



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Malte Kantak

Automatische Segmentierung von
Gitarrenaufnahmen in Einzeltöne und deren
Klassifikation

Malte Kantik

Automatische Segmentierung von
Gitarrenaufnahmen in Einzeltöne und deren
Klassifikation

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Wolfgang Fohl
Zweitgutachter : Prof. Dr.-Ing. Andreas Meisel

Abgegeben am 28. Juli 2011

Malte Kantik

Thema der Bachelorarbeit

Automatische Segmentierung von Gitarrenaufnahmen in Einzeltöne und deren Klassifikation

Stichworte

Signalverarbeitung, Audioanalyse, Gitarre, Töne, Segmentierung, Skriptsprache Matlab

Kurzzusammenfassung

Diese Bachelorarbeit beschäftigt sich mit der Entwicklung von Verfahren zum Auffinden von Einzeltönen und Artefakten in Soloaufnahmen verschiedener Konzertgitarren. Weiterhin werden Verfahren zur Analyse der gefundenen Artefakte entwickelt, um Informationen über diese zu bestimmen und sie dahingehend zu klassifizieren, ob es sich jeweils um einen Einzelton, einen Akkord oder ein Arpeggio handelt. Mit Hilfe dieser gefundenen Artefakte soll es möglich sein, Rückschlüsse auf die Gitarre zu ziehen, auf welcher dieses Solo gespielt wurde.

Malte Kantik

Title of the paper

Automated segmentation of guitar recordings into individual sounds and their classification

Keywords

Signal processing, audio analysis, guitar, tones, segmentation, scripting language Matlab

Abstract

This bachelor thesis deals with the development of methods for finding individual notes and artifacts in various solo classical guitar recordings. Furthermore it deals with the development of methods for analysis of the found artifacts to determine information about these and to classify them as to whether in each case it is a single note, a chord or an arpeggio. Using these Artifacts it should be possible to draw conclusions about the guitar, on which this solo was played. In this work, the generated methods are applied exemplary to solo recordings of various guitar players on three different guitars.

*Für meinen Großvater
Dr. med. Hans-Jürgen Kühl
(*1925, †1995)*

Danksagung

An dieser Stelle möchte ich den vielen Leuten danken, die den erfolgreichen Abschluss dieser Arbeit ermöglicht haben.

Besonderer Dank gilt Professor Dr. Wolfgang Fohl, welcher mir immer neue Denkanstöße gab, wenn ich mit dem Kopf durch die Wand wollte, und die Entstehung dieser Arbeit von der Bereitstellung des Themas bis zur Fertigstellung begleitet hat. Dieser Dank gilt ebenso für Professor Andreas Meisel.

Ich danke Dirk Gladiator dafür, dass er mich als Werkstudent eingestellt und immer Verständnis für meine studienbedingte dynamische Anwesenheit aufgebracht hat.

Mareike danke ich für ihr Vertrauen in meine Arbeit, ihre Geduld und Ihre Motivation.

Weiterhin danke ich meiner Familie und meinen Freunden für die Unterstützung während des gesamten Studiums.

Für das Korrekturlesen danke ich besonders Dr. Stephan Kantak, Mareike Thiel und Peter Hansen.

Inhaltsverzeichnis

Tabellenverzeichnis	9
Abbildungsverzeichnis	10
Listings	11
1. Einleitung	12
1.1. Ähnliche Arbeiten	12
1.2. Problemstellung	13
1.3. Ziel der Arbeit	14
1.4. Aufbau	14
1.5. Abgrenzung	14
2. Grundlagen	15
2.1. Gitarrenklang	15
2.2. Verwendete Audiodateien	16
2.3. Artefakttypen	18
2.3.1. Einzelton	18
2.3.2. Akkord	18
2.3.3. Arpeggierter Akkord	19
2.3.4. Unbekanntes Artefakt	19
3. Gängige Verfahren zur Segmentierung von Audiosignalen	20
3.1. Allgemeiner Aufbau	20
3.1.1. Vorverarbeitung	21
3.1.2. Reduktion	21
3.1.3. „Peak Picking“	21
3.2. Übersicht	22
3.2.1. Signalenergiebasierte Verfahren	23
3.2.2. Frequenzbasierte Verfahren	23
3.2.3. Hybridverfahren	24
3.2.4. Weitere Verfahren	26
3.3. Anforderungen	27

3.4. Bewertung	28
4. Das entwickelte Verfahren zur Segmentierung	30
4.1. Vorverarbeitung	31
4.1.1. Hervorheben von Anstiegen in der Signalenergie	31
4.1.2. Glättung	32
4.2. Reduktion	32
4.3. „Peak-Picking“	35
4.3.1. Nachverarbeitung	36
4.3.2. Thresholding	37
4.3.3. „Peak-Picking“	37
4.4. Bewertung des entwickelten Verfahrens	39
5. Analyse der Artefakte	40
5.1. Eigenschaften der Artefakte	40
5.1.1. Länge des Artefakts	40
5.1.2. Grundfrequenz und Grundton	40
5.1.3. Unterartefakte	41
5.1.4. Spektrogramm	41
5.2. Art der Artefakte	42
5.2.1. Arpeggierte Akkorde	42
5.2.2. Differenzierung zwischen Einzelton und Akkord	43
6. Ergebnisse	45
6.1. Versuchsaufbau	45
6.2. Resultate	48
6.2.1. Spielerabhängigkeit	50
6.2.2. Gitarrenabhängigkeit	52
6.3. Diskussion der Ergebnisse	55
7. Fazit und Ausblick	58
7.1. Erreichte Ziele	58
7.2. Ausblick und Weiterentwicklung	59
Literaturverzeichnis	60
A. Quellcode	64
A.1. Allgemeiner Aufbau	64
A.2. Programm zur Vorverarbeitung	64
A.3. Programm zur Erkennung von Artefaktanfängen	67
A.4. Prototyp zur Artefaktanalyse	71
A.5. Verwendete Programme/Codeteile	75

Tabellenverzeichnis

6.1. Erzeugte Report-Dateien	47
6.2. Artefakt-Eigenschaften	48
6.3. Ergebnisse des entwickelten Segmentierungsverfahrens	50
6.4. Ergebnisse zu den Aufnahmen von Hoffmann	50
6.5. Ergebnisse zu den Aufnahmen von Oba	50
6.6. Ergebnisse zu den Aufnahmen von Ossig	51
6.7. Ergebnisse zu den Aufnahmen der Gitarre von Hense	53
6.8. Ergebnisse zu den Aufnahmen der Gitarre von Marin	53
6.9. Ergebnisse zu den Aufnahmen der Gitarre von Wichmann	55

Abbildungsverzeichnis

2.1. Die verschiedenen Phasen des Tonverlaufs	16
2.2. Partitur zu „Prelude“	17
3.1. Beispiel für ein Blackman-Fenster in MATLAB	25
3.2. Ablaufschema einer hybridbasierten Segmentierung	26
4.1. Prozess der Informationsgewinnung in der Signalverarbeitung	30
4.2. Prozess der Audiosignalsegmentierung nach [BDA ⁺ 05]	30
4.3. Signal vor und nach der Anwendung der Glättungsfunktion	33
4.4. Beispiele für möglicher Anfänge von Artefakten	34
4.5. Verminderung der Informationen durch Blockbildung	35
4.6. Prozess des „Peak-Pickings“ nach [BDA ⁺ 05]	36
4.7. Überstieg des Schwellenwertes zur Bestimmung eines Tonanfangs	38
4.8. Mehrere erkannte Tonanfänge in Folge (arpeggierter Akkord)	39
5.1. Beispiel für ein erzeugtes Spektrogramm	42
6.1. Versuchsaufbau zur Analyse der Audiodateien	46
6.2. Ausgabe der Auswertungsanwendung	49
6.3. Erkennungsraten zu Spielern	51
6.4. Erkennungsraten zu Gitarren	54
6.5. Beispiel für einen verdeckten Tonanfang	56
6.6. Beispiel für zu niedrige Signalenergie verglichen mit Nachbartönen	56
A.1. Systemarchitektur des erzeugten Programms	65
A.2. Systemarchitektur der Vorverarbeitung	66
A.3. Systemarchitektur der Detektion	68
A.4. Systemarchitektur der Analyse	72

Listings

6.1. Beispiel Reportdatei	47
A.1. Aufruf der Vorverarbeitung aus process.m	64
A.2. preprocessing.m	66
A.3. normalize.m	66
A.4. Aufruf der Detektion aus process.m	67
A.5. detect.m	67
A.6. mainArtifacts.m	69
A.7. slicedPeaks.m	70
A.8. findArtifacts.m	70
A.9. Aufruf der Analyse aus process.m	72
A.10.isAccordIterative.m	73
A.11.getSpectrogram.m	75
A.12.getF0.m	75
A.13.tonname.m	76

1. Einleitung

Allen fühlenden Lebewesen ist die Eigenschaft zu eigen, den Informationsstrom, welcher sich aus deren Wahrnehmung ergibt, innerhalb von Sekundenbruchteilen zu analysieren und die für die jeweilige Situation wichtigen Informationen herauszufiltern. Noch anerkannter wird diese Leistung, wenn beabsichtigt wird, diese Eigenschaft mit Hilfe einer Maschine künstlich nachzubilden. Der Vorgang der Sinnesverarbeitung und Filterung an sich ist etwas, worüber wir im Allgemeinen nicht nachdenken, da wir diesen Vorgang ständig und millionenfach am Tag ausführen. Noch komplexer nachzubilden ist ein weiterer, mit dem ersten Verfahren eng verknüpfter Vorgang: Die Erkennung von Mustern in einer dieser Sinneswahrnehmungen.

Diese Arbeit beschäftigt sich mit nur einer der Sinneswahrnehmungen, der Audioverarbeitung. Und auch hier wird das Gebiet stark auf ein definiertes Problem eingeschränkt. Die Art der Verarbeitung und Klassifizierung hängt stark mit dem zu behandelnden Problem zusammen. Viel erstaunlicher ist es also, dass wir als Menschen intuitiv in dem entsprechenden Signal die Merkmale aus der Sinneswahrnehmung extrahieren, welche dann zur späteren Klassifikation des Artefakts im Signal erforderlich sind.

1.1. Ähnliche Arbeiten

Es gibt viele Arbeiten, die sich mit dem Thema der Audiosegmentierung befassen. Einige werden im Rahmen dieser Arbeit Erwähnung finden, doch die meisten dieser Arbeiten setzen sich mit anderen Instrumenten oder vor dem Hintergrund anderer Ziele mit dem Thema auseinander. Einige der verwendeten Verfahren finden Einfluss in das Kapitel 3 und werden dort erläutert, sodass sie an dieser Stelle nicht explizit erwähnt werden müssen.

Neben diesen Arbeiten gibt es aber auch solche, die sich zwar nicht mit der Thematik der Audiosegmentierung auseinander setzen, aber trotzdem in einem ähnlichen Kontext stehen. Sowohl die Veröffentlichung von Kerstin Dosenbach et al. [DFM08] als auch die Diplomarbeit von Henrik von Coler [Col09] befassen sich mit der Analyse von Audiomaterial, welches von den selben Spielern auf den selben Gitarren gespielt wurde wie die hier verwendeten Musikstücke.

Die Arbeit von Kerstin Dosenbach et al. befasste sich mit der Klassifizierung einzelner Töne dahingehend, die Gitarre zu ermitteln, auf welcher dieser Ton gespielt wurde. Dabei verwendete sie als Klassifikator eine Support Vector Machine¹ und erreichte dabei mit der besten Merkmalskombination eine Quote von 81,58% richtig klassifizierter Einzeltöne.

Henrik von Coler befasste sich in seiner Arbeit mit Verfahren zur Synthese von Gitarrenklängen mit Hilfe von spektraler Modellierung der tonalen und transienten Anteile. Er konnte im Rahmen von Hörtests zeigen, dass die Anschlaggeräusche einen sehr wichtigen Anteil an dem von einer Gitarre erzeugten Klang haben. Ein synthetisierter Ton klingt demnach deutlich natürlicher, wenn das Anschlaggeräusch mit modelliert wird.

1.2. Problemstellung

Indirekt ist diese Arbeit eine Fortführung der Arbeit von Kerstin Dosenbach et al. [DFM08]. Hier wurden drei Gitarren anhand von separat aufgezeichneten Tönen mit Hilfe von Support Vector Machines (SVM) klassifiziert. Es werden in dieser Arbeit Soloaufnahmen derselben Gitarren analysiert, mit dem Ziel, die Töne so zu extrahieren, dass die von Kerstin Dosenbach entwickelten Verfahren in einer Folgearbeit auf diese Töne angewendet werden können, um die Gitarre zu klassifizieren, auf der diese Aufnahme gespielt wurde. Bei Dosenbach et al. wird mit dem effizientesten Verfahren eine Klassifikationsrate von durchschnittlich 81,6% erzielt. Da diese Ergebnisse auf Basis separat und normiert aufgezeichneter Einzeltöne erzielt wurden, ist zu erwarten, dass die durchschnittliche Klassifikationsrate für die in dieser Arbeit extrahierten Artefakte deutlich hinter den 81,6% zurückbleiben wird. Trotzdem könnte die Klassifikationsrate per analysiertem Solo über denen der Einzeltöne bei Dosenbach et al. liegen, da hier nicht nur ein Ton für die Klassifikation herangezogen werden kann, sondern eine Vielzahl von Artefakten.

Was die Segmentierung der Audioaufzeichnungen besonders problematisch macht, ist zum einen die Verunreinigung des Audiosignals durch Störgeräusche und unsauber gespielte Töne, zum anderen ist die Erkennung der Tonanfänge und die spätere Differenzierung der unterschiedlichen Artefakttypen auf Basis des Audiosignals kein triviales Problem. In dieser Arbeit sollen die erwähnten Probleme mit Hilfe von vorhandenen oder noch zu entwickelnden Verfahren gelöst werden.

¹SVM

1.3. Ziel der Arbeit

Es ist Ziel dieser Arbeit ein Verfahren zu entwickeln, um Artefakte wie Einzeltöne, Akkorde und Arpeggios innerhalb eines gegebenen Audiosignals zu identifizieren, zu isolieren und zu extrahieren. Das Verfahren soll Informationen über das gefundene Artefakt liefern. Es sollen Art des Artefakts, Tonart, Maximalamplitude, Länge und Frequenz/-en bestimmt werden. Die Artefakte sollen in einem Merkmalsvektor gehalten und nach der Verarbeitung persistent gespeichert werden. Dies beinhaltet einen Bericht über die gefundenen Artefakte sowie die Artefakte selbst als separate Audiodateien.

1.4. Aufbau

Kapitel 2 (Grundlagen) gibt einen kurzen Überblick über die verwendeten Audiodateien, den allgemeinen Aufbau von Gitarrentönen und die verschiedenen Artefakttypen. Das 3. Kapitel (Gängige Verfahren zur Segmentierung von Audiosignalen) beschäftigt sich mit der Analyse des Problems und der Untersuchung gängiger Verfahren auf deren Anwendbarkeit auf das hier behandelte Problem. Kapitel 4 (Das entwickelte Verfahren zur Segmentierung) beschreibt die Realisierung und Entwicklung der in dieser Arbeit vorgestellten Verfahren zur Lösung der Probleme. Es folgen in Kapitel 5 (Analyse der Artefakte) die Analyseverfahren und die Ansätze der Differenzierungsverfahren. In Kapitel 6 (Ergebnisse) werden die Ergebnisse der Implementierung des Verfahrens vorgestellt, und abschließend werden in Kapitel 7 (Fazit und Ausblick) die erzielten Fortschritte dieser Arbeit zusammengefasst und bewertet. Es werden Ausblicke für eine mögliche Fortführung dieser Arbeit aufgezeigt, abschließend wird ein Fazit gezogen.

1.5. Abgrenzung

Die in dieser Bachelorarbeit entwickelten Verfahren sind nicht ohne weiteres für andere Aufgabengebiete anwendbar. Die Verfahren werden auf Grundlage der hier in dieser Arbeit behandelten Soloaufnahmen von Konzertgitarren entwickelt, und deren Anwendbarkeit auf andere Audiosignale muss vor der Anwendung überprüft werden.

Die Erkennung der Gitarre oder des Spielers auf Basis der extrahierten Artefakte ist nicht Teil dieser Arbeit. Die Arbeit behandelt ausschließlich die Analyse des Eingangssignals, die Extraktion und Analyse der enthaltenen Artefakte sowie die Bewertung der entwickelten Verfahren.

2. Grundlagen

Bevor wir uns in dieser Arbeit mit der Betrachtung des Problems oder der Entwicklung von Verfahren zur Lösung dieser beschäftigen, beginnen wir mit einer kurzen Einführung in die grundlegenden Gegenstände dieser Arbeit. Es wird auf den allgemeinen Aufbau von Gitarrenklängen eingegangen, die Beschaffenheit und Entstehung der verwendeten Audiodateien beleuchtet und eine Einteilung der in dem Eingangssignal zu erwartenden Artefakte vorgenommen.

2.1. Gitarrenklang

Gitarren gehören zur Gruppe der Saiteninstrumente (Chordophone). Diese zeichnen sich dadurch aus, dass die gespielten Töne durch eine in Schwingung gebrachte Saite erzeugt werden [Tom10]. Im Gegensatz zu anderen Saiteninstrumenten wie der Violine, haben die Gitarrenklänge einen harten, perkussiven Tonanfang [DSD02]. Im Allgemeinen wird der Tonverlauf in verschiedene Phasen unterteilt. Diese Phasen werden in Abbildung 2.1 verdeutlicht.

Es handelt sich dabei um die Phasen: „Attack“, „Decay“, „Sustain“ und „Release“. Die „Attack“-Phase beschreibt den schnellen Anstieg der Lautstärke des Tons auf sein Maximum. Bei der Gitarre wird diese Phase durch das Anschlagen der entsprechenden Seite eingeleitet. Direkt nachdem das Maximum erreicht ist, beginnt die Lautstärke wieder abzunehmen. Diese Phase wird als „Decay“-Phase bezeichnet. Sie dauert an, bis der Ton eine bestimmte Grundlautstärke erreicht hat. Diese Grundlautstärke wird durch den „Sustain“ beschrieben. Ist diese Grundlautstärke erreicht, beginnt die „Sustain“ Phase, in welcher der Ton eine näherungsweise gleichbleibende Lautstärke aufweist und nur sehr langsam an Lautstärke verliert. Die letzte Phase ist die so genannte „Release“-Phase. Diese beginnt, wenn die Lautstärke des Tons rapide abnimmt. Bei Tasteninstrumenten können die letzten drei Phasen relativ klar differenziert werden. Die „Decay“-Phase geht klar erkennbar in die „Sustain“-Phase über. Diese hält dann so lange an, bis die Taste wieder los gelassen wird. Die „Sustain“-Phase geht dann in die „Release“-Phase über. Bei Gitarren ist diese klare Differenzierung nicht in allen Fällen möglich. Ein Ton beginnt mit dem Anschlagen einer Saite und somit mit der „Attack“-Phase. Danach darf die Saite nicht mehr fixiert sein, da sie sonst

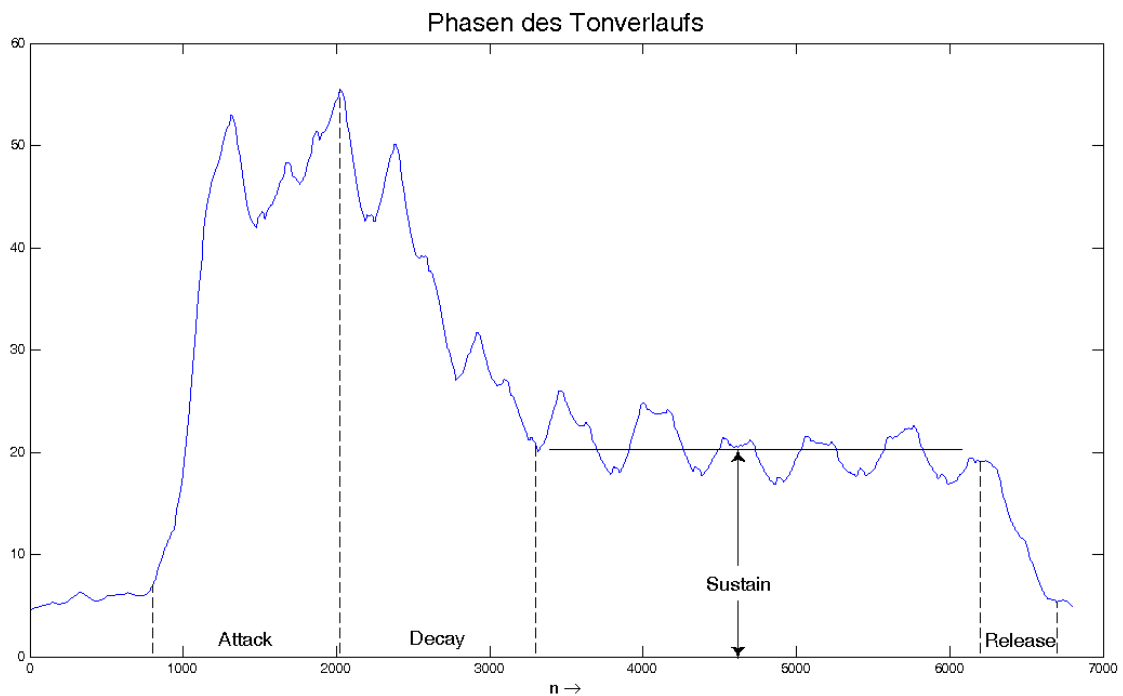


Abbildung 2.1.: Die verschiedenen Phasen des Tonverlaufs

nicht schwingen könnte. Das bedeutet, dass die „Sustain“-Phase, also das Ausschwingen des Tons, nur dann beendet werden kann, wenn die Schwingung der Saite gestoppt wird. Dadurch fällt die Lautstärke schlagartig ab, was der „Release“-Phase entspricht. Das stoppen der Saitenschwingung kann zu Störartefakten am Ende eines Tons führen.

2.2. Verwendete Audiodateien

Es werden Soloaufnahmen von drei verschiedenen Gitarren untersucht. Jede Gitarre wurde von verschiedenen Spielern gespielt. Aufgenommen wurde immer das Stück „Prelude No. 1“ vom brasilianischen Komponisten Heitor Villa-Lobos (*1887, †1959) [WEBd]. In Abbildung 2.2 wird die Partitur des Stückes dargestellt. Gespielt wurden jeweils ab „Più mosso“ die folgenden acht Takte.

Bei den bespielten Gitarren handelt es sich um Gitarren von drei verschiedenen Gitarrenbauern. Deshalb werden die Gitarren auch im Verlauf dieser Arbeit mit dem Namen des jeweiligen Gitarrenbauers bezeichnet. Unterschieden wird zwischen:

3

rit. **Più mosso**

rall. **Poco meno**

allarg. poco a poco

M.E. 6731

Abbildung 2.2.: Partitur zu „Prelude“

- Hense¹
- Marin²
- Wichmann³

Es liegen nur von drei Spielern Aufnahmen von allen drei Gitarren vor. Die anderen Spieler haben jeweils zwei der oben genannten Gitarren bespielt. Relevant wird dieser Hinweis bei der Auswertung der Ergebnisse hinsichtlich der Vergleichbarkeit der Segmentierungsrate abhängig vom jeweiligen Spieler.

2.3. Artefakttypen

Es wird in dieser Arbeit zwischen vier Artefakttypen differenziert. Ein Artefakt kann sein: ein Einzelton, ein Akkord, ein arpeggierter Akkord oder etwas Unbekanntes.

2.3.1. Einzelton

Ein Einzelton zeichnet sich dadurch aus, dass der Spieler genau eine Saite anschlägt. Dieser Anschlag muss klar von dem vorangegangenen und dem nachfolgenden Anschlag abgegrenzt sein. Ein reiner Einzelton im eigentlichen Sinne darf nicht durch andere Einflüsse verunreinigt sein.

2.3.2. Akkord

Als Akkord bezeichnet man ein Artefakt, bei dem mehrere Töne gleichzeitig gespielt werden. Im Eingangssignal sind diese von den Einzeltönen sehr schwer zu differenzieren, da der Akkord als Artefakt auch von seinem Vorgänger und Nachfolger klar abgegrenzt ist und sich nur im Frequenzbereich von einem Einzelton unterscheiden lässt. Diese Unterscheidung ist nur für spezielle Akkorde trivial durchzuführen. Im Verlauf dieser Arbeit wird dieses Unterscheidungsproblem ausführlich thematisiert und erläutert.

¹Dieter Hense

²Santiago Marin (*1946), gemessene Gitarre Baujahr 1968 (vgl. [WEBa] und [WEBb])

³Michael Wichmann, gemessene Gitarre Baujahr 2000, Hamburg (vgl. [WEBc])

2.3.3. Arpeggierter Akkord

Diese spezielle Form eines Akkords entsteht, wenn ein Akkord aufgefächert wird. Das Spielen von Akkorden in arpeggierter Variante ist ein sehr beliebtes Stilmittel unter Gitarrenspielern und gibt dem gespielten Stück eine individuelle Note. Diese Art von Artefakten kann im Rahmen dieser Arbeit zu einem relativ guten Anteil erkannt werden, da sich das Artefakt im Zeitbereich als kaskadierendes Artefakt mit mehreren Tonanfängen in kurzer Abfolge darstellt.

2.3.4. Unbekanntes Artefakt

Es kommt immer wieder vor, dass im Signal Artefakte auftauchen, bei denen die Signalenergie explosiv ansteigt, welche aber nicht die nötige Intensität, Frequenz oder Form eines Tons aufweisen. Es handelt sich hierbei um sogenannte Störgeräusche. Die Artefakte dieser Kategorie werden als „Unbekannt“ klassifiziert.

3. Gängige Verfahren zur Segmentierung von Audiosignalen

Der Schwerpunkt dieser Arbeit liegt auf der Segmentierung eines spezifischen Eingangssignals in die enthaltenen Einzeltöne. Um eine solche Segmentierung durchführen zu können, benötigt man zu jedem Ton zwei Informationen. Zum einen den Beginn des Tones („Onset“) und zum anderen das Ende („Offset“). Mit diesen beiden Informationen kann man klar definiert die Position des Tons innerhalb des Audiomaterials angeben. Das Finden des „onsets“ und des „offsets“ ist allerdings kein triviales Problem. Aufgrund der Tatsache, dass das Auffinden von bestimmten Artefakten innerhalb eines gegebenen Audiosignals nicht nur für das hier behandelte Problem eine Rolle spielt, sondern beispielsweise auch für Query-by-Humming Systeme (vgl. [ABW06],[MSW+96] und [WLL+06]), bei denen bestimmte Musikstücke anhand einer vom Suchenden vorgeschummten Melodie innerhalb kürzester Zeit in einer dazugehörigen Datenbank gefunden werden sollen, oder für Audiotools zum Schneiden von Musikstücken (vgl. [Kla99]), gibt es auf diesem Gebiet viele verschiedene Ansätze, welche grob in drei Gruppen eingeteilt werden.

In diesem Kapitel wird zuerst der allgemeine Aufbau der Transientenermittlung dargestellt (3.1). Danach wird auf die drei gängigen Verfahrensgruppen eingegangen (3.2). Es werden weitere Verfahren vorgestellt, welche sich bewusst von diesen drei gängigen Verfahren abgrenzen (3.2.4). Es werden die Anforderungen (3.3) an das zu entwickelnde System aufgeführt und zusammengefasst. Zuletzt werden die vorgestellten Verfahrensweisen bewertet und eines davon für die Entwicklung des Verfahrens dieser Arbeit ausgewählt (3.4).

3.1. Allgemeiner Aufbau

Unabhängig von dieser Einteilung wird der Vorgang der Transientenermittlung häufig in drei Phasen unterteilt (vgl. [BDA+05]):

- Vorverarbeitung
- Reduktion
- „Peak Picking“

Diese drei Phasen sollen im Folgenden erläutert werden.

3.1.1. Vorverarbeitung

Die Vorverarbeitung dient im Allgemeinen dazu, bestimmte Teilaspekte oder Eigenschaften des Audiosignals hervorzuheben oder so aufzubereiten, dass die nachgelagerten Analyseverfahren mit der veränderten Form des Signals dieses möglichst einfach und schnell verarbeiten können (Vgl. [BDA⁺05]). Diese Phase der Segmentierung ist optional und kann ausgelassen werden, wenn zum Beispiel das Eingangssignal schon in einer gut zu verarbeitenden Form vorliegt. Im Kontext der Transientensuche gilt es, in dieser Phase das Signal so aufzubereiten, dass die Abschnitte starker Veränderungen in der Signalenergie besonders hervorgehoben werden.

3.1.2. Reduktion

Dies ist die Hauptphase der Transientenermittlung. Die Informationen, welche noch in dem vorverarbeiteten Signal enthalten sind, sollen stark reduziert werden, um die anschließende Suche zu vereinfachen. Hier wird unterschieden zwischen Verfahren, welche zur Reduzierung explizit vordefinierter Eigenschaften des Signals verwendet werden, und solchen, bei denen eine Reduzierung auf Basis von Wahrscheinlichkeitsmodellen vorgenommen wird (Vgl. [BDA⁺05]).

3.1.3. „Peak Picking“

In diesem letzten Schritt werden die „Peaks“ in dem reduzierten Signal mit Hilfe einer „Peak Picking“ Funktion markiert. Bei dieser Funktion ist es wichtig, dass sie die Fähigkeit besitzt die „richtigen“ Peaks im Signal zu identifizieren, das heißt, dass sie Tonanfänge erkennt und von Störsignalen und anderen Signalfragmenten unterscheiden kann. Hierzu sind nach [BDA⁺05] drei Schritte notwendig:

- Nachverarbeitung
- Thresholding
- Peak-Picking

Im Folgenden werden die bei [BDA⁺05] dargestellten Vorgänge in den einzelnen Schritten zusammengefasst und erläutert.

Nachverarbeitung

Dieser Schritt ist, wie auch schon die Vorverarbeitung (3.1.1), optional und davon abhängig, ob das Signal schon in einer verwertbaren Form vorliegt. Ziel dieses Schritts ist es das Signal idealerweise soweit zu modifizieren, dass die „Onsets“ als klar erkennbare lokale Maxima vorhanden sind. Weiterhin soll die Bildung von Parametern für Verfahren mit adaptivem Threshold durch die Nachverarbeitung erleichtert werden.

Thresholding

Um nun aus dem resultierenden Signal die tatsächlichen Peaks zu markieren, welche „Onsets“ zu Tönen sind, ist es notwendig, einen Schwellenwert („Threshold“) zu definieren. Dieser Schwellenwert kann auf zwei Arten festgelegt werden. Entweder handelt es sich bei dem Schwellenwert um einen im Vorwege definierten festen Wert, welcher meist durch verschiedene Versuche auf das spezifische Problem abgestimmt ist, oder es wird ein Schwellenwert adaptiv auf Basis von bestimmten Eigenschaften im Signal bestimmt (vgl. [VTB04]). Diese Eigenschaften können verschiedener Natur sein und können eng mit dem Problem verknüpft sein. Ein Beispiel für einen adaptiven Schwellenwert ist in Gleichung 3.1 dargestellt. λ ist der Skalierungsfaktor, $D[n_m]$ ist ein Ausschnitt aus dem modifizierten Eingangssignal D auf dem Intervall n_m mit $n_m = [n - m, n + m]$ und δ ist eine Konstante zur Feinanpassung (vgl. [BBP04]).

$$\delta_t(n) = \lambda \cdot \text{median}(D[n_m]) + \delta \quad (3.1)$$

Peak-Picking

Beim Vorgang des „Peak-Picking“ werden nun die Peaks im Signal ausgewählt, welche oberhalb des Thresholds liegen. Diese Stellen im Signal werden dann als Onset des Tons markiert. Die Markierung des passenden „Offsets“ kann nun durch eine sukzessive Analyse des Signals ab dem jeweiligen Onset erfolgen. Diese Onset/Offset-Paare bilden dann die Markierung für das Isolieren der Töne.

3.2. Übersicht

Bei der Bestimmung des „Onsets“ wird versucht, Transienten innerhalb des Audiomaterials zu ermitteln. Transienten sind die schnell veränderlichen Teile eines Tones (vgl. 2.1). Die

Verfahren werden anhand der Art und Weise eingeteilt, auf welche diese Töne gesucht werden.

Unterteilt werden die Verfahren grob in die Gruppen:

- Signalenergiebasierte Verfahren
- Frequenzbasierte Verfahren
- Hybridverfahren

Weiterhin werden in diesem Abschnitt unter [3.2.4](#) weitere Verfahren präsentiert, welche nicht in die genannten Kategorien passen.

3.2.1. Signalenergiebasierte Verfahren

Signalenergiebasierte Verfahren zeichnen sich dadurch aus, dass sie versuchen die Transienten durch Schwankungen in der Signalenergie über die Zeit zu identifizieren. Sie stützen sich auf die Annahme, dass sich jede neu gespielte Note durch eine Zunahme der Signalenergie bemerkbar macht (vgl. [\[DBDS03\]](#)). Diese abrupte Zunahme der Signalenergie lässt in den meisten Fällen auf eine Veränderung in der gespielten Melodie schließen, da es sich hier um ein neues akustisches Signal handelt. Diese Verfahren haben allerdings den Nachteil, dass ihnen neue Töne entgehen, bei denen die Signalenergie keine signifikante Zunahme verzeichnet. Das ist bei allen Instrumenten der Fall, welche keine oder einen geringen Anteil an perkussiven Tonanfänge aufweisen (vgl. [\[BS03\]](#)). Auch ist es aufwändiger leise gespielte Tonanfänge zu erkennen, da diese im Verhältnis zu laut gespielten Passagen deutlich geringere Energiezunahmen aufweisen. Vorteilhaft sind allerdings die schnelle und einfache Implementation (vgl. [\[BS03\]](#)) und die guten Adaptionismöglichkeiten an das jeweilige Problem.

3.2.2. Frequenzbasierte Verfahren

Frequenzbasierte Verfahren analysieren den Verlauf des Signals im Frequenzbereich. Hier erzeugen Onsets eine Veränderung der im Signal vorhandenen Frequenzen. Auch geht das Auftreten von Onsets im Frequenzbereich mit einem Anstieg der Energie einher. Die meisten Verfahren dieser Gruppe betrachten den Frequenzverlauf in verschiedenen Frequenzbändern. Um den Frequenzbereich sichtbar zu machen, wird das Signal mit Hilfe der Short-time Fourier Transformation¹, welche das Zeit-Frequenz-Spektrum eines Signal ermittelt, bearbeitet. Diese wandelt das Eingangssignal mit Hilfe des Prinzips der Fourier-Transformation um.

¹STFT

Dabei wird die Fourier-Transformation jeweils auf kurze Zeitabschnitte angewendet, welche mit einer Fensterfunktion verknüpft sind. Es ergibt sich für die Berechnung der STFT die Funktion 3.2 [KSW08].

$$S(t, f) = \int_{-\infty}^{\infty} s(\tau) \cdot w(\tau - t) e^{-j2\pi f \tau} d\tau \quad (3.2)$$

Es ist notwendig eine Fensterfunktion zu verwenden, da die Fourier-Transformation davon ausgeht, dass das zu bearbeitende Signal periodischer Natur ist. Damit die betrachteten Abschnitte als periodisch angesehen werden können, müssen vorher der Beginn und das Ende des Fensters mit Hilfe einer Fensterfunktion stetig gemacht werden.

Ein Beispiel für eine solche Fensterfunktion ist das Blackman-Fenster (vgl. Gleichung 3.3 [KJ08]). N ist hierbei die Länge des Fensters für die STFT.

$$w_N(n) = 0,42 - 0,5 \cdot \cos\left(\frac{2n\pi}{N}\right) + 0,08 \cdot \cos\left(\frac{4n\pi}{N}\right), 0 \leq n \leq N - 1 \quad (3.3)$$

Zur Veranschaulichung ist das Blackman-Fenster in Abbildung 3.1, wie es in MATLAB verwendet wird, für $M = 40$ dargestellt. Zu beachten ist, dass die Werte der x-Achse von der Definition des Fensters abweichen, was damit zusammenhängt, dass in MATLAB Funktionen als Arrays aufgefasst werden. Für diese Arrays sind nur Indexe i mit $i \in \mathbb{N} \setminus \{0\}$ zugelassen.

Nachdem das Eingangssignal auf diese Weise vom Zeit- in den Frequenzbereich transferiert wurde, wird der resultierende Frequenzverlauf über die Länge des Eingangssignals auf Änderungen im hörbaren Frequenzbereich analysiert, um die Anfänge der Artefakte im Signal zu lokalisieren.

3.2.3. Hybridverfahren

Hybridverfahren versuchen die Vorteile der signalenergiebasierten und der frequenzbasierten Verfahren zu kombinieren und damit die Nachteile der beiden Verfahren zu eliminieren. Solche Verfahren nutzen zum Beispiel zunächst eine signalenergiebasierte und danach eine frequenzbasierte Onset-Bestimmung. Wie schon in den beiden vorangegangenen Abschnitten beschrieben, gibt es jeweils zur energiebasierten und frequenzbasierten Onset-Bestimmung einige Tonanfänge, die unter bestimmten Bedingungen nicht gefunden werden können. Zum Beispiel werden bei der Analyse von Violinenaufnahmen viele Tonanfänge nicht gefunden, wenn nur ein energiebasiertes Verfahren zur Anwendung kommt, da diese nicht gefundenen Tonanfänge sich nur durch eine Frequenzänderung auszeichnen und über

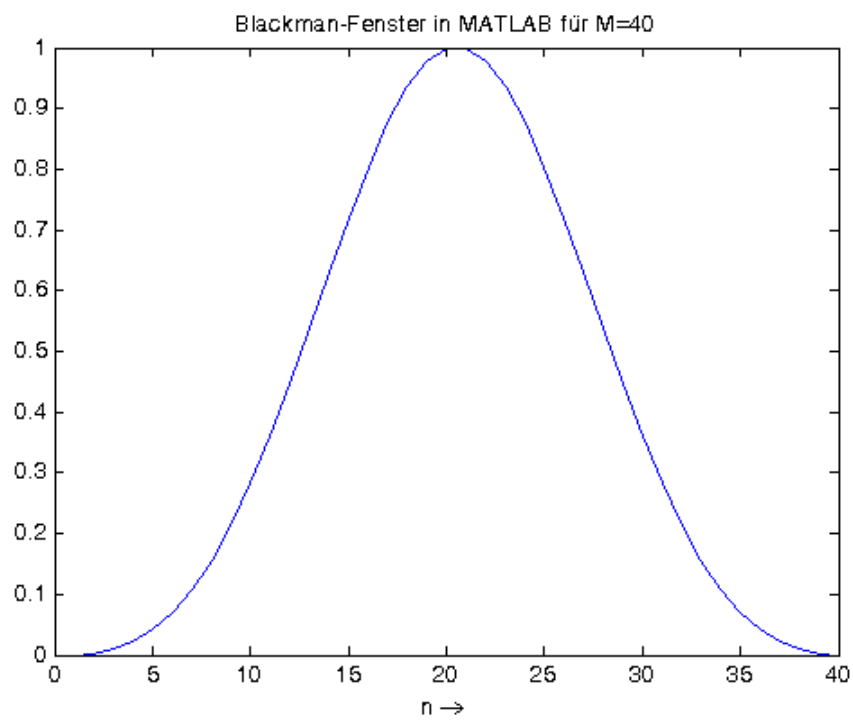


Abbildung 3.1.: Beispiel für ein Blackman-Fenster in MATLAB

keine nennenswerte Ausprägung in der Signalenergie verfügen. Da es auch für die frequenzbasierten Verfahren ähnliche Beispiele gibt, ist es sinnvoll, beide Verfahren zur Anwendung zu bringen und im Anschluss die Resultate beider Verfahren zusammen zu führen. Der Ablauf dieses Verfahren ist schematisch in Abbildung 3.2 dargestellt.

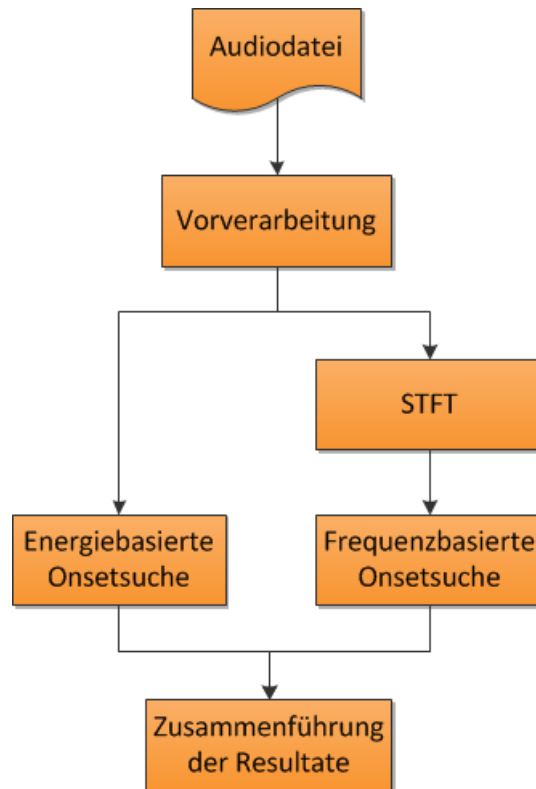


Abbildung 3.2.: Ablaufschema einer hybridbasierten Segmentierung

3.2.4. Weitere Verfahren

Neben den drei genannten Verfahrensgruppen gibt es noch weitere spezialisierte Verfahren, welche durch bestimmte gezielte Modifikation versuchen, die Verfahren noch weiter zu optimieren.

Phase Based

Dieser Typ von Segmentierungsverfahren ist eine Erweiterung eines frequenzbasierten Ansatzes (vgl. [BS03]). Das Resultat der Diskreten Fourier Transformation² wird hier als komplexe Zahl interpretiert:

$$\underline{S}(u) = R(u) + j \cdot I(u) \quad (3.4)$$

Wobei $R(u)$ den Kosinus-Anteil und $I(u)$ den Sinus-Anteil enthält. Die Phase φ ergibt sich nun als:

$$\varphi(u) = \arctan \left(\frac{I(u)}{R(u)} \right) \quad (3.5)$$

Tonanfänge werden nun auf Basis der Differenz zwischen dem erwarteten und dem tatsächlich vorhandenen Wert der folgenden Phase ermittelt. Hier kommt, wie auch schon bei den anderen Verfahren, ein Schwellenwert für die Beurteilung der Phasendifferenz zum Einsatz.

Complex Based

„Complex Based“ Verfahren erweitern die phasenbasierten Verfahren um die Amplitude der STFT. Die Amplitude ergibt sich aus:

$$|\underline{S}(u)| = \sqrt{R^2(n) + I^2(n)} \quad (3.6)$$

Aus den Gleichungen 3.5 und 3.6 lässt sich eine Komplexe Zahl erzeugen, welche sowohl die Amplituden als auch die Phasenveränderung berücksichtigt (vgl. [Dix06] und [DBDS03]).

3.3. Anforderungen

Nachdem nun verschiedene gebräuchliche Verfahren dargestellt und erläutert wurden, muss eines dieser Verfahren für die Implementierung ausgewählt werden. Das Verfahren sollte möglichst gut zu der in dieser Arbeit behandelten Problemstellung passen und einigen Anforderungen gerecht werden.

²DFT

Im Kapitel 2.1 ist gezeigt wurde, wie ein Gitarrenton im Allgemeinen aufgebaut ist. Man kann hier eine kurze, stark ausgeprägte „Attack“ Phase erkennen. Dies ist charakteristisch für Gitarrentöne, sie weisen einen eher perkussiven Tonanfang auf (vgl. [DSD02, S. 1]), was sich durch eine starke Änderung in der Signalenergie widerspiegelt.

Darüber hinaus soll das gesuchte Verfahren im Rahmen dieser Arbeit umsetzbar sein. Ideal wäre ein Verfahren, das relativ schnell und einfach zu implementieren ist, damit es möglich ist, im Laufe dieser Arbeit verwertbare Ergebnisse zu erzeugen und zu präsentieren.

Zusammengefasst sollte das Verfahren gute Resultate beim Auffinden von perkussiven Tonanfängen aufweisen. Es sollte möglichst gut nachvollziehbar sein und im Rahmen dieser Arbeit umzusetzen sein.

3.4. Bewertung

Es gibt sowohl Vorteile auf Seiten der energiebasierten Verfahren (vgl. 3.2.1) als auch der frequenzbasierten (vgl. 3.2.2). Für die energiebasierten Verfahren spräche, dass sie einfach und schnell zu implementieren sind und dass sie sehr gut geeignet sind für die Detektion von perkussive Tonanfängen. Auch frequenzbasierte Verfahren können bei perkussiven Tonanfängen zu verwertbaren Ergebnissen führen, sie sind allerdings komplizierter in der Umsetzung. Eine Kombination der beiden Verfahren, wie in Kapitel 3.2.3 vorgestellt, würde natürlich dazu führen, dass die Nachteile der beiden Verfahren minimiert werden würden. Dies würde dann allerdings zu Lasten der Performanz gehen und die Komplexität des Verfahrens zusätzlich erhöhen.

Auch wurden in Kapitel 3.2.4 verschiedene Verfahren vorgestellt, die eines der beiden Grundverfahren abwandeln, um bessere Ergebnisse zu erzielen. Dabei steigt im Allgemeinen die Komplexität des jeweiligen Verfahrens. Dazu kommt auch, dass die Resultate dieser Verfahren durch die geringe Verbreitung relativ schlecht quantifizierbar sind.

Das in dieser Arbeit zu entwickelnde Verfahren soll die genannten Anforderungen aus Kapitel 3.3 erfüllen. Zu den genannten Anforderungen passt am besten ein energiebasiertes Verfahren, da es sowohl perkussive Tonanfänge auffinden kann und relativ einfach umzusetzen ist. Auf Basis der in diesem Kapitel vorgestellten Analyse der gängigen Verfahren und Anforderungen wird im nächsten Kapitel die Entwicklung eines solchen energiebasierten Verfahrens dokumentiert.

Neben den Verfahren, welche in diesem Kapitel vorgestellt wurden, gibt es noch viele andere zum Auffinden von Tonanfängen. In einem sehr interessanten Ansatz wird zum Beispiel versucht, Tonanfänge auf Basis der Deskriptoren des 2002 verabschiedeten ISO-Standards „MPEG-7“ zu bestimmen (vgl. [SCB10]). Als Detektionsfunktion werden hier die Differenzen

verschiedener Deskriptoren zwischen aufeinanderfolgenden Fenstern verwendet. Auf diese Ansätze werden in dieser Arbeit nicht weiter verfolgt.

4. Das entwickelte Verfahren zur Segmentierung

Auf Basis der Analyse in Kapitel 3.4 ist es in dieser Arbeit notwendig, ein signalenergiebasiertes Verfahren zu entwickeln. Üblicherweise orientiert sich der Prozess der Informationsgewinnung in der Signalverarbeitung an einem festen Ablaufschema. Dieser Prozess ist in Abbildung 4.1 dargestellt. Der erste Teil des Prozesses beschreibt die Digitalisierung des analogen Eingangssignals. Hier wird ein kontinuierliches, analoges Signal auf ein zeitdiskretes, digitales Signal abgebildet. Auf diesen Teil des Prozesses wird in dieser Arbeit nicht näher eingegangen. Der zweite Teil beschreibt die Vorverarbeitung, welche notwendig ist, um das digitale Signal für gewünschte Analyseverfahren anzupassen. In diesem Schritt wird das Signal durch Normierung, Glättung und Filterverfahren für die Analyse aufbereitet. Als letztes wird, auf Basis des in Schritt 2 aufbereiteten Signals, die Analyse durchgeführt.



Abbildung 4.1.: Prozess der Informationsgewinnung in der Signalverarbeitung

In Kapitel 3.1 wurde ein allgemeiner Aufbau für Segmentierungssysteme beschrieben, so wie er bei [BDA⁺05] vorgestellt wurde. Im Rahmen dieser Arbeit soll sich das zu entwickelnde Verfahren an diesem Aufbau orientieren. Der beschriebene Ablauf ist in Abbildung 4.2 visualisiert, um den Aufbau besser zu verdeutlichen. Dieser Aufbau ist eine Spezialisierung des in Abbildung 4.1 gezeigten Ablaufs und teilt den Analyseabschnitt explizit in die Reduktion und das so genannte „Peak-Picking“.

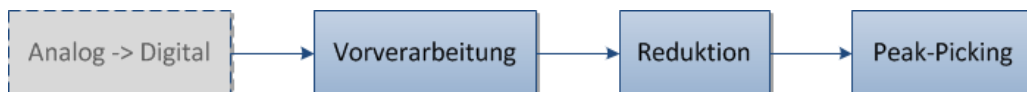


Abbildung 4.2.: Prozess der Audiosignalsegmentierung nach [BDA⁺05]

4.1. Vorverarbeitung

Die Vorverarbeitung ist bei der Arbeit mit Signalen aus der realen Welt nicht nur besonders wichtig, sondern unerlässlich. Das hat verschiedene Gründe:

1. Das Eingangssignal beinhaltet sehr viele Informationen, welche für die eigentliche Analyse irrelevant sind.
2. Die Darstellungsform, in welcher das Eingangssignal digitalisiert wurde, ist hinderlich für eine effiziente Analyse.
3. Es kann störende Elemente innerhalb des Signals geben, die beseitigt werden müssen.

4.1.1. Hervorheben von Anstiegen in der Signalenergie

Im Rahmen dieser Arbeit sollen Töne aus einer zusammenhängen Musikaufnahme gefiltert werden. Dazu ist es im ersten Schritt notwendig, die Tonanfänge zu bestimmen. Ein solcher Tonanfang zeichnet sich durch eine deutliche Zunahme der Signalenergie im Zeitbereich aus. Das Eingangssignal liegt als Signalenergie über die Zeit vor, was bedeutet, dass die Tonanfänge im Signal Orte schnellen Wachstums sind. Das entspricht großen, positiven Steigungswerten über einen längeren Zeitraum hinweg. Damit diese Orte besser zu identifizieren sind, wird auf das Signal zuerst ein Verfahren angewendet, welches in der Diplomarbeit von Henrik von Coler [Col09, S. 48,49] beschrieben wird. Dazu wird die Signalenergie jeweils vor und nach jedem Punkt n bestimmt. Dies gelingt mit Hilfe von zwei Fensterfunktionen:

$$\begin{aligned}
 E_L(n) &= \frac{1}{L} \cdot \sum_{k=n-L}^{n-1} x^2(k) \\
 E_R(n) &= \frac{1}{L} \cdot \sum_{k=n+1}^{n+L} x^2(k)
 \end{aligned}
 \tag{4.1}$$

Daraus wird bei [Col09] die Signalenergie abgeleitet als:

$$C(n) = c \cdot \log \left(\frac{E_R(n)}{E_L(n)} \right) \cdot E_R(n), c \in \mathbb{R}
 \tag{4.2}$$

Da es sich bei dem Eingangssignal in dieser Arbeit um eine Abfolge von aufgezeichneten Tönen handelt, kann nicht das erste lokale Maximum zur Bestimmung des Transientenbeginns

verwendet werden (wie in [Col09]), sondern es bedarf deutlich mehr Aufwand, wie in Kapitel 4.2 beschrieben wird. Es hat sich gezeigt, dass die besten Resultate für eine Fenstergröße von $L = 5$ und einen Gewichtungsfaktor von $c = 2$ erzielt werden können.

4.1.2. Glättung

Um Störungen im Eingangssignal zu beseitigen und die Orte der Tonanfänge noch weiter hervorzuheben, wird das Signal mit Hilfe eines gewichteten gleitenden Mittelwertes geglättet (vgl. [SS01]). Jeder Punkt n wird abgebildet auf die Summe der Werte aller benachbarter Punkte $[n - R, n + R]$ multipliziert mit einer Fensterfunktion w :

$$G(n) = \sum_{k=n-R}^{n+R} S(k) \cdot w(k - n + R + 1) \quad (4.3)$$

Für die Problemstellung dieser Arbeit wurde die Konstanten R mit $R = 200$ belegt und für w wurde ein Gauß-Fenster (vgl. Gleichung 4.4) mit einer Fenstergröße $N = 2 \cdot R + 1$ verwendet. Das verwendete Gauß-Fenster der MATLAB Bibliothek ist definiert bei [MATb] als:

$$w(n) = e^{-\frac{1}{2} \left(\alpha \frac{n}{N/2} \right)^2} \quad (4.4)$$

α wurde mit $\alpha = 2.5$ belegt und durch N wird die Fensterlänge beschrieben. Das Gauß-Fenster lässt sich aus der Standard-Gauß-Funktion (vgl. Gleichung 4.5) herleiten.

$$w(x) = \frac{P}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4.5)$$

Der Effekt dieser Glättung kann anhand von Abbildung 4.3 betrachtet werden.

4.2. Reduktion

Für die Detektion der Artefakte im Eingangssignal ist es von essentieller Bedeutung deren Anfänge zu lokalisieren. Stark vereinfacht wird ein Verfahren gesucht, welches den raschen Anstieg der Signalenergie im Eingangssignal findet. Bei [Col09] wird beschrieben, dass anschließend an die in Kapitel 4.1.1 vorgestellte Vorverarbeitung nach dem ersten lokalen Minimum gesucht werden sollte. Dies ist auf Basis eines zusammengesetzten Eingangssignals

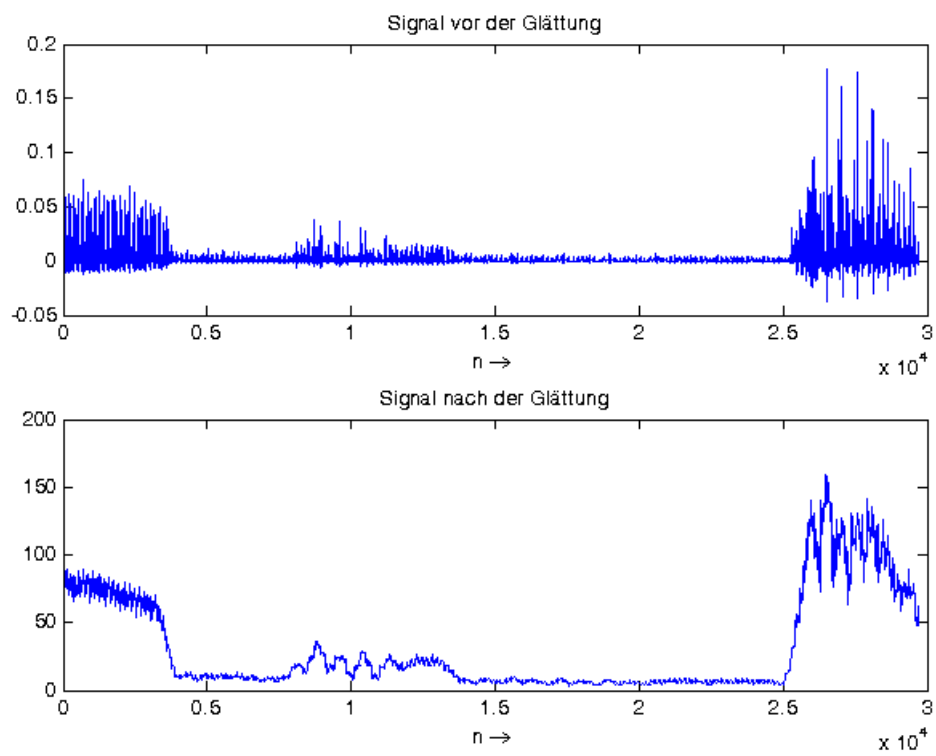


Abbildung 4.3.: Signal vor und nach der Anwendung der Glättungsfunktion

aus vielen verschiedenen Artefakten nicht möglich, da es eine beliebige Vielzahl von lokalen Minima gibt, welche als Tonanfang in Frage kommen würden (vgl. Abb. 4.4).

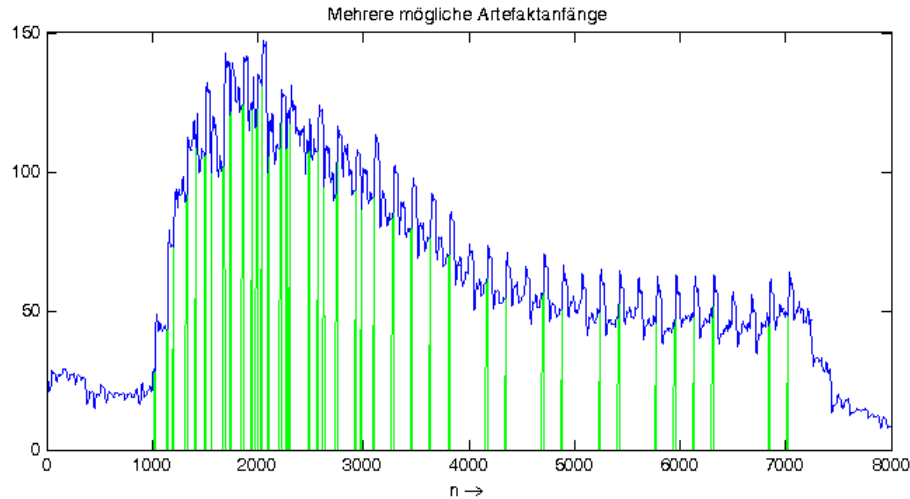


Abbildung 4.4.: Beispiele für möglicher Anfänge von Artefakten

Die Suche nach lokalen Minima auf Basis des oben beschriebenen Verfahrens ist ohne weitere Vorverarbeitung sehr rechenaufwändig und vor allem ungenau, da jeder einzelne Funktionswert in Betracht gezogen werden müsste und durch Fluktuationen in der Signalenergie sehr viele Stellen mit starker Steigung existieren. Um dieser Rechenlast und der Ungenauigkeit entgegen zu wirken und die Analyse performanter zu gestalten, wurde eine Verminderung der Funktionswerte auf Basis der Bestimmung von globalen Maxima in sich nicht überlappenden, aufeinander folgenden Fenstern bestimmter Größe vorgenommen. Dazu wird das zu untersuchende Signal in Blöcke mit einer Größe von 400 Samplen unterteilt. In diesen Blöcken wird jeweils der Index mit dem größten Funktionswert in das gefilterte Signal übernommen. Die zu analysierenden Informationen wurden somit um den Faktor 400 verringert. Das Verfahren wird in Abbildung 4.5 verdeutlicht und deckt sich mit der Beschreibung der Reduktionsphase bei [BDA⁺05].

Man könnte an dieser Stelle auch das Standard-Verfahren der Abwärtsabtastung (engl. *downsampling*) zum Einsatz bringen. Dieses Verfahren liefert ähnliche Resultate, wie das hier entwickelte Verdichtungsverfahren. Bei der Abwärtsabtastung wird das Eingangssignal zunächst mit einem Tiefpassfilter gefiltert und danach wird jeder n -te Wert in das Zielsignal übernommen. Auch bei diesem Verfahren wird die Anzahl an Informationen reduziert. Der Faktor der Reduktion kann mit Hilfe von n bestimmt werden. In MATLAB ist dieses Verfahren unter der Funktion *downsample* definiert (vgl. [MATa]).

Im Rahmen dieser Arbeit wurde darauf verzichtet dieses Verfahren zur Anwendung zu brin-

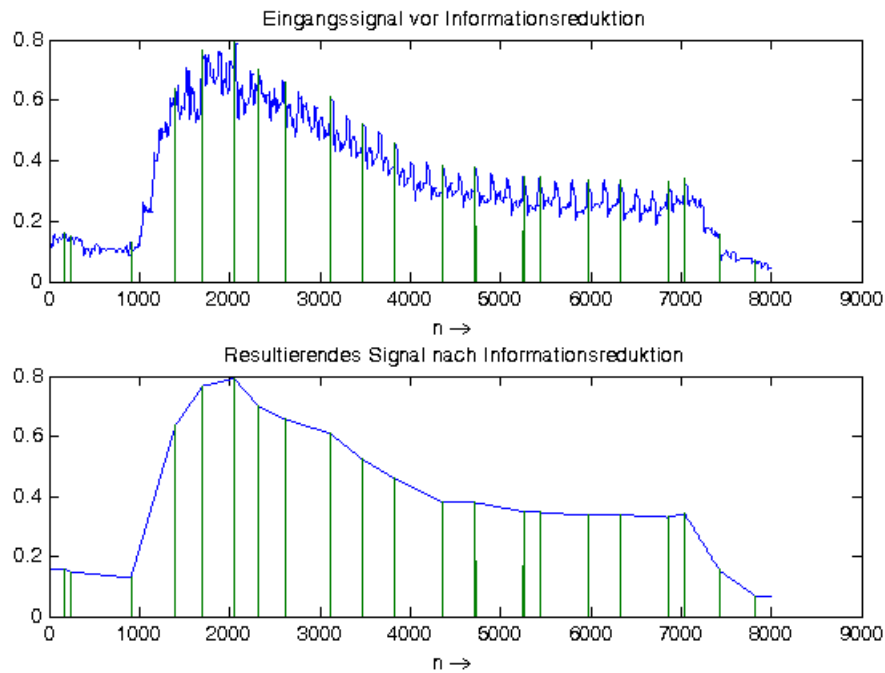


Abbildung 4.5.: Verminderung der Informationen durch Blockbildung

gen, da ein besonderer Wert auf die Pivotalisierung der gewählten Stützstellen als lokale Maxima gelegt wurde.

4.3. „Peak-Picking“

Nachdem das Datenmaterial stark reduziert wurde, sollen im nächsten Schritt die einzelnen Tonanfänge gefunden werden. Dieser Vorgang wird bei [BDA⁺05] (wie in Kapitel 3.1.3 beschrieben) in drei Unterphasen 'Nachverarbeitung - Thresholding - Peak-Picking' eingeteilt (vgl. Abb. 4.6). Die Nachverarbeitung wird bei [BDA⁺05] als Optional beschrieben. In dem hier entwickelten Verfahren ist es aus verschiedenen Gründen sehr wichtig diesen Schritt zu implementieren (vgl. dazu Kapitel 4.3.1). Das Thresholding gestaltet sich hingegen im Rahmen dieser Arbeit als relativ einfach. Das „Peak-Picking“ wird dann auf Basis einer Differenzfunktion unter Zuhilfenahme des Thresholds durchgeführt.

Abbildung 4.6.: Prozess des „Peak-Pickings“ nach [BDA⁺05]

4.3.1. Nachverarbeitung

In dem nach der Reduktion vorhandenen Signal ist es immer noch nicht ohne weitere Nachverarbeitung möglich die Artefakte klar abzugrenzen, da immer noch unterschiedlich laut gespielte Passagen in dem Audiomaterial existieren. Dadurch würden bei der Anwendung eines statischen Schwellenwerts viele Töne nicht erkannt werden können, da sie über eine zu geringe Signalenergie verfügen. Für die Analyse über das gesamte Signal ist es also erforderlich, das Eingangssignal in einzelne Blöcke von Samplen aufzuteilen, welche einzeln für sich auf eine Maximalamplitude von 1 normiert werden (vgl. Gleichung 4.6).

$$N(x) = \frac{S(x)}{\max(S)} \quad (4.6)$$

Dieses Verfahren wird genauer, je kürzer man die Länge der Blöcke ansetzt. Begründet ist dies dadurch, dass sich die Wahrscheinlichkeit mit sinkender Blockgröße verringert, innerhalb eines Blocks unterschiedlich laut gespielte Artefakte vorzufinden. Für diese Arbeit wurde die Blockgröße auf eine Länge von einer Sekunde angesetzt, was bei einer Samplefrequenz von 44100 Hz gerade 44100 Samples entspricht. Danach können die einzelnen Töne innerhalb dieses Blocks mit Hilfe eines statischen Schwellenwertes ermittelt werden.

Probleme mit Artefaktanfängen zu Beginn eines Blocks

Die Normierung und Blockbildung (s. o.) bringt unweigerlich ein neues Problem mit sich. Bei der Analyse mit dem in Kapitel 4.3 beschriebenen Verfahren kann bei einer stetigen Steigung zu Beginn des Blocks nicht gesichert entschieden werden, ob das Artefakt an dieser Stelle beginnt oder ob es die Fortsetzung eines Artefakts ist, welches in dem vorangegangenen Block schon erkannt wurde. Vorstellbar ist auch, dass die Differenz bei der Erkennung im vorangegangenen Block nicht ausreichend war, um das Artefakt zu erkennen. Um dieses Problem zu umgehen und zu gewährleisten, dass auch diese Tonanfänge erkannt werden können, wurde die Analyse dahingehend erweitert, dass zum einen eine Analyse der Segmentierung des Eingangssignals in Blöcke der Größe von 44100 Samplen beginnend am Start des Signals und zum anderen eine Analyse derselben Segmentierung um eine halbe Blockgröße nach

hinten verschoben durchgeführt wird. Die jeweiligen Ergebnisse werden dann zusammengeführt, wobei hier ein gewisser Toleranzbereich Berücksichtigung finden muss, da abhängig von unterschiedlichen Normierungen derselben Abschnitte in sich überlappenden Blöcken minimal abweichende Indizes als Artefaktanfang markiert werden. Diese Varianz wurde für die Arbeit auf $v = 25$ links und rechts des jeweiligen Artefakts gesetzt.

4.3.2. Thresholding

Für diese Anwendung wurde ein fester Schwellenwert (Threshold) auf Basis mehrerer Versuche eingestellt. In anderen Arbeiten (z.B. [BBP04]) wird auf die Verwendung eines dynamischen Thresholds zurückgegriffen. Ein dynamischer Threshold soll verhindern, dass leise gespielte Passagen im Audiostück nicht aufgrund eines statischen Thresholds ignoriert werden. Das ist bei dem hier entwickelten Verfahren nicht erforderlich, da die leisen Passagen schon durch die Blockbildung im Rahmen der Nachverarbeitung (vgl. 4.3.1) hervorgehoben wurden. Somit ist hier die Verwendung eines festen Schwellenwerts zulässig.

4.3.3. „Peak-Picking“

Beim eigentlichen Peak-Picking sollen jetzt die Offsets (also die lokalen Minima vor den Tonanfängen) mit Hilfe des Thresholds gekennzeichnet werden. Die in Frage kommenden Minima zeichnen sich dadurch aus, dass nach einem Gefälle eine sehr große Steigung in der Signalenergie folgt. Um die Suche nach den richtigen lokalen Minima so zu modifizieren, dass das Ergebnis für das hier beschriebene Problem anwendbar ist, wird die Differenz des Anstiegs untersucht, damit nicht fälschlicherweise Schwankungen in der Signalenergie als Artefaktanfänge identifiziert werden. Übersteigt diese Differenz einen bestimmten Schwellenwert, wie zum Beispiel in Abbildung 4.7, so wird die Basis der Steigungsmessung als Anfang eines Artefakts markiert.

Für die Bestimmung der Peak-Höhe wird eine abgewandelte Version einer „First Order Difference“¹ verwendet. Eine FOD betrachtet die Differenz jedes Wertes einer Funktion f zu seinem Vorgänger, also: $FOD(n) = f(n) - f(n - 1)$ (vgl. [LGWM03] und [DSD02]). Da die Differenz zwischen zwei aufeinander folgenden Samplen sehr klein ist und wenig Aussagekraft besitzt, wird die Differenz zwischen einem Minimum und dem darauf folgenden Maximum bestimmt. Auch hier können Schwankungen in der Signalenergie dazu führen, dass das folgende Maximum nicht das tatsächliche Maximum des Tonanstiegs ist. Es handelt sich damit um eine Teilspitze des Tons. Das im nächsten Abschnitt beschriebene Verfahren hilft,

¹FOD

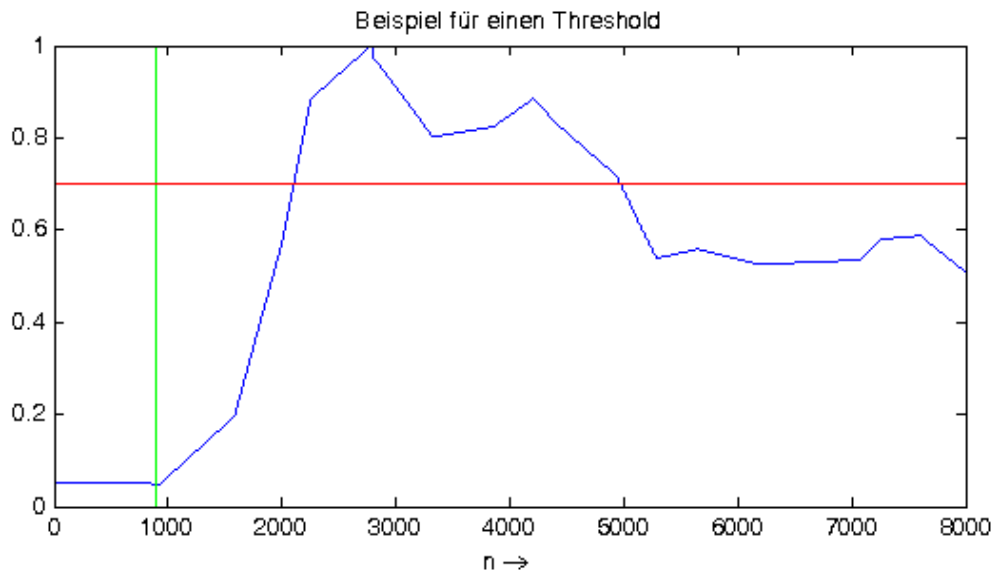


Abbildung 4.7.: Überstieg des Schwellenwertes zur Bestimmung eines Tonanfangs

dieses Problem zu minimieren. Sollten trotzdem Zwischenspitzen gefunden werden, wird deren Behandlung im nachfolgenden Abschnitt beschrieben.

Arpeggierte Akkorde

Während der Betrachtung der Ergebnisse, die von der bis hier entwickelten Version des Segmentierungsverfahrens geliefert wurden, fiel auf, dass es an einigen Stellen vermehrt zu einer Abfolge von erkannten Tonanfängen kam (vgl. Abb. 4.8). Es handelt sich bei diesen Gruppen von Tonanfängen nicht, wie vom Algorithmus deklariert, um mehrere Einzeltöne, sondern um ein einzelnes Artefakt. Das hier gefundene Artefakt entsteht, wenn ein Akkord aufgefächert wird, das heißt, die einzelnen Töne des Akkords werden mit einem leichten zeitlichen Versatz nacheinander gespielt. Dadurch entsteht ein solches Artefakt wie in Abbildung 4.8. Damit hier nur ein Artefakt erkannt wird, wurde der Segmentierungsalgorithmus dahingehend erweitert, dass ein Artefakt eine bestimmte Mindestlänge aufweisen muss. Alle weiteren gefundenen Artefakte innerhalb dieser Zeitspanne werden als so genannte „Unterartefakte“ dem ersten gefundenen Artefakt zugeordnet, um später Rückschlüsse auf die Art des Artefakts ziehen zu können.

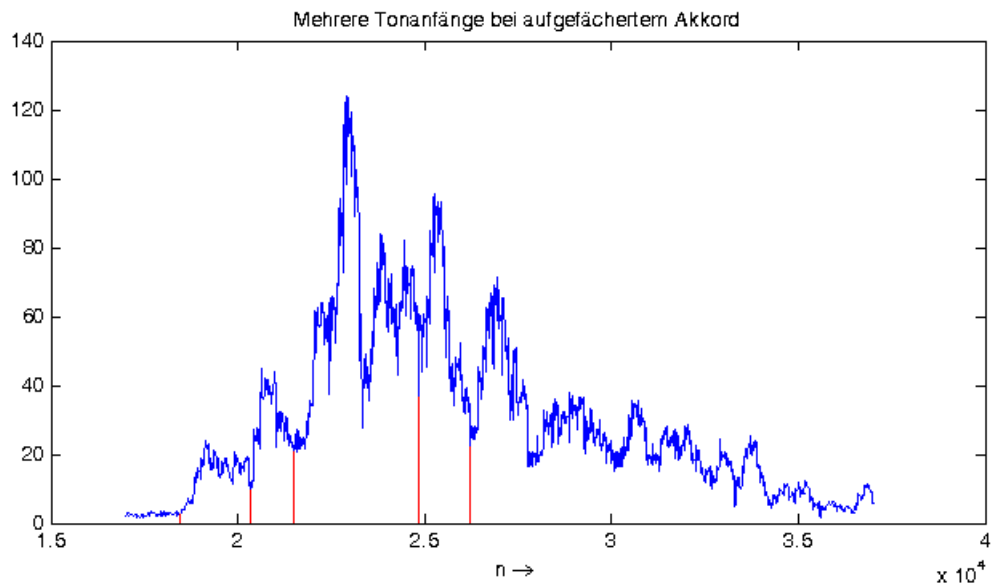


Abbildung 4.8.: Mehrere erkannte Tonaufänge in Folge (arpeggierter Akkord)

4.4. Bewertung des entwickelten Verfahrens

Das hier entwickelte Verfahren zur Detektion von Tonaufängen mag auf den ersten Blick trivial wirken, liefert aber für das in dieser Arbeit behandelte Problem ausreichend gute Resultate. Ein genaueres Bild der Leistungstärke des Verfahrens wird im nächsten Kapitel (6) vermittelt werden.

Trotz aller Bemühungen, das Verfahren so gut wie möglich an die Aufgaben anzupassen, gibt es doch noch einige bekannte Schwachstellen im Verfahren:

- Bedingt durch die Notwendigkeit zwischen tatsächlichen und falschen Artefaktanfängen zu differenzieren, musste das Verfahren mit Hilfe einer Mindestlänge für Töne implementiert werden. Dadurch kann es zu Situationen kommen, in denen der Beginn des nächsten Artefakts als Unterartefakt des vorangegangenen markiert wird.
- Eine weitere Schwachstelle ist, dass trotz blockweiser Normierung einige sehr leise gespielte Töne nicht erkannt werden. Die Rate dieser nicht erkannten Töne konnte allerdings durch die Verwendung der in Kapitel 4.3.1 beschriebenen Abwandlung des Verfahrens stark minimiert werden.

Diese Schwachstellen könnten in folgenden Arbeiten beseitigt werden. Näher wird darauf noch im Kapitel 7.2 eingegangen.

5. Analyse der Artefakte

Nachdem nun die Anfänge der Artefakte kenntlich gemacht wurden, gilt es Informationen über diese Artefakte zu ermitteln, welche Rückschlüsse auf deren Art zulassen. Diese Informationen können für spätere Arbeiten von großem Nutzen sein. Alle gewonnenen Informationen werden in Form eines Reports persistent gespeichert. Außerdem werden alle gefundene Artefakte als Audiodatei im Ausgangsverzeichnis abgelegt.

5.1. Eigenschaften der Artefakte

Zu den gefundenen Artefakten sollen Eigenschaften bestimmt werden. Im Rahmen dieser Arbeit werden Länge des Artefakts, Grundfrequenz und Grundton, Unterartefakte, Maximalamplitude und Index des Peaks bestimmt. Weiterhin soll das Spektrogramm jedes Artefakts, sowie das Artefakt selbst gespeichert werden.

5.1.1. Länge des Artefakts

Bei der Bestimmung der Länge eines Artefakts wird zuerst überprüft, ob das Artefakt eine bestimmte Mindestlänge aufweist. Nach dieser Mindestlänge wird der nächste Index im Signal ermittelt, welcher eine geringere Amplitude aufweist als die Startamplitude des Artefakts. Diese Distanz wird vorerst als Länge des Artefakts markiert. Anschließend wird überprüft, ob zwischen der Mindestlänge und dem ermittelten Ende des Artefakts weitere Artefakte markiert wurden. Sollte das der Fall sein, so wird die Länge des Artefakts definiert als die Distanz zwischen dem Beginn des Artefakts und dem des nachfolgenden.

5.1.2. Grundfrequenz und Grundton

Zur Bestimmung der Grundfrequenz und des Grundtons wird das Verfahren nach Cepstrum verwendet. Dazu wird das Verfahren so verwendet, wie es in der Veröffentlichung von Kers-

tin Dosenbach et al. [DFM08] verwendet wird¹. Der Begriff Cepstrum ergibt sich durch eine Permutation der ersten vier Buchstaben des Wortes „Spectrum“. Das Cepstrum eines Signals wird erzeugt durch die Anwendung der inversen Fouriertransformation² auf das logarithmierte Spektrum des Signals $s(t)$, wobei bei dieser Transformation der Wertebereich des Spektrums erhalten bleibt:

$$C(q) = IFFT(\log |FFT(s(t))|) \quad (5.1)$$

Die Gewinnung der Grundfrequenz aus dem Cepstrum ist möglich, da sich die Grundfrequenz des Spektrums im Cepstrum als Maximum darstellt. Bei der Rücktransformation aus dem Frequenzbereich werden die Phaseninformationen unterdrückt. Dadurch liegt der erste Impuls im Cepstrum immer an der Stelle $d = 0$ und der direkt darauf folgende Impuls bei $d = T_0$. Durch die Differenz Δd kann somit nach [DFM08] und [Hes06] die Grundfrequenz berechnet werden durch:

$$f_0 = \frac{1}{\Delta d \cdot \frac{1}{f}} = \frac{f}{\Delta d} \quad (5.2)$$

Wobei hier f als Samplefrequenz von 44100 Hz gegeben ist.

5.1.3. Unterartefakte

Als Unterartefakte werden alle Artefakte bezeichnet, welche sich im Bereich eines anderen Artefakts befinden. Sie treten vor allem bei arpeggierten Akkorden auf. Damit diese nicht fälschlicherweise als einzelne Töne erkannt werden, werden diese Artefakte dem Artefakt zugeordnet, in dessen Verlauf sie anzutreffen sind.

5.1.4. Spektrogramm

Ein Spektrogramm beschreibt die Intensitätsverteilung der Frequenzen zu bestimmten Zeitpunkten in einem Signal. Dazu wird die Short-Time Fourier Transformation genutzt, um zu jeweils kleinen überlappenden Abschnitten des Signals die Frequenzen zu bestimmen. Das erzeugte Spektrogramm wird zusammen mit der Datei abgespeichert, in welcher sich das Artefakt als Audiosignal befindet. Abbildung 5.1 ist ein Beispiel für ein solches Spektrogramm dargestellt.

¹Ausführlich beschrieben ist das Verfahren in der dazu gehörigen Diplomarbeit von Kerstin Dosenbach (vgl. [Dos07])

²IFFT

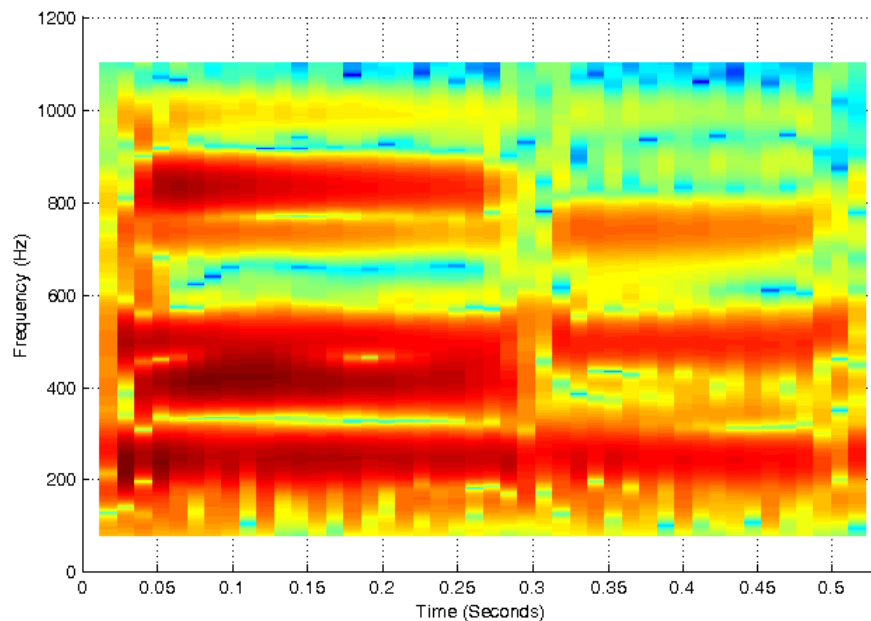


Abbildung 5.1.: Beispiel für ein erzeugtes Spektrogramm

Auf der x-Achse ist die Zeit in Sekunden angegeben. Die y-Achse beschreibt die Frequenzen in Hz. Durch die Farbe der einzelnen Punkte im Spektrogramm wird die Intensität beschrieben, mit welcher diese Frequenz im Signal zu der jeweiligen Zeit vorhanden ist.

5.2. Art der Artefakte

Es kann verschiedene Arten von Artefakten geben. Unterschieden wird in dieser Arbeit zwischen Einzeltönen, Akkorden und arpeggierten Akkorden, sowie „unbekannten Artefakten“.

5.2.1. Arpeggierte Akkorde

Ein arpeggierter Akkord ist ein Akkord, bei dem einzelne differenzierbare Töne mit einem kurzen zeitlichen Versatz gespielt werden und sich immer mehr vermischen, bis sie sich in einem Akkord vereinen. Mit Hilfe der Unterartefakte kann daher solch ein Akkord im Rahmen dieser Arbeit identifiziert werden.

5.2.2. Differenzierung zwischen Einzelton und Akkord

Bezüglich einer Differenzierung zwischen Einzelton und Akkord stellte sich heraus, dass diese Unterscheidung im Rahmen dieser Arbeit nicht mehr umgesetzt werden kann. Prinzipiell ist es möglich, im Frequenzbereich die verschiedenen Grundtöne zu isolieren. Dies gilt aber nur für einen eingeschränkten Bereich. Werden beispielsweise in einem Akkord zwei Töne gespielt, die sich um eine Oktave unterscheiden, so ist die Frequenz des einen Tones ein Vielfaches der Frequenz des anderen. Das bedeutet allerdings, dass sich dieser Grundton mit einem der Obertöne des anderen Tons überlagert und so nicht mit Sicherheit gesagt werden kann, ob es sich bei dem speziellen Artefakt um einen Einzelton oder einen Akkord handelt.

Im folgenden werden zwei Verfahren vorgestellt. Das erste ist prototypisch im Rahmen dieser Arbeit umgesetzt worden, das zweite ist als Ausblick und Erweiterungsmöglichkeit für das erste Verfahren zu verstehen und wurde hier nicht implementiert.

Iterative Betrachtung des Frequenzbereiches

Dabei handelt es sich um ein Verfahren, bei dem der erste auftretende Ton und seine Obertöne aus dem Frequenzspektrum entfernt werden. Danach wird das Frequenzspektrum dahingehend untersucht, ob noch ein weiterer Ton gefunden werden kann. Wird kein weiterer Ton gefunden, handelt es sich entweder um einen Einzelton oder einen Akkord. In diesem Fall wird angenommen, dass es sich um einen Einzelton handelt. Wird ein weiterer Ton gefunden, so beginnt das Verfahren von vorne. Es werden der zweite Ton und seine Obertöne aus dem Frequenzspektrum entfernt. Danach wird erneut nach einem dritten Ton gesucht. Im zweiten Fall handelt es sich mit großer Sicherheit um einen Akkord.

Implementiert wurde diese Verfahren so, dass durch Fast-Fourier Transformation³ das Frequenzspektrum des Artefakts bestimmt wird. Als Fensterfunktion dient hier das Hanning-Fenster (vgl. Gleichung 5.3, [KJ08]), wobei hier durch N die Länge des Fensters für die FFT beschreiben wird.

$$W_N(n) = 0.5 - 0.5 \cdot \cos\left(\frac{2\pi \cdot n}{N}\right), 0 \leq n \leq N - 1 \quad (5.3)$$

Das Spektrum wird im ersten Iterationsdurchlauf auf eine Maximalamplitude von 1 normiert, damit später ein Threshold zur Bestimmung „relevanter“ Frequenzen verwendet werden kann. Um die erste Grundfrequenz zu finden, wird das erste Maximum im Frequenzbereich

³FFT

von 80 Hz bis 700 Hz bestimmt, welches über dem Threshold liegt. Diese Frequenz und alle ihre Vielfachen werden nun aus dem Spektrum entfernt. Dazu wird iterativ die gefundene Grundfrequenz f_0 mit $n = [1, x]$, $x \cdot f_0 \leq 700$ multipliziert und zu jedem Produkt das nächste lokale Maximum gesucht. Von diesem Maximum ausgehend werden dann die nächsten p Frequenzen links und rechts davon auf „0“ gesetzt. Das so entstandene Spektrum wird mit Hilfe der Inversen-Fast-Fourier-Transformation⁴ wieder in den Zeitbereich transferiert. Mit dem so entstandenen Signal wird das Verfahren erneut aufgerufen. Zurückgegeben wird jeweils eine Liste mit gefundenen Grundfrequenzen. Der erneute Aufruf des Verfahrens mit dem gefilterten Signal erfolgt nur solange, bis drei Grundfrequenzen identifiziert wurden, da hier von einem Akkord ausgegangen werden kann.

Dieses Verfahren erkennt zwar sehr zuverlässig alle Akkorde, bei denen mindestens ein Ton aus der Menge der Töne des Akkordes nicht dieselbe Frequenz aufweist, wie die eines Obertons der anderen Töne, versagt aber bei Akkorden, bei denen diese Bedingung nicht erfüllt ist, da hier die anderen Töne des Akkordes von dem Ton mit der niedrigsten Frequenz verdeckt werden.

Betrachtung des Frequenzbereiches über die Zeit

Um nun auch die verdeckten Grundfrequenzen erkennen zu können, wäre es denkbar, ein Verfahren zu entwickeln, welches den zeitlichen Verlauf des Spektrums mithilfe der Short-Time Fourier Transformation⁵ verfolgt. Man könnte zum Beispiel das oben beschriebene Verfahren jeweils auf einen STFT transformierten Abschnitt des Artefakts anzuwenden. Es müsste beim Vergleich der Frequenzdifferenzen darauf geachtet werden, dass die Grundfrequenz im zeitlichen Verlauf eines Tones dazu tendiert, sich in Richtung der x-Achse zu verschieben. Somit müsste beim Vergleich eine gewisse Toleranz zur Anwendung kommen.

⁴IFFT

⁵STFT

6. Ergebnisse

Das in den beiden vorangegangenen Kapiteln vorgestellte Verfahren wurde für die Anwendung in der kommerziellen Software MATLAB¹ von der Firma TheMathWorks Inc.² erstellt. Diese Software ist auf dem Gebiet der Lösung mathematischer Probleme die gängige Standardsoftware. Es gibt auch frei verfügbare Alternativen zu MATLAB, auf die der Quellcode aus MATLAB mit relativ wenigen Anpassungen portierbar ist. Eine häufig verwendete Alternative ist das leistungsstarke GNU-Projekt „GNU-Octave“³.

In diesem Kapitel wird zuerst der Versuchsaufbau erläutert. Danach werden die erzielten Ergebnisse veranschaulicht und im letzten Abschnitt dieses Kapitels diskutiert und bewertet.

6.1. Versuchsaufbau

Der Programmablauf stellt sich wie in Abbildung 6.1 gezeigt dar. Die zu analysierenden Audiodateien werden im Eingangsverzeichnis hinterlegt. Danach wird aus MATLAB die Verarbeitung aufgerufen. Hier kann mit Hilfe von Parametern definiert werden, ob alle Dateien aus dem Eingangsverzeichnis verarbeitet werden sollen oder nur eine bestimmte Audiodatei. Außerdem kann auch die zu analysierende Zeitspanne eingestellt werden.

Für die Auswertung, welche zu den in diesem Kapitel vorgestellten Ergebnissen führen, wurde immer eine separate Verarbeitung je einer Audiodatei durchgeführt. Die zu analysierende Zeitspanne wurde für alle Audiodateien nicht eingeschränkt. Es wurden also alle Audiodateien einzeln und in voller Länge untersucht.

Nach dem Durchlauf des Programms werden die erzeugten Informationen persistent im Ausgangsverzeichnis (vgl. Abb. 6.1) abgelegt. Dabei ergibt sich eine Ordnerstruktur, in welcher verschiedene Informationen abgelegt werden. Existiert im Ausgangsverzeichnis kein Ordner mit dem Namen der analysierten Datei, so wird ein solcher Ordner erstellt. In diesem Ordner wird ein neuer angelegt, welcher als Name eine eindeutige Kombination aus Datum und Uhrzeit aufweist, so dass ein Überschreiben von früheren Resultaten ausgeschlossen

¹<http://www.mathworks.de/products/matlab/>

²<http://www.mathworks.de/>

³<http://www.gnu.org/software/octave/>

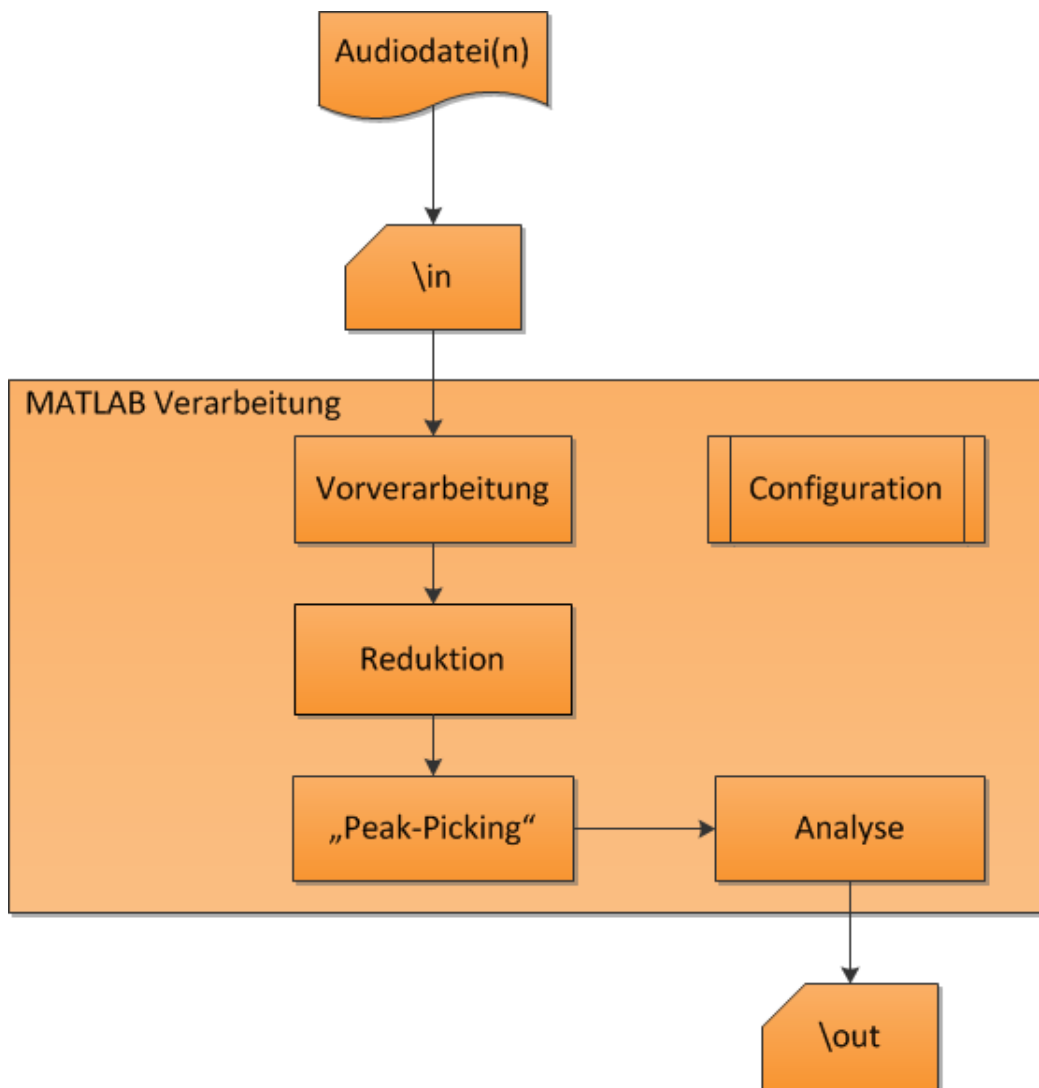


Abbildung 6.1.: Versuchsaufbau zur Analyse der Audiodateien

und zusätzlich eine sinnvolle Sortierung anhand der Ordernamen möglich ist. In diesem Ordner werden üblicherweise fünf Dateien und ein Ordner abgelegt. Der Ordner trägt den Namen „artifacts“ und enthält die einzelnen gefundenen Artefakte als „wav“-Dateien sowie als Spektrogramm. Die fünf erzeugten Dateien sind in Tabelle 6.1 dargestellt und erläutert.

Datei	Beschreibung
<i>graph.fig</i>	MATLAB Figure Datei des analysierten Signals mit markierten Tönen.
<i>graph.png</i>	Die MATLAB Figure Datei als Portable Network Graphics (PNG) Datei exportiert.
<i>{Filename}.mat</i>	Enthält eine exportierte MATLAB Variable 'res', welche die gefundenen Tonanfänge enthält. Diese Datei wird im späteren Verlauf für die Auswertung und Visualisierung der Resultate verwendet.
<i>pgm.log</i>	Enthält eine Kopie der während des Analyse geschriebenen Konsolenausgabe.
<i>report.txt</i>	Enthält den eigentlichen Analysebericht.

Tabelle 6.1.: Erzeugte Report-Dateien

Für die weitere Auswertung der Resultate ist hier die Datei „{Filename}.mat“ von Interesse. {Filename} ist hier der Name der Quelldatei ohne die Dateierdung. In dieser Datei ist die MATLAB Variable 'res' gespeichert, welche auf der gesamten Länge der Eingangsdatei die Tonanfänge indiziert. „0“ bedeutet, es wurde für diesen Index kein Tonanfang erkannt, „1“ bedeutet, dass hier vom Programm ein Tonanfang erkannt wurde.

Die Datei „report.txt“ enthält diverse Informationen über die analysierte Datei. Das Listing 6.1 zeigt den Aufbau der Datei. Zuerst wird der Name der analysierten Datei ausgegeben, danach die Spanne der Untersuchungsdauer. In diesem Fall beginnt die Analyse mit der ersten Sekunde und endet mit der neunundsechzigsten.

Listing 6.1: Beispiel Reportdatei

```

1 Report zu Audiodatei: in/hoffmannL_wichmann_solo.wav
2
3 Von: 0.000 sek
4 Bis: 69.000 sek
5
6 ## Artefakt Nr. 1 ##
7 Start:    47905
8 Ende:     52905
9 Maximum:  0.0002 (Bei 47906)
10 Grundton: e-2 (f= 83)
11 Art:      Singlenote (t= 1)
12
13 ## Artefakt Nr. 2 ##

```

```

14 Start:      62320
15 Ende:      67320
16 Maximum:   0.0068 (Bei 62321)
17 Grundton:  h-2 (f= 123)
18 Art:       Accord (t= 2)
19
20 ...

```

Danach werden die einzelnen Artefakte aufgezählt und jeweils die dazugehörigen Informationen ausgegeben (vgl. Tabelle 6.2).

Eigenschaft	Beschreibung
<i>Start</i>	Startindex des Artefakts relativ zum Anfang der Analyse.
<i>Ende</i>	Endindex des Artefakts relativ zum Anfang der Analyse.
<i>Maximum</i>	Angegeben wird die maximalen Amplitude des Artefakts im Eingangssignal und der Index des Maximums relativ zum Anfang der Analyse.
<i>Grundton</i>	Grundton des Artefakts ermittelt nach Cepstrum (vgl. Kapitel 5.1.2) namentlich und als Frequenzangabe.

Tabelle 6.2.: Artefakt-Eigenschaften

Nachdem nun die Resultate im Ausgangsverzeichnis vorliegen, werden die in der Datei `Filename.mat` hinterlegten Tonanfänge in ein Analyseprogramm geladen. Das Analyseprogramm ist ebenso wie die Verarbeitung in MATLAB geschrieben und zeigt zwei Graphen an (vgl. Abb. 6.2).

Der obere Graph zeigt das Eingangssignal normiert auf „1“ und die händisch markierten Tonanfänge. Der untere Graph zeigt im Vergleich dazu die vom Segmentierungsalgorithmus identifizierten Tonanfänge. Die angezeigte Zeitspanne und die Verweildauer der Graphen kann beim Aufruf des Programms über Parameter modifiziert werden. Alle erzeugten Graphen werden nach dem gleichen Schema wie die Sicherung der Analyseinformationen in einem Ausgangsverzeichnis zur späteren Betrachtung gespeichert. Auf Basis dieser Betrachtung kann die Trefferquote des Segmentierungsverfahrens ermittelt werden.

6.2. Resultate

Im Folgenden werden die Resultate, welche sich aus dem Versuchsaufbau (vgl. Kapitel 6.1) ergaben, dokumentiert. Dabei wird zu jeder analysierten Aufnahme in Tabelle 6.3 der Dateiname, der Spieler, die Gitarre, die totale Anzahl an erkannten Tonanfängen, die Anzahl

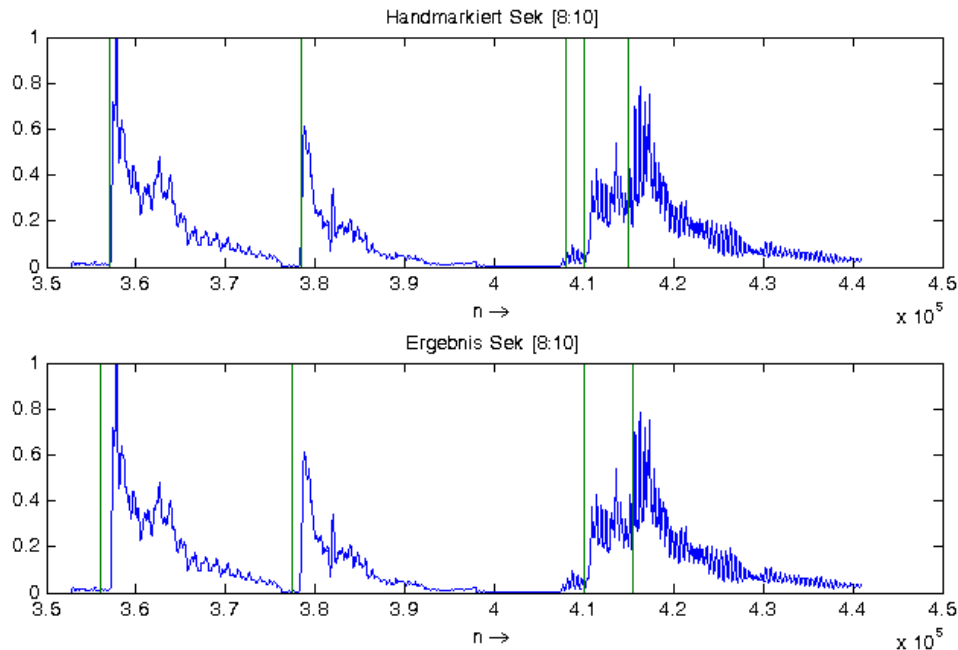


Abbildung 6.2.: Ausgabe der Auswertungsanwendung

der richtig identifizierten Tonanfänge, die Anzahl an falsch identifizierten und die Anzahl an nicht identifizierten mit deren jeweiligen prozentualen Anteilen aufgelistet.

Dabei steht $\sum V_h$ für die Anzahl der händisch markierten Tonanfänge. $\sum Res$ zeigt die Summe aller vom Algorithmus markierten Tonanfänge. Es folgt die absolute Anzahl an richtig markierten Tonanfängen (*true positives* = TP). Die direkt folgenden zwei Spalten beschreiben den prozentualen Anteil der *true positives* an der Summe der händisch ($\frac{TP}{\sum V_h}$) und automatisch ($\frac{TP}{\sum Res}$) markierten Tonanfänge. Nachstehend sind zum einen die falsch erkannten Tonanfänge (*false positives* = FP) sowie deren Anzahl an der Summe der automatisch markierten Tonanfänge ($\frac{FP}{\sum Res}$) und zum anderen die Anzahl der nicht erkannten Tonanfänge (*false negatives* = FN) und deren Anteil an der Summe der händisch markierten ($\frac{FN}{\sum V_h}$).

In der letzten Zeile wird der Durchschnitt für die jeweilige Spalte angezeigt. Es zeigt sich, dass ca. 88% aller im Audiosignal handmarkierten Einzeltöne von dem Verfahren erkannt werden. Hinzu kommen noch ca. 18% Tonanfänge, welche zu keinen Artefakten gehören und ca. 12% nicht erkannten Tonanfängen.

Als nächstes soll anhand der Ergebnisse untersucht werden, ob die Resultate des Verfahrens in Relation zu dem Spieler der Aufnahme, beziehungsweise zu der Gitarre auf welcher gespielt wurde stehen.

Spieler	Gitarre	$\sum Vh$	$\sum Res$	TP	$\frac{TP}{\sum Vh}$	$\frac{TP}{\sum Res}$	FP	$\frac{FP}{\sum Res}$	FN	$\frac{FN}{\sum Res}$
Hoffmann	Hense	113	136	106	93,81%	77,94%	30	22,06%	7	6,19%
Hoffmann	Marin	117	121	99	84,62%	81,82%	22	18,18%	18	15,38%
Hoffmann	Wichmann	120	127	111	92,50%	87,40%	16	12,60%	9	7,50%
Oba	Hense	112	138	92	82,14%	66,67%	46	33,33%	20	17,86%
Oba	Marin	107	124	95	88,79%	76,61%	29	23,39%	12	11,21%
Oba	Wichmann	91	93	78	85,71%	83,87%	15	16,13%	13	14,29%
Ossig	Hense	64	69	61	95,31%	88,41%	8	11,59%	3	4,69%
Ossig	Marin	69	69	56	81,16%	81,16%	13	18,84%	13	18,84%
Ossig	Wichmann	65	61	55	84,62%	90,16%	6	9,84%	10	15,38%
Ø		~95	~104	~84	87,63%	81,56%	~21	18,44%	~12	12,37%

Tabelle 6.3.: Ergebnisse des entwickelten Segmentierungsverfahrens

6.2.1. Spielerabhängigkeit

In der Tabelle 6.4 sind die Resultate zu den Aufnahmen dargestellt, welche von Hoffmann gespielt wurden. Es wurde noch eine weitere Zeile angefügt, welche die maximale Abweichung der ermittelten Werte vom gebildeten Durchschnitt darstellt. Diese Angabe soll ein Indikator für die Aussagekraft des Durchschnitts sein.

Gitarre	$\sum Vh$	$\sum Res$	TP	$\frac{TP}{\sum Vh}$	$\frac{TP}{\sum Res}$	FP	$\frac{FP}{\sum Res}$	FN	$\frac{FN}{\sum Res}$
Hense	113	136	106	93,81%	77,94%	30	22,06%	7	6,19%
Marin	117	121	99	84,62%	81,82%	22	18,18%	18	15,38%
Wichmann	120	127	111	92,50%	87,40%	16	12,60%	9	7,50%
Ø	~117	~128	~105	90,31%	82,39%	~23	17,61%	~11	9,69%
max Ø - X	~4	8	~6	5,69%	5,01%	~7	5,01%	~7	5,69%

Tabelle 6.4.: Ergebnisse zu den Aufnahmen von Hoffmann

Simultan dazu sind in den Tabellen 6.5 und 6.6 die Resultate für die Spieler „Oba“ und „Ossig“ aufgetragen.

Gitarre	$\sum Vh$	$\sum Res$	TP	$\frac{TP}{\sum Vh}$	$\frac{TP}{\sum Res}$	FP	$\frac{FP}{\sum Res}$	FN	$\frac{FN}{\sum Res}$
Hense	112	138	92	82,14%	66,67%	46	33,33%	20	17,86%
Marin	107	124	95	88,79%	76,61%	29	23,39%	12	11,21%
Wichmann	91	93	78	85,71%	83,87%	15	16,13%	13	14,29%
Ø	~103	~118	~88	85,55%	75,72%	30	24,28%	15	14,45%
max Ø - X	~12	~25	~10	3,40%	9,05%	16	9,05%	5	3,40%

Tabelle 6.5.: Ergebnisse zu den Aufnahmen von Oba

Gitarre	$\sum Vh$	$\sum Res$	TP	$\frac{TP}{\sum Vh}$	$\frac{TP}{\sum Res}$	FP	$\frac{FP}{\sum Res}$	FN	$\frac{FN}{\sum Res}$
Hense	64	69	61	95,31%	88,41%	8	11,59%	3	4,69%
Marin	69	69	56	81,16%	81,16%	13	18,84%	13	18,84%
Wichmann	65	61	55	84,62%	90,16%	6	9,84%	10	15,38%
\emptyset	66	~66	~57	87,03%	86,58%	9	13,42%	~9	12,97%
$\max \emptyset - X $	3	~5	~4	8,28%	5,42%	4	5,42%	~6	8,28%

Tabelle 6.6.: Ergebnisse zu den Aufnahmen von Ossig

Auf Basis der in den Tabellen 6.4, 6.5 und 6.6 vorgestellten Resultate kann man die Erkennungsraten für Tonanfänge in Audiomaterial der drei Spieler miteinander vergleichen. Um diesen Vergleich besser zu visualisieren, wurden die Erkennungsraten in Abbildung 6.3 gegeneinander aufgetragen. Auf der x-Achse sind zu jedem Spieler jeweils vier Balken dargestellt. Beschrieben werden der Anteil der richtig erkannten von allen automatisch markierten Tonanfängen, der richtig erkannten von den handmarkierten Tonanfängen, sowie jeweils der nicht und falsch markierten Tonanfänge. Auf der y-Achse befindet sich eine prozentuale Skala von 0% bis 100%, welche die jeweilige Erkennungsrate repräsentiert.

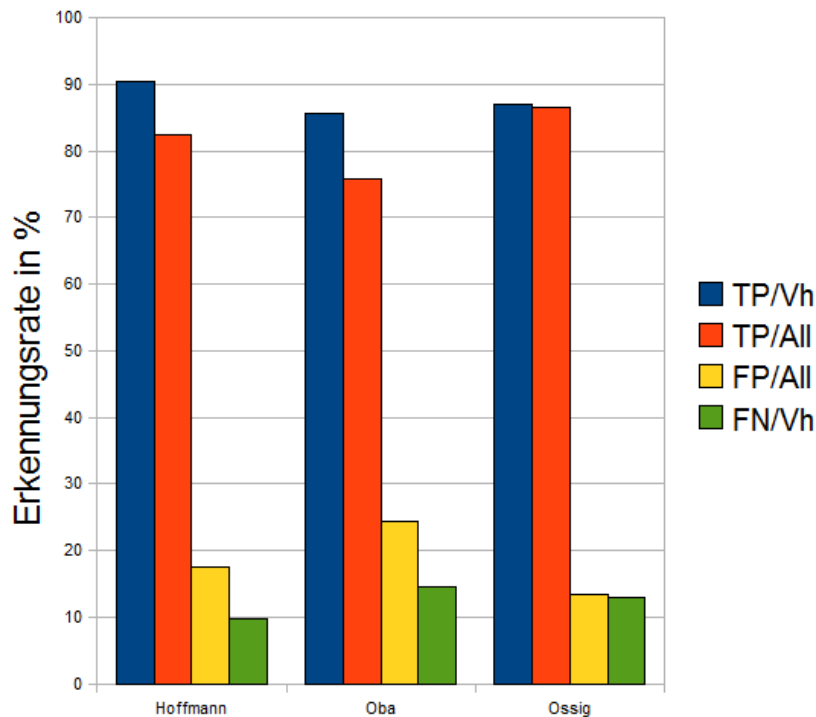


Abbildung 6.3.: Erkennungsraten zu Spielern

Es zeigt sich, dass das vorliegende Audiomaterial des Spielers „Hoffmann“ die beste Erkennungsrate für Tonanfänge, mit einer Erkennungsrate von 90,31% der handmarkierten Tonanfänge aufweist. Auch der Anteil der richtig markierten Tonanfänge der handmarkierten von 82,39% ist bei diesem Spieler ein relativ hoher Wert. Der Anteil nicht markierter Tonanfänge ist hier am geringsten mit einer Rate von 9,69%, was bedeutet, dass nur jeder zehnte handmarkierte Tonanfang nicht erkannt wurde. Trotzdem sind 17,61% der markierten Tonanfänge im Audiomaterial nicht real vorhanden.

Die zweitbeste Erkennungsrate wird für das Audiomaterial des Spielers „Ossig“ erzielt. Es wurden 87,03% aller handmarkierten Tonanfänge erkannt. 86,58% aller markierten Tonanfänge finden eine Entsprechung bei den handmarkierten Tonanfängen, damit sind nur 13,42% falsch markierte Tonanfänge. Damit wird für diesen Spieler die niedrigste Rate an falsch markierten Tonanfängen erzielt. Trotzdem werden 12,97% aller handmarkierter Tonanfänge nicht identifiziert.

Die schlechtesten Resultate liefert das entwickelte Verfahren für den Spieler „Oba“. Hier werden nur 85,55% aller handmarkierten Tonanfänge korrekt identifiziert. Darüber hinaus ist der Anteil der richtig erkannten Tonanfänge in Relation zu allen markierten verglichen mit den anderen Spielern mit einer Rate von 75,72% sehr gering. Im Schnitt findet jeder vierte Tonanfang keine Entsprechung bei den handmarkierten Tonanfängen. Daraus ergibt sich, dass die Raten der nicht und der falsch markierten Tonanfänge für diesen Spieler mit 14,45% und 24,28% deutlich höher liegen als bei den anderen beiden untersuchten Spielern.

Berücksichtigt man die maximale Abweichung vom Durchschnitt für die jeweiligen Spieler und prozentualen Anteile, so findet man hier eine relativ große Abweichung. Die maximale Abweichung vom Durchschnitt der totalen Zahlen weist dagegen für die Spieler „Ossig“ und „Hoffmann“ eine eher geringe Abweichung auf.

6.2.2. Gitarrenabhängigkeit

Nachdem die Abhängigkeiten der Ergebnisse von den Spielern analysiert wurde, soll nun die Abhängigkeit der Ergebnisse in Relation zum gespielten Gitarrenmodell betrachtet werden.

In den Tabellen 6.7, 6.8 und 6.9 sind die Resultate zu den Aufnahmen dargestellt, welche mit den Gitarren der Gitarrenbauer Hense, Marin bzw. Wichmann gespielt wurden.

Wie schon bei der vorangegangenen Untersuchung der Resultate bezüglich des Spielers, sollen auch für die einzelnen Gitarren die Erkennungsraten der Tonanfänge im Audiomaterial auf Basis der Tabellen 6.7, 6.8 und 6.9 verglichen werden. Dieser Vergleich ist grafisch in Abbildung 6.4 dargestellt. Auf der x-Achse befinden sich zu den drei untersuchten Gitarren

Spieler	$\sum Vh$	$\sum Res$	TP	$\frac{TP}{\sum Vh}$	$\frac{TP}{\sum Res}$	FP	$\frac{FP}{\sum Res}$	FN	$\frac{FN}{\sum Res}$
Hoffmann	113	136	106	93,81%	77,94%	30	22,06%	7	6,19%
Oba	112	138	92	82,14%	66,67%	46	33,33%	20	17,86%
Ossig	64	69	61	95,31%	88,41%	8	11,59%	3	4,69%
\emptyset	~96	~114	~86	90,42%	77,67%	28	22,33%	10	9,58%
$\max \emptyset - X $	~32	~45	~25	8,28%	11,00%	20	11,00%	10	8,28%

Tabelle 6.7.: Ergebnisse zu den Aufnahmen der Gitarre von Hense

Spieler	$\sum Vh$	$\sum Res$	TP	$\frac{TP}{\sum Vh}$	$\frac{TP}{\sum Res}$	FP	$\frac{FP}{\sum Res}$	FN	$\frac{FN}{\sum Res}$
Hoffmann	117	121	99	84,62%	81,82%	22	18,18%	18	15,38%
Oba	107	124	95	88,79%	76,61%	29	23,39%	12	11,21%
Ossig	69	69	56	81,16%	81,16%	13	18,84%	13	18,84%
\emptyset	~98	~105	~83	84,85%	79,86%	~21	20,14%	~14	15,15%
$\max \emptyset - X $	~29	~36	~27	3,93%	3,25%	~8	3,25%	~4	3,93%

Tabelle 6.8.: Ergebnisse zu den Aufnahmen der Gitarre von Marin

jeweils vier Erkennungsraten. Diese Erkennungsraten beschreiben den Anteil der richtig erkannten bezüglich aller automatisch markierten Tonanfänge, den der richtig erkannten aller handmarkierten Tonanfänge, sowie jeweils den der nicht und falsch markierten Tonanfänge. Auf der y-Achse befindet sich eine prozentuale Skala von 0% bis 100%, welche die jeweilige Erkennungsrate repräsentiert.

Die mit Abstand besten Resultate werden hier für die Gitarre „Hense“ erzielt. Es werden insgesamt 90,42% aller handmarkierten Tonanfänge markiert. Im Schnitt konnte nur jeder zehnte Tonanfang nicht markiert werden. Allerdings ist der Anteil der richtig markierten zu allen markierten Tonanfängen mit nur 77,67% verglichen mit den anderen beiden Gitarren am geringsten. Es wurden 22,33% falsch und 9,58% nicht markiert.

Die zweitbeste Rate wurde für die Gitarre „Wichmann“ erzielt. Hier konnten 87,61% der handmarkierten Tonanfänge erkannt werden. Der Anteil der falsch markierten Tonanfänge ist für diese Gitarre am geringsten und beziffert sich auf 12,85%, was einer Rate von 87,15% korrekt markierter Tonanfänge entspricht. Die Rate der nicht erkannten Tonanfänge beläuft sich auf 12,39%.

Für die Gitarre „Marin“ konnte nur eine Erkennungsrate von 84,85% der handmarkierten Tonanfänge erreicht werden. Somit werden 15,15% nicht erkannt. Von allen automatisch markierten Tonanfängen wurden 79,86% richtig und damit 20,14% falsch markiert.

Betrachtet man die maximalen Abweichungen vom Durchschnitt für die jeweiligen Gitarren, wird deutlich, dass die Abweichungen der totalen Zahlen, verglichen mit den Abweichungen

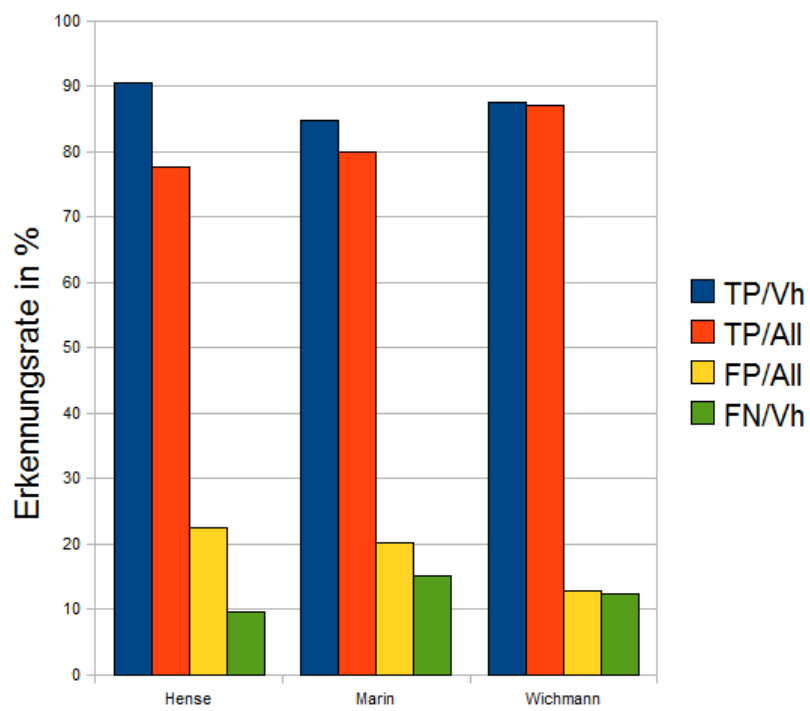


Abbildung 6.4.: Erkennungsraten zu Gitarren

Spieler	$\sum Vh$	$\sum Res$	TP	$\frac{TP}{\sum Vh}$	$\frac{TP}{\sum Res}$	FP	$\frac{FP}{\sum Res}$	FN	$\frac{FN}{\sum Res}$
Hoffmann	120	127	111	92,50%	87,40%	16	12,60%	9	7,50%
Oba	91	93	78	85,71%	83,87%	15	16,13%	13	14,29%
Ossig	65	61	55	84,62%	90,16%	6	9,84%	10	15,38%
\emptyset	92	~94	~81	87,61%	87,15%	~12	12,85%	~11	12,39%
$\max \emptyset - X $	28	~33	~30	4,89%	3,72%	~6	3,27%	~2	4,89%

Tabelle 6.9.: Ergebnisse zu den Aufnahmen der Gitarre von Wichmann

bei den Spielern im vorangegangenen Abschnitt, deutlich größer sind. Die maximale Abweichung für die prozentualen Anteile ist hingegen, mit Ausnahme der Gitarre „Hense“, sehr gering.

6.3. Diskussion der Ergebnisse

Wie sich gezeigt hat, erkennt das hier vorgestellte Verfahren durchschnittlich 87,63% der handmarkierten Tonanfänge. Von allen vom Programm markierten Tonanfängen sind im Schnitt 81,56% unter den handmarkierten wiederzufinden, was bedeutet, dass 18,44% der markierten Tonanfänge in Wirklichkeit nicht existieren. Das Verfahren bietet nun Optimierungsmöglichkeiten in zwei Richtungen: zum einen eine Steigerung der Erkennungsrate, zum anderen eine Minimierung der falsch markierten Tonanfänge.

Um die Erkennungsrate zu optimieren, können die Beispiele für nicht erkannte Tonanfänge analysiert werden. Eine stichprobenartige Analyse ergab, dass viele Tonanfänge, die nicht als solche erkannt werden, durch zu kurze Abstände zwischen den Tonanfängen (vgl. Abb. 6.5) oder leise gespielte Töne inmitten von laut gespielten (vgl. Abb. 6.6) zu erklären sind. Bei zu kurzen Abständen zwischen den Tönen wird der Folgeton von dem vorangegangenen „verdeckt“. Der Folgeton liegt in der definierten Mindestdauer für einen Ton. Bei zu leise gespielten Tönen, welche von laut gespielten umgeben sind, führt die Normierung in dem jeweiligen Block dazu, dass die Amplitudendifferenz während der „attack“-Phase nicht groß genug ist, um den Threshold zu übersteigen. In beiden Fällen bestünde die Möglichkeit, mittels verschiedener Versuche die dazugehörigen Parameter besser zu justieren. Dies würde zum einen das Verfahren stärker auf das vorliegende Audiomaterial fixieren und zum anderen würde eine feinere Einstellung der Parameter unweigerlich zu einer höheren Rate an falsch markierten Tonanfängen führen.

Um die falsch markierten Tonanfänge zu minimieren, wäre es denkbar, ein Verfahren zu implementieren, welches die vom Detektionsverfahren gelieferten Kandidaten für mögliche

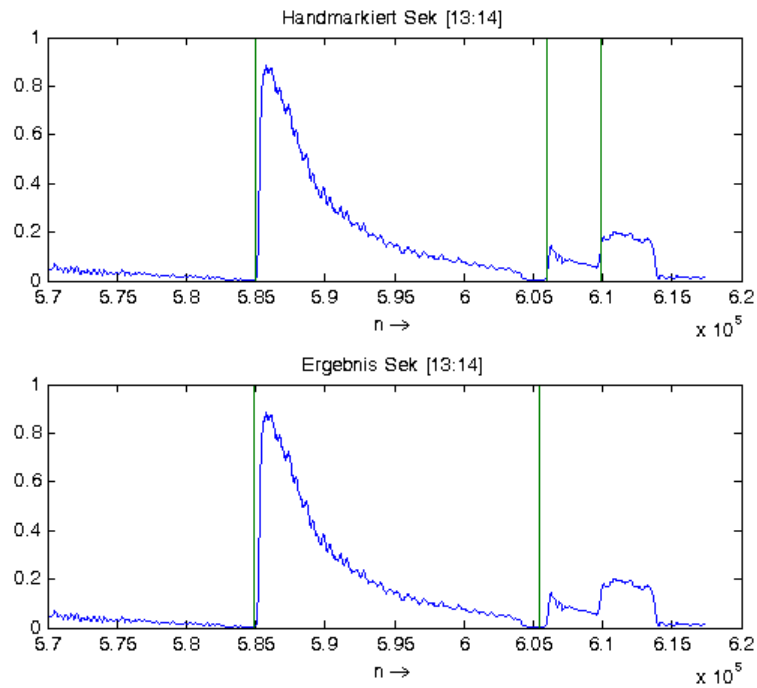


Abbildung 6.5.: Beispiel für einen verdeckten Tonanfang

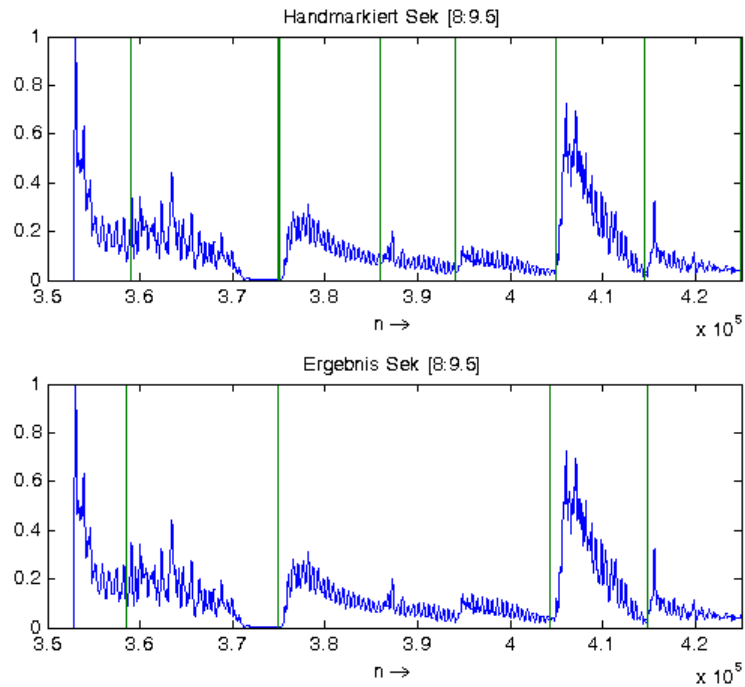


Abbildung 6.6.: Beispiel für zu niedrige Signalenergie verglichen mit Nachbartönen

Tonanfänge noch genauer untersucht, um falsche Tonanfänge schon frühzeitig zu eliminieren. Sobald hier ein leistungsstarkes Verfahren gefunden würde, wäre es durchaus möglich, dass die Parameter des Detektionsverfahrens feiner eingestellt werden könnten, sodass dadurch auch die Erkennungsrate gesteigert würde.

Die Untersuchung hinsichtlich einer Abhängigkeit zwischen Erkennungsrate und Spielern konnte zu keinem eindeutigen Ergebnis ausgewertet werden. Die jeweiligen Durchschnittswerte für die Erkennungsraten der einzelnen Spieler zeigen zwar, dass die Aufnahmen von „Hoffmann“ eine um 4,76% höhere Erkennungsrate haben als die Aufnahmen des Spieler „Oba“, doch zeigt die maximale Abweichung von diesem Durchschnitt pro Spieler durchschnittlich eine Abweichung von 5,79%. Dass diese Abweichung größer ist als die maximale Differenz der Durchschnittswerte, zeigt, dass sich eine Abhängigkeit vom Spieler im Mittel zwar tendenziell abzeichnet, aber nicht klar durch die Messwerte belegt werden kann. In Anbetracht der oben beschriebenen Probleme, welche dazu führen, dass Tonanfänge nicht erkannt werden, wäre es durchaus plausibel eine Abhängigkeit vom Spieler zu vermuten. Spielt ein Spieler gerne schneller als andere Spieler, so könnte das dazu führen, dass die Tonanfänge zu schnell aufeinander folgen und deswegen nicht erkannt werden. Diese These lässt sich mit den dargestellten Ergebnissen tendenziell untermauern, da sich die Erkennungsrate von „Hoffmann“ (90,31%) über „Ossig“ (87,03%) zu „Oba“ (85,55%) an der Spielgeschwindigkeit zu orientieren scheint.

7. Fazit und Ausblick

Rückblickend sollen an dieser Stelle die erzielten Ergebnisse noch einmal aufgegriffen und mit der Zielsetzung in Zusammenhang gebracht werden (vgl. 7.1). Zum Abschluss der Arbeit sollen Möglichkeiten der Verbesserung und der Weiterentwicklung des Verfahrens aufgezeigt werden (vgl. 7.2).

7.1. Erreichte Ziele

Ziel dieser Arbeit war es ein Verfahren zur Segmentierung von aufgezeichneten Soloaufnahmen verschiedener Konzertgitarren zu entwickeln. Zu Beginn wurde in Kapitel 2 ein kurzer Überblick über die verschiedenen Grundlagen und Gegenstände dieser Arbeit gegeben. Danach wurden gängige Verfahrensklassen vorgestellt und abgewogen. Ausgewählt wurde hier ein energiebasiertes Segmentierungsverfahren, da es sich besonders für perkussive Töne eignet.

Es wurde ein solches Verfahren implementiert und in Kapitel 4 beschrieben. Eine Validierung der Ergebnisse ergab, dass das Verfahren im Mittel eine Erkennungsrate von ca. 88% der handmarkierten Tonanfänge identifiziert. Unter der Prämisse, dass dieses Verfahren Einzeltöne liefern soll, um mit diesen das Verfahren von Dosenbach et al. [DFM08] zur Erkennung der Gitarre durchführen zu können, ist diese Rate durchaus akzeptabel. In [DFM08] wurde angeführt, dass die Klassifikationsrate für eine Kombination aus drei Tönen der selben Gitarre bei 90% läge. Im Schnitt liefert das hier entwickelte Verfahren ca. 84 Töne pro Aufnahme. Natürlich ist die Qualität der Einzeltöne nicht zu vergleichen mit der der für das Klassifikationsverfahren verwendeten Einzeltöne, dennoch sollte es auf Basis dieser Einzeltöne möglich sein, das Verfahren von Dosenbach et al. zur Anwendung zu bringen.

Die Differenzierung von Einzeltönen und Akkorden konnte in dieser Arbeit nicht zu Ende geführt werden. In Kapitel 5.2.2 und 5.2.2 wurden zwei Ansätze zur Lösung dieser Problems skizziert. Der erste Ansatz wurde prototypisch implementiert und liefern relativ vielversprechende Resultate, allerdings müsste dieser noch weiter ausgearbeitet werden, um verwertbare Resultate zu erzielen. Auch wäre möglich die Leistungsfähigkeit des Verfahrens durch das in Kapitel 5.2.2 beschriebene Verfahren zu verbessern.

7.2. Ausblick und Weiterentwicklung

Diese Arbeit bietet viel Spielraum für Folgearbeiten, Erweiterungen und Verbesserungen.

Das in dieser Arbeit vorgestellte Verfahren basiert auf einer Analyse des Audiomaterials auf Energiebasis. Ein interessanter Ansatz wäre es zu untersuchen, ob ein frequenzbasiertes Verfahren oder gar ein Hybridverfahren vergleichbare oder sogar besser Ergebnisse leistet.

Auch sollte versucht werden, das hier vorgestellte Verfahren zu verbessern, indem man zum Beispiel das Detektionsverfahren zu optimieren versucht. Wichtig zu entwickeln wäre eine Nachanalyse der markierten Tonanfänge, um Aussagen über die Qualität des gefundenen Artefakts treffen zu können. Somit könnte die Zahl der *false positives* deutlich gesenkt werden. Weiterhin bestünde so auch die Möglichkeit, dass mehr *true positives* detektiert werden können, da diese Teilweise durch *false negatives* verdeckt werden. Ansätze zur Optimierung wurden in Kapitel 6.3 dargelegt.

Die in Kapitel 5.2.2 und 5.2.2 vorgestellten Verfahren zur Differenzierung von Einzeltönen und Akkorden sind in dieser Arbeit über den Entwicklungsstand eines Prototypen nicht hinaus gekommen. Die beiden Verfahren könnten in einer folgenden Arbeit weiterentwickelt und verglichen werden. Da nur das erste vorgestellte Verfahren implementiert wurde, kann nicht abschließend bewertet werden, welches der beiden Verfahren leistungsstärker ist. Dennoch liegt die Vermutung nahe, dass das zweite Verfahren bessere Resultate liefern würde. Es erweitert das erste Verfahren und identifiziert somit die Akkorde, welche vom Ersten gefunden werden und darüber hinaus noch solche, welche das erste Verfahren nicht bestimmen kann.

Wie in der Einleitung erwähnt, ist der Hintergrund zur späteren Klassifikation der Gitarre auf Basis des bei Dosenbach et al. (vgl. [DFM08]) vorgestellten Verfahrens die Segmentierung des Audiomaterials. Eine Folgearbeit sollte also versuchen, Merkmale über die aus diesem Verfahren extrahierten Einzeltöne zu bilden und diese mit dem Verfahren von Dosenbach zu klassifizieren.

Denkbar wäre es auch, die gespielten Lieder anhand der Partitur und des hier vorgestellten Segmentierungsverfahrens auf Richtigkeit zu überprüfen. Dieser Ansatz entspricht in etwa der Entwicklung einer automatischen Korrekturfunktion.

Es gibt sehr viele Anwendungsgebiete für das hier entwickelte Verfahren. Dem entsprechend sind auch die Möglichkeiten der Weiterentwicklung und Verwendung eines solchen Verfahrens sehr vielfältig. Von Klassifikation über Audioanalyse zu Partiturabgleich und Query-by-humming Suchsystemen sind der Anwendung keine Grenzen gesetzt.

Literaturverzeichnis

- [ABW06] ADAMS, Norman H. ; BARTSCH, Mark A. ; WAKEFIELD, Gregory H.: Note segmentation and quantization for music information retrieval. In: *IEEE Transactions on Audio, Speech, and Language Processing* 14 (2006), Jan., Nr. 1, S. 131 – 141. – ISSN 1558–7916. – <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.124.9688&rep=rep1&type=pdf>
- [BBP04] BROSSIER, Paul ; BELLO, Juan P. ; PLUMBLEY, Mark D.: Real-time temporal segmentation of note objects in music signals. In: *Proc. of the Int. Computer Music Conference*. Miami, Florida, USA, Nov. 1–6 2004. – http://homepages.nyu.edu/~jb2843/Publications_files/Brossier-ICMC-2004.pdf
- [BDA⁺05] BELLO, Juan P. ; DAUDET, Laurent ; ABDALLAH, Samer ; DUXBURY, Chris ; DAVIES, Mike ; SANDLER, Mark B.: A Tutorial on Onset Detection in Music Signals. In: *IEEE Transactions on Speech and Audio Processing* 13 (2005), Sept., Nr. 5, S. 1035 – 1047. – ISSN 1063–6676. – http://homepages.nyu.edu/~jb2843/Publications_files/Bello-TSAP-2005.pdf
- [BS03] BELLO, Juan P. ; SANDLER, Mark: Phase-based note onset detection for music signals. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* Bd. 5, 2003. – ISSN 1520–6149, S. V – 441–4 vol.5. – <http://138.37.35.209/people/juan/Documents/Bello-ICASSP-2003.pdf>
- [Col09] COLER, Henrik von: *Ein Verfahren zur Synthese von Gitarrenklängen mit separater Modellierung der tonalen und der transienten Anteile*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2009
- [DBDS03] DAXBURY, Chris ; BELLO, Juan P. ; DAVIES, Mike ; SANDLER, Mark: Complex domain onset detection for musical signals. In: *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-03)*. London, UK, Sept. 8–11, 2003. – <http://www.elec.qmul.ac.uk/dafx03/proceedings/pdfs/dafx81.pdf>

- [DFM08] DOSENBACH, Kerstin ; FOHL, Wolfgang ; MEISEL, Andreas: Identification of individual guitar sounds by support vector machines. In: *Proc. of the 11th Int. Conference on Digital Audio Effects (DAFx-08)*. Espoo, Finland, Sept. 1–4, 2008. – http://www.acoustics.hut.fi/dafx08/papers/dafx08_56.pdf
- [Dix06] DIXON, Simon: Onset Detection Revisited. In: *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*. Montreal, Quebec, Canada, Sept. 18–20 2006, S. 133–137. – http://www.dafx.ca/proceedings/papers/p_133.pdf
- [Dos07] DOSENBACH, Kerstin: *Klassifikation von Audiosignalen mit Support Vector Machines*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2007. http://opus.haw-hamburg.de/volltexte/2007/391/pdf/DA_Dosenbach.pdf
- [DSD02] DAXBURY, Chris ; SANDLER, Mark ; DAVIES, Mike: A hybrid approach to musical note onset detection. In: *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-02)*. Hamburg, Germany, Sept. 26–28 2002, S. 33–38. – http://www2.hsu-hh.de/ant/dafx2002/papers/DAFX02_Duxbury_Sandler_Davis_note_onset_detection.pdf
- [Hes06] HESS, Wolfgang: Grundlagen der Sprachsignalverarbeitung. (2006). http://www.ikp.uni-bonn.de/lehre/informationen-materialien/informationen-und-materialien-kopho/materialien-1/hess/grundlagen-der-sprachsignalverarbeitung/gsv_4b.pdf/at_download/file. – Zugriff: Juli, 2011
- [KJ08] KIENCKE, U. ; JÄKEL, H.: *Signale und Systeme*. Oldenbourg, 2008 <http://books.google.de/books?id=tmiFhy8VNe8C>. – ISBN 9783486587340
- [Kla99] KLAPURI, Anssi: Sound onset detection by applying psychoacoustic knowledge. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* Bd. 6, 1999, S. 3089–3092. – http://www.ece.ul.ie/homepage/j_walker/papers/NAE_smac05.pdf
- [KSW08] KIENCKE, Uwe ; SCHWARZ, Michael ; WEICKERT, Thomas: *Signalverarbeitung: Zeit-Frequenz-Analyse und Schätzverfahren*. Oldenbourg, 2008 <http://books.google.de/books?id=hS6E94yp-rkC>. – ISBN 9783486586688
- [LGWM03] LIU, Ruolun ; GRIFFITH, Niall ; WALKER, Jacqueline ; MURPHY, Peter: Time domain note average energy based music onset detection. In: *Proc. of the Stockholm Music Acoustics Conference (SMAC 03)*. Stockholm, Sweden, Aug. 6–9,

- 2003, S. 553–556. – http://www.ece.ul.ie/homepage/j_walker/papers/NAE_smac05.pdf
- [MATa] *Decrease sampling rate by integer factor - MATLAB.* <http://www.mathworks.com/help/toolbox/signal/downsample.html>. – Zugriff: Juli, 2011
- [MATb] *Gaussian window - MATLAB.* <http://www.mathworks.com/help/toolbox/signal/gausswin.html>. – Zugriff: Juli, 2011
- [MSW⁺96] MCNAB, Rodger J. ; SMITH, Lloyd A. ; WITTEN, Ian H. ; HENDERSON, Clare L. ; CUNNINGHAM, Sally J.: Towards the digital music library: tune retrieval from acoustic input. In: *Proceedings of the first ACM international conference on Digital libraries*. New York, NY, USA : ACM, 1996 (DL '96). – ISBN 0–89791–830–4, S. 11–18. – <http://doi.acm.org/10.1145/226931.226934>
- [SCB10] SMITH, D. ; CHENG, E. ; BURNETT, I. S.: Musical Onset Detection using MPEG-7 Audio Descriptors. In: *Proc. of the 20th Int. Congress on Acoustics (ICA 2010)*. Sydney, Australia, Aug. 23–27 2010. – http://www.acoustics.asn.au/conference_proceedings/ICA2010/cdrom-ICA2010/papers/p884.pdf
- [SS01] SCHLITTGEN, Rainer ; STREITBERG, B.H.J.: *Zeitreihenanalyse*. Oldenbourg, 2001 (Oldenbourg Lehr- und Handbücher der Statistik). <http://books.google.de/books?id=FdpX9yt5ZS0C>. – ISBN 9783486257250
- [Tom10] TOMECEK, Stephen M.: *Music*. Infobase Publishing, 2010 (Experimenting With Everyday Science). <http://books.google.de/books?id=T6ub1HJiBigC>. – ISBN 9781604131697
- [VTB04] VELIKIC, Gordana ; TITLEBAUM, Edward L. ; BOCKO, Mark F.: Musical note segmentation employing combined time and frequency analyses. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)* Bd. 4, 2004. – ISSN 1520–6149, S. IV–277 – IV–280 vol.4. – http://www.ece.umassd.edu/Faculty/acosta/ICASSP/Icassp_2004/pdfs/0400277.pdf
- [WEBa] *Francisco Paco Santiago Marin.* <http://www.guitarsint.com/detail.cfm/id/484>. – Zugriff: Juli, 2011
- [WEBb] *Gitarrenbau Michael Wichmann.* <http://www.wichmann-guitars.de/handel.html#meister>. – Zugriff: Juli, 2011
- [WEBc] *Gitarrenbau Michael Wichmann.* <http://www.wichmann-guitars.de>. – Zugriff: Juli, 2011

- [WEBd] *Villa-Lobos, Heitor (1887-1959) | Heitor Villa-Lobos Website*. <http://www.villalobos.ca/node/2015>. – Zugriff: Juli, 2011
- [WLL⁺06] WU, Xiao ; LI, Ming ; LIU, Jian ; YANG, Jun ; YAN, Yonghong: A Top-down Approach to Melody Match in Pitch Contour for Query by Humming. In: *Proceedings of the 5th International Symp. Chinese Spoken Language Process*, 2006. – <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.110.1802&rep=rep1&type=pdf>
- [ZA02] ZÖLZER, Udo ; AMATRIAIN, Xavier: *DAFX: digital audio effects*. Wiley, 2002. – ISBN 0-471-49078-4

A. Quellcode

Im folgenden sollen zum einen der allgemeine und spezielle Aufbau des erzeugten Programms und zum anderen ausgewählte Ausschnitte aus dem Quellcode dargestellt werden. Darüber hinaus werden in Abschnitt [A.5](#) die Quellcodeteile vorgestellt, welche von anderen Autoren übernommen wurden.

A.1. Allgemeiner Aufbau

Der allgemeine Aufbau ist in Abbildung [A.1](#) dargestellt. Die Module „Vorverarbeitung“, „Detektion“ und „Analyse“ werden in den folgenden Abschnitten näher erläutert. Der Programmablauf richtet sich in der Reihenfolge nach den Nummern (1) bis (5). Die Hauptsteuerung der Verarbeitung wird in „process.m“ beschrieben. Für das Abspeichern der Ergebnisse ist „saveResults.m“ zuständig.

A.2. Programm zur Vorverarbeitung

Die Vorverarbeitung gliedert sich in zwei Aufrufe (vgl. [A.1](#)). Abbildung [A.2](#) zeigt den Aufbau bzw. Inhalt des Moduls zur Vorverarbeitung. Zunächst wird das Signal in „preprocessing.m“ (vgl. [A.2](#)) mit Hilfe des bei Henrik von Coler [[Col09](#)] beschriebenen Verfahrens vorverarbeitet und danach mit einem gleitenden Mittelwert geglättet. Die Verfahren sind in Kapitel [4.1](#) beschrieben. Danach wird das Signal in „normalize.m“ (vgl. [A.3](#)) auf eine Maximalamplitude von 1 normiert und in einer globalen Variablen abgelegt.

Listing A.1: Aufruf der Vorverarbeitung aus process.m

```
30 ...
31 % Preprocessing
32 signal = signal(tstart*FS+1:tend*FS+1); % windowing
33 O_TONE = signal; % store original tone
34 signal = preprocessing(signal); % do preprocess
35 N_SIGNAL = normalize(signal); % normalize signal
36 ...
```

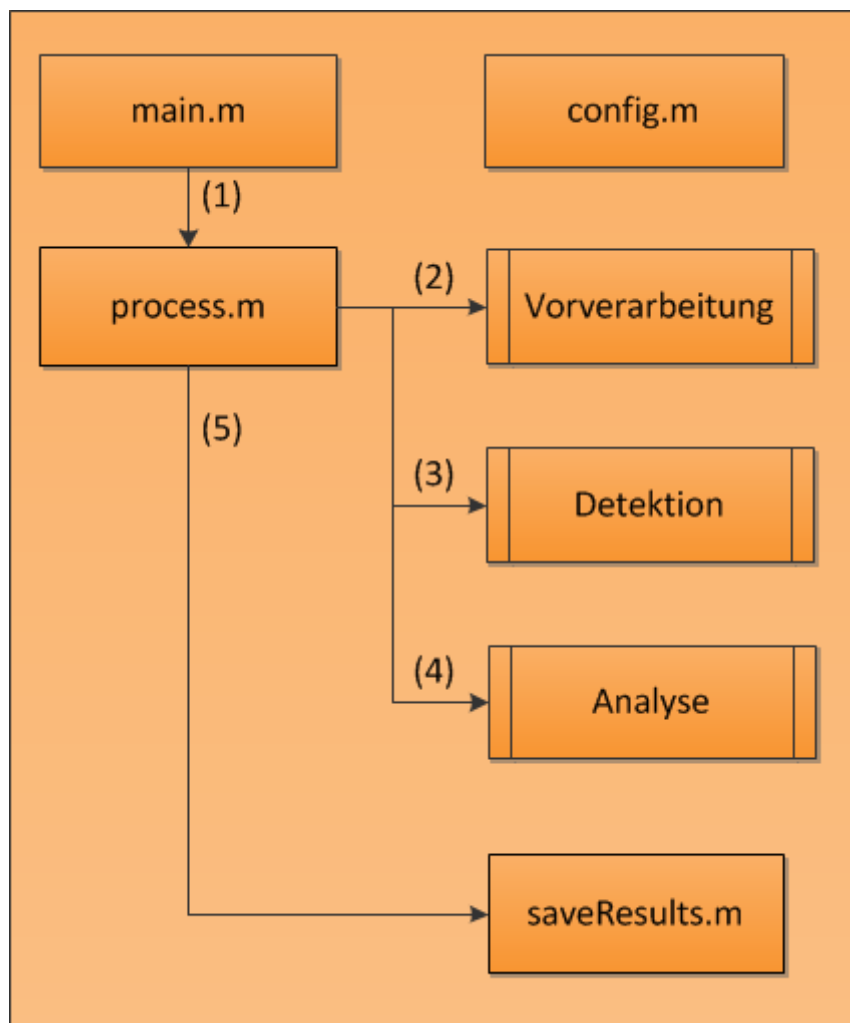



Abbildung A.1.: Systemarchitektur des erzeugten Programms

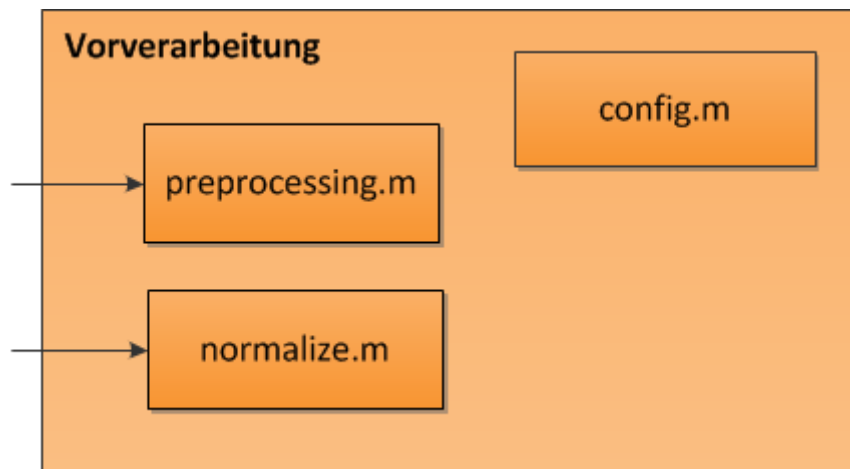


Abbildung A.2.: Systemarchitektur der Vorverarbeitung

Listing A.2: preprocessing.m

```

1  % Function for preprocessing
2  function [signal]=preprocessing(signal)
3      global c L range weight;
4
5      % for transient detection
6      squaredSignal = arrayfun(@(x) x*x, signal);
7      C = zeros(size(signal));
8      for n=1+L:length(signal)-L
9          % Calculate left window and right window
10         el = 1/L * sum(squaredSignal(n-L:n-1));
11         er = 1/L * sum(squaredSignal(n+1:n+L));
12         % Calculate transient detection function
13         C(n) = c * log(er/el) * er;
14     end
15     signal = C;
16
17     % equilisation
18     W = gausswin(2*range+1) .* weight;
19     C = zeros(size(signal));
20     for n=range+1 : length(signal)-range
21         C(n) = sum(signal(n-range:n+range) .* W);
22     end;
23     signal = C;
24 end

```

Listing A.3: normalize.m

```

1 % Normalizes an input to max. amplitude = 1
2 function [signal]=normalize(signal)
3     signal = signal/max(signal);
4 end

```

A.3. Programm zur Erkennung von Artefaktanfängen

Die Detektion der Artefaktanfänge wird aus „process.m“ nach der Vorverarbeitung aufgerufen (vgl. Listing A.4). Der Aufbau der Detektion ist in Abbildung A.3 dargestellt. Zunächst wird in „detect.m“ (vgl. Listing A.5) das Signal in Blöcke bestimmter Länge eingeteilt, wie in Kapitel 4.3 beschrieben. In jedem dieser Blöcke wird nach möglichen Tonanfängen gesucht. Diese Suche wird über „mainArtifacts.m“ (vgl. Listing A.6) gesteuert. Hier wird zunächst bei (1) die Konfiguration blockspezifisch umgestellt. Danach wird der Block für sich genommen normalisiert (vgl. Listing A.3). Damit die Tonanfänge einfacher und schneller gefunden werden können, wird die Datenmenge stark reduziert. Dies geschieht in „slicedPeaks.m“ (vgl. Listing A.7). Im Anschluss ist der Block soweit vorbereitet, dass er auf Tonanfänge analysiert werden kann. Mögliche Kandidaten für Tonanfänge werden in „findArtifacts.m“ markiert. Der Quellcode dieser Funktion ist im Listing A.8 dargestellt.

Listing A.4: Aufruf der Detektion aus process.m

```

36 ...
37 % Detect artifacts
38 artifacts = detect(tstart,tend);
39 ...

```

Listing A.5: detect.m

```

1 function [artifacts] = detect(tstart, tend)
2     global signal FS;
3
4     % find onset candidates
5     artifacts = zeros(size(signal));
6     for n=0:ceil(tend)-floor(tstart)-1
7         % set block ranges
8         if n == 0, from1 = 1/FS; from2 = 0.5;
9         else from1 = n; from2 = n+0.5; end
10        if n+1 > tend-tstart, to1 = tend-tstart-1/FS;
11        else to1 = n+1; end
12        if n+1+0.5 > tend-tstart, to2 = -1;
13        else to2 = n+1+0.5; end
14

```

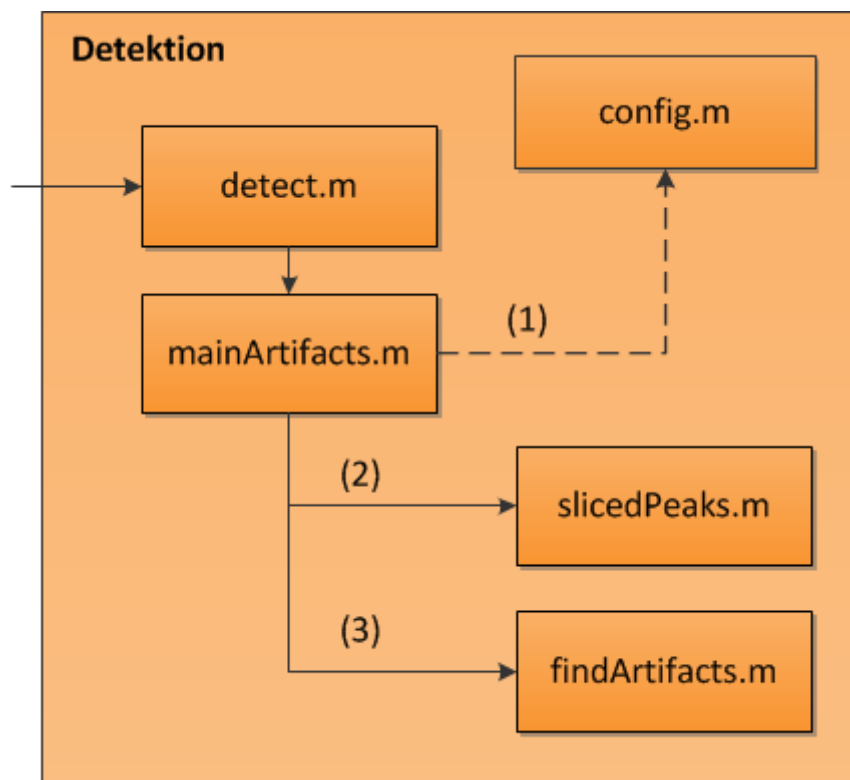


Abbildung A.3.: Systemarchitektur der Detektion

```

15     % find onset candidates in first block + match
16     temp_artifacts = mainArtifacts(from1,tol);
17     artifacts      = match(temp_artifacts, artifacts);
18
19     % find onset candidates in overlapping block + match
20     temp_artifacts = mainArtifacts(from2,to2);
21     artifacts      = match(temp_artifacts, artifacts);
22     end
23 end
24
25 % Find matching onsets to avoid duplicate tone onsets.
26 % Sets the index in artifacts to 1 if it matched and
27 % to 0.5 if it didn't.
28 function [artifacts] = match(t_artifacts, artifacts)
29     global MATCHING_RANGE FS;
30
31     for z=1:length(t_artifacts)
32         if t_artifacts(z) > 0
33             match = 0;
34
35             % Look for matching onsets
36             for u=-MATCHING_RANGE:MATCHING_RANGE;
37                 if artifacts((n+0.5)*FS+z+u) ~= 0
38                     artifacts((n+0.5)*FS+z+u) = 1;
39                     match = 1;
40                 end
41             end
42
43             % No match? Set to 0.5
44             if match == 0;
45                 artifacts((n+0.5)*FS+z) = 0.5;
46             end
47         end
48     end
49 end

```

Listing A.6: mainArtifacts.m

```

1 function [peakArtifacts]=mainArtifacts(secf,sect)
2     global l r signal tsignal;
3
4     % Configuration
5     config(secf, sect);
6
7     % Signal Modifications

```

```

8     tsignal = signal(l+1:r+1);    % block building
9
10    % normalisation
11    tsignal = normalize(tsignal);
12
13    % get onset candidates
14    peaks      = slicedPeaks(tsignal);
15    peakArtifacts = findArtifacts(peaks);
16 end

```

Listing A.7: slicedPeaks.m

```

1 % finde peaks in windows of range {DET_PEAK_RANGE}
2 function [peaks]=slicedPeaks(signal)
3     global DET_PEAK_RANGE;
4
5     peaks = zeros(size(signal));
6     for n=0:floor(length(signal)/DET_PEAK_RANGE)
7         t_start = n*DET_PEAK_RANGE + 1;
8         if((n+1)*DET_PEAK_RANGE < length(signal))
9             t_end = (n+1)*DET_PEAK_RANGE;
10        else
11            t_end = length(signal);
12        end
13        [maxval,imax] = max(signal(t_start:t_end));
14        peaks(imax+t_start) = maxval;
15    end
16 end

```

Listing A.8: findArtifacts.m

```

1 % detects artifacts within a peak signal
2 function [artifacts]=findArtifacts(peaks)
3     global ART_THRESHOLD MAX_ATTACK_SPAN tsignal;
4
5     last = -1;
6     base = -1;
7     artifacts = zeros(size(peaks));
8
9     for n=1:length(peaks)
10        if peaks(n) > 0
11            % Is there a tone base defined
12            if base > 0
13                % If the last peak wasn't the estimated tone base
14                % or the bases value is smaller then the current

```

```

15         if base ~= last || peaks(base) < peaks(n)
16             % If the last peak was bigger then the current
17             % or the maximum attack length was reached
18             % => Top or sub-top of tone was found
19             if peaks(last) > peaks(n)
20                 || n - base > MAX_ATTACK_SPAN
21                 % Is the amplitude difference big enough?
22                 if peaks(last)-peaks(base) > ART_THRESHOLD
23                     % Base to local minimum between top and
24                     % base
25                     [~,imin] = min(tsignal(base:last));
26                     artifacts(base+imin) = 1;
27                 end
28                 % re-define base
29                 base = n;
30             end
31             else % re-define base
32                 base = n;
33             end
34             else % define first tone base
35                 if last > 0 && peaks(last) > peaks(n), base=n; end
36             end
37             last = n;
38         end
39     end
40
41     % Check if there is an onset but the top is outside the block
42     if last > 0 && base > 0
43         && peaks(last) - peaks(base) > ART_THRESHOLD
44         [~,imin] = min(tsignal(base:last-1));
45         artifacts(base+imin) = 1;
46     end
47 end

```

A.4. Prototyp zur Artefaktanalyse

Die Gliederung der Analyse der gefundenen Tonanfänge gliedert sich, wie in Abbildung A.4 dargestellt. Sie wird nach der Detektion aus „process.m“ aufgerufen (vgl. Listing A.9). In „analyse.m“ wird für jeden gefundenen Kandidaten für einen Tonanfang die Funktion „classifyArtifact.m“ aufgerufen. Hier werden die in Kapitel 5.1 beschriebenen Eigenschaften zu jedem gefundenen Kandidaten bestimmt. Es wird überprüft, ob es sich bei dem Kandidaten

um ein Subartefakt handelt. Es wird mit Hilfe der Funktion „getF0.m“ (vgl. Listing A.12) die Grundfrequenz ermittelt und im Anschluss mit Hilfe von „tonname.m“ (vgl. Listing A.13) der Tonname zu der Frequenz bestimmt.

Listing A.9: Aufruf der Analyse aus process.m

```
39 ...  
40 % Get artifact features  
41 [features,facts,omits] = analyse(artifacts);  
42 ...
```

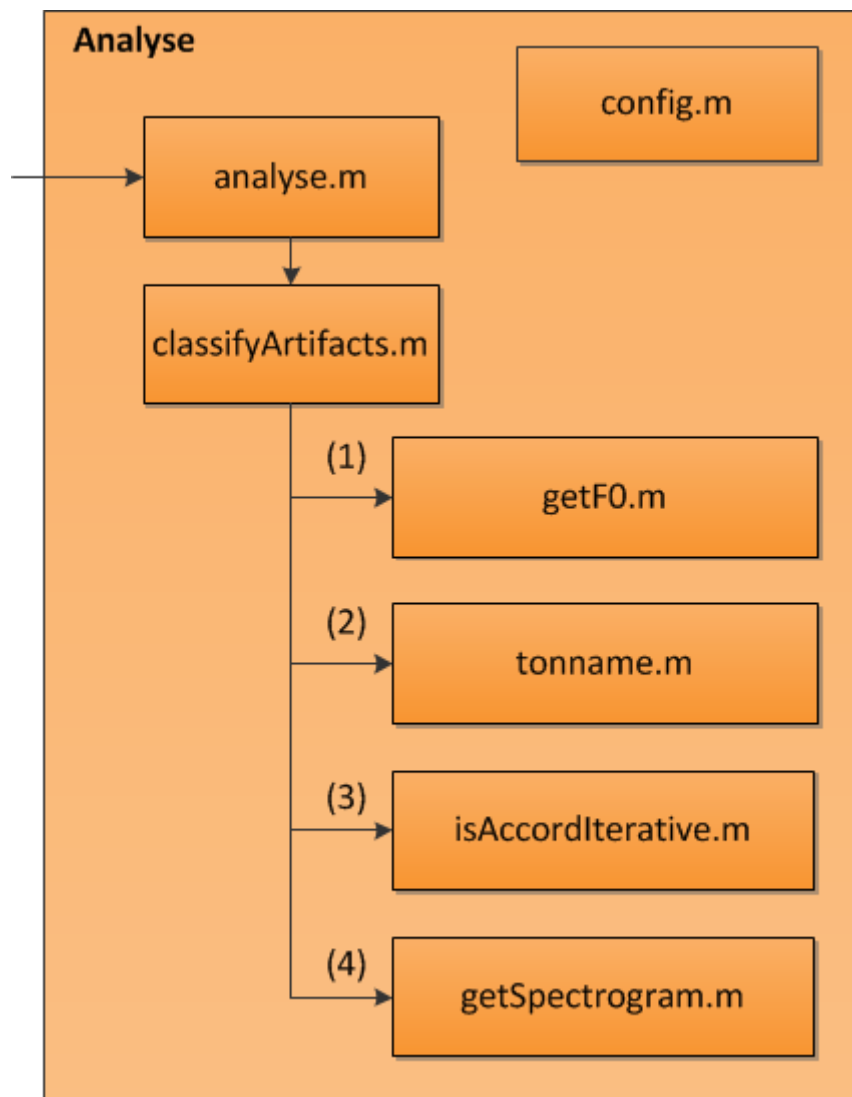


Abbildung A.4.: Systemarchitektur der Analyse

Weiterhin wird an dieser Stelle der Prototyp zur Differenzierung von Einzeltönen und Akkorden aufgerufen. Dieser ist nach der Beschreibung aus Kapitel 5.2.2 implementiert und befindet sich in „isAccordIterative.m“ (Listing A.10).

Listing A.10: isAccordIterative.m

```

1  % checks it with iteration within the frequency spectrum
2  function [out] = isAccordIterative(signal, Fs, freqs)
3      DEVIATION      = 10;
4      THRESHOLD     = 0.2;
5      HF            = 700; % 700 Hz: hoechste vorkommende Frequenz
6                      %           bei den vorliegenden Audiodateien
7      LF            = 80; % 80 Hz: niedrigste vorkommende Frequenz
8                      %           bei den vorliegenden Audiodateien
9
10     % handle parameters
11     if nargin == 2
12         freqs = [];
13     end
14
15     if length(freqs) > 2
16         out = freqs;
17         return
18     end
19
20     % windowing
21     N = length(signal);
22     W = hanning(N);
23     sw = signal.*W;
24
25     % spectrum
26     fsw = abs(fft(sw));
27     fsw = fsw(1:ceil((length(fsw) + 1) / 2));
28     if isempty(freqs)
29         fsw = fsw/max(fsw);
30     end
31     fsln = length(fsw);
32     fr   = (0:(length(fsw)-1))/fsln * Fs/2;
33     ifr  = (0:(length(fsw)-1))/Fs * 2 * fsln;
34
35     % get base frequency f0
36     if HF > length(ifr); HF = length(ifr); end
37     search = fsw(round(ifr(LF)):round(ifr(HF))) - THRESHOLD;
38
39     sidx   = find(search > 0, 1);

```

```
40     bl = sidx-DEVIATION;
41     br = sidx+DEVIATION;
42     if bl < 1; bl = 1; end
43     if br > fsln; br = fsln; end
44     rng = bl:br;
45     [~,imx] = max(fsw(rng));
46
47     if isempty(imx); out = freqs; return; end
48     if0 = imx+bl+round(ifr(LF));
49     if0t = if0;
50     freqs(length(freqs)+1) = fr(if0);
51
52     f0v = getF0(signal, Fs);
53
54     % remove base frequency from signal
55     fir = ones(size(fsw));
56     n = 1;
57     while if0t+2*DEVIATION < fsln;
58         % search window for next peak
59         bl = if0t-DEVIATION;
60         br = if0t+DEVIATION;
61         if bl < 1; bl = 1; end
62         if br > fsln; br = fsln; end
63         rng = bl:br;
64         [~,imx] = max(fsw(rng));
65         imx = imx+bl;
66
67         % erase region for filter
68         bl = imx-DEVIATION;
69         br = imx+DEVIATION;
70         if bl < 1; bl = 1; end
71         if br > fsln; br = fsln; end
72         rng = bl:br;
73         fir(rng) = 0;
74         n = n+1;
75         if0t = if0 * n;
76     end
77     fsw = fsw.*fir;
78
79     % transform back
80     for x = 1:length(signal)-fsln;
81         fsw(x+fsln) = fsw(fsln+1-x);
82     end
83     sig = ifft(fsw, 'symmetric');
```

```

84
85     % recursion with filtered signal
86     out = isAccord1(sig, Fs, freqs);
87 end

```

Zur Evaluation und zur weiteren Informationsbestimmung zu den gefundenen Artefakten wird in „getSpectrogram.m“ (Listing A.11) das Spektrogramm des Artefakts mit Hilfe der MATLAB-Funktion „spectrogram“ ermittelt. Diese führt eine Short-Time Fourier Transformation des Signals durch. Angezeigt und gespeichert werden die Spektrogramme im Ausgangsverzeichnis als zweidimensionale Darstellung. Der zeitliche Verlauf wird auf der x-Achse dargestellt, auf der y-Achse befinden sich die Frequenzen in Hz. Die Intensität der Frequenzen wird durch die Farbe der Punkte repräsentiert.

Listing A.11: getSpectrogram.m

```

1 function [out] = getSpectrogram(x, Fs)
2     r      = 80:1100;
3     NFFT = length(r);
4     Win   = hanning(NFFT);
5     [S,F,T,P] = spectrogram(x,Win,ceil((NFFT+1)/2),r,Fs);
6     figure(2)
7     surf(T,F,10*log10(P),'edgecolor','none'); view(0,90);
8
9     H = figure(2);
10    clf(H);
11    surf(T,F,10*log10(P),'edgecolor','none');
12    view(0,90); xlabel('Time (Seconds)'); ylabel('Frequency (Hz)');
13    out = H;
14 end

```

A.5. Verwendete Programme/Codeteile

Zur Bestimmung der Grundfrequenz f_0 wurde ein Skript von Kerstin Dosenbach verwendet und geringfügig angepasst. Es handelt sich dabei um die Datei „getF0.m“ (vgl. Kapitel A.4). Im Listing A.12 ist der Quellcode dieser Datei dargestellt.

Listing A.12: getF0.m

```

1 % -----
2 % Grundfrequenz nach der Cepstrum-Methode berechnen
3 % Autor: Kerstin Dosenbach
4 % Autor: Malte Kantak (Anpassung)

```

```

5  % -----
6  function f0 = getF0(inwav, fs)
7      % Indizes der Bereiche innerhalb derer nach dem Maximum gesucht
8      % werden soll berechnen
9      HF = fix(fs/700); % 700 Hz: hoechste vorkommende Frequenz bei
10         % den vorliegenden Audiodateien
11      LF = fix(fs/80); % 80 Hz: niedrigste vorkommende Frequenz bei
12         % den vorliegenden Audiodateien
13
14      % Cutting the attack & windowing
15      linwav = length(inwav);
16      wnd     = hanning(linwav);
17      inwav   = inwav .* wnd;
18
19      % Cepstrum berechnen
20      ceps = rceps(inwav);
21
22      % Innerhalb des zuvor definierten Bereichs nach dem Maximum
23      % suchen
24      [~, idx] = max(ceps(HF:LF));
25
26      % Über den ermittelten Index die Frequenz berechnen
27      f0 = fs/(HF + idx);
28 end

```

Weiterhin wurde für die Benennung der gespielten Töne das Skript „tonname.m“ übernommen. Das Skript ist in Listing A.13 abgebildet. Es wurde von Professor Dr. rer. nat. Wolfgang Fohl implementiert und zur Verfügung gestellt.

Listing A.13: tonname.m

```

1  % -----
2  % Tonnamen zu einer bestimmten Frequenz finden
3  % Autor: Wolfgang Fohl
4  % -----
5  function tname = tonname(freq)
6      C0 = 440*2^(3/12 - 5);
7      CHROMATIK = linspace(0,1,13);
8      CHROMATIK = CHROMATIK(1:12);
9
10     NAMEN=[ 'c_'
11             'c#'
12             'd-'
13             'd#'
14             'e-'

```

```
15         'f-'
16         'f#'
17         'g-'
18         'g#'
19         'a-'
20         'a#'
21         'h-'];
22
23     log2freq = log2(freq/C0);
24     oktave = fix(log2freq);
25     ton = log2freq-oktave;
26     [~, iton] = min(abs(CHROMATIK - ton));
27     tname = [NAMEN(iton,:) num2str(oktave)];
28 end
```

B. Inhalt der DVD

Auf der beigefügten DVD befinden sich die im Rahmen dieser Bachelorarbeit entwickelten Quelldateien und Verfahrensimplementationen. Darüber hinaus sind auf der DVD auch die für die Verifikation verwendeten Audiodateien sowie die Resultate der Experimente enthalten.

Die Verzeichnisstruktur und die Bedeutung der einzelnen Verzeichnisse werden in der folgenden Auflistung verdeutlicht:

Tonsegmentierung/	Stammverzeichnis der DVD
└─ Audiomaterial/	Stammverzeichnis der verwendeten Audiodateien
└─ Hense/	Audiodateien der Hense Gitarre
└─ Marin/	Audiodateien der Marin Gitarre
└─ Wichmann/	Audiodateien der Wichmann Gitarre
└─ Experimente/	Stammverzeichnis der Experimente
└─ Segmentierung/	Resultate der Segmentierung
└─ {filename}/	Pro Datei ein Ausgabeverzeichnis
└─ artifacts/	Extrahierte Artefakte und Spektrogramme
└─ Verifikation/	Resultate der Verifikation
└─ resultIn/	Die verwendeten Ergebnisdateien
└─ {filename}/	Pro Datei ein Ausgabeverzeichnis
└─ Software/	Stammverzeichnis der erzeugten Programme
└─ Segmentierung/	MATLAB Projekt des Segmentierungsverfahrens
└─ in/	Eingangsverzeichnis für Audiodateien
└─ Verifikation/	MATLAB Projekt des Verifikationstools
└─ handIn/	Dateien mit handmarkierten Onsets
└─ resultIn/	Dateien mit Ergebnissen der Segmentierung
└─ soundIn/	Eingangsverzeichnis für Audiodateien

Mit *{filename}* bezeichnete Verzeichnisse stellen Platzhalter für verschiedene Verzeichnisse dar, wobei jedes Verzeichnis den Namen der analysierten Datei trägt.

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 28. Juli 2011

Ort, Datum

Unterschrift