eine Strukturmatrix A approximieren können. Hierbei machen wir jedoch

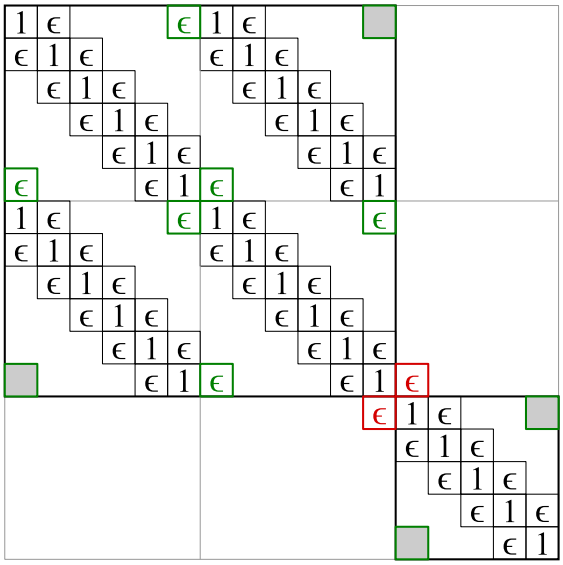


Abbildung 3.8.: Beispielmatrix zur Illustration der Korrekturterme. Die schwarzen Einträge sind durch A' modelliert worden, die grünen Einträge entsprechen den in A^J hinzugefügten Korrekturtermen. Die roten ε-Einträge werden in unserer Modellierung nicht berücksichtigt, treten aber in der Praxis auf.

einen systematischen Fehler, da die Segmentgrenzen bei normaler Glättung bzw. Filterung der Selbstähnlichkeitsmatrix nicht berücksichtigt werden.

Zum Verständnis dieser Abweichung definieren wir die $k \times k$-*Korrekturmatrix* J_k^ε wie folgt:

$$J_k^\varepsilon := \begin{pmatrix} 0 & 0 & \cdots & 0 & \varepsilon \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ \varepsilon & 0 & \cdots & 0 & 0 \end{pmatrix} \tag{3.7}$$

Hiermit lässt sich die in Gleichung 3.2 modellierte Approximation weiter verfeinern:

$$A^J = \left(\bigoplus_{n=1}^{N} \mathbb{1}^{o_n \times o_n} \otimes (I_{\ell_n}^\varepsilon + J_{\ell_n}^\varepsilon) \right) - J_N^\varepsilon,$$

Diese Modellierung wird in Abbildung 3.8 illustriert.

Die Korrekturterme J^ε führen wegen

$$J_k^\varepsilon \cdot X = (\varepsilon X_k, 0, \ldots, 0, \varepsilon X_1)^\top$$

zu keinen wesentlichen Unterschieden zwischen den Eigenwertzerlegungen von A' und A^J, wenn die Mindestlänge θ der Segmente (und damit die Mindestkantenlänge der Kacheln)

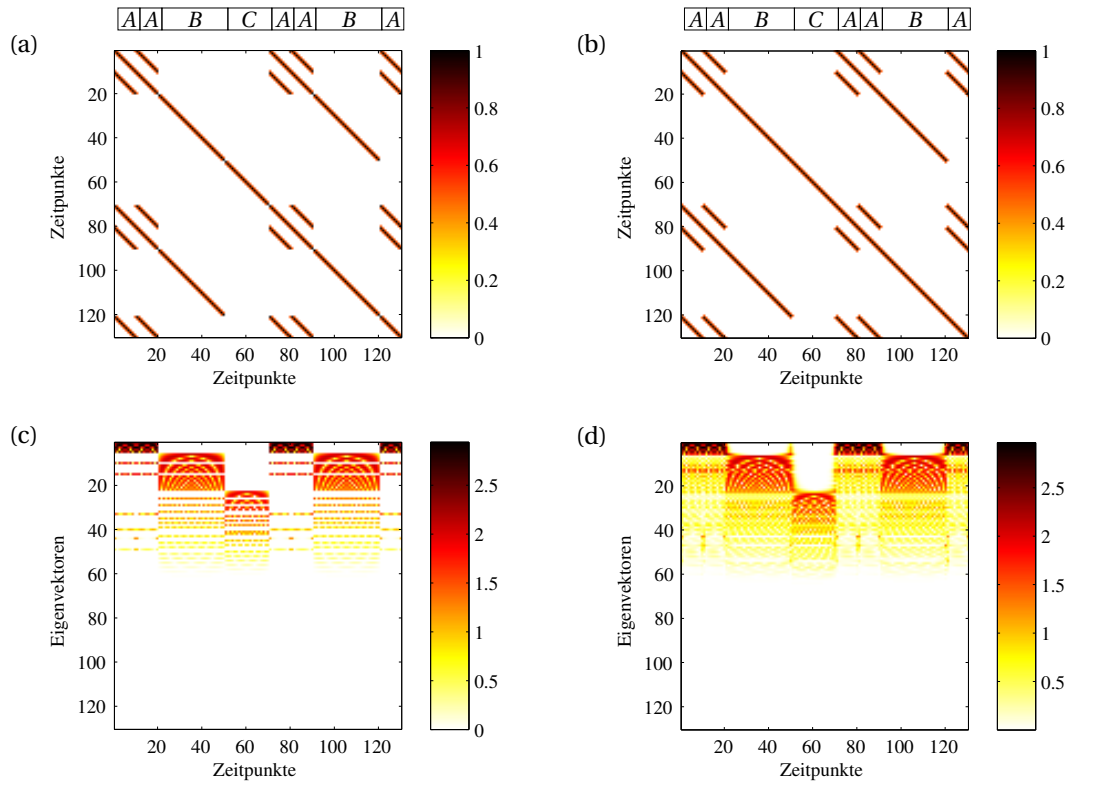


Abbildung 3.9.: Selbstähnlichkeitsmatrizen und dazugehörige Eigenvektoren. Hier ist $\varepsilon = 0{,}5$ gesetzt. **(a)** Theoretisch modellierte Matrix A^J als SSM. **(b)** Tatsächliche SSM bei Vorhandensein der roten ε-Einträge aus Abbildung 3.8, ebenfalls mit $\varepsilon = 0{,}5$. **(c)** Im theoretischen Fall sind die Träger der Eigenvektoren disjunkt. **(d)** Im praktischen Fall treten deutliche Vermischungseffekte auf.

groß genug gewählt wurde, da nur der erste und der letzte Wert jedes Eigenvektors von A^J ungleich 0 ist. Praktische Experimente legen für θ einen Wert von 8 bis 20 Sekunden bei einer Merkmalsauflösung von 10 Hz nahe, wodurch wir davon ausgehen können, dass selbst kleine Segmente mindestens durch eine 80×80-Submatrix dargestellt werden. Also ergeben sich innerhalb der Blöcke aus gleichen Segmenten keine wesentlichen Unterschiede.

Durch die Glättung wird jedoch auch die angenommene Blockstruktur von A^J gestört, was in Abbildung 3.8 durch die beiden roten ε-Einträge dargestellt wird. Durch den Wegfall der angenommen Blockstruktur erhalten wir Überlappungen in den Trägern der Eigenvektoren, die weitreichende Folgen haben:

In Abbildung 3.9a ist eine Matrix abgebildet, bei der die Positionen der beiden roten ε-Einträge in Abbildung 3.8 auf 0 gesetzt werden. Dies entspricht exakt dem in A^J modellierten Fall, die Träger der Eigenvektoren sind disjunkt und segmentieren daher perfekt die einzelnen

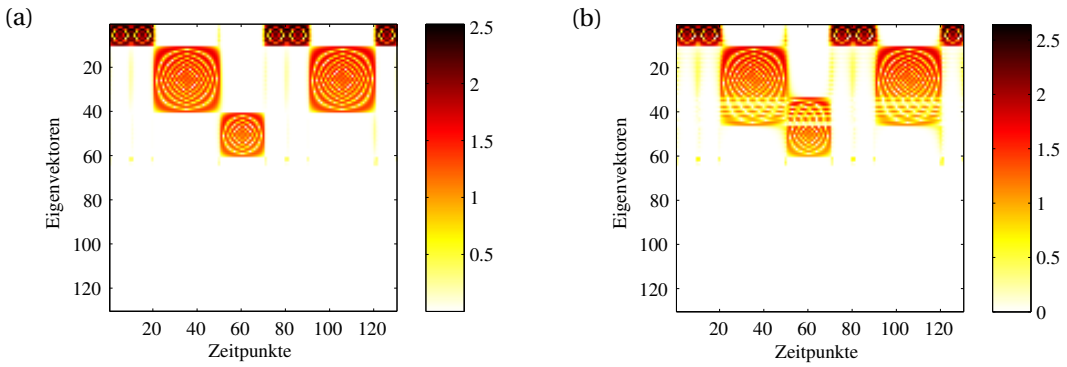


Abbildung 3.10.: Einfluss der ε-Werte auf die berechneten Eigenvektor-Matrizen. **(a)** Bei $\varepsilon = 0{,}1$ stimmen die berechneten Matrizen noch gut mit dem theoretisch vorhergesagten Resultat überein. **(b)** Erste Vermischungen der Eigenvektoren zu verschiedenen Segmentklassen sowie unscharfe Kanten an den Segmentgrenzen treten bei $\varepsilon = 0{,}2$ auf.

Segmentklassen, siehe Abbildungsteil (c). Zum Vergleich ist in Teil (b) die in der Praxis häufig vorkommende Struktur bei Vorhandensein der beiden roten ε-Einträge abgebildet. Obwohl hier die globale Blockmatrix-Struktur nicht mehr vorhanden ist, beschreiben weiterhin einige der Eigenvektoren die Segmente hinreichend gut für eine automatische Segmentierung, siehe Abbildung 3.9d.

Werden die Werte für ε klein genug gewählt, so treten diese Effekte kaum auf, siehe dazu den nächsten Unterabschnitt.

Einfluss des Parameters ε

Die Modellierung mit einem kleinen Wert für ε in Abbildung 3.10a zeigt, dass die theoretisch berechneten Eigenvektoren gut mit den beobachteten Strukturen übereinstimmen und folglich der im vorherigen Abschnitt diskutierte Effekt der in der Modellierung unberücksichtigten ε-Einträge nicht gravierend ist. Die Träger sind zwar nicht völlig disjunkt, allerdings werden in den theoretisch disjunkten Regionen nur vergleichsweise kleine Werte angenommen, wodurch die Strukturen klar erkennbar bleiben. Wird der ε-Wert nun geringfügig erhöht (auf $\varepsilon = 0{,}2$), so vermischen sich die Eigenvektoren bereits deutlich und die Grenzen zwischen den einzelnen Segmenten verschwimmen, siehe Abbildung 3.10b.

Bei den in der Praxis üblicherweise vorkommenden größeren Werten für ε treten die Eigenvektoren in gar keiner sinnvollen Reihenfolge mehr auf, da bei der Eigenwertzerlegung keine Reihenfolge der Basisvektoren berücksichtigt wird und wir daher a priori die normierten Eigenvektoren nach der Größe ihres jeweiligen Eigenwertes sortieren. Da diese für die Gesamtmatrix die Gestalt

$$\mu_n = \#\text{rep} \cdot \left(1 + \varepsilon \cdot 2\cos\tfrac{n\pi}{N+1}\right)$$

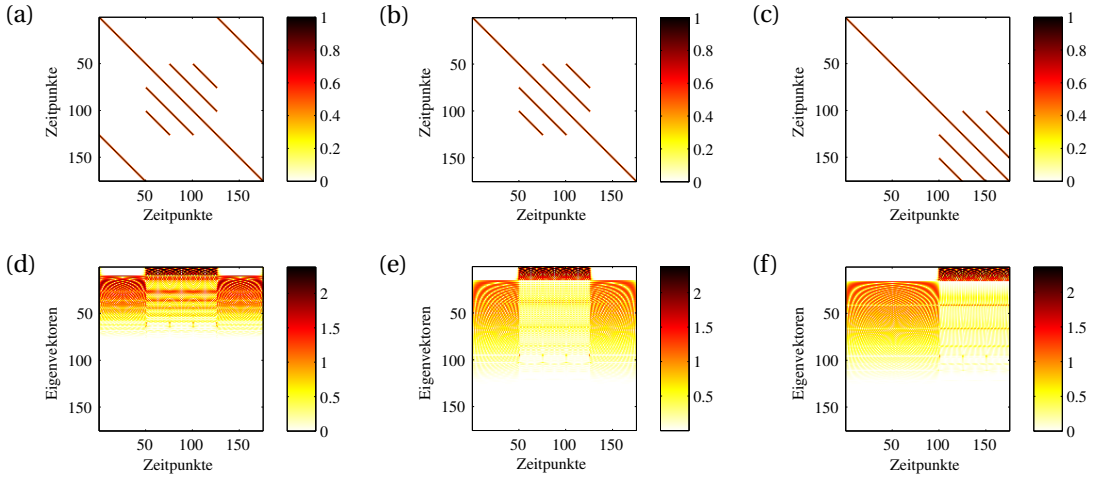


Abbildung 3.11.: Drei ähnliche Wiederholungsstrukturen mit dazugehörigen Eigenvektoren: **(a),(d)** Das erste Segment wird am Ende wiederholt ($ABBBA$), **(b),(e)** am Anfang und am Ende stehen jeweils einmalig auftretende Segmente ($ABBBC$), **(c),(f)** zum Vergleich die unzulässige Wiederholungsstruktur $ACBBB$ (entspricht $A'BBB$).

haben, wobei #rep die Anzahl der Wiederholungen des entsprechenden Segments bezeichnet, dominiert die Anzahl der Wiederholungen die Sortierung nur bei kleinen Werten für ε, bei größeren Werten hingegen nicht mehr. Daher verwenden wir bei unserem Algorithmus ein *k-means*-Clusteringverfahren, um eine Sortierung der Eigenvektoren zu erreichen, bei der die Träger benachbarter Vektoren möglichst große Schnittmengen aufweisen.

Segmente ohne Wiederholungen

Der in Abschnitt 3.4.3 ausführlich beschriebene Permutationsansatz ist auch die Erklärung für den bereits erwähnten Effekt, dass das Verfahren nicht zwischen mehreren nur einmal auftretenden Segmenten unterscheiden kann. Bei unserer Modellierung übergehen wir die Tatsache, dass ein Segment nicht zerschnitten werden kann, was zur Folge hat, dass alle einmalig auftretenden Segmente als Bestandteile eines einzigen Segmentes aufgefasst werden. Da unsere Modellierung eine minimale Anzahl von Segmentlabeln vorsieht, werden von diesem Ansatz alle nicht-wiederholten Segmente zu einer Segmentklasse zusammengefasst.

In Abbildung 3.11 ist dieses Phänomen mittels einiger Beispiele illustriert. Im ersten Beispiel wird das erste Segment am Ende wiederholt, was von unserem Verfahren korrekt erkannt wird. Im zweiten Beispiel steht am Ende ein neues Segment, sodass das erste und letzte Segment jeweils nicht wiederholt wird. Diese beiden Segmente werden durch das Verfahren ebenfalls zu einem Segment zusammengefasst. Die obige Erklärung dieses Phänomens wird durch das dritte Beispiel illustriert, das eine segmentweise Permutation des zweiten Beispiels darstellt. Hier treten die beiden unwiederholten Segmente A und C direkt hintereinander auf

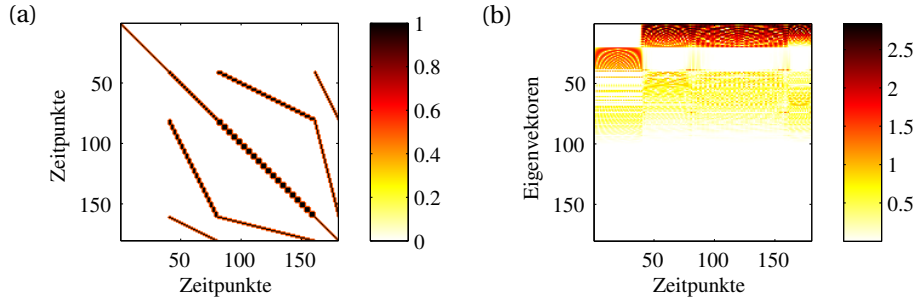


Abbildung 3.12.: Wiederholungsstruktur $AB_1B_2B_3$, wobei B_2 und B_3 Wiederholungen von B_1 in halbem bzw. doppeltem Tempo darstellen. **(a)** Selbstähnlichkeitsmatrix, **(b)** daraus abgeleitete Eigenvektoren.

(was eine Verletzung unserer Zulässigkeitsbedingung in Abschnitt 2.5 darstellt), wodurch die Interpretation als $A'BBB$ unseres Programms deutlich wird.

Eine mögliche Lösung für dieses Problem ist die manuelle Verlängerung der Selbstähnlichkeitsmatrix um ein zusätzliches einmaliges Segment. Anschließend werden Eigenvektorzerlegung, Berechnung der homogenen Selbstähnlichkeitsmatrix und anschließende sparse-NMF-Segmentierung auf dieser erweiterten Selbstähnlichkeitsmatrix durchgeführt. Das zu dieser künstlichen Erweiterung korrespondierende Segment beschreibt dann die Klasse der einmalig auftretenden Segmente und kann zu deren Identifikation genutzt werden.

Wiederholungen mit abweichendem Tempo

Bislang sind wir davon ausgegangen, dass alle Wiederholungen durch Diagonalstrukturen modelliert werden können, was musikalisch gesehen einer exakten Wiederholung in exakt demselben Tempo entspricht. Obwohl die Beibehaltung eines absolut exakten Tempo sehr schwierig und oftmals künstlerisch auch nicht gewünscht ist, können wir annehmen, dass viele Wiederholungen in der Praxis durchaus in annähernd gleichem Tempo auftreten. Dennoch ist die Variation des Tempos auch ein musikalisches Stilmittel, das bei der Suche nach Wiederholungen nicht außer Acht gelassen werden sollte.

Eine solche Tempoabweichung zeigt sich in der Selbstähnlichkeitsmatrix als Pfadstruktur, die nicht exakt einer Diagonallinie entspricht, sondern (bei konstantem Tempo) einem Geradensegment in leicht veränderter Richtung gleicht oder (bei gleichmäßig variiertem Tempo) eine bogenförmige Struktur aufweist. Unabhängig von der tatsächlichen Form werden diese von unserem Ansatz als Wiederholungen erkannt, vergleiche die Illustration in Abbildung 3.12. Die hier abgebildete synthetische Wiederholungsstruktur besteht aus einem nicht-wiederholten A-Teil gefolgt von drei B-Teilen $B_1B_2B_3$, wobei B_2 eine Wiederholung von B_1 in halbem Tempo und B_3 eine Wiederholung in doppeltem Tempo von B_1 darstellt, folglich weist B_3 ein vierfach schnelleres Tempo als B_2 auf. Wie in Abbildung 3.12b illustriert wird, spiegelt sich dies auch in

den entsprechenden Einträgen der Eigenvektoren wieder, was insbesondere bei dem zu B_2 korrespondierenden Teil der Eigenvektor-Matrix deutlich erkennbar ist.

Zum Verständnis dieses Phänomens betrachten wir ein Stück mit der Struktur $A_1 A_2$, bei dem die Wiederholung A_2 in einem um einen rationalen Faktor $\frac{p}{q}$ ($p, q \in \mathbb{N}$) abweichenden Tempo gespielt wird. Bei gleicher Merkmalsauflösung beschreiben somit die p Zeitpunkte des Segments A_1 denselben musikalischen Inhalt wie die q Zeitpunkte der Wiederholung A_2. Im Folgenden nehmen wir $p \leq q \leq 2p$ an. Weiterhin begnügen wir uns mit der Betrachtung des vereinfachten Falls, dass die synthetische Selbstähnlichkeitsmatrix mit $\varepsilon := 0$ modelliert wird. Folglich erhalten wir die Blockmatrix

$$\mathcal{M}_{p,q} := \begin{pmatrix} I_p & I_{p,q} \\ I_{q,p} & I_q \end{pmatrix},$$

wobei mit I_n die $n \times n$-Einheitsmatrix und mit $I_{p,q}$ eine $p \times q$-Binärmatrix bezeichnet wird, deren Einsen alle auf einem von (1,1) nach (p, q) verlaufendem Geradensegment stehen, wobei jede Spalte nur eine Eins aufweist, etwa

$$I_{2,3} := \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{und} \quad I_{3,4} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Hierdurch sind $d := q - p$ Spalten Duplikate einer ihrer Nachbarspalten, wobei wir annehmen, dass keine Spalte mehr als zweimal vorkommt. Man beachte, dass an dieser Stelle nicht konkret definiert wird, welche Spalten doppelt auftreten, da dies für die Berechnung der Eigenwerte nicht relevant ist. Analog weist $I_{q,p} := I_{p,q}^{\top}$ genau d jeweils doppelt auftretende Zeilen auf.

Da $\mathcal{M}_{p,q}$ eine Blockmatrix mit invertierbaren Submatrizen auf der Diagonale ist, können ihre Eigenwerte λ mithilfe des Satzes über Determinanten von Blockmatrizen (»Kästchensatz«, siehe etwa [54, S. 137 f]) mittels

$$0 = \det(\mathcal{M}_{p,q} - \lambda I_{p+q}) = \det\big((1-\lambda) I_q\big) \cdot \det\big((1-\lambda) I_p - I_{p,q}((1-\lambda) I_q)^{-1} I_{q,p}\big)$$

berechnet werden. Da $I_{p,q} \cdot I_{q,p}$ eine $p \times p$-Diagonalmatrix ist, deren Diagonale aus genau d Zweien und $p - d$ Einsen besteht, folgt daraus

$$0 = (1-\lambda)^d \cdot \big((1-\lambda)^2 - 2\big)^d \cdot \big((1-\lambda)^2 - 1\big)^{p-d},$$

also erhalten wir die Eigenwerte 1, $1 + \sqrt{2}$ und $1 - \sqrt{2}$ jeweils mit Vielfachheit d sowie 0 und 2 mit Vielfachheit $p - d$. Im allgemeineren Fall $k \cdot p \leq q \leq (k+1) \cdot p$ für $k \in \mathbb{N}$ und mit $d := q - k \cdot p$ ergeben sich die Eigenwerte 1 und $1 \pm \sqrt{k+1}$ mit Vielfachheit d und $1 \pm \sqrt{k}$ mit Vielfachheit $p - d$.

Zur Illustration der Eigenschaften der Eigenvektoren berechnen wir diese für $\mathcal{M}_{2,3}$ explizit und vergleichen sie mit den Eigenvektoren von $\mathcal{M}_{2,2}$. Die beiden betrachteten Matrizen haben

die folgende Gestalt:

$$\mathcal{M}_{2,3} := \left(\begin{array}{cc|cc} 1 & & \mathbf{1} & \mathbf{1} \\ & 1 & & 1 \\ \hline \mathbf{1} & & \mathbf{1} & \\ \mathbf{1} & & & \mathbf{1} \\ & 1 & & 1 \end{array}\right) \quad \text{sowie} \quad \mathcal{M}_{2,2} := \left(\begin{array}{cc|cc} 1 & & \mathbf{1} & \\ & 1 & & 1 \\ \hline \mathbf{1} & & \mathbf{1} & \\ & 1 & & 1 \end{array}\right),$$

wobei uns die Eigenschaften der Eigenvektoren an den fett gedruckten Positionen interessieren, die zur modellierten zeitlichen Streckung korrespondieren. Mit $B_\lambda(A)$ sei eine Basis des Eigenraums von A zum Eigenwert λ bezeichnet. Da alle Eigenwerte von $\mathcal{M}_{2,3}$ einfach auftreten, erhalten wir als Basis beispielsweise die folgenden Eigenvektoren:

$$\begin{aligned} B_{1-\sqrt{2}}(\mathcal{M}_{2,3}) &= \left\{\begin{pmatrix} -\sqrt{2} & 0 & \mathbf{1} & \mathbf{1} & 0 \end{pmatrix}^\top\right\}, \\ B_0(\mathcal{M}_{2,3}) &= \left\{\begin{pmatrix} 0 & 1 & \mathbf{0} & \mathbf{0} & -1 \end{pmatrix}^\top\right\}, \\ B_1(\mathcal{M}_{2,3}) &= \left\{\begin{pmatrix} 0 & 0 & \mathbf{1} & \mathbf{-1} & 0 \end{pmatrix}^\top\right\}, \\ B_2(\mathcal{M}_{2,3}) &= \left\{\begin{pmatrix} 0 & 1 & \mathbf{0} & \mathbf{0} & 1 \end{pmatrix}^\top\right\}, \\ B_{1+\sqrt{2}}(\mathcal{M}_{2,3}) &= \left\{\begin{pmatrix} \sqrt{2} & 0 & \mathbf{1} & \mathbf{1} & 0 \end{pmatrix}^\top\right\}. \end{aligned}$$

Bei $\mathcal{M}_{2,2}$ weisen die Eigenwerte 0 und 2 beide Vielfachheit 2 auf, wir erhalten die entsprechenden Eigenvektoren

$$\begin{aligned} B_0(\mathcal{M}_{2,2}) &= \left\{\begin{pmatrix} -1 & 0 & \mathbf{1} & 0 \end{pmatrix}^\top, \begin{pmatrix} 0 & 1 & \mathbf{0} & -1 \end{pmatrix}^\top\right\}, \\ B_2(\mathcal{M}_{2,2}) &= \left\{\begin{pmatrix} 0 & 1 & \mathbf{0} & 1 \end{pmatrix}^\top, \begin{pmatrix} 1 & 0 & \mathbf{1} & 0 \end{pmatrix}^\top\right\}. \end{aligned}$$

Beim Vergleich dieser beiden Basen stellen wir fest, dass die Vektoren aus $B_0(\mathcal{M}_{2,2})$ strukturell den Vektoren aus $B_0(\mathcal{M}_{2,3}) \cup B_{1-\sqrt{2}}(\mathcal{M}_{2,3})$ entsprechen und ebenso $B_2(\mathcal{M}_{2,2})$ der Menge $B_2(\mathcal{M}_{2,3}) \cup B_{1+\sqrt{2}}(\mathcal{M}_{2,3})$ entspricht. Der verbleibende Vektor $B_1(\mathcal{M}_{2,3})$ findet als Entsprechung in $B(\mathcal{M}_{2,2})$ den Nullvektor. Die Unterschiede treten ausschließlich in den fett gedruckten Komponenten auf und korrespondieren somit zu der in Abbildung 3.12b visualisierten »zeitlichen Streckung« der Eigenvektoren.

3.5. Evaluation und Experimente

Um zu zeigen, wie sich unser Konvertierungsansatz auf echten Daten verhält, diskutieren wir nun eine Reihe expliziter Beispiele (Abschnitt 3.5.1) und führen einige quantitative Experimente durch (Abschnitt 3.5.2). Man beachte, dass die Untersuchung und Optimierung der speziellen Rolle der verschiedenen Parameter nicht das Ziel dieser Experimente ist. Unser Hauptziel liegt nicht in der numerischen Verbesserung eines spezifischen Strukturanalyse-Resultats,

sondern im Hervorheben der konzeptionellen Neuheit dieses Ansatzes. Insbesondere zeigen wir, dass die für homogenitätsbasierte Strukturanalyse entwickelten Methoden wie das in Abschnitt 2.6.1 beschriebene sparse-NMF-Verfahren nun auch für wiederholungsbasierte Strukturanalyse nutzbar sind.

In unseren Experimenten verwenden wir diese NMF-Variante mit zusätzlichen Bedingungen an die Dünnbesetztheit der Aktivierungsmatrix [83], wobei wir den entsprechenden sNMF-Parameter β wieder auf $4 \cdot \text{mean}(\mathcal{S}^{\text{Block}})$ setzen und den Rangparameter auf 6, wodurch wir maximal sechs verschiedene musikalische Segmentklassen beschreiben können. Diese Methode wenden wir dann auf die Matrix $\mathcal{S}^{\text{Block}}$ an.

3.5.1. Qualitative Evaluation

Wir beginnen mit der Diskussion einiger spezieller Beispiele zur Demonstration sowohl des Potentials als auch der Grenzen unseres Konvertierungsansatzes. In den hierzu verwendeten Abbildungen zeigen wir die als Eingabe für unseren Konvertierungsansatz dienende Pfadmatrix als Überlagerung der daraus berechneten Blockstrukturmatrix $\mathcal{S}^{\text{Block}}$. Ebenfalls zeigen wir die aus $\mathcal{S}^{\text{Block}}$ mittels *sparse NMF* (vgl. Abschnitt 2.6.1) abgeleitete Strukturannotation (obere Strukturierung) sowie als Vergleich die manuell generierte Referenzannotation (untere Segmentierung). Bei allen Abbildungen ist die Zeit jeweils in Sekunden angegeben.

Wir beginnen mit unserem Hauptbeispiel, dem vierten *Pomp and Circumstance*-Marsch von Edward Elgar, illustriert in Abbildung 3.13. Verwenden wir die Pfadmatrix $\mathcal{S}$ wie in Abbildung 3.4a dargestellt als Eingabe, so produziert unsere Konvertierungsmethode die Blockstrukturmatrix $\mathcal{S}^{\text{Block}}$, dargestellt in Abbildung 3.4d. Wie Abbildung 3.13a illustriert, werden die Pfadstrukturen wiederholter Segmente korrekt in Blockstrukturen überführt. Weiterhin zeigt Teil a dieser Abbildung die aus $\mathcal{S}^{\text{Block}}$ abgeleitete Strukturierung und die bereits in Kapitel 2 vorgestellte, manuell generierte Strukturannotation. Hierbei wird deutlich, dass der homogenitätsbasierte, auf $\mathcal{S}^{\text{Block}}$ angewendete Clustering-Ansatz ein nahezu perfektes wiederholungsbasiertes Strukturanalyse-Resultat ergibt, obwohl die Pfadmatrix einige Störungen und Inkonsistenzen aufweist. Nur direkt aufeinander folgende Segmente wie die beiden C-Teile $C_1 C_2$ sowie deren weitere Unterteilung (die deutlich durch Pfadstrukturen erkennbar sind), können von diesem punktweisen Segmentierungsansatz nicht erkannt werden, vergleiche dazu die Anmerkungen in Abschnitt 2.5.

Im Allgemeinen zeigt unser Verfahren bessere Resultate, wenn die als Eingabe verwendete Pfadstrukturmatrix möglichst dünnbesetzt ist. Daher sind die in Abschnitt 3.2 vorgestellten Schritte zur Glättung, Rauschunterdrückung und Schwellwertberechnungen nötig, um die Pfadstruktur ausreichend zu verstärken, wie es auch in den meisten wiederholungsbasierten Strukturanalyseverfahren geschieht [28, 147].

Wie bereits erwähnt, ist die Ausgabe des Algorithmus stark von dem gewählten Wert für den Glättungsparameter abhängig, der für die Verstärkung der Pfadstrukturen eine wesentliche Rolle spielt. In Abbildung 3.13a wurden hierfür 8 Sekunden gewählt. Teil (b) dieser Abbildung

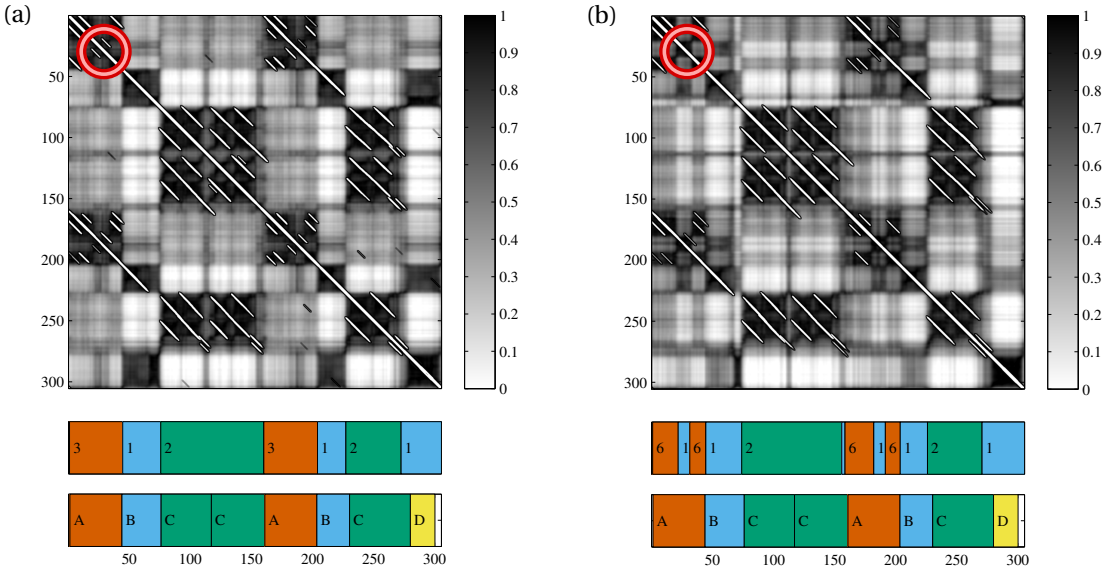


Abbildung 3.13.: Einfluss des Glättungsparameters auf die finalen Segmentierungen beim Elgar-Beispiel. **(a)** Bei einer Fensterlänge von 8 Sekunden kann die manuelle Grobsegmentierung zu sehr großen Teilen rekonstruiert werden. **(b)** Bei 12 Sekunden kann die Wiederholung des zweiten Themas im A-Teil nicht mehr gefunden werden, wodurch der A-Teil zerfällt.

zeigt das Ergebnis bei Verwendung von 12 Sekunden. Die beiden Pfadmatrizen unterscheiden sich hauptsächlich durch das Fehlen der Wiederholung des zweiten Themas innerhalb des A-Teils (vergleiche hierzu auch die Feinstruktur in Abbildung 2.1). Da bei beiden Parameterisierungen die Feinstruktur des B-Teils nicht durch einen Pfad abgebildet wird, erkennt das Verfahren in der in Abbildung 3.13b vorliegenden Situation den Mittelteil des A-Teils als zum B-Teil zugehörig und benennt folglich beide Segmente gleich.

Als nächstes betrachten wir die vier in Abbildung 3.14 illustrierten Beispiele. Das erste, in Abbildung 3.14a gezeigte Beispiel ist eine Aufnahme des Instrumentalstückes *Ungarischer Tanz Nr. 5* von Johannes Brahms. Es weist eine ternäre Struktur auf, wobei die Randsegmente in zwei Untersegmente zerfallen, die wir in der Referenzannotation als A und B bezeichnen. Diese Segmente sind in der Blockstrukturmatrix deutlich erkennbar. Weiterhin treten starke Tempounterschiede innerhalb der B-Segmente auf (B_2 und B_4 werden etwa doppelt so schnell gespielt wie B_1 und B_3), die durch Berücksichtigung mehrerer Tempi bei der pfadverstärkenden Vorverarbeitung zu den musikalisch sinnvollen Blockstrukturen führt. Wie auch im vorherigen Elgar-Beispiel werden die direkt wiederholten A- bzw. B-Segmente nicht separiert.

Unser zweites Beispiel ist eine Aufnahme des ABBA-Songs *The winner takes it all*, siehe Abbildung 3.14b. In diesem Beispiel möchten wir zeigen, dass fehlende oder verkürzte Pfadstrukturen, wie sie insbesondere beim letzten A-Segment zu beobachten sind, dennoch in der

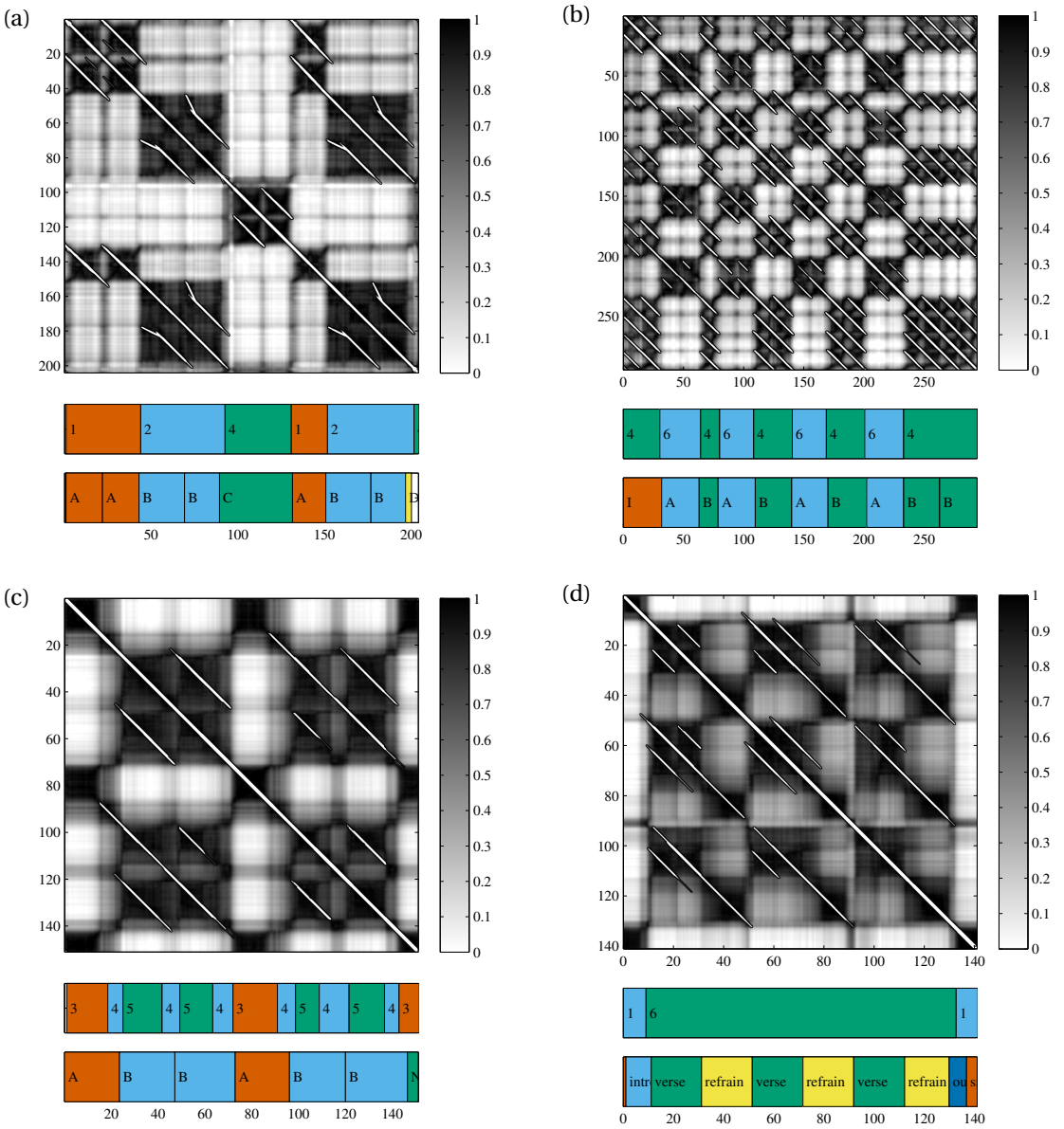


Abbildung 3.14.: Ergebnisse für vier verschiedene Musikstücke: **(a)** *Ungarischer Tanz Nr. 5* von Johannes Brahms, **(b)** der Song *The winner takes it all* von ABBA, **(c)** die *Bayernhymne* (2 Strophen, a-capella), **(d)** der Song *Help!* von The Beatles.

Blockstruktur »korrigiert« werden. Da die Zerlegung in Eigenvektoren eine *globale* Analyse der gesamten Matrix $\mathcal{S}$ darstellt, werden lokale Abweichungen und fehlende Verbindungen ausbalanciert, wodurch eine Art Transitivität in der Blockstrukturmatrix realisiert wird. Weiterhin zeigt dieses Beispiel, dass selbst die hier vorliegende, bis auf Segmentgrenzen perfekte wiederholungsbasierte Struktur nicht alleine dazu geeignet ist, die Referenzannotation adäquat zu beschreiben. Da die mit I gekennzeichnete *Introduction* des Musikstückes eine rein instrumentale Version des im Folgenden mit B gezeichneten Refrains darstellt, werden von unserem Verfahren die I- und B-Teile mit derselben Bezeichnung versehen.

Im nächsten Beispiel (Abbildung 3.14c) wird eine Chorversion der *Hymne des Freistaates Bayern* (»Bayernhymne«) analysiert. Diese besteht aus zwei Strophen, von denen jeweils der zweite Teil wiederholt wird; es ergibt sich also für jede der beiden Strophen eine ABB-Struktur. Eine Schwierigkeit für unser Verfahren liegt darin, dass der Text der beiden Strophen nicht identisch ist. Dies trägt neben anderen musikalischen Gründen dazu bei, dass nur die Wiederholung des letzten Teiles der A-Segmente erkannt wird, wohingegen der überwiegende Teil der A-Segmente als einmalig vorkommendes Material behandelt wird. Als Folge davon können wir bei den B-Teilen eine Übersegmentierung beobachten, die ein typisches Phänomen automatischer Strukturanalysen darstellt [104].

Abschließend stellen wir in Abbildung 3.14d mit dem Beatles-Song *Help!* ein Beispiel für eine typische Schwäche des sparse-NMF-Verfahrens vor. In der Pfadstruktur ist deutlich zu erkennen, dass die zwei Segmentklassen »verse« und »refrain« durch eine wiederholungsbasierte Strukturierung erfasst werden können. Auch in der durch unser Verfahren berechneten Blockstrukturmatrix sind markante Blockstrukturen erkennbar, die zwar etwa die erste Hälfte der refrain-Segmente dem jeweilig vorangehenden verse-Segment zuschlagen wollen, aber ansonsten die musikalische Struktur gut widerspiegeln. Durch die Wahl eines zu strikten *sparsity*-Parameters sowie eine unglücklich gewählte[5] Initialisierungsmatrix für das NMF-Verfahren werden jedoch beide Segmentklassen zusammengefasst. Das Ergebnis ist eine untersegmentierte, nahezu unbrauchbare Schätzung der musikalischen Struktur.

3.5.2. Quantitative Evaluation

Abschließend haben wir unser Verfahren auch quantitativ auf einigen der in Abschnitt 2.7 vorgestellten Datensätzen ausgewertet und die Ergebnisse mit anderen Verfahren verglichen. Als Evaluationsmaße wurden die bereits beschriebenen Standardmaße Precision $\boldsymbol{P}$, Recall $\boldsymbol{R}$ und F-measure $\boldsymbol{F}$ sowohl für den paarweisen Vergleich der Segmentbenennungen zu bestimmten Zeitpunkten als auch für die Segmentgrenzen (mit der üblichen Toleranz von 3 Sekunden) verwendet. Bedingt durch die Verwendung des homogenitätsbasierten Segmentierungsansatzes können Segmentgrenzen zwischen Segmenten gleicher Benennung nicht gefunden werden (siehe dazu auch Abschnitt 2.5), wodurch wir schwächere Ergebnisse bei der Evaluation der Segmentgrenzen erwarten.

[5] Beim NMF-Verfahren wird die Initialisierungsmatrix zufällig gewählt. Bei unseren Experimenten haben wir den MATLAB-Zufallszahlengenerator fixiert, um wiederholbare Ergebnisse zu erhalten.

Für [62] haben wir die Beatles-Songs mit den TUT-Annotationen[6], verwendet und aus den 2792 Einzelaufnahmen des Mazurka-Datensatzes[7] mit manuell generierten Strukturannotationen die drei vollständigen (d. h. alle 49 Stücke umfassenden) Serien der Pianisten *Artur Rubinstein* (1966), *Patrick Cohen* und *Masako Ezaki* ausgewählt. Für den Vergleich haben wir neben den Ergebnissen aus [79, 146] auch das auf [174] basierende und mit SMGA bezeichnete Verfahren verwendet, das bei der MIREX2012-Auswertung[8] das höchste Ergebnis erzielen konnte. Für SMGA haben wir die Ergebnisse für zwei verschiedene Parametereinstellungen verglichen, die zum besten (+) und zum schlechtesten (-) Ergebnis geführt haben. Die Zahlen hierzu stammen aus der Literatur. Für das vorgestellte Konvertierungs-Verfahren wählten wir eine feste Fensterlänge von 12 s und berücksichtigten keine Transponierung. Das Verfahren mit diesen Einstellungen bezeichnen wir mit $\mathbf{Konv}_1$.

Im Anschluss daran haben wir eine verbesserte Version des Algorithmus' zur Pfadverstärkung entwickelt, bei der sich andere Parameterbereiche als zielführender erwiesen haben: In den weiteren Experimenten verwendeten wir eine Fensterlänge von $0{,}65 \cdot \sqrt{T}$ Sekunden, wobei T für die Gesamtlänge des Stückes (in Sekunden) steht. Der Pfadverstärkung liegen nun transpositionsinvariante Selbstähnlichkeitsmatrizen zugrunde, die zum Zwecke schnellerer Berechnungen auf eine Größe von 600×600 skaliert wurden. Dieses Verfahren bezeichnen wir im Folgenden mit $\mathbf{Konv}_2$.

Bei dem als *Baseline* beschriebenen Verfahren wird jedes Stück als ein Segment interpretiert und dies gegen die Referenzannotation(en) ausgewertet. Dies führt bei der paarweisen Betrachtung der Segmentbezeichnungen zu einem konstanten Recall-Wert von 1 sowie bei den Segmentgrenzen zu einem Precision-Wert von 1, da die erste und letzte Segmentgrenze des Stückes fast sicher gefunden wird. Hierzu siehe auch die Erläuterungen zu den grundlegenden Eigenschaften der verwendeten Evaluationsmaße in Abschnitt 2.7. Bei den Segmentgrenzen ergeben sich auf dem Mazurka-Datensatz Abweichungen zu diesem Idealfall, da durch die automatische Synchronisation die Position der letzten Segmentgrenze nicht exakt mit dem Ende des Stückes übereinstimmen muss.

Tabelle 3.1 zeigt die Ergebnisse sowohl bei Verwendung der ursprünglichen Methode zur Pfadglättung als auch des verbesserten Verfahrens. Man beachte, dass wir einen ähnlichen NMF-basierten Segmentierungsansatz wie in [79] verwendet haben, allerdings im Unterschied zu dem dortigen Verfahren nicht auf einer Selbstähnlichkeitsmatrix $\mathcal{S}$ mit Blockstrukturen, sondern auf unserer konvertierten Blockstrukturmatrix $\mathcal{S}^{\text{Block}}$. Dies führte zu einer deutlichen Verbesserung gegenüber [79] auf dem Beatles-Datensatz bezüglich der paarweisen Evaluationsmaße. Im Vergleich mit den SMGA-Ergebnissen liegen wir bezüglich dieser Werte in einem ähnlichen Bereich, wobei wir für die Segmentgrenzen deutlich schlechtere Werte erzielt haben. Dies ist wenig überraschend, da unser Ansatz als ein rein homogenitätsbasiertes und nur punktweise agierendes Verfahren keine optimierte Erkennung der Segmentgrenzen beinhaltet. Im Gegensatz dazu werden beim SMGA-Verfahren zuerst Kandidaten für Segmentgrenzen

[6] `http://www.cs.tut.fi/sgn/arg/paulus/structure.html`
[7] `http://www.mazurka.org.uk`
[8] `http://nema.lis.illinois.edu/nema_out/mirex2012/results/struct/mrx09/`

Datensatz	Methode	Segmentbezeichnungen			Segmentgrenzen (3 s)		
		F	***P***	***R***	***F***	***P***	***R***
BeatlesTUT	**Konv**$_1$	0,68	0,714	0,688	0,614	0,58	0,695
	Konv$_2$	0,675	0,656	0,753	0,578	0,632	0,571
	[79]	0,608	0,615	0,646	N/A	N/A	N/A
	[146]	0,599	0,729	0,546	N/A	N/A	N/A
	SMGA(-)	0,658	0,709	0,659	0,696	0,681	0,729
	SMGA(+)	0,718	0,651	0,8	0,753	0,734	0,791
	Baseline	0,599	0,443	0,997	0,293	0,819	0,182
Mazurka49-Rub	**Konv**$_1$	0,723	0,701	0,787	0,606	0,663	0,605
Mazurka49-Coh	**Konv**$_1$	0,70	0,693	0,742	0,627	0,653	0,659
Mazurka49-Eza	**Konv**$_1$	0,714	0,69	0,774	0,644	0,707	0,641
Mazurka2792	**Konv**$_2$	0,712	0,686	0,791	0,574	0,696	0,528
	SMGA(-)	0,681	0,752	0,652	0,659	0,703	0,653
	SMGA(+)	0,719	0,758	0,716	0,692	0,724	0,695
	Baseline	0,504	0,351	1	0,233	0,734	0,141

Tabelle 3.1.: Ergebnisse der automatischen Strukturanalyse unseres Verfahrens **Konv** im Vergleich zu anderen Methoden.

bestimmt und diese Informationen anschließend zum Schätzen geeigneter Segmentbezeichnungen genutzt. Das Verfahren **Konv**$_2$ haben wir wie das Vergleichsverfahren SMGA auf allen 2792 Aufnahmen des Mazurka-Datensatzes ausgewertet, was auch die drei Versionen in den Einzelexperimenten mit der Konfiguration **Konv**$_1$ umfasst. Sowohl für den vollständigen Datensatz als auch für die drei ausgewählten Pianisten erzielten wir Evaluationswerte, die mit denen für SMGA erreichten Werten vergleichbar sind. Die *Baseline*-Ergebnisse werden von allen Verfahren deutlich übertroffen.

In einem zweiten Schritt verwendeten wir die ein breiteres Spektrum musikalischer Stilrichtungen abdeckenden Datensätze *Isophonics* und *SALAMI* für eine umfassendere Analyse.

Die Ergebnisse für *Isophonics* sind in Tabelle 3.2 dargestellt. Das in [78] vorgestellte Verfahren verwendet einen Fusionsansatz zur Kombination von Pfad- und Blockstrukturen, bei dem sowohl Novelty-basierte Schätzungen von Segmentgrenzen als auch wiederholungsbasierte Informationen verwendet werden. Mit (Wdh.) sind die Evaluationswerte für das ausschließlich wiederholungsbasierte Verfahren gekennzeichnet, mit (Fusion1) und (Fusion2) die Ergebnisse bei Verwendung der beiden dort diskutierten Fusionsansätze. Wie beim vorher diskutierten Vergleich erreichen wir auch hier beim paarweisen Vergleich der Segmentbezeichnungen ähnliche Werte, bei der Auswertung der Segmentgrenzen fallen die von unserem Verfahren erzielten Werte wiederum stark ab.

Bei der Aufteilung der Ergebnisse nach Interpreten wird deutlich, dass die auf den Beatles-Songs erzielten Ergebnisse die der anderen Künstler übertreffen. Dies stimmt mit der in [181] vorgestellten Analyse der Performance mehrerer Verfahren auf dem *Isophonics*-Datensatz

Methode	Interpret	#Songs	Segmentbezeichnungen			Segmentgrenzen (3 s)		
			F	***P***	***R***	***F***	***P***	***R***
Konv$_2$	Carole King	14	0,555	0,446	0,799	0,533	0,624	0,492
	Michael Jackson	38	0,513	0,391	0,823	0,504	0,653	0,434
	Queen	51	0,489	0,371	0,808	0,436	0,588	0,381
	Beatles	180	0,671	0,633	0,783	0,581	0,688	0,541
	Zweieck	18	0,552	0,439	0,795	0,53	0,711	0,44
	TOTAL	301	0,608	0,538	0,794	0,541	0,665	0,492
[78] (Wdh.)	TOTAL	301	0,599	0,657	0,583	0,643	0,636	0,676
(Fusion1)	TOTAL	301	0,596	0,642	0,596	0,634	0,528	0,834
(Fusion2)	TOTAL	301	0,621	0,624	0,667	0,652	0,667	0,659
Baseline	Carole King	14	0,391	0,246	1	0,287	1	0,168
	Michael Jackson	38	0,366	0,227	1	0,251	1	0,145
	Queen	51	0,378	0,238	1	0,292	1	0,172
	Beatles	180	0,582	0,425	1	0,34	1	0,208
	Zweieck	18	0,388	0,244	1	0,263	1	0,152
	TOTAL	301	0,5	0,349	1	0,313	1	0,189

Tabelle 3.2.: Ergebnisse der automatischen Strukturanalyse auf *Isophonics* unterteilt nach Interpreten mit Vergleichswerten.

überein. Von den 20 Songs mit den durchschnittlich höchsten Evaluationswerten waren 17 Beatles-Songs, wohingegen der Anteil an den 20 Songs mit den niedrigen Werten nur 2 beträgt. In [181] wird die Vermutung geäußert, dass dies ein Indiz für die Anpassung der Algorithmen an diesen weitverbreiteten und beliebten Datensatz ist. Die Vergleiche mit dem Mazurka-Datensatz lassen allerdings auch die Vermutung zu, dass die Annotationen für den Beatles-Datensatz eher auf Wiederholungen basieren, für deren Erkennung bessere Algorithmen existieren als beispielsweise für verschiedene Genres, die der Struktur des Stückes *Bohemian Rhapsody* von Queen zugrunde liegen.

Der ebenfalls in Abschnitt 2.7 vorgestellte, öffentlich zugängliche Teil des SALAMI-Datensatzes [183] besteht aus 779 Stücken, von denen 498 über Annotationen von zwei verschiedenen Personen verfügen. In Abbildung 3.15 sind die beiden Annotationen der Grobstruktur eines Stückes aus dem *SALAMI*-Datensatz zusammen mit dem von uns berechneten Resultat illustriert. Diese Abbildung verdeutlicht den großen Unterschied der beiden Referenzannotationen, wodurch die von uns berechnete, mit der zweiten Referenz gut übereinstimmende Struktur im Vergleich zur ersten Referenzannotation wenig Übereinstimmungen zeigt. Ein Vergleich der beiden Annotationen auf allen 498 doppelt annotierten Stücken hat ergeben, dass solche Unterschiede bei allen Genres vorkommen, siehe dazu Abschnitt A.1.

Daher haben wir für die in Tabelle 3.3 dargestellten Evaluationsergebnisse mehrere Annotationen als Referenz zugelassen. Hierzu wurde das Strukturergebnis jedes Stückes gegen bis zu vier Annotationen (sowohl Grob- als auch Feinstruktur von jeweils bis zu zwei Personen)

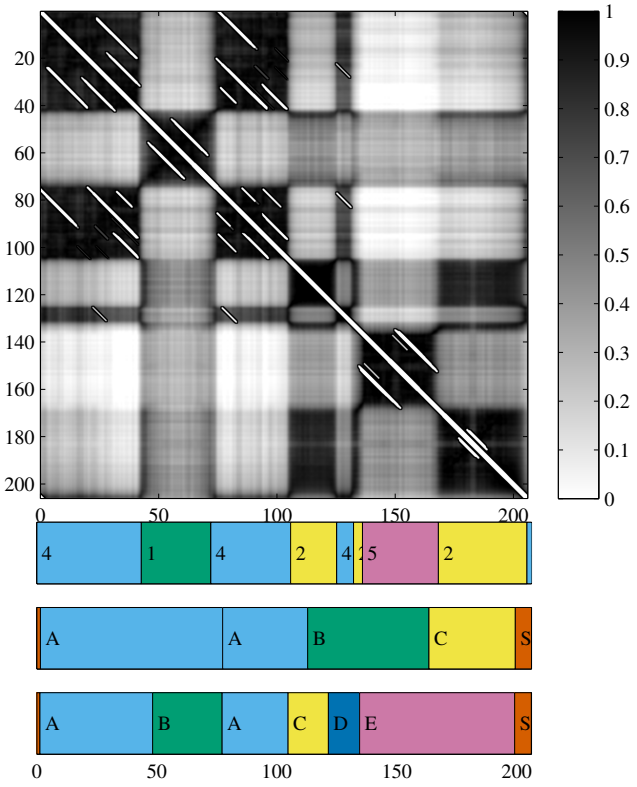


Abbildung 3.15.: Wiederholungsbasierte Analyse des Duetts »Là ci darem la mano« aus der Oper »Don Giovanni« von Wolfgang A. Mozart (KV 527). Die drei Strukturierungen zeigen die automatisch bestimmte (oben) sowie die beiden Referenzannotationen aus dem *SALAMI*-Datensatz.

ausgewertet und jeweils die Referenz mit dem höchsten paarweisen F-measure verwendet. Dies führt dazu, dass in der Evaluation im Schnitt höhere Werte erreicht werden als beim *Isophonics*-Datensatz, da beim letzteren nur mit jeweils einer Referenzannotation verglichen wird. Zu einer ausführlicheren Diskussion dieser Problematik siehe auch Abschnitt 2.7. Somit erreichen wir ein durchschnittliches F-measure von 0,66 für die Segmentbezeichnungen, wobei auf dem Genre »Jazz« mit 0,693 der höchste und auf »Classical« mit 0,604 der niedrigste Durchschnittswert erzielt wurde.

Einen weiteren Beitrag zu den im Vergleich zu *Isophonics* höheren Evaluationswerten leisten einige Audiodateien des *SALAMI*-Datensatzes, die nur aus gesprochenem Text bestehen und teilweise eine Länge von nur wenigen Sekunden aufweisen. Diese bestehen folglich auch nur aus einem Segment und werden daher sowohl vom *Baseline*-Verfahren als auch von unserem nach Wiederholungen in einer Chroma-basierten Merkmalsfolge suchendem Verfahren als ein Segment erkannt. Auch der hohe Precision-Wert des *Baseline*-Verfahrens für das Genre »Jazz« (für eine Liste der Subgenres siehe Abschnitt A.1) zeigt, dass die automatische Evaluation nicht mit einer empirischen Beurteilung der Segmentierungsgüte übereinstimmen muss.

Methode	Genre	#Songs	Segmentbezeichnungen			Segmentgrenzen (3 s)		
			F	***P***	***R***	***F***	***P***	***R***
Konv$_2$	Live_Music	257	0,669	0,628	0,792	0,363	0,681	0,29
	classical	112	0,604	0,628	0,681	0,47	0,526	0,496
	jazz	122	0,693	0,718	0,732	0,419	0,604	0,378
	popular	162	0,657	0,629	0,772	0,477	0,664	0,427
	unknown	15	0,652	0,607	0,756	0,529	0,768	0,445
	world	111	0,663	0,657	0,762	0,422	0,617	0,405
	TOTAL	779	0,66	0,646	0,757	0,422	0,636	0,381
Baseline	Live_Music	257	0,638	0,499	1	0,214	0,998	0,127
	classical	112	0,621	0,491	1	0,446	1	0,308
	jazz	122	0,756	0,643	1	0,315	0,996	0,193
	popular	162	0,616	0,477	0,998	0,34	1	0,231
	unknown	15	0,544	0,386	1	0,25	1	0,146
	world	111	0,652	0,522	1	0,345	0,991	0,229
	TOTAL	779	0,649	0,517	1	0,309	0,997	0,2

Tabelle 3.3.: Ergebnisse der automatischen Strukturanalyse auf dem *SALAMI*-Datensatz unterteilt nach Genres mit Vergleichswerten.

Vergleichen wir das von unserer Methode erzielte F-measure von 0,66 mit dem des Baseline-Verfahrens von 0,649, so lässt sich a priori keine wesentliche Verbesserung feststellen. Dies lässt die folgenden Schlüsse zu: Entweder ist das vorgestellte Verfahren zur automatischen Strukturierung unzureichend, oder die verwendeten Annotationen sind ungeeignet für eine wiederholungsbasierte Segmentierung, oder die Evaluationsmethode beschreibt die Qualität der Ergebnisse nicht adäquat. Die qualitative Auswertung zeigt jedoch, dass das Verfahren bei Vorliegen einer durch Wiederholungen gegliederten Struktur zu durchaus sinnvollen Ergebnissen führt. Leider liegen für die Referenzannotationen keine erläuternden Kommentare vor, aus denen ersichtlich ist, ob und in welchem Maße die manuellen Segmentierungen durch Wiederholungen beschrieben werden können. Folglich ist es möglich, dass zumindest ein Teil des hier betrachteten Phänomens auf abweichende Segmentierungskriterien zurückgeführt werden kann. Die konzeptionellen Schwächen des Evaluationsverfahrens wurden bereits in Abschnitt 2.7 formuliert. Zusammenfassend stellt dieser Vergleich der beiden Segmentierungsmethoden einen Indikator für die kombinierten Unzulänglichkeiten der automatischen Evaluation von automatisch generierten Strukturierungen dar und mag zu einer kritischen Diskussion über die Aussagekraft solcher Werte beitragen.

3.6. Zusammenfassung und Ausblick

In diesem Abschnitt haben wir eine neuartige Methode zur Konvertierung einer Selbstähnlichkeitsmatrix mit Pfadstrukturen in eine Selbstähnlichkeitsmatrix mit Blockstrukturen mittels

Eigenwertzerlegung vorgestellt. Als vorrangigen technischen Beitrag haben wir die charakteristischen Eigenschaften der Eigenvektoren spezieller Pfadstrukturen diskutiert. Weiterhin haben wir als eine Anwendung unserer Konvertierungsmethode gezeigt, wie ein Verfahren zur homogenitätsbasierten Strukturanalyse auf die konvertierte Pfadmatrix angewendet werden kann, um so eine wiederholungsbasierte Strukturanalyse zu ermöglichen. Experimente haben gezeigt, dass die mittels dieses Verfahrens erzielten Ergebnisse vergleichbar zu denen spezialisierter Systeme sind. Weiterhin bietet die Ähnlichkeit der hier vorgestellten Methode zum *spektralen Clustering* die Möglichkeit, hierarchische Strukturen direkt zu erfassen. Ein erster Ansatz hierzu wird in [109] präsentiert.

Wir hoffen, dass dieser Ansatz nicht nur konzeptionell interessant ist, sondern auch neue Wege zur gemeinsamen Analyse verschiedener Segmentierungsprinzipien bereits zu einem frühen Zeitpunkt im Verlauf einer Strukturanalyse erschließen kann. Insbesondere scheint es ein vielversprechender Ansatz zu sein, blockähnliche Selbstähnlichkeitsmatrizen zur Beschreibung homogener musikalischer Eigenschaften mit konvertierten pfadähnlichen Selbstähnlichkeitsmatrizen, die wiederholungsbasierte musikalische Eigenschaften abbilden, in einer gemeinsamen Verarbeitung zu kombinieren. Hierzu ist allerdings vor allem ein tiefgreifenderes Verständnis der musikalischen Eigenschaften einer manuell annotierten Struktur notwendig.

4. Fallstudie: Schuberts »Winterreise«

Bei der automatischen Musikstrukturanalyse handelt es sich um ein oftmals nur vage definiertes Problem, welches zusätzlich durch Fehlen einer objektiven Auswertungsmöglichkeit erschwert wird. Vielfach ist Vorwissen sowohl über die Eigenarten der betreffenden Musikstücke als auch der zum Vergleich herangezogenen Referenzannotationen für ein brauchbares Ergebnis erforderlich.

In diesem Kapitel beschreiben wir einen neuen Datensatz mit mehreren Annotationen, der zusätzlich über Motivationstexte zu jeder manuell erstellten Segmentierung verfügt. Weiterhin stellen wir zwei Modifikationen bestehender Merkmalsdarstellungen vor, die gemäß des obigen Vorwissens benötigt werden. Dies verwenden wir anschließend zur Diskussion eines exemplarischen Stückes, bei dem genau dieses Vorwissen zur Interpretation der Ergebnisse und zum Entwurf einer kombinierten Strukturanalyse verwendet wird. Hierbei wird deutlich, dass insbesondere noch lokale Kriterien zur Beschreibung der Relevanz verschiedener musikalischer Aspekte oder Prinzipien entwickelt werden müssen. Als einen ersten Schritt in diese Richtung entwickeln wir eine interaktive Benutzerschnittstelle zur synchronen Verbindung der graphischen Repräsentation technischer Merkmale mit der zugrundeliegenden Musik.

4.1. Einleitung

Der Liederzyklus *Winterreise* wurde 1827 von Franz Schubert (D 911, op. 89) komponiert und basiert auf einer Gedichtsammlung von Wilhelm Müller. Der Zyklus besteht aus 24 Einzelliedern für einen männlichen Sänger (üblicherweise ein Tenor oder Bariton) mit Klavierbegleitung, siehe auch [32]. Protagonist der Winterreise ist ein einsamer Wanderer, der sich nach seiner unerreichbaren Geliebten sehnt und dessen Verzweiflung sich im Verlauf der Komposition bis zur Selbstaufgabe im letzten Lied steigert. Die Lieder der Winterreise sind strukturell komplexer und vielfältiger als etwa Schuberts Vertonung der Gedichtsammlung »Die schöne Müllerin« desselben Dichters [209]. Es existieren motivische Verbindungen zwischen den einzelnen Liedern, die häufig Bewegungsmotive aufweisen [41].

Die formale Struktur von Schuberts Liedern umfasst die auf Wiederholungen basierenden Formen des Strophenliedes, die ternäre ABA-Form und die Barform AAB, allerdings auch die durchkomponierte Form, bei der für jede Strophe verschiedenes musikalisches Material verwendet wird. Einige Stücke liegen als variiertes Strophenlied vor, wobei der Typ der Variation sich jeweils unterscheidet. In Tabelle 4.1 geben wir in einer kurzen Übersicht die grobe musikalische Form jedes Liedes an sowie die durchschnittliche Aufnahmedauer gemittelt über

Nr.	Titel	Grobform	Tonart	øDauer
1	Gute Nacht	$A\,A\,A\,A'$	d-Moll	06:35
2	Die Wetterfahne	$A\,B\,A'\,A'$	a-Moll	02:06
3	Gefror'ne Tränen	$A\,B\,C\,C$	f-Moll	03:03
4	Erstarrung	$A\,B\,A'$	c-Moll	03:32
5	Der Lindenbaum	$A\,A'\,B\,A$	E-Dur	05:49
6	Wasserflut	$A\,B\,A\,B$	e-Moll	05:20
7	Auf dem Flusse	$A\,B\,C$	e-Moll	04:17
8	Rückblick	$A\,B\,A'$	e-Moll	02:41
9	Irrlicht	$A\,A\,B$	h-Moll	03:08
10	Rast	$A\,B\,A\,B$	c-Moll	03:53
11	Frühlingstraum	$\|\!: A\,B\,C :\!\|$	A-Dur	05:07
12	Einsamkeit	$A\,B\,C\,C$	h-Moll	03:16
13	Die Post	$A\,B\,A\,B$	Es-Dur	02:38
14	Der greise Kopf	$A\,B\,A'$	c-Moll	03:44
15	Die Krähe	$A\,B\,A'$	c-Moll	02:32
16	Letzte Hoffnung	$A\,B\,C\,D$	Es-Dur	02:41
17	Im Dorfe	$A\,B\,A\,C$	D-Dur	03:55
18	Der stürmische Morgen	$A\,B\,A'$	d-Moll	01:01
19	Täuschung	$A\,A\,B\,A$	A-Dur	01:42
20	Der Wegweiser	$(A\,B\,A\,C)$	g-Moll	05:02
21	Das Wirtshaus	$(A\,A\,B\,C)$	F-Dur	05:11
22	Mut	$A\,A\,B\,B$	g-Moll	01:41
23	Die Nebensonnen	$A\,A\,B\,A$	A-Dur	03:37
24	Der Leiermann	$A\,A\,B$	a-Moll	04:22

Tabelle 4.1.: Übersicht der 24 Einzelstücke des Liederzyklus' *Winterreise*.

9 verschiedene Aufnahmen jedes Liedes. Die Stücke mit eingeklammerten Formen können unserer Meinung nach mit verschiedenen Formen adäquat beschrieben werden, von denen hier nur eine angegeben ist. Eine detaillierte Beschreibung der musikalischen Struktur jedes Liedes ist in Abschnitt A.2 zu finden.

Für die automatische Strukturanalyse ist Schuberts *Winterreise* zweifellos eine Herausforderung. Bezüglich der Instrumentierung stellt sie eine Zwischenstufe zwischen dem *Mazurka*-Datensatz, welcher ausschließlich aus Klaviermusik besteht, und den komplexeren Datensätzen aus Kammer- und Orchestermusik sowie populärer Musik dar, bei denen die Instrumentierung häufig wechselt und in vielen Fällen nicht eindeutig annotiert werden kann. Ähnlich wie beim populärmusikalischen *Isophonics*-Datensatz sind die vorkommenden musikalischen Strukturen vielfältig und nicht mittels eines einzigen Segmentierungsprinzips nachvollziehbar, wohingegen die Strukturen des *Mazurka*-Datensatzes und teilweise des Beatles-Datensatzes sich allein durch wiederholungsbasierte Strukturierung schon verhältnismäßig gut rekonstruieren lassen. Obwohl der Winterreise-Datensatz nur aus 24 Einzelliedern besteht, kann durch Berücksichtigung mehrerer Aufnahmen dieses vielfach aufgenommenen Liederzyklus' unkompliziert ein größerer Testkorpus aufgebaut werden. Wir verwenden

9 Komplettaufnahmen sowie die 5 Einzelstücke aus dem *SMD*-Datensatz[1] wodurch sich eine Anzahl von insgesamt 221 Aufnahmen ergibt.

In diesem Kapitel stellen wir zunächst in Abschnitt 4.2 die von uns angefertigten Strukturannotationen für den *Winterreise*-Datensatz vor. Diese beruhen hauptsächlich auf dem Segmentierungsprinzip der Wiederholung, auf Homogenitätsbereichen bezüglich lokaler Tonarten sowie den zeitlichen Positionen der Liedstrophen innerhalb eines Stückes. Für die Ermittlung der Wiederholungsstruktur wurde bereits in Kapitel 3 ein Verfahren diskutiert. In Abschnitt 4.3 stellen wir ein Merkmal für die homogenitätsbasierte Segmentierung nach Tonarten vor und diskutieren eine hierarchische Darstellung harmonischer Informationen. In Abschnitt 4.4 entwickeln wir ein neuartiges Merkmal zur Unterscheidung zwischen Gesangs- und Klavierpassagen, um so die Positionen der Liedstrophen zu schätzen. Abschließend zeigen wir in Abschnitt 4.5 mittels einer detaillierten Analyse zweier ausgewählter Stücke aus der Winterreise, wie alle diese Komponenten zur Ermittlung der musikalischen Struktur notwendig sind. Damit illustrieren wir exemplarisch die Möglichkeiten und Grenzen der in dieser Arbeit vorgestellten Methodik zur automatischen Musikstrukturanalyse.

4.2. Manuelle Annotationen

Für die 24 Lieder der Winterreise haben wir[2] Annotationen der musikalischen Strukturen basierend auf [1] angefertigt. Bedingt durch die maschinelle Art der Datenverarbeitung verlangt jede automatisierte Segmentierung die Definition fester Segmentgrenzen sowie eine eindeutige Benennung der Segmente, vergleiche hierzu Abschnitt 2.5. Daher haben wir im Gegensatz zu den zahlreichen Analysen der Winterreise weder eine Interpretation der Musikstücke durchgeführt noch eine Formanalyse im musikwissenschaftlichen Sinne. Stattdessen stand das Erkennen musikalischer Strukturen für die Verwendung als erste Annäherung eines semantisch sinnvollen Analysemodells im Vordergrund. Unsere Strukturannotationen sind somit als reine Struktur- und nicht als Formanalyse zu verstehen.

Um einem größeren Kreis von Interessenten einen eigenen Einblick in diesen Datensatz zu ermöglichen, sind unsere Annotationen zusammen mit zwei freien Aufnahmen und einigen graphischen Darstellungen des musikalischen Materials unter `http://winterreise.sechsachtel.de` öffentlich zugänglich. Für den Datensatz stellen wir neben der Strukturanalyse zwei weitere Annotationen zur Verfügung, dies sind zum einen die zeitlichen Positionen der Gedichtstrophen innerhalb jedes Stücks der Winterreise und zum anderen eine Beschreibung der lokalen Tonarten, siehe auch Abbildung 4.1. Analog zum *Mazurka*-Datensatz (vgl. Abschnitt 2.7) wurden die Annotationen auf Basis des Notentextes angefertigt. Die Synchronisation mit den einzelnen Aufnahmen wurde ebenfalls durch Verwendung von MIDI-Dateien und dem in [47] vorgestellten Verfahren realisiert.

[1] *Saarland Music Data* [125], `http://www.mpi-inf.mpg.de/resources/SMD/`

[2] Diese Annotationen sind in enger Zusammenarbeit mit der Musikwissenschaftlerin Polina Gubaidullina entstanden.

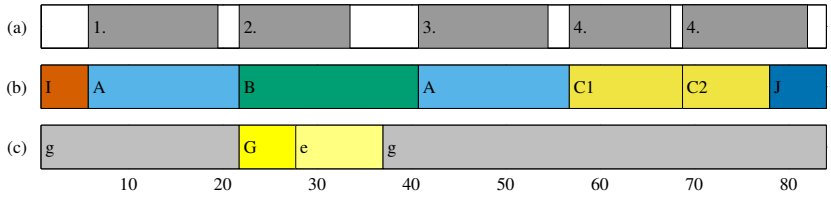


Abbildung 4.1.: Manuelle Annotationen für das 20. Stück der Winterreise, Zeitachse in Takten. **(a)** Liedstrophen, **(b)** Struktur, **(c)** lokale Tonarten.

Die *Gedichtstrophen* stellen eine einfache und objektive Segmentierung des Musikstückes dar, die durch Identifikation der Strophen des zugrundeliegenden Gedichtes im Musiksignal erfolgt. Durch die gelegentlich vorkommende Wiederholung einzelner Strophen bzw. in einem Falle auch des Weglassens einer Strophe wurden die Segmente mit der Nummer der entsprechenden Gedichtstrophe bezeichnet. In Abbildung 4.1a sind die Positionen dieser Strophen innerhalb des Musikstückes dargestellt, die vierte Strophe wird wiederholt.

Die *Strukturanalyse*, also die Segmentierung der Musikstücke nach rein musikalischen Gesichtspunkten ohne Berücksichtigung der Strophen des zugrundeliegenden Gedichtes, ist weitaus komplexer als die Annotierung der Gedichtstrophen. Als Kriterien für diese Segmentierungen haben wir uns sowohl exakte als auch variierte Wiederholungen angesehen, gefolgt von Tonartwechseln und – in seltenen Fällen – auch textuellen Änderungen. Im allgemeinen existiert weder eine Hierarchie dieser Kriterien noch eine objektive oder gar »richtige« Segmentierung für jedes Lied. Wir haben versucht, musikalisch sinnvolle Segmente zu identifizieren, allerdings mussten wir in einigen Fällen mehr als eine Segmentierung für dasselbe Musikstück angeben. Daher sind auch die Segmentgrenzen nicht eindeutig, da in manchen Fällen Segmente zusammengefasst werden können. Das in Abbildung 4.1b illustrierte Beispiel der Strukturannotation für das 20. Stück stellt beispielsweise nur eine von drei nahezu gleichwertigen Möglichkeiten dar. Jede unserer Segmentierungen wird in Abschnitt A.2 durch einen kurzen Begleittext begründet und erläutert. In Abbildung 4.2 wird eine graphische Übersicht der Strukturannotationen aller Stücke gegeben.

Die Bestimmung der *Tonarten* der Winterreise stellt wegen der fortgeschrittenen Harmonik und den teilweise häufigen Modulationen auch für menschliche Experten keine triviale Aufgabenstellung dar. Wir haben uns daher entschieden, nur die Bereiche zu annotieren, für die wir eine eindeutig vorherrschende Tonart identifizieren konnten. Daher weisen die Tonartenannotationen in einzelnen Stücken größere unannotierte Lücken auf, siehe beispielsweise das 2. Stück in Abschnitt A.2. Eine automatisierte Tonartenanalyse wird weiterhin durch häufig von Interpreten vorgenommene Transponierungen erschwert.

Bei der Benennung der Segmente haben wir uns an die Konvention gehalten, verschiedene Segmente mit lateinischen Großbuchstaben beginnend bei A, B usw. zu bezeichnen. Dies mag für einfach strukturierte Stücke ausreichend sein, aber im Allgemeinen – und speziell im Fall

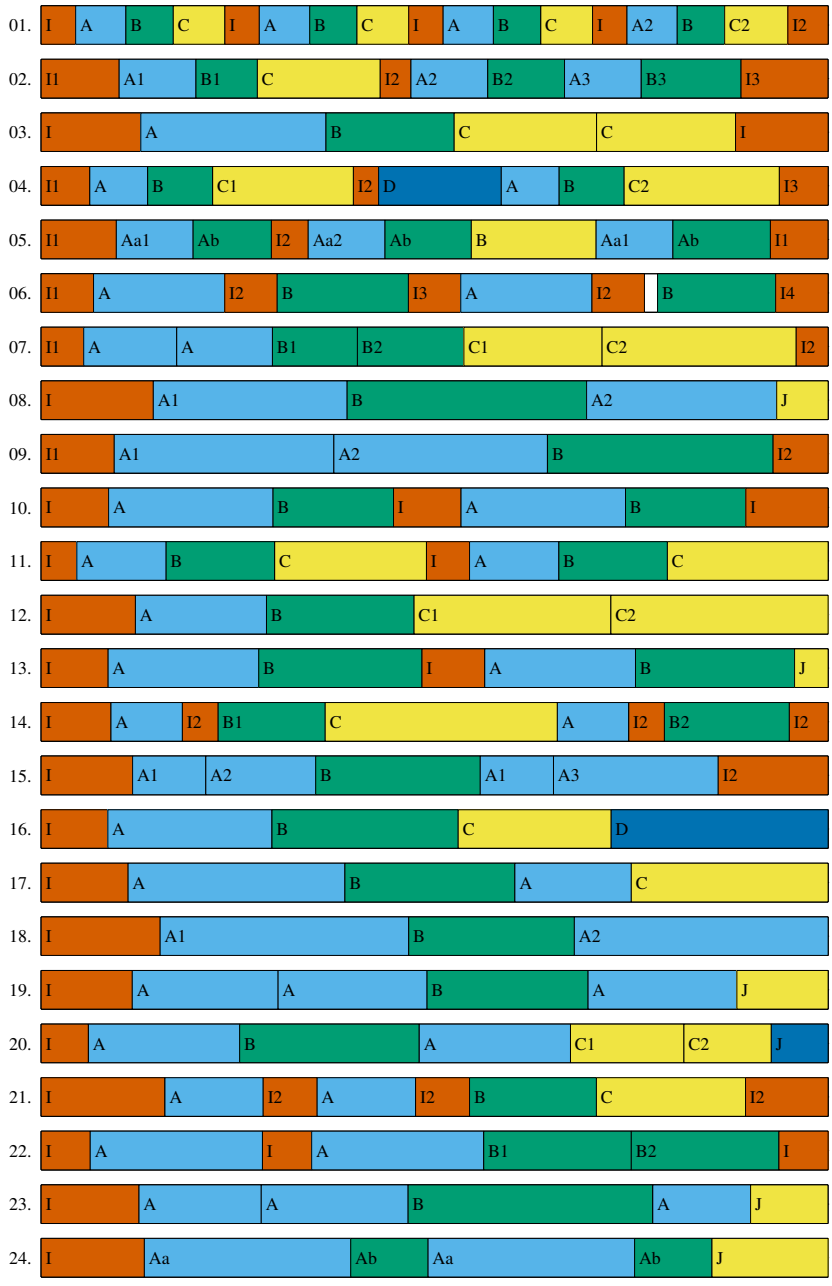


Abbildung 4.2.: Übersicht der *Winterreise*-Strukturannotationen.

der *Winterreise* – müssen wir zwischen verschiedenen Graden der Ähnlichkeit unterscheiden, siehe hierzu die detaillierten Beschreibungen in Abschnitt 2.7.

- A, B, C: Zwei Segmente werden durch verschiedene Buchstaben beschrieben, wenn sie verschiedenes musikalisches Material enthalten.
- $A1, A2$: Bei Vorliegen einer variierten Wiederholung bzw. der motivischen Verwandtschaft zweier Segmente verwenden wir denselben Großbuchstaben und zeigen mit einer zusätzlichen Zahl die Variation an. So bezeichnen beispielsweise $A2$ und $A3$ die erste bzw. zweite Variation des Segments $A1$. Dies ist nicht mit den in Abschnitt 2.1 beschriebenen Indizes für (gleichartige) Wiederholungen A_1, A_2 zu verwechseln!
- $Aa, Ab, Aa1, Aa2$: In manchen Fällen werden nur Teile eines Segmentes im Laufe des Musikstückes wiederholt. In diesen Fällen unterteilen wir das Segment, um diese wiederholten Passagen exakt zu kennzeichnen. Die Kleinbuchstaben beschreiben dabei die Position einer solchen Passage innerhalb des Segments. Falls notwendig, können durch Nummern auch Variationen dieser Passagen gekennzeichnet werden.

Für eine automatische Analyse empfehlen wir, nicht stur die Bezeichnungen paarweise miteinander zu vergleichen, sondern je nach Aufgabenstellung die Bezeichnungen zu vereinfachen. Bei einer wiederholungsbasierten Segmentierung auf einer feinen Granularitätsstufe würde beispielsweise zwischen Aa und Ab unterschieden werden, auf einer gröberen Skala würden die beiden Segmente Aa und Ab hingegen zu einem Segment A zusammenfallen. Werden nur exakte bzw. sehr wenig variierte Wiederholungen gesucht, so sind $A1$ und $A2$ als verschiedene Segmentklassen anzusehen, erlaubt man auch gröbere Variationen bzw. ist man an einer vereinfachten Struktur interessiert, so müssten diese Bezeichnungen derselben Segmentklasse zugeordnet werden.

Bei unseren Untersuchungen der Winterreise haben wir neben den beiden auf der Webseite veröffentlichten Aufnahmen von *Hüsch* (1933) und *Scarlata* (2006) noch sieben weitere Interpretationen verwendet, siehe Tabelle 4.2 für eine Liste der von uns verwendeten Aufnahmen.

4.3. Lokale Tonarten und harmonische Hierarchie

Die Bestimmung der in einem Stück vorkommenden Akkorde (engl. *chords*) sowie Tonarten (*keys*) ist eine zentrale Problemstellung im *Music Information Retrieval* [38, 70, 107, 141, 167, 177]. Insbesondere die auf einem gröberen zeitlichen Raster wirkenden *lokalen Tonarten* sind als homogenes Merkmal auch ein wichtiges Indiz für die Strukturanalyse [139, 150]. Die Popmusik etwa weist die *Rückung* als ein gängiges Stilmittel auf, bei der das Stück die lokale Tonart abrupt verlässt und ohne vorherige Modulation einen Halbton höher fortgesetzt wird. In der klassischen Musik tritt in der Sonate das zweite Thema zuerst in der Dominanttonart auf und wird erst in der Reprise gegen Ende des Stücks in der Tonika wiederholt. Bei der in dieser Fallstudie diskutierten Winterreise spielt die Harmonik in vielen Stücken eine wichtige

Sänger	Pianist	Jahr	Anmerkungen
Gerhard Hüsch	Hanns-Udo Müller	1933	Schallplattenaufnahme, mittlerweile *public domain*, European Archive
Randall Scarlata	Jeremy Denk	2006	Live-Aufnahme des Isabella Stewart Gardner Museums (Boston), *Creative Commons*-Lizenz
Thomas Allen	Roger Vignoles	1998	CD, ASIN: B004IESDBW, Virgin Classics, EMI
Thomas Oliemans	Bert van den Brink	2006	CD, ASIN: B000HLDD3S, Fineline
Thomas Quasthoff	Charles Spencer	1998	CD, ASIN: B00000DFKL, RCA Red Seal (Sony)
Roman Trekel	Ulrich Eisenlohr	1999	CD, ASIN: B000031WH6, Naxos
Dietrich Fischer-Dieskau	Daniel Barenboim	1980	CD, ASIN: B0000012ZU, Deutsche Grammophon, 1997
Dietrich Fischer-Dieskau	Jörg Demus	1966	CD, ASIN: B000001GQE, Deutsche Grammophon, 1995
Dietrich Fischer-Dieskau	Gerald Moore	1955	CD, ASIN: B00006BCDM, EMI Classics, 2002

Tabelle 4.2.: Verwendete Aufnahmen von Schuberts Liederzyklus *Winterreise*. Die beiden zuerst aufgeführten Einspielungen sind auch unter `http://winterreise.sechsachtel.de` verfügbar.

Rolle für die Annotierung der musikalischen Struktur, siehe hierzu die Beschreibungen unserer Annotationen in Abschnitt A.2.

Zur Bestimmung der lokal vorherrschenden Tonarten wurde von Krumhansl [90] die Verwendung von Schablonen (*templates*) vorgeschlagen, die empirisch ermittelte Werte zur Beschreibung der Zugehörigkeit jeder der 12 Chroma-Komponenten zu einer Tonart beinhalten. Obwohl aus musikalischer Sicht eine Tonart- und Akkordanalyse immer eine zu berücksichtigende zeitliche Komponente beinhaltet (vgl. [165]), ist dieses Modell ein wesentlicher Grundbaustein der automatischen Tonartenanalyse geworden. Eine weitere Motivation für diesen einfachen Ansatz sowie darauf aufbauende Modelle kann aus [21, 22] abgeleitet werden, da hier experimentell gezeigt wurde, dass die Qualität von Akkorderkennungsprogrammen so stark von den eingesetzten Glättungsfiltern abhängt, dass die Auswirkungen eines etwaigen methodischen Vorteils komplexerer Ansätze im Allgemeinen durch geschickte Wahl der Filterparameter deutlich übertroffen wird.

Solche komplexeren Methoden nutzen beispielsweise geometrische Darstellungen zur Erkennung von harmonischen Wechseln [70] oder auch Hidden Markov Models [96, 177] zur Modellierung der zeitlichen Progression von Akkordfolgen. Eine weitere Variante stellt die Einbeziehung musikalischen Vorwissens zur Konstruktion probabilistischer Modelle dar [149]. Stehen mehrere Aufnahmen desselben Stückes zur Verfügung, bieten sich auch versionsübergreifende Verfahren zur Ermittlung harmonisch eindeutiger Passagen an, wodurch die Robustheit der schablonenbasierten Ansätze gesteigert werden kann [86, 87].

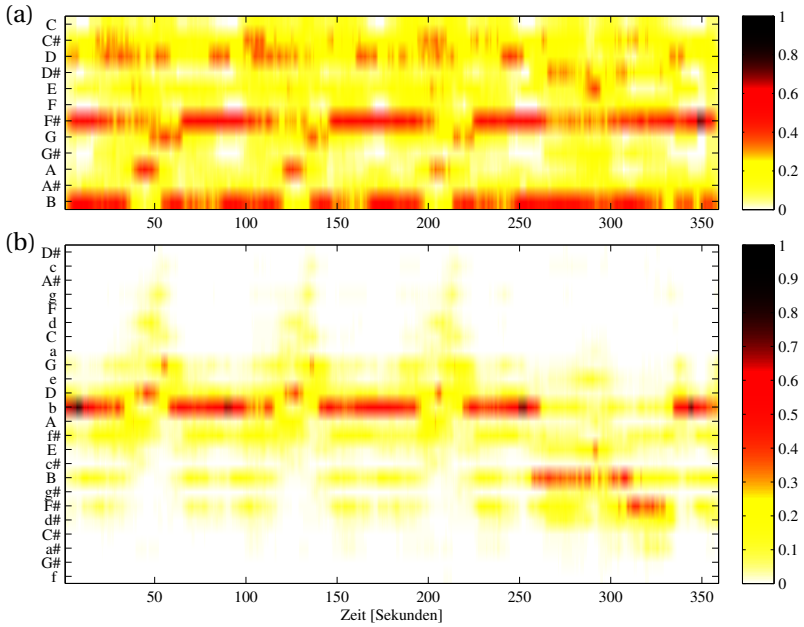


Abbildung 4.3.: Harmoniebasierte Merkmalsdarstellungen des ersten Stücks der Winterreise in der Aufnahme von Quasthoff bei einer Auflösung von 12 Sekunden. **(a)** Chroma-Merkmale, **(b)** daraus abgeleitete Key-Merkmale.

4.3.1. Schablonen und Tonartenmerkmale

Wir verwenden im Folgenden zur Beschreibung der 24 enharmonischen Tonarten der modernen westlichen Musik (Dur und Moll-Schema) die Menge $[0,1]^{24}$ zur Beschreibung aller möglichen Merkmale im Sinne von Abschnitt 2.3. Hierbei dienen die ersten 12 Komponenten zur Beschreibung der Dur-Tonarten und die verbleibenden 12 Komponenten zur Beschreibung der Moll-Tonarten. Jedem Zeitpunkt des Musikstückes wird nun eine Wahrscheinlichkeitsverteilung bezüglich dieser 24 Tonarten zugeordnet, analog zu den Chroma-Merkmalen nennen wir diese Vektoren auch Tonarten- oder Key-Merkmale. Die lokal vorherrschende Tonart kann für jeden Zeitpunkt mittels eines Maximum-Likelihood-Verfahrens, d. h. durch Auffinden der Position der Komponente mit dem höchsten Wert, bestimmt werden, siehe auch Abbildung 4.3 für eine gemeinsame Darstellung von Chroma- und Key-Merkmalen.

Diese Key-Merkmale berechnen wir aus den vorliegenden Chroma-Merkmalen mittels eines üblichen Schablonenansatzes, welche die in einem Chroma-Vektor vorliegende Verteilung der Signalenergie auf die einzelnen Chromabänder in die oben genannte Wahrscheinlichkeitsverteilung bezüglich der 24 Tonarten umwandelt. Die Schablone selbst besteht aus zwei prototypischen Chroma-Vektoren, welche die Verteilung der Energie der einzelnen Halbtöne in den Tonarten C-Dur und c-Moll angeben. Durch zyklische Verschiebung lassen sich die Prototypen für die anderen Tonarten erreichen. Diese Prototypen werden anschließend zu einer

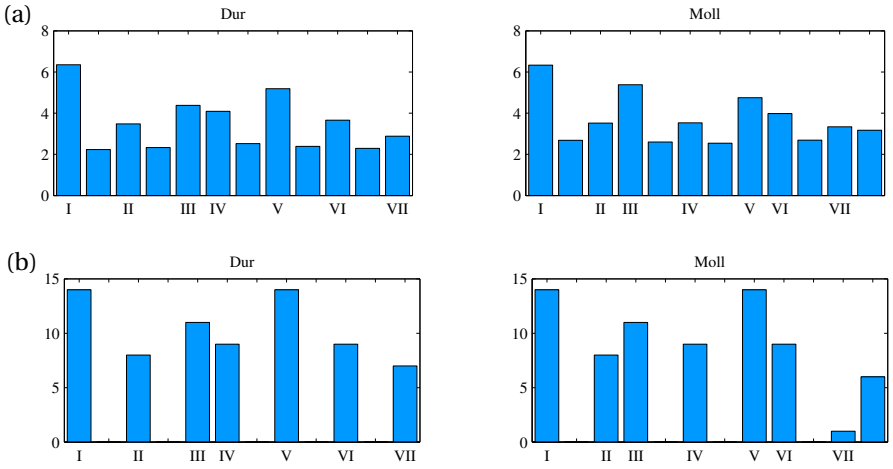


Abbildung 4.4.: Die beiden zur Ermittlung von Tonarten verwendeten Schablonen. Diese ordnen jedem Ton einer chromatischen Tonleiter abhängig vom angenommenen Grundton einen Gewichtungsfaktor zu. Die Stufen der diatonischen Tonleitern sind mit römischen Ziffern gekennzeichnet. **(a)** Empirisch ermittelte Werte von Krumhansl [90], **(b)** funktionsharmonisch motivierte Werte.

24×12-Matrix zusammengeführt, die von links an jeden Chroma-Vektor multipliziert wird. Das Ergebnis wird anschließend ℓ^1-normalisiert[3], um eine Wahrscheinlichkeitsverteilung zu erreichen. Zur Erhöhung der Deskriptivität insbesondere der Visualisierungen wenden wir vor dem Normalisierungsschritt eine Exponentialfunktion an, um die häufig bei der Berechnung von Chroma-Merkmalen eingesetzte Logarithmierung des Spektrogramms (vergleiche etwa [155]) bzw. bei Verwendung eines Filterbank-Ansatzes die logarithmische Quantisierung der Chroma-Bänder [116] aufzuheben. Diese logarithmische Skalierung der Einträge $x \in [0,\infty)$ der Chroma-Vektoren kann näherungsweise durch eine Abbildung der Form $x \mapsto \log(cx+1)$ mit einer Konstanten $c \geq 1$ ausgedrückt werden und dient zur Anpassung an das menschliche Hörempfinden, welches Lautstärke auf einer logarithmischen Skala (wie beispielsweise der bekannten *Dezibel*-Skala) wahrnimmt. Bei der Berechnung der Key-Merkmale wird dies durch Anwendung der Umkehrfunktion $x \mapsto \frac{1}{c} \cdot (\exp(x) - 1)$ aufgehoben. Dieser Schritt hat keinen Einfluss auf die Bestimmung der Tonart, da wir hierzu nur die Position des Maximums verwenden.

Neben den bereits vorgestellten Werten von Krumhansl [90] wurden auch von Temperley [191] leicht modifizierte und an die diatonische Skala angepasste Werte vorgestellt. Eine weitere Modifikation nutzte Gómez [56] durch Verwendung einer feineren Unterteilung des Chromagramms zur Verringerung von Störungen durch unsaubere Intonation. Für eine Übersicht siehe etwa [23, 156]. In [198, 199] wird eine musikwissenschaftliche Herangehensweise verfolgt, bei der die Schablonenwerte nicht empirisch, sondern durch musikalische Skalen wie

[3] Die ℓ^1-normalisierte Version v' eines Vektors $v \geq 0$ erhält man durch Teilung der einzelnen Komponenten durch die Summe aller Komponenten, $v' := v / \sum_i v_i$.

etwa der Dur- und Moll-Tonleiter oder der Pentatonik bestimmt werden. Wir verfolgen einen ähnlichen Ansatz und verwenden für unsere Experimente auf grundlegender Funktionsharmonik beruhende Schablonenwerte. Hierzu ordnen wir jedem Hauptfunktionsakkord (Tonika, Subdominante und Dominante) und jedem Nebenfunktionsakkord (die entsprechenden Parallelen) einen im Allgemeinen monoton fallenden empirischen Gewichtungsfaktor $w = (w_\mathrm{T}, w_\mathrm{D}, w_\mathrm{S}, w_\mathrm{Tp}, w_\mathrm{Sp}, w_\mathrm{Dp})$ zu. Zur Bestimmung der beiden Chroma-Prototypen der w-Schablone wird nun für jede Chromakomponente ermittelt, in welchen Akkorden diese vorkommt und die entsprechenden Gewichte aufsummiert. Beispielsweise kommt in der Tonart C-Dur der Ton C im Tonikaakkord als Grundton, im Subdominantakkord F-Dur als Quintton und in der Tonikaparallelen a-Moll als Terzton vor. Somit erhält der C-Dur-Schabloneneintrag für das Chromaband C den Wert $w_\mathrm{T} + w_\mathrm{S} + w_\mathrm{Tp}$. Bei den Molltonarten werden zur Abdeckung von sowohl reinem als auch harmonischem Moll die Gewichte der Dominate dem Durakkord zugeordnet, wodurch sich für die große Septime der Wert w_D und für die kleine Septime der Wert w_Dp ergibt.

Durch diese Modellierung werden nur den Intervallen der diatonischen Tonleitern positive Gewichtungsfaktoren zugeordnet. In unseren Experimenten hat sich herausgestellt, dass die mittels des Gewichtsparameters $w := (7,6,4,3,2,1)$ berechneten Schablonenwerte zu aussagekräftigen Tonartenmerkmalen führen, wobei das Verfahren tolerant gegenüber kleinen Änderungen dieser Parameter ist. In Abbildung 4.4 werden diese Schablonenwerte sowie die empirisch ermittelten Werte von Krumhansl visualisiert, indem der Gewichtungsfaktor für jeden chromatischen Ton (ausgehend vom angenommenen Grundton) angegeben wird. Die in der diatonischen Tonleiter der Dur- bzw. Molltonart enthaltenen, sogenannten *leitereigenen* Töne sind durch die Angabe ihrer Stufe in römischen Ziffern gekennzeichnet. Im Falle der Schablone für die Tonarten C-Dur und c-Moll werden die Gewichte also der Reihe nach auf die Halbtöne $\mathrm{C}, \mathrm{C}^\sharp, \mathrm{D}, \ldots, \mathrm{H}$ angewendet, bei E-Dur und e-Moll entsprechend auf $\mathrm{E}, \mathrm{F}, \mathrm{F}^\sharp, \ldots, \mathrm{D}^\sharp$. Mittels des trivialen Gewichtsparameters $w := (1,0,0,0,0,0)$ lässt sich dieses Verfahren auch unmittelbar zur Akkordbestimmung von reinen Dur- und Mollakkorden verwenden.

Bei unseren Schablonenwerten führen wir eine zusätzliche, heuristisch motivierte Anpassung ein: In vielen Musikstilen wird beim sogenannten authentischen Schluss, also der Akkordfolge Dominante-Tonika, zusätzlich die Leittonwirkung[4] der reinen Quarte (IV) in die große Terz (III) verwendet, indem der Dominantakkord (in C-Dur und c-Moll ist dies der G-Dur-Akkord) zum sogenannten Dominantseptakkord erweitert wird, der zusätzlich die kleine Septime über dem Dominantton enthält, was bezüglich des Grundtons der reinen Quarte entspricht (in C-Dur und c-Moll ist dies der Ton F). Zur Modellierung dieses Septakkordes erhöhen wir das Gewicht der Quarte zusätzlich um das halbe Gewicht des Dominantakkordes, wodurch wir für IV den Wert $w_\mathrm{S} + w_\mathrm{Sp} + w_\mathrm{Dp}/2 = 4 + 2 + 3 = 9$ erhalten. Unsere finalen Werte betragen somit $(14,0,8,0,11,9,0,14,0,9,0,7)$ für die Dur-Tonarten sowie $(14,0,8,11,0,9,0,14,9,0,1,6)$ für die Moll-Tonarten, siehe auch Abbildung 4.4b.

[4] Als *Leitton* wird ein Ton bezeichnet, der »durch seine melodische oder harmonische Bedeutung zur Auflösung in einen anderen Ton strebt. Leit- und Zielton sind stets einen Halbtonschritt voneinander entfernt.« [43].

Vergleich der beiden Schablonen

In Abbildung 4.5 sind vier Selbstähnlichkeitsmatrizen des ersten Stücks der Winterreise in der Aufnahme von Quasthoff illustriert, die mittels harmonischer Merkmale berechnet wurden. Hierzu verwenden wir CENS-Merkmale mit einer Glättung von 6 s bei einer Merkmalsauflösung von 4 Hz. In Abbildungsteil (a) wird der Transpositionsindex einer transpositionsinvariante Selbstähnlichkeitsmatrix wie bereits bei Abbildung 2.6 farblich gekennzeichnet. Die in Abbildung 2.7 vorgestellte funktionsharmonisch motivierte Farbgebung findet sich sowohl in der Annotation als auch bei der Selbstähnlichkeitsmatrix wieder. Bei dieser Herangehensweise werden keine expliziten Schablonen verwendet, sondern die Abschätzung der harmonischen Funktionen ergeben sich aus der Selbstähnlichkeit der zyklisch verschobenen Chroma-Merkmale [117], siehe hierzu die ausführliche Beschreibung in Abschnitt 2.4. Somit erfolgt hier keine Bestimmung der Tonarten, wodurch auch eine Unterscheidung nach Tongeschlecht nicht möglich ist. Beim Vergleich zweier Passagen in den gleichnamigen Tonarten d-Moll und D-Dur erhalten wir je nach vorliegenden Tönen entweder Transpositionsindex 0 (beispielsweise beim Vergleich des Dur-Dominantakkords, die in beiden Tonarten A-Dur-Akkord ist) oder Transpositionsindex 3, da die D-Dur-Skala dieselben leitereigenen Tönen wie die (reine) h-Moll-Skala aufweist. Die zweite Matrix (Teil b) ist ein typisches Beispiel einer Chroma-basierten Selbstähnlichkeitsmatrix. Die Passagen in F- und B$^\flat$-Dur werden markant von den Abschnitten in d-Moll und D-Dur unterschieden, alle weiteren Unterscheidungen sind nur sehr schwach ausgeprägt. So hebt sich weder die letzte Strophe in D-Dur gut erkennbar von den vorherigen Strophen in der gleichnamigen Molltonart ab, noch werden die (Sub-)dominant-Beziehungen zwischen F- und B$^\flat$-Dur sowie zwischen D- und G-Dur deutlich.

In den unteren beiden Teilbildern von Abbildung 4.5 werden Selbstähnlichkeitsmatrizen gezeigt, die aus Key-Merkmalen berechnet wurden. Die Merkmale für die Matrix in (c) wurde mittels der von Krumhansl angegebenen empirischen Werte berechnet, die in (d) mittels der oben vorgestellten funktionsharmonisch motivierten Gewichte. Beide Matrizen stellen die verschiedenen Tonarten gut dar, wobei die Darstellung in (d) etwas kontrastreicher ist und insbesondere der Unterschied zwischen den Abschnitten in d-Moll und D-Dur klarer erkannt wird. Durch die relativ kurze Länge der Glättung von nur 6 Sekunden weisen die harmonisch weitgehend homogenen Blöcke in d-Moll einige Unterstrukturen auf, die sich allerdings insbesondere bei der Matrix (d) im Vergleich zu den »sinnvollen« Strukturen nur wenig voneinander unterscheiden.

Bei diesem Beispiel ist das zur Matrix (d) korrespondierende Verfahren unter den hier vorgestellten am besten geeignet, um ein gegebenes Musikstück nach lokalen Tonarten zu segmentieren. In einem zweiten Schritt führen wir zur Überprüfung der Allgemeingültigkeit dieser Beobachtung anhand der neun vollständigen Aufnahmen der Winterreise eine Tonartanalyse mit beiden Schablonen durch. Hierzu berechnen wir Chroma-Merkmale mit einer Fensterlänge von 22,5 s und führen anschließend eine punktweise Tonartschätzung mittels der oben beschriebenen punktweisen Anwendung der Tonartenschablonen auf die Chroma-Vektoren mit darauffolgender Ermittlung der Maximalposition durch.

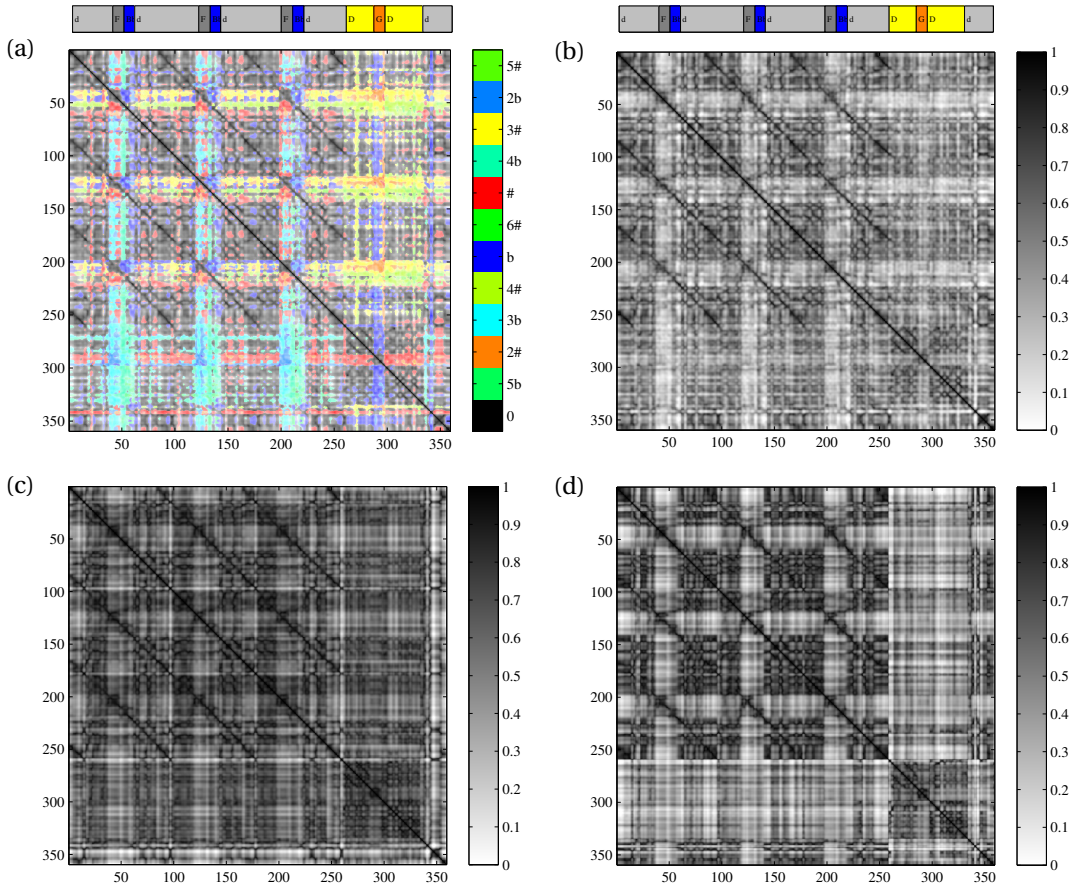


Abbildung 4.5.: Harmoniebasierte Selbstähnlichkeitsmatrizen (1. Stück, Quasthoff) bei einer Glättung von 6 Sekunden und einer Merkmalsauflösung von 4 Hz. **(a)** Transpositionsinvariante SSM mit Chroma-Merkmalen. Die Färbung gibt den Transpositionsindex an. **(b)** SSM mit Chroma-Merkmalen, **(c)** SSM mit mittels Krumhansl-Schablone berechneten Key-Merkmalen, **(d)** SSM mit mittels der vorgestellten Schablone berechneten Key-Merkmalen.

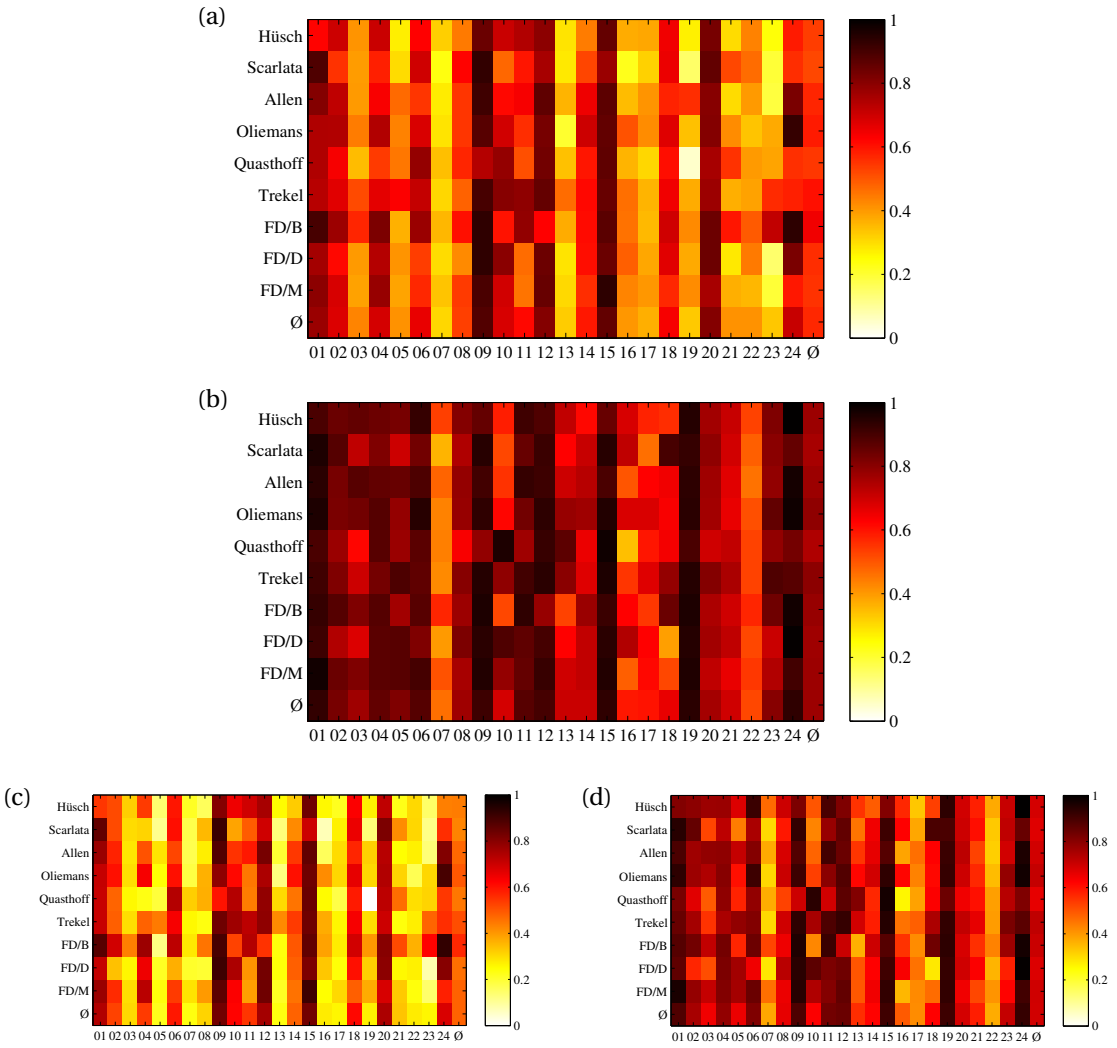


Abbildung 4.6.: Detaillierte Übersicht der Ergebnisse bei der automatischen Tonartschätzung auf 9 Komplettaufnahmen der *Winterreise*. **(a)** MIREX-Score bei empirisch ermittelten Werten (Krumhansl), **(b)** MIREX-Score bei funktionsharmonisch motivierten Werten; **(c)** Genauigkeit für (a), **(d)** Genauigkeit für (b).

Bei der automatischen Auswertung der Experimente ordnen wir jedem Zeitpunkt des jeweiligen Stückes bei einer Auflösung von 4 Hz eine Tonart zu und vergleichen diese mit der entsprechenden Position in der Referenzannotation. Die Bereiche, für die mangels Eindeutigkeit keine Tonartenannotationen existieren, werden bei dieser Auswertung nicht berücksichtigt. Die Information über Transponierungen durch die Interpreten liegen durch die Synchronisation vor und werden hier als bekannt vorausgesetzt. Zur Evaluation verwenden wir zwei verschiedene Maße:

- *Genauigkeit* (engl. *accuracy*). Jeder übereinstimmende Zeitpunkt wird mit 1 gewertet, jeder abweichende mit 0. Die Genauigkeit stellt den Durchschnitt dieser Wertungen dar.
- *MIREX-Score*. Wie oben, allerdings mit den im *MIREX-2005 Key Estimation Contest* verwendeten Punktzahlen: 1 für übereinstimmende Tonarten, 0,5 für Abweichungen von einer reinen Quinte, 0,3 für die parallele Dur-/Molltonart und 0,2 für die gleichnamige Dur-/Molltonart, vgl. [156].

Bei Verwendung der empirischen Schablonenwerte von Krumhansl [90] erhalten wir einen durchschnittlichen MIREX-Score von 0,57 und eine Genauigkeit von 0,481. Bei Verwendung der funktionsharmonisch motivierten Schablonenwerte wird ein MIREX-Score von 0,772 und eine Genauigkeit von 0,693 erreicht. Die in Abbildung 4.6 dargestellten Einzelergebnisse für alle Aufnahmen zeigen die teilweise erheblichen Unterschiede bei den einzelnen Stücken. Auffällig sind die niedrigen Werte in Abbildungsteil (a) bei den Stücken, die zu großen Teilen in Dur stehen; diese Passagen werden bei Verwendung der empirischen Schablonenwerte oftmals als Moll geschätzt. Die Tonarten des 7. Stücks der Winterreise sind sehr feingranular annotiert, was bei beiden diskutierten Verfahren zu Schwierigkeiten führt.

Durch dieses Experiment konnte die in Abbildung 4.5 beobachtete höhere Deskriptivität der funktionsharmonisch motivierten Tonarten-Merkmale bestätigt werden. Insbesondere auf dem Winterreise-Datensatz sind diese Merkmale folglich besser zur Bestimmung lokaler Tonarten geeignet. Daher werden wir im Folgenden für die Berechnung von Key-Merkmalen ausschließlich diese Schablone verwenden.

4.3.2. Hierarchische Darstellung

Die Bestimmung lokaler Tonarten und die automatische Akkorderkennung sind eng miteinander verbundene Problemstellungen, deren Übergänge fließend sind. In [166, 167] wird eine graphische Repräsentation der harmonischen Struktur auf sämtlichen Hierarchieebenen eingeführt und diskutiert, siehe Abbildung 4.7 für eine ähnliche Darstellung.

Bei diesen sogenannten *Scape-Plots* wird für jede mögliche Anzahl simultan betrachteter Chroma-Vektoren eine Akkord- bzw. Tonartenanalyse durchgeführt, was durch die sukzessive Verwendung aller möglichen Fensterlängen der Glättungsfilter realisiert wird. In der graphischen Darstellung werden die so ermittelten Tonarten durch verschiedene Farben illustriert, und die Intensität durch die sogenannte *clarity*, wofür in [166] die Differenz zwischen dem höchsten und zweithöchsten Tonartwert verwendet wird. Die Verwendung von kurzen Glät-

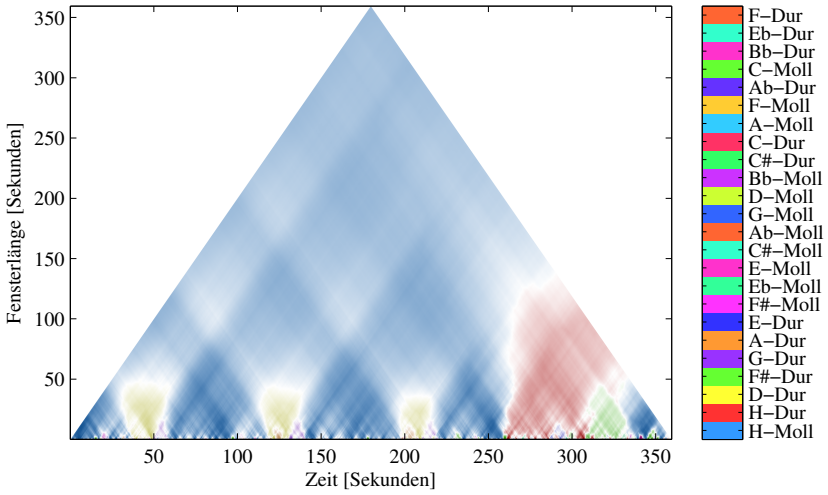


Abbildung 4.7.: Scape-Plot-Darstellung der harmonischen Struktur des ersten Stücks der Winterreise in der Aufnahme von Quasthoff, nach [166]. Die Farben stehen für die wahrscheinlichste Tonart, die Intensität illustriert den Abstand zur zweitwahrscheinlichsten Tonart. Die globale Tonart dieser Aufnahme ist h-Moll.

tungsfenstern entspricht dabei eher einer Akkordanalyse, die mittleren bis etwa 30 Sekunden beschreiben den Bereich lokaler tonaler Zentren, der anschließend fließend in den Bereich der (lokalen) Tonarten übergeht.

In [167] werden Scape-Plot-Darstellungen mehrerer Stücke miteinander verglichen, indem für gleiche Tonarten stets dieselbe Farbe genommen wird. Dies bietet sich aus zwei Gründen für den Winterreise-Datensatz nicht an: Zum einen werden die einzelnen Lieder von den verschiedenen Interpreten oftmals transponiert, wodurch die erkannten Tonarten pro Aufnahme voneinander abweichen. Dies ließe sich zwar durch die Verwendung zusätzlicher Informationen über die vorliegende Transposition ausgleichen, verliert dabei jedoch die universelle Anwendbarkeit auf jedes Audiomaterial unabhängig von der vorherrschenden Transponierung. Zum anderen umfassen die in der Winterreise vorkommenden Tonarten einen Großteil des Quintenzirkels, sodass die Unterscheidbarkeit der Farben bei einzelnen Tonarten nicht immer gegeben ist. Weiterhin verwenden wir bei unserer Darstellung die Farbattribute Helligkeit und Sättigung bereits zur Illustration der *clarity*, wodurch nur noch der Farbton zur Darstellung der Tonart verwendet werden kann. Um eine hinreichend gute Unterscheidbarkeit der einzelnen Tonarten zu gewährleisten, ist die Menge der nutzbaren Farbtöne jedoch stark eingeschränkt. Weiterhin erlaubt die Zuordnung einer beliebigen Farbe zu einer Tonart wie in Abbildung 4.7 zwar eine gute Unterscheidung der verschiedenen tonalen Bereiche, ordnet diesen allerdings keine harmonische Semantik zu.

Für die visuelle Erfassung harmonischer Kontextinformationen ist die in Abbildung 2.7 vorgestellte funktionsharmonisch motivierte Farbskala besser geeignet, bei der deutlich erkennbare

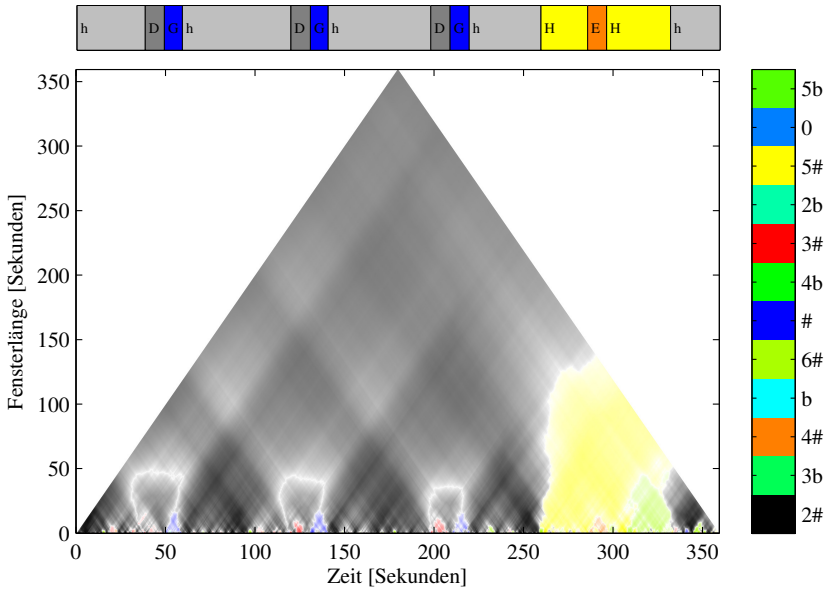


Abbildung 4.8.: Scape-Plot-Darstellung der harmonischen Struktur des ersten Stücks der Winterreise in der Aufnahme von Quasthoff. Die schwarze Farbe kennzeichnet die erkannte Moll-Tonika h-Moll sowie die Parallele D-Dur, gelb die gleichnamige Durtonart H-Dur, rot die Dominante $f^{\sharp}$-Moll und blau die Subdominante e-Moll. Man beachte, dass der Interpret das Stück im Vergleich zur Annotation in Abbildung 4.5 um eine kleine Terz nach unten transponiert hat.

Farbunterschiede für Tonarten nahe der Tonika und kleine für entfernte Tonarten verwendet werden. Weiterhin führt die strikte Trennung von Blautönen für die subdominantischen und Rottönen für die dominantischen Nachbartonarten zu einer klaren Darstellung der harmonischen Beziehungen, was in Abbildung 4.8 dargestellt wird. Man beachte beim Vergleich der absoluten Tonarten bzw. der durch die Anzahl der Vorzeichen angegebene Position auf dem Quintenzirkel, dass bei dieser Aufnahme das Stück um eine kleine Terz nach unten transponiert wurde, die Tonika hier also h-Moll, die gleichnamige Durtonart H-Dur und die Durparallele D-Dur ist. Die Referenzannotation in der Abbildung wurde daher ebenfalls transponiert.

Diese Darstellung hat jedoch den Nachteil, dass nur die Position auf dem Quintenzirkel verwendet wird und die Information über das Tongeschlecht (Dur, Moll) unberücksichtigt bleibt, wodurch die jeweiligen Paralleltonarten (in diesem Fall die Tonika h-Moll und ihre Paralleltonart D-Dur, beide 2♯) in derselben Farbe (hier schwarz) dargestellt werden. Auch die (Moll-)Dominante fis-Moll bzw. die Dominantparallele A-Dur mit jeweils 3♯ werden beide durch rot, die Subdominante e-Moll und ihre Parallele G-Dur mit 1♯ durch blau gekennzeichnet. Folglich genügt eine reine Scape-Plot-Darstellung mit dieser Farbkodierung nicht zur

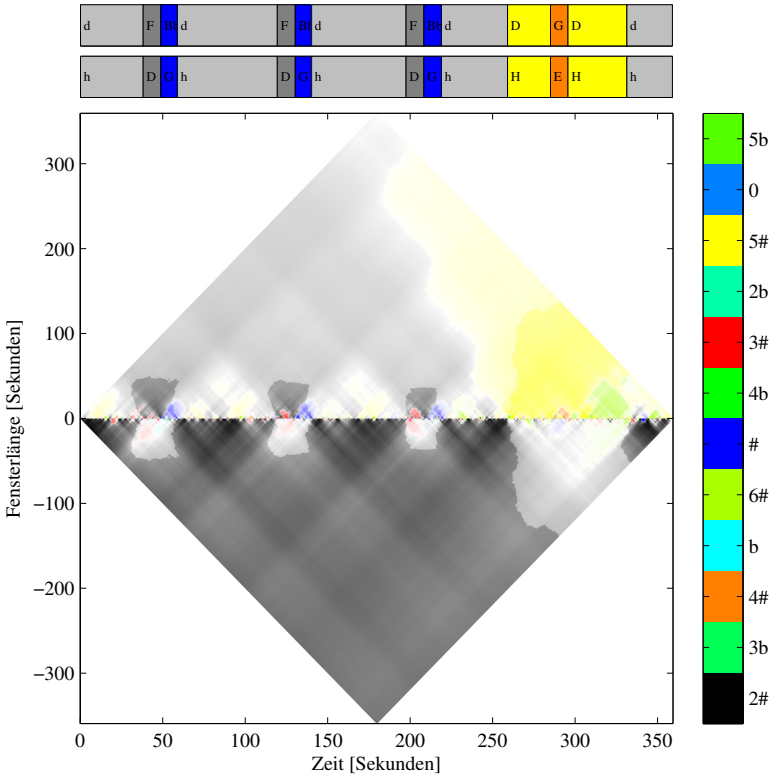


Abbildung 4.9.: Darstellung als Rhombus-Plot des ersten Stücks der Winterreise in der Aufnahme von Quasthoff. Das obere Dreieck beschreibt die Dur-, das untere die Molltonarten. Die obere Annotation bezeichnet die notierten Tonarten, die untere die tatsächlichen, die mit den Vorzeichen aus dem Rhombus-Plot übereinstimmen.

Unterscheidung aller Tonarten, weswegen wir eine Erweiterung zu einem Doppeldreieck (oder *Rhombus-Plot*) vorschlagen.

Bei dieser Darstellung betrachten wir die Abschätzungen für die beiden Tongeschlechter separat, bestimmen also für Dur und Moll unabhängig voneinander die wahrscheinlichsten Tonarten und als *clarity*-Werte die Abstände zur jeweiligen zweitwahrscheinlichsten. Somit entstehen zwei Scape-Plots, die wir in Form eines Doppeldreiecks anordnen, wobei wir nach Konvention das obere Dreieck für die Durtonarten und das untere für die Molltonarten verwenden[5]. In einem zweiten Schritt vergleichen wir für jeden Punkt den *clarity*-Wert für die geschätzte Dur- mit dem für die geschätzte Molltonart. Wir gehen davon aus, dass das tatsächlich vorliegende Tongeschlecht über den höheren Wert verfügt. Um dies in der Visualisierung

[5] Diese Konvention ist durch die als *Tonnetz* bezeichnete Anordnung der Halbtöne einer Oktave motiviert, bei der ein Durdreiklang einem nach oben zeigendem und ein Molldreiklang einem nach unten zeigendem Dreieck entspricht, vgl. [70].

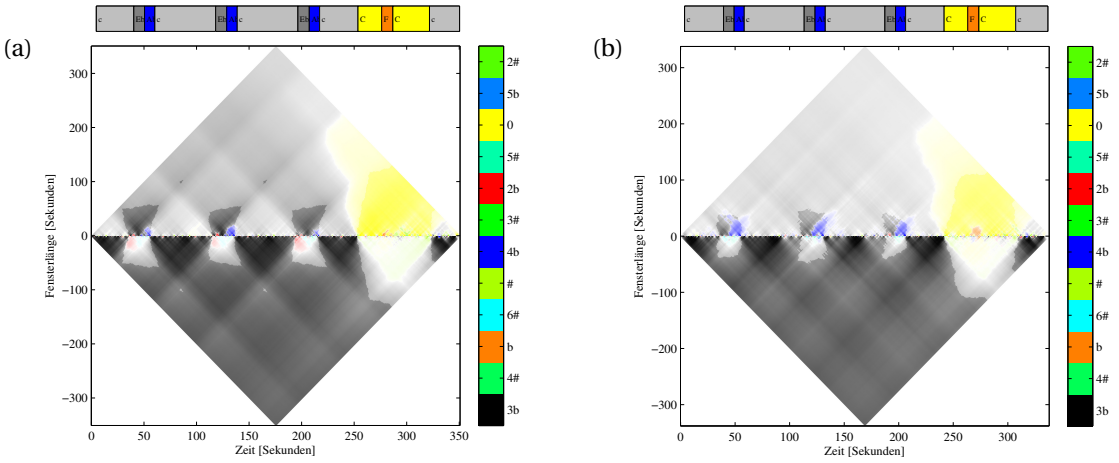


Abbildung 4.10.: Darstellung als Rhombus-Plot des ersten Stücks der Winterreise. **(a)** Allen, **(b)** Fischer-Dieskau (mit Moore).

kenntlich zu machen, verringern wir die Farbintensität des Punktes mit dem niedrigeren Wert zusätzlich auf 33% des ursprünglichen Wertes.

In Abbildung 4.9 ist diese Darstellung für das erste Stück der Winterreise in der Aufnahme von Quasthoff illustriert. Dieses Stück besteht aus vier Strophen, wobei die ersten drei in d-Moll stehen und die letzte in D-Dur. Die Strophen selbst sind dreigeteilt, wobei im Mittelteil kurz nach F-Dur (bzw. auf einer feineren Skala zuerst nach F-Dur und anschließend nach $B^{\flat}$-Dur) moduliert wird. Der Schluss des Stückes steht wieder in d-Moll. Wir erinnern an dieser Stelle noch einmal an die Transponierung um eine kleine Terz nach unten, die Tonika der Referenzannotation ist d-Moll, die der Aufnahme und des Scape-Plots h-Dur. Die obere der beiden Annotationen zeigt die notierte Tonart, die untere die mit dem Scape-Plot übereinstimmenden Tonarten dieser Aufnahme.

In der Abbildung ist dies durch das stark gefärbte, schwarze nach unten weisende Dreieck gekennzeichnet. Die vierte Strophe entspricht dem gelben Bereich im oberen Dreieck (gleichnamige Durtonart). Die Mittelteile der ersten drei Strophen sind durch dunkelgrauen Bereiche im oberen Dreieck gekennzeichnet, dort sind auch einige blaue Flecken zu erkennen, welche für die Subdominante (bzw. deren Paralleltonart) stehen. Auch sind einige kleinere rote Bereiche erkennbar, die auf Verwendung der Dominanttonart hinweisen sowie gegen Ende der vierten Strophe ein gelbgrüner Bereich, der für die Dominante der gleichnamigen Durtonart steht.

Ein Vergleich dieser Darstellung für mehrere Aufnahmen dieses Stückes zeigt, dass die Schätzungen der lokal vorherrschenden Tonart relativ stabil sind. Dennoch zeigen sich zwischen den Einspielungen von Quasthoff (Abb. 4.9), Allen (Abb. 4.10a) und Fischer-Dieskau (Abb. 4.10b) leichte Abweichungen, insbesondere bei der Intensität der Subdominante im Mittelteil der ersten drei Strophen. Da zur Festlegung der Farbinformationen ausschließlich die Tonart ver-

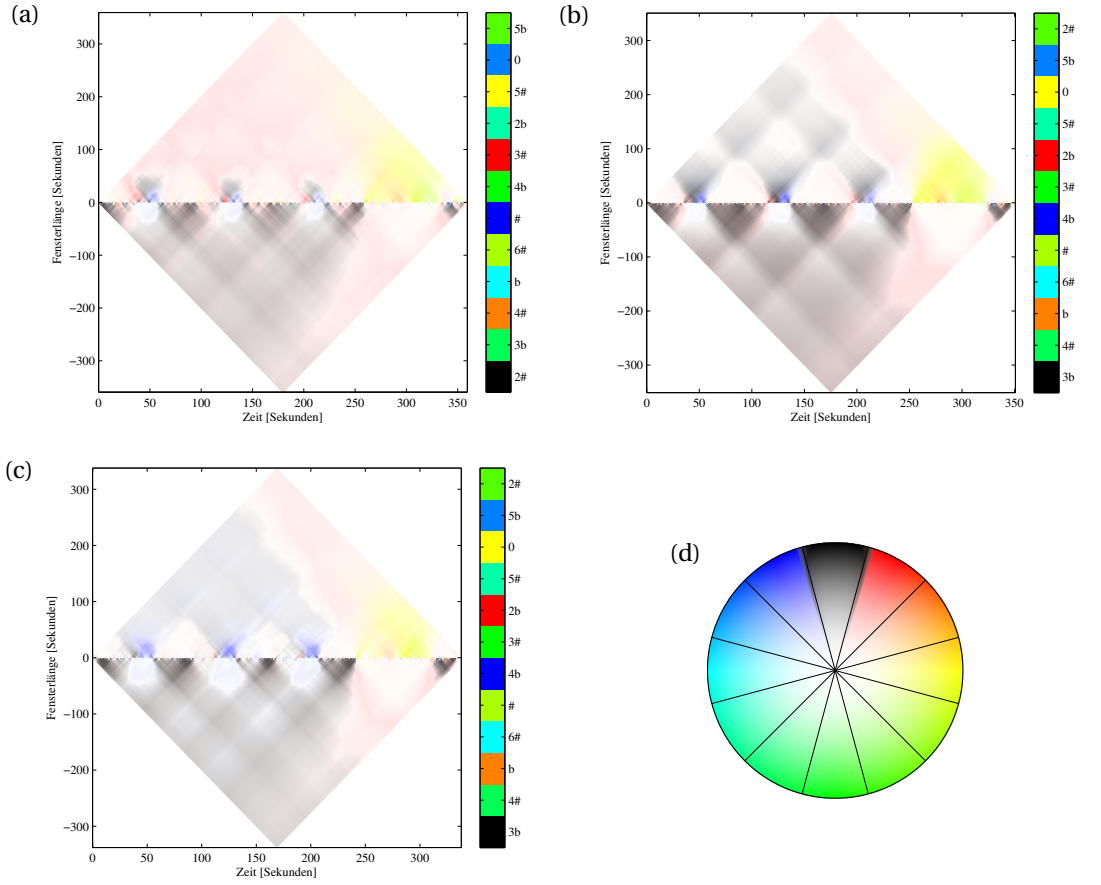


Abbildung 4.11.: Rhombus-Plot-Darstellungen mittels kontinuierlicher Farbgebung für die Interpretationen des ersten Stücks der Winterreise von **(a)** Quasthoff, **(b)** Allen, **(c)** Fischer-Dieskau (mit Moore). **(d)** Kontinuierliche Variante der Farbskala für Tonarten aus Abbildung 2.7.

wendet wird, deren entsprechende Komponente maximale Energie innerhalb des einzelnen Key-Vektors aufweist, stellt sich nun die Frage, ob die Unterschiede in den verschiedenen Interpretationen durch diese harten Entscheidungen verursacht worden sind oder ob sie aus Eigenschaften der zugrundeliegenden Chroma-Merkmale folgen. Ein Beispiel für ein aus diesen Entscheidungen resultierendes Artefakt ist bei der Aufnahme von Allen (Abb. 4.10a) an den Positionen 80 s und 160 s bei Fensterlänge 100 s zu sehen.

Hierzu untersuchen wir zum Abschluss dieses Abschnitts eine kontinuierliche Variante unserer Rhombus-Plot-Darstellung, bei der die Tonart inklusive Modus nicht mittels Betrachtung eines Maximums, sondern aus einer Art Schwerpunktberechnung des gesamten Key-Merkmalsvektors berechnet wird. Hierzu betrachten wir wie oben die Komponenten jedes

Key-Vektors separat für die Dur- wie für die Moll-Tonarten. Mit den beiden Vektoren

$$k_{\text{Dur}} = (k_{\text{C}}, k_{\text{G}}, \cdots, k_{\text{F}}) \in [0,1]^{12} \quad \text{und} \quad k_{\text{Moll}} = (k_{\text{a}}, k_{\text{e}}, \cdots, k_{\text{d}}) \in [0,1]^{12}$$

bezeichnen wir die zur Beschreibung der Dur- bzw. Moll-Tonarten verwendete Hälfte eines ℓ^1-normalisierten Key-Vektors, also $\sum k_{\text{Dur}} + \sum k_{\text{Moll}} = 1$. Weiterhin bezeichnen wir mit $\mathbf{e} = \left(e^{2\pi i n/12}\right)_{n\in[0:11]}$ die Menge der 12-ten Einheitswurzeln.

Die Winkel der komplexen Zahlen

$$z_{\text{Dur}} = \mathbf{e} \cdot k_{\text{Dur}}^{\top} \quad \text{und} \quad z_{\text{Moll}} = \mathbf{e} \cdot k_{\text{Moll}}^{\top}$$

können dann als kontinuierliche Variante der geschätzten Dur- bzw. Moll-Tonart interpretiert werden und die Absolutbeträge als deren Konfidenz. Mittels der kontinuierlichen Tonarten-Farbskala in Abbildung 4.11d wird so jedem Punkt des Scape-Plots eine Farbe zugeordnet.

Diese Variante der Farbgebung führt zu weichen Übergängen zwischen den Tonarten, insbesondere entfällt die harte Dur-/Moll-Entscheidung, wodurch die oben erwähnten Artefakte reduziert werden. Andererseits ist eine solche Darstellung anfällig gegenüber Wechseln in weiter entfernte Tonarten, da durch die kontinuierlichen Farbübergänge somit alle Zwischentonarten fälschlicherweise als lokale Tonarten interpretiert werden, wodurch sie nur als Erweiterung, nicht aber als Ergänzung der oben vorgestellten Variante gesehen werden sollte.

Ein Vergleich der in Abbildung 4.11a–c dargestellten kontinuierlichen Scape-Plots zeigt auch hier große Unterschiede zwischen den drei betrachteten Interpretationen. Folglich resultieren diese nicht aus den bei Erstellung der »normalen« Scape-Plot-Darstellungen zu treffenden Entscheidungen, sondern sind bereits in den Key- und damit auch in den Chroma-Merkmalen enthalten. Diese Schlussfolgerung wird auch durch die teilweise sehr großen Unterschiede zwischen den einzelnen Interpreten bei der automatischen Tonartschätzung in Abbildung 4.6 gestützt. Bei Vorliegen mehrerer synchronisierter Aufnahmen können diese Abweichungen im Rahmen einer harmonischen Strukturanalyse dazu verwendet werden, bezüglich ihrer Harmonik übereinstimmende Passagen mit einer höheren Konfidenz zu bewerten als abweichende und somit die Akkord- oder Tonartenerkennung zu stabilisieren [86].

4.4. Ein Merkmal für Gesangserkennung

Die für die Strukturierung des *Winterreise*-Datensatzes relevanten musikalischen Aspekte umfassen neben den bereits besprochenen Wiederholungen und der harmonischen Progression auch die Position der Gesangspassagen. Während die musikalischen Komponenten Rhythmus und Harmonik recht eindeutig verschiedenen physikalischen Aspekten eines Musiksignales zugeordnet werden können, bleiben die mit dem Begriff der Klangfarbe assoziierten Eigenschaften oftmals nur wage definiert und werden meistens mit der Energieverteilung in den Spektralvektoren beschrieben [2, 160, 175].

Wie bereits in Abschnitt 2.3 vorgestellt, werden zur Analyse von Musiksignalen bezüglich ihrer Klangfarbe häufig die *unteren MFCC-Merkmale* verwendet, welche die grobe Form der spektralen Energieverteilung beschreiben [192]. Ein anderer Aspekt ist die Obertonverteilung jedes Instruments, die allerdings nicht rein instrumentenspezifisch ist, sondern zusätzlich von der Tonhöhe und der verwendeten Spieltechnik abhängt. So weist beispielsweise ein Konzertflügel in den tieferen Lagen eine deutlich andere Obertonstruktur auf als in den hohen Lagen, ebenso unterscheiden sich die Obertöne bei einem leicht angeschlagenen Ton deutlich von einem *espressivo* gespielten.

Weiterhin handelt es sich bei der Klangfarbe um ein sehr komplexes Phänomen, das nicht nur von der Instrumentierung, sondern auch von den Aufnahmebedingungen abhängt. Schon 1939 war Architekten bekannt, wie der umgebende Raum über Echo- und Nachhalleffekte einen nicht unerheblichen Einfluss auf die spektrale Energieverteilung hat, da die verwendeten Baustoffe verschiedene Frequenzen in unterschiedlichem Maße dämpfen [201, 202]. Nach [201] führen etwa verschieden lange Nachhallzeiten bei tiefen und hohen Tönen zu einem dumpfen bzw. schrillen Klangeindruck, was der Verteilung der Schallenergie im Bereich der tiefen bzw. hohen Frequenzen entspricht. Der umgebende Raum stellt also ein akustisches Filter mit einer zusätzlich von den Standorten des Schallerzeugers und des Aufzeichnungsgerätes abhängigen Frequenzantwort dar. Nicht zuletzt weisen auch die Geräte zur Schallaufzeichnung individuelle Eigenschaften auf, die ebenfalls Einfluss auf die im Spektrogramm dargestellten Signaleigenschaften nehmen. Eine Repräsentation als MFCC-Vektoren beinhaltet daher zwangsläufig immer auch akustische Merkmale, die nicht direkt in Bezug zum aufgenommenen Stück, sondern zu den Umständen der Aufnahme selbst gehören.

Im Gegensatz zur komplexen Problemstellung des Erkennens einzelner Instrumente in einem polyphonen Audiosignal (siehe [52] für eine Übersicht), können wir uns bei der Winterreise auf die Unterscheidung zwischen reinen Klavierpassagen und Zeitintervallen mit Gesang konzentrieren. Hierzu werden häufig allgemeine Deskriptoren für die Energieverteilung in den Spektralvektoren des Magnitudenspektrogramms verwendet, die anschließend mittels verschiedener Techniken maschinellen Lernens zur Klassifikation herangezogen werden, siehe hierzu etwa [164, 184, 214]. In [185] wird beobachtet, dass die Obertöne bei gesprochenem Text üblicherweise in der Frequenz variieren. Daraus wird ein Merkmal zur Erkennung von Sprache in einem Audiosignal entwickelt, indem die spektralen Muster von nahe beieinander liegenden Zeitpunkten verglichen und diese Frequenzabweichungen mittels eines auf Entscheidungsbäumen basierenden Lernverfahrens zur Klassifikation verwendet werden. In [98] wird dieses Merkmal in Kombination mit den unteren fünf MFCCs und statistischen Kennzahlen der Spektralvektoren zur Erkennung von Gesang eingesetzt. Ebenfalls auf eine Analyse der Abweichungen in den Obertönen beruht das in [136] vorgestellte Verfahren. Hier werden zusätzliche hohe Werte in den Spektralvektoren gesucht, die nicht an den üblichen Positionen der Obertöne auftreten, und diese zur Erkennung von Gesangspassagen verwendet.

In [97] wird angemerkt, dass im Allgemeinen die einfachen MFCCs zu mindestens ebenso guten Ergebnissen führen wie verfeinerte und speziell auf konkrete Anwendungen optimierte Merkmale. Für uns legt dies den Schluss nahe, dass für die spezialisierten Verfahren des

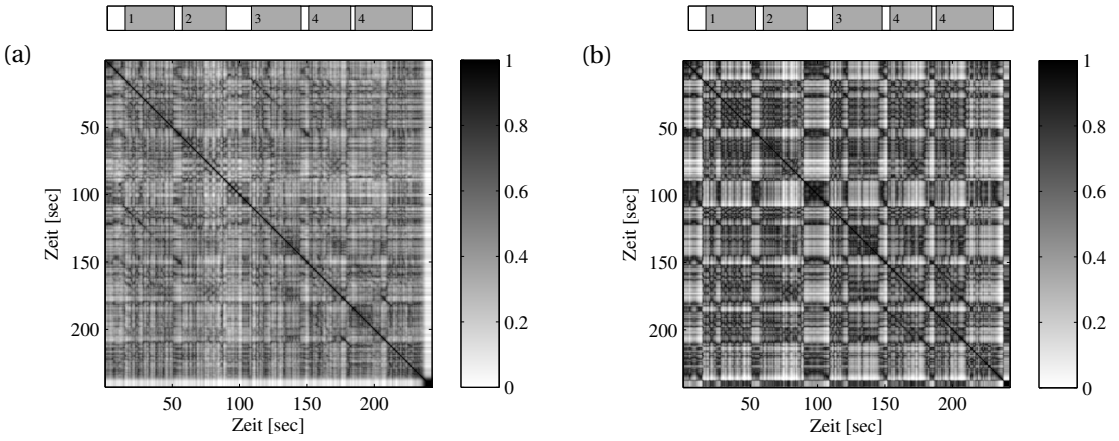


Abbildung 4.12.: Selbstähnlichkeitsmatrizen für Klangfarbenmerkmale des 20. Stücks in der Aufnahme von Hüsch mit Annotation der gesungenen Strophen. **(a)** Verwendung der üblichen MFCC-Merkmale, **(b)** Verwendung modifizierter MFCC-Merkmale.

maschinellen Lernens eine präzise binäre Aufgabenstellung vorliegt, die gut durch die in den Merkmalen erhaltenen Informationen gelöst werden kann, da die MFCCs als eine komprimierte Version des Spektrogramms gesehen werden können. Folglich scheint eine Optimierung der Merkmale bei Verwendung dieser automatischen Lernmethoden nicht notwendig zu sein.

Die zur Gesangserkennung relevanten Informationen liegen zudem in dem hier betrachten Fall der Unterscheidung zwischen Gesang und Klavier deutlich genug in den MFCCs vor, um eine Lösung auch mittels homogenitätsbasierter Segmentierung durch nicht-negative Matrixfaktorisierung (vgl. Abschnitt 2.6.1) erhalten zu können. In Abbildung 4.12a ist eine Selbstähnlichkeitsmatrix des 20. Stückes der Winterreise »Der Wegweiser« in der Aufnahme des European Archive (Hüsch, 1933) bei Verwendung der üblichen MFCC-Merkmale illustriert. Diese Matrix weist zwar einige Blockstrukturen auf der Diagonalen auf, die zum Teil mit den gesungenen Strophen bzw. den Instrumentalpassagen zwischen diesen übereinstimmen, allerdings sind diese nicht sehr ausgeprägt und weisen kaum klare Kanten auf. Weiterhin fehlen die Blöcke außerhalb der Diagonalen, die zur Benennung der Segmente notwendig sind.

In diesem Abschnitt stellen wir eine Erweiterung dieser MFCC-Merkmale vor, bei deren Verwendung die Selbstähnlichkeitsmatrix deutlichere Blockstrukturen (siehe Abbildung 4.12b) aufweist, die zur Strukturierung des Stückes nach Gesangspassagen und rein instrumentalen Zwischenspielen geeignet ist. Die Kernidee des vorgestellten Verfahrens ist eine Abschätzung, ob die Frequenz eines einmal angeschlagenen Tons in der Klangphase (engl. *sustain*) noch Änderungen aufweist. Dieses Phänomen tritt beispielsweise bei Gesang sehr stark auf, in schwächerem Maße auch bei Streich- und Blasinstrumenten, insbesondere bei *Vibrati*. Bei einem Klavier oder einer Orgel geschieht dies in der Regel nicht, die Tonhöhe eines einmal angeschlagenen Tones verändert sich nicht mehr.

Ausgehend von einer Kurzzeit-Fouriertransformation verwenden wir die zeitliche Änderung der Phaseninformationen innerhalb der einzelnen Frequenzbänder zur Schätzung dieser Frequenz-Stabilität der einzelnen Töne. Dies nutzen wir anschließend zum Abschwächen der frequenzstabilen Töne im Magnitudenspektrogramm. Von dem so modifizierten Spektrogramm ausgehend berechnen wir die üblichen unteren MFCC-Merkmale, stellen die Selbstähnlichkeitsmatrix auf und wenden darauf die homogenitätsbasierte Segmentierung mittels *sparse-NMF* an, vgl. Kapitel 2.

Die Phaseninformationen werden intensiv im Umfeld der *Phase Vocoder*- Verfahren verwendet. Das aus den Begriffen *voice* und *coder* zusammengesetzte Kunstwort *vocoder* beschreibt eine sogenannte Analyse-Synthese-Technik, bei der ein Eingabesignal gemäß eines zeitabhängigen Modells (Analyse) in ein ggf. modifiziertes Ausgabesignal (Synthese) umgewandelt wird. Beim *Phase Vocoder* wird die Phaseninformation für Skalierungen sowohl im Zeit- als auch im Frequenzbereich verwendet, bei denen die charakteristischen Variationen in den Frequenzbändern erhalten bleiben [35]. Der *Phase Vocoder* findet ebenfalls Anwendung in der Quellentrennung mittels der Trennung zwischen transienten und stabilen Zuständen (*transient/steady-state (TSS) separation*) [42]. In [4] werden einige Verfahren zur Erhöhung der Auflösung (engl. *reassignment methods*) in Zeit-Frequenz-Darstellungen eines Signals durch Verwendung der Ableitung von Phaseninformationen in Zeitrichtung (in diesem Kontext als *instantaneous frequency* bekannt) diskutiert. In [53] wird eine algorithmische Verbindung zwischen der gefensterten Fouriertransformation und dem Konzept der *instantaneous frequency* hergestellt. Die Erkennung von Noteneinsatzzeiten (*onset detection*) kann ebenfalls mittels Zuhilfenahme der Phaseninformation verbessert werden, für eine Übersicht siehe [6, 72]. Außerhalb der Musikanwendung werden Phaseninformationen beispielsweise auch zum Erkennen von Materialeigenschaften verwendet [101].

4.4.1. Algorithmus

Die Kurzzeit-Fouriertransformation (engl. *Short-Time Fourier Transform*, STFT) eines diskreten Musiksignals $x : \mathbb{Z} \to \mathbb{C}$ mit einer Fensterfunktion $w \in \ell^2(\mathbb{Z})$ der Länge N und einer Schrittweite (engl. *hopsize*) von $h \in \mathbb{N}$ Zeitpunkten (engl. *samples*) ist gegeben durch

$$X(k,t) = \sum_{n=0}^{N-1} x(n+ht) \cdot w(n) \cdot e^{-2\pi i k n/N} .$$

Als Fensterfunktion verwenden wir üblicherweise ein Hamming-Fenster[6] der Länge $N = 1024$, als Schrittweite h die halbe Fensterlänge. Da unsere Musiksignale im Allgemeinen eine Abtastrate von 22,05 kHz aufweisen, ergibt sich für unsere STFT eine zeitliche Auflösung von 23 ms und eine Frequenzauflösung von 21,5 Hz.

[6] Das *Hamming-Fenster* der Länge $N \in \mathbb{N}$ ist für $n \in [0 : N-1]$ definiert als $w(n) = \frac{1}{46} \cdot \left(25 - 21 \cdot \cos\left(\frac{2\pi n}{N-1}\right)\right)$.

Der quadrierte Absolutbetrag der STFT entspricht dem *Spektrogramm* (das manchmal auch als Magnitudenspektrogramm, engl. *power spectrum*, bezeichnet wird) $S(k,t) := |X(k,t)|^2$, analog dazu verwenden wir den Ausdruck *Phasenspektrogramm* für die Matrix der Phaseninformationen $\Phi(k,t) := \arctan(X(k,t))$.

Bei Betrachtung der STFT eines Tons mit konstanter Frequenz erwarten wir bedingt durch die konstante Schrittweite, dass sich die Phasen zweier aufeinanderfolgender Spektralvektoren um einen konstanten, von der exakten Frequenz abhängigen Wert ändern – hierbei betrachten wir diese Differenzen modulo 2π, da die Phaseninformationen üblicherweise im Bereich $(-\pi,\pi]$ bzw. $[0,2\pi)$ angegeben werden. Dies bezeichnen wir im Folgenden als diskrete Ableitung der Phaseninformation in Zeitrichtung und schreiben $\frac{\partial}{\partial t}\,\Phi(k,t)$. Da der Wert dieser konstanten Differenz selbst für unsere Anwendung unerheblich ist, berechnen wir analog die zweite diskrete Ableitung in Zeitrichtung

$$\Delta(k,t) := \begin{cases} \frac{\partial^2}{\partial t^2}\,\Phi(k,t) & t \in (0,T), \\ 0 & t \in \{0,T\}, \end{cases}$$

wobei wir diesmal die Werte im Intervall $(-\pi,\pi]$ betrachten.

Die Matrix Δ enthält nun Nullwerte in den Zeit-Frequenz-Zellen, bei denen ein frequenzstabiler Ton ausgehalten wird. Bei punktweiser Multiplikation von Δ mit dem Magnitudenspektrogramm S werden nun diese Töne abgeschwächt und die Töne mit veränderlichen Frequenzen bzw. Toneinsätze bleiben erhalten. Das durch die Gewichtung modifizierte Spektrogramm bezeichnen wir mit

$$\tilde{S}(k,t) := S(k,t) \cdot \tfrac{1}{\pi}|\Delta(k,t)|.$$

In Abbildung 4.13 wird der Effekt dieses Vorgehens illustriert. Die betrachtete Stelle besteht aus den Takten 22–27 des ersten Stückes der Winterreise, in denen die Gesangsstimme eine Pause von zwei Takten Länge aufweist, siehe Abbildung 4.13a. Im Abbildungsteil (b) wird nun das Spektrogramm der entsprechenden Passage in der Aufnahme von Hüsch dargestellt. Hierbei zeigen die Helligkeitswerte die Energieverteilung im Magnitudenspektrogramm an, wohingegen die Farben die aus dem Phasenspektrogramm gewonnenen Δ-Werte anzeigen. Die Werte von Δ werden dabei auf den Farbkreis abgebildet, somit wird ein kleiner Wert mittels roter Farbe dargestellt, die sporadisch auftretenden mittleren Werte in gelb bzw. violett und die seltenen hohen Werte in grün, türkis oder blau. In ebendieser Abbildung sind aufgrund der hohen Auflösung diese Farbnuancen meist nur als Grauton wahrzunehmen. In Abbildung 4.13c ist das ursprüngliche Spektrogramm S illustriert, welchem in Teil (d) das durch punktweise Multiplikation mit den Betragswerten von Δ gewichtete Spektrogramm $\tilde{S}$ gegenübergestellt wird. In den Abbildungen 4.13e und 4.13f wird der entsprechende Ausschnitt der Selbstähnlichkeitsmatrizen illustriert, die mittels Berechnung von MFCC-Merkmalen aus den beiden Spektrogrammen entstanden sind. Hierbei wird deutlich, dass diese Modifikation zu einer merklich schärferen Unterscheidung von Klavier- und Gesangspassagen führt.

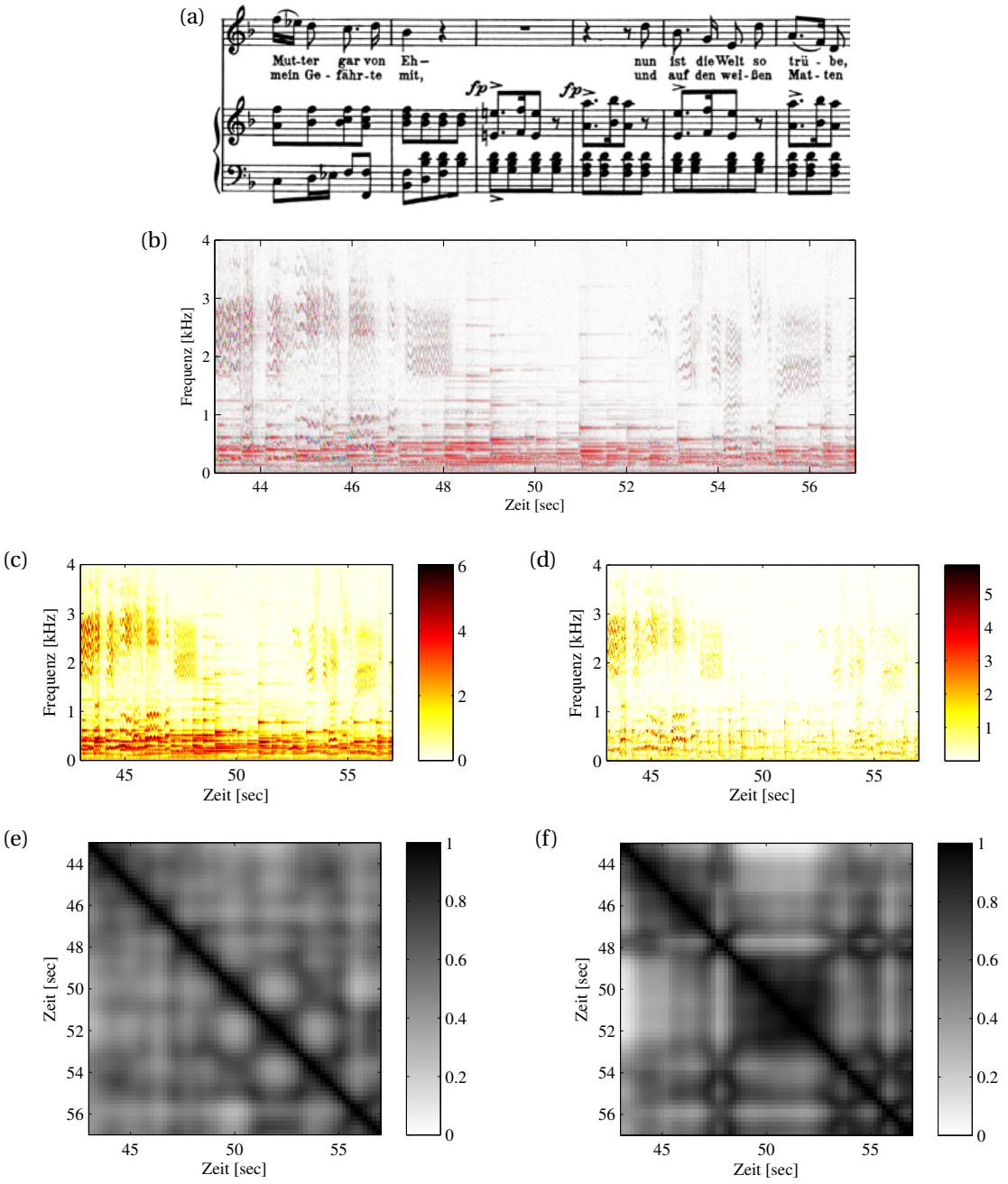


Abbildung 4.13.: Auszug aus dem ersten Stück »Gute Nacht« (Takte 22–27): **(a)** Notentext, **(b)** Spektrogramm überlagert mit den farbkodierten Werten von Δ (rot ist nahe Null), **(c)** ursprüngliches Spektrogramm S, **(d)** mit den Betragswerten von Δ punktweise gewichtetes Spektrogramm $\tilde{S}$, **(e)**, **(f)** entsprechende Auszüge aus den resultierenden Selbstähnlichkeitsmatrizen bei Verwendung von MFCC-Merkmalen.

4.4.2. Evaluation

Im Folgenden überprüfen wir die Deskriptivität der modifizierten MFCC-Merkmale für die homogenitätsbasierte Segmentierung. Hierzu vergleichen wir eine Segmentierung nach Gesangspassagen und instrumentalen Zwischenspielen mittels verschiedener Techniken. Diese umfassen zum einen die Segmentierung mittels sparse-NMF der Selbstähnlichkeitsmatrizen aus den unteren 13 MFCC-Komponenten (ohne die erste), die zum einen aus dem originalen und zum anderen aus dem modifizierten Spektrogramm abgeleitet wurden. Für das NMF-Verfahren erlauben wir nur zwei Klassen und verwenden eine Kontrastverstärkung der Selbstähnlichkeitsmatrizen, um sicherzustellen, dass die beiden Klassen tatsächlich die Klavier- und Gesangspassagen abbilden. Zusätzlich werden die Matrizen mit einem Gaußfenster der Länge 3 s geglättet.

Weiterhin vergleichen wir[7] diese Ergebnisse mit einem aktuellen, auf [97] basierenden Verfahren, welches auf einer automatischen Merkmalsauswahl und maschinellem Lernen beruht. Dieses Verfahren wurde mittels zufällig gewählter Zeitpunkte trainiert, die insgesamt 10% der ersten 8 Stücke aller Interpreten ausmachen. Als Baseline dient die Annahme, dass das gesamte Stück aus Gesang besteht.

Der verwendete Datensatz für diese Evaluation umfasst die vier Aufnahmen der Winterreise von Allen, Oliemans, Quasthoff und Trekel. Als Referenz dient die Gesangsstimme in den mit den einzelnen Aufnahmen synchronisierten MIDI-Dateien. Daher beinhalten diese Annotationen auch einige sehr kurze Segmente, die perzeptuell nicht als Instrumentalpassage bezeichnet würden.

Als Beispiel illustrieren wir in Abbildung 4.14 die erhaltene Segmentierung nach Gesangspassagen anhand der normalen (Teil a) und der modifizierten (Teil b) MFCC-Merkmale. Obwohl die Strukturen in der Matrix (a) deutlich schwächer sind als in (b), konnte das sparse-NMF-Verfahren das instrumentale Vor- und Nachspiel erfolgreich separieren. Auch die in der Mitte auftretenden kurzen Zwischenspiele wurden erkannt, allerdings nicht in ihrer vollen Länge. Bei Verwendung der modifizierten Merkmale (Abb. 4.14b) treten einerseits die Strukturen in der Matrix deutlicher zutage, zum anderen weist die Segmentierung mehr Detailstrukturen auf. Insbesondere die Abfolge zwischen dem längeren und dem kurzen Zwischenspiel um die 40. Sekunde des Stücks wurde exakt erkannt. Die Zwischenspiele mit minimaler Länge wurden aufgrund der Matrixglättung nicht gefunden.

Für eine quantitative Analyse wählen wir eine punktweise Benennung bei einer Auflösung von 5 Hz, bei der wir jedem Zeitpunkt mittels der betrachteten Verfahren entweder die Bezeichnung »Gesang« oder »kein Gesang« zuordnen. Bei den beiden NMF-basierten Segmentierern führen wir diese Zuordnung mittels einer empirisch ermittelten Regel durch: Für beide Bezeichnungen wird der Durchschnitt aller Merkmalsvektoren mit derselben Bezeichnung berechnet. Derjenige Durchschnittsvektor, der über weniger Gesamtenergie in den unteren drei

[7] Dieses Verfahren wurde von Christian Dittmar in den *International Audio Laboratories Erlangen* implementiert, der auch die Experimente mit diesem Verfahren durchgeführt hat.

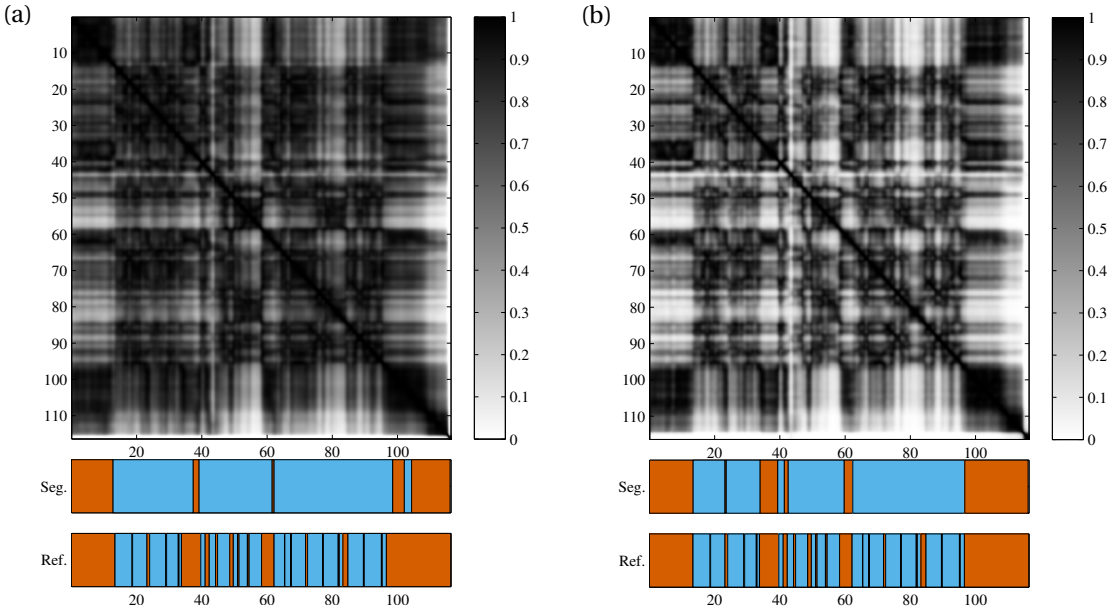


Abbildung 4.14.: Automatische Segmentierung des 15. Stücks der Winterreise nach Gesang (blau) und Klavier (orange) mittels der dargestellten Selbstähnlichkeitsmatrix. Unten die manuelle Annotation als Referenz. **(a)** Standard-MFCC, **(b)** modifizierte MFCC.

Komponenten verfügt, wird als Gesang angesehen. Diese heuristische Regel hat bei allen 96 betrachteten Einzelaufnahmen zum richtigen Ergebnis geführt. Die so erhaltene punktweise Zuordnung vergleichen wir anschließend mittels des üblichen F-Measures mit den aus den synchronisierten MIDI-Daten erhaltenen Referenzannotationen.

In Tabelle 4.3 sind die Ergebnisse der Evaluation auf allen 96 betrachteten Aufnahmen dargestellt. Diese zeigt, dass durch Verwendung der modifizierten MFCC-Merkmale mit einem durchschnittlichen F-Measure von 0,893 eine spürbare Verbesserung gegenüber den ursprünglichen MFCCs mit einem F-Measure von 0,855 erzielt wurde. Beide homogenitätsbasierten Segmentierungen liegen deutlich über dem Baseline-Wert von 0,745. Das Resultat des auf diesen Datensatz mittels maschinellen Lernens optimierten Verfahrens von 0,951 konnte nicht erreicht werden.

Folglich ist die homogenitätsbasierte Segmentierung zur Unterscheidung zwischen Klavier- und Gesangspassagen grundsätzlich geeignet. Eine Steigerung der Güte wurde durch die Verwendung von Phaseninformationen zur Modifikation der betrachteten MFCC-Merkmale erreicht, die allerdings nicht an die durch maschinelles Lernen erzielbaren Ergebnisse heranreicht.

Die in Abbildung 4.15 dargestellte Einzelanalyse weist auf Schwankungen der Güte des Verfahrens bei den einzelnen Stücken von etwa 0,7 bis 0,98 hin, wobei die Ergebnisse bei einem Großteil der Stücke oberhalb des Mittelwerts von 0,893 liegen. Die in der letzten Zeile für jedes

Merkmal	F-Measure		Kommentar
	Mittelwert	Std.abw.	
modifizierte MFCC	0,893	0,051	SSM und sparse-NMF
Standard-MFCC	0,855	0,063	SSM und sparse-NMF
Variante von [97]	0,951	0,019	maschinelles Lernen
Baseline	0,745	0,060	Annahme: Nur Gesang

Tabelle 4.3.: Ergebnisse der automatischen Gesangserkennung auf 4 Komplettaufnahmen der *Winterreise*.

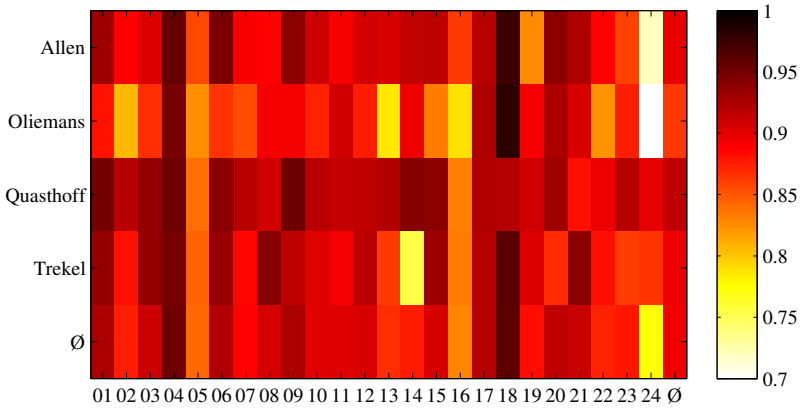


Abbildung 4.15.: Detaillierte Übersicht der Ergebnisse bei der automatischen Gesangserkennung auf 4 Komplettaufnahmen der *Winterreise*.

Stück einzeln angegebenen Durchschnittswerte über die Ergebnisse aller vier Interpreten zeigen, dass die Gesangserkennung bei den Stücken 01, 04, 17 und 18 besonders gut und die bei den Stücken 05, 16 und 24 deutlich schlechter gelungen ist. Eine Einzelanalyse der anderen Verfahren unterstützt diese Beobachtung, weswegen wir dies eher auf Besonderheiten der Stücke selbst als auf Eigenschaften der betrachteten Verfahren zurückführen. So weist Stück Nr. 05 »Der Lindenbaum« insbesondere in der 4. und in der 6. Strophe einige lang ausgehaltene Töne auf, die in den Aufnahmen nur sehr leise zu hören sind und von der an diesen Stellen exponierten Klavierbegleitung übertönt werden. Die Klavierbegleitung in Stück Nr. 16 »Letzte Hoffnung« besteht überwiegend aus Staccati, die sich durch das Fehlen einer längeren Ausklingphase spektral deutlich von den übrigen Klavierpassagen unterscheiden. Das letzte Stück »Der Leiermann« wird von vielen Sängern vergleichsweise zurückhaltend gesungen, wodurch sich auch hier die Klavierbegleitung teilweise deutlicher abhebt.

Auch beim Vergleich der Durchschnittswerte über alle Lieder eines einzelnen Interpreten (siehe letzte Spalte) ergeben sich Unterschiede von etwa 0,05 Punkten, wobei zumeist die besseren Ergebnisse beim Sänger Quasthoff erzielt werden konnten sowie die schwächeren

bei Oliemans. Bei den anderen beiden Verfahren liegt die niedrigste Güte der vier Interpreten ebenfalls bei Oliemans.

Speziell bei dem hier vorgestellten Verfahren fallen beim letzten Stück einige sehr niedrige Evaluationswerte auf, die bei den anderen Verfahren nicht auftreten. Eine mögliche Erklärung für dieses Phänomen liegt in der Gewichtung des Magnitudenspektrogramms mit den abgeleiteten Phaseninformationen, was einige Eigenschaften eines Schwellwertverfahrens aufweist. Da insbesondere die Gesangsstimme des Interpreten Oliemans bei diesem Stück im Vergleich zum Klavier sehr leise ist, heben sich die Gesangspassagen im Spektrum kaum hervor und werden durch die punktweise Gewichtung noch weiter reduziert, was zur Folge hat, dass bei diesem Stück einige Gesangspassagen nicht als solche erkannt werden.

Abschließend wurden drei Vergleichsexperimente durchgeführt, bei denen einzelne Aspekte der beiden untersuchten Verfahren gegeneinander ausgetauscht wurden. Im ersten Experiment wurde aus den maschinell ausgewählten 40 Merkmalskomponenten eine Selbstähnlichkeitsmatrix berechnet, welche anschließend mittels sparse-NMF zur binären Klassifikation nach Gesang bzw. Klavier verwendet wurde. Das erhaltene F-Measure von 0,880 ist höher als das mit den normalen MFCC-Merkmalen erzielte, kommt allerdings nicht ganz an die Güte der sparse-NMF-Klassifikation mittels der modifizierten MFCC-Merkmale heran. In einem zweiten Experiment wurde das maschinelle Lernverfahren ausschließlich auf den modifizierten MFCC-Merkmalen durchgeführt. Die dabei erzielten Einzelergebnisse wiesen eine hohe Übereinstimmung mit den Ergebnissen des regulären Verfahrens auf, allerdings traten einige Ausreißer nach unten auf. Folglich konnten die modifizierten Merkmale das maschinelle Lernverfahren nicht verbessern. Diese Tendenz bestätigte sich in einem dritten Experiment; diesmal wurde dem maschinellen Auswahlverfahren als Eingabewerte neben den auf normalen MFCCs beruhenden Merkmalen auch die entsprechenden, aus den modifizierten MFCCs abgeleiteten Merkmale zur Verfügung gestellt und überprüft, welche Merkmale von dem Verfahren als deskriptiv ausgewählt wurden. Hierbei zeigte sich, dass das Verfahren den Standard-MFCCs in weiten Teilen den Vorzug gegenüber den phasengewichteten MFCCs gab.

In diesen Experimenten zeigte sich, dass die untersuchte Methode maschinellen Lernens mit den allgemeineren MFCC-Merkmalen leicht bessere Ergebnisse erzielen konnte als mit den modifizierten Merkmalen. Bei Verwendung von sparse-NMF zur binären Klassifikation konnte die Deskriptivität der modifizierten Merkmale durch automatische Merkmalsauswahl nicht erreicht werden. Hier bestätigt sich die in [97] geäußerte Beobachtung insofern, als dass Modifikationen von MFCC-Merkmalen in Verbindung mit maschinellem Lernen zu keiner Verbesserung der erreichten Klassifikationsergebnisse führen.

4.5. Kombinierte Strukturanalyse

Bei der Vorstellung der Annotationen für diese Fallstudie haben wir Wiederholungen, Harmonik sowie die Unterscheidung zwischen Klavier- und Gesangspassagen verwendet. Mittels der

in diesem Kapitel diskutierten Merkmale sowie der in Kapitel 3 vorgestellten Methode können wir diese drei musikalischen Aspekte als Blockstrukturmatrix darstellen. Die in [182] vorgestellte Analyse der Deskriptivität spezieller Merkmale zur Beschreibung einzelner Segmente hat ergeben, dass innerhalb eines Stückes die verschiedenen Segmente mitunter durch verschiedene Merkmale ermittelt werden können. Bei dieser Betrachtung ist allerdings nicht klar, ob der solcherart ermittelte musikalische Aspekt tatsächlich der menschlichen Annotation zugrunde liegt oder ob diese Übereinstimmung durch einen Zufall zustande gekommen ist. Bei unserer Fallstudie liegt diese Information in Form der erläuternden Texte in Abschnitt A.2 vor. Somit können wir überprüfen, ob uns die Rekonstruktion der Referenzannotation durch automatisierte Verfahren nach denselben Entscheidungskriterien gelingt.

Im Folgenden analysieren wir das erste Lied »Gute Nacht« der Winterreise, dessen harmonische Eigenschaften bereits in Abschnitt 4.3.2 vorgestellt wurden. Das Stück besteht aus vier Strophen und fünf instrumentalen Zwischenspielen (die mit I bzw. $I2$ bezeichnet sind), wobei diese wie die ersten drei Strophen in Moll stehen. Die letzte Strophe steht abweichend in Dur. Die Strophen selbst sind dreigeteilt ($A\,B\,C$ bzw. $A2\,B\,C2$), wobei sich der mittlere Teil (B) harmonisch von den anderen Strophen unterscheidet; dieser steht in den Moll-Strophen in der parallelen Durtonart, in der Dur-Strophe in der Subdominanten.

In Abbildung 4.16 sind drei Strukturierungen dieses Stücks in der Aufnahme von Allen sowie eine für die Aufnahme von Fischer-Dieskau (begleitet von Moore) illustriert. Jeder Abbildungsteil besteht aus einer Selbstähnlichkeitsmatrix, der daraus mittels sparse-NMF abgeleiteten Segmentierung sowie der strukturellen Referenzannotation.

Für die in Abbildung 4.16a dargestellte Segmentierung wurden die in Abschnitt 4.4 vorgestellten Merkmale zur Ermittlung von Gesangspassagen verwendet. Der Vergleich der beiden Segmentierungen ergibt, dass erwartungsgemäß die instrumentalen Zwischenspiele erfolgreich von den restlichen Segmenten getrennt wurden. Da die Strophen ebenfalls kurze Gesangspausen aufweisen und da der Interpret zu Beginn der vierten Strophe sehr leise singt, wurden zusätzlich Bereiche aus den Strophen den Zwischenspielen zugeordnet. In der zweiten Segmentierung (Abbildungsteil b) wurden ausschließlich die in Abschnitt 4.3 vorgestellten funktionsharmonisch motivierten Key-Merkmale verwendet. Diese eignen sich in besonderem Maße zum Nachvollziehen der harmonischen Strukturierungselemente des Liedes – in diesem Beispiel betrifft das die mit B bezeichneten Mittelteile der Moll-Strophen sowie die Auszeichnung der Dur-Strophe. Sämtliche anderen Unterscheidungskriterien werden von diesem Merkmal nicht erfasst, auch der harmonisch nah verwandte Mittelteil der Dur-Strophe wird nicht erkannt.

Die in Abbildung 4.16c dargestellte Selbstähnlichkeitsmatrix stellt das punktweise Minimum der Matrizen aus den Teilen (a) und (b) dar. In dieser Matrix werden also zwei Passagen des Musikstücks nur dann als ähnlich ausgewiesen, wenn sie sowohl bezüglich Harmonik als auch bezüglich Instrumentierung übereinstimmen. Durch die in Abschnitt A.2 aufgeführte Erklärung zur manuellen Segmentierung wissen wir, dass damit die wesentlichen musikalischen Aspekte berücksichtigt werden. Die automatische Segmentierung zeigt folglich auch

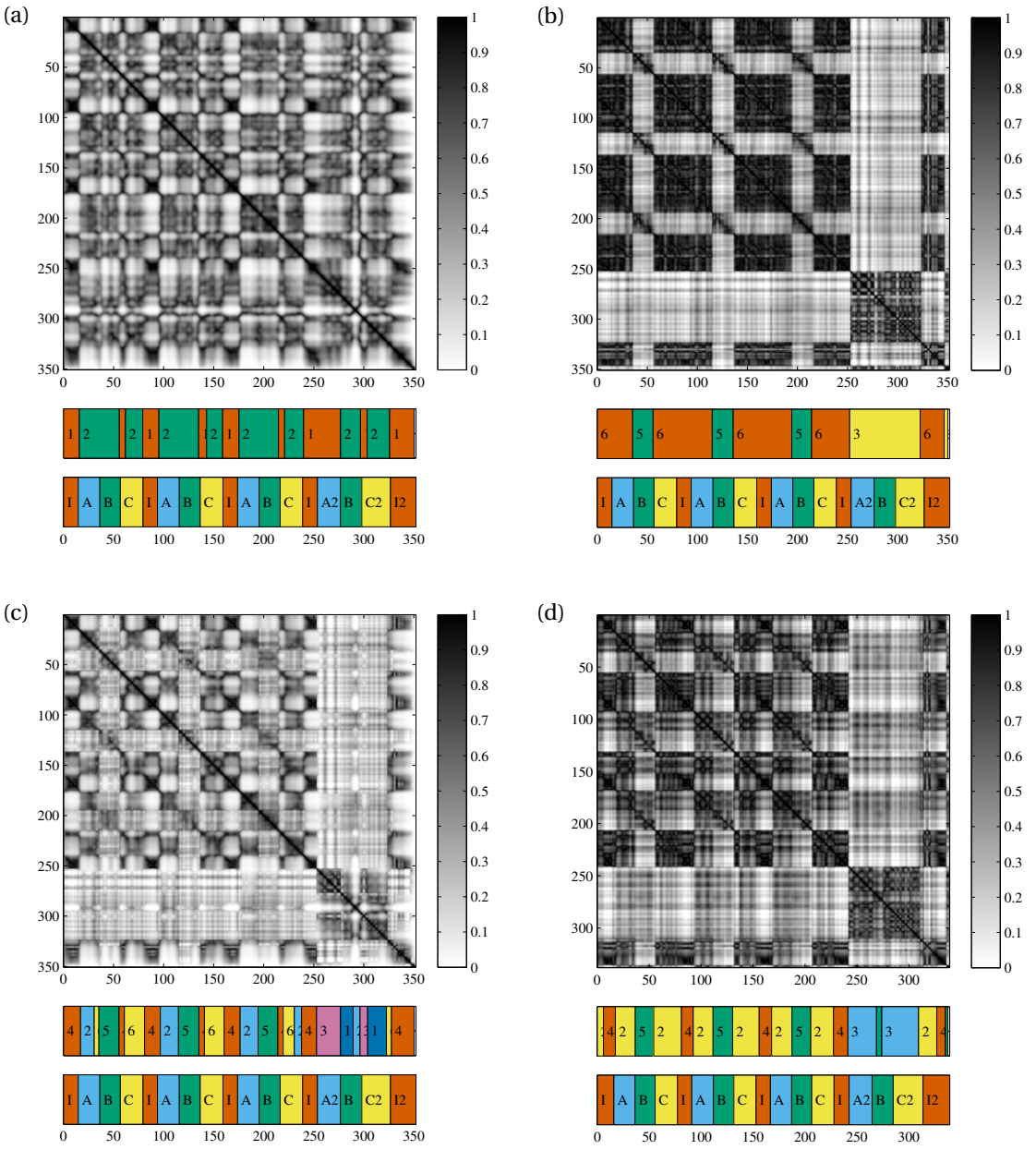


Abbildung 4.16.: Strukturanalyse durch Kombination zweier homogener Merkmale des ersten Stücks der Winterreise in der Aufnahme von Allen. **(a)** Klangfarbe (Gesangs-Merkmale), **(b)** Harmonik (Key-Merkmale), **(c)** Kombination der Selbstähnlichkeitsmatrizen für Harmonik und Klangfarbe. **(d)** Alleinige Verwendung von Key-Merkmalen bei der Aufnahme von Fischer-Dieskau und Moore.

eine hohe Übereinstimmung mit der Referenz, die Mollstrophen stimmen bis auf die kleinen Gesangspausen und je einem kleinen Artefakt in der ersten und der dritten Strophe fast vollständig überein. Die weder in (a) noch in (b) ermittelte Unterscheidung zwischen den mit *A* und *C* bezeichneten Strophenbestandteilen lassen sich durch leichte Abweichungen in der Klangfarbe erklären, die bei der binären Segmentierung in (a) nicht berücksichtigt wurden. Die Struktur der Durstrophe wurde weniger gut erkannt, was wir auf bereits bei Betrachtung der einzelnen Aspekte gemachte Fehler zurückführen. So wurde das Segment *A*2 ursprünglich nicht als Gesang erkannt, weswegen es hier mit dem kurzen Zwischenspiel zwischen *B* und *C*2 in eine Segmentklasse zusammengefasst wird. Die Segmente *B* und *C*2 selbst werden ebenfalls weitgehend als zur selben Segmentklasse zugehörig gezählt, da sich diese harmonisch zu wenig unterscheiden.

Der hohe Grad an Abhängigkeit dieser Betrachtungen von der konkreten Aufnahme zeigt die in Abbildung 4.16d dargestellte Interpretation von Fischer-Dieskau mit Moore. Die dargestellte Matrix basiert ausschließlich auf Key-Merkmalen, welche allerdings durch die starken Obertöne des Gesangs soweit gestört werden, dass sie – zwar nur in geringem Maße, aber ausreichend für das Segmentierungsverfahren – zur Unterscheidung zwischen den Instrumental- und Gesangspassagen innerhalb der Mollbereiche geeignet sind. Folglich erzielen wir hier durch diese »Artefakte« eine deutlich genauere Segmentierung als bei Abbildungsteil (b). Dies zeigt auch eine große Schwierigkeit bei der automatischen Evaluation eines Datensatzes auf. Gute numerischen Ergebnisse können leicht zu der Fehlannahme führen, dass ein Merkmal besonders gut zur Beschreibung eines allgemeinen Segmentierungsaspekts geeignet ist, obwohl manche Resultate nur auf geeignet gewählte Schwellwerte oder Glättungsparameter zurückzuführen sind. So mag man bei diesem Spezialfall zum Ergebnis kommen, dass die Key-Merkmale zur Beschreibung dieser Segmentierung nahezu ausreichend sind, obwohl dies nur auf dieses spezielle Stück unter der von uns gewählten Standardparametrisierung zutrifft.

Diese Beobachtung wollen wir durch ein weiteres Beispiel untermauern. Das in Abbildung 4.17 illustrierte 20. Lied »Der Wegweiser« besteht aus vier Strophen, von denen die letzte wiederholt wird. Die Segmente stimmen weitgehend mit den Strophen überein und stellen alle mehr oder weniger starke Variationen desselben musikalischen Materials dar, wodurch die Angabe einer möglichst allgemeingültigen Referenzannotation deutlich erschwert wird. Alle Segmente stehen in g-Moll mit Ausnahme von *B*, welches in G-Dur beginnt und in e-Moll fortgesetzt wird. Das zweite *A*-Segment könnte auch als leichte, *B* als starke Variation des ersten *A*-Segments angesehen werden. Auch die beiden *C*-Segmente sind rhythmisch und harmonisch Variationen von *A*, wobei hier die Melodik stark abweicht. Folglich wäre auch eine Segmentbenennung der Form *I A*1 *A*2 *A*3 *A*4 *A*5 möglich. Wir haben uns hier entschieden, die stärkeren Variationen durch eigene Buchstaben zu kennzeichnen.

Die Kombination der homogenen Merkmale Klangfarbe und lokale Tonarten, die beim ersten Stück zu einer weitgehenden Übereinstimmung der berechneten mit der annotierten Struktur führte, liefert auf diesem Stück in der Aufnahme von Fischer-Dieskau/Moore kein befriedigendes Ergebnis, siehe Abbildung 4.17a. Die berechnete Segmentierung stellt im wesentlichen eine untersegmentierte Variante der erkannten Gesangspassagen (Gesang wird durch die

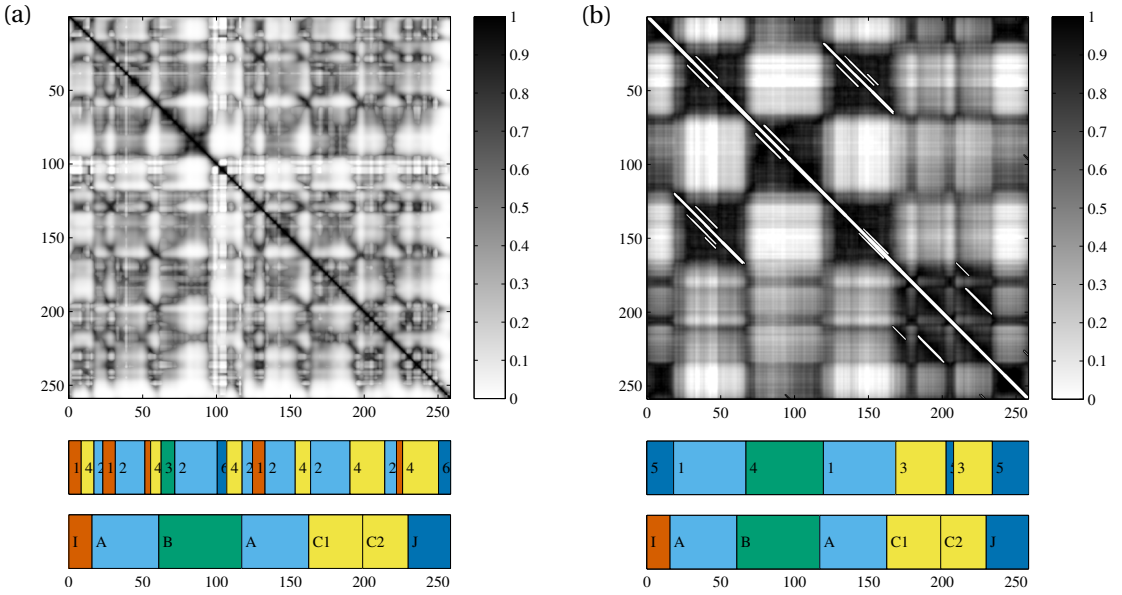


Abbildung 4.17.: Kombinierte Strukturanalyse des 20. Stücks der Winterreise in der Aufnahme von Fischer-Dieskau begleitet von Moore. **(a)** Kombination von Klangfarbe und Harmonik, **(b)** rein wiederholungsbasiert.

Segmente mit den Bezeichnungen 2 und 3 dargestellt, Klavier durch die übrigen Bezeichnungen) mit zusätzlicher Einbeziehung der harmonischen Unterteilung des B-Segments dar. Im Gegensatz dazu erhält man bei dieser Aufnahme mittels einer rein wiederholungsbasierten Segmentierung wie in Abbildungsteil (b) dargestellt ein nahezu mit der Referenz übereinstimmendes Resultat. Hierbei ist wiederum anzumerken, dass unsere manuelle Annotation nur eine von mehreren sinnvollen Möglichkeiten zur Beschreibung der musikalischen Struktur dieses Liedes darstellt, wodurch auch diese Aussage keine Allgemeingültigkeit beanspruchen darf.

Zusammenfassend kommen wir zu dem Ergebnis, dass eine Kombination verschiedener Selbstähnlichkeitsmatrizen nicht auf direktem Wege möglich ist. Dennoch zeigt das erste der diskutierten Beispiele, dass diese Herangehensweise über das Potential verfügt, ein besseres Resultat zu erzielen als es bei Verwendung der einzelnen Segmentierungen möglich wäre. Eine Möglichkeit für weitere Forschungen auf diesem Gebiet wäre das Einbeziehen von Vorwissen über die der manuellen Segmentierung zugrundeliegenden Eigenschaften. Diese könnten analog zu [182] auch lokale Eigenschaften abbilden und somit die nur an wenigen Stellen relevanten Merkmale mittels geeigneter Maskierungen auch nur an diesen Stellen berücksichtigen. Weiterhin wäre langfristig die Entwicklung einer Art Plausibilitätsüberprüfung für Segmentierungen denkbar, die aus jeweils einem Merkmal berechnet wurden. Diese würde eine Abschätzung erlauben, welches Merkmal im Einzelfall strukturell relevante Informationen liefert. Als ein erster Schritt für eine solche Herangehensweise erscheint es uns sinnvoll, eine

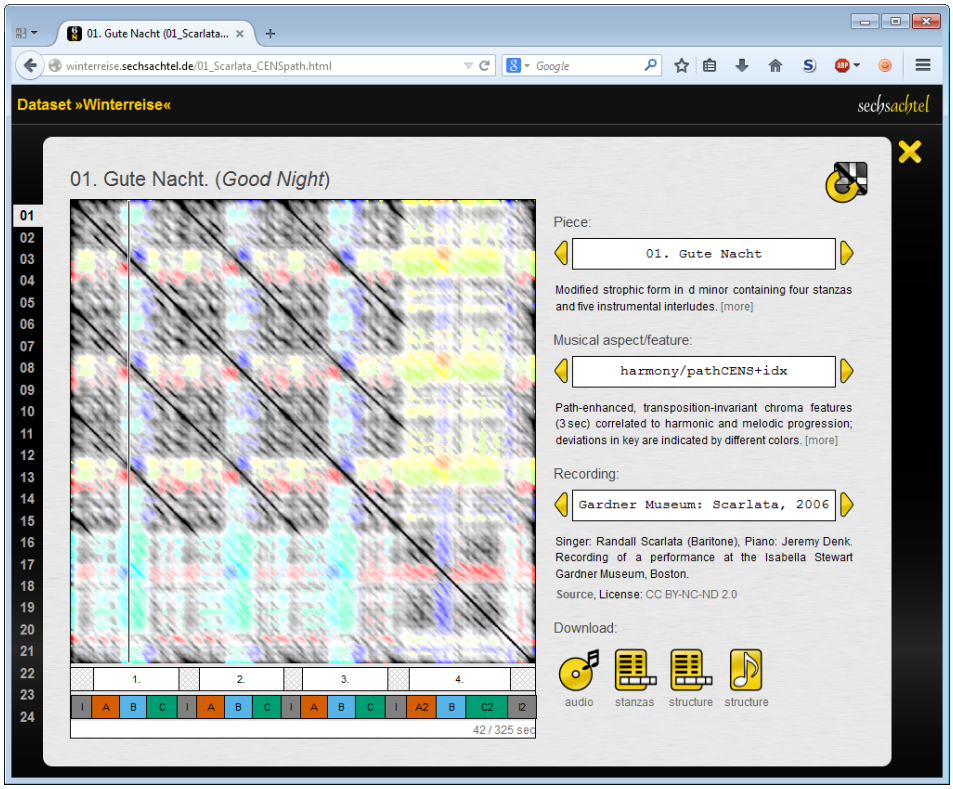


Abbildung 4.18.: Screenshot der Website `http://winterreise.sechsachtel.de` mit aktiver `makePlotPlayable`-Benutzerschnittstelle.

gemeinsame Betrachtung von Musik, Referenzannotation und der verschiedenen Selbstähnlichkeitsmatrizen zu ermöglichen.

Um einen intuitiven Zugang zu Merkmalsdarstellungen zu bekommen, haben wir in [122] eine bereits in [40] erfolgreich eingesetzte MATLAB-Funktion namens `makePlotPlayable` vorgestellt, die es dem Benutzer erlaubt, eine Audio-Datei synchron mit jeder denkbaren Merkmalsdarstellung oder Selbstähnlichkeitsmatrix abzuspielen. Diese Funktion wird nach Erstellung eines beliebigen Plots aufgerufen und erlaubt, durch einfachen Klick auf eine beliebige Position die Wiedergabe der Audio-Datei an der dazu korrespondierenden Stelle zu starten. Während der Wiedergabe zeigt die Funktion innerhalb der Abbildung ihre jeweilige Position durch eine bewegte vertikale Linie an. Diese einfache Funktionalität stellt eine große Hilfestellung bei der Analyse dar und trägt zum Verstehen der Eigenschaften einer Merkmalsdarstellung in einer musikalisch-intuitiven Weise bei.

Auf der begleitenden Website `http://winterreise.sechsachtel.de` zu diesem Kapitel kommt eine analoge Funktionalität mittels der neuen Möglichkeiten von *HTML5* in Verbindung mit *JavaScript* zur interaktiven Darstellung verschiedener Selbstähnlichkeitsmatrizen

sowie der Scape-Plot-Repräsentationen von zwei öffentlich verfügbaren Komplettaufnahmen der Winterreise zum Einsatz. Die Einzelseiten zu den jeweiligen Aufnahmen enthalten zudem auf gleicher zeitlicher Achse die Strukturannotationen sowie die Positionen der Gedichtstrophen, um somit einen direkten Zugang zur Verbindung des musikalischen Materials mit den Eigenschaften der verschiedenen Merkmalsdarstellungen und den zur Segmentierung herangezogenen musikalischen Aspekten zu eröffnen. In Abbildung 4.18 ist diese Ansicht für das erste Stück in der Aufnahme von Scarlata dargestellt. Bei der dort abgebildeten Merkmalsdarstellung handelt es sich um eine transpositionsinvariante, pfadgeglättete Chroma-Selbstähnlichkeitsmatrix.

Mittels der rechten Spalte kann auf Zusatzinformationen zur abgebildeten Matrix zugegriffen werden sowie zwischen den einzelnen Stücken, Merkmalsdarstellungen und Interpreten navigiert werden. Unter *Piece* finden sich die auch in Abschnitt A.2 aufgeführten Begleittexte zu den einzelnen Stücken. Der zweite Punkt *Musical aspect/feature* enthält Informationen zum Merkmal, aus dem die aktuelle Selbstähnlichkeitsmatrix berechnet worden ist, und ermöglicht das Blättern durch die alternativen Selbstähnlichkeitsmatrizen. Diese umfassen sowohl chromabasierte als auch die in Abschnitt 4.4 diskutierten Klangfarbenmerkmale sowie Tempo- und Rhythmusmerkmale. Weiterhin umfasst die Website die in Abschnitt 4.3 vorgestellten doppelten Scape-Plots. Darunter kann in der Rubrik *Recording* zwischen den verschiedenen Aufnahmen des Stückes gewechselt sowie die Lizenzinformationen für die jeweilige Aufnahme eingesehen werden. Abschließend können die jeweiligen Audio-Dateien, die Strophen- und Strukturannotationen und eine Notenansicht mit farblich hervorgehobenen Strukturinformationen heruntergeladen werden.

Wir sind davon überzeugt, dass in vielen Fällen keine eindeutige Lösung für das Problem der automatischen Strukturanalyse für Musiksignale gefunden werden kann, dass aber in der Beschäftigung mit dieser Frage und der Analyse des Zusammenspiels zwischen musikalischen und algorithmischen Aspekten wichtige Erkenntnisse über die Eigenschaften und Grenzen der Segmentierungsmethoden für Musikstücke gewonnen werden können. Mittels der in dieser Fallstudie vorgestellten Musikstücke, manuell erstellten Annotationen sowie mehrerer Merkmalsdarstellungen soll die Webseite die Vielschichtigkeit des Problems nicht nur darstellen, sondern auch intuitiv erfahrbar machen und somit der weiteren Forschung auf diesem Gebiet dienlich sein.

5. Ermittlung rhythmischer Informationen in MIDI-Dateien

Das MIDI-Format wurde ursprünglich für den Austausch von Kontrollsequenzen zwischen verschiedenen elektronischen Instrumenten entwickelt. Im Laufe der Zeit hat sich MIDI als Quasi-Standard für die Darstellung und Speicherung von partiturbezogenen Informationen etabliert. Die einer MIDI-Datei zugrundeliegenden Zeitinformationen können sowohl in musikalisch-symbolisch (wie bei einem Notenblatt) als auch in physikalisch-absoluter Form in Sekunden (um eine spezifische Interpretation abzubilden) vorliegen. Allerdings haben bei etlichen MIDI-Dateien die Zeitinformationen nur eine physikalische Bedeutung, wenn bei mangelndem kontextuellen Bezug die symbolischen Werte nur auf Standardwerte gesetzt sind, die in keiner Beziehung zum tatsächlichen musikalischen Inhalt stehen.

In diesem Kapitel[1] stellen wir eine Methode zur Bestimmung des musikalischen Schlagrasters einer solchen MIDI-Datei mit rein physikalischen Zeitinformationen vor. Ein Hauptbeitrag ist die globale Schätzung der Taktart, die wir zur Korrektur von lokalen Fehlern des vorher abgeschätzten Schlagrasters verwenden. Im Unterschied zur MIDI-Quantisierung, bei der MIDI-Noten an ein musikalisches Raster angeglichen werden, ist unser Ziel die Bestimmung eines solchen Rasters. In diesem Sinne kann unsere Methode in Kombination mit bereits existenter MIDI-Quantisierungssoftware verwendet werden, um physikalische MIDI-Dateien in semantisch angereicherte symbolische MIDI-Dateien umzuwandeln.

5.1. Einleitung

MIDI (*Music Instrument Digital Interface*, dt. Digitale Schnittstelle für Musikinstrumente) ist ein Standard-Protokoll zur Steuerung und Synchronisation elektronischer Instrumente und Synthesizer [74]. Obwohl MIDI ursprünglich nicht für symbolische Musikdaten konzipiert wurde und daher viele Einschränkungen in Bezug auf die Darstellung musikalisch relevanter Informationen aufweist [114, 171], verbreitete es sich aufgrund seiner relativen Flexibilität in den letzten drei Jahrzehnten. Die besondere Relevanz, die dieses Format für die Speicherung musikalischer Informationen trotz weiterentwickelter Formate wie *MusicXML* [106] auch noch heute hat, resultiert unter anderem sowohl aus der unüberschaubaren Menge frei im Internet verfügbarer Musikstücke im MIDI-Format als auch der MIDI-Unterstützung nahezu aller Programme zur elektronischen Verarbeitung symbolischer Musikdaten.

[1] Dieses Kapitel ist eine erweiterte Version von [63].

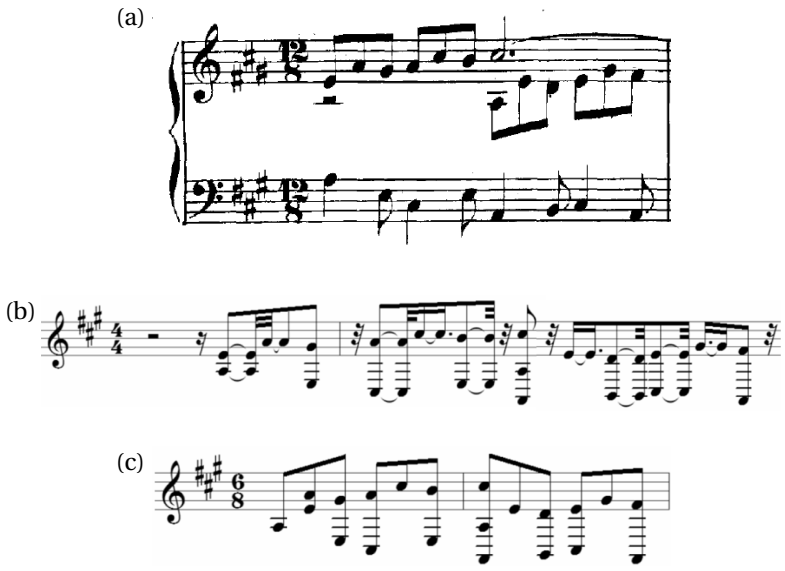


Abbildung 5.1.: Erster Takt des Präludiums BWV 888 von J. S. Bach. **(a)** Originalnoten. **(b)** Notendarstellung einer P-MIDI-Datei einer Live-Aufnahme ohne musikalische Zeitinformation. **(c)** Notendarstellung einer S-MIDI-Datei basierend auf einem geschätzten Schlagraster.

Eine besonderes Merkmal des MIDI-Formats ist es, sowohl musikalisch-symbolische als auch physikalische Einsatzzeiten und Notendauern bereit zu stellen. Insbesondere spezifiziert der Kopf einer MIDI-Datei die Anzahl der Zeiteinheiten (im MIDI-Kontext als *ticks* bezeichnet) pro Viertelnote. Die physikalische Zeitinformation wird dann durch zusätzliche Tempo-Befehle kodiert, welche ihrerseits die Anzahl der Mikrosekunden pro Viertelnote spezifizieren.

Einerseits ermöglicht das Entfernen dieser Tempo-Befehle die Erstellung einer mechanischen Version in konstantem Tempo, das stark mit der musikalischen Zeitachse (etwa in Viertelschlägen) eines Notenblattes korreliert ist. Andererseits erlaubt das Hinzufügen solcher Tempo-Informationen zu einer Partitur-generierten MIDI-Datei die Erstellung einer interpretierten Version mit einer sinnvollen physikalischen Zeitachse (in Sekunden). Allerdings folgen etliche der verfügbaren MIDI-Dateien dieser Konvention nicht. So werden zum Beispiel viele MIDI-Dateien durch direktes Einspielen auf einem MIDI-fähigen Instrument erzeugt, ohne dass das Tempo näher spezifiziert wird. Dies führt dazu, dass weder die Angabe zur Anzahl der Ticks pro Viertelnote noch die expliziten Tempo-Informationen musikalisch korrekt erfasst werden. Stattdessen werden diese Parameter auf Standardwerte gesetzt, die zwar die Rekonstruktion der physikalischen, nicht aber der musikalisch-symbolischen Zeitinformationen erlaubt.

Im Folgenden unterscheiden wir zwei Typen von MIDI-Dateien: Wenn die Taktschläge und Tempoinformationen auf musikalisch sinnvolle Art und Weise gesetzt sind, kann daraus neben der physikalischen auch eine musikalische Zeitachse analog zu einem (symbolischen) Notentext abgeleitet werden. In diesem Falle sprechen wir von einer MIDI-Datei *mit symbolischen*

Zeitinformationen, oder kurz von S-MIDI. Wenn im Gegensatz dazu die Tempoinformationen fehlen oder falsch gesetzt sind und die Schlagzeiten ohne musikalische Bedeutung sind, sprechen wir von einer MIDI-Datei *mit rein physikalischen Zeitinformationen*, oder kurz von P-MIDI. In diesem Kapitel behandeln wir das allgemeine Problem, wie eine P-MIDI-Datei in eine (sinnvolle Approximation einer) S-MIDI-Datei umgewandelt werden kann. Der Hauptschritt dieses Verfahrens liegt darin, ein auf musikalischen Informationen fundiertes Schlagraster abzuschätzen, aus dem die musikalische Zeitachse abgeleitet werden kann. Es ist zu beachten, dass die Extraktion solcher musikalischer Zeitinformationen aus MIDI-Dateien notwendig ist, bevor Programme zur *Quantisierung* und Notenblatt-Erstellung auf sinnvolle Art und Weise angewendet werden können.

Die in diesem Kapitel vorgestellte Methode schätzt Schlagraster und globale Tonart einer P-MIDI-Datei ab und wandelt diese anschließend mithilfe der ermittelten Informationen in eine S-MIDI-Datei um, wobei die physikalischen Zeitinformationen weitgehend erhalten bleiben. Dies ist in Abbildung 5.1 illustriert, welche neben dem originalen Notenmaterial zwei aus MIDI-Dateien generierte Partiturausschnitte zeigt. Der erste dieser Ausschnitte entspricht dem direkten Import der P-MIDI-Datei in ein Programm zur Musiknotation[2], für den zweiten Ausschnitt wurde diese Datei vorher automatisch in eine S-MIDI-Datei umgewandelt und anschließend in das Notationsprogramm importiert.

Das vorgestellte Verfahren basiert auf der Idee, einen ursprünglich für das Erkennen von rhythmischen Impulsen in Audio-Dateien entwickelten *beat tracker* für die Schätzung eines ersten Impulsrasters zu verwenden. Obwohl dieses Raster im Allgemeinen nicht fehlerfrei ist, nutzen wir es als erste Approximation des musikalischen Schlagrasters. Insbesondere genügen diese Informationen zur Ableitung der Taktart. Diese wiederum wird anschließend verwendet, um einzeln auftretende Unregelmäßigkeiten des Impulsrasters zu korrigieren.

Nach einem kurzen Überblick über die relevanten Spezifikationen des MIDI-Formats in Abschnitt 5.2 betrachten wir den aktuellen Stand der Forschung im Gebiet der Rhythmus-Erkennung und des Beat-Trackings (Abschnitt 5.3). In Abschnitt 5.4 beschreiben wir die algorithmischen Details der vorgestellten Methode. Im darauffolgenden Abschnitt 5.5 werten wir unser Verfahren aus und diskutieren mittels einiger expliziter Beispiele seine Vorzüge und Grenzen. Weiterhin gehen wir auf einige Erweiterungen ein (Abschnitt 5.6) und stellen in Abschnitt 5.7 eine konkrete Anwendung vor, bei der durch die vorgestellte Methode ein wesentlicher Fortschritt bei einer real auftretenden Problemstellung ermöglicht wird. Abschließend wird in Abschnitt 5.8 neben einer kurzen Zusammenfassung auch auf weitere Anwendungsmöglichkeiten eingegangen und ein Ausblick auf mögliche weitere Entwicklungen gegeben.

[2] In diesem Fall wurde das Notationsprogramm *capella* verwendet.

5.2. Das MIDI-Format

Zur Veranschaulichung des Unterschieds zwischen symbolischen und rein physikalischen Zeitinformationen ist ein generelles Verständnis der technischen Umsetzung und internen Speicherung musikalischer Inhalte im MIDI-Format nötig. Daher geben wir in diesem Abschnitt einen Überblick über den grundlegenden Aufbau eines Datenstroms von MIDI-Befehlen und wenden uns insbesondere der Darstellung von Notenwerten und Rhythmus zu. Durch den Vergleich zweier inhaltsgleicher S-MIDI- und P-MIDI-Dateien wird deutlich, warum die dem MIDI-Format zugrundeliegenden Zeitinformationen bei P-MIDI-Dateien nicht von sich aus geeignet sind, die rhythmischen Inhalte in eine symbolische Darstellung wie etwa eine Partitur zu überführen.

Die von verschiedenen Musikgeräte-Herstellern gebildete *MIDI Manufactors Association* beschreibt in der Einleitung der detaillierten Spezifikation [113] des MIDI 1.0-Formats die Zielsetzung und den Einfluss von MIDI wie folgt:

> »MIDI, the Musical Instrument Digital Interface, was established as a hardware and software specification which would make it possible to exchange information (musical notes, program changes, expression control, etc.) between different musical instruments or other devices such as sequencers, computers, lighting controllers, mixers, etc. This ability to transmit and receive data was originally conceived for live performances, although subsequent developments have had enormous impact in recording studios, audio and video production, and composition environments.«

Vor diesem Hintergrund werden einige Besonderheiten des MIDI-Formats deutlich. Beispielsweise wird eine Note in MIDI durch zwei unabhängige Datenpakete realisiert, von denen das erste einen Ton einer speziellen Tonhöhe (engl. *pitch*) mit speziellen Parametern startet (entspricht dem Drücken einer Taste auf einem elektronischen Klavier) und ein zweiter Befehl das Loslassen dieser Taste modelliert und damit das Erklingen des Tons beendet. Diese Art der Modellierung musikalischer Information resultiert unmittelbar aus der Echtzeit-Übertragung von Spielanweisungen auf einem Musikinstrument an einen separaten Tonerzeuger. Diese Trennung zwischen Eingabegerät und Klangerzeuger ermöglicht die große Variabilität bezüglich einsetzbarer Geräte ohne zusätzliche Anforderungen an die einzelnen Geräte zu stellen, was bei einer Modellierung von symbolischen Notenobjekten deutlich komplexer werden würde.

Die MIDI-Inhalte werden im Binärformat realisiert und sind als solche nicht für leichte menschliche Lesbarkeit optimiert. Geläufig ist die Darstellung der einzelnen Bytes als zweistellige Hexadezimalzahl. Um diese besser von den üblichen Dezimalzahlen unterscheiden zu können, verwenden wir eine Schreibmaschinenschrift: Die Zahl `10` ist als Hexadezimalzahl zu lesen und entspricht der Dezimalzahl 16. Für einige MIDI-Inhalte werden auch englische Begriffe verwendet, diese werden in Großbuchstaben geschrieben: `MIDI_COMMAND`.

00	C0	00	00	FF	51	03	0B
71	B0	00	90	3C	40	9E	00
80	3C	00	00	FF	2F	00	

Abbildung 5.2.: Ausschnitt aus einer MIDI-Datei in hexadezimaler Darstellung mit farblicher Hervorhebung von Statusbytes (grau), Datenbytes (weiß), Zeitinformationen (blau) und Kommandos mit variabler Anzahl (dunkelgelb) von Datenbytes (hellgelb). Diese Sequenz kodiert das Abspielen einer ganze Note auf dem Klavier, vgl. Codebeispiel 5.1.

5.2.1. Ereignisse und Befehle

Im Folgenden geben wir eine kurze Übersicht über den Aufbau solcher MIDI-Datenpakete. Ein solches Paket wird auch als MIDI-Ereignis (engl. *event*) bezeichnet und besteht aus einer Zeitinformation und einem MIDI-Befehl (*message*). Neben den MIDI-Befehlen im engeren Sinne wie `NOTE_ON` und `NOTE_OFF` existieren weiterhin die Gruppe der Meta-Befehle wie Tempoangaben und die der systemspezifischen Befehle (*system exclusive*) [171]. Letztere unterscheiden sich oftmals von Hersteller zu Hersteller und sind für uns nicht relevant. Die MIDI-Befehle ihrerseits sind kurze Byte-Sequenzen, bestehend aus einem Statusbyte und mehreren Datenbytes. Die Anzahl der Datenbytes pro Befehl ist bei vielen Befehlen auf zwei festgelegt, kann aber abhängig vom Typ des Befehls auch variabel sein. So weist der MIDI-Befehl `PROGRAM_CHANGE` zum Ändern des Abspielprogramms bzw. Instruments nur ein Datenbyte zum Speichern der Programmnummer auf.

Status- und Datenbytes lassen sich grundsätzlich dadurch unterscheiden, dass das erste Bit der Statusbytes auf `1`, das der Datenbytes auf `0` gesetzt wird. In der üblichen Notation der MIDI-Befehle als hexadezimale Bytes liegen somit die Statusbytes im Bereich zwischen `80` und `FF`, während die Datenbytes im Bereich `00` bis `7F` zu finden sind. Beispielsweise kodiert der Befehl `90 3C 40` die Operation `NOTE_ON` mit den zwei Datenbytes 60 und 64, was dem Beginn eines Tons auf c' bei mittlerer Anschlagsstärke beschreibt; für eine Illustration siehe Abbildung 5.2. Die MIDI-Kommandos mit variablen Längen werden im nächsten Abschnitt erläutert.

Das MIDI-Format nutzt zur Adressierung 16 Kanäle (*channels*), die grob den verschiedenen Instrumenten oder auch den Notenzeilen einer Partitur entsprechen. Bei den nur einzelne Kanäle betreffenden Befehlen (*Channel voice messages*) wie etwa `NOTE_OFF 8x`, `NOTE_ON 9x` und `PROGRAM_CHANGE Cx` ist die Art des Befehls im ersten Halbbyte kodiert, die zweite Hälfte kodiert den betreffenden Kanal. Der Befehl `NOTE_ON` wird also durch `90` für den ersten bis hin zu `9F` für den sechzehnten Kanal kodiert.

Sollen nun mehrere MIDI-Geräte synchron betrieben werden, ist die Definition gemeinsamer Zeitpunkte wichtig. Diese Rolle kann bei einem MIDI-Datenstrom von der sogenannten *MIDI Beat Clock* bzw. *MIDI Time Clock* übernommen werden, wobei die *Beat Clock* eine relative Zeitangabe vorgibt (vergleichbar mit dem von einer Perkussionsgruppe vorgegebenen rhyth-

mischen Grundschlag) und die *Time Clock* absolute Zeitangaben bereitstellt (wie sie etwa für die Anforderungen der Filmindustrie wichtig sind). Technisch wird hier zu definierten Zeitpunkten ein spezieller MIDI-Befehl zur Synchronisation der angeschlossenen MIDI-Geräte gesendet. Diese Zeitangaben werden mit den in den MIDI-Ereignissen übermittelten Zeitinformationen in Beziehung gebracht, um beispielsweise synchrone Vibrati und Arpeggien zu ermöglichen. Im MIDI-Standard [113] wird weiterhin ein zusätzlicher Indikator für Taktanfänge `BAR_MARKER` definiert, welcher allerdings in den verbreiteten MIDI-Dateien scheinbar keine Beachtung gefunden hat.

Die absoluten Zeitangaben erfolgen in speziell definierten Perioden von 24 Hz, 25 Hz, 29,97 Hz bzw. 30 Hz, was mit den Bildwiederholraten der unterschiedlichen Filmformate (Film, PAL, NTSC mit *drop frame* bzw. NTSC ohne *drop frame*) zusammenhängt. Diese Formate werden nach der *Society of Motion Picture and Television Engineers* mit SMPTE bezeichnet. Die relativen Zeitangaben richten sich nach dem Grundschlag (modelliert als Viertelnote), welcher in 24 Teile (*ticks*) unterteilt wird. Dieses Format wird daher mit PPQ (*pulses per quarter*) bezeichnet; die physikalische Dauer dieser Ticks wird mittels eines speziellen Meta-Events spezifiziert, welches die Dauer einer Viertelnote in Mikrosekunden vorgibt. Für die Modellierung symbolisch-rhythmischer Informationen ist nur das PPQ-Zeitformat interessant.

5.2.2. Standard-Dateiformat

Schon bald nach Aufkommen der ersten MIDI-Geräte wurde im Jahr 1988 auch ein Format für das Abspeichern von MIDI-Informationen in speziellen Dateien spezifiziert [112]. Diese Dateien bestehen aus einen Kopf (engl. *header*), in dem einige globale Parameter festgelegt werden, und einer oder mehrerer sogenannter Spuren (engl. *tracks*), die Sequenzen von MIDI-Ereignissen beinhalten. Im Gegensatz zur Echtzeit-Wiedergabe eines dynamisch generierten MIDI-Befehlsstroms sind bei einer statischen Datei die Zeitinformationen essentiell zur Rekonstruktion der darin enthaltenen musikalischen Informationen.

Diese Zeitangaben werden in einer MIDI-Datei in Form von *delta time*-Werten angegeben, d. h. als Zeitdifferenz zwischen dem aktuellen und dem nächsten Ereignis. Ähnlich zu den bereits vorgestellten MIDI-Befehlen, bei denen mittels der Belegung des ersten Bits zwischen Status- und Datenbytes unterschieden wird, zeigt bei der Kodierung dieser Zeitinformationen das erste Bit jedes Bytes an, ob noch weitere Bytes zur Beschreibung der Zeitangabe folgen oder ob das letzte Byte erreicht wurde. Somit werden variable Bytelängen für die Zeitangaben ermöglicht (im Standard wird sie auf eine Maximallänge von 4 Byte beschränkt). Um eine eindeutige Trennung zwischen Zeitinformationen und MIDI-Befehlen zu erhalten, ist dieses Bit bei allen Bytes außer dem letzten auf 1 gesetzt. Zum Beispiel entspricht eine *delta time*-Angabe mit dem Wert `9e 00` der Binärsequenz `(1)0011110 (0)0000000`, wobei die geklammerten Werte die Statusbits markieren, die nicht bei der Berechnung der Zeitangabe berücksichtigt werden. Hier wird also die Binärzahl `1111 00000000` kodiert, was der Dezimalzahl 3840 entspricht.

In MIDI-Dateien wird mit dem Statusbyte `FF` ein Meta-Befehl eingeleitet, dessen genauer

```
1   4d 54 68 64          "MThd"
2   00 00 00 06          chunk size: "6" (the following header contains 6 bytes data)
3   00 00                format type: "0" (single track MIDI file)
4   00 01                number of tracks: "1"
5   03 c0                time division: "960" (top bit "0" means 960 PPQ)
6
7   4d 54 72 6b          "MTrk"
8   00 00 00 17          chunk size: "23" (the following track contains 23 bytes data)
9   00                   delta time: "0" (immediately)
10  c0 00                PROGRAM_CHANGE, channel "0", program "0"
11  00                   delta time: "0" (immediately)
12  ff 51 03 0b 71 b0 SET_TEMPO, length "3", data "750000" (microseconds per beat)
13  00                   delta time: "0" (immediately)
14  90 3c 40             NOTE_ON, channel "0", pitch "60" (C4), velocity "64"
15  9e 00                delta time: 3840 (four quarters of 960 ticks each)
16  80 3c 00             NOTE_OFF, channel "0", pitch "60" (C4), velocity "0"
17  00                   delta time: "0" (immediately)
18  ff 2f 00             END_OF_TRACK, length "0".
```

Codebeispiel 5.1: Kommentiertes Minimalbeispiel einer MIDI-Datei. Die Zeilen 9 bis 18 entsprechen der in Abbildung 5.2 gezeigten Kommandosequenz.

Typ in seinem ersten Datenbyte spezifiziert wird. Da die Meta-Befehle zur Kodierung vieler verschiedener Informationen verwendet werden, ist die Anzahl der Datenbytes variabel. Im zweiten Datenbyte wird die Anzahl der darauffolgenden weiteren Datenbytes, welche die eigentliche Information transportieren, explizit angegeben [112], vgl. Abbildung 5.2. Nicht alle Meta-Befehle werden von jedem MIDI-Gerät tatsächlich umgesetzt, da viele dieser Befehle für die Wiedergabe nicht notwendige Informationen wie Ton- und Taktart, Liedtexte oder Angaben zum Urheber enthalten.

Zur Kodierung rhythmisch-musikalischer Informationen sind die folgenden beiden Meta-Befehle von besonderer Bedeutung:

- Der `SET_TEMPO`-Befehl spezifiziert des aktuelle Tempo durch `FF 51 03 tttttt`, wobei die 6-stellige Hexadezimalzahl `tttttt` die Anzahl der Mikrosekunden pro Viertelnote angibt.
- Der `TIME_SIGNATURE`-Befehl legt die Taktart in der Form `FF 58 04 nn dd cc bb` fest, wobei das Byte `nn` den Zähler und `dd` die negative Zweierpotenz des Nenners angibt (also 2 für Viertel-, 3 für Achtelnoten etc.). Der Parameter `cc` kann zur Spezifikation eines Metronomschlags verwendet werden (Anzahl MIDI-Clock-Ereignisse pro Schlag) und `bb` zum Überschreiben der Definition einer MIDI-Viertelnote durch Angabe einer alternativen Anzahl an Zweiunddreißigstelnoten pro »Viertelnote«. Die letzten beiden Parameter sind für die symbolische Weiterverarbeitung der MIDI-Informationen nicht relevant.

In Codebeispiel 5.1 wird ein kommentiertes Beispiel einer einfachen MIDI-Datei vorgestellt,

die lediglich eine ganzen Note c' beschreibt. Die ersten 5 Zeilen sind der Kopf, in dem das Format der MIDI-Datei festgelegt (das Format »0« unterstützt nur eine Spur pro Datei) und die Anzahl von 960 Ticks pro Viertelschlag vorgegeben wird. Dass dies eine PPQ-Zeiteinheit ist, wird durch Setzen des führenden Bits auf 0 eindeutig festgelegt, ein Wert von 1 an dieser Position wird zum Kodieren der SMPTE-Zeitparameter verwendet. Die darauffolgende Spur enthält nach der einführenden Zeichenkette »MTrk« und der Längenangabe eine Folge von MIDI-Ereignissen, die jeweils aus einer *delta time*-Angabe und einem MIDI-Befehl bestehen. Wir finden etwa in Zeile 12 die Tempoangabe 750 000 als Länge eines Schlags in Mikrosekunden, was einem Tempo von 80 BPM (*beats per minute*) entspricht. In Zeile 14 wird das Abspielen des Tons c' gestartet und nach 3840 Ticks (Zeile 15) wieder beendet (Zeile 16). Da zwischen Beginn und Ende des gespielten Tons die Zeit von vier Viertelschlägen vergangen ist, beschreiben diese beiden MIDI-Ereignisse eine ganze Note. Durch die Tempoinformation ist festgelegt, dass ein MIDI-Abspielgerät zum Abspielen dieser Note genau drei Sekunden benötigt.

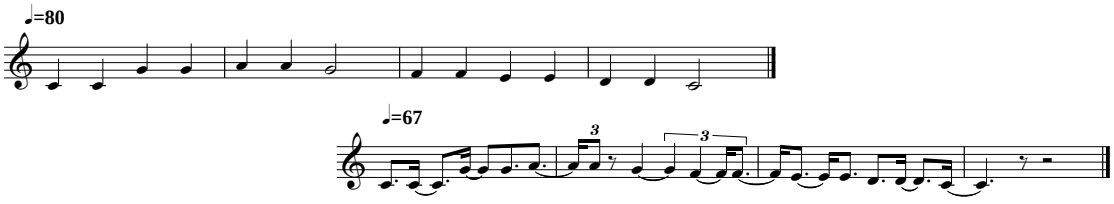
5.2.3. Musikalische und physikalische Zeitachse

In diesem Abschnitt stellen wir einen Bezug zwischen der musikalischen Zeitachse von MIDI-Dateien zu den physikalischen Abspielzeiten der einzelnen MIDI-Ereignisse her und vertiefen damit die bereits eingeführten Begriffe der symbolischen S-MIDI- und der physikalischen P-MIDI-Datei.

Die Zeitinformationen einer MIDI-Datei mit SMPTE-Zeitangaben sind immer physikalisch, d. h. in Sekunden bzw. bei Filmen in Einzelbildern, welche ebenfalls linear mit einer physikalischen Zeitachse zusammenhängen.

In dem für uns interessanteren Fall einer MIDI-Datei mit zugrundeliegendem PPQ-Zeitformat beziehen sich die MIDI-Ticks als zeitliche Minimalschritte immer auf eine gedachte Viertelnote. Im vorherigen Abschnitt wurde bereits die Zerlegung einer Viertelnote in Ticks vorgestellt. Durch Setzen der Tempoinformationen mittels eines oder mehrerer SET_TEMPO-Ereignisse wird ein Bezug zwischen Viertelnoten (und damit Ticks) und einer physikalischen Zeitachse (bei MIDI grundsätzlich in Mikrosekunden) hergestellt. Ist diese Information nicht gegeben, so wird nach dem Standard für MIDI-Dateien ein Wert von 500 000 μs pro Viertelnote (120 BPM) angenommen [112]. Diese Definition setzt voraus, dass die *delta time*-Zeitangaben zwischen den einzelnen MIDI-Befehlen im Wesentlichen mit dem im Kopf vordefinierten Raster übereinstimmen. Dies ist beispielsweise grundsätzlich beim Export aus Notensatzprogrammen der Fall, kann aber auch beim Einspielen eines Stückes durch metronomartige Vorgabe des Grundschlags durch das MIDI-Aufzeichnungsgerät erfolgen. Die so entstandenen S-MIDI-Dateien basieren auf dieser musikalisch-symbolischen Zeitachse, welche ihrerseits eine direkte Umrechnung in physikalische Einsatzzeiten erlaubt.

Umgekehrt gilt dies nicht: Stimmt das Raster der *delta time*-Angaben nicht mit der PPQ-Angabe des MIDI-Kopfes überein oder liegt – wie etwa beim Mitschnitt einer rhythmisch




```
1   4d 54 68 64       "MThd"
2   00 00 00 06       "6" header length
3   00 00             "0" format type
4   00 01             "1" number of tracks
5   00 18             "24" time division
6
7   4d 54 72 6b       "MTrk"
8   00 00 00 7b       "123" track length
9   00                delta time, dt
10  ff 51 03 0b 71 b0 SET_TEMPO 750000
11                                (80 BPM)
12  (dt  message)
13  00   90 3c 40     NOTE_ON   c
14  17   80 3c 00     NOTE_OFF
15  01   90 3c 40     NOTE_ON   c
16  17   80 3c 00     NOTE_OFF
17  01   90 43 40     NOTE_ON   g
18  17   80 43 00     NOTE_OFF
19  (...)
20  01   90 3c 40     NOTE_ON   c
21  30   80 3c 00     NOTE_OFF
22  00   ff 2f 00     END_OF_TRACK
```

```
1   4d 54 68 64       "MThd"
2   00 00 00 06       "6" header length
3   00 00             "0" format type
4   00 01             "1" number of tracks
5   00 18             "24" time division
6
7   4d 54 72 6b       "MTrk"
8   00 00 00 7b       "123" track length
9   00                delta time, dt
10  ff 51 03 0d bb a0 SET_TEMPO 900000
11                                (66.7 BPM)
12  (dt  message)
13  00   90 3c 40     NOTE_ON   c
14  13   80 3c 00     NOTE_OFF
15  01   90 3c 40     NOTE_ON   c
16  13   80 3c 00     NOTE_OFF
17  01   90 43 40     NOTE_ON   g
18  13   80 43 00     NOTE_OFF
19  (...)
20  01   90 3c 40     NOTE_ON   c
21  2a   80 3c 00     NOTE_OFF
22  00   ff 2f 00     END_OF_TRACK
```

Codebeispiel 5.2: Zwei MIDI-Dateien mit denselben Noten auf gleicher physikalischer Zeitachse, deren musikalische Zeitachsen voneinander abweichen, erkennbar an den unterschiedlichen *delta time*-Angaben von 17 (links) und 13 (rechts) vor den NOTE_OFF-Befehlen. Bei der linken Datei handelt es sich um eine S-MIDI, bei der rechten um eine P-MIDI.

freien Interpretation – den MIDI-Ereignissen kein starres Raster zugrunde, so ist die Extraktion rhythmischer Informationen nicht direkt möglich. Insbesondere werden Änderungen des musikalischen Tempos nicht durch entsprechende SET_TEMPO-Ereignisse ausgedrückt, sondern in einer nicht explizit dargestellten Veränderung des *delta time*-Rasters.

In Codebeispiel 5.2 sieht man die Notendarstellungen und einen Teil der MIDI-Daten für zwei MIDI-Dateien, die sich lediglich im Raster ihrer musikalischen Zeitachsen unterscheiden. Die Köpfe der beiden Dateien sind identisch, insbesondere werden stets 24 Ticks (hexadezimal 18) auf eine Viertelnote gezählt. Bei der linken Datei stimmt dies mit dem Abstand der NOTE_ON-Befehle überein (jeweils 23 Ticks Abstand zum dazugehörigen NOTE_OFF, dann einen weiteren bis zum nächsten NOTE_ON), wodurch diese MIDI-Datei von einem Notensatzprogramm[3] musikalisch sinnvoll dargestellt werden kann (oberes Notenbeispiel). Bei der rechten Datei beträgt dieser Abstand nur 20 Ticks (hexadezimal 14), was nicht mit der Auflösung im MIDI-

[3] Für dieses Beispiel wurde das Programm *Sibelius* verwendet.

Kopf übereinstimmt. Trotz exakt äquidistantem Abstand der Viertelnoten handelt es sich hier um eine P-MIDI-Datei, da wegen dieser Diskrepanz die semantische Bedeutung der Notendauern nicht mehr erkennbar ist. Dies hat zur Folge, dass das Notensatzprogramm alle Viertelnoten nur mit $\frac{5}{6}$ ihrer Länge darstellt, also als 5 triolisch unterteilte Sechzehntelnoten. Durch Quantisierungseffekte des Notensatzprogramms ergibt sich das untere Notenbeispiel. Da das Tempo des rechten Beispiels genau $\frac{5}{6}$ des Tempos der linken Datei entspricht, ist die physikalische Zeitachse nahezu identisch (durch Verwendung des konstanten Abstands von einem Tick zwischen je zwei Einzelnoten ergibt sich eine kleine Abweichung bei den Notenenden von 5,2 ms).

5.3. Stand der Forschung

Das grundsätzliche Problem, aus Darstellungen von Musik (dies umfasst sowohl MIDI als auch Audio-Aufnahmen) rhythmische Informationen zu extrahieren, ist eine komplexe Aufgabenstellung. Eine zusätzliche Schwierigkeit liegt darin, dass Art und Weise der zu ermittelnden rhythmischen Inhalte oftmals nicht klar definiert sind.

Die folgende Beschreibung des Begriffs *Metrum* aus *The New Grove Dictionary of Music and Musicians* unterstreicht die subjektive Komponente der zeitlichen Untergliederung eines Musikstücks in rhythmische Strukturen:

> »[Metre is] the temporal hierarchy of subdivisions, beats and bars that is maintained by performers and inferred by listeners which functions as a dynamic temporal framework for the production and comprehension of musical durations. In this sense, metre is more an aspect of the behaviour of performers and listeners than an aspect of the music itself.« [102]

Auch im Standardwerk »Musik in Geschichte und Gegenwart« findet man die Aussage »Rhythmus ist ein ästhetisches Phänomen« [170]. Daher müssen wir davon ausgehen, dass es kein automatisches Rhythmusanalyse-Verfahren gibt, welches eine adäquate Partiturdarstellung aus jeder beliebigen MIDI-Datei generieren kann. Dennoch können einige rhythmische Grundkonzepte wie Grundschläge und Takte in vielen Fällen in hinreichend guter Qualität gefunden werden.

Man beachte, dass die in diesem Kapitel definierten und verwendeten rhythmischen Begriffe im musikwissenschaftlichen Umfeld nicht im Sinne allgemein gültiger Definitionen gebraucht werden, sondern je nach Kontext und Lehrmeinung leicht abweichende Bedeutungen aufweisen, vgl. [170]. Dort wird u. a. der Ursprung des Rhythmischen in menschlichen Bewegungen hervorgehoben und die Parallelen zwischen musikalischem Pulsieren und menschlichen Schritten betont, der musikalische Puls dient somit als Bewegungsimpuls.

Wir unterscheiden im Folgenden zwischen diesem physikalischen Puls, dessen Hochpunkte wir als *Impulse* (engl. *pulses*) bezeichnen werden und dem daraus abgeleiteten musikalischen *Schlag* bzw. *Grundschlag* (*beat*). Diese Grundschläge bilden ein Raster (*beat grid*), welches wie-

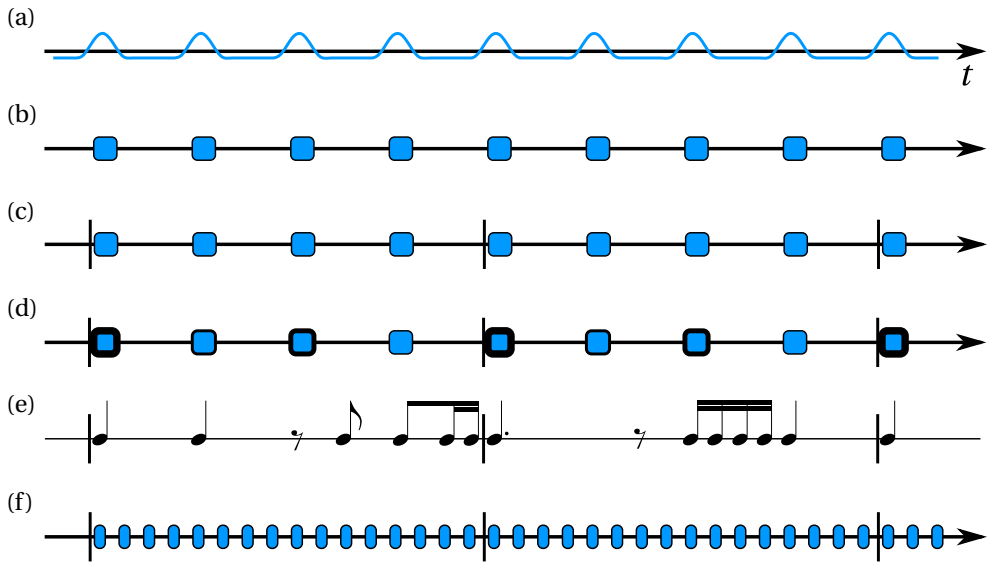


Abbildung 5.3.: Zusammenhang zwischen rhythmischen Begriffen, nach [205]: **(a)** Impulskurve, **(b)** Raster aus Grundschlägen, **(c)** Gruppierung zu Takten, **(d)** durch Betonungswerte gegebenes Metrum, **(e)** konkreter Rhythmus eines Stückes, **(f)** daraus abgeleitetes Tatum-Raster.

derum periodisch wiederkehrende Abfolgen verschiedener Gewichtungen aufweist, vgl. [43, Takt]. Die Folge der Hauptgewichte definiert hierbei die Anfänge der *Takte* (*measures* oder *bars*), das Betonungsmuster innerhalb der Takte definiert das Metrum (*meter*). Das Metrum ist also die periodische Komponente der Musik, die dem konkreten Rhythmus (*rhythm*) des jeweiligen Musikstückes zugrunde liegt[4] [170]. Mit dem Begriff *Tatum* [8] wird die kleinste rhythmisch sinnvolle Unterteilung des Schlagrasters bezeichnet. Im Allgemeinen wird diese global gewählt (wie in Notensatzprogrammen bei der Quantisierung auf eine Mindest-Notenlänge), wir verwenden in Abschnitt 5.4.3 eine lokale Variante für einen adaptiven Quantisierungsansatz. In Abbildung 5.3 wird der Zusammenhang zwischen den hier definierten und in den folgenden Abschnitten verwendeten Begriffen illustriert. Bei der Vorstellung von Ansätzen zur automatischen Extraktion rhythmischer Inhalte sind die oben erwähnten Mehrdeutigkeiten der Bezeichnungen zu beachten.

In [61] wird zwischen Programmen zur Herleitung von Grundschlag und Tempo, zur Generierung vollständiger rhythmischer Transkriptionen und zur Bestimmung von rhythmischen Merkmalen wie Tempo- oder Taktänderungen unterschieden. Allen gemein ist jedoch, dass entweder aus Audiodaten oder direkt aus symbolischen Daten Merkmale wie Aktivitätskurven generiert werden. Selbst wenn die Einsatzzeiten der einzelnen Noten explizit bekannt sind

[4] Diese Charakterisierung des Begriffspaars von Rhythmus und Metrum ist in dieser Form nur für die Musik ab der Musikepoche der Klassik (ab Mitte/Ende des 18. Jahrhunderts) zutreffend.

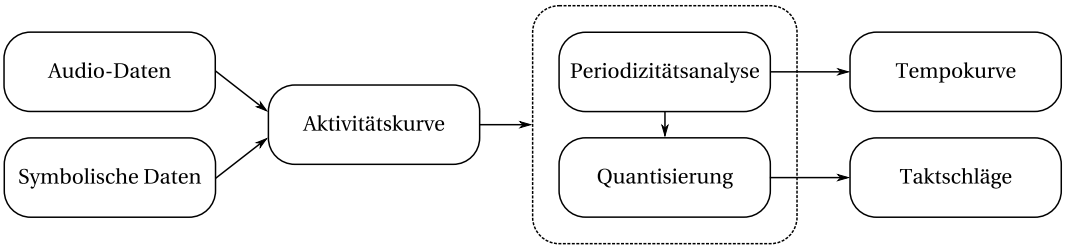


Abbildung 5.4.: Schematische Darstellung der Funktionsweise von *beat tracking*-Ansätzen zur Ermittlung von Grundschlag und Tempo, aufbauend auf [30].

(wie es bei MIDI-Dateien der Fall ist), ist das Ermitteln von Schlagzeiten und Takten bei weitem nicht trivial – insbesondere, wenn man MIDI-Aufnahmen von konkreten musikalischen Darbietungen mit dynamischen Tempoänderungen analysiert.

Ebenfalls ist diese spezielle Aufgabenstellung ein Teil des allgemeineren Problems der automatischen Musik-Transkription, also der Überführung einer Audiodatei in eine geeignete Partiturdarstellung. Dies kann in zwei Schritten geschehen, wobei im ersten die vorkommenden Töne geschätzt werden und im zweiten die so gewonnene Tonhöhen-Zeit-Darstellung in Notentext übersetzt wird [7]. Die Rhythmus-Transkription ist hier neben der Stimmentrennung ein wesentlicher Bestandteil des zweiten Schrittes.

Frühere Arbeiten wie [13, 15, 162, 189, 191] beschränken sich meist auf symbolische Musikdaten im MIDI-Format. Durch Vorschalten eines *onset detectors* zur Schätzung von Notenanfängen sind sie allerdings leicht auf die Verwendung von Audio-Daten erweiterbar [85]. In diesen Arbeiten wird bereits der hierarchische Aspekt des Metrums hervorgehoben. So wird in [191] ein Verfahren zur Erkennung und Schätzung der metrischen Hierarchie von MIDI-Dateien vorgestellt, es werden also nicht nur der Grundschlag, sondern auch übergeordnete rhythmische Konzepte wie Takte oder Phrasen gefunden. Dieses Verfahren basiert auf einem Ausgleich dreier »Regeln« für die Rhythmuserkennung: Korrelation von Taktschlägen zu Noteneinsatzzeitpunkten, Zuordnung betonter Taktschläge zu längeren Notenwerten und einem möglichst konstanten Abstand der Schläge je Hierarchiestufe. Weiterhin werden zusätzlich harmonische Informationen miteinbezogen. Ein ähnlicher Ansatz wird in [13] zur Herleitung eines rhythmisch sinnvollen Notentextes aus gegebenen MIDI-Daten vorgestellt. Hierbei werden charakteristische Abfolgen betonter und unbetonter Taktschläge zur Schätzung der Position der einzelnen Noten innerhalb eines Taktes verwendet.

In [58] wird ein regelbasiertes System vorgestellt, bei dem in Echtzeit auch ohne markante Schlagzeug-Stimme zwischen mehreren möglichen Schlagkandidaten für verschiedene rhythmische Muster im 4/4-Takt gewählt wird. Umgekehrt wird in [110] von einem gegebenen Grundschlag ausgehend mittels Autokorrelation eine hierarchische Gruppierung der Schläge vorgenommen, welche den metrischen Hierarchiestufen entspricht.

In Abbildung 5.4 wird schematisch die zugrundeliegende Funktionsweise der Verfahren zur

Ermittlung von Taktschlägen dargestellt: Ausgehend von einem Musikstück, das entweder als symbolische Repräsentation (etwa als MIDI-Datei) vorliegt oder auch durch eine Audioaufnahme gegeben sein kann, werden zuerst die Einsatzzeiten der Noten extrahiert bzw. im Fall einer Audio-Datei geschätzt und anschließend eine Aktivitätskurve (*onset curve*) berechnet, die jedem Zeitpunkt eines geeigneten Rasters einen Wert für die damit verbundenen Noteneinsatzzeiten (*onsets*) zuordnet, für Details siehe [6]. Bei manchen Verfahren wird die Aktivitätskurve nicht explizit aufgestellt, sondern die Folge der Abstände zwischen zwei Noteneinsatzzeiten (*inter onset intervals, IOI*) verwendet. Diese Information wird dann vom eigentlichen *beat tracker* verarbeitet, wobei einige Ansätze wie etwa [30, 34, 58, 65, 157] die Schätzung von Tempo und die Ermittlung des Grundschlags trennen, wohingegen in anderen Verfahren (darunter [15, 45, 85, 162, 188, 204]) beide Teilprobleme simultan gelöst werden.

In [30] wird ein Verfahren zur Erkennung von Schlagzeiten mittels eines zweistufigen Modells vorgestellt. In einem ersten Schritt werden die vorherrschende Periodizität und damit das lokale Tempo sowie Kandidaten für mögliche Schlagzeiten mittels Autokorrelation ohne Vorwissen ermittelt. In einem zweiten Schritt werden kontextbezogene Informationen über Tempo und frühere Schlagzeiten zur Verbesserung der extrahierten Schlagzeiten verwendet.

Eine verwandte Möglichkeit zur Ermittlung des Grundschlages basiert auf dem Optimierungsverfahren der *dynamischen Programmierung*: In [45] werden die beiden obigen Bedingungen für den Grundschlag aus [191] kombiniert, nämlich die Übereinstimmung von Taktschlägen mit möglichst vielen Noteneinsatzzeiten und die Gleichmäßigkeit der zeitlichen Abfolge der Grundschläge in einer parametrisierten Bewertungsformel (*objective function*). Durch Umwandlung in eine rekursive Darstellung dieser Formel kann mittels dynamischer Programmierung ein Optimum berechnet werden, welches einem Ausgleich zwischen den beiden Anforderungen entspricht. Ein ähnlicher Ansatz wird in [212] zur Tatum-Ermittlung und darauf aufbauender rhythmischer Quantisierung verwendet.

In [16] wird ein probabilistischer Ansatz für die rhythmische Quantisierung bei einem vorgegebenen Tempo vorgestellt. Dies wird in [15] zu einem System zur gleichzeitigen Tempobestimmung und rhythmischen Quantisierung mittels probabilistischer Methoden erweitert. Genauer werden hier *Markov Chain Monte Carlo* und sequenzielle Monte-Carlo-Methoden genutzt und das Verfahren mittels eines künstlichen synkopischen Rhythmus' und zweier Beatles-Songs ausgewertet.

Bei diesen Verfahren wird ein Kompromiss zwischen möglichst geringer Komplexität des resultierenden Notentextes und Kontinuität im Tempo gesucht. Ein ähnlicher Ansatz wurde bereits zuvor in [162] vorgestellt, bei dem die Verteilung der Zeitspannen zwischen je zwei Tonanfängen untersucht wird.

Eine Variation dieser Methoden wird durch Einführung eines speziellen Markers für Taktanfänge in [204] vorgestellt. Dieser wird als verborgener Zustand modelliert und mittels Bayesscher Methoden abgeschätzt. Ausgewertet wird diese Methode auf einem echten und einem künstlichen Stück, beide mit wechselndem Rhythmus. In [203] wird eine leichte Variation dieses Verfahrens zusätzlich auf einen konstant triolischen Rhythmus angewendet.

Hierauf aufbauend wird in [88] ein Verfahren vorgestellt, welches rhythmische Muster zur Beschreibung metrischer Strukturen in Audiodaten (Tanzmusik) verwendet. Das Verfahren beruht auf einem Hidden Markov-Ansatz, der die Gewinnung der Rhythmen direkt aus annotierten Daten erlaubt.

In [188, 189] werden anhand eines Katalogs von rhythmischen Grundmustern allgemeinere Ansätze zum Detektieren von Schlagzeiten, Takten und rhythmischen Informationen entwickelt. Ähnlich zu Methoden der Sprachverarbeitung werden hier rhythmische Ereignisse zwischen zwei Schlägen als Lexikoneinträge modelliert. Durch die Verwendung eines Hidden Markov Models zur Beschreibung dieser Abstände in Abhängigkeit von einem lokalen Tempo können so gleichzeitig eine möglichst konstante Tempokurve wie auch eine möglichst gute rhythmische Transkription bestimmt werden. Der gleiche Ansatz wird in [80] zur simultanen Schätzung von Taktschlägen und Takten (*downbeats*) verwendet.

Eine weitere Variation solcher Verfahren wird in [194] zur automatischen rhythmischen Transkription von MIDI-Dateien mittels eines generativen Bayes-Modells verwendet. Auch hier erfolgen Ermittlung des Tempos und rhythmische Quantisierung gleichzeitig. Bei der Konstruktion des Notentextes werden die Einzelstimmen hier allerdings separat betrachtet, um z. B. Artefakte durch verschiedene rhythmische Strukturen in Melodie und Begleitung zu reduzieren.

Das von uns vorgestellte Verfahren basiert auf der in [64, 65] vorgestellten Methode zur Detektion von Tempo und Impulsraster mittels Fouriertransformation, wobei nach Ermittlung der lokal vorherrschenden Periodizität (*Predominant Local Pulse, PLP*) aus diesen lokalen Tempoinformationen eine globale Impulsfolge generiert wird. Diese Methode wird detailliert im folgenden Abschnitt beschrieben.

5.4. Algorithmus

In diesem Abschnitt beschreiben wir unsere Methode zur Konvertierung von P-MIDI- nach S-MIDI-Dateien mittels Abbildung der physikalischen Zeitachse der P-MIDI-Datei auf eine geeignete musikalische Zeitachse, siehe Abbildung 5.5 für eine Übersicht. Nach Extraktion einer Aktivitätskurve aus der P-MIDI-Datei führen wir eine Periodizitätsanalyse durch, bei der wir einen Puls-Schätzer zur Berechnung einer Folge von möglichen Kandidaten für die Schlagzeiten erhalten, siehe Abschnitt 5.4.1. In Abschnitt 5.4.2 stellen wir eine Methode zur Schätzung der globalen Taktart durch Analyse der Betonungsverteilung der Kandidaten für die Schlagzeiten vor. Diese Information dient nun wiederum dazu, Inkonsistenzen der Betonungsfolge (und damit auch der Schlagzeiten) zu erkennen und aufzulösen.

Im Folgenden verwenden wir die Notation $[a:n:b] := (\lceil a \rceil + n\mathbb{N}_0) \cap [a,b]$, wobei $[a,b] := \{t \in \mathbb{R} \mid a \leq t \leq b\}$ für $a,b \in \mathbb{R}$ und $n \in \mathbb{N}$. Für den Fall $n = 1$ schreiben wir auch kurz $[a:b] := [a:1:b]$. Beispielsweise kann die Menge $\{3,8,13,18,23\}$ auch durch $[3:5:23]$ beschrieben werden.

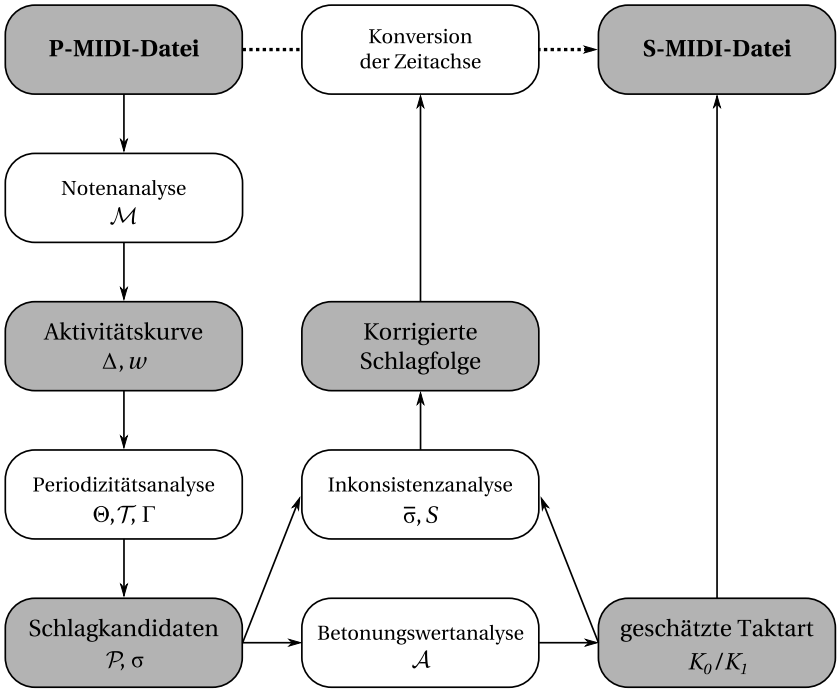


Abbildung 5.5.: Bestandteile des vorgestellten Verfahrens.

5.4.1. Erkennung der Schlagzeiten

Für die Erkennung der Schlagzeiten verwenden und erweitern wir die von [65] vorgestellte Methode. Bei dieser wird die lokal vorherrschende Periodizität von Aktivitätskurven bestimmt und eine Kurve generiert, welche die Positionen mit der höchsten Wahrscheinlichkeit für die Grundschläge bzw. Impulse angibt. Obwohl diese Methode ähnlich wie andere Beat-Tracking-Methoden (etwa [30, 45]) für Audio-Dateien entwickelt worden ist, funktioniert sie ebenfalls mit aus MIDI-Dateien generierten Aktivitätskurven.

Wir nehmen an dieser Stelle an, dass die MIDI-Datei durch geeignete Vorverarbeitung bereits eine explizite physikalische Zeitachse $[0, T]$ besitzt, wobei T das Ende der letzten MIDI-Note beschreibt, und die einzelnen Noten bereits extrahiert wurden. Somit liegt uns für eine geeignete Indexmenge $I \subset \mathbb{N}$ eine *Liste von MIDI-Noten* vor:

$$\mathcal{M} := \left(t_i, d_i, p_i, v_i\right)_{i \in I},$$

wobei $t_i \in [0, T)$ den Startpunkt der i-ten MIDI-Note beschreibt, d_i ihre Dauer in Sekunden, $p_i \in [0:127]$ ihre Notenhöhe (MIDI pitch) und $v_i \in [0:127]$ ihre Anschlagsgeschwindigkeit bzw. Lautstärke.

Ausgehend von dieser Darstellung der MIDI-Noten definieren wir für einen Gewichtungspara-

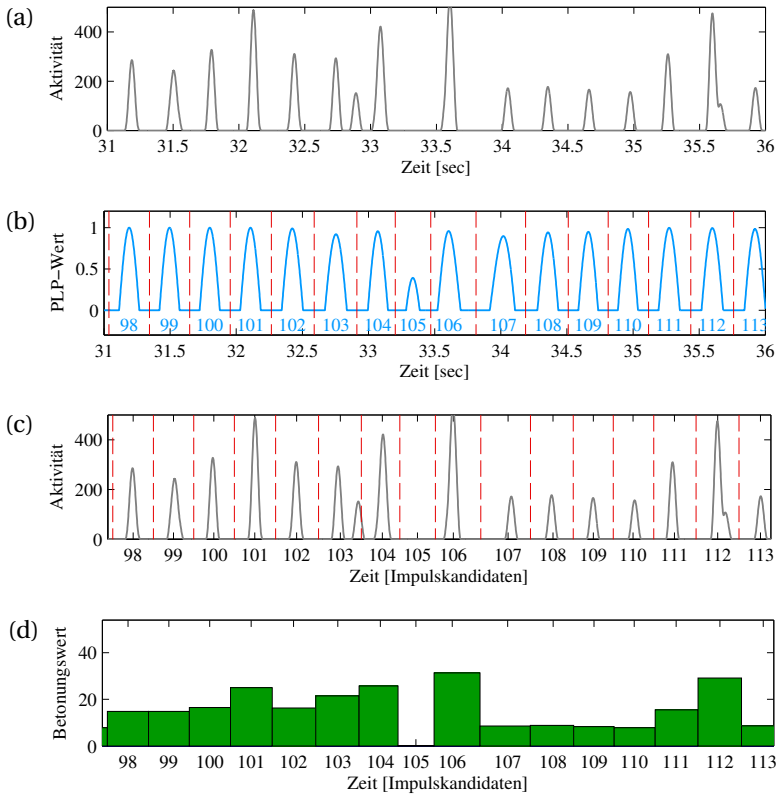


Abbildung 5.6.: Berechnung der Betonungsmuster für einen fünfsekündigen Ausschnitt aus BWV 888: **(a)** MIDI-Aktivitätskurve Δ, **(b)** PLP-Kurve Γ mit den Begrenzungen b der einzelnen Schlagzeiten, **(c)** Aktivitätskurve Δ mit Schlagzeit-Begrenzungen b, **(d)** Betonungswerte σ für die Schlagzeit-Kandidaten.

meter $w = (w_1, w_2, w_3) \in \mathbb{R}^3$ eine *MIDI-Aktivitätskurve*

$$\Delta_w(t) := \sum_{i \in I} (w_1 + w_2 \cdot d_i + w_3 \cdot v_i) \cdot h(t - t_i) \,,$$

für $t \in [0, T]$, wobei h ein um 0 zentriertes Hann-Fenster der Länge 50 ms beschreibt. In unserer Implementation fixieren wir $w := \left(1, 20, \frac{50}{128}\right)$ zum Ausbalancieren der Komponenten Länge und Anschlagsstärke. Unsere Experimente haben gezeigt, dass die Methode robust bezüglich kleinerer Änderungen dieser Werte ist. Weiterhin erlauben wir, durch einen zusätzlichen Parameter d^* eine Obergrenze für die Gewichtung der Notendauer vorzugeben (Standard: 3 s). In Abbildung 5.6a ist eine Aktivierungskurve für einen kurzen Ausschnitt aus einer MIDI-Datei illustriert.

Mittels Kurzzeit-Fouriertransformation wird aus Δ ein *Tempogramm* $\mathcal{T} : [0, T] \times \Theta \to \mathbb{C}$ für eine vorgegebene Menge Θ von zu berücksichtigen Tempo-Werten in Schlägen pro Minute (engl. *beats per Minute*, BPM) sowie zusätzlichen Parametern für Glättung (Fensterlänge) und

Zeitgranularität (Schrittweite) berechnet. Für die Details zu dieser Berechnung verweisen wir auf die ausführliche Erläuterung in [65].

Zuerst berechnen wir ein grobes Tempogramm $\mathcal{T}^{\text{coarse}}$ (vgl. hierzu auch [17]) mittels der Tempomenge $\Theta = [40:4:240]$, einer Fensterlänge von 8 s und einer Schrittweite von 1 s. Das vorherrschende globale Tempo T_0 wird nun bestimmt durch zeilenweises Aufsummieren der Absolutbeträge der Werte aus $\mathcal{T}^{\text{coarse}}$ und Ermittlung des Maximums. Typischerweise können wir von der Eindeutigkeit dieses Maximums ausgehen; sollte dies in Einzelfällen mehrdeutig sein, wählen wir per Konvention die Maximalstelle mit dem niedrigsten Index.

Anschließend berechnen wir ein zweites Tempogramm $\mathcal{T}^{\text{fine}}$ basierend auf der neuen BPM-Menge $\Theta = \left[\frac{1}{\sqrt{2}} \cdot T_0, \sqrt{2} \cdot T_0\right] \cap \mathbb{N}$, was der *Tempo-Oktave* um T_0 entspricht. Für dieses Tempogramm beträgt die Fensterlänge $5 \cdot \frac{60}{T_0}$ s und wir verwenden eine feinere Schrittweite von 0,2 s. Die Auswahl des BPM-Bereiches auf diese Art und Weise verhindert unerwünschte Sprünge zwischen Vielfachen des detektierten Tempos. Die Wahl der Fensterlänge von fünf erwarteten Schlagzeiten basiert auf der Annahme, dass ein stabiles Tempo etwa für fünf Schläge konstant bleibt.

Analog zu [65] schätzen wir mittels des Tempogramms $\mathcal{T}^{\text{fine}}$ das vorherrschende Tempo für jede Zeitposition und verwenden diese Informationen zur Bestimmung sinus-ähnlicher Kerne, welche jeweils die lokale Periodizität (engl. *predominant local pulse*, PLP) der zugrundeliegenden Aktivitätskurve Δ am besten beschreiben. Diese Kerne werden zu einer *PLP-Kurve* $\Gamma : [0, T] \to [0, 1]$ zusammengefügt, welche die Kandidaten für Impulse bzw. Schlagzeiten auf der physikalischen Zeitachse beschreibt, siehe auch Abbildung 5.6b. Die den lokalen Maximalstellen von Γ entsprechenden Zeitpunkte bilden eine Folge

$$\mathcal{P} = (\mathcal{P}_1 < \ldots < \mathcal{P}_N),$$

welche eine gute erste Approximation der musikalischen Grundschläge darstellt. Allerdings ist es gerade bei Aufnahmen mit vielen Tempoänderungen sehr wahrscheinlich, dass diese Folge einige zusätzliche Elemente enthält, die keine musikalische Schlagzeit beschreiben, oder bei der musikalisch relevante Taktschläge nicht als solche erkannt worden sind. Gerade bei starken Ritardandi, wie sie etwa in der Barockmusik üblich sind, wird durch das verwendete Verfahren oftmals ein zusätzlicher Schlag eingefügt. Im nächsten Abschnitt stellen wir eine Methode zur Nachverarbeitung vor, die in einem gewissen Rahmen zum Aufspüren und Beheben dieser Fehler geeignet ist.

5.4.2. Optimierung der Liste möglicher Schlagzeiten

Das in diesem Abschnitt beschriebene Verfahren beruht auf der Ermittlung einer globalen Taktart und Verwendung derselben für das Entfernen von Inkonsistenzen der im vorangegangen Abschnitt berechneten Impulsfolge. Wir nehmen an, dass die Taktart des betrachteten Musikstücks sich im Verlauf des Stückes nicht ändert. Die Taktart kann durch die Analyse wie-

derkehrender Betonungen in der Impulsfolge durch eine Kurzzeit-Autokorrelation abgeschätzt werden. In einem zweiten Schritt vergleichen wir die relative Position jedes Impulskandidaten mit einem durch die Taktart induziertem Taktraster und ermitteln Abweichungen, um so einzelne fälschlich gesetzte Impulse zu korrigieren. Abschließend werden die durch die korrigierte Folge beschriebenen Zeitpunkte als Grundschläge und damit als neue musikalische Zeitachse interpretiert, und die Tick-Positionen aller MIDI-Events auf diese neue Zeitachse abgebildet.

Im Folgenden beschreiben wir dieses Optimierungsverfahren detaillierter. Zuerst akkumulieren wir die Aktivität für den n-ten Impulskandidaten durch Definition seines *Betonungswertes* (engl. *salience*):

$$\sigma(n) := \int_{b(n-1)}^{b(n)} \Delta(t)\, dt \quad (n \in [1:N]), \tag{5.1}$$

wobei die Grenzen der zu diesem Schlag gehörenden Zeitspanne durch die arithmetischen Mittel des Schlags mit seinen Nachbarschlägen definiert sind: $b(n) = \frac{1}{2} \cdot (\mathcal{P}_n + \mathcal{P}_{n+1})$ für $1 \leq n < N$, $b(0) = 0$ und $b(N) = T$. Zur Illustration der Berechnung von b und σ siehe Abbildung 5.6b-d.

Unser nächstes Ziel ist die Berechnung einer Schätzung für die Taktart K_0/K_1, zu dessen Zweck wir eine Betonungsanalyse mittels Autokorrelation durchführen. Um sicher zu stellen, dass Fehler in $\mathcal{P}$ und σ nur lokale Auswirkungen haben, verwenden wir eine Kurzzeit-Autokorrelation: Für eine feste Fensterbreite $K > 12$ (in unserer Implementierung: $K = 32$) betrachten wir die $K \times N$-Matrix

$$\mathcal{A}(k,n) := |I_k|^{-1} \sum_{i \in I_k} \sigma(n+i) \cdot \sigma(n+i+k),$$

wobei $I_k := [0:k:K-k-1]$ und $\sigma(n) := 0$ für alle $n \in \mathbb{Z} \setminus [1:N]$. Somit gibt $\mathcal{A}(k,n)$ ein Maß für die Plausibilität für die Vermutung an, dass die betonten Taktschläge im Bereich des n-ten Impulses periodisch mit Periodenlänge k auftreten.

Die vorherrschende Periodenlänge K_0, welche den Zähler der geschätzten Taktart darstellt, bestimmen wir nun durch zeilenweises Aufsummieren einer relevanten Submatrix von $\mathcal{A}$ und Ermittlung des Index mit dem maximalen Wert:

$$K_0 := \operatorname*{arg\,max}_{k \in [3:12]} \sum_{n=1}^{N} \mathcal{A}(k,n).$$

Zur Steigerung der Robustheit haben wir die Fälle $k < 3$ und aus musikalischen Gründen die Fälle $k > 12$ aus dem Bereich für die Suche der maximalen Zeilensumme ausgeschlossen. (Der Ausschluss des Falles $k = 2$ verursacht keine ernsthaften Schwierigkeiten, da wir beispielsweise die Taktart 4/8 als Ersatz für 2/4 nutzen können.) Die relevanten Zeilen der Matrix $\mathcal{A}$ sind in Abbildung 5.7b dargestellt, wobei hier $K_0 = 6$ gilt. Der Nenner K_1 der Taktart, also die musikalische Einordnung des Grundschlags, ist für die Berechnung nicht nötig. Er wird in Abhängigkeit des vorherrschenden Tempos T_0 so gesetzt, dass sich ein Wert zwischen 70 und 140 Viertelschlägen pro Minute ergibt. Wurde das Haupttempo beispielsweise auf 200 BPM

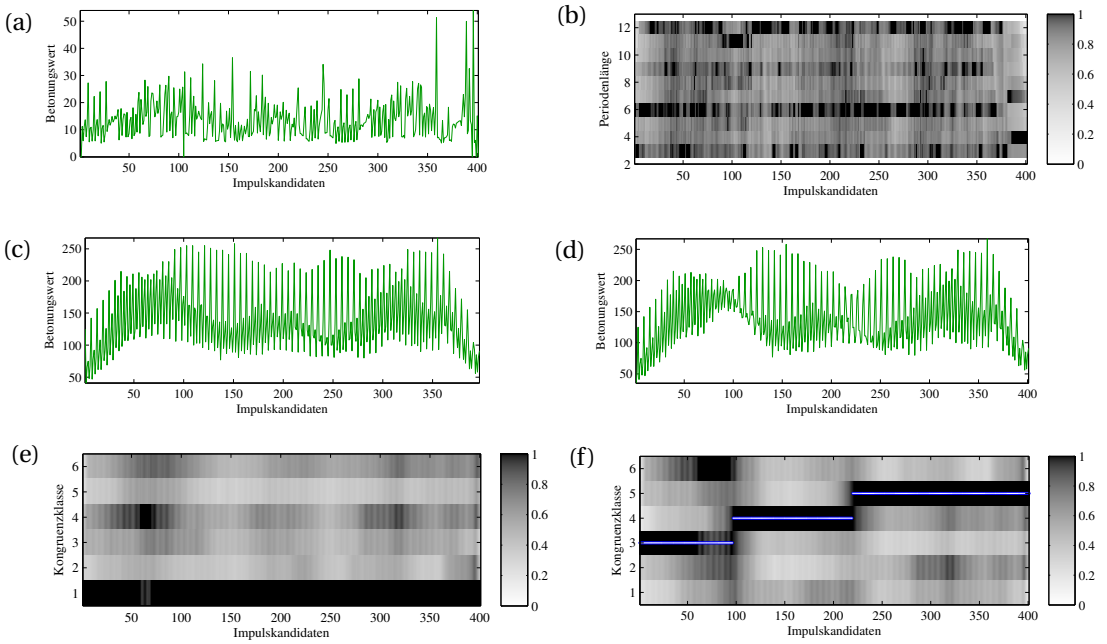


Abbildung 5.7.: Illustration der Einzelschritte des Verfahrens zur Erkennung der Inkonsistenzen in der Betonungsfolge anhand des Beispiels BWV 888: **(a)** Betonungsfolge σ wie in Abbildung 5.6d. **(b)** Ausschnitt der Kurzzeit-Autokorrelationsmatrix $\mathcal{A}$ von σ mit höchster Zeilensumme in Zeile 6. **(c)** 6-gekämmte Betonungsfolge $\bar{\sigma}$, falls alle Schlagkandidaten korrekt erkannt wurden. **(d)** 6-gekämmte Betonungsfolge $\bar{\sigma}$ falls zwei zusätzliche Schläge eingefügt wurden. **(e)** Stressgramm mit maximalen Betonungswerten in der 1. Kongruenzklasse. **(f)** Stressgramm mit zwei Wechseln der Kongruenzklasse maximaler Betonungswerte und Pfad der lokalen Kandidaten für die Klasse der Taktanfänge.

geschätzt, so wird der Grundschlag als Achtelnote interpretiert (bei einem Tempo von 100 Vierteln pro Minute).

Mittels K_0 sind wir nun in der Lage, die Impulsfolge auf Inkonsistenzen zu untersuchen. Zur Motivation betrachten wir vorerst den Fall, dass alle detektierten Impulskandidaten tatsächlich korrekte musikalische Grundschläge sind. In diesem idealisierten Szenario beschreibt die Einschränkung von $\mathcal{P}$ auf die n-te K_0-Kongruenzklasse $[n : K_0 : N]$, $n \in [1 : K_0]$, alle Impulse auf der n-ten Zählzeit auf semantisch sinnvolle Weise. Insbesondere korrespondiert die erste Klasse ($n = 1$) genau zu allen Taktanfängen, falls das betrachtete Stück nicht mit einem Auftakt beginnt. Eine analoge Zerlegung angewendet auf σ führt zu Betonungsmustern einer jeden Zählzeit. Trotz rhythmischer Variationen erwarten wir, dass die erste Kongruenzklasse (und damit auch die erste Zählzeit) meistens den höchsten Betonungswert aufweist. Zur Stärkung der Robustheit wird σ lokal innerhalb der K_0-Kongruenzklassen geglättet,

$$\bar{\sigma}(n) := \sigma(n) + \sum_{k=1}^{\lfloor K/K_0 \rfloor} \sigma(n \pm k \cdot K_0), \tag{5.2}$$

wie es in Abbildung 5.7c illustriert wird. Da die Einschränkung auf je eine Kongruenzklasse an einen Kamm erinnert, nennen wir $\bar{\sigma}$ auch die K_0-*gekämmte* Version von σ.

Falsch detektierte Impulskandidaten stören die Zuordnung aller Taktanfänge zu einer speziellen Kongruenzklasse. In diesem Fall beobachten wir einen Wechsel der die starken Betonungswerte enthaltenen Klasse an einigen Zeitpunkten. Zur Analyse dieses Phänomens definieren wir eine $K_0 \times N$-Matrix $\mathcal{S}$, welche die lokale Betonungsverteilung der Kongruenzklassen enthält. Präziser definieren wir

$$S(k,n) := \bar{\sigma}(k) \cdot \delta(k \equiv_{K_0} n),$$

wobei δ ein Kronecker-Delta bezeichnet, also $\delta(A) := 1$ falls die Aussage A gilt und ansonsten 0. Glätten wir S entlang der Zeitachse (zeilenweise) mit einem Hann-Fenster der Länge $2 \cdot K_0$, erhalten wir ein sogenanntes *Stressgramm* (engl. *stress* = Betonung). Solche Stressgramme sind für das idealisierte Szenario (Abbildung 5.7e) ebenso visualisiert wie bei Anwesenheit zweier zusätzlich detektierter Impulse (Abbildung 5.7f).

Diesen Fall wollen wir nun etwas detaillierter diskutieren: Zu Beginn stellen wir fest, dass die Schätzung von K_0 nur lokal gestört wird, was nicht zu einer Änderung der vermuteten Taktart führt (vgl. hierzu Abbildung 5.7b). Hingegen stimmt die Zerlegung in K_0-Kongruenzklassen nicht länger semantisch mit den Zählzeiten überein, da alle Impulse hinter dem zusätzlich eingefügten um eine Schlagposition verschoben sind. Im Stressgramm $\mathcal{S}$ wird dies durch einen Wechsel der Zeile mit hohem Betonungswert deutlich.

Wiederum zur Stärkung der Robustheit wechseln wir zu einer gröberen Sichtweise durch Berechnung eines maximalen Pfades durch das Stressgramm von links nach rechts mittels dynamischer Programmierung. Jeder Punkt in diesem Pfad zeigt die Kongruenzklasse mit der höchsten Wahrscheinlichkeit zur Repräsentation der Taktanfänge. Dies bedeutet: Wenn die Taktanfänge in der Klasse mit dem Index i lokalisiert sind, dann kann in der Nähe des zusätzlichen Impulses ein Wechsel zur Klasse $i+1 \mod K_0$ beobachtet werden, wie es zweimal in Abbildung 5.7f geschieht. Der Fall eines fehlenden Impulskandidaten verhält sich ähnlich, hier ändert sich der besagte Zeilenindex auf $i-1 \mod K_0$.

Diese so ermittelten Inkonsistenzen können nun gelöst werden, indem entweder falsch detektierte Impulskandidaten entfernt oder Positionen ermittelt werden, an denen ein offensichtlich fehlender Impuls ergänzt werden kann. Details hierzu werden in Abschnitt 5.4.3 diskutiert.

Abschließend definiert die so korrigierte Impulsfolge ein Schlagraster in der P-MIDI-Datei, woraus eine Sequenz von Tick-Positionen abgeleitet werden kann, die musikalisch sinnvollen Grundschlägen entsprechen. Durch Abbildung dieser Tick-Positionen auf neue Tick-Werte, die alle in einem festen Abstand (etwa 960 Ticks) zueinander stehen, durch Hinzufügen sinnvoller MIDI-Befehle zur Tempoänderung und durch lineare Interpolation aller Ticks zwischen den ermittelten Impulsen kann die frühere Zeitachse der P-MIDI durch eine musikalische Zeitachse einer neuen S-MIDI ersetzt werden. Im Falle eines Auftakts werden zusätzliche Grundschläge zu Beginn des Stückes eingefügt, sodass die erste K_0-Kongruenzklasse den er-

mittelten Taktanfängen entspricht. Als letztes wird die Taktart K_0/K_1 am Beginn (Tick-Position 0) der neuen S-MIDI eingefügt.

5.4.3. Einfügen und Löschen von Impulskandidaten

In der Beschreibung unseres Verfahrens zur Optimierung einer Folge von Impulskandidaten haben wir eine Möglichkeit zur zeitlichen Lokalisation einer Abweichung vom Taktraster vorgestellt. Der damit ermittelte Zeitpunkt stimmt allerdings wegen Glättungseffekten nicht notwendigerweise mit einem Fehler in der Impulsfolge überein, sondern dient lediglich zur Angabe einer problematischen Region. Somit verbleibt die Klärung der Frage, wie genau ein musikalisch bedeutungsloser Impulskandidat ermittelt bzw. der Zeitpunkt zum Einfügen eines zusätzlichen Grundschlags festgelegt werden kann.

Eine Möglichkeit hierzu besteht darin, innerhalb jeder dieser Problemregionen den Schlagkandidaten mit dem niedrigsten Betonungswert σ oder dem niedrigsten PLP-Wert[5] zu ermitteln. Um einen zusätzlichen Schlag einzufügen, genügt es nach zwei benachbarten relativ niedrigen Werten der PLP-Kurve zu suchen und den Schlag zwischen diesen beiden Werten einzufügen. In unserer Implementierung verwenden wir zur Ermittlung der zu korrigierenden Impulskandidaten eine Kombination dieser Ansätze sowie ein einfaches Verfahren zur *adaptiven rhythmischen Quantisierung*.

Prinzipiell wird bei einem Quantisierungsverfahren von einem minimalen musikalisch sinnvollen Zeitintervall ausgegangen – etwa einer Sechzehntel- oder Zweiunddreißigstelnote, oder auch von entsprechenden Triolen – und daraus ein gleichmäßiges Raster erzeugt, an das alle Notenanfänge und -enden durch möglichst kleine Verschiebungen angepasst werden. Solche Methoden gehören zur grundlegenden Funktionalität jedes MIDI-Sequenzers und finden sich auch in den gängigen Notensatzprogrammen wie *Sibelius*, *Finale* oder *capella*. Die Ergebnisse dieser Quantisierungen sind bei regelmäßigen Rhythmen sehr gut, weisen aber bei sporadisch auftretenden deutlich schnelleren rhythmischen Läufen im Allgemeinen Fehler in Form von Clustern der Einzelnoten auf.

Bei dem von uns vorgestellten Verfahren verwenden wir ein adaptives Raster, indem wir verschiedene musikalische Zeitintervalle zulassen und den Grad der notwendigen Verschiebung für jedes Raster bestimmen. Genauer betrachten wir eine Menge von zulässigen *Schlagunterteilungen* $J := \{1,2,3,4,6,8\}$, was bei Annahme eines Grundschlages in Viertelnoten einer erlaubten Unterteilung in Viertel, Achtel, Achteltriolen, Sechzehntel, Sechzehnteltriolen und Zweiunddreißigsteln entspricht. Für jeden Schlagkandidaten und jede Unterteilung $j \in J$ berechnen wir Fensterfunktionen $w_{\bullet,j}$ (vgl. Abbildung 5.8) und Quantisierungskosten p mittels

[5] Der PLP-Wert kann analog zu Gleichung 5.1 berechnet werden, indem in dieser Gleichung Δ durch Γ ersetzt wird.

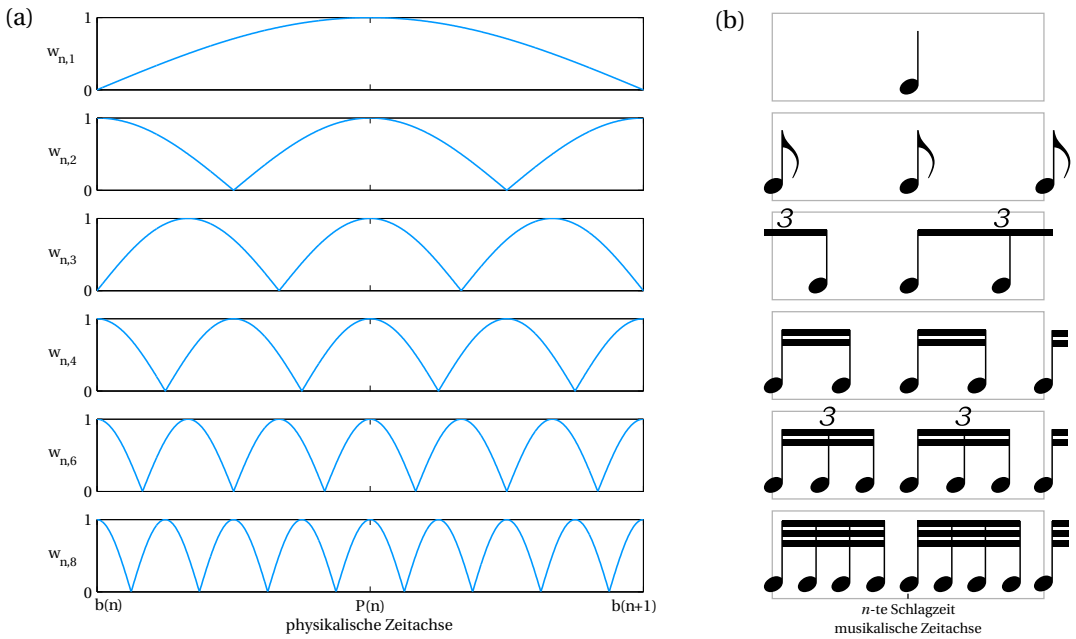


Abbildung 5.8.: Mögliche Unterteilungen der Schlagkandidaten. **(a)** Quantisierungskurven $w_{n,j}$ für $j \in \{1,2,3,4,6,8\}$. **(b)** Entsprechende Notenwerte.

eines Dämpfungsparameters α (in unserer Implementation haben wir $\alpha = 0{,}1$ gesetzt):

$$\begin{aligned} w_{n,j}(t) &= \left|\cos\left(j \cdot \pi \cdot \left(\tfrac{t-b(n)}{b(n+1)-b(n)} - 0{,}5\right)\right)\right| \\ p(n,j) &= \int_{b(n)}^{b(n+1)} w_{n,j}(t) \cdot \Delta(t)^{\alpha}\, dt. \end{aligned}$$

Als geschätzte Schlagunterteilung des n-ten Schlagkandidaten wählen wir den Index seiner minimalen Quantisierungskosten

$$\bar{j} := \operatorname*{arg\,min}_{j \in J} p(n,j)\,,$$

der Wert an dieser Stelle $p(n) := p(n,\bar{j})$ beschreibt dann die Quantisierungskosten dieses Kandidaten.

Diese Information kann nun einerseits dazu verwendet werden, die Noteneinsatzzeiten sowie die Notendauern der resultierenden S-MIDI entweder auf die Schlagzeiten selbst oder adaptive Schlagunterteilungen zu quantisieren. Hierbei ist allerdings eine weitere heuristische Optimierung notwendig, um zu verhindern, dass Noten mit einer Länge von 0 Ticks erzeugt werden. Eine solche Note würde bedeuten, dass ein `NOTE_ON`-Befehl gleichzeitig mit oder sogar kurz nach seinem entsprechenden `NOTE_OFF`-Befehl gesendet werden würde, was zum endlosen Abspielen dieses Tons führen würde (sogenannte »MIDI-Hänger«). Andererseits wird durch

die Quantisierungskosten auch abgebildet, wie gut die Notenwerte im Umfeld eines Schlagkandidaten mit diesem korrespondieren. Weist ein Schlag im Vergleich zu seinem Umfeld hohe Quantisierungskosten auf, so ist die Wahrscheinlichkeit größer, dass genau dieser Kandidat eine fehlerhafte Schätzung des PLP-Algorithmus darstellt. In unserer Implementierung ordnen wir jedem Schlagkandidaten als »Gesamtpunktwert« den Quotienten aus PLP-Wert und Quantisierungskosten zu und ermitteln für die Korrekturen in jeder Problemregion den Kandidaten mit minimalen Gesamtpunktwert.

5.5. Evaluation

Die Auswertung der Ausgabe eines Programms zur Grundschlagerkennung ist eine nichttriviale Problemstellung, was unter anderem an der unspezifischen Definition der Schlagzeiten liegt, siehe auch [34, S. 21]. Insbesondere die Ermittlung der Granularität des Grundschlagrasters, also der Entscheidung zwischen ähnlichen Taktarten wie 6/8 und 3/4 oder Vielfachen wie 4/4 oder 8/4, scheint ein schlecht gestelltes Problem darzustellen, weswegen wir es in den folgenden Betrachtungen als unwesentlich ansehen werden. Auch für den Menschen kann diese Aufgabe sehr herausfordernd sein, insbesondere wenn das betrachtete Stück viele rhythmische Variationen aufweist oder das Tempo in einer expressiven musikalischen Aufführung sehr stark schwankt. Unsere Art der Evaluation orientiert sich sowohl an [188], in dem unter anderem der visuelle Eindruck des berechneten Notentextes bewertet wird, als auch an [34], in dem ein Vergleich mit manuell extrahierten Taktschlägen sowie Hörtests für die subjektive Bewertung der wahrgenommenen Qualität vorgeschlagen werden.

Bedingt durch die Modellierung ist das von uns vorgestellte Verfahren nicht effektiv auf sämtliche Typen von P-MIDI-Dateien anzuwenden. Der in Abschnitt 5.4.1 beschriebene Ansatz mittels PLP-Kurven stellt einige Bedingungen an die zu untersuchenden MIDI-Dateien wie einen meistens regelmäßigen Rhythmus oder ein nahezu gleichbleibendes Tempo für einen gewissen Zeitraum (in unserer Implementierung beträgt die Länge dieses Zeitfensters etwa fünf Sekunden). Weiterhin werden Sprünge in andere Tempooktaven (doppeltes/halbes Tempo) nicht berücksichtigt. Für den in Abschnitt 5.4.2 beschriebenen Optimierungsschritt wird eine globale, unveränderliche Taktart vorausgesetzt. Weiterhin müssen Taktanfänge durch Betonung oder gehäuftes Vorkommen langer Notenwerte erkennbar sein.

Im Folgenden werden wir einige typische Beispiele für P-MIDI-Dateien detaillierter betrachten und anschließend eine automatische Analyse auf einem kleinen Testdatensatz künstlich gestörter MIDI-Dateien durchführen.

5.5.1. Qualitative Evaluation

Die Leistungsfähigkeit der vorgestellten Prozedur ist bereits mittels des *Präludiums BWV 888* von Johann Sebastian Bach durch die Abbildungen 5.1, 5.6 und 5.7 gezeigt worden. Die beiden durch Ritardandi verursachten zusätzlichen Impulse wurden richtig erkannt und korrigiert,

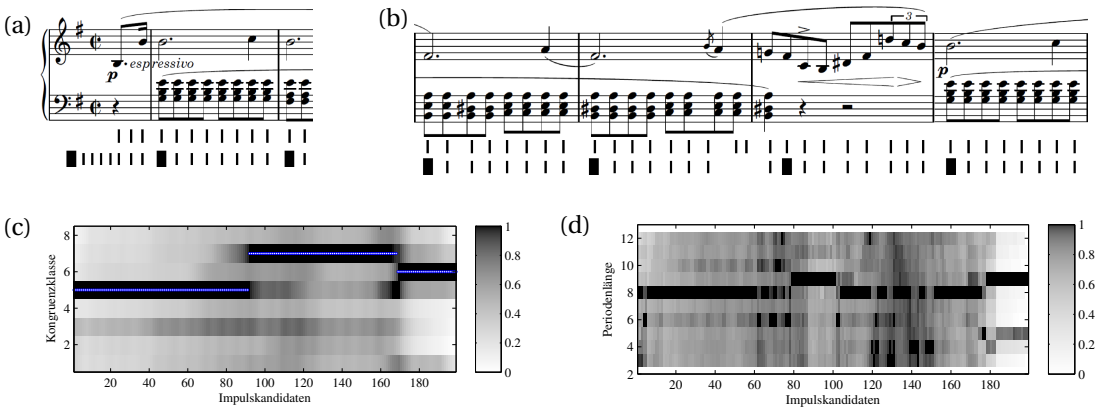


Abbildung 5.9.: Stück Nr. 4 aus Chopins op. 28. Die Partiturausschnitte zeigen die erkannten Impulskandidaten (obere Reihe) und die in der Nachverarbeitung ermittelten Taktschläge. Die Taktanfänge werden durch breitere Striche angezeigt. **(a)** Korrekt erkannter Auftakt. **(b)** Gemeinsame Korrektur zweier aufeinanderfolgender Fehler. **(c)** Stressgramm mit hervorgehobenem Pfad maximaler Betonungswerte. **(d)** Kurzzeit-Autokorrelationsmatrix.

auch die fehlerhaften Pausen zu Beginn des Stücks wurden entfernt. Die geschätzte Taktart 6/8 ist der notierten Taktart 12/8 wesensähnlich.

Als zweites Beispiel sehen wir uns das vierte Stück aus den *24 Préludes* (op. 28) von Frédéric Chopin an. Die Abbildung 5.9 zeigt zwei Notenausschnitte gemeinsam mit den erkannten Impulskandidaten und der geschätzten Taktstruktur ebenso wie das dazugehörige Stressgramm und die Kurzzeit-Autokorrelationsmatrix für das gesamte Stück.[6] Präludium Nr. 4 beinhaltet einige lange Notenwerte an den Taktanfangspositionen, was zu einer stabilen Erkennung des Taktrasters führt. Als Grundschlag wurden Achtelschläge erkannt (siehe Abbildung 5.9d), da diese eine durchgehend starke Präsenz in der linken Hand aufweisen. Das Stück beginnt mit einem Auftakt von einer Viertelnote. Da das MIDI-Format Auftakte nicht direkt unterstützt, wurden von unserer Methode zusätzliche Impulse zu Beginn des Stückes eingefügt, sodass der erste Impuls immer in der Kongruenzklasse der Taktanfänge lokalisiert wird. (Abb. 5.9a).

Bedingt durch deutliche Tempowechsel gemeinsam mit kurzen Appoggiatura und einer Triole im Umfeld des 90. Impulses werden vom verwendeten PLP-Verfahren zwei zusätzliche Impulse in zwei aufeinanderfolgenden Takten detektiert (Abb. 5.9b). Im Stressgramm wird dies durch einen Sprung des Betonungspfades über zwei Kongruenzklassen angezeigt (Abb. 5.9c). Man beachte, dass dieser Fehler keinen großen Einfluss auf die Berechnung der Taktart hat (Abb. 5.9d). Obwohl das Entfernen eines korrekt detektierten Schlags im zweiten Takt von Abb. 5.9b zu einem falsch detektierten Taktanfang im darauffolgenden Takt führt, wird das globale Taktraster im vierten Takt wiederhergestellt. Dies zeigt, wie unser Verfahren das Taktraster optimiert, ohne dabei jeden einzelnen Fehler in der Impulsfolge exakt korrigieren zu müssen.

[6] Die Beispiele aus diesem Abschnitt sind Aufnahmen eines MIDI-Klaviers und entnommen aus *Saarland Music Data* [125], (`http://www.mpi-inf.mpg.de/resources/SMD/`). Die Notenbeispiele stammen von *Mutopia* (`http://www.mutopiaproject.org/`).

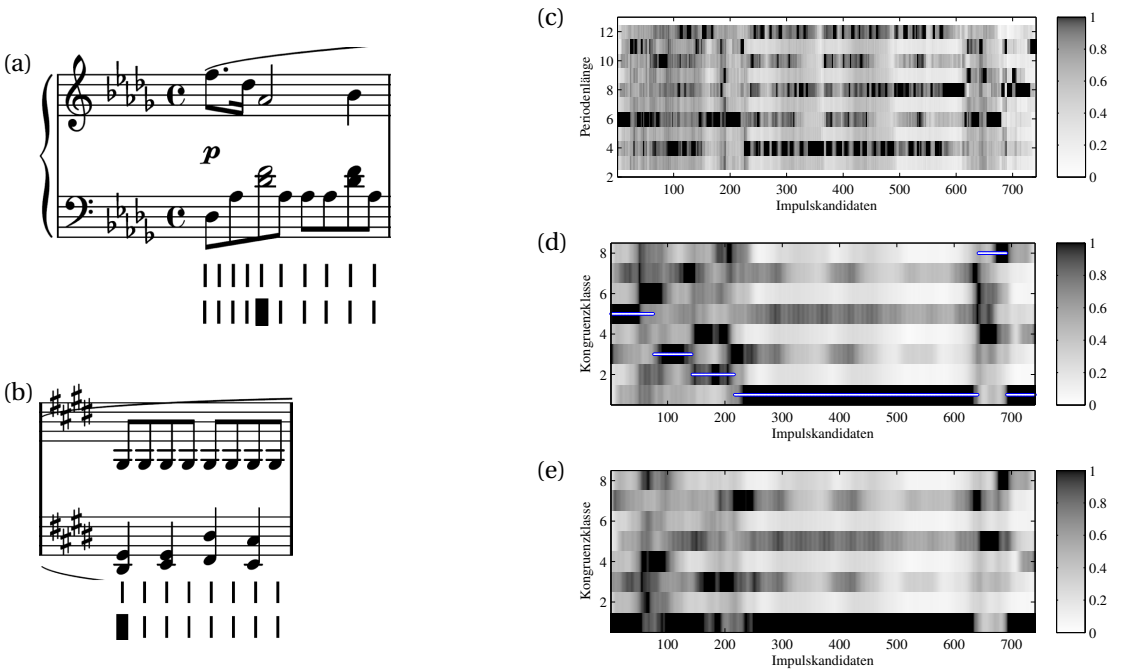


Abbildung 5.10.: Stück Nr. 15 aus Chopins op. 28. Die Partiturausschnitte zeigen die erkannten Impulskandidaten (obere Reihe) und die in der Nachverarbeitung ermittelten Taktschläge. Die Taktanfänge werden durch breitere Striche angezeigt. **(a)** Synkopische Stelle zu Beginn des Stückes. **(b)** Rhythmisch homogene Stelle im Mittelteil. **(c)** Kurzzeit-Autokorrelationsmatrix. **(d)** Stressgramm mit hervorgehobenem Pfad maximaler Betonungswerte. **(e)** Stressgramm der modifizierten Kandidatenliste.

Mit einem weiteren Beispiel aus diesem Zyklus illustrieren wir die Beschränkungen und Grenzen unserer Methode. Das Stück Nr. 15 »Regentropfen-Prélude« aus derselben Sammlung weist eine typische ternäre Form auf, die sich auch in verschiedenen Rhythmen äußert: Die beiden Randsegmente bestehen aus einem stark synkopischen Rhythmus mit vielen kleinen Tempoabweichungen, wohingegen der große Mittelteil eine rhythmisch weitgehend homogene Struktur aufweist. Die Kombination aus instabilem Tempo und Betonungsverschiebungen führt dazu, dass sowohl die ursprüngliche Schätzung der Schlagkandidaten als auch die Korrekturen durch die Stressgramm-Darstellungen fehlschlagen.

In Abbildung 5.10a und b sind zwei kurze Passagen aus dem Notentext zusammen mit den erkannten Schlägen dargestellt. Abbildungsteil a zeigt, dass direkt zu Beginn die Betonung nicht auf den ersten Schlag erfolgt, sodass unser Verfahren den ersten Taktanfang auf den zweiten Viertelschlag schätzt. Teil b zeigt einen Takt aus dem rhythmisch homogenen Mittelteil, bei dem trotz leichter Tempovariationen der Grundschlag fehlerfrei geschätzt wird. In der Autokorrelationsmatrix in Abbildung 5.10c ist zu sehen, dass bei diesem Stück selbst die Schätzung der Taktart massiv erschwert wird. Die dargestellten Spektrogramme sowohl vor (5.10d) als auch

nach erfolgter Anpassung durch Modifikation der Liste möglicher Schlagkandidaten (5.10e) zeigen, dass zu Beginn und Ende des Stückes keine einheitliche Betonungsfolge auf einer schweren Zählzeit rekonstruiert werden kann. Somit ist auch die Ermittlung und Beseitigung einzelner »Fehler« in der Kandidatenliste nicht möglich.

In Abschnitt 5.6.1 stellen wir eine Benutzeroberfläche für unser Verfahren vor, welches sowohl die Tempokurve als auch die Stressgramme darstellt. Hierdurch erhält der Benutzer einen unmittelbaren Einblick in die geschätzte Güte des Ergebnisses und kann andere Parameterkonfigurationen ausprobieren, die je nach Stück gegebenenfalls zu einem besseren Ergebnis führen.

5.5.2. Automatische Evaluation

Weiterhin haben wir unser Verfahren auf symbolischen MIDI-Dateien evaluiert, indem die S-MIDI-Dateien automatisch durch zusätzliche abrupte Tempowechsel gestört wurden. Eine ähnliche Herangehensweise wurde in [64] verwendet, um die Fähigkeiten der PLP-Methode zum Auffinden weicher Tempoänderungen zu demonstrieren.

Da das Ziel unseres Verfahrens insbesondere im Auffinden von korrekten Taktpositionen liegt, nutzen wir die in Abschnitt 2.7 vorgestellten Evaluationsmaße *precision* (***P***), *recall* (***R***) und *F-measure* (***F***) auf der Menge aller MIDI-Noten. Dabei wird eine Note als »relevant« angesehen, wenn sie in der S-MIDI an einer Taktanfangsposition beginnt, und sie gilt als »erkannt«, wenn sie durch unsere Methode wieder auf eine Taktanfangsposition abgebildet wird. Bedingt durch fehlende rhythmische Quantisierung erlauben wir eine Toleranz von ±5% für den Bereich des Taktanfangs.

Durch Verwerfen der Information über die musikalische Zeitachse und alleinige Verwendung der physikalischen Zeitpunkte (in Millisekunden) aller MIDI-Ereignisse, simulieren wir aus einer S-MIDI-Datei eine interpretierte P-MIDI. Die systematischen Störungen werden durch Hinzufügen von Tempowechseln in Höhe von ±20% um das ursprüngliche Tempo in zehnsekündigem Abstand während des gesamten Stückes realisiert.

Als Testdatenmenge verwenden wir S-MIDI-Dateien der *Fünfzehn Fugen* von Ludwig van Beethoven aus der Musikdatenbank IMSLP[7]. Bei diesen Stücken sind die Notendauern ausreichend für eine gute Schätzung der Impulse durch die PLP-Methode. Das Hinzunehmen von variierenden Tonanschlagsstärken aus echten MIDI-Einspielungen lässt eine weitere Verbesserung der Ergebnisse erwarten.

Die Ergebnisse der automatischen Evaluation sind in Tabelle 5.1 dargestellt. Wir haben sowohl die von der in Abschnitt 5.4.1 beschriebenen PLP-Tracking-Methode berechnete, ursprüngliche Impulsfolge als auch die nachverarbeitete Version ausgewertet. In beiden Fällen haben wir die geschätzte Taktart zur Identifizierung der Schläge an den Taktanfängen verwendet. Alle Stücke außer Fuge Nr. 11 weisen die Taktart 2/2 auf, welche meistens als 4/4 und manchmal als

[7] Petrucci Music Library, `http://imslp.org/wiki/15_Fugues_(Beethoven,_Ludwig_van)`

Stück	Vollständig			PLP			# Korrekturen		
	F	P	R	F	P	R	+	-	Auftakt
Nr. 1	0,477	0,587	0,402	0,372	0,509	0,293	0	2	0
Nr. 2	0,978	1	0,956	0,397	0,56	0,308	1	0	1
Nr. 3	0,656	0,663	0,649	0,144	0,196	0,113	1	0	0
Nr. 4	0,945	0,984	0,909	0,738	0,804	0,682	2	0	0
Nr. 5	0,966	0,971	0,962	0	0	0	0	0	2
Nr. 6	0,996	1	0,993	0,996	1	0,993	0	0	0
Nr. 7	0,826	0,832	0,82	0,324	0,386	0,28	1	4	1
Nr. 8	0,953	0,985	0,923	0,821	0,945	0,725	0	1	0
Nr. 9	0,896	0,916	0,876	0,787	0,855	0,73	1	0	1
Nr. 10	0,581	0,579	0,582	0,008	0,013	0,005	2	2	1
Nr. 11	1	1	1	1	1	1	0	0	0
Nr. 12	0,994	1	0,988	0,792	0,842	0,748	3	1	0
Nr. 13	0,656	0,884	0,522	0,245	0,393	0,178	0	2	2
Nr. 14	0,975	0,995	0,957	0,432	0,75	0,303	1	3	0
Nr. 15	0,692	0,98	0,535	0,633	0,992	0,465	0	0	4
ø	0,839	0,892	0,805	0,513	0,616	0,455	0,8	1	0,8

Tabelle 5.1.: Evaluationsergebnisse für die Fünfzehn Fugen von Beethoven sowohl für die vollständige Methode als auch für reine PLP-basierte Impulserkennung.

8/4 detektiert worden ist. Verglichen mit den Ergebnissen des PLP-Verfahrens zur Schätzung des Grundschlags, welches nicht für die Erkennung von Takten entwickelt worden ist, sind die Ergebnisse für einige Stücke durch das vorgestellte Verfahren signifikant verbessert worden. Zum Beispiel hat unsere Nachverarbeitungsmethode in Fuge Nr. 7 einen Schlag hinzugefügt und vier andere entfernt. Zu Beginn des Stückes wurde ein einzelner zusätzlicher Schlag zur Vermeidung von (durch Auftakte bedingte) Verschiebungen hinzugefügt. Diese Veränderungen haben zu einer Verbesserung des $\boldsymbol{F}$-Wertes von 0,324 auf 0,826 geführt, was große Auswirkungen etwa auf die Menge an zusätzlichem Aufwand bei der manuellen Optimierung hat, wenn diese MIDI-Datei in ein Notensatzprogramm importiert wird.

5.6. Erweiterungen

Nun stellen wir einige Erweiterungen der in den vorherigen Abschnitten beschriebenen Methode zum Umwandeln einer interpretierten P-MIDI-Datei in eine partiturnahe S-MIDI-Datei vor. An erster Stelle steht eine graphische Benutzeroberfläche (Abschnitt 5.6.1), die sowohl das einfache Einstellen der relevanten Parameter als auch eine manuelle Qualitätskontrolle durch Sonifizierung des detektierten Taktrasters und Bereitstellung von Plots analog zu Abbildung 5.7 erlaubt. Weiterhin werden einige Verbesserungsmöglichkeiten unseres Verfahrens vorgestellt. Diese können sowohl zu einer stabileren Schätzung des Schlagrasters beitragen und somit die

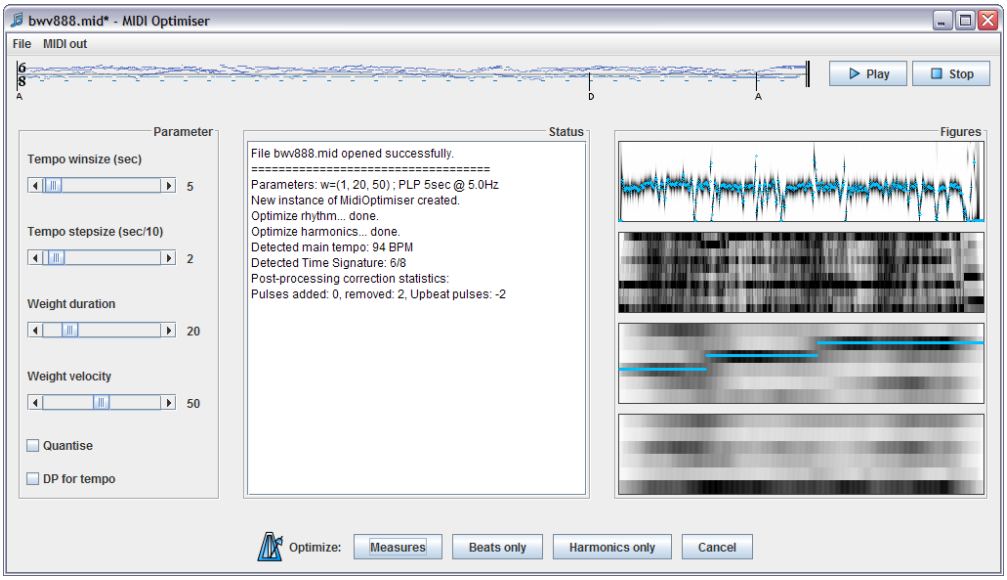


Abbildung 5.11.: Java-Benutzerschnittstelle »MidiOptimizer« mit geöffneter MIDI-Datei (BWV 888).

Anzahl der Fehler in der Folge der Grundschläge verringern als auch durch Miteinbeziehung weiterer Informationen die Aussagekraft der Betonungswerte erhöhen (Abschnitt 5.6.2).

Da die Ausgabe unseres Verfahrens wieder eine MIDI-Datei ist, kann es durch Vorverarbeitung von MIDI-Dateien ohne musikalisch sinnvolle Zeitinformationen generell in Kombination mit jedem MIDI-fähigen Quantisierungsprogramm verwendet werden. Hierbei bleibt die physikalische Zeitachse zumeist unverändert, sodass es weiterhin in Kombination mit anderen Ansätzen zur rhythmischen Transkription genutzt werden kann. Die Herleitung einer musikalischen Zeitachse ohne weitere Quantisierungsschritte ist auch für die Echtzeit-Interaktion mit MIDI-Synthesizern sinnvoll, zum Beispiel als eine Variation des in [29] vorgestellten Systems.

Aufgrund seines schrittweisen Aufbaus kann unser Verfahren auch leicht durch Einbeziehen zusätzlicher rhythmischer oder harmonischer Aspekte erweitert werden. Beispielsweise werden in [142] Informationen über Akkordwechsel zum Erkennen von Taktanfängen verwendet. Als eine weitere Möglichkeit zu weiteren Verbesserungen mag die Kombination mit musterbasierten Ansätzen wie [188] erscheinen, bei denen eine Liste rhythmischer Figuren als Schablone verwendet wird. Hier könnte das Verfahren zum Beschreiben selten vorkommender Muster verwendet werden, die nicht von den Rhythmen in der Liste abgedeckt werden.

5.6.1. Graphische Benutzeroberfläche

Das vorgestellte Verfahren wurde vollständig in Java implementiert. Dies ermöglichte eine verhältnismäßig einfache Erstellung einer graphischen Benutzeroberfläche zum Öffnen und Speichern von MIDI-Dateien sowie die Kontrolle von Programmdurchläufen mit und ohne

Nachverarbeitung jeweils mit einstellbaren Parametern, siehe Abbildung 5.11. Weiterhin bietet die Benutzerschnittstelle die Möglichkeit, die geöffnete MIDI-Datei auf einem beliebigen MIDI-Gerät abzuspielen. Die von dem Programm erzeugten S-MIDI-Dateien können ebenfalls direkt angehört werden, hierzu werden die ermittelten Schläge und Taktanfänge durch eine zusätzliche Schlagwerkstimme sonifiziert. Im Dateimenü kann weiterhin die Funktion zum Export der aktuell geöffneten MIDI-Datei in eine kommentierte Textdatei ähnlich zu Codebeispiel 5.1 aufgerufen werden.

Die linke Spalte beinhaltet Schieberegler zum Einstellen der im Programm verwendeten Parameter. Hierzu zählen neben der Fensterlänge für die Berechnung der PLP-Kurve auch die Gewichtungsparameter für die Berechnung der Aktivitätskurve (Abschnitt 5.4.1). Weiterhin kann die Schrittweite für die PLP-Kurve gesetzt werden. Im unteren Bereich können neben der in Abschnitt 5.4.3 vorgestellten Quantisierungsmethode auch einige der im folgenden Abschnitt vorgestellten algorithmischen Erweiterungen zur Korrektur spezifischer Probleme des Verfahrens aktiviert werden, wie etwa die Nutzung von dynamischer Programmierung für die Ermittlung der lokalen Tempoinformationen. Hierbei werden in Einzelfällen deutliche Verbesserungen gegenüber der vorab beschriebenen Standardmethode erreicht.

Der Platz in der Mitte wird vom Ausgabefenster für die durch das Hauptprogramm erzeugten Statusmeldungen verwendet. Das Programm gibt bei jedem Durchlauf die verwendeten Parametereinstellungen an, das erkannte globale Tempo T_0 und die geschätzte Taktart. Ähnlich zu Tabelle 5.1 wird weiterhin die Anzahl der in der Nachverarbeitung hinzugefügten und entfernten Schläge angegeben sowie die Anzahl im Rahmen der Auftaktkorrektur erfolgten Änderungen.

In der rechten Spalte werden die Ergebnisse analog zu Abbildung 5.7 graphisch dargestellt: Der erste Plot zeigt das für die PLP-Berechnung verwendete Tempogramm $\mathcal{T}^{\text{fine}}$ mit Hervorhebung der maximalen Indizes, welche die lokalen Tempi bestimmen. Im nächsten Plot sind die relevanten Zeilen aus der Kurzzeit-Autokorrelationsmatrix $\mathcal{A}$ dargestellt. Der dritte Plot zeigt das im Rahmen der Nachverarbeitung berechnete Stressgramm mit hervorgehobener Kongruenzklasse der geschätzten Taktanfänge. Der letzte Plot schließlich zeigt das Stressgramm nach Durchführung der Korrekturen. Im Optimalfall zeigt die unterste Zeile durchgehend maximale Energie. Somit kann der Benutzer visuell abschätzen, ob das Ergebnis zufriedenstellend ist oder ein weiterer Programmdurchlauf mit anderen Parametern durchgeführt werden sollte.

5.6.2. Algorithmische Verbesserung

Die in diesem Abschnitt vorgestellten Ansätze dienen zur Behebung konkreter Schwierigkeiten, die bei Verwendung des vorgestellten Systems bei einzelnen Stücken aufgetreten sind. Die einzelnen Techniken werden anhand von Beispielen motiviert und qualitativ evaluiert, wobei zu beachten ist, dass diese Techniken anhand konkreter Stücke entwickelt wurden und nicht für den generellen Einsatz optimiert sind. Sie basieren sowohl auf der Nutzung musikalischen Wissens als auch auf technischen Eigenschaften unseres Ansatzes.

Alternative Bestimmung des lokalen Tempos

Wie in Abschnitt 5.4.1 ausführlich dargestellt wurde, erfolgt die Bestimmung des lokal vorherrschenden Tempos in zwei Stufen. Nach Ermittlung eines globalen Tempos T_0 wird zur Vermeidung von Sprüngen in Tempovielfache ein Fenster zulässiger Tempi von der Größe einer Tempooktave um T_0 gewählt. Das konkrete lokale Tempo wird dann mittels spaltenweiser Maximumsbildung ermittelt. Bei der ursprünglichen Modellierung ist T_0 das geometrische Mittel des Fensters. Diesem Modell liegt die Annahme zugrunde, dass Beschleunigung und Verlangsamung in selber Häufigkeit und Intensität auftritt. Wenn nun durch zusätzliche Informationen über das Musikstück stärkere Ritardandi zu erwarten sind, bieten sich alternative Tempooktaven wie etwa $\left[\frac{3}{5} \cdot T_0, \frac{6}{5} \cdot T_0\right] \cap \mathbb{N}$ an. Im diskutierten Bach-Beispiel führt dies zu einem Wegfall der ersten Sprungstelle beim in Abbildung 5.7f illustrierten Stressgramm.

Eine andere Möglichkeit zur Bestimmung der lokalen Tempowerte bei gleichzeitiger Robustheit gegenüber Fehlern der Tempooktave besteht darin, einen Pfad maximalen Tempos mittels dynamischer Programmierung zu berechnen, vergleiche auch die in [157] vorgestellte HMM-ähnliche Vorgehensweise. Wir verwenden hier das in Abschnitt 5.4.2 verwendete Verfahren zur Bestimmung der Taktanfangs-Kongruenzklasse zur Ermittlung eines Pfades durch das Tempogramm. Die in Abschnitt 5.6.1 vorgestellte Benutzerschnittstelle erlaubt die Auswahl zwischen diesem Ansatz sowie der im ursprünglichen Verfahren vorgestellten Verwendung des Maximalindex' innerhalb der betrachteten Tempooktave.

Betrachtung mehrerer Tempokurven

Viele Interpretationen von Musikstücken weisen Passagen mit kleinen oder gar keinen Tempovariationen auf. An diesen Stellen bietet es sich an, die Robustheit des Systems zur Grundschlag-Erkennung zu erhöhen, indem längere Fensterlängen bei der Berechnung des Tempogramms (vgl. Abschnitt 5.4.1) verwendet werden, wodurch eine höhere Auflösung des lokal vorherrschenden Tempos erzielt wird. Andererseits verlangen sporadisch auftretende, kurze rhythmische Variationen die Verwendung kleiner Fensterlängen, die maximal ein paar Schläge betragen dürfen.

Für eine kurze Passage aus dem bereits diskutierten Barockstück BWV 888 sind in Abbildung 5.12 Tempogramme sowohl auf einem groben Niveau von geschätzten 36 Schlägen (5.12a) als auch auf einer feinen zeitlichen Auflösung von etwa 5 Schlägen dargestellt (5.12b). Die aus dem groben Tempogramm abgeleiteten PLP-Kurven bilden die Periodizität des Schlagrasters sehr gut ab, verfehlen allerdings einige kurzzeitige Tempoabweichungen wie das Ritardando am Ende der im Beispiel diskutierten Passage, siehe Abbildung 5.12c. Zeitlich feiner aufgelöste Tempogramme können diese Phänomene deutlich besser abbilden, allerdings ist hier eine starke Anpassung an die Aktivitätskurve zu beobachten, die bei Vorschlagnoten oder allgemeiner synkopischen Rhythmen zu ungewollten Verschiebungen des Grundrasters führt.

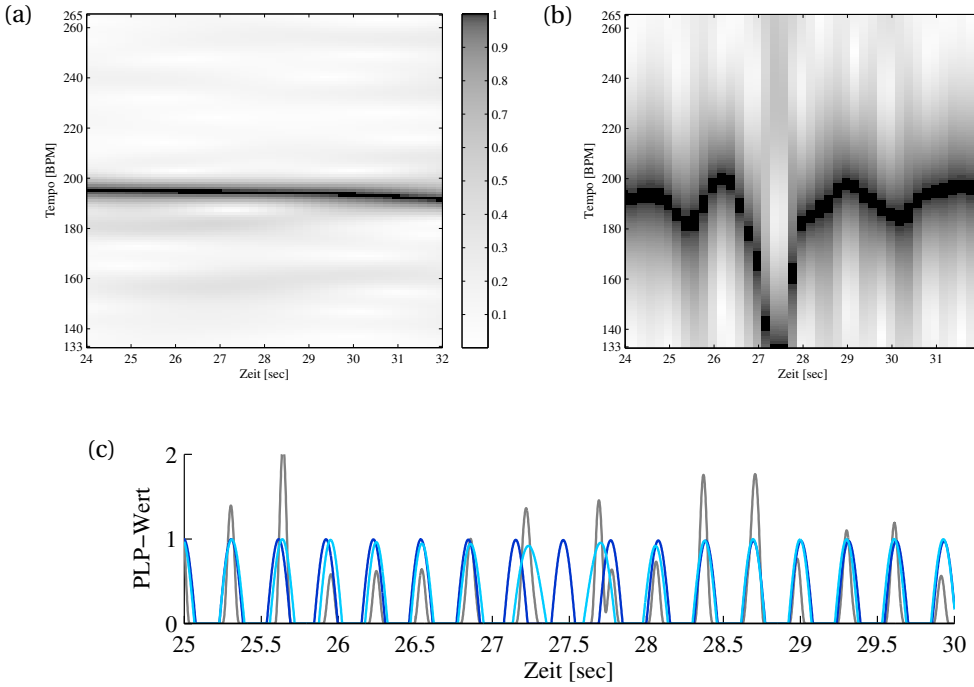


Abbildung 5.12.: Berechnung von PLP-Kurven für verschiedene Fensterlängen. **(a)** Tempogramm bei einer groben Zeitauflösung von 36 Schlägen, **(b)** Tempogramm bei einer feinen Auflösung von 5 Schlägen, **(c)** Aktivitätskurve (grau) und abgeleitete PLP-Kurven für die obigen Fensterlängen von 36 (dunkelblau) und 5 (hellblau) Schlägen. Man beachte, dass bei Verwendung des groben Fensters das leichte Ritardando zwischen 27 und 28 Sekunden nicht gefunden wird.

Daher schlagen wir ein Verfahren für einen Kompromiss aus Periodizität und moderatem Anpassen an die Aktivitätskurve vor, indem mehrere Tempogramme basierend auf verschiedenen Fensterlängen[8] betrachtet werden. Durch mehrfaches Anwenden der vorgestellten Methode können wir für jede Fensterlänge eine PLP-Kurve, eine Autokorrelationsmatrix sowie ein Stressgramm berechnen. Durch starkes Glätten der PLP-Kurve erhalten wir für jede Auflösung eine »Konfidenzkurve«, welche den durchschnittlichen PLP-Wert der musikalischen Zeitpunkte bestimmt.

Weiterhin stellt die Ermittlung der Inkonsistenzen in den verschiedenen Folgen möglicher Schlagkandidaten und die Extraktion der dazugehörigen Fehlerregionen weitere Informationen über die Güte der einzelnen Fensterfunktionen bereit. Wird etwa bei der Verwendung eines speziellen Fenster an einer Stelle ein Sprung in der Zuordnung der Kongruenzklassen erkannt und bei Verwendung eines anderen nicht, so kann daraus abgeleitet werden, dass diese Stelle mit dem anderen Fenster besser beschrieben werden kann. Das Stressgramm

[8] In unserer Implementierung verwenden wir Fensterlängen von 5, 12 und 36 Schlägen.

und die daraus ermittelten Korrekturstellen können somit als ein unabhängig von Vorwissen nutzbares Kriterium für die Güte eines Parameters verwendet werden.

5.7. Anwendungsbeispiel: Nintendo Sound Format

Die Musik in Videospielen stellt ein sehr junges Genre dar. Trotz zahlreicher Parallelen zur Filmmusik wird die Videospielmusik aufgrund der höheren Interaktivität der Spiele durchaus eigenständig betrachtet, vgl. [25]. Aufgrund der technischen Beschränkungen der Spielekonsolen vor der Jahrtausendwende lag die Musik für deren Spiele weitgehend in symbolischer Form vor und wurde von der Konsole mittels Wellengeneratoren in Audiosignale umgesetzt, siehe [33] für weitere Informationen zur technischen Umsetzung und [20] für einen Überblick über diese Konsolen.

Der sich durch die Verwendung dieser Wellengeneratoren ergebende einzigartige Klang führte zur Etablierung des neuen Musikgenres der *Chiptunes*. Dieser Ausdruck bezeichnet speziell für diese Art der Klangerzeugung neu komponierte Musikstücke, die auf künstlerische Art und Weise die Beschränkungen der Technik als Ausdrucksmittel nutzt, siehe hierzu [33, 143]. Andererseits haben sich die Musikstücke dieser »klassischen« Spiele selbst zu einem Phänomen der Populärkultur entwickelt. So finden seit einigen Jahren weltweit zumeist ausverkaufte Konzerte in renommierten Häusern statt, bei denen bekannte Orchester Arrangements von Musikstücken spielen, die ursprünglich für einfache Wellengeneratoren zur Untermalung eines Videospiels geschrieben wurden [137].

In diesem Abschnitt soll unser Verfahren zur Schätzung der musikalischen Zeitachse als ein Baustein eines Verfahrens zur Extraktion der symbolischen Musikinformationen aus den Spielen genutzt werden, um so die ursprüngliche Form dieser Musik in Notenschrift lesbar zu machen. Hierbei beschränken wir uns mit dem »Nintendo Entertainment System« (NES, Abbildung 5.13a) auf eine sehr populäre Konsole der japanischen Firma Nintendo, auf der viele bekannte Videospieltitel ihre Premiere feierten. Durch Ausreizen der technischen Möglichkeiten und Komposition berühmter Soundtracks wie etwa die Themen zu »Super Mario Bros.« (Nintendo, 1985) und »The Legend of Zelda« (Nintendo, 1986) durch Koji Kondo oder die Musik zur »Final Fantasy«-Serie (Squaresoft, erstmals 1987) durch Nobou Uematsu kann das NES als Beginn der Videospielmusik angesehen werden [51]. Andere Konsolen dieser Generation nutzen ähnliche Verfahren zum Speichern und zur Wiedergabe musikalischer Informationen, sodass die grundlegende Herangehensweise ähnlich sein wird.

Das NES erschien 1985 in den USA und Europa als leichte Modifikation der bereits 1983 für den japanischen Markt entwickelten Konsole »Famicon« und ist mit über 60 Millionen verkauften Einheiten eine der weltweit erfolgreichsten Spielekonsolen [33]. Seine CPU stellt eine um Sound- und Musikgeneratoren erweiterte Variante des weitverbreiteten 8-Bit-Mikroprozessors

(a)

(b)

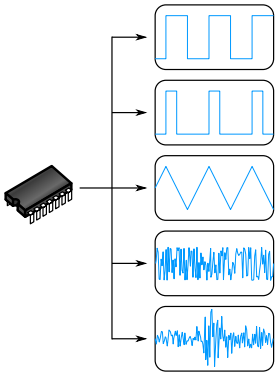


Abbildung 5.13.: Klangerzeugung der Videospielkonsole »Nintendo Entertainment System«. **(a)** Konsole mit einem Spielkontroller (Bild: Public Domain, [206]), **(b)** die fünf zur Klangerzeugung verwendeten Kanäle.

MOS 6502[9] dar, welche von RICOH unter der Bezeichnung 2A03 (für die NTSC-Version) und 2A07 (für die PAL-Version) produziert wurde [208].

Zur Klangerzeugung werden beim NES fünf (in der japanischen Version vier) Kanäle, verwendet, vgl. Abbildung 5.13b. Hierbei dienen zwei Kanäle zur Erzeugung von Rechteckschwingungen mit einstellbarer Pulsbreite (12,5%, 25%, 50% oder 75%), die Abbildung zeigt 50% auf Kanal 1 und 25% auf Kanal 2. Die verschiedenen Pulsbreiten werden aufgrund ihrer Obertonspektren zur Realisierung unterschiedlicher Klangfarben verwendet. Für Lautstärkepegel stehen 16 mögliche Werte zur Verfügung, für die Angabe der Frequenz 2048 mögliche Werte aus dem Bereich zwischen 54 Hz und 28 kHz. Ein weiterer Kanal wird für eine Dreiecksschwingung mit fester Lautstärke und Frequenzen zwischen 27 Hz und 56 kHz verwendet. Zur Darstellung perkussiver Elemente und Spielgeräusche existiert ein Rauschgenerator, der weißes Rauschen in 16 Lautstärkegraden mit 16 vorprogrammierten Frequenzen erzeugt. Der letzte (in der japanischen Version nicht vorhandene) Kanal kann zum Abspielen von 6-Bit-Audiosignalen mit Samplingraten zwischen 4,2 kHz und 33,5 kHz genutzt werden, wurde aber in der Praxis kaum verwendet. Für Details siehe [33, 190, 208].

Nach [25] wurden üblicherweise in den für das NES entwickelten Spielen die drei tonalen Kanäle für Melodiestimme, Begleitung und Bass verwendet. Wegen der mangelnden Variabilität in Lautstärke und Klangfarbe diente die Dreiecksschwingung oftmals zur Erzeugung der Bassbegleitung. In den anderen Kanälen hingegen konnten durch Modulation der Kurven musikalische Effekte wie Vibrato, Tremolo, Glissando, Echo-Effekte usw. realisiert werden. Teilweise wurden auch alle drei Kanäle gleichzeitig zum Abspielen von Akkorden verwendet.

Um diese Musikinformation aus den Spielen zu extrahieren, wurde vom Hobby-Programmierer *Kevin Horton*[10] das »NES Sound Format« (NSF) [207] entwickelt, welches aus den extrahierten

[9] Mikroprozessoren dieser Familie wurden u. a. von Apple, Atari und Commodore eingesetzt.
[10] `http://kevtris.org`

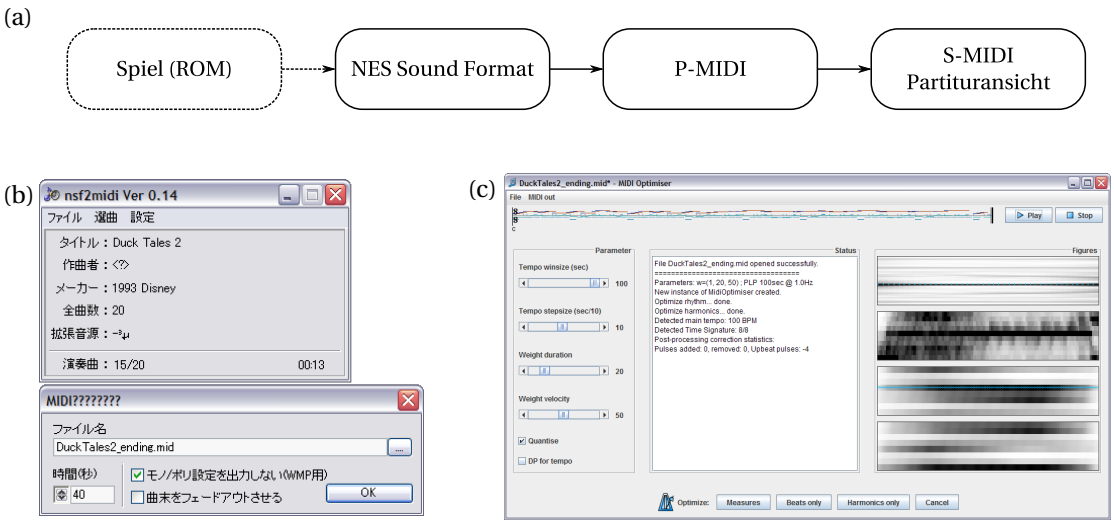


Abbildung 5.14.: **(a)** Erzeugung von musikalisch sinnvollen Partiturdaten aus NES-Spielen. **(b)** Programm `nsf2midi` mit Export-Dialog zur Konvertierung von NSF-Dateien nach MIDI. **(c)** Die daraus entstandene MIDI-Datei geöffnet in unserem Programm und mit geeigneten Parametern verarbeitet.

Sound- und Musikinformationen aus dem Binärcode der Spiele und einem hinzugefügten Dateikopf mit Meta-Informationen besteht. Aufgrund der weiten Verbreitung des NES ist dieses Format heutzutage auch bei Komponisten aus der Chiptune-Musikszene sehr beliebt. Für die technischen Spezifikationen dieses Formats siehe [73].

In Abbildung 5.14a ist die Abfolge der Bearbeitungsschritte für die Extraktion musikalischer Informationen aus NES-Spielen illustriert. Abhängig vom jeweiligen Spiel kann die Extraktion der musikalischen Informationen aus dem Code relativ einfach bis technisch sehr aufwendig sein, weswegen bislang kein Programm zum vollständig automatischen Lösen dieser Aufgabenstellung existiert. Hierzu besteht aus Sicht der Nutzer auch kein Anlass, da im Internet sehr viele NSF-Dateien leicht zu finden sind.

Für den nächsten Schritt, die Umwandlung der NSF-Dateien in das MIDI-Format, existiert ein Hilfsprogramm namens `nsf2midi`[11], womit NSF-Dateien geöffnet, die verschiedenen Soundclips eines Spiels angewählt und einzeln als MIDI-Dateien exportiert werden können, siehe Abbildung 5.14b. In diesem Beispiel wurde aus der NSF-Datei zum Spiel »DuckTales 2« (Disney/Capcom, 1993) das 15. von 20 Stücken geöffnet und die ersten 40 Sekunden als MIDI-Datei exportiert. Bedingt durch die technische Umsetzung der Abspielfunktion des NES ist keine musikalisch sinnvolle symbolische Zeitachse vorhanden, sondern es sind lediglich physikalische Zeitinformationen nutzbar, vgl. Abbildung 5.15a.

Auf die so erzeugte MIDI-Datei wird nun das in den vorherigen Abschnitten beschriebene

[11] `http://gigo.retrogames.com/download.html#nsf2midi`

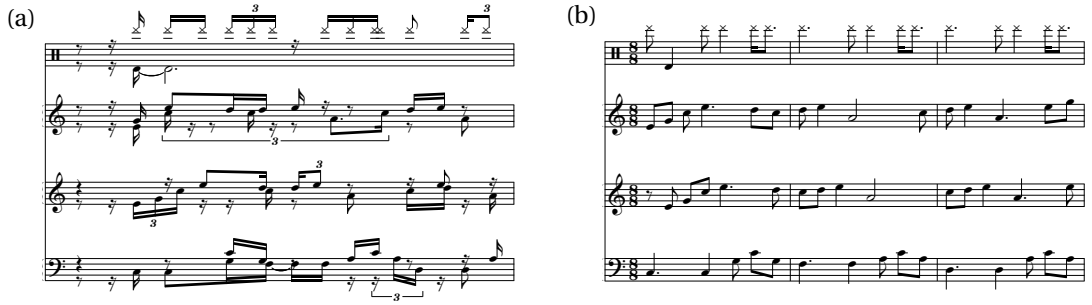


Abbildung 5.15.: Beginn der Schlussmusik aus dem Spiel »DuckTales 2« automatisch gesetzt vom Notensatzprogramm *Sibelius* mit Quantisierung auf Sechzehntelnoten. Zur besseren Lesbarkeit wurde in der 4. Zeile der Bass-Schlüssel manuell gesetzt. Die Noten wurden aus folgenden Eingabedaten erzeugt: **(a)** Die in `nsf2midi` erzeugte P-MIDI-Datei, **(b)** die automatisch umgewandelte S-MIDI-Datei.

Verfahren zur Schätzung der musikalischen Zeitachse von P-MIDI-Dateien und automatische Konvertierung in eine semantisch angereicherte S-MIDI-Datei angewendet. Bei diesen Spielen können wir aus technischen Gründen zumeist von einem konstanten Tempo ausgehen, was im beschriebenen Verfahren durch große Fensterlängen realisiert wird. In Abbildung 5.14c sieht man, dass zu allen Zeitpunkten identische Tempowerte gewählt worden sind, was zu einer robusten Schätzung des Grundschlags führt. Durch das in Abschnitt 5.4.2 beschriebene Nachverarbeitungsverfahren wurden lediglich die Pausen am Beginn des Stücks entfernt. Die so gewonnene S-MIDI-Datei wurde anschließend in das Notensatzprogramm Sibelius importiert, siehe Abbildung 5.15b.

Im Vergleich zeigt sich, dass die fehlende Semantik der symbolischen Zeitachse in der P-MIDI-Datei zu einer zu groben Quantisierung geführt hat, da viele Läufe zu Clustern zusammengezogen worden sind. Weiterhin wurde von Sibelius versucht, den musikalischen Grundschlag durch Triolen zu approximieren, was ebenfalls zur Unlesbarkeit des Notenbildes beiträgt. Das übersichtliche und aufgeräumte Notenbild der S-MIDI-Datei hingegen ist nicht nur für Menschen leichter nutzbar, sondern erlaubt auch das Erkennen vorher nicht sichtbarer musikalischer Strukturen, wie etwa in der 3. Zeile das um eine Achtelnote versetzte Echo der Melodie aus der 2. Zeile. Lediglich der in Achteln geschätzte Grundschlag ist eine Granularitätsstufe feiner als das für Menschen prägnantere Raster in Viertelnoten.

5.8. Zusammenfassung und Ausblick

In diesem Kapitel haben wir einen Algorithmus zur Schätzung einer musikalisch sinnvollen Zeitachse aus einer MIDI-Datei mit rein physikalischen Zeitinformationen und zur Umwandlung in eine semantisch angereicherte partiturnahe MIDI-Datei vorgestellt. Dieses Verfahren optimiert eine geschätzte Folge von Schlagzeitkandidaten durch Einfügen fehlender und Ent-

fernen fälschlich erkannter Kandidaten, um so ein möglichst global konsistentes Taktraster zu erhalten.

Da die Ausgabe der vorgestellten Methode wiederum eine MIDI-Datei ist, kann das Verfahren in Kombination mit jeder MIDI-Quantisierungssoftware als Vorverarbeitungsschritt verwendet werden. Insbesondere bleiben die physikalischen Zeitinformationen unverändert, sodass unser Ansatz ebenfalls mit Verfahren zur Rhythmus-Transkription verbunden werden kann. Das Ableiten einer musikalischen Zeitachse ohne Quantisierung ist beispielsweise für die Entwicklung von Programmen für die Echtzeit-Interaktion mit MIDI-Synthesizern sinnvoll [29], die durch Benutzerschnittstellen zur automatischen Erkennung von Dirigierbewegungen [129, 130] zu einer intuitiven und reizvollen musikalischen Erfahrung beitragen können, vgl. auch [12, 94, 95, 151] für einige Beispiele ähnlicher Systeme.

Durch seine Allgemeinheit kann unser Verfahren auch leicht um weitere rhythmische oder harmonische Aspekte erweitert werden. So werden beispielsweise in [142] Informationen über Akkordwechsel zum Schätzen von Taktwechseln verwendet. Auch eine Verbindung mit Mustererkennungssystem wie [188] erscheint sinnvoll, die eine maschinell gelernte Liste rhythmischer Schablonen verwendet. Unser Verfahren mag an dieser Stelle zum Erfassen der selten vorkommenden und daher in solchen Listen fehlenden Muster eingesetzt werden.

A. Ergänzende Informationen zur Strukturanalyse

A.1. Vergleich der SALAMI-Annotationen

Wie bereits in Abschnitt 2.7 diskutiert, weisen die Evaluationsmaße zur Bewertung der Güte von Strukturierungsverfahren nur eine begrenzte Aussagekraft auf. Um eine Referenz für unsere Ergebnisse zu bekommen, haben wir überprüft, welche Evaluationsmaße wir für den Vergleich zweier Annotationen desselben Stückes erhalten, wenn diese von Menschen statt von Maschinen angefertigt wurden. Dies erlaubt uns eine grobe Orientierung, welche Abweichung von dem optimalen Evaluationswert 1 noch im Rahmen der musikalischen Interpretation liegt und welche Abweichung als sicheres Indiz für ein unzureichendes Verfahren angesehen werden kann.

Für dieses Experiment haben wir aus dem öffentlich zugänglichen Teil des SALAMI-Datensatzes diejenigen Stücke ausgewählt, die jeweils von zwei verschiedenen Personen[1] annotiert worden sind. Jede Person hat dabei sowohl die grobe Struktur (Segmentbezeichnungen in Großbuchstaben) als auch die Feinstruktur (Segmentbezeichnungen in Kleinbuchstaben) berücksichtigt. Wir haben dies in zwei separate Auswertungen aufgeteilt. Jede dieser Auswertungen berechnet die Evaluationswerte Precision $\boldsymbol{P}$, Recall $\boldsymbol{R}$ und F-measure $\boldsymbol{F}$ wie sie in Abschnitt 2.7 definiert worden sind. Nach unserer Konvention stellt die erste Annotation die Referenz dar, gegen welche die zweite ausgewertet wird.

Da für die paarweisen Evaluationswerte die Bezeichnungen zu jeweils denselben Zeitpunkten miteinander verglichen werden, haben wir diese Zeitpunkte äquidistant mit einer Auflösung von 5 Hz gewählt. Somit können keine Strukturen einer Länge unter 200 ms gefunden werden, was für Strukturannotationen keine Auswirkungen haben sollte.

Die Ergebnistabellen zeigen die Durchschnittswerte nach musikalischen Stilrichtungen (engl. *genres*), die Anzahl der Stücke pro Genre ist ebenfalls angegeben. Zur besseren Übersicht zeigen wir durch horizontale Linien die grobe Einteilung der Anbieter des Datensatzes in die vier Hauptgenres *Classical, Jazz, Popular* und *World* sowie die Aufnahmen aus dem *Live Music Archive* an. Die erzielten Werte sind ähnlich zu den Ergebnissen des Vergleichs aus [183], der auf dem kompletten Datensatz durchgeführt wurde.

[1] Insgesamt haben 8 Teilnehmer eines Graduiertenstudiengangs Musiktheorie oder Komposition die Annotationen für die Stücke angefertigt [183].

Grobstruktur

Genre	#Songs	*F*	*P*	*R*
Classical - 20th Century Classical	10	0,689	0,758	0,744
Classical - Baroque	16	0,815	0,876	0,812
Classical - Classical	11	0,65	0,698	0,663
Classical - Renaissance & Med	12	0,846	0,858	0,859
Classical - Romantic	16	0,696	0,708	0,787
Blues - Contemporary Blues	6	0,739	0,868	0,687
Blues - Country Blues	7	0,855	0,904	0,851
Blues - Urban Blues	6	0,77	0,839	0,777
Jazz - Acid Jazz	7	0,584	0,618	0,646
Jazz - Avant-Garde Jazz	6	0,794	0,832	0,778
Jazz - Bebop	6	0,606	0,743	0,658
Jazz - Cool Jazz	6	0,815	0,847	0,88
Jazz - Dixieland	6	0,761	0,827	0,78
Jazz - Hard Bop	4	0,906	0,927	0,894
Jazz - Latin Jazz	7	0,736	0,765	0,788
Jazz - Post-Bop	6	0,81	0,805	0,84
Jazz - Soul Jazz	6	0,756	0,906	0,677
Jazz - Swing	6	0,731	0,671	0,841
R B - Contemporary R B	7	0,829	0,805	0,873
R B - Funk	6	0,76	0,708	0,854
R B - Gospel	7	0,836	0,843	0,873
R B - Rock & Roll	6	0,699	0,858	0,688
R B - Soul	7	0,792	0,906	0,781
Alternative Pop & Rock	8	0,632	0,631	0,673
Country	8	0,722	0,722	0,77
Dance Pop	7	0,78	0,861	0,756
Electronica	7	0,69	0,584	0,911
Hip Hop & Rap	6	0,743	0,718	0,84
Humour	7	0,794	0,814	0,849
Instrumental Pop	7	0,789	0,798	0,853
Modern Folk - Alternative Folk	8	0,78	0,75	0,888
Modern Folk - Singer & Songwriter	6	0,778	0,77	0,812
Reggae	7	0,69	0,686	0,747
Rock - Alternative Metal & Punk	6	0,727	0,755	0,731
Rock - Classic Rock	7	0,82	0,749	0,925
Rock - Metal	7	0,768	0,776	0,788
Rock - Roots Rock	8	0,781	0,738	0,867
World - African	6	0,796	0,879	0,785
World - Americas	3	0,714	0,848	0,703
World - Arabic	3	0,807	0,754	0,899
World - Asian	5	0,701	0,828	0,653
World - Balkan	4	0,627	0,572	0,753
World - Calypso	1	0,695	0,719	0,672
World - Celtic	6	0,661	0,886	0,579

Fortsetzung auf der folgenden Seite

Genre	#Songs	*F*	*P*	*R*
World - Chanson	4	0,737	0,792	0,758
World - Cuban	6	0,809	0,909	0,756
World - European	5	0,818	0,896	0,83
World - Flamenco	6	0,7	0,797	0,64
World - Fusion	5	0,719	0,799	0,735
World - Gypsy	4	0,87	0,933	0,825
World - Indian	3	0,655	0,804	0,63
World - Klezmer	3	0,918	0,974	0,88
World - Latin American	4	0,713	0,77	0,694
World - Mixed Traditional	4	0,808	0,872	0,828
World - Tango	4	0,763	0,762	0,806
World - US-Traditional	5	0,641	0,697	0,648
Live Music Archive	141	0,679	0,672	0,753
TOTAL	498	0,731	0,756	0,773

Feinstruktur

Genre	#Songs	*F*	*P*	*R*
Classical - 20th Century Classical	10	0,664	0,757	0,628
Classical - Baroque	16	0,544	0,533	0,693
Classical - Classical	11	0,559	0,502	0,723
Classical - Renaissance & Med	12	0,736	0,778	0,74
Classical - Romantic	16	0,588	0,565	0,711
Blues - Contemporary Blues	6	0,611	0,714	0,611
Blues - Country Blues	7	0,729	0,701	0,775
Blues - Urban Blues	6	0,812	0,838	0,817
Jazz - Acid Jazz	7	0,543	0,627	0,601
Jazz - Avant-Garde Jazz	6	0,637	0,88	0,564
Jazz - Bebop	6	0,656	0,603	0,791
Jazz - Cool Jazz	6	0,562	0,607	0,61
Jazz - Dixieland	6	0,674	0,643	0,74
Jazz - Hard Bop	4	0,791	0,78	0,865
Jazz - Latin Jazz	7	0,601	0,729	0,551
Jazz - Post-Bop	6	0,644	0,631	0,67
Jazz - Soul Jazz	6	0,595	0,633	0,641
Jazz - Swing	6	0,601	0,567	0,658
R B - Contemporary R B	7	0,74	0,762	0,767
R B - Funk	6	0,785	0,786	0,813
R B - Gospel	7	0,736	0,745	0,789
R B - Rock & Roll	6	0,683	0,737	0,704
R B - Soul	7	0,662	0,805	0,655
Alternative Pop & Rock	8	0,547	0,508	0,753
Country	8	0,535	0,899	0,434
Dance Pop	7	0,595	0,576	0,842

Fortsetzung auf der folgenden Seite

Genre	#Songs	*F*	*P*	*R*
Electronica	7	0,424	0,434	0,818
Hip Hop & Rap	6	0,568	0,528	0,818
Humour	7	0,394	0,681	0,534
Instrumental Pop	7	0,592	0,869	0,531
Modern Folk - Alternative Folk	8	0,582	0,834	0,543
Modern Folk - Singer & Songwriter	6	0,33	0,641	0,417
Reggae	7	0,523	0,699	0,526
Rock - Alternative Metal & Punk	6	0,561	0,523	0,773
Rock - Classic Rock	7	0,665	0,657	0,814
Rock - Metal	7	0,627	0,634	0,683
Rock - Roots Rock	8	0,645	0,6	0,757
World - African	6	0,71	0,839	0,683
World - Americas	3	0,719	0,817	0,656
World - Arabic	3	0,551	0,792	0,524
World - Asian	5	0,458	0,771	0,468
World - Balkan	4	0,611	0,647	0,656
World - Calypso	1	0,456	0,623	0,36
World - Celtic	6	0,669	0,728	0,66
World - Chanson	4	0,684	0,821	0,649
World - Cuban	6	0,697	0,77	0,693
World - European	5	0,679	0,753	0,665
World - Flamenco	6	0,544	0,604	0,558
World - Fusion	5	0,598	0,517	0,792
World - Gypsy	4	0,735	0,658	0,844
World - Indian	3	0,541	0,753	0,474
World - Klezmer	3	0,763	0,903	0,662
World - Latin American	4	0,598	0,594	0,634
World - Mixed Traditional	4	0,733	0,747	0,817
World - Tango	4	0,584	0,689	0,594
World - US-Traditional	5	0,621	0,587	0,791
Live Music Archive	141	0,688	0,738	0,713
TOTAL	498	0,636	0,695	0,688

Vollständiger Vergleich

In den folgenden drei Tabellen haben wir die Auswertung der Annotationen gegeneinander gruppiert und nach Genres unterteilt. Die folgende Tabelle zeigt somit eine Zusammenfassung der beiden oberen Tabellen, bei denen zuerst die beiden Grobstrukturen miteinander verglichen wurden (Index cc), anschließend die beiden Feinstrukturen (ff).

Genre	#Songs	F_{cc}	P_{cc}	R_{cc}	F_{ff}	P_{ff}	R_{ff}
Live Music Archive	141	0,679	0,672	0,753	0,688	0,738	0,713
classical	65	0,744	0,783	0,779	0,611	0,615	0,701
jazz	112	0,763	0,813	0,786	0,668	0,71	0,698
popular	99	0,749	0,738	0,815	0,545	0,654	0,658
world	81	0,744	0,822	0,735	0,635	0,714	0,657
TOTAL	498	0,731	0,756	0,773	0,636	0,695	0,688

In der folgenden Tabelle haben wir die Grobstruktur des ersten Annotators verglichen mit der Feinstruktur des zweiten und vice versa.

Genre	#Songs	F_{cf}	P_{cf}	R_{cf}	F_{fc}	P_{fc}	R_{fc}
Live Music Archive	141	0,632	0,654	0,692	0,595	0,622	0,642
classical	65	0,517	0,424	0,839	0,535	0,782	0,459
jazz	112	0,555	0,497	0,777	0,534	0,783	0,454
popular	99	0,509	0,452	0,823	0,517	0,779	0,485
world	81	0,62	0,59	0,767	0,564	0,804	0,493
TOTAL	498	0,573	0,538	0,768	0,553	0,74	0,52

In der letzten Tabelle betrachten wir für jedes Stück individuell das maximale paarweise F-measure der vorgestellten vier möglichen Auswertungen.

Genre	#Songs	F_{max}	P_{max}	R_{max}
Live Music Archive	141	0,776	0,792	0,799
classical	65	0,802	0,837	0,803
jazz	112	0,811	0,838	0,822
popular	99	0,796	0,806	0,826
world	81	0,789	0,826	0,796
TOTAL	498	0,793	0,817	0,809

A.2. Die Annotationen des *Winterreise*-Datensatzes

In diesem Abschnitt stellen wir[2] die vollständigen Annotationen des Winterreise-Datensatzes vor und erläutern zu jedem Stück unsere Motivation für die angegebenen Segmentierungen. Für jedes Stück haben wir drei Segmentierungen angefertigt. Dabei zeigt die erste die Positionen der Gedichtstrophen, die zweite die Strukturannotationen und die dritte die Regionen homogener Tonarten. Die Zeitachse ist bei allen Stücken in Takten[3] angegeben.

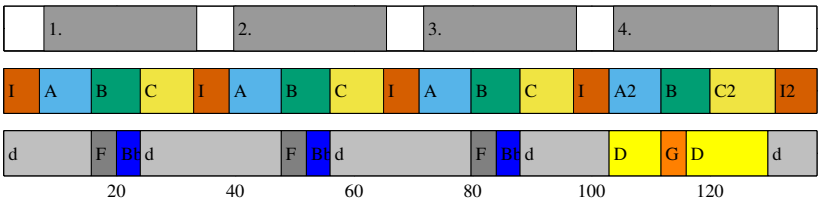
[2] Diese Texte sind in enger Zusammenarbeit mit der Musikwissenschaftlerin Polina Gubaidullina entstanden.

[3] Wir beziehen uns bei Takt- und Tonartenangaben auf die folgende Edition:
Herausgeber: Max Friedlaender (1852-1934), Gesänge für eine Singstimme mit Klavierbegleitung, Leipzig: Edition Peters, No. 20a, n. d. Plate 9023, verfügbar unter `http://imslp.org/wiki/File:Schubert_-_Winterreise.pdf`.

A. Ergänzende Informationen zur Strukturanalyse

Bei den Strukturannotationen werden Segmente mit unterschiedlichem musikalischen Material mit verschiedenen Großbuchstaben $A, B, \ldots$ gekennzeichnet. Eine angehängte Nummer bedeutet, dass das Material variiert wird, ein zusätzlicher Kleinbuchstabe gibt eine feinere Unterteilung an. Aufgrund der hierarchischen Struktur der Musikstücke nehmen wir in den begleitenden Texten gelegentlich Rückgriff auf die dann ebenfalls angegebene *musikalische Form*. Die Farbgebung für die Tonarten orientiert sich an Abbildung 2.7, wobei die Molltonarten durch pastellartige Farben angezeigt werden. Bereiche, die wir nicht einer speziellen Tonart zuordnen können, sind hier ausgelassen. Für eine ausführliche Beschreibung der Annotationen siehe Abschnitt 4.2.

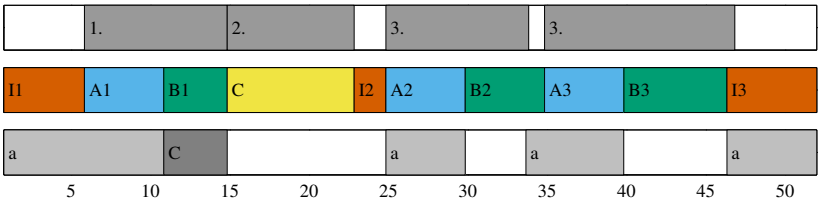
1. Gute Nacht



Variiertes Strophenlied in d-Moll mit vier Strophen und fünf rein instrumentalen Zwischenspielen.

Die Strophen sind in sich dreigeteilt, wobei die einzelnen Teile sich in der Tonart unterscheiden: d-Moll / F-Dur / d-Moll. Auf einer feineren Stufe lässt sich die zweite Hälfte der Mittelteile auch als B-Dur identifizieren. Eine Ausnahme bildet die letzte Strophe mit folgenden Tonarten: D-Dur / G-Dur / D-Dur. Die instrumentalen Zwischenspiele stehen alle in d-Moll und sind weitestgehend identisch.

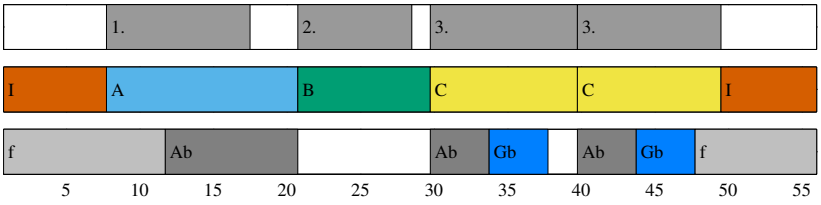
2. Die Wetterfahne



Variiertes Strophenlied in a-Moll mit drei Strophen und drei instrumentalen Zwischenspielen.

Aus harmonischer Sicht kann die erste Strophe in zwei Teile mit verschiedenen Tonarten unterteilt werden: a-Moll und C-Dur. Die zweite Strophe beginnt in G-Dur, ist danach harmonisch uneindeutig und enthält nicht-wiederholtes musikalisches Material. Sowohl die dritte Strophe als auch ihre geringfügig variierte Wiederholung können ebenfalls in zwei Teile untergliedert werden. Aus musikalischer Sicht können sie als Variation der ersten Strophe gesehen werden.

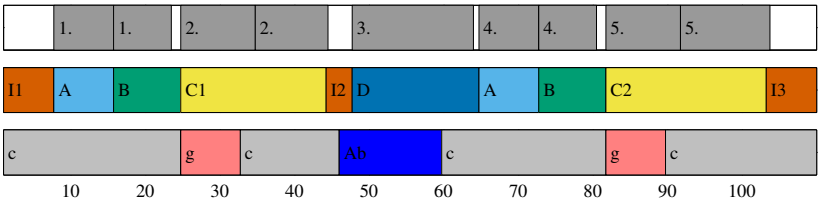
3. Gefror'ne Tränen



Durchkomponiertes Lied in f-Moll mit einem Vorspiel, drei Strophen und einem Nachspiel.

Die erste und die letzte Strophe stehen überwiegend in f-Moll und der parallelen Durtonart As-Dur. Die zweite Strophe ist harmonisch uneinheitlich, ihre erste Hälfte kann grob der Durdominante C-Dur, die zweite der Molldominant-Parallele Es-Dur zugeordnet werden. Dabei kommen die erste und die zweite Strophe jeweils nur einmal vor, die dritte Strophe dagegen wird einmal wiederholt. Durch eine tonale und rhythmische Verwandtschaft zwischen der ersten und der dritten Strophe (mit den Buchstaben A und C gekennzeichnet) könnte man das Lied auch mit einer ABA-Form annotieren. Die entsprechende Segmentierung würde folgendermaßen aussehen: $I\,A\,B\,A'\,A''\,I$.

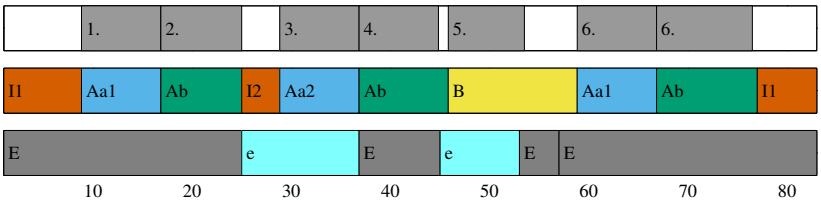
4. Erstarrung



Variiertes Strophenlied in c-Moll mit fünf Strophen.

Grobsegmentierung als dreiteilige Form ABA'. Der erste Teil besteht aus vier Segmenten $I1$, A, B und $C1$ der Feinsegmentierung. Dabei stellen A und B die Vertonung der ersten Strophe sowie ihrer Wiederholung dar. Die rein instrumentale Einleitung $I1$ sowie die gesungenen Teile A und B stehen in c-Moll. Das Segment $C1$ steht teilweise in der Tonart g-Moll. Der Mittelteil (D in der Feinsegmentierung) hat ein viertaktiges instrumentales Vorspiel und steht in As-Dur. Es handelt sich dabei um die Vertonung der dritten Strophe. Der letzte Teil stellt eine variierte Form des ersten Teils dar, wobei das Einleitungssegment $I1$ ganz entfällt. Des Weiteren enthält dieser Teil eine Coda ($I3$) in c-Moll. Die nach D folgenden Segmente A und B stimmen mit der vierten Strophe und ihrer Wiederholung überein. $C2$ stellt die Vertonung der fünften und letzten Strophe dar.

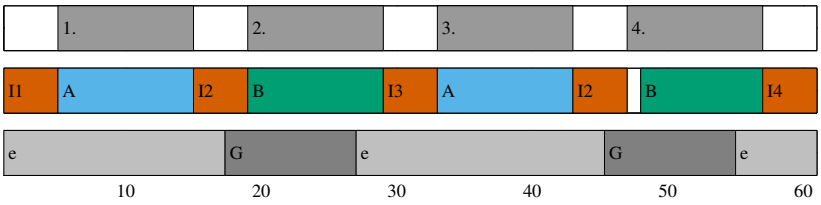
5. Der Lindenbaum



Variiertes Strophenlied in E-Dur mit sechs Strophen.

Grobsegmentierung als dreiteilige ABA'-Form. Der erste Teil enthält eine instrumentale Einleitung in E-Dur sowie zwei untereinander leicht variierte und durch ein instrumentales Zwischenspiel in e-Moll getrennte Segmente. Beide Segmente bestehen jeweils aus zwei Teilen: $Aa1$ und Ab bzw. $Aa2$ und Ab. Dabei stimmt $Aa1$ mit der ersten Textstrophe, Ab mit der zweiten, $Aa2$ mit der dritten und die Wiederholung von Ab mit der vierten Textstrophe überein. Das erste Segment $Aa1$ Ab steht in E-Dur. Das zweite Segment ist tonal zweigeteilt: $Aa2$ steht ebenso wie das vorangestellte Zwischenspiel $I2$ in e-Moll, Ab steht wieder in E-Dur. Der Mittelteil B besteht aus zwei Abschnitten: einem gesungenen in e-Moll und einer instrumentalen Überleitung in E-Dur zum letzten Teil des Liedes. Dieser Teil besteht musikalisch gesehen aus den Segmenten $Aa1$, Ab und $I1$ des ersten Teils und stellt die Vertonung der sechsten Strophe sowie ihrer Wiederholung dar.

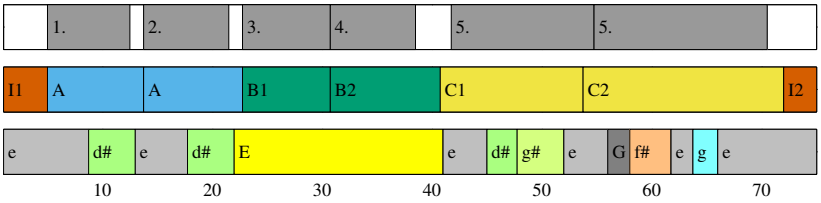
6. Wasserflut



Einfaches Strophenlied in e-Moll mit vier Strophen.

Die Segmentierung entspricht der Strophe neinteilung, wobei die einzelnen Textstrophen durch instrumentale Zwischenspiele getrennt sind. Die Zwischenspiele ($I1$–$I4$) sind nahezu identisch und unterscheiden sich lediglich geringfügig im jeweils letzten Takt. Die erste und die dritte Strophen bzw. die zweite und die vierte Strophen sind musikalisch gesehen jeweils identisch.

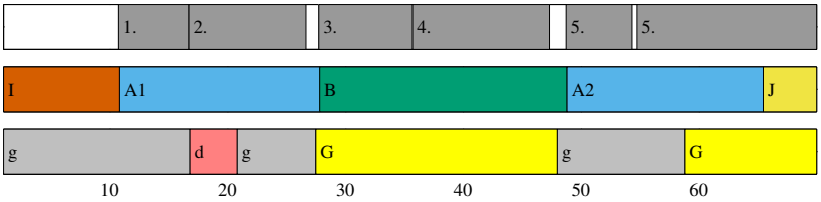
7. Auf dem Flusse



Variiertes Strophenlied in e-Moll mit fünf Textstrophen.

Musikalisch gesehen gibt es hier insgesamt drei Arten von Strophen: *A*, *B* und *C*. Diese werden jeweils zweimal hintereinander gespielt. Die Segmentierung stimmt weitestgehend mit der Stropheneinteilung überein, wobei die fünfte Textstrophe als einzige wiederholt wird. Die Wiederholung von *A* ist mit *A* identisch. Die Wiederholung *B*2 unterscheidet sich von *B*1 lediglich rhythmisch in der Begleitstimme (triolisch statt duolisch). Die Wiederholung *C*2 weist gegenüber *C*1 harmonische Unterschiede und einen anderen Schluss auf.

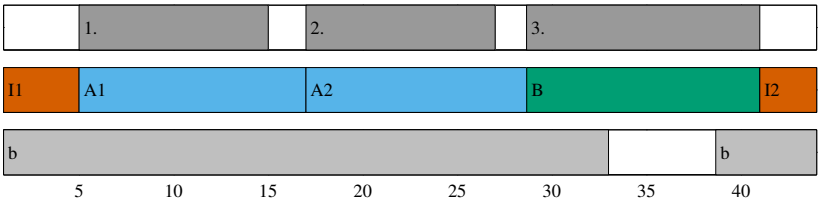
8. Rückblick



Variiertes Strophenlied in g-Moll mit fünf Textstrophen.

Grobsegmentierung als dreiteilige ABA'-Form. Die Segmentierung entspricht im Wesentlichen der Stropheneinteilung, wobei immer zwei Strophen auf ein musikalisches Segment fallen. Die fünfte Strophe wird wiederholt. Die Grobsegmente *A* (bestehend aus *I* und *A*1) und A' (*A*2 und *J*) unterscheiden sich harmonisch geringfügig voneinander. Außerdem hat A' einen anderen Schluss, der in unserer Segmentierung mit *J* bezeichnet wird.

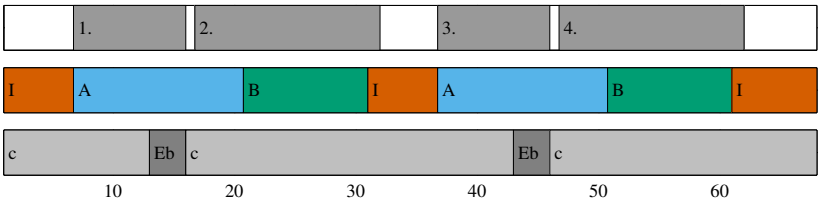
9. Irrlicht



Variiertes Strophenlied (Barform) in h-Moll mit drei Strophen.

Die erste und die zweite Strophe (*A*1 und *A*2) unterscheiden sich geringfügig in der Gesangsmelodie. Die dritte Strophe setzt sich harmonisch sowie melodisch deutlich von den ersten beiden Strophen ab. Das instrumentale Nachspiel stellt eine Variation des Vorspiels dar.

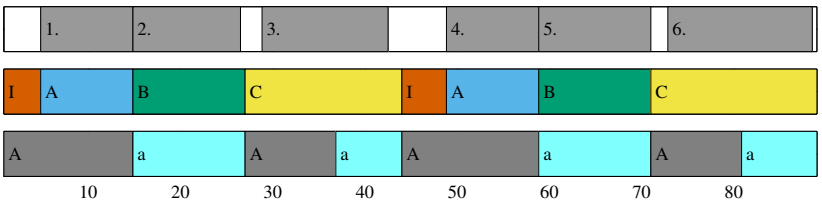
10. Rast



Variiertes Strophenlied in c-Moll mit vier Strophen.

Die Segmentierung entspricht weitestgehend der Strophe neinteilung. Es wird keine Textstrophe wiederholt. Die erste und die dritte Strophe (beide mit *A* gekennzeichnet) sowie die zweite und die vierte Strophe (beide mit *B* gekennzeichnet) sind musikalisch identisch und unterscheiden sich lediglich im Text. Das Vor-, das Zwischen- und das Nachspiel sind nahezu identisch.

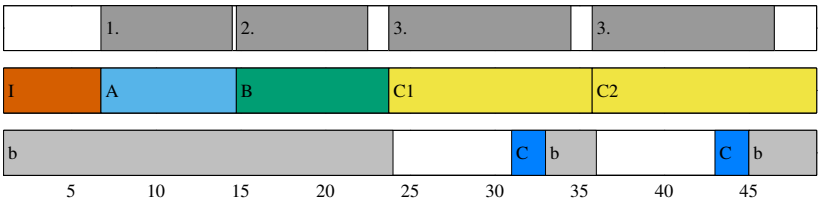
11. Frühlingstraum



Variiertes Strophenlied in A-Dur mit sechs Strophen.

Die Segmentierung entspricht weitestgehend der Strophe neinteilung. Es wird keine Textstrophe wiederholt. Die Gesamtstruktur des Liedes ist zweigeteilt, wobei beide Teile musikalisch gesehen absolut identisch sind: *I A B C* | *I A B C*. Die Segmente *I*, *A* und *C* stehen in A-Dur. *B* hingegen ist harmonisch gesehen heterogen.

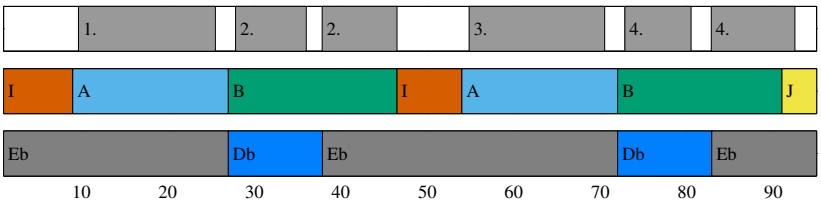
12. Einsamkeit



Durchkomponiertes Lied in h-Moll mit drei Strophen.

Die Segmentierung entspricht der Strophe neinteilung, wobei die dritte Strophe als Einzige wiederholt wird (Segmente $C1$ und $C2$). Der gesungene Part lässt sich musikalisch in drei Segmente unterteilen: A, B und C. Das achttaktige Segment A lässt sich wiederum in zwei Teile aufspalten; ein viertaktiges Motiv (T. 7–10 mit Auftakt) und seine genaue Wiederholung (T. 11–14 mit Auftakt). Das achttaktige Segment B lässt sich ebenso wie A in zwei Teile aufspalten; ein viertaktiges Motiv (T. 15–18 mit Auftakt) und seine leicht variierte Wiederholung (T. 19–22 mit Auftakt). Auch das Segment C besteht aus zwei nahezu identischen Teilen; $C1$ und $C2$, wobei $C2$ einen anderen Schluss aufweist als $C1$.

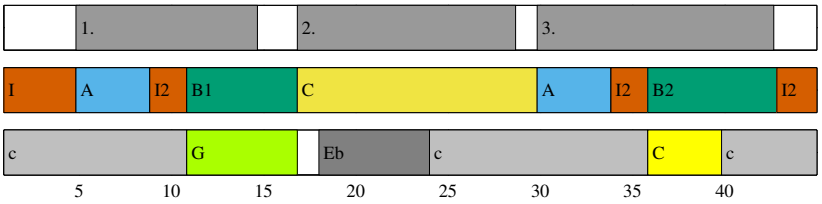
13. Die Post



Variiertes Strophenlied in Es-Dur mit vier Strophen.

Die Segmentierung entspricht der Strophe neinteilung, wobei die zweite und die vierte Strophe wiederholt werden. Das Lied ist zweiteilig aufgebaut, wobei beide Teile musikalisch gesehen nahezu identisch sind: $I\,A\,B\,|\,I\,A\,B\,J$. Das Vor- und Zwischenspiel I sind ebenfalls identisch. Das Nachspiel J kommt nur ein einziges Mal vor. B stellt eine starke Variation von A dar. Es gibt rhythmische, harmonische und melodische Unterschiede. Die Verwandtschaft lässt sich u.a. in den Passagen mit dem Text »Mein Herz« erkennen.

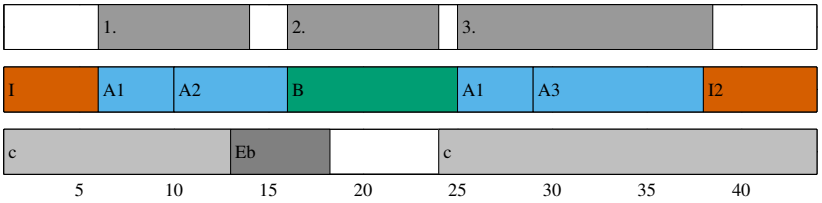
14. Der greise Kopf



Variiertes Strophenlied in c-Moll mit drei Strophen.

Dreiteilige Form ABA', was in der Feinsegmentierung $I1\,A\,I2\,B1\,|\,C\,|\,A\,I2\,B2\,I2$ entspricht. Das Lied hat ein instrumentales Vorspiel $I1$, zwei instrumentale Zwischenspiele $I2$ und ein Nachspiel $I2$, das mit den Zwischenspielen identisch ist. Der erste Teil umfasst das Vorspiel $I1$, das Segment A, ein Zwischenspiel $I2$ sowie das Segment $B1$ und stellt die Vertonung der ersten Textstrophe dar. A und $B1$ weisen sowohl in der Harmonik als auch in der Melodik starke Unterschiede auf, sind sich rhythmisch jedoch sehr ähnlich. Der Mittelteil (C in der Feinsegmentierung) hat ebenfalls rhythmische Ähnlichkeit mit A und $B1$. Hier wird die zweite Strophe des Gedichts vertont. Der letzte Teil ist im Wesentlichen mit dem ersten Teil identisch, wobei $B2$ eine melodische Variation von $B1$ ist. Dieser Teil entspricht der dritten Textstrophe.

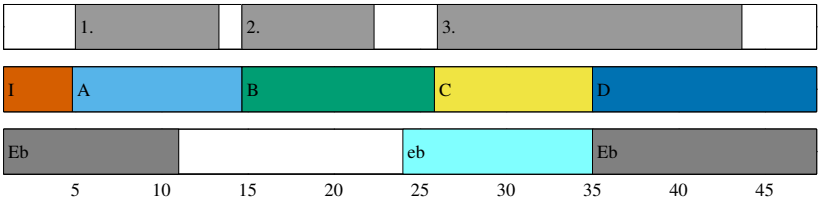
15. Die Krähe



Variiertes Strophenlied in c-Moll mit drei Strophen.

Es lässt sich eine dreiteilige ABA'-Form mit einem instrumentalen Vor- und Nachspiel erkennen ($I\,A1\,A2\,|\,B\,|\,A1\,A3\,I2$ in der Feinsegmentierung). Die dreiteilige Struktur stimmt mit der Strophe einteilung überein. Das instrumentale Vorspiel I stimmt in den ersten vier Takten mit der Strophe $A1$ überein, wobei die Gesangsmelodie von der rechten Hand des Klaviers übernommen wird. $A2$ stellt eine leichte Variation von $A1$ dar. In B wechselt trotz gleicher Art von Begleitung die Stimmung: Dur statt Moll. In $A3$ wird $A1$ sowohl bzgl. der Harmonik als auch bzgl. der Melodik stark variiert. Das Nachspiel $I2$ stellt eine leichte Variation des Vorspiels I dar.

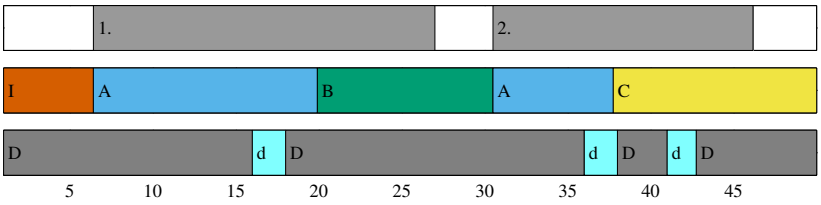
16. Letzte Hoffnung



Durchkomponiertes Lied in Es-Dur mit drei Strophen.

Die Segmentierung wurde nach inhaltlich-textuellen und melodischen Kriterien durchgeführt. *A* besteht aus zwei Wiederholungen eines viertaktigen Motivs. Das Lied steht bis auf das Segment *B* mit schwer zu bestimmender Tonart weitestgehend in Es-Dur.

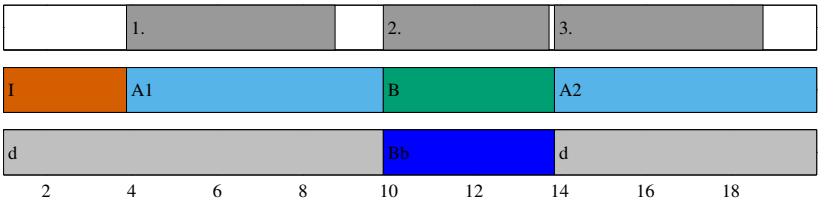
17. Im Dorfe



Variiertes Strophenlied in D-Dur mit zwei Strophen.

Die Segmentierung entspricht im Wesentlichen der Strophe neinteilung, wobei auf jede Textstrophe zwei musikalische Segmente fallen. Die musikalische Struktur des Liedes lässt drei verschiedene Segmente erkennen: *A*, *B* und *C*, wobei nur *A* im Laufe des Stücks wiederholt wird. Die Wiederholung von *A* variiert in der zweiten Hälfte, daher könnte das zweite Vorkommen von *A* auch mit A' bezeichnet werden. Ebenfalls könnte man *A* auch in $A1$ (T. 6–11) und $A2$ (T. 12–19) sowie A' in $A1$ (T. 30–35) und $A3$ (T. 36–37) unterteilen, oder nur zwischen *A* und A' (genauer: $A1$ und $A2$) ohne weitere Unterteilung unterscheiden. Die Strophe *C* ist in der Melodik und in der Harmonik näher an *A* als *B* an *A*.

18. Der stürmische Morgen

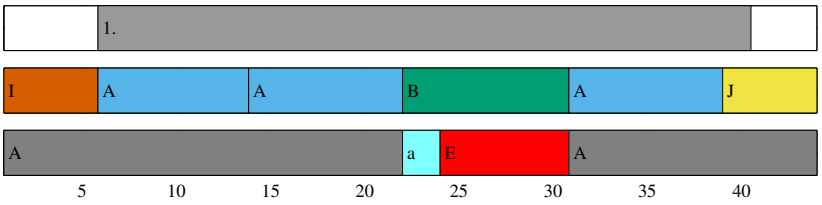


Durchkomponiertes Strophenlied in d-Moll mit drei Strophen.

Es lässt sich eine dreiteilige ABA'-Form erkennen mit einem instrumentalen Vorspiel *I*. Diese entspricht der Feinsegmentierung und im Wesentlichen der Stropheneinteilung. Es wird keine Strophe wiederholt.

Der erste und der letzte Teil stehen zum großen Teil in d-Moll. Der Mittelteil steht in B-Dur. *A*2 stellt eine Variation von *A*1 dar, wobei die ersten zwei Takte von *A*2 mit dem dritten und dem vierten Takt von *A*1 übereinstimmen. Auch die Schlusspassage von *A*2 ist mit der Schlusspassage von *A*1 identisch.

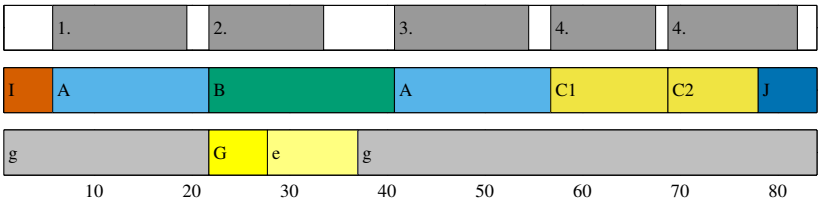
19. Täuschung



Durchkomponiertes Lied in A-Dur mit nur einer Textstrophe.

Es lässt sich eine dreiteilige ABA'-Form oder Barform mit instrumentalen Vor- und Nachspiel erkennen (*I A A B A J*). Das instrumentale Vorspiel wird im gesamten Stück als Begleitung gespielt und auch im Nachspiel motivisch verarbeitet. Im ersten Teil wird das Segment *A* zweimal hintereinander gespielt, im letzte Teil dagegen nur einziges Mal. Der Mittelteil setzt sich in der Harmonik und in der Melodik von der Strophe *A* ab.

20. Der Wegweiser



Variiertes Strophenlied in g-Moll mit vier Strophen.

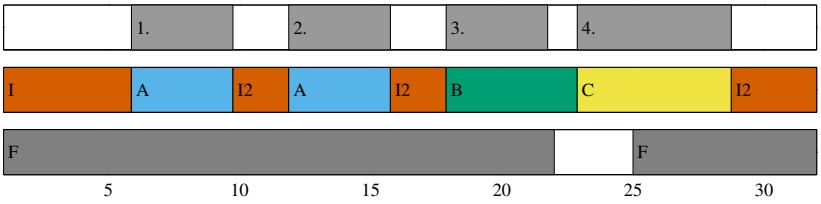
Die annotierte Segmentierung entspricht nahezu vollständig der Stropheneinteilung, wobei die vierte Textstrophe wiederholt wird. Die beiden *A*-Segmente sind leicht variiert. *B* kann als eine starke Variation von *A* angesehen werden. *C* basiert rhythmisch und harmonisch auf dem Material von *A*, variiert jedoch stark in der Melodik; die Begleitung ändert sich im Vergleich

zu *A* und *B* nur wenig. Das Nachspiel ist sehr kurz und zur Hälfte gesungen und muss daher nicht unbedingt von *C*2 abgetrennt werden.

Je nach Gewichtung der Variationen ergeben sich die alternativen Segmentierungen:

- *I A*1 *A*2 *A*3 *C*1 *C*2
- *I A*1 *A*2 *A*3 *A*4 *A*5

21. Das Wirtshaus



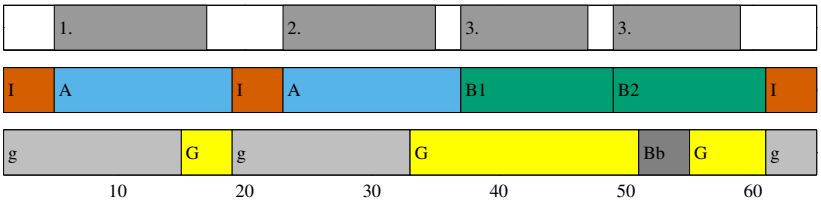
Variiertes Strophenlied in F-Dur mit vier Strophen.

Das instrumentale Vorspiel bildet im Wesentlichen die Begleitstimme des gesamten Liedes. Die Zwischenspiele *I*2 stellen die zweite Hälfte des Vorspiels *I*1 dar. *B* und *C* können als starke Variationen von *A* angesehen werden, wobei die Rhythmik immer gleich bleibt. Die Harmonik und die Melodik weichen zuweilen stark von dem Original *A* ab.

Zwei gleichberechtigte Segmentierungen:

- *I*1 *A I*2 *A I*2 *B C I*2
- *I*1 *A*1 *I*2 *A*1 *I*2 *A*2 *A*3 *I*2

22. Mut

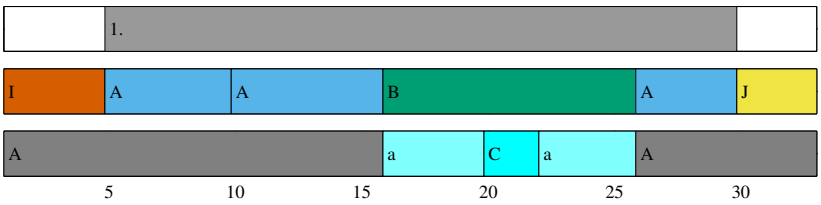


Variiertes Strophenlied in g-Moll mit drei Strophen.

Die Segmentierung entspricht im Wesentlichen der Strophe neinteilung. Das Vor-, Zwischen- und Nachspiel sind identisch. Es lassen sich zwei Strophenarten unterscheiden: *A* und *B*. Diese weisen untereinander starke melodische und leichte harmonische Unterschiede auf. Die Rhythmik bleibt sowohl in der Gesangs- als auch in der Klavierstimme weitestgehend gleich. Sowohl *A* als auch *B*1 und *B*2 sind zweiteilig aufgebaut und bestehen aus zwei Wiederholungen

eines 6- bis 7-taktigen Motivs. Die zweiten Wiederholungen weichen jeweils leicht von dem Original ab.

23. Die Nebensonnen



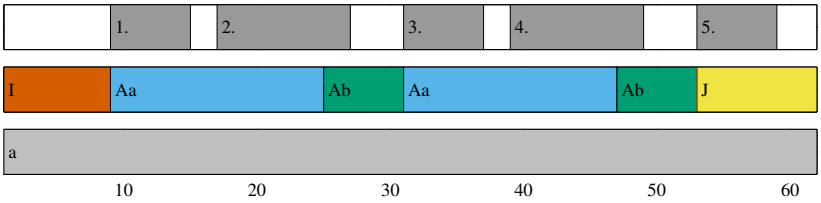
Durchkomponiertes Lied in A-Dur mit nur einer Textstrophe.

Der Aufbau lässt sich genauso gut als dreiteilige ABA-Form oder Barform beschreiben. Die *A*-Segmente sind identisch. *B* ist eine Moll-Variation von *A*. *I* und *J* weisen starke Ähnlichkeiten auf.

Mögliche Segmentierungen:

- *I A A B A J*
- *I* A1 A1 *A*2 A1 *J*

24. Der Leiermann



Variiertes Strophenlied in a-Moll mit fünf Strophen.

Die Segmentierung wurde wiederholungsbasiert durchgeführt und entspricht nicht der Stropheneinteilung. Die erste und die dritte Strophe des Gedichts sind gleich vertont. Das entsprechende musikalische Segment *Aa* kann in vier viertaktige Abschnitte unterteilt werden, wobei der erste und der zweite sowie der dritte und der vierte Abschnitt jeweils identisch sind. Alle vier Abschnitte sind durch ein zweitaktiges instrumentales Zwischenspiel getrennt. *Ab* ist sechstaktig und stellt den Abschluss der zweiten bzw. der vierten Strophe des Gedichts dar, wobei die ersten beiden Takte gesungen und die letzten vier rein instrumental sind. Das letzte Segment *J* kommt nur ein einziges Mal vor. Die zweite Hälfte ist rein instrumental.

Literaturverzeichnis

[1] Frans G. J. Absil. Musical analysis: Visiting the great composers. `http://www.fransabsil.nl/archpdf/musanbk.pdf`, 2009. Retrieved 11.05.2012.

[2] Jean-Julien Aucouturier, Francois Pachet, and Mark Sandler. "The way it sounds": Timbre models for analysis and retrieval of music signals. *IEEE Transactions on Multimedia*, 7(6):1028–1035, 2005.

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

[4] François Auger and Patrick Flandrin. Improving the readability of time-frequency and time-scale representations by the reassignment method. *IEEE Transactions on Signal Processing*, 43(5):1068–1089, 1995.

[5] 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.

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

[7] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.

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

[9] Frédéric Bimbot, Emmanuel Deruty, Gabriel Sargent, Emmanuel Vincent, et al. Semiotic structure labeling of music pieces: Concepts, methods and annotation conventions. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 235–240, Porto, Portugal, 2012.

[10] Friedrich Blume and Ludwig Finscher, editors. *Musik in Geschichte und Gegenwart*. Bärenreiter, Kassel, Germany, 2nd edition, 1994.

[11] Michael J. Bruderer, Martin McKinney, and Armin Kohlrausch. Structural boundary perception in popular music. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 198–201, Victoria, Canada, 2006.

[12] Bernd Bruegge, Christoph Teschner, Peter Lachenmaier, Eva Fenzl, Dominik Schmidt, and Simon Bierbaum. Pinocchio: Conducting a virtual symphony orchestra. In *Proceedings of the international conference on Advances in computer entertainment technology*, pages 294–295. ACM, 2007.

[13] Emilios Cambouropoulos. From MIDI to traditional musical notation. In *Proceedings of the AAAI Workshop on Artificial Intelligence and Music*, volume 30, 2000.

[14] Michael A. Casey, Remco Veltkamp, 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.

[15] Ali Taylan Cemgil and Bert Kappen. Monte Carlo methods for tempo tracking and rhythm quantization. *Journal of Artificial Intelligence Research*, 18(1):45–81, 2003.

[16] Ali Taylan Cemgil, Bert Kappen, and Peter Desain. Rhythm quantization for transcription. *Computer Music Journal*, 24(2):60–76, 2000.

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

[18] Wei Chai. Structural analysis of musical signals via pattern matching. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages V–549–52 vol.5, Hong Kong, China, 2003.

[19] Wei Chai. Semantic segmentation and summarization of music: methods based on tonality and recurrent structure. *IEEE Signal Processing Magazine*, 23(2):124–132, 2006.

[20] KyuSik Chang, GyuBeom Kim, and TaeYong Kim. Video game console audio: Evolution and future trends. In *Computer Graphics, Imaging and Visualisation (CGIV)*, pages 97–102, 2007.

[21] Taemin Cho and Juan Pablo Bello. On the relative importance of individual components of chord recognition systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 22(2):477–492, 2014.

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

[23] Ching-Hua Chuan and Elaine Chew. Creating ground truth for audio key finding: When the title key may not be the key. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 247–252, Porto, Portugal, 2012.

[24] Andrzej Cichocki, Rafal Zdunek, and Anh Huy Phan. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation.* John Wiley and Sons, 2009.

[25] Karen Collins. *Game Sound: an introduction to the history, theory, and practice of video game music and sound design.* MIT Press, 2008.

[26] David Damm. *A Digital Library Framework for Heterogeneous Music Collections–from Document Acquisition to Cross-Modal Interaction.* PhD thesis, University of Bonn, 2013.

[27] David Damm, Harald Grohganz, Frank Kurth, Sebastian Ewert, and Michael Clausen. SyncTS: Automatic synchronization of speech and text documents. In *Proceedings of the AES International Conference Semantic Audio*, pages 98–107, Ilmenau, Germany, 2011.

[28] 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.

[29] Roger B. Dannenberg and Christopher Raphael. Music score alignment and computer accompaniment. *Communications of the ACM, Special Issue: Music information retrieval*, 49(8):38–43, 2006.

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

[31] Steven B. Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Readings in Speech Recognition*, pages 65–74, 1990.

[32] Otto Erich Deutsch and Werner Aderhold (ed.). *Franz Schubert, thematisches Verzeichnis seiner Werke in chronologischer Folge.* Bärenreiter, Kassel, 1978.

[33] Nils Dittbrenner. *Soundchip-Musik: Computer-und Videospielmusik von 1977-1994*, volume 9. Electronic Publishing, 2007.

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

[35] Mark Dolson. The phase vocoder: A tutorial. *Computer Music Journal*, 10(4):14–27, 1986.

[36] Edward R. Dougherty. *An Introduction to Morphological Image Processing.* SPIE Optical Engineering Press, Bellingham, WA, USA, 1992.

[37] J. Stephen Downie. Music information retrieval. *Annual Review of Information Science and Technology (Chapter 7)*, 37:295–340, 2003.

[38] 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.

[39] J. Stephen Downie, Andreas F. Ehmann, Mert Bay, and M. Cameron Jones. The music information retrieval evaluation exchange: Some observations and insights. In *Advances in music information retrieval*, pages 93–115. Springer, 2010.

[40] Jonathan Driedger, Harald Grohganz, Thomas Prätzlich, Sebastian Ewert, and Meinard Müller. Score-informed audio decomposition and applications. In *Proceedings of the ACM International Conference on Multimedia (ACM-MM)*, pages 541–544, Barcelona, Spain, 2013.

[41] Walther Dürr. Schubert, Franz (Peter). In Blume and Finscher [10], pages 75–205.

[42] Chris Duxbury, Mike Davies, and Mark Sandler. Improved time-scaling of musical audio using phase locking at transients. In *Audio Engineering Society Convention*, 4 2002.

[43] Hans Heinrich Eggebrecht and Gerhard Kwiatkowski. *Meyers Taschenlexikon Musik in 3 Bänden*. Bibliographisches Institut, 1984.

[44] Andreas F. Ehmann, Mert Bay, J. Stephen Downie, Ichiro Fujinaga, and David De Roure. Music structure segmentation algorithm evaluation: Expanding on mirex 2010 analyses and datasets. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 561–566, Miami, FL, USA, 2011.

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

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

[47] 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.

[48] 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.

[49] Jonathan T. Foote and Matthew L. Cooper. Media segmentation using self-similarity decomposition. *Storage and Retrieval for Media Databases*, 5021(1):167–175, 2003.

[50] Christian Fremerey. *Automatic Organization of Digital Music Documents – Sheet Music and Audio*. PhD thesis, University of Bonn, 2010.

[51] Melanie Fritsch. History of Video Game Music. In Moormann [115], pages 11–40.

[52] Ferdinand Fuhrmann. *Automatic musical instrument recognition from polyphonic music audio signals*. PhD thesis, UPF Barcelona, 2012.

[53] Sean A Fulop and Kelly Fitz. Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications. *The Journal of the Acoustical Society of America*, 119(1):360–371, 2006.

[54] Karl-Heinz Goldhorn and Hans-Peter Heinz. *Mathematik für Physiker 1: Grundlagen aus Analysis und Linearer Algebra*. Mathematik für Physiker / Karl-Heinz Goldhorn. Springer London, Limited, 2007.

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

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

[57] 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.

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

[59] 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.

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

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

[62] 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.

[63] Harald Grohganz, Michael Clausen, and Meinard Müller. Estimating musical time information from performed MIDI files. In *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR)*, pages 35–40, Taipei, Taiwan, 2014.

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

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

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

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

[68] George Grove, Stanley Sadie, and John Tyrrell, editors. *The New Grove Dictionary of Music and Musicians*. Macmillan Publishers Ltd., London, UK, 2nd edition, 2001.

[69] Christopher Harte and Mark Sandler. Automatic chord identification using a quantised chromagram. In *Proceedings of the Audio Engineering Society Convention*, Barcelona, Spain, 2005.

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

[71] Frank Hoffmann, editor. *Encyclopedia of Recorded Sound*. Routledge, New York, NY, USA, second edition, 2005.

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

[73] Kevin Horton. NES Music Format Spec. `http://kevtris.org/nes/nsfspec.txt`, 2000. Retrieved 09.06.2014.

[74] David Miles Huber. *The MIDI manual*. Focal Press, 3rd edition, 2006.

[75] Özgür İzmirli. Localized key finding from audio using nonnegative matrix factorization for segmentation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 195–200, Vienna, Austria, 2007.

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

[77] Florian Kaiser, Marina Georgia Arvanitidou, and Thomas Sikora. Audio similarity matrices enhancement in an image processing framework. In *International Workshop on Content-Based Multimedia Indexing (CBMI)*, Madrid, Spain, 2011.

[78] Florian Kaiser and Geoffroy Peeters. A simple fusion method of state and sequence segmentation for music structure discovery. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 257–262, Curitiba, Brazil, 2013.

[79] 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.

[80] Maksim Khadkevich, Thomas Fillon, Gaël Richard, and Maurizio Omologo. A probabilistic approach to simultaneous extraction of beats and downbeats. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 445–448. IEEE, 2012.

[81] Clemens Kühn. Rhythmus, Metrum, Takt. In Blume and Finscher [10], pages 607–643.

[82] Clemens Kühn. *Formenlehre*. Bärenreiter-Verlag, Kassel, Germany, eighth edition, 2007.

[83] Jingu Kim and Haesun Park. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 353–362, Pisa, IT, 2008.

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

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

[86] Verena Konz and Meinard Müller. A cross-version approach for harmonic analysis of music recordings. In Meinard Müller, Masataka Goto, and Markus Schedl, editors, *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups*, pages 53–72. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2012.

[87] Verena Konz, Meinard Müller, and Rainer Kleinertz. A Cross-Version Chord Labelling Approach for Exploring Harmonic Structures? – A Case Study on Beethoven's Appassionata. *Journal of New Music Research*, pages 1–17, 2013.

[88] Florian Krebs, Sebastian Böck, and Gerhard Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, Curitiba, Brazil, 2013.

[89] Verena Kriesel. *Music Synchronization, Audio Matching, Pattern Detection, and User Interfaces for a Digital Music Library System*. PhD thesis, Universität Bonn, 2013.

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

[91] Alison Latham, editor. *The Oxford companion to music*. Oxford University Press, Oxford, UK, 2002.

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

[93] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Proceedings of the Neural Information Processing Systems (NIPS)*, pages 556–562, Denver, CO, USA, 2000.

[94] Eric Lee, Thorsten Karrer, and Jan Borchers. Toward a framework for interactive systems to conduct digital audio and video streams. *Computer Music Journal*, 30(1):21–36, 2006.

[95] Eric Lee, Teresa Marrin Nakra, and Jan Borchers. You're the conductor: a realistic interactive conducting system for children. In *Proceedings of the conference on New interfaces for musical expression (NIME)*, pages 68–73. National University of Singapore, 2004.

[96] Kyogu Lee and Malcolm Slaney. Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):291–301, 2008.

[97] Bernhard Lehner, Reinhard Sonnleitner, and Gerhard Widmer. Towards light-weight, real-time-capable singing voice detection. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 53–58, Curitiba, Brazil, 2013.

[98] Bernhard Lehner, Gerhard Widmer, and Reinhard Sonnleitner. On the reduction of false positives in singing voice detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7480–7484, Florence, Italy, 2014. IEEE.

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

[100] 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.

[101] François Léonard. Phase spectrogram and frequency spectrogram as new diagnostic tools. *Mechanical Systems and Signal Processing*, 21(1):125–137, 2007.

[102] Justin London. Metre. In Grove et al. [68], page 531.

[103] 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.

[104] 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.

[105] Namunu C. Maddage, Changsheng Xu, Mohan S. Kankanhalli, and Xi Shao. Content-based music structure analysis with applications to music semantics understanding. In *Proceedings of the ACM International Conference on Multimedia*, pages 112–119, New York, NY, USA, 2004.

[106] MakeMusic, Inc. MusicXML for Exchanging Digital Sheet Music, `http://www.musicxml.com`, Retrieved 26.12.2013.

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

[108] 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.

[109] Brian McFee and W. Ellis, Daniel P.˙ Analyzing song structure with spectral clustering. In *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR)*, pages 405–410, Taipei, Taiwan, 2014.

[110] Benoit Meudic. Automatic meter extraction from midi files. In *Proc. Journées d'informatique musicale*. Citeseer, 2002.

[111] Michael Michaelis. Musik & Form. `http://www.michael-michaelis.de/htdocs/musikalischeformen`, Retrieved 18.06.2009.

[112] International MIDI Association et al. *Standard MIDI Files 1.0*. International MIDI Association, 1988.

[113] MIDI Manufacturers Association et al. *The complete MIDI 1.0 detailed specification: incorporating all recommended practices*. MIDI Manufacturers Association, 1996.

[114] F. Richard Moore. The dysfunctions of MIDI. *Computer Music Journal*, 12(1):19–28, 1988.

[115] Peter Moormann, editor. *Music and Game*. Springer VS, 2013.

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

[117] 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.

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

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

[120] 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.

[121] 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.

[122] 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.

[123] Meinard Müller, Nanzhu Jiang, Harald Grohganz, and Michael Clausen. Strukturanalyse für Musiksignale. In *GI-Edition: Lecture Notes in Informatics*, pages 2943–2957, Koblenz, Germany, 2013.

[124] 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.

[125] Meinard Müller, Verena Konz, Wolfgang Bogler, and Vlora Arifi-Müller. Saarland music data (SMD). In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR): Late Breaking session*, 2011.

[126] 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.

[127] 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.

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

[129] Declan Murphy. Tracking a conductor's baton. In *Proceedings of the 12th Danish Conference on Pattern Recognition and Image Analysis*, volume 2003, page 05, 2003.

[130] Declan Murphy, Tue Haste Andersen, and Kristoffer Jensen. Conducting audio files via computer vision. In *Gesture-based communication in human-computer interaction*, pages 529–540. Springer, 2004.

[131] Klaus Wolfgang Niemöller and Bram Gätjen, editors. *Perspektiven und Methoden einer Systemischen Musikwissenschaft*. Peter Lang, 1998.

[132] Oriol Nieto and Morwaread Farbood. Identifying polyphonic musical patterns from audio recordings using music segmentation techniques. In *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR)*, pages 411–416, Taipei, Taiwan, 2014.

[133] 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)*, pages 236–240. IEEE, 2013.

[134] Oriol Nieto and Jordan B. L. Smith. 2013 late-break session on music segmentation. ISMIR.net, `http://ismir2013.ismir.net/wp-content/uploads/2014/03/lbd6.pdf`. Retrieved 11.06.2014.

[135] Silvia Noschese, Lionello Pasquini, and Lothar Reichel. Tridiagonal toeplitz matrices: properties and novel applications. *Numerical Linear Algebra with Applications*, 20(2):302–326, 2013.

[136] Tin Lay Nwe, Arun Shenoy, and Ye Wang. Singing voice detection in popular music. In *Proceedings of the ACM International Conference on Multimedia*, pages 324–327. ACM, 2004.

[137] Matthias Oborski. Spielemusik: Geschichte. `http://spielemusikkonzerte.de/hintergrund/geschichte`, 2014. Retrieved 11.06.2014.

[138] Lawrence O'Gorman and Rangachar Kasturi. *Document image analysis*, volume 39. IEEE Computer Society Press Los Alamitos, CA, 1995.

[139] Bee Suan Ong. *Structural Analysis and Segmentation of Music Signals*. PhD thesis, University Pompeu Fabra, Barcelona, Spain, 2007.

[140] Nicloa Orio. Music retrieval: A tutorial and review. *Foundation and Trends in Information Retrieval*, 1(1):1–90, 2006.

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

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

[143] Matthias Pasdzierny. Geeks on Stage? Investigations in the World of (Live) Chipmusic. In Moormann [115], pages 171–190.

[144] Jouni Paulus. *Signal Processing Methods for Drum Transcription and Music Structure Analysis*. PhD thesis, Tampere University of Technology, Tampere, Finland, 2010.

[145] Jouni Paulus and Anssi Klapuri. Music structure analysis using a probabilistic fitness measure and an integrated musicological model. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 369–374, Philadelphia, PA, USA, 2008.

[146] 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.

[147] 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.

[148] Johan Pauwels, Florian Kaiser, and Geoffroy Peeters. Combining harmony-based and novelty-based approaches for structural segmentation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 601–606, Curitiba, Brazil, 2013.

[149] Johan Pauwels and Jean-Pierre Martens. Integrating musicological knowledge into a probabilistic framework for chord and key extraction. In *Audio Engineering Society Convention 128*, London, UK, 2010. AES.

[150] Johan Pauwels and Geoffroy Peeters. Segmenting music through the joint estimation of keys, chords and structural boundaries. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 741–744. ACM, 2013.

[151] Osemwaro Pedro. A conductible virtual orchestra. Master's thesis, Imperial College London, London, UK, 2006.

[152] 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.

[153] 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.

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

[155] Geoffroy Peeters. Chroma-based estimation of musical key from audio-signal analysis. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 115–120, Victoria, Canada, 2006.

[156] Geoffroy Peeters. Musical key estimation of audio signal based on hidden markov modeling of chroma vectors. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 127–131, Montreal, Quebec, Canada, 2006.

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

[158] 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.

[159] Geoffroy Peeters and Victor Bisot. Improving music structure segmentation using lag-priors. In *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR)*, pages 337–342, Taipei, Taiwan, 2014.

[160] Geoffroy Peeters, Bruno L Giordano, Patrick Susini, Nicolas Misdariis, and Stephen McAdams. The timbre toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, 130(5):2902–2916, 2011.

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

[162] Christopher Raphael. Automated rhythm transcription. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2001.

[163] 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.

[164] Martın Rocamora and Perfecto Herrera. Comparing audio descriptors for singing voice detection in music audio files. In *Brazilian Symposium on Computer Music (SBCM)*, pages 187–196, Sao Paulo, Brazil, 2007.

[165] Peter Rummenhöller. Harmonielehre. In Blume and Finscher [10], pages 132–153.

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

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

[168] Jaakko Sauvola and Matti Pietikäinen. Adaptive document image binarization. *Pattern recognition*, 33(2):225–236, 2000.

[169] Mikkel N. Schmidt and Morten Mørup. Nonnegative matrix factor 2-d deconvolution for blind single channel source separation. In *Independent Component Analysis and Blind Signal Separation*, pages 700–707. Springer, 2006.

[170] Wilhelm Seidel. Rhythmus, Metrum, Takt. In Blume and Finscher [10], pages 257–317.

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

[172] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., Orlando, FL, USA, 1984.

[173] 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.

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

[175] William A Sethares. *Tuning, Timbre, Spectrum, Scale*, volume 2. Springer, 2005.

[176] 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.

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

[178] Paris Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation (Lecture Notes in Computer Science 31959*, pages 494–499, Grenada, Spain, 2004.

[179] Gordon D. Smith. *Numerical Solution of Partial Differential Equations*. Clarendon Press, Oxford, UK, 1965.

[180] Jordan B. L. Smith. A comparison and evaluation of approaches to the automatic formal analysis of musical audio. Master's thesis, McGill University, Montreal, Quebec, Canada, 2010.

[181] Jordan B. L. Smith and Elaine Chew. A meta-analysis of the MIREX structure segmentation task. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 469–474, Curitiba, Brazil, 2013.

[182] 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.

[183] 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.

[184] Liming Song, Ming Li, and Yonghong Yan. Automatic vocal segments detection in popular music. In *International Conference on Computational Intelligence and Security (CIS)*, pages 349–352. IEEE, 2013.

[185] Reinhard Sonnleitner, Bernhard Niedermayer, Gerhard Widmer, and Jan Schlüter. A simple and effective spectral feature for speech detection in mixed audio signals. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2012.

[186] Joachim Staib. Interaktive Analyse und Visualisierung der Mehrdeutigkeit von Nichtnegativen Matrixfaktorisierungen. Master's thesis, Technische Universität Dresden, Dresden, Germany, 2012.

[187] Wolfram Steinbeck. *Struktur und Ähnlichkeit*, volume 25 of *Kieler Schriften zur Musikwissenschaft*. Bärenreiter, Kassel, 1982.

[188] Haruto Takeda, Takuya Nishimoto, and Shigeki Sagayama. Rhythm and tempo analysis toward automatic music transcription. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4. IEEE, 2007.

[189] Haruto Takeda, Naoki Saito, Tomoshi Otsuki, Mitsuru Nakai, Hiroshi Shimodaira, and Shigeki Sagayama. Hidden markov model for automatic transcription of MIDI signals. In *IEEE Workshop on Multimedia Signal Processing*, pages 428–431. IEEE, 2002.

[190] Brad Taylor. 2a03 sound channel hardware documentation. `http://nesdev.com/NESSOUND.txt`, 2003. Retrieved 09.06.2014.

[191] David Temperley and Daniel Sleator. Modeling meter and harmony: A preference-rule approach. *Computer Music Journal*, 23(1):10–27, 1999.

[192] Hiroko Terasawa, Malcolm Slaney, and Jonathan Berger. The thirteen colors of timbre. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 323–326, 2005.

[193] Christian Thurau, Kristian Kersting, and Christian Bauckhage. Yes we can: simplex volume maximization for descriptive web-scale matrix factorization. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 1785–1788. ACM, 2010.

[194] Masato Tsuchiya, Kazuki Ochiai, Hirokazu Kameoka, and Shigeki Sagayama. Probabilistic model of two-dimensional rhythm tree structure representation for automatic transcription of polyphonic midi signals. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–6, 2013.

[195] Rainer Typke, Frans Wiering, and Remco C. Veltkamp. A survey of music information retrieval systems. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 153–160, London, GB, 2005.

[196] Gregory H Wakefield. Mathematical representation of joint time-chroma distributions. In *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, pages 637–645. International Society for Optics and Photonics, 1999.

[197] Beiming Wang and Mark D. Plumbley. Musical audio stream separation by non-negative matrix factorization. In *Proceedings of the DMRN summer conference*, pages 23–24, 2005.

[198] Christof Weiß, Estefania Cano, and Hanna Lukashevich. A mid-level approach to local tonality analysis: Extracting key signatures from audio. In *Proceedings of the AES Conference on Semantic Audio*, London, GB, 2014.

[199] Christof Weiß and Julian Habryka. Chroma-based scale matching for audio tonality analysis. In *Proceedings of the 9th Conference on Interdisciplinary Musicology (CIM)*, Berlin, Germany, 2014.

[200] Ron J. Weiss and Juan Pablo Bello. Identifying repeated patterns in music using sparse convolutive non-negative matrix factorization. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 123–128, Utrecht, The Netherlands, 2010.

[201] Karlhans Weisse. *Nachhallgestaltung in Sälen für Tonfilm-Wiedergabe*. Bauwelt-Verlag, 1939.

[202] Karlhans Weisse. *Leitfaden der Raumakustik für Architekten*. Verlag des Druckhauses Tempelhof, 1949.

[203] Nick Whiteley, Ali Taylan Cemgil, and Simon Godsill. Sequential inference of rhythmic structure in musical audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4. IEEE, 2007.

[204] Nick Whiteley, Ali Taylan Cemgil, and Simon J Godsill. Bayesian modelling of temporal structure in musical audio. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 29–34, 2006.

[205] Wikimedia Commons. Datei:Metrum-takt-rhythmus.svg. `https://de.wikipedia.org/wiki/Datei:Metrum-takt-rhythmus.svg`, 2010. Retrieved 10.06.2014.

[206] Wikimedia Commons. File:NES-Console-Set.jpg. `http://commons.wikimedia.org/wiki/File:NES-Console-Set.jpg`, 2010. Retrieved 09.06.2014.

[207] Wikipedia. NES Sound Format. `http://en.wikipedia.org/wiki/NES_Sound_Format`, 2014. Retrieved 09.06.2014.

[208] Wikipedia. Nintendo Entertainment System technical specifications. `https://en.wikipedia.org/wiki/Nintendo_Entertainment_System_technical_specifications`, 2014. Retrieved 09.06.2014.

[209] Robert Winter. Schubert, Franz. In Grove et al. [68], pages 655–729.

[210] Guanglei Xiong. Local adaptive thresholding. MATLAB Central, `http://www.mathworks.com/matlabcentral/fileexchange/8647`, Retrieved 25.07.2014.

[211] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceeding of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 267–273, Toronto, Canada, 2003.

[212] Aaron C. Yang, Elaine Chew, and Anja Volk. A dynamic programming approach to adaptive tatum assignment for rhythm transcription. In *Seventh IEEE International Symposium on Multimedia*. IEEE, 2005.

[213] Dirk von Zeddelmann and Frank Kurth. A construction of compact MFCC-type features using short-time statistics for applications in audio segmentation. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 1504–1508, Glasgow, Scotland, UK, 2009.

[214] Tong Zhang. Automatic singer identification. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*, volume 1, pages I–33 – I–36. IEEE, 2003.

Index