Automatic Segmentation and Labelling

Diplomarbeit

durchgeführt von

Peter Gutmann

vorgelegt am
Institut für Elektronische Musik und Akustik
der Universität für Musik und darstellende Kunst Graz

Begutachter: O.Univ.Prof. Mag.art. DI Dr.techn. Robert Höldrich

Betreuer: DI Dr.techn. Alois Sontacchi

Graz, im April 2008

Kurzfassung

Durch die schier unüberschaubare Menge an Musikstücken, die durch die Verfügbarkeit von kostengünstigen Speichermöglichkeiten und vor allem durch die Entwicklung des MPEG-1 Audio Layer 3 (MP3)1 Standards zur Verfügung steht, ist eine automatische Strukturierung der Archive erstrebenswert. Um den Überblick in großen Sammlungen zu behalten, ist das rasche Auffinden einer bezeichnenden Stelle des Stückes hilfreich.

Die Schwierigkeit liegt nun darin, dass im Gegensatz zu Musikdaten in Symboldarstellung (MIDI2) die Audiodaten zuerst aufbereitet werden müssen, um eine Segmentierung und anschließende Strukturierung und Benennung zu ermöglichen.

Die vorliegende Arbeit hat die Strukturanalyse von Audiodateien aus dem Genre Pop/Rock zum Ziel. Dies soll einerseits durch harmonische Analyse und damit verbundener Bestimmung der Formteile in einem Stück erreicht werden, andererseits wird die Suche nach repetitiven Elementen und die damit einhergehende Bestimmung des Refrains zur Auswertung herangezogen.

Diese beiden Ansätze werden in der Programmierumgebung Matlab als Prozeduren implementiert, um in Folge deren Praxistauglichkeit zu evaluieren. Für die harmonische Analyse wird ein neues Modell eingeführt, das zwei bekannte Vorgangsweisen miteinander verknüpft. Auf ein zum Tempo des Musikstücks synchronisiertes Chromagramm wird die sogenannte Harmonic Change Detection Function angewendet, die in der Lage ist, harmonische Veränderungen aufzuzeigen. Die so gewonnenen Segmente werden für die Strukturanalyse herangezogen. Ein gänzlich anderer Weg wird mit dem zweiten Programm beschritten. Hier erfolgt die Suche nach Ähnlichkeiten innerhalb eines Stückes. Es wird davon ausgegangen, dass der Refrain möglichst unverändert an mehreren Stellen im Musiktitel auftaucht und dementsprechend erkannt werden kann. Als Hilfsmittel dienen Distanzmatrizen und Methoden der Bildverarbeitung zur Mustererkennung. Die qualitative Auswertung erfolgt als Vergleich zur manuellen Bestimmung der Formteile bzw. des Refrains.

Es wird gezeigt, dass das vorgeschlagene harmonische Modell das Potential zur Strukturerkennung aufweist, allerdings treten Probleme auf, die es für zukünftige Versionen des Algorithmus auszumerzen gilt. Die Suche auf Basis der Ähnlichkeiten zeigt gute Ergebnisse, solange eine einfache Form (z.B. Strophe-Refrain-Refrain) eingehalten wird.

[1] *MPEG-1 Audio Layer 3 (MP3)* ist ein Dateiformat zur verlustbehafteten Audiodatenkompression nach psychoakustischen Kriterien.

[2] *Musical Instrument Digital Interface (MIDI)* ist ein Datenübertragunsprotokoll zur Übermittelung musikalischer Steuerinformationen.

Abstract

Information overload from a continuously rising number of sources is becoming increasingly unmanageable, this is also true for music collections. An overview with traditional means is impossible. The answer could be automatic structuring to be able to search specifically and to enable quick information retrieval.

The problem, as opposed to musical data in symbolic representation (MIDI), lies in the need for a pre-processing stage to find segments meaningful for the structure of the piece and finally label them.

This thesis deals with the analysis of the structure of musical data from the genre pop/rock. The first approach uses harmonic analysis to obtain the musical form of the piece, secondly a search for repeating elements is conducted to determine the position of the chorus in the song.

The implementation of modular programming-routines, as well as evaluation, takes place in Matlab software. For harmonic analysis a new model is presented, which combines two previously known approaches. Beat-synchronous calculation of the Chromagram is followed by a Harmonic Change Detection Function, to find segments suitable for the analysis of a musical structure. The second model searches for similarities within the song. The chorus is assumingly the part occurring quite similarly at different positions of a song. With the help of distance matrices and methods borrowed from image-processing and pattern-search applications, the localisation is made possible. Evaluation is performed by comparing manual annotations of musical form with automatic detections.

It will be shown that the presented harmonic model is potentially capable of finding song structures, though problems arise to be solved in future versions of the algorithm. The search for similarities functions well, as long as a basic musical form (e.g. Verse-Chorus-Chorus) is met in the song.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Motivation

Die großflächige Verfügbarkeit von Breitband Internetzugängen und die zunehmende Nutzung von online Musikportalen, sowie die einfache und kostengünstige Speicherung großer Mengen an Musiktiteln durch die Entwicklung des *MPEG-1 Audio Layer 3 (MP3)*[1] Standards einerseits und die Verbreitung portabler Abspielgeräte in Verbindung mit niedrigen Speicherpreisen führt zu unüberschaubaren Archiven, sogar für private Anwender. Die alleinige Auskunft über Titel und Interpret ist für eine gezielte Suche nicht ausreichend.

Dieses Problem wird bereits durch die im MPEG-7[2] Standard definierten Metadaten in Angriff genommen, die unter anderem eine inhaltliche Organisation zum Ziel haben. Auf unterster Ebene werden Merkmale eines Audiosignals wie Energie, Harmonizität oder Timbre beschrieben (*low-level* Deskriptoren), eine höhere Abstraktionsebene bietet inhaltlich abhängige Merkmale (*high-level* Deskriptoren). Der Standard enthält diese beschreibende Kriterien, nicht jedoch die Methoden diese beschreibenden Merkmale aus Musikdaten zu gewinnen (vgl. [Yoshioka]). Eine automatische Auswertung dieser Kriterien ist unumgänglich, nur so kann die automatische Strukturierung der Archive erreicht werden. Das Thema wird zurzeit rege erforscht, dementsprechend zahlreich ist die vorhandene Literatur und vielschichtig die verschiedenen Herangehensweisen.

[1] *MPEG-1 Audio Layer 3 (MP3)* ist ein Dateiformat zur verlustbehafteten Audiodatenkompression nach psychoakustischen Kriterien.

[2] *Multimedia Content Description Interface (MPEG-7)* ist ein Standard zur Beschreibung multimedialer Daten mit Metainformationen.

Die Bestimmung des Refrains kann beispielsweise dazu dienen, den Überblick über ein großes Musikarchiv zu behalten, indem man schnell eine bezeichnende Stelle eines Stückes findet.

Aufgabenstellung

Die vorliegende Arbeit hat die Strukturanalyse von Audiodateien aus dem Genre Pop/Rock zum Ziel. Diese Einschränkung wurde gewählt, da bei dieser Musik die das Tempo durch die klar ausgeprägten rhythmischen Bestandteile gut detektierbar ist (und es außerdem meist konstant bleibt), die Harmonien relativ einfach gehalten sind und die Liedform in groben Zügen dem Schema Intro-Strophe-Refrain-Strophe-Refrain-Solo-Refrain entspricht.

Die Schwierigkeit liegt nun dabei, dass im Gegensatz zu Musikdaten in Symboldarstellung (MIDI[1]) die Audiodaten zuerst aufbereitet werden müssen, um eine Segmentierung und anschließende Strukturierung und Benennung zu ermöglichen. Dies geschieht mit Hilfe der Berechnung von besonderen Merkmalen (*features*), bezogen auf Tempo und Rhythmus (*beat-tracker*), die Klangfarbe (*MFCC*) oder die Tonhöhe (Chromagramm). Im Anschluss folgt eine Form der Klassifizierung der so erhaltenen Eigenschaften und damit die Möglichkeit der Strukturbeschreibung.

1.1 Merkmale zur Strukturbeschreibung

Tempo und Rhythmus

Die Bestimmung des Tempos ist ein wichtiger erster Schritt für die automatisierte Erkennung von Musik im Computer, weil es sehr schwierig ist, westliche Musik ohne den Einfluss des *beats* (er entspricht im Allgemeinen der Viertelzählzeit) wahrzunehmen, weil der *beat* die grundsätzliche rhythmische Struktur darstellt (vgl. [Goto1]).

[1] *Musical Instrument Digital Interface (MIDI)* ist ein Datenübertragunsprotokoll zur Übermittelung musikalischer Steuerinformationen.

Für den Menschen ist es ohne weiteres möglich zur Musik im Takt mitzuwippen. Die Studie von McKinney (vgl. [McKinney]) zeigt, dass sowohl Musiker als auch Nicht-Musiker in der Lage sind, das musikalische Metrum[1] zu erfassen. Die Bestimmung ist nicht eindeutig, da es subjektive Unterschiede in der Auffassung gibt, die Unterschiede stehen allerdings immer im ganzzahligen Verhältnis zueinander. Die Umsetzung dieser kognitiven Fähigkeit des Menschen in ein Computermodell ist nicht trivial lösbar.

Die Wichtigkeit wird auch darin unterstrichen, dass enorme Forschung in dieses Thema investiert wurde, allen voran Scheirer (vgl. [Scheirer]), der als Basis für eine Vielzahl weiterer Arbeiten dient (vgl. [Klapuri], [Seppänen] und [Ellis1]).

Die sogenannten *beat-tracker* leisten mehr als die Tempobestimmung, sie erstellen ein Tempoprofil für das gesamte Musikstück und sind auch in der Lage Tempoänderungen zu folgen. Sie werden eingesetzt, um eine weiterführende Auswertung von Merkmalen synchron zum Tempo zu bestimmen. Diese nun tempounabhängigen Eigenschaften können als Vergleichskriterien dienen, um gleiche Formteile trotz rhythmischer Variationen zu erkennen, oder überhaupt unterschiedliche Interpretationen desselben Stückes zu erkennen (vgl. [Ellis2]).

Klangfarbe (Timbre)

Auch die empfundene Klangfarbe und somit die damit verbundenen Merkmale (z.B. MFCC) können zur Bewertung herangezogen werden. Einige Beispiele aus der Forschung werden im Folgenden genannt: Herrera beschäftigt sich mit der automatischen Erkennung von Instrumenten (vgl. [Herrera]), Tzanetakis erforscht die automatische Klassifizierung des Genres. Dabei wird versucht eine Kategorisierung von Musiktiteln in Genres anhand von spektralen oder temporalen Eigenschaften zu erreichen und die Ergebnisse mit einer Datenbank händisch klassifizierter Stücke verglichen (vgl. [Tzanetakis]). Auch Allamanche bestimmt Musiktitel mittels einer Datenbank. Er benutzt Algorithmen zur Mustererkennung von *low-level* Merkmalen

[1] Das Metrum (von griechisch métron) ist das grundlegende Betonungsmuster, auf das sich alle übrigen rhythmischen Strukturen eines Stückes beziehen.

und bestimmt sogenannte *fingerprints,* mit deren Hilfe unbekannte Musikstücke mit einer Referenz-Datenbank verglichen werden, um Ähnlichkeiten festzustellen. (vgl. [Allamanche]). Aucouturier behandelt polyphone Musikstücke und komplexe Instrumentierungen, um eine globale Beschreibung des Timbres durch MFCC (vgl. Abschnitt 2.3) zu erhalten und mit Näherungsmethoden und *Gaussian Mixture Model (GMM)* zu klassifizieren (vgl. [Aucouturier]). Foote benutzt ebenfalls MFCC als Eingangsmerkmal um mit einem Lernalgorithmus Musikstücke zu suchen (vgl. [Foote]), Cooper generiert mit MFCC als Merkmal Ähnlichkeitsmatrizen und bildet Cluster der Segmente mit statistischen Methoden (vgl. [Cooper]).

Tonhöhe

Die wohl wichtigste Eigenschaft zur Ermittlung der sogenannten *pitch-class* (entspricht der oktav-invarianten Tonhöhe) ist das Chromagramm (vgl. Abschnitt 2.5). Es liefert einen Vektor entsprechend der detektierten *pitch-classes* eines Ausschnittes eines Stückes.

Um festzustellen, ob es sich bei einem Abschnitt um eine Wiederholung eines vorangegangenen handelt, ist die Bestimmung der Ähnlichkeit der akustischen oder signaltheoretischen Eigenschaften der jeweiligen Stellen notwendig. Das Problem dabei ist, dass durch Variationen der Begleitung oder der Singstimme nie zwei Bereiche exakt gleich sind. Das Leistungsspektrum oder auch MFCC sind dazu als Beschreibung nicht optimal geeignet, da solche Variationen zu stark in die Auswertung einfließen (vgl. [Goto2]). Die Verwendung eines Chromagramms bietet sich an, denn der Vergleich der Harmonien ist auch bei Variationen (z.B. veränderte Instrumentierung) gültig.

Ein weiteres Beispiel liefert Ellis bei der Erkennung von Coverversionen (vgl. [Ellis2]). Wie zuvor schon erläutert, ist zur Bestimmung von Ähnlichkeiten nicht nur die Unabhängigkeit vom Tempo erwünscht, sondern auch von der Besetzung und dem Musikstil generell.

Bartsch und Wakefield verwenden neben dem Chromagramm Ähnlichkeitsmatrizen zur Bestimmung des Refrains (vgl. [Bartsch]), Ong und Herrera benutzen ein Modell, dass

in der ersten Phase MFCC und Subband-Energien ermittelt, in einer zweiten Phase mit Hilfe von *low-level* Eigenschaften eine Verfeinerung der Segmentierung erlaubt (vgl. [Ong]).

Klassifikation

Herrera vergleicht für die Klassifizierung monophoner Klänge einige Verfahren, beispielsweise *k-nearest neighbour, support vector machines* oder Neurale Netze (vgl. Herrera). Bello und Pickens benutzen Chroma Eigenschaften und ein *hidden-markov-model* (HMM) (vgl. [Bello2]). Chai sucht nach übereinstimmenden Mustern mit Hilfe von dynamischer Programmierung (vgl. [Chai]).

Nachdem Wiederholungen bestimmt und einzelne Segmente zusammengefasst sind, werden die Segmente der Gesamtstruktur schließlich benannt.

Diese kurze Übersicht soll den Einblick in dieses interessante Forschungsgebiet ermöglichen. Durch das Ausmaß der vorhandenen Literatur ist die Vermittelung des Gesamtbildes schwierig, um die Möglichkeiten dieses relativ neuen Forschungszweiges zu erkennen allerdings ausreichend.

Kapitelübersicht

In Kapitel 2 erfolgt die Erklärung der in den nachfolgend vorgestellten Programmen verwendeten speziellen Eigenschaften *(features)* und Verfahren. Kapitel 3 beschreibt die erstellten Programme zur Bestimmung der harmonischen Struktur sowie die Ähnlichkeitsbestimmung von Teilen eines Stückes. Es folgt die qualitative Auswertung sowie die Besprechung erkannter Probleme und Verbesserungsvorschläge. Abschließende Betrachtungen den Erfolg der verwendeten Methoden betreffend, sowie ein Ausblick auf weiterführende Verbesserungen der Modelle, finden sich in Kapitel 4.

1.2 Erstellte Programme

Die beiden hier behandelten Ansätze werden in der Programmierumgebung Matlab als Prozeduren implementiert um in Folge deren Praxistauglichkeit zu evaluieren und miteinander zu vergleichen.

Modell zur harmonischen Analyse

Der erste in dieser Arbeit behandelte Ansatz bestimmt die Formteile eines Stückes durch harmonische Analyse. Es wird ein neues Modell eingeführt, das zwei bekannte Vorgangsweisen miteinander verknüpft. Auf ein zum Tempo des Musikstücks synchronisiertes Chromagramm wird die sogenannte *Harmonic Change Detection Function* (vgl. Abschnitt 2.6) angewendet, die in der Lage ist, harmonische Veränderungen aufzuzeigen. Die so gewonnenen Segmente werden für die Strukturanalyse herangezogen.

Modell zur Ähnlichkeitsbestimmung

Ein gänzlich anderer Weg wird mit dem zweiten Programm beschritten. Hier erfolgt die Suche nach Ähnlichkeiten innerhalb eines Stückes, repetitive Elemente und die damit einhergehende Bestimmung des Refrains werden zur Auswertung herangezogen. Es wird davon ausgegangen, dass der Refrain möglichst unverändert an mehreren Stellen im Musiktitel auftaucht und dementsprechend erkannt werden kann. Als Hilfsmittel dienen Distanzmatrizen und Methoden der Bildverarbeitung zur Mustererkennung.

Die qualitative Auswertung erfolgt jeweils als Vergleich zur manuellen Bestimmung der Formteile bzw. des Refrains.

Ziel

Das Ziel der untersuchten Modelle ist es einerseits durch Segmentierung harmonisch gleicher Teile und das Auffinden gleicher Akkordsequenzen die Struktur zu bestimmen, andererseits durch die Bestimmung von Ähnlichkeiten repetitive Teile zu erkennen und unter der Bedingung, dass im Genre Pop/Rock einfache Liedformen vorherrschen, den Refrain zu bestimmen.

Es wird gezeigt, dass das vorgeschlagene harmonische Modell das Potential zur Strukturerkennung besitzt. Aufgrund der „bottom-up"- Strategie (die Bestimmung von *low-level* Merkmalen führt zur Tempoerkennung, daraus resultiert die Mittelung der nachfolgenden Eigenschaft, usw.) weist das Modell geringe Robustheit bezüglich der Fehlertoleranz auf. Für zukünftige Versionen des Algorithmus gilt es, diese Robustheit zu verbessern. Die Suche auf Basis der Ähnlichkeiten zeigt gute Ergebnisse, solange eine einfache Form (z.B. Strophe-Refrain-Refrain) eingehalten wird.

Es sei noch darauf hingewiesen, dass fast die gesamte Literatur in englischer Sprache verfasst wurde, darum wird auf eine Übersetzung der üblichen Begriffe zur Erhöhung der Verständlichkeit größtenteils verzichtet.

2 Verwendete Verfahren

Von entscheidender Bedeutung für die Analyse ist eine inhaltliche Segmentierung der Audiodaten anhand von bestimmten Eigenschaften *(features)*. Diese Eigenschaften können grob in die Kategorien zur Erkennung des Rhythmus, der Klangfarbe und der Tonhöhe eingeteilt werden.

Die nachfolgenden Abschnitte (2.1 bis 2.6) behandeln die tatsächlich für die hier verwendeten Modelle verwendeten *features*, oder für deren Verständnis wichtige Ausgangspunkte.

2.1 Onset detection

Musik wird ereignisorientiert wahrgenommen. Noten folgen einander in einem bestimmten Rhythmus, Veränderungen der Klangfarbe oder Instrumentierung bestimmen den Ausdruck. Der Beginn dieser musikalischen Ereignisses wird als *onset* bezeichnet und ist für eine Automatisierung der Erkennung verschiedener Merkmale notwendig.

Definitionen

In nachfolgender Abbildung (Abb. 2-1) ist der Idealfall einer einzelnen Note dargestellt. Als *attack* gilt das Zeitintervall, in dem die Einhüllende der Amplitude anwächst, deren Abklingphase bezeichnet man als *decay*. Der Transient entspricht hier jenem kurzen Zeitabschnitt, in dem das Musiksignal entsteht (z.B. durch die Anregung einer Saite). Der Beginn einer Note (*onset*) entspricht nun dem Anfang des Transienten.

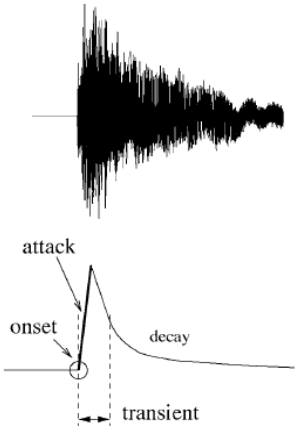

Abb. 2-1: Einzelne Note mit entsprechendem *attack, decay, transient* und *onset*[Bello1]

Bestimmung

Prinzipiell wird vom ursprünglichen Audiosignal nach einer Vorverarbeitung eine Detektor-Funktion abgeleitet, auf die wiederum eine Form der Suche nach Spitzenwerten (*peak-picking*) angewandt wird, um die *onsets* zu bestimmen.

Es folgt eine kurze Übersicht verschiedener Ansätze zur Bestimmung des *onset.*

Mit Hilfe der Vorverarbeitung (*preprocessing*) ist es möglich, Eigenschaften eines Signals zu betonen oder zu unterdrücken. Zwei Verfahren sind hier üblich: zum einen die Unterteilung des Signals in mehrere Frequenzbänder, andererseits die Trennung in transiente und stabile Zustände.

Die Detektor-Funktion kann aus folgenden Merkmalen bestimmt werden:

- Temporale Eigenschaften: im einfachsten Fall wird dabei die Einhüllende der Amplitude untersucht.

- Spektrale Eigenschaften: Energieschwankungen in Verbindung mit Transienten scheinen als breitbandiges Ereignis auf. Da die meiste Energie des Musiksignals sich im tieffrequenten Bereich befindet, kann mit einer Gewichtung des Spektrums eine Verbesserung der Erkennung der Transienten erreicht werden.

- Phase: Am Beginn einer Note entsteht meist ein Phasensprung, dieser kann als Indikator für den *onset* dienen.

- Wahrscheinlichkeitsmodelle: Beispielsweise werden Zeitpunkte, an denen plötzliche Veränderungen im Signal wahrscheinlich sind, in Modellen nachgebildet.

Abschließend folgt die Auswertung mit einer Detektion der Spitzen (*peak*-detection) dieser Detektor-Funktionen. Die Bestimmung des *onset* ist ein grundsätzlicher Bestandteil der im nachfolgenden Abschnitt behandelten Tempoerkennung.

2.2 Tempobestimmung

Meter Analysis oder *Beat-Tracking* beschreibt die Untersuchung des Metrums bzw. des Tempos eines Musikstückes. Das Erkennen des Metrums ist essentiell für das Musikverständnis (vor allem für westliche Musikstücke). Dabei handelt es sich um eine kognitive Fähigkeit des Menschen – auch ohne musikalische Ausbildung (vgl. [McKinney]).

Als erster Schritt wird nun eine Tempoanalyse durchgeführt, die sich im Pop/Rockbereich einfacher gestaltet als in anderen Genres, da das Tempo durch die klar ausgeprägten rhythmischen Bestandteile gut detektierbar ist (und es außerdem meist konstant bleibt).

Die vorliegende Arbeit hat die Strukturanalyse von Audiodateien aus dem Genre Pop/Rock zum Ziel. Diese Einschränkung wurde gewählt, da bei dieser Musik die das Tempo durch die klar ausgeprägten rhythmischen Bestandteile gut detektierbar ist (und es außerdem meist konstant bleibt), die Harmonien relativ einfach gehalten sind und die Liedform in groben Zügen dem Schema Intro-Strophe-Refrain-Strophe-Refrain-Solo-Refrain entspricht.

Als Grundschlag für den Rhythmus und als Zählzeit für das Metrum gilt normalerweise die Viertelnote. Das absolute Tempo lässt sich mit der Schlagzahl pro Minute (*beats per minute,* BPM) festlegen, ausgehend vom Mälzelschen Metronom (1816). Bei

M.M. = 60 kommen 60 Schläge pro Minute vor, also eine Viertelzählzeit pro Sekunde (vgl. [Michels]).

2.2.1 Übersicht bekannter Algorithmen

Modell von Scheirer

Wegbereiter für die rhythmische Analyse von Audiodateien ist die Arbeit von Scheirer, auf der zahlreiche aktuelle Verfahren basieren (vgl. [Scheirer]). In psychoakustischen Versuchen erforscht er die Wahrnehmung des Rhythmus und entwickelt daraus eine Filterbank zur Bestimmung der Einhüllenden in Subbändern. Der rhythmische Impuls wird mit Hilfe einer Frequenz und Phase beschrieben, wie das auch für eine periodische Schallwelle üblich ist. Diese Frequenz (bzw. die Periodendauer) entspricht dem Tempo, die Phase der Position des *beat.*

Das Audiosignal wird mit Hilfe einer Filterbank in 6 Bänder geteilt und die Einhüllende der Amplitude jedes dieser Subbänder wird bestimmt, anschließend differenziert. Das so erhaltene Signal durchläuft eine weitere Filterbank (*tuned resonators*), in der jeweils ein Resonator pro Subband so abgestimmt wird, dass die Resonanzfrequenz der Periodendauer der Einhüllenden entspricht. Die Resonanzfrequenzen der Subbänder werden summiert, die so bestimmte Frequenz entspricht dem Temposchätzwert des Signals. Abschließend wird die Phase ausgewertet und damit das Tempo bestimmt (Abb. 2-2).

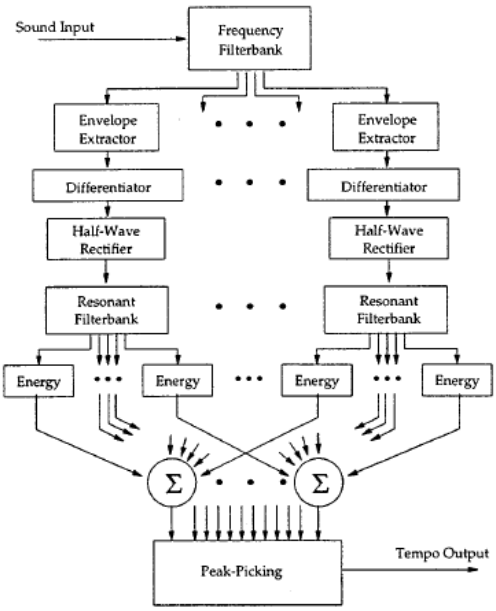

Abb. 2-2: Übersicht des Modells zur Tempobestimmung nach Scheirer [Scheirer]

Modell von Klapuri

Laut Klapuri sind zur Tempoerkennung folgende drei Schritte nötig (vgl. [Klapuri]):

1. Musikalische Betonungen müssen als Zeitfunktionen erfasst werden.
2. Die Periodendauer und Phase der zugrunde liegenden metrischen Pulse ist abzuschätzen.
3. Die rhythmische Ebene entsprechend der gewünschten Zählzeit (normalerweise der *beat*) wird ausgewählt.

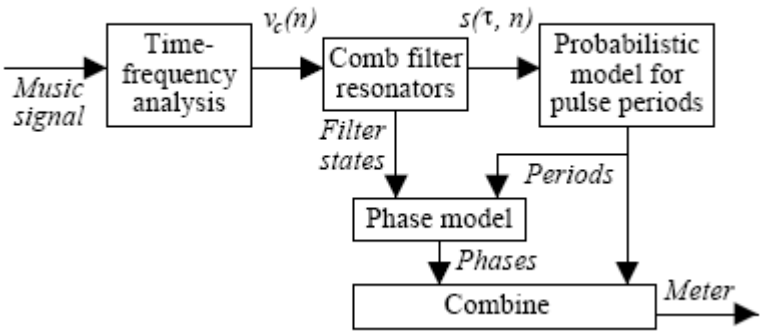

Abb. 2-3: Übersicht des Modells zur Tempobestimmung nach Klapuri [Klapuri]

Um dies zu erreichen wird zuerst eine Filterbank entsprechend der Bark-Bänder implementiert und damit sogenannte musikalische Akzente berechnet, gefolgt von einer Kammfilterbank zur Bestimmung von Periodendauer und Phase. Die so erhaltenen Kandidaten werden schließlich einer wahrscheinlichkeitstheoretischen Auswertung (HMM[1]) unterzogen (Abb. 2-3). Mittels eines anschließenden Phasenmodells, mit dessen Hilfe die genaue Phasenlage ermittelt wird, erfolgt die Bestimmung des Tempos des Musikstückes.

Modell von Seppänen

Da die Berechnung von fein abgestimmten Kammfilterresonatoren, die Rechenleistung betreffend, sehr aufwendig ist, erfolgt im Modell von Seppänen die Vorstellung einer effizienteren Methode. Die Ergebnisse sind dabei durchaus mit Klapuri vergleichbar (vgl. [Seppänen]).

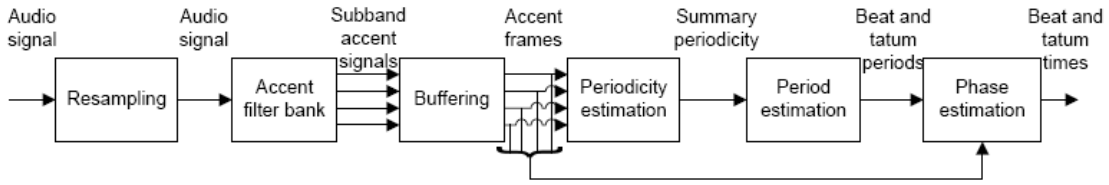

Abb. 2-4: Übersicht des Modells zur Tempobestimmung nach Seppänen [Seppänen]

Die Filterbank wird hier mit Hilfe von QMF Filtern[2] realisiert, außerdem werden zur Auswertung nur noch zwei Kammfilter benötigt, die schon zuvor abgestimmt werden. Auch dieser Algorithmus verwendet musikalisches Grundwissen in Form von Wahrscheinlichkeitsfunktionen, allerdings in einfacherer Form als bei Klapuri. Ebenfalls wird abschließend die Phase ermittelt und damit das Tempo bestimmt (Abb. 2-4).

[1] *Hidden Markov Model (HMM):* statistisches Modell, benannt nach dem russischen Mathemaiker Andrei Andrejewitsch Markov

[2] *Quadrature mirror filter (QMF)* werden in der Signalverarbeitung benutzt um ein Eingangssignal in zwei Bänder aufzuspalten. Die zwei resultierenden hoch- bzw. tiefpassgefilterten Signale werden oft um den Faktor 2 dezimiert und man erhält so eine zweikanalige Repräsentation des Originalsignals.

2.2.2 Modell von Ellis

Der tatsächlich im Programm verwendete Algorithmus zur Tempoerfassung stammt vom LabROSA[1] Projekt „*Cover Song Identification*“ von Dan Ellis (vgl. [Ellis1]).

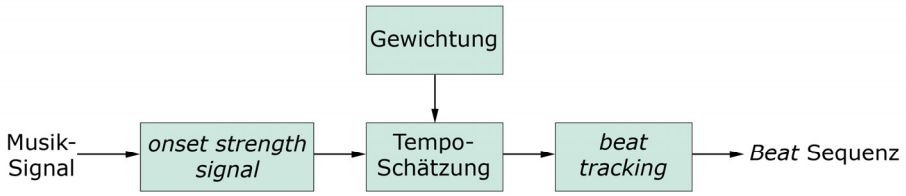

Abb. 2-5: Modell zur Tempobestimmung nach Ellis

Der hier verwendete Algorithmus besteht aus zwei Hauptteilen, einer Tempo Schätzung und nachfolgender dynamischen Programmierung des *beat-trackers* (Abb. 2-5).

Der Tempo-Schätzer basiert auf der Erkennung des *onset-strength* Parameters, der dazu dient, jene Stellen im Audiosignal zu erkennen, an denen Noten beginnen. Dafür wird das Spektrum des Signals den Mel-Frequenzbändern (vgl. Abschnitt 2.3.1) entsprechend zusammengefasst und anschließend logarithmiert, um der menschlichen Wahrnehmung besser zu entsprechen. Die so entstandenen Subbänder des Signals werden differenziert, um die Detektion der Spitzen, die den *onset* Änderungen entsprechen, zu erleichtern. Nach der Einweggleichrichtung, bei der negative Signalanteile Null gesetzt werden, erfolgt die Summierung der Subbänder und die anschließende Glättung des Signals (Abb. 2-6).

[1] *LabROSA: Laboratory for the Recognition and Organization of Speech and Audio*

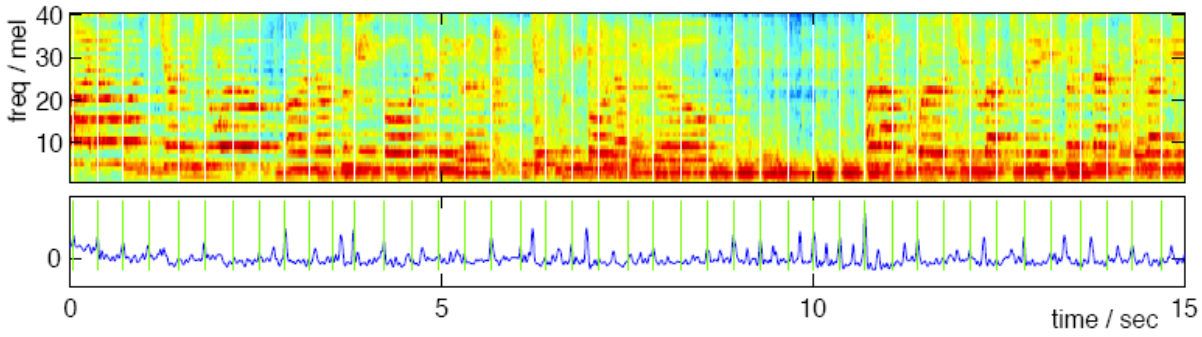

Abb. 2-6: Mel Spektrogramm (oben), geglättete *onset-strenght* Einhüllende mit detektierten *beats* (unten) eines Auszugs eines Musikstücks [Ellis1]

Die nachfolgende Autokorrelation des gesamten Signals erfolgt bis zu einer maximalen Verzögerung von 4 Sekunden. Mittels einer zuvor experimentell ermittelten Gewichtung (vgl. [McKinney]), realisiert durch eine Gauss-Fensterung (mit dem Mittelwert bei τ_0=0,5 s, das entspricht 120 BPM), wird das sogenannte primäre Tempo des Stückes bestimmt. Es entspricht dem *beat*, also der Viertelzählzeit. Das sekundäre Tempo, es entspricht der am zweithäufigsten von der Versuchsgruppe erkannten Geschwindigkeit und steht im rationalen Verhältnis zum Primärtempo (meist handelt es sich um die Achtelzählzeit), wird zusätzlich ausgegeben.

Untenstehende Abbildung verdeutlicht den Einfluss der Gewichtung auf die berechnete Autokorrelationsfunktion. Das obere Bild zeigt die Autokorrelationsfunktion des Signals bis zur maximalen Verzögerung von 4 s, unten erkennt man den Einfluss der Gewichtung. Die höchste Spitze wird als Primärtempo erkannt, die höchste Spitze mit dem Verhältnis 0,33, 0,5, 2 oder 3 dazu entspricht dem sekundären Tempo (Abb. 2-7).

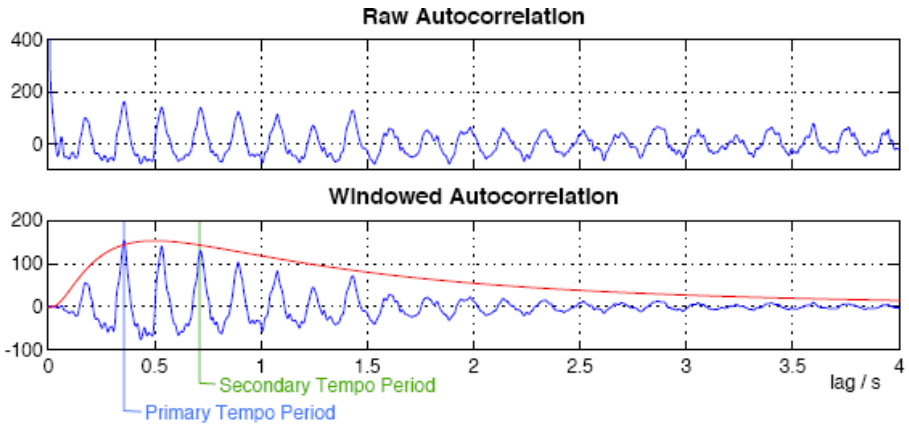

Abb. 2-7: Autokorrelationsfunktion eines Auszugs eines Musikstücks mit Gewichtungsfunktion (rot) und detektiertem primärem (grün) und sekundärem Tempo (blau) [Ellis1]

Dieses globale Tempo (Primärtempo) dient als Eingangswert der nachfolgenden dynamischen Programmierung, dem eigentlichen *beat-tracker*. Ziel ist es, dasjenige Tempo zu finden, das sowohl den Zusammenhang mit der Stärke des *onset*-Signals als auch den Abstand zwischen den *beats* optimiert. Zu diesem Zweck erfolgt die Bewertung aller möglichen *beat*-Sequenzen, um die gesamte Kostenfunktion (bestehend aus einem lokalen Wert, entsprechend der *onset* Stärke, und einer Bewertung des Übergangs) zu optimieren. Für jeden Zeitpunkt erfolgt eine Suche der vorangegangenen 0,5 bis 2 *beat*-Periodendauern. Eine Übergangswahrscheinlichkeit, ebenfalls ein Gauss-Fenster mit der Mittenfrequenz entsprechend des idealen Tempos des vorhergegangenen *beats* und der Breite als Systemparameter, wird eingeführt. Die Auswertung erfolgt vorerst für die paar letzten Takte des Stücks, für alle Kandidaten. Der Beste wird daraufhin zurückverfolgt, um die gesamte Sequenz zu erhalten. Um die Ausgewogenheit zwischen momentanem Spitzenkandidat und vorherigem Tempo zu erhalten wird eine Gewichtungskonstante für den momentanen Kandidaten eingeführt.

Der Vorteil der dynamischen Programmierung liegt darin, dass alle möglichen *beat*-Instanzen berücksichtigt werden und somit die global beste *beat*-Sequenz gefunden werden kann, sogar wenn lokale Probleme in der Tempoerkennung auftreten (Pausen oder lang ausgehaltene Noten ohne rhythmische Information).

Der Grund für die Wahl dieses Algorithmus ist eine gute Erkennungsrate bei relativ kurzer Laufzeit, wie aus dem MIREX[1] 2006 Bericht hervorgeht (vgl. [MIREX], Tab. 2-1 und Tab. 2-2).

Platzierung	**Kandidat**	**Ergebnis (*P-score*)**
1.	Dixon	0,575
2.	Davies & Plumbley	0,571
3.	Klapuri	0,564
4.	Ellis	0,552
5.	Brossier	0,453

Tab. 2-1: Ergebnis des MIREX 2006 *beat-tracking* Wettbewerbs

Platzierung	**Kandidat**	**Computer**	**Laufzeit in Sekunden**
1.	Brossier	LINUX	139
2.	Ellis	LINUX	498
3.	Dixon	FAST	639
4.	Klapuri	LINUX	1218
5.	Davies & Plumbley	FAST	1394

Tab. 2-2: Laufzeiten der *beat-tracking* Wettbewerbsprogramme (MIREX 2006)

[1] MIREX: Music Information Retrieval Evaluation eXchange

2.3 Mel Frequenz Cepstrum Koeffizienten (MFCC)

Mel-Frequenz-Cepstrum-Koeffizienten (*mel frequency cepstral coefficients*) werden häufig in der automatischen Spracherkennung verwendet, da eine kompakte Darstellung des Spektrums mit ihrer Hilfe möglich ist. Ein weiteres Anwendungsgebiet findet sich in der Analyse musikalischer Signale.

Die eigentliche Grundlage der Erzeugung von MFCC ist die lineare Modellierung bei der Sprachsynthese. Ein periodisches Anregungssignal (Simulation der Stimmbänder) wird linear gefiltert (das entspricht der Nachbildung von Mund, Zunge und Rachenraum). Die Spracherkennung interessiert sich vor allem für dieses Filter, es entspricht dem sprachlichen Inhalt, ohne Tonhöheninformation. Die Berechnung der MFCC ist eine elegante Methode, das Anregungssignal und die Impulsantwort des Filters zu trennen.

Eng mit der Entstehung der MFCC verbunden ist die Berechnung der Mel-Filterbank, diese wird zunächst kurz erklärt.

2.3.1 Tonhöhenskalen

Die Bark Skala[1] ist eine psychoakustische Skala zur Bewertung der wahrgenommenen Tonhöhe (Tonheit), eine Verdoppelung des Barkwertes bedeutet eine als doppelt so hoch empfundene Tonhöhe.

Die Tonheit in Bark berechnet sich nach Zwicker (vgl. [Zwicker]) aus der Frequenz wie folgt (vgl. auch Abb. 2-8):

$$z = 13\arctan(0,76f/1000) + 3,5\arctan(f/7500)^2$$

Bei niedrigen Frequenzen unter 500 Hz ergibt sich ein nahezu linearer Zusammenhang, bei höheren Frequenzen stellt sich ein logarithmisches Verhältnis derer Frequenzen ein, die als doppelt so hoch empfunden werden.

[1] Bark Skala: benannt nach Heinrich Barkhausen

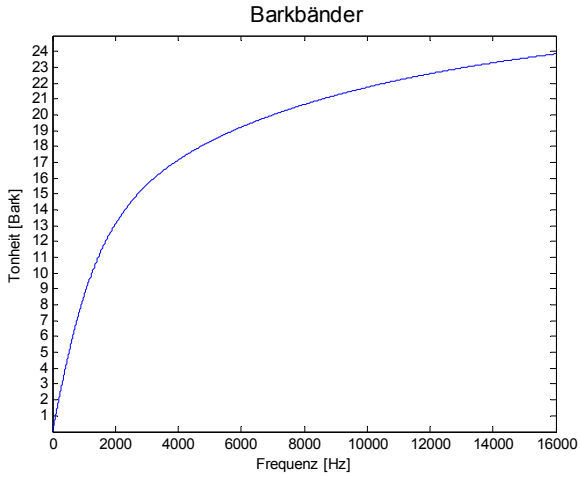

Abb. 2-8: Einteilung der hörbaren Frequenzen in Barkbänder

Eine weitere Darstellungsform ist die Tonheit z in Mel, die Umrechnung erfolgt folgendermaßen:

1 Bark = 100 Mel

Wie in untenstehender Abbildung zu sehen (Abb. 2-9) hängt die Empfindung der Tonhöhe davon ab, an welcher Stelle im Innenohr Nervenzellen erregt werden. Ein Ton einer bestimmten Frequenz führt an einem bestimmten Ort der Basilarmembran zu einem Erregungsmaximum der Haarzellen. Dargestellt sind der Ort des Erregungsmaximums und seine Beziehung zur wahrgenommenen Tonhöhe (der Tonheit z in Mel), sowie zur Frequenz f des Signals.

Die Bildung der 24 Frequenzgruppen (entspricht der nachfolgenden Bezeichnung Mel-Filterbank) modelliert diesen Zusammenhang.

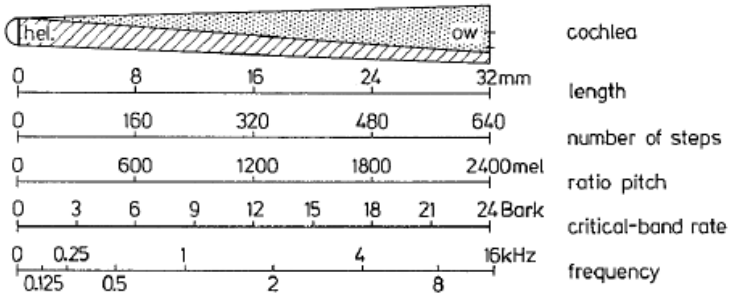

Abb. 2-9: Zusammenhang zwischen Länge der Basilarmembran, Tonheit z in Bark und Frequenz f [Zwicker]

2.3.2 Berechnung der MFCC

MFCC sind das hauptsächlich verwendete Merkmal in der Spracherkennung. Logan hat als erste die Möglichkeiten zur musikalischen Analyse untersucht, zunächst in Hinblick auf die Unterscheidung von Sprache und Musik, und zeigt, dass sie auch als Merkmale für Musik gut geeignet sind (vgl. [Logan]).

Abb. 2-10: Berechnung der MFCC nach Logan

Das Audiosignal wird in *frames* unterteilt, um stationäre Abschnitte zu erhalten (typisch 20 ms), ein Hanning Fenster beseitigt dabei Randeffekte. Mit Hilfe der Diskreten Fourier Transformation (DFT) wird das Signal in den Frequenzbereich gebracht, das Amplitudenspektrum erstellt und schließlich logarithmiert, um besser der wahrgenommenen Lautstärke zu entsprechen. Auf Phaseninformation wird dabei gänzlich verzichtet. Nach der Glättung des Spektrums erfolgt die Zusammenfassung der Frequenzbänder entsprechend der Mel-Filterbank. Als letzter Schritt wird für die Transformation in den Cepstralbereich die Diskrete Kosinustransformation (DCT) angewandt. (vgl. Abb. 2-10).

2.4 Constant-Q Transformation

Die Frequenzen der in der westlichen Musik vorkommenden Skalen liegen in geometrischem Abstand zueinander. Dies führt dazu, dass die Diskrete Fourier Transformation (DFT) Komponenten beinhaltet, die kein exaktes Abbild der musikalischen Frequenzen ermöglichen. Der Grund dafür ist, dass die mit der DFT berechneten Frequenzkomponenten mit konstanter Schrittweite der Frequenz und somit konstanter Auflösung ermittelt werden. Die *constant-Q* Transformation ist der DFT ähnlich, die Mittenfrequenzen der Filter sind allerdings geometrisch angeordnet.

$f_k = f_0 2^{\frac{k}{b}} (k = 0,...)$ b...Anzahl der Filter pro Oktave

Um die Filter einander angrenzen zu lassen, wird die Bandbreite des k-ten Filters folgendermaßen gewählt:

$$\Delta_k^{cq} = f_{k+1} - f_k = f_k \left(2^{\frac{1}{b}} - 1 \right)$$

Die Güte bleibt immer konstant, das Verhältnis von Mittenfrequenz zu Auflösung bleibt dasselbe.

$$Q = \frac{f_k}{\Delta_k^{cq}} = \frac{1}{2^{\frac{1}{b}} - 1}$$

Die erwünschte Bandbreite wird mit der Fensterlänge eingestellt:

$N_k = \frac{f_s}{\Delta_k^{cq}} = Q \frac{f_s}{f_k}$ f_s...*Samplingrate*

Die Berechnung erfolg schließlich nach folgender Formel:

$x^{cq}[k] = \frac{1}{N_k} \sum_{n<N_k} x[n] w_{N_k}[n] e^{-2\pi j n Q / N_k}$ w...Hanningfenster

Besonders nützlich ist die *constant-Q* Transformation zur musikalischen Analyse. Wählt man beispielsweise $f_0 = 110Hz$ (entspricht der Note A) und $b = 12$ (entspricht einer Halbtonauflösung) sind direkt nach der Transformation die Noten ablesbar. Des Weiteren steigt die zeitliche Auflösung mit ansteigender Frequenz, ebenso wie das bei der menschlichen Wahrnehmung der Fall ist (vgl. [Blankertz]).

In der Darstellung der *constant-Q* Transformation (Abb. 2-11) ist ein deutliches Muster der harmonischen Frequenzkomponenten zu erkennen. Das bringt Vorteile bei der Identifikation von gespielten Noten und der Erkennung von Instrumenten, und

ermöglicht eine einfache Signaltrennung mit Hilfe von Mustererkennung (vgl. [Brown]).

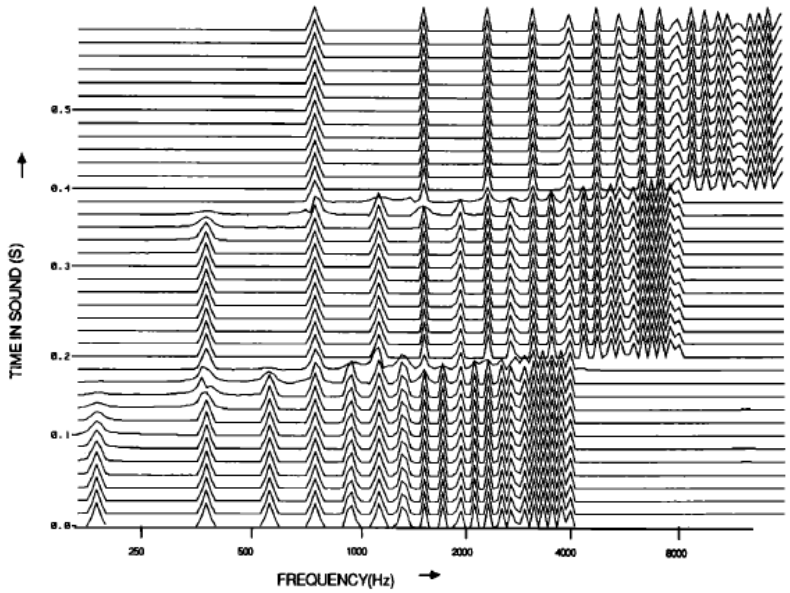

Abb. 2-11: *Constant-Q* Transformation dreier komplexer Klänge mit den Grundtönen G_3 (196 Hz), G_4 (392 Hz) und G_5 (784 Hz) mit jeweils 20 Harmonischen mit gleicher Amplitude [Brown]

2.5 Chromagramm

Der Psychologe Roger Shepard findet anhand von aufwendigen Versuchsreihen mit diversen Probanden in den "Bell Telephone Laboratories" in New Jersey heraus, dass unter bestimmten Bedingungen Tonfortschreitungen in steigender oder fallender Richtung außerhalb der Grenzen des Oktavraumes nicht mehr eindeutig einer bestimmten absoluten Lage zugeordnet werden können. Er prägt den Begriff Chroma, eine zyklische Tonhöhenbeschreibung mit der Oktave als Periodendauer. Sind zwei Töne durch ein ganzzahliges Oktavverhältnis getrennt, weisen sie denselben Chroma Wert auf. (vgl. [Shepard] und Abb. 2-12).

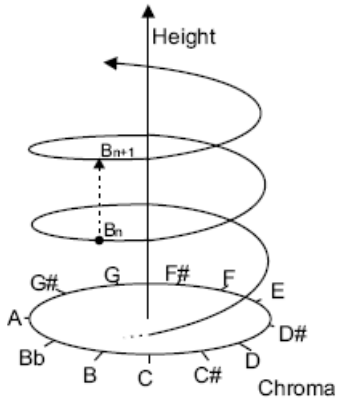

Abb. 2-12: Helix der Tonhöhen und Abbildung auf das Chromagramm [Harte1]

Wie bereits von Bartsch und Wakefield vorgeschlagen wird diese Eigenschaft der menschlichen Wahrnehmung als beschreibendes Merkmal bei der Analyse von Audiodaten eingesetzt. Das Frequenzspektrum wird durch Abbildung in ein Chroma-Spektrum überführt und bildet die Basis für das Chromagramm, welches die 12 Halbtonschritte einer Oktave pro *frame* darstellt (vgl. [Bartsch]). Die Abbildung der Frequenzen kann auf unterschiedliche Weise erfolgen, beispielsweise unter Zuhilfenahme von Filterbänken, der Fourier Transformation oder der *Constant-Q* Transformation (vgl. Abschnitt 2.4). Die so erhaltene Energieverteilung, bezogen auf die Halbtöne, ermöglicht die harmonische Analyse eines Stückes.

Als Beispiel dient hier der Klavierakkord C-Dur, für den ein Chromagramm erstellt wird (Abb. 2-13). Deutlich zu erkennen ist der Aufbau des Dreiklangs aus Grundton C, großer Terz E und der Quinte G (es handelt sich hierbei um oktav-invariante Tonhöhen, also *pitch-classes*).

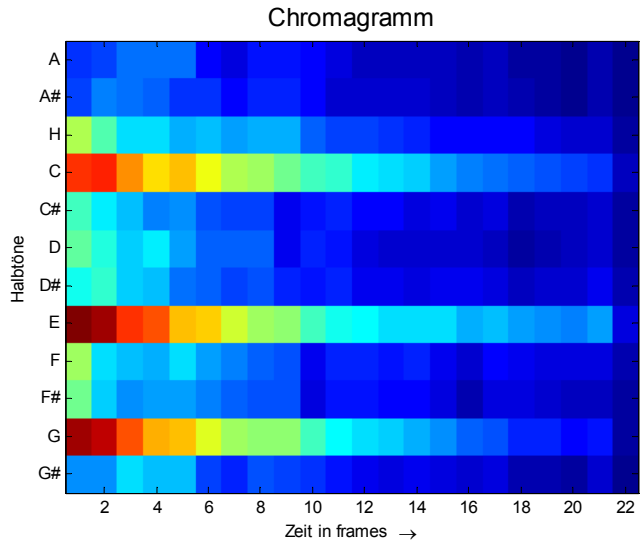

Abb. 2-13: Chromagramm eines C-Dur Klavierakkordes

2.5.1 Berechnung

Das Audiosignal wird mittels der *constant-Q* Transformation in den Spektralbereich übertragen. Anschließend erfolgt die Bestimmung des Chroma-Vektors in 36 Frequenzgruppen pro Oktave, das entspricht einer Auflösung der Tonhöhe in Achteltönen und dient dazu, einen Abgleich mit der verwendeten Grundstimmung vorzunehmen, sollte das A nicht exakt 440 Hz entsprechen. In der Orchestermusik sind durchaus abweichende Grundstimmungen üblich, ebenso sind im Popularbereich Musikstücke aus der Zeit ohne elektronische Stimmgeräte manchmal anders gestimmt (z.B. Beatles).

Um die Frequenzauflösung für eine untere Grenzfrequenz von 110 Hz in der *constant-Q* Transformation zu erreichen, ist mindestens eine Fenstergröße von 468 ms notwendig (vgl. [Brown]), was einem sehr großen Analysebereich der musikalischen Vorgänge entspricht. Um den Wechsel der Harmonien genauer bestimmen zu können, erfolgt die Wahl der Schrittweite mit 93 ms pro *frame* (vgl. [Harte1]).

Aus den so gewonnenen 36 Spektralbändern wird die Leistung berechnet und die entsprechenden Oktaven werden zusammengefasst, um einen Chroma-Vektor pro *frame* zu erhalten und somit das Chromagramm zu bestimmen.

2.6 Harmonic Change Detection Function

Diese sogenannte *Harmonic Change Detection Function* (HCDF) ist die von Harte und Sandler entwickelte Methode, um Veränderungen in der harmonischen Struktur eines Audiosignals zu erkennen (vgl. [Harte2]), eine Übersicht des Algorithmus liefert folgende Abbildung (Abb. 2-14).

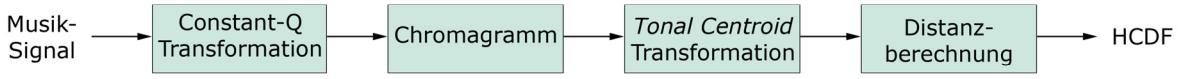

Abb. 2-14: Berechnung der HCDF nach Harte & Sandler

Die Grundlage dafür bildet das sogenannte „Tonnetz“ (Abb. 2-15), das erstmals bei Euler[1] Erwähnung findet. Es ist eine planare Repräsentation der Verhältnisse der Tonhöhen zueinander. Von links nach rechts folgen Quinten aufeinander, von links unten nach rechts oben sind es große Terzen und von links oben nach rechts unten kleine Terzen in Folge. Bei der Annahme reiner Stimmung ist das Tonnetz unendlich ausgedehnt, aufgrund der Tatsache, dass eigentlich in unterschiedlichen Lagen auch minimal unterschiedliche Tonhöhen vorherrschen (die reine Stimmung folgt den natürlichen Intervallproportionen, die temperierte Stimmung teilt die Oktave mathematisch in 12 Abstände von je $\sqrt[12]{2}^{-1}$, vgl. [Michels]). Wird nun angenommen, dass derselbe Notenname in einer Zeile dem einer anderen entspricht ($F\#_1 = F\#_2$), entsteht aus der Ebene eine Röhre, mit den Quinten als Helix auf ihrer Oberfläche. In der temperierten Stimmung ergeben sich durch die enharmonische Gleichheit der Tonhöhen nur 12 verschiedene Töne innerhalb einer Oktave (vgl. Chromagramm, Abschnitt 2.5). Diese Tatsache führt zur Verbindung der Enden der zuvor entstandenen Röhre, sie wird zum Hypertorus, um den sich der Quintenzirkel dreimal schlingt (Abb. 2-16).

[1] Leonhard Euler (1707-1783), ein bedeutender Mathematiker

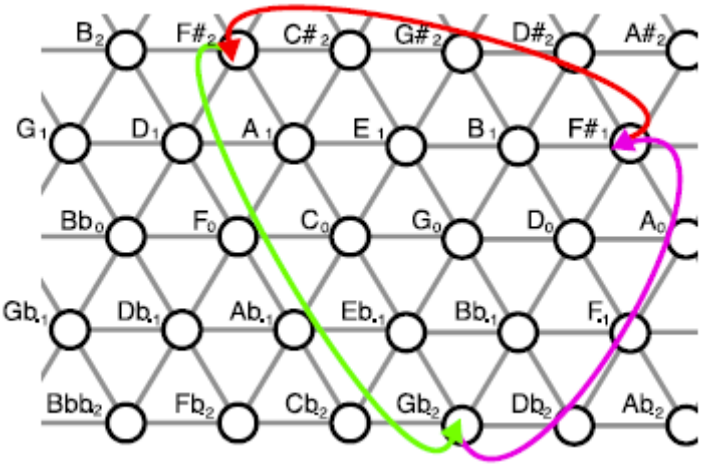

Abb. 2-15: Das „Tonnetz“, eine Repräsentation der Verhältnisse von Tonhöhen zueinander [Harte2]

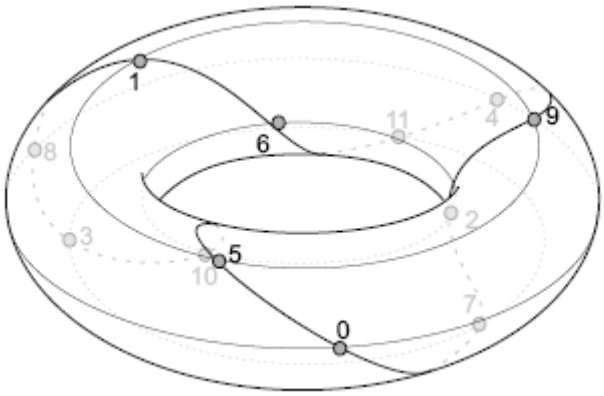

Abb. 2-16: Die Projektion des „Tonnetzes“ auf die Oberfläche eines Hypertorus [Harte2]

Jedem Akkord entspricht ein tonales Zentrum (*tonal centroid*), das mit 3 Koordinatenpaaren festgelegt ist. Die Darstellung der tonalen Zentren erfolgt als Projektion auf den Kreis der Quinten, sowie der kleinen und großen Terzen (Abb. 2-17), als Beispiel ist der A-Dur Akkord in rot eingezeichnet. Dieser aus der Abbildung der Chroma-Vektoren erhaltene Vektor des tonalen Zentrums wird anschließend geglättet und die HCDF ergibt sich schließlich aus dessen Änderungsrate, die aus der euklidischen Entfernung zwischen dem nachfolgenden und vorangegangenen tonalen Zentrum bestimmt wird.

Auf diese Weise können Harmonieänderungen bestimmt und deren Stärke erfasst werden.

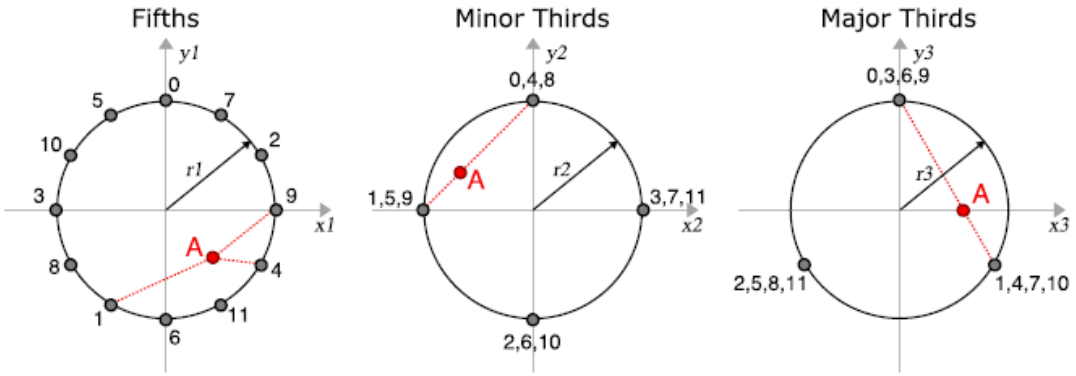

Abb. 2-17: Visualisierung des Tonraums als Kreis der Quinten (links), Kreis der kleinen Terzen (Mitte) und Kreis der großen Terzen (rechts), das tonale Zentrum für den Akkord A-Dur ist eingezeichnet [Harte2]

Transformationsmatrix Φ:

$$\Phi_l = \begin{bmatrix} \Phi(0,l) \\ \Phi(1,l) \\ \Phi(2,l) \\ \Phi(3,l) \\ \Phi(4,l) \\ \Phi(5,l) \end{bmatrix} = \begin{bmatrix} r_1 \sin l\frac{7\pi}{6} \\ r_1 \cos l\frac{7\pi}{6} \\ r_2 \sin l\frac{3\pi}{2} \\ r_2 \cos l\frac{3\pi}{2} \\ r_3 \sin l\frac{2\pi}{3} \\ r_3 \cos l\frac{2\pi}{3} \end{bmatrix} \qquad 0 \le l \le 11$$

Tonales Zentrum:

$$\zeta_n(d) = \frac{1}{\|c_n\|_1} \sum_{l=0}^{11} \Phi(d,l) c_n(l) \qquad \begin{matrix} 0 \le d \le 5 \\ 0 \le l \le 11 \end{matrix}$$

HCDF:

$$\xi_n = \sqrt{\sum_{d=0}^{5} \left[\zeta_{n+1}(d) - \zeta_{n-1}(d)\right]^2}$$

3 Strukturanalyse

Um Audiodateien aus dem Genre Pop/Rock zu strukturieren, werden einerseits durch harmonische Analyse die Formteile in einem Stück bestimmt, andererseits wird die Suche nach repetitiven Elementen und die damit einhergehende Bestimmung des Refrains zur Auswertung herangezogen.

Der erste in dieser Arbeit behandelte Ansatz (vgl. Abschnitt 3.1) bestimmt die Formteile eines Stückes durch harmonische Analyse. Es wird ein neues Modell eingeführt, das zwei bekannte Vorgangsweisen miteinander verknüpft. Auf ein zum Tempo des Musikstücks synchronisiertes Chromagramm wird die sogenannte *Harmonic Change Detection Function* (vgl. Abschnitt 2.6) angewendet, die in der Lage ist, harmonische Veränderungen aufzuzeigen. Die so gewonnenen Segmente werden für die Strukturanalyse herangezogen.

Ein gänzlich anderer Weg wird mit dem zweiten Programm beschritten (vgl. Abschnitt 3.2). Hier erfolgt die Suche nach Ähnlichkeiten innerhalb eines Stückes, repetitive Elemente und die damit einhergehende Bestimmung des Refrains werden zur Auswertung herangezogen. Es wird davon ausgegangen, dass der Refrain möglichst unverändert an mehreren Stellen im Musiktitel auftaucht und dementsprechend erkannt werden kann. Als Hilfsmittel dienen Distanzmatrizen und Methoden der Bildverarbeitung zur Mustererkennung.

Die qualitative Auswertung erfolgt jeweils als Vergleich zur manuellen Bestimmung der Formteile bzw. des Refrains.

Das Ziel der untersuchten Modelle ist es einerseits durch Segmentierung harmonisch gleicher Teile und das Auffinden gleicher Akkordsequenzen die Struktur zu bestimmen, andererseits durch die Bestimmung von Ähnlichkeiten repetitive Teile zu erkennen und

unter der Bedingung, dass im Genre Pop/Rock einfache Liedformen vorherrschen, den Refrain zu bestimmen.

Die Herausforderungen dabei werden von Goto folgendermaßen beschrieben (vgl. [Goto]):

Der Refrain ist normalerweise derjenige Abschnitt eines Musikstücks mit den häufigsten Wiederholungen. Das heißt um den Refrain zu finden müssen sich wiederholende Stellen gefunden werden und jene mit den meisten Wiederholungen entsprechen dem Refrain.

Für die automatische Auswertung stellen sich dabei folgende Probleme:

Um festzustellen, ob es sich bei einem Abschnitt um eine Wiederholung eines vorangegangenen handelt, ist die Bestimmung der Ähnlichkeit der entsprechenden akustischen oder signaltheoretischen Eigenschaften der jeweiligen Stellen notwendig. Das Problem dabei ist, dass durch Variationen der Begleitung oder der Singstimme nie zwei Bereiche exakt gleich sind. Das Leistungsspektrum oder MFCC sind dazu als Beschreibung nicht optimal geeignet, da solche Variationen zu stark in die Auswertung einfließen. Als Lösung bietet sich die Verwendung des Chroma Merkmales an, im Speziellen die *beat*-synchrone Chroma Bestimmung, um neben der Variation der Instrumentierung oder der Gesangsstimme auch eine Veränderung des Tempos nicht mehr für die Erkennung der Ähnlichkeit zum Tragen kommt.

Der Schwellwert der notwendigen Ähnlichkeit zur Erkennung des Refrains ist von Stück zu Stück sehr unterschiedlich, abhängig davon, wie ähnlich sich einzelne Phrasen und Teile sind. Er sollte automatisch angepasst werden, eine adaptive Formulierung des Kriteriums zur Schwellwertbestimmung erscheint sinnvoll.

Die Bestimmung des Anfangs- und Endpunktes des Refrains ist problematisch. Bei einem Stück mit der Struktur A B C B C C wird BC als Wiederholung erkannt, der Refrain ist aber tatsächlich nur der Teil C. Abhilfe schafft die Suche der Wiederholungen und eine anschließende Gruppierung mit Berücksichtigung der Relationen im gesamten Stück.

3.1 Modell zur harmonischen Analyse

3.1.1 Übersicht

Das hier untersuchte Modell (Abb. 3-1) basiert auf *beat*-synchroner Chromaberechnung. Der *beat* ist in diesem Fall allerdings nicht die Viertelzählzeit, sondern die Achtelzählzeit, um die Ergebnisse der Chromamittelung zu verbessern (vgl. [Ellis]). Auf diese Chromawerte wird im Anschluss eine *Harmonic Change Detection Function* (vgl. Abschnitt 2.6) angewandt. Eine nachfolgende zweite Mittelung innerhalb gleich erkannter Harmonien soll die Erkennungsrate der berechneten Akkorde verbessern. Die nun erhaltenen Akkordfolgen werden nach musikalischer Gewichtung mittels eines Suchalgorithmus ausgewertet. Diese Anordnung ermöglicht die Erkennung gleicher Formteile eines Stückes.

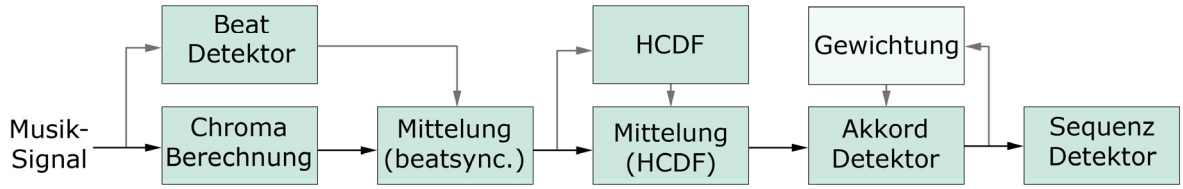

Abb. 3-1: Modell zur harmonischen Analyse

3.1.2 Detaillierte Beschreibung

Beat Detektor

Der *beat* Detektor (also der *beat-tracking* Algorithmus) stützt sich auf das Modell von Ellis (vgl. Abschnitt 2.2.2).

Das Tempo des Musiksignals wird durch die Auswertung rhythmischer Akzente erfasst, um mit dem *beat-tracker* die *beats* zu bestimmen und somit eine weiterführende beatsynchrone Analyse zu ermöglichen. Dies hat den Vorteil, dass in weiterer Folge *features* bezogen auf den *beat* verarbeitet werden, da aus musikalischer Sicht zwischen zwei *beats* nicht mit harmonischen Änderungen gerechnet werden muss.

Die nachfolgende Grafik zeigt einen Ausschnitt (die ersten 11 Sekunden) aus dem Stück „Sonnet“ von „The Verve“ mit den zugehörigen detektierten *beats* (Abb. 2-5).

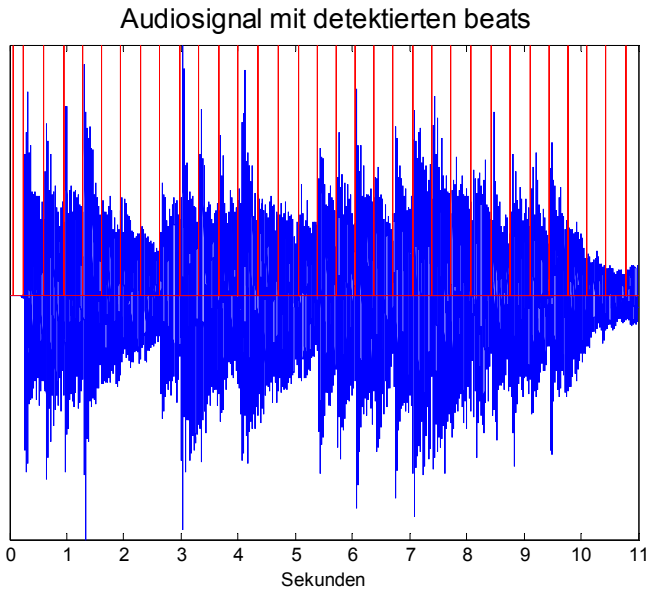

Abb. 3-2: Audiosignal (Sonnet, 11s) mit detektierten *beats*

Die nachfolgende Tabelle zeigt den Vergleich der detektierten Tempi mit der manuellen Auswertung, außerdem das im Algorithmus als *beat* verwendete Tempo, das der Achtelzählzeit entspricht. Die manuelle BPM-Bestimmung erfolgt durch mittippen des *beats* von ca. einer Minute des Musikstückes und anschließender Mittelung. Die Ergebnisse der automatischen Berechnung sind als sehr gut einzustufen, die Abweichungen zum händisch ermittelten Tempo minimal (Tab. 3-1).

Titel	***beat* (manuell)**	***beat* (detektiert)**	**verwendetes Tempo**
Let it Be (The Beatles)	72,3 BPM	72,8 BPM	146 BPM
Like a Virgin (Madonna)	119,3 BPM	119,0 BPM	238 BPM
Sonnet (The Verve)	87,4 BPM	88,2 BPM	176 BPM
Real World (Matchbox 20)	117,9 BPM	119,0 BPM	238 BPM

Tab. 3-1: Vergleich der detektierten Tempi mit der manuellen Auswertung der als Beispiele verwendeten Stücke

Chroma Berechnung

Parallel dazu wird das Chromagramm des Musikstücks berechnet, zu dessen Erstellung die Vorgehensweise aus Abschnitt 2.5.1 Verwendung findet.

Der Spektralanalyse mit *Constant-Q* folgt die Berechnung des Chromagramms. Dabei wird ein Zeitfenster von 468 ms mit einer Schrittweite von 93 ms gewählt. Als Signal dient wiederum derselbe Auszug aus dem Stück „Sonnet“ (Abb. 3-3).

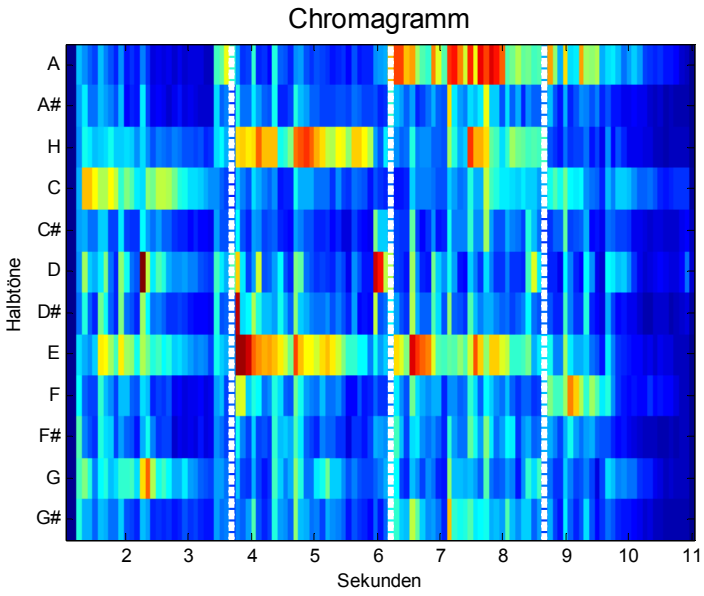

Abb. 3-3: Chromagramm (Sonnet, 11s), 93ms pro *frame*, strichliert: Zeitpunkt der Akkordwechsel

Chroma Mittelung (beatsynchron)

Dieser erste Chroma Mittelwert wird bestimmt, indem über die berechneten Chromawerte innerhalb eines *beats* gemittelt wird. Alle Chromawerte innerhalb der Zeitgrenze werden zusammengefasst und als neuer Chromawert gespeichert (Abb. 3-4).

Mit dieser Methode ist es möglich, transiente Störungen im Übergangsbereich zu unterdrücken, man erhält stationäre harmonische Zustände innerhalb der zuvor ermittelten Zeitabschnitte.

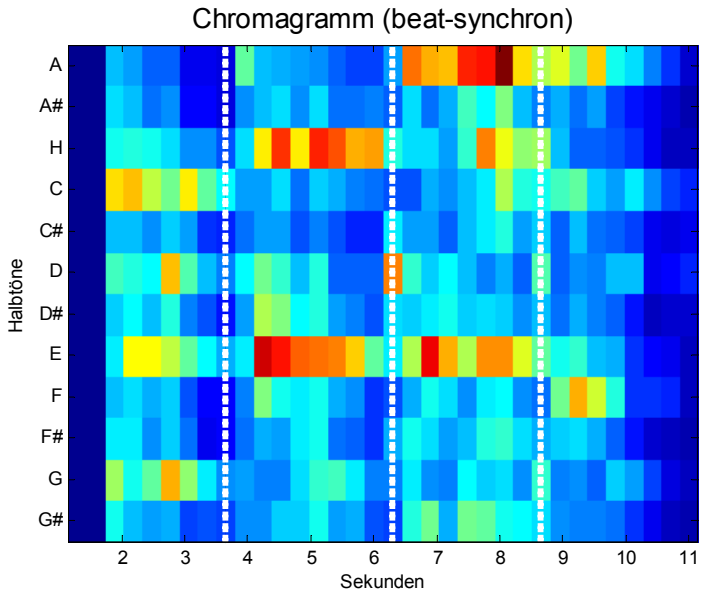

Abb. 3-4: Chromagramm (Sonnet, 11s), *frames beat*-synchron, 324 ms pro *frame,* strichliert: Zeitpunkt der Akkordwechsel

Harmonic Change Detection Function (HCDF)

Diese Funktion wird wie in Abschnitt 2.6 beschrieben implementiert, allerdings hier auf die zuvor gefundenen, *beat*-synchronen Chroma-Mittelwerte angewandt, nicht auf die einzelnen Chroma-Vektoren des Signals selbst. Die detektierten Spitzen (*peaks*) sind die neuen Grenzen für die folgende zweite Mittelung (Abb. 3-5).

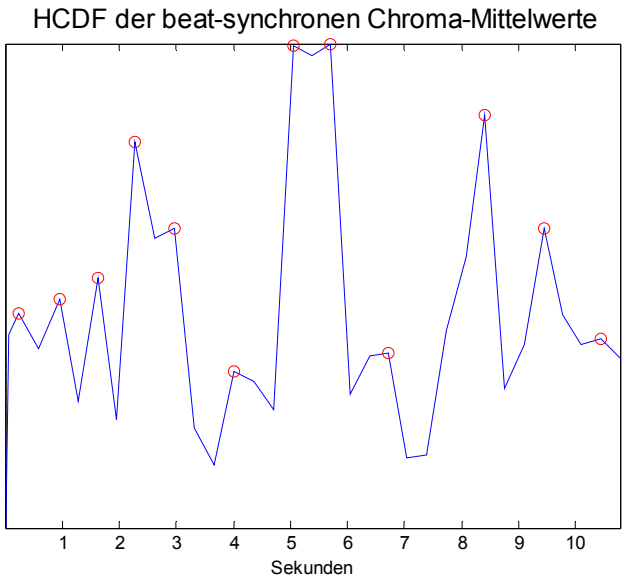

Abb. 3-5: HCDF (Sonnet, 11s)

Chroma Mittelung (HCDF)

Die HCDF zeigt Veränderungen der harmonischen Struktur auf, Akkordwechsel werden damit aufgespürt. Während harmonisch weitgehend gleichbleibender Phasen wird nun ein weiteres Mal gemittelt, um die Erkennungsrate der Akkorde durch verbesserte Chroma Auswertung zu erhöhen.

Das so erhaltene Chromagramm (Abb. 3-6) zeigt nun die erwarteten stationären Zustände, vor allem der Grundton ist deutlich erkennbar, was daran liegt, dass im Bassbereich die höchste Energie vorhanden ist.

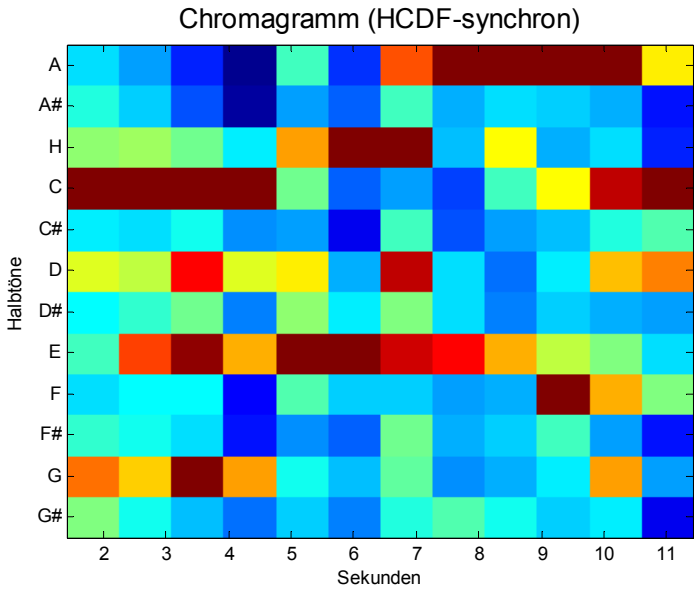

Abb. 3-6: Chromagramm (Sonnet, 11s), *frames* entsprechend der HCDF gemittelt

Akkord Detektor

In der momentanen Implementierung erfolgt nur die Unterscheidung nach Tongeschlecht (Dur- oder Moll-Akkord). Für die Bestimmung der Akkorde wird zunächst je eine Akkordmaske für Dur und Moll erstellt (Grundton, große Terz und Quinte für den Durakkord, Grundton, kleine Terz und Quinte für den Mollakkord).

$$Maske\_Dur = [1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0]$$

$$Maske\_moll = [1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]$$

Aus diesen Masken werden durch zirkuläre Rotation der Elemente, entsprechend des Akkordgrundtons, die jeweiligen Zeilen der Akkordmatrix bestimmt.

$$\begin{bmatrix} 1&0&0&0&1&0&0&1&0&0&0&0 \\ 0&1&0&0&0&1&0&0&1&0&0&0 \\ 0&0&1&0&0&0&1&0&0&1&0&0 \\ 0&0&0&1&0&0&0&1&0&0&1&0 \\ 0&0&0&0&1&0&0&0&1&0&0&1 \\ 1&0&0&0&0&1&0&0&0&1&0&0 \\ 0&1&0&0&0&0&1&0&0&0&1&0 \\ 0&0&1&0&0&0&0&1&0&0&0&1 \\ 1&0&0&1&0&0&0&0&1&0&0&0 \\ 0&1&0&0&1&0&0&0&0&1&0&0 \\ 0&0&1&0&0&1&0&0&0&0&1&0 \\ 0&0&0&1&0&0&1&0&0&0&0&1 \end{bmatrix} \bullet \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{12} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{12} \end{bmatrix}$$

c...Chromavektor
b...Bewertungsvektor

Das Skalarprodukt dieser Matrix (die hier dargestellte Matrix gilt für Durakkorde) mit dem momentanen Chroma-Vektor bildet den Vektor zur Akkordbewertung (b). Das Maximum dieses Vektors wird detektiert und somit der Akkord bestimmt (vgl. [Harte1]).

Als Beispiel dient wiederum ein Ausschnitt aus „Sonnet“, diesmal bis zum ersten Refrain. Zum Vergleich wurde der Abschnitt auch händisch transkribiert (Abb. 3-7). Bis auf den Bereich am Ende, wo fälschlicherweise d und g erkannt wird, stimmen die Ergebnisse überein.

Die Einleitung besteht aus den ersten beiden Akkordfolgen (C e a F / C e a F), wobei zuerst nur eine Akustikgitarre spielt und erst beim zweiten Akkorddurchlauf die Band inklusive Schlagzeug einsetzt. Die dritte und vierte Folge (C e a F / C e a F) beinhaltet die Strophe, inklusive Gesang. Danach folgt der Refrain (F G e F / F G e F e F), der am Schluss noch eine kleine Wiederholung aufzeigt.

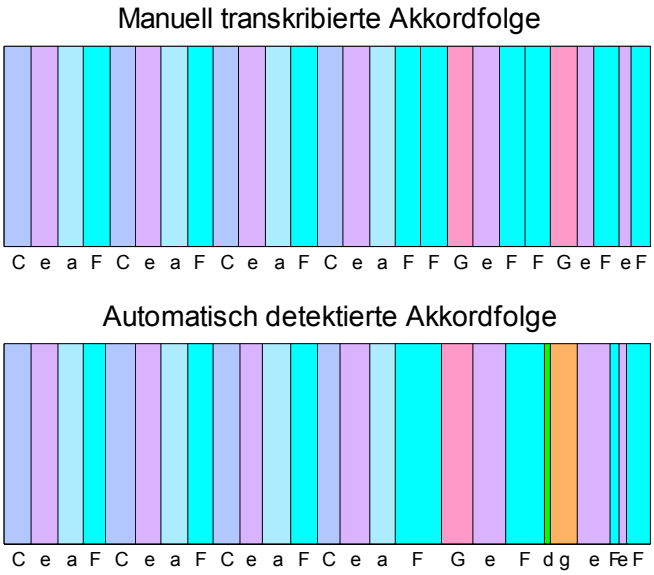

Abb. 3-7: Akkordfolge (Sonnet, 68s), manuell transkribiert (oben) und die automatische Erkennung (unten)

Die Akkordfortschreitung kann nach musikalischen Gesichtspunkten bewertet werden. Die benachbarten Töne im Quintenzirkel sind wahrscheinlichere Nachfolger, als weit entfernte (Abb. 3-8). Das bedeutet, dass beispielsweise nach dem momentanen Akkord CDur der nachfolgende Akkord sehr wahrscheinlich FDur, GDur oder aMoll bzw. e-Moll sein kann, gänzlich unwahrscheinlich aber FisDur.

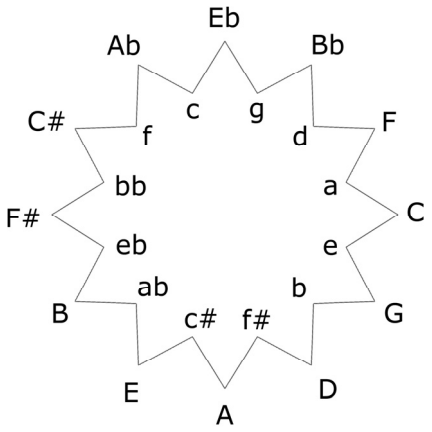

Abb. 3-8: Verschachtelter doppelter Quintenzirkel, mit den Molldreiklängen versetzt zu den Durdreiklängen [Bello2]

Die so gefundene Akkordfolge wird nun nach sich wiederholenden gleichen Abfolgen durchsucht.

Sequenz Detektor

Als Suchalgorithmus dient in diesem Programm „Sequitur“ von Nevill-Manning und Witten (vgl. [Nevill]).

Dieser Algorithmus wurde für die Sprachkodierung entwickelt. Es wird nach sich wiederholenden Buchstabenpaaren in einem Text gesucht, die dann durch ein neues Symbol (hier Großbuchstaben) ersetzt werden. Dieser wird zur späteren Dekodierung im Wörterbuch gespeichert (vgl. Abb. 3-9).

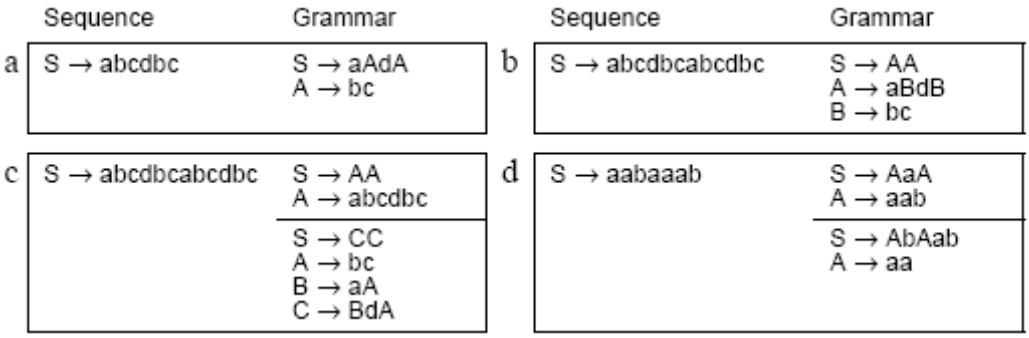

Abb. 3-9: Beispiel „Sequitur“ [Nevill]

Der Vorteil dabei ist, dass die längsten möglichen gleichen Teile erkannt werden. Leider ist dieser Algorithmus in seiner ursprünglichen Form für die hier gewollte Erkennung der Akkordstrukturen nicht optimal, da es keinen Zeitbezug gibt. Für musikalische Anwendungen wäre gerade das aber wichtig, da Formteile normalerweise einen Zusammenhang mit der Anzahl der Takte und somit dem *beat* aufweisen.

Ein weiterer Nachteil liegt darin, dass die Segmentierung niemals eindeutig ist. Die Folge ABAB kann gleichbedeutend als CC dargestellt werden, der Bezug zu einem weiteren erkannten Segment AB geht damit verloren.

Im Prinzip kann allerdings trotzdem gezeigt werden, dass eine Erkennung der Formteile auf diese Art realisiert werden kann. In Abb. 3-10 wird die sich wiederholende Akkordsequenz der Strophe korrekt als wiederkehrende Sequenz erkannt.

Die Bezeichnung der detektierten Formteile mit römischen Zahlen ist als Folge des verwendeten Suchalgorithmus entstanden und hat keine musikalische Relevanz.

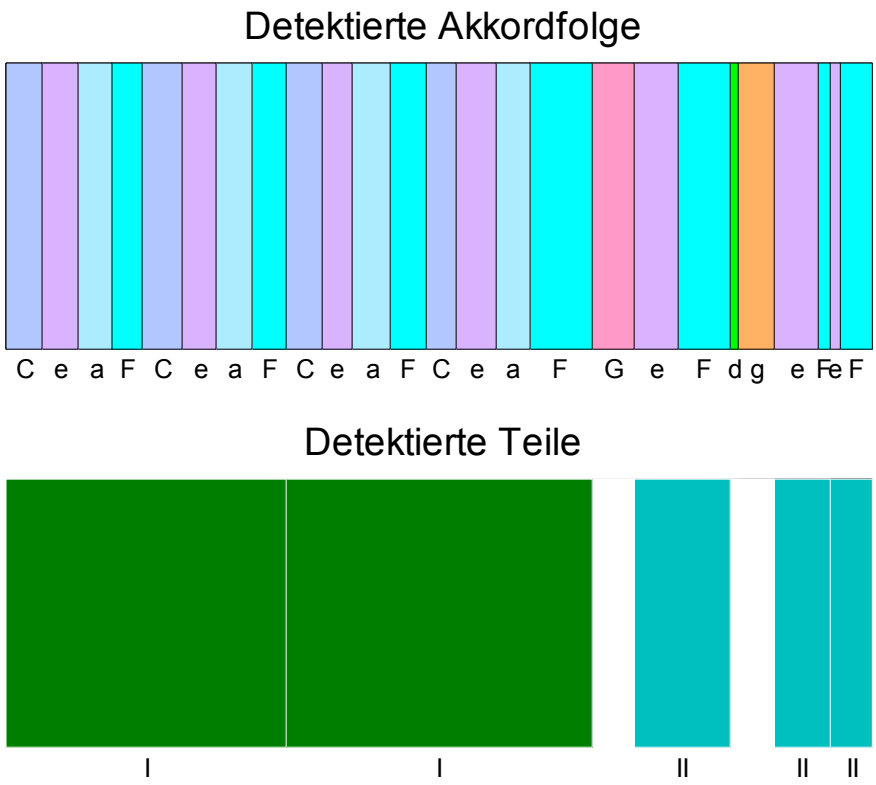

Abb. 3-10: Sequenzerkennung (Sonnet, 68s), detektierte Akkordfolge (oben) und detektierte Formteile (unten)

Der folgende Abschnitt beinhaltet die Besprechung der Ergebnisse und der auftretenden Probleme.

3.1.3 Ergebnisse, Probleme und Verbesserungsvorschläge

Das Programm liefert zur Veranschaulichung als Endergebnis die Wellenformdarstellung der Audiodatei mit eingezeichneten erkannten *beats*, die *Harmonic Change Detection Function* des gesamten Stücks mit den relevanten *peaks,* die gesamte erkannte Akkordfolge, sowie die daraus folgenden detektierten Teile. In nachfolgender Abbildung wird aus Gründen der Übersichtlichkeit auf die Darstellung der *beats*, sowie der *peaks* der HCDF verzichtet, Es handelt sich hierbei um die ersten 68 Sekunden des Stückes „Sonnet“ (zur Übersichtlichkeit im Bild), der Refrain kann noch nicht erkannt werden, da er erst einmal vorkommt, die Einleitung und Strophe (bestehend aus derselben Akkordfolge) werden korrekt erkannt (Abb. 3-11).

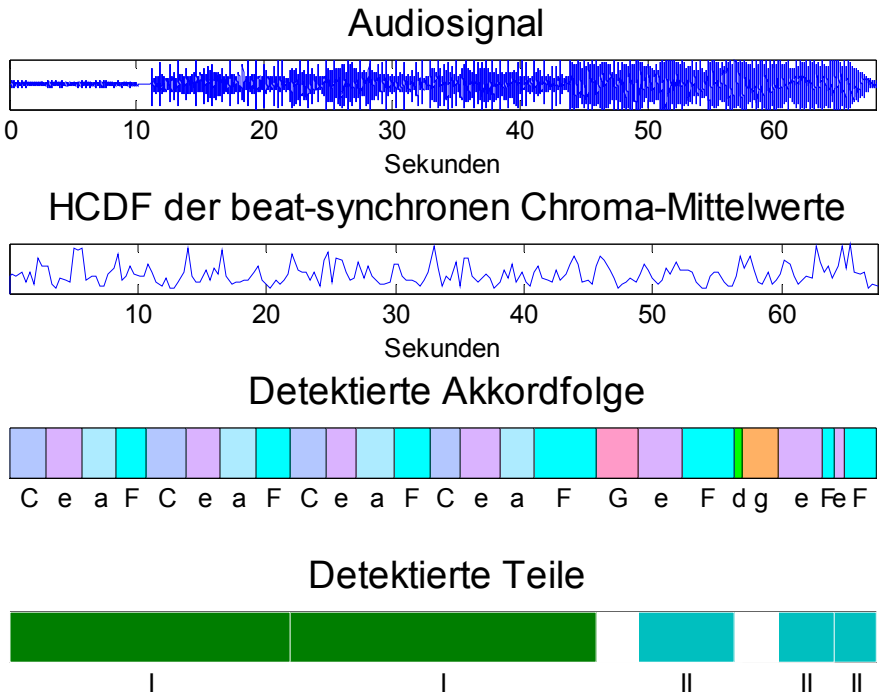

Abb. 3-11: Programmausgabe: Darstellung des Audiosignals, der HCDF, der detektierten Akkordfolge, sowie der detektierten Teile (Sonnet, 68s)

Verbesserung durch die Mittelungen

Die folgende Darstellung (Abb. 3-12) zeigt den Vergleich der Mittelung nach der HCDF (mit zusammengefassten Akkordwiederholungen) mit der *beat*-synchronen Mittelung und den daraus direkt bestimmten Akkorden, mit und ohne den Einfluss der Gewichtung. Es zeigt sich, dass wie erwartet eine Verbesserung der Erkennungsrate gegenüber der *beat*-synchronen Auswertung erzielt werden kann. Akkorde, die für einen kurzen Zeitbereich falsch erkannt werden (der braune Streifen wird als b-moll ausgewertet) fallen durch die Mittelung heraus. Ebenso zeigt die Gewichtung Wirkung, ohne sie (im Bild ganz unten) steigt die Falscherkennung beträchtlich an, einige im Stück nicht vorhandene Akkorde (violett=E-Dur, rot=B-Dur, grün=f-moll, petrol=A-Dur, gelb=Bb-Dur) werden ohne die Gewichtung detektiert.

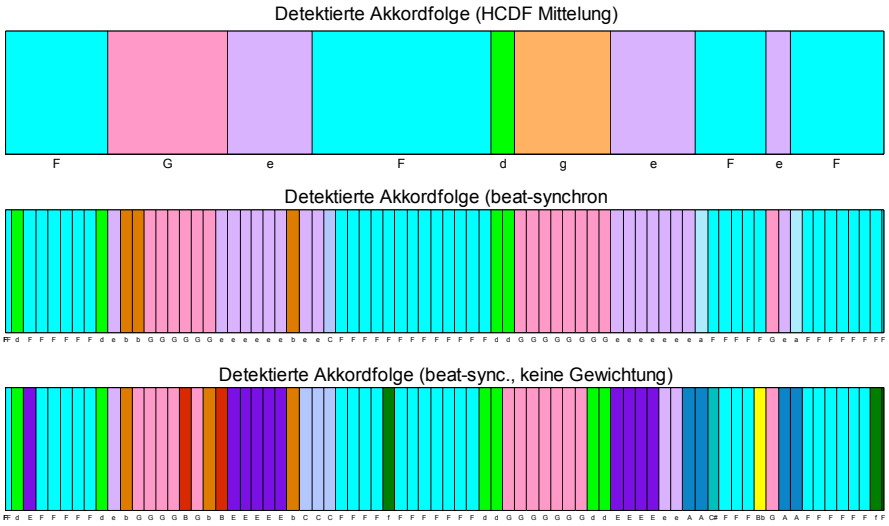

Abb. 3-12: Verbesserungen des Verfahrens zur Akkorddetektion: Mittelung nach HCDF (oben), beat-synchrone Mittelung mit Gewichtung (Mitte) und beat-synchrone Mittelung ohne Gewichtung (unten), (Sonnet, 43 s bis 68 s, der erste Refrain)

Zusammenfassung gleicher aufeinanderfolgender Akkorde

Durch Überbestimmung der Akkordwechselzeitpunkte durch die HCDF (es werden mehr Akkordwechsel erkannt, als tatsächlich vorhanden sind) ist es notwendig, gleiche Akkorde zusammenzufassen (Abb. 3-13 zeigt den Vergleich der Ausgabe der detektierten Akkordfolge ohne Akkordwiederholungen und entsprechend der HCDF tatsächlich bestimmten Akkorde inklusive Wiederholungen). Es wurde vorerst darauf verzichtet, die Zeitbasis bei der Akkordabfolge zu berücksichtigen, was sich hier darin äußert, dass der erste Akkord im Refrain fehlt, da es derselbe ist wie der letzte der Strophe. Für die Erkennung gleicher Teile ist das ein Problem, da unterschiedliches Zusammenfassen an sich gleicher Teile auf tatsächlich unterschiedliche Teile schließen lässt. Würde bei der Zusammenfassung der Akkorde Bezug auf den Takt (abgeleitet vom detektierten *beat*) genommen werden, könnte dieses Problem bereinigt werden.

Zusätzlich kommt es noch zur Falscherkennung von Akkorden. Die Ursache dafür sind wahrscheinlich Durchgangstöne und Vorhalte, die ohne musikalisches Wissen zwar in diesem Moment korrekt erkannt werden, für die Bestimmung der Akkordfolge trotzdem

unbrauchbar sind. Ebenso kann die oft sehr in den Vordergrund gemischte Gesangsmelodie die Akkorderkennung negativ beeinflussen.

Konkret wird in diesem Beispiel d-moll falsch erkannt, durch die nachfolgende Gewichtung wird der eigentlich richtig als G-Dur erkannte Folgeakkord ein g-moll. Die Parameter der Gewichtung sind anscheinend nicht optimal gewählt, hier herrscht Verbesserungspotential. Im Speziellen könnte eine Bewertung der Akkorddetektion erfolgen, die ihrerseits den Einfluss der Gewichtung steuert. Sicher erkannte Akkorde gehen damit vollständig in die Bewertung nachfolgender Akkorde ein, weniger sicher erkannte werden für die Gewichtung kaum oder gar nicht herangezogen.

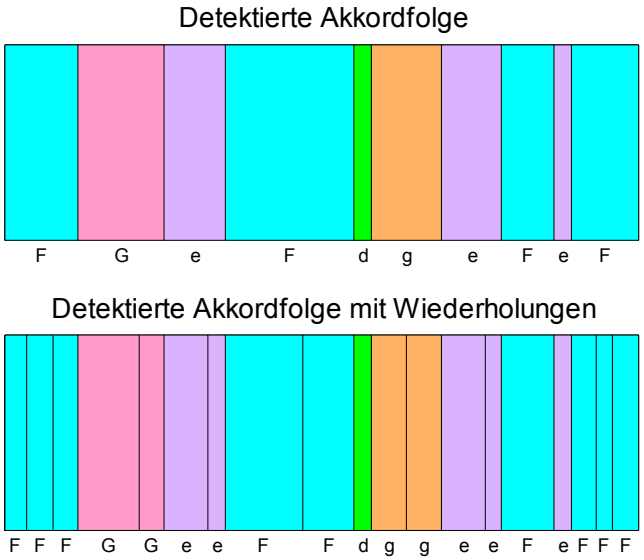

Abb. 3-13: Vergleich der detektierten Akkordfolge mit und ohne der Zusammenfassung gleichnamiger Akkorde (Sonnet, 43 s bis 68 s, der erste Refrain)

Probleme

Zur Veranschaulichung der Probleme der Strukturerkennung wird vorerst ein Ausschnitt aus „Sonnet“ bis zum zweiten Refrain betrachtet (Abb. 3-14).

Bei korrekter Akkorderkennung werden auch die Akkordmuster richtig erkannt, hier entspricht die Akkordfolge der Strophe (C-e-a-F) dem erkannten Formteil III. Im

Refrain gibt es Probleme bei der Erkennung, die richtige Abfolge (F-G-e-F / F-G-e-F / e-F)[1] wird nur teilweise korrekt bestimmt.

Es wird deutlich, dass zwar die Abfolge d-g an der richtigen Stelle noch einmal erkannt wird, der Teil allerdings trotzdem nicht entsprechend zugeordnet werden kann, da es zu zusätzlichen Fehldetektionen kommt. Statt einer weiteren Folge von e-F am Schluss des Refrains, wird e-a-F erkannt, womöglich durch diverse Verzierungen im Spiel hervorgerufen.

Wie zuvor schon erklärt, erscheint wegen der Zusammenfassung gleichnamiger Akkorde der erkannte Refrain kürzer als in Wirklichkeit. Zusätzlich sind geringfügige Variationen im Stück vorhanden, die in diesem Zusammenhang die Analyse dann allerdings zu genau machen und eine richtige Erkennung darunter leidet.

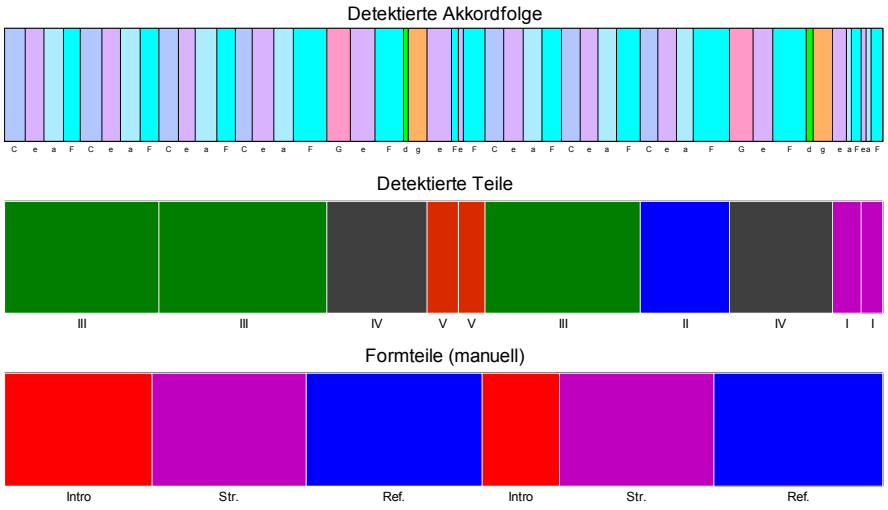

Abb. 3-14: Ergebnisse Sonnet (125 s, bis zum zweiten Refrain)

Die untenstehende Tabelle zeigt den Vergleich der detektierten Teile mit der manuellen Transkription der Akkorde (Tab. 3-2).

[1] Eigentlich e7, ist aber hier nicht relevant, da im Programm nur nach Tongeschlecht unterschieden wird.

Abschnitt	detektiert	manuell
1. Refrain, 1.Teil	G – e – F	F – G – e – F
1. Refrain, 2.Teil	d – g – e – F	F – G – e – F
1. Refrain, 3.Teil	e – F	e – F

Tab. 3-2: Vergleich der automatisch detektierten Teile mit der manuellen Transkription des Refrains von Sonnet

Programmparameter

Durch eine Vielzahl veränderbarer Parameter im Programm ist es möglich die Erkennung für einzelne Stücke zu optimieren. Als veränderbare Parameter bieten sich unter anderem folgende an: die untere Grenzfrequenz der *constant-Q* Transformation (die Wertigkeit des Bassbereichs kann so bestimmt werden), damit zusammenhängend die Fenstergröße und Schrittweite derselben (vgl. *constant-Q* Transformation, Abschnitt 2.5.1), außerdem ein Glättungsparameter in der HCDF, der großen Einfluss auf die Erkennung der Spitzen ausübt, des Weiteren ein Grenzwert zur Bestimmung relevanter Spitzen und der Abgleich der Menge der insgesamt gefundenen *peaks* in der HCDF, sowie die gesamte Gewichtung im Akkorddetektor.

Zusammenfassung der Probleme

- Die Akkorddetektion wird durch musikalische Geschehnisse (Durchgangstöne, Bassläufe, Gesang) beeinflusst und ist im harmonischen Zusammenhang so nicht immer gültig.

- Damit verbunden ist eine Optimierung der Gewichtung nötig, da falsch erkannte Akkorde eventuell zu noch mehr Falscherkennungen führen.

- Die Zusammenfassung gleicher Akkorde (wegen Überdetektion der Harmonieänderung durch die HCDF) erfolgt nicht *beat-,* oder taktsynchron, dadurch wird es für den Suchalgorithmus erschwert gleiche Teile zu finden.

- Der Suchalgorithmus ist nicht auf musikalische Bedürfnisse abgestimmt, auch hier wäre ein Zusammenhang mit der Taktzahl wünschenswert und würde zu besseren Ergebnissen führen.

Verbesserungsvorschläge

Um dem gerade angesprochenen Problem der Überdetektion der Harmonieänderung durch die HCDF entgegenzuwirken, wäre die Verbesserung der Bestimmung der Spitzenwerte der HCDF erstrebenswert. Eine Möglichkeit ist die Suche nach *peaks* unter Einbeziehung der Nachbarwerte und der Entfernung zu lokalen Minima, es ist auch eine adaptive Lösung denkbar, angepasst and das jeweilige Musikstück. Eventuell ist auch eine inhaltliche Verbindung der Bestimmung der Akkordmuster zu der Auswertung der HCDF möglich, entweder durch die taktsynchrone Auswertung oder die Auswahl von Stellen im Stück, an denen aus musikalischer Sicht Wechsel wahrscheinlich sind.

Die Akkorddetektion verfügt im momentanen Zustand nur über die Unterscheidung zwischen Dur und Moll. Eine Erweiterung zur Erkennung übermäßiger und verminderter Dreiklänge, sowie die Einbeziehung von Septakkorden ist möglich.

Wie auch Harte und Sandler berichten, ist darüber hinaus eine extra Auswertung des Bassbereichs denkbar, da in momentanen Lösungen der Akkordgrundton nicht erkannt werden kann (vgl. [Harte]). Durch die Bestimmung des Basstones, die Chroma Auswertung erfolgt beispielsweise von 55 Hz bis 110 Hz, wird es aber möglich, Akkorde genauer zu analysieren und z.B. Umkehrungen zu erkennen.

Außerdem ist die Möglichkeit in Betracht zu ziehen, die Tonart des Stückes zu erfassen und die nachfolgende Auswertung tonartspezifisch zu gewichten.

Im Zuge der Programmierung wurden Akkordprofile der gesamten Stücke gespeichert, sie können beispielsweise in einem zweiten Durchlauf des Detektionsalgorithmus als optimierte Maske für das jeweiligen Stück dienen.

Der „Sequitur“ Algorithmus ist an sich nicht das Optimum für die vorliegende Aufgabenstellung. Man sollte die Einbeziehung von Zeitinformation in die Erkennung

der Akkordfolgen überlegen (taktsynchrone Auswertung). Sehr ähnliche Segmente sollten idealerweise als gleich eingestuft und aus dem Kontext erkannt werden, kleinere Variationen im Stück sollen die Erkennung nicht dazu bewegen, neue Teile zu bestimmen.

3.1.4 Auswertung

Zur Auswertung wird nun jeweils eine von Hand erstellte Version der Formteile als Vergleich zu den automatisch erkannten Teilen herangezogen.

Sonnet (The Verve) (Abb. 3-15)

Die richtige Abfolge lautet:

Intro–Strophe–Refrain–Intro(WH)–Strophe–Refrain–Überleitung (Bridge)–Strophe–Outro

Da sich das Intro und die Strophe aus derselben Akkordfolge zusammensetzen, ist die Erkennung anfangs richtig. Der Teil IV entspricht dem Intro, der zweite Teil IV der Strophe und Teil VI dem Refrain. Da das Intro und die Strophe dieselben Akkorde verwenden ist auch die händische Auswertung an dieser Stelle nicht eindeutig, die automatische zumindest nicht falsch, wenn man Teil IV und Teil III als Intro, das an dieser Stelle wiederholt wird, mit Strophe bezeichnet. Teil VI entspricht dem zweiten Refrain, die Bridge wiederholt sich nicht und kann deshalb nicht als repetitiver Teil erkannt werden. Das in Abschnitt 3.1.3 behandelte Problem der nicht *beat*-synchronen Zusammenfassung von gleichen Akkorden führt nun dazu, dass die dritte Strophe als eigener Teil erkannt wird (Teil V). Ebenso wie die Bridge ist das Outro auch ein einmaliges Ereignis und wird als neuer Teil erkannt.

Die Stellen, an denen sich ein Refrain befindet, werden gefunden, die detektierte Länge ist dabei allerdings nicht korrekt.

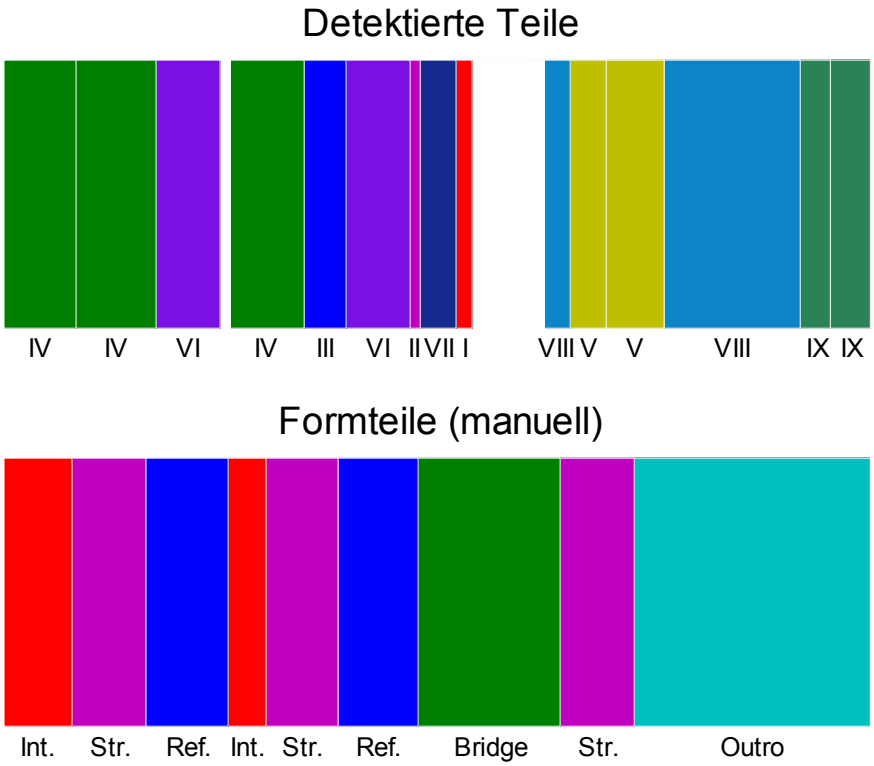

Abb. 3-15: Vergleich der detektierten Teile mit der manuellen Auswertung der Formteile (Sonnet)

Like a Virgin (Madonna) (Abb. 3-16)

Die richtige Abfolge lautet:

Intro–Strophe–Refrain–Strophe–Refrain–Bridge–Strophe–Refrain–Outro

Auch dieses Stück zeigt ähnliches Verhalten. Der Refrain wird an zwei von drei Stellen richtig erkannt (Teil V), der Strophe wird größtenteils die Abfolge der Teile II und IV zugewiesen. Auffällig ist hier, dass der erste Refrain zu lange detektiert wird und damit eine Erkennung der zweiten Strophe nicht mehr möglich ist. Das oben genannte und in Abschnitt 3.1.3 erläuterte Problem, nämlich die Zusammenfassung der Akkorde, ist die Ursache dafür.

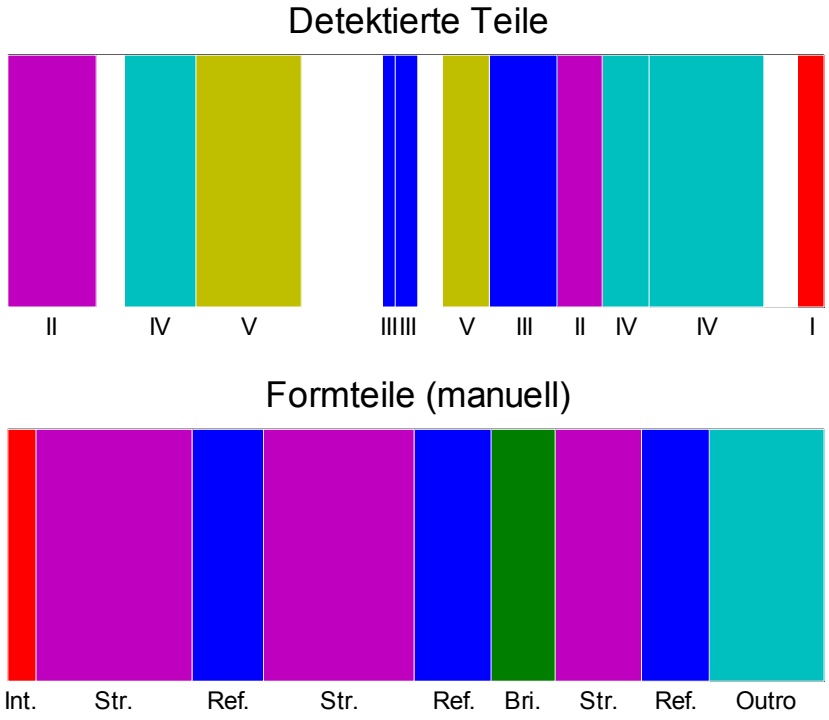

Abb. 3-16: Vergleich der detektierten Teile mit der manuellen Auswertung der Formteile (Like a Virgin)

Let it be (Beatles) (Abb. 3-17)

Die richtige Abfolge lautet:

Intro–Strophe–Refrain–Strophe–Refrain–Solo–Refrain–Strophe–Refrain–Outro

Obwohl sich die Teile dieses Stücks harmonisch stark ähnlich sind und trotz der beschriebenen Probleme im Suchalgorithmus konnten zwei der vier Refrainstellen zumindest teilweise erkannt werden (der Teil IV entspricht dem Beginn des Refrains), bei den Strophen sind keine eindeutigen Entsprechungen auszumachen. Die erste Strophe wird ebenso wie der erste und letzte Refrain als Teil I detektiert, die zweite Strophe zeigt sich als Wiederholung der Einleitung (Teil II) und dem zweiten Teil des Refrains (Teil III). Die dritte Strophe besteht allerdings aus den neuen Teilen V und VI. Der letzte Refrain schließlich, wird wie der erste als Teil I detektiert.

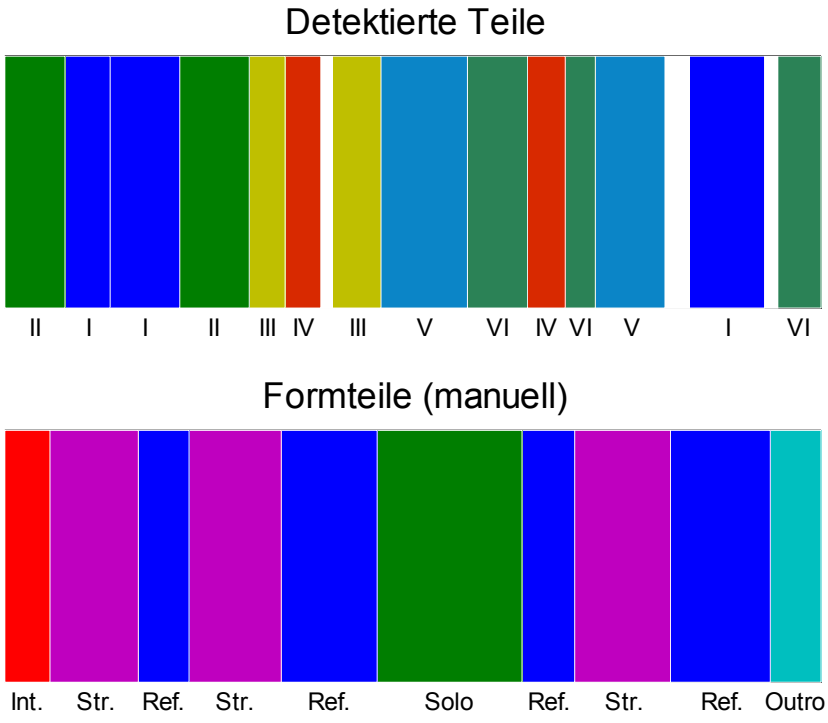

Abb. 3-17: Vergleich der detektierten Teile mit der manuellen Auswertung der Formteile (Let it be)

Real World (Matchbox 20) (Abb. 3-18)

Die richtige Abfolge lautet:

Intro–Strophe(1)–Strophe(2)–Strophe(1)–Strophe(2)–Refrain–Solo–Strophe(1)–Strophe(2)–Refrain–Bridge–Solo–Refrain–Outro

Bei diesem Titel scheint das Programm aufgrund der komplexeren Struktur größere Schwierigkeiten zu haben, trotz allem werden zwei von drei Stellen an denen sich richtigerweise der Refrain befindet zumindest teilweise erkannt (Teil X). Da die Strophe in sich in zwei deutlich unterscheidbare Abschnitte gegliedert ist und dadurch zusätzliche Muster erkannt werden, ist die Detektion hier mangelhaft, Teil IV könnte man eventuell als richtig werten.

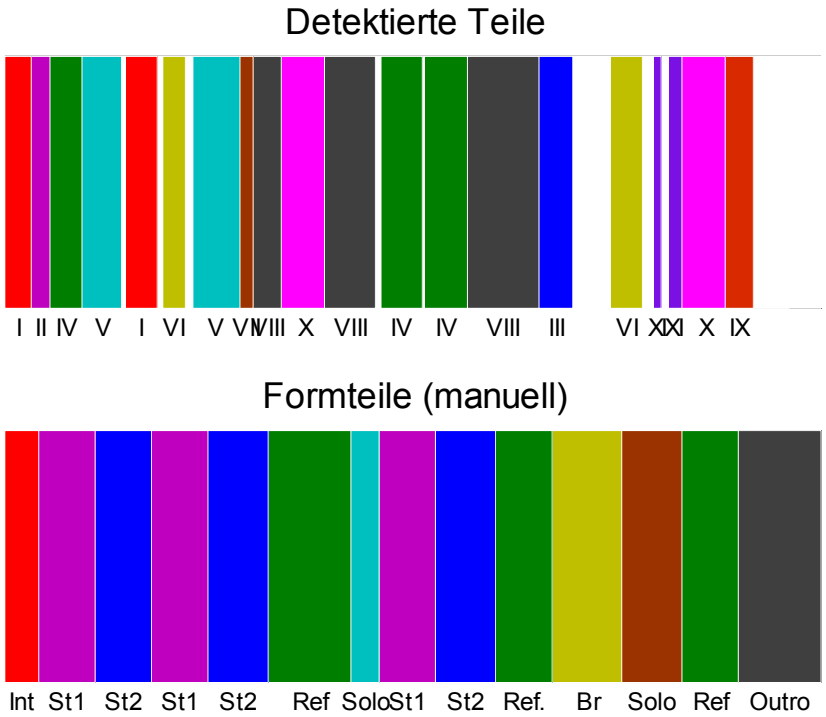

Abb. 3-18: Vergleich der detektierten Teile mit der manuellen Auswertung der Formteile (Real World)

Es wird gezeigt, dass das vorgeschlagene harmonische Modell das Potential zur Strukturerkennung aufweist und gegenüber der einfachen *beat*-synchronen Version durchaus Vorteile bringt, allerdings treten Probleme auf, die es für zukünftige Versionen des Algorithmus auszumerzen gilt.

3.2 Modell zur Ähnlichkeitsbestimmung

Dieses Modell wurde anhand der Unterlagen von Antti Eronen (vgl. [Eronen]) erstellt.

3.2.1 Übersicht

Eine Distanzmatrix (bestehend aus der Summe der *beat*-synchronen MFCC und Chroma Distanzmatrizen) dient als Ausgangspunkt für die Suche nach diagonalen Segmenten, die am wahrscheinlichsten einem Refrain entsprechen. Für weiterführende Filterung werden Methoden der Bildverarbeitung angewandt. Repetitive Teile können so gefunden werden, durch ein eigenständiges Bewertungssystem sollen daraus die dem Refrain entsprechenden Abschnitte ausgegeben werden (Abb. 3-19).

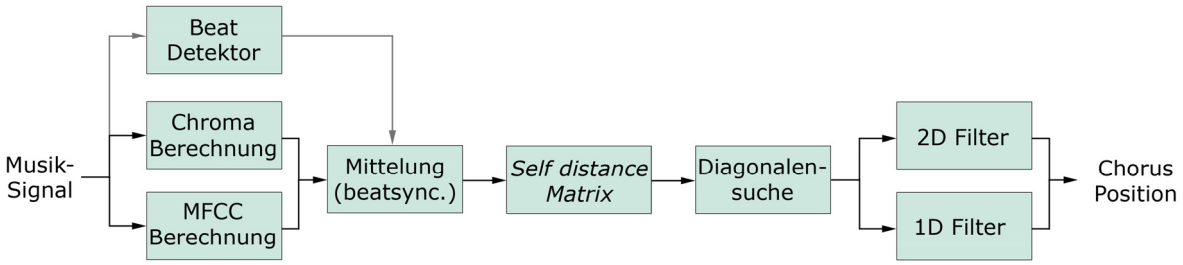

Abb. 3-19: Modell zur Ähnlichkeitsbestimmung

3.2.2 Detaillierte Beschreibung

Beat Detektor

Als *beat*-Detektor dient wiederum der *beat-tracking* Algorithmus von Ellis (vgl. Abschnitt 2.2.2).

Chroma Berechnung

Die Berechnung des Chromagramms erfolgt in diesem Programm mit Hilfe der FFT (Länge der *frames* ist 186 ms), nicht wie zuvor mit *Constant-Q* Berechnung. Es wird weiters mit der normalisierten Energie des Chroma Vektors gearbeitet.

MFCC Berechnung

Die Berechnung der MFCC erfolgt wie in Abschnitt 2.3.2 beschrieben, die Fensterlänge beträgt hier 30 ms.

Mittelung (*beat*-synchron)

Die *beat*-synchrone Mittelung erfolgt auch hier basierend auf dem ermittelten Tempo des *beat-trackers.* Der *beat* wird wie schon zuvor als Achtelzählzeit verstanden.

Distanzmatrizen

Die *self-distance matrix (SDM)* wird so berechnet, dass jeder Punkt dieser Matrix D(i, j) der euklidischen Distanz des Musiksignals zum Zeitpunkt i mit sich selbst zum Zeitpunkt j entspricht.

In nachstehender Abbildung (Abb. 3-20) ist erkennbar, dass entlang der Hauptdiagonale der Abstand natürlich Null beträgt. Außerdem sind weitere Diagonalen erkennbar, diese entsprechen den Kandidaten zur Bestimmung des Refrains.

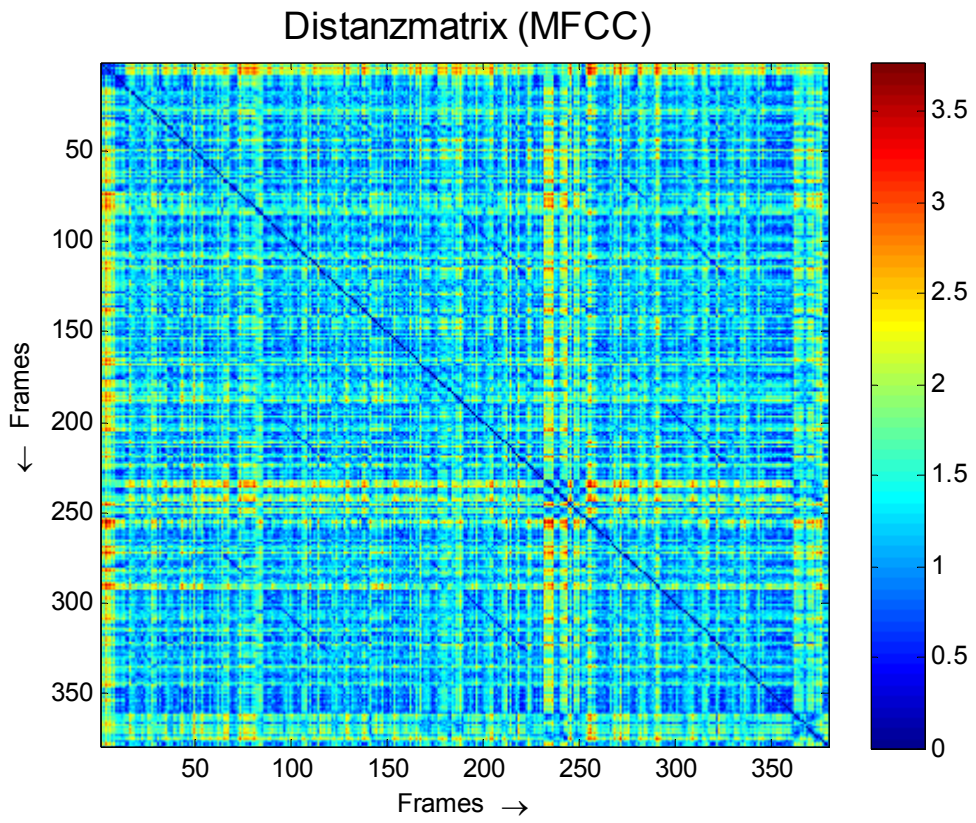

Abb. 3-20: MFCC Distanzmatrix (Like a virgin)

Dasselbe Verfahren wird auch auf die Chroma Vektoren angewandt, um die entsprechende Distanzmatrix zu erhalten (Abb. 3-21).

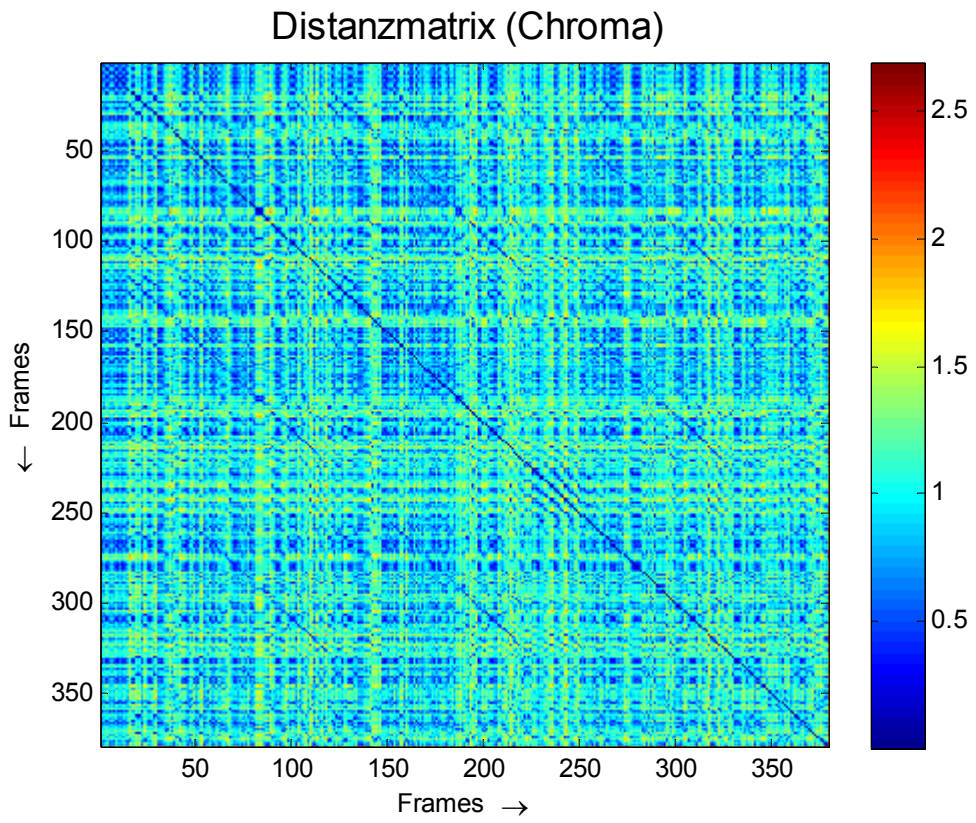

Abb. 3-21: Chroma Distanzmatrix (Like a virgin)

Um die Diagonalsegmente besser zu betonen wird die Chroma Distanzmatrix einer Filterung unterzogen und liefert die verbesserte (*enhanced*) Chroma Distanzmatrix (Abb. 3-22).

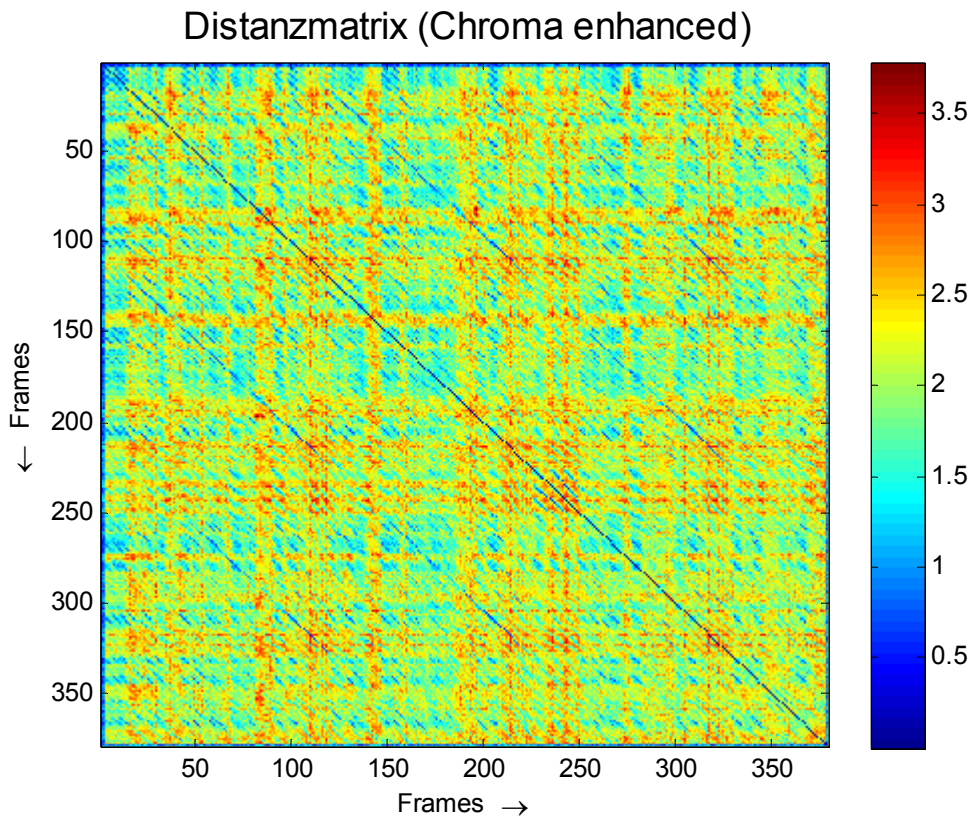

Abb. 3-22: *Enhanced* Chroma Distanzmatrix (Like a virgin)

Schließlich werden die MFCC Distanzmatrix und die gefilterte Version der Chroma Distanzmatrix summiert (Abb. 3-23).

Diese Matrix dient folglich als Ausgangspunkt weiterführender Berechnungen.

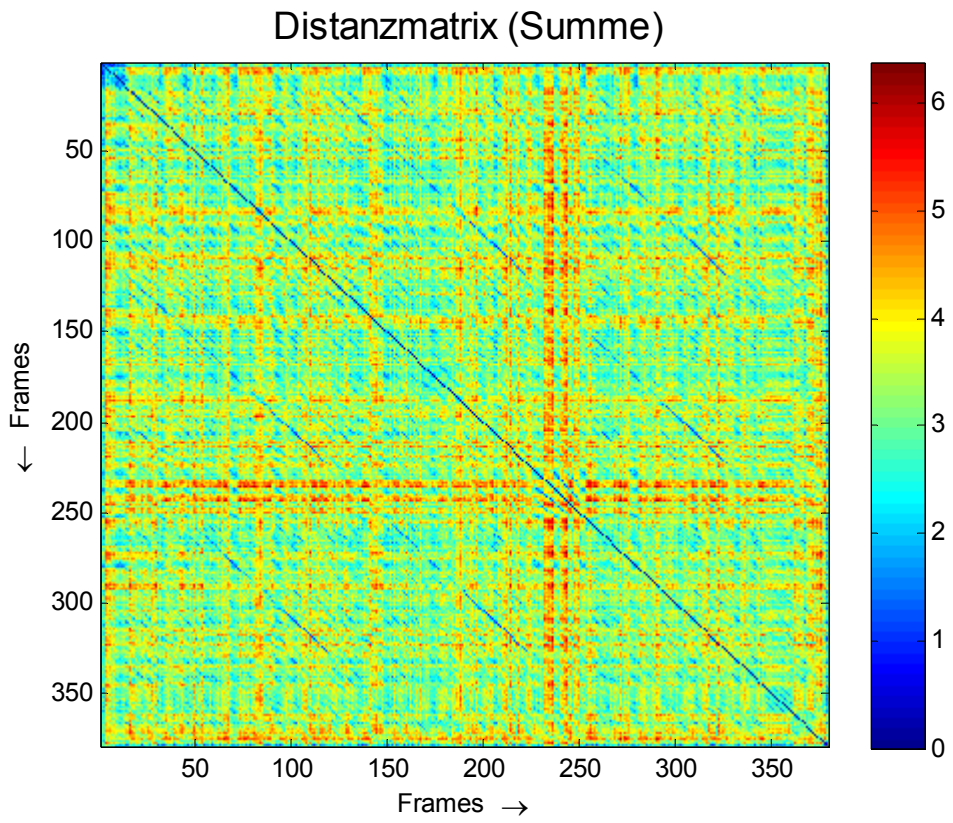

Abb. 3-23: Summe der Distanzmatrizen (MFCC und Chroma)

Um aus der summierten Distanzmatrix Diagonalen bestimmen zu können, wird eine Binarisierung durchgeführt.

Dazu erfolgt eine Suche nach Diagonalen mit geringen Summenwerten, ähnliche Teile haben ja eine kleine Distanz zueinander. Eine bestimmte Anzahl von Diagonalen (mindestens 10, je nach den Werten der Matrix) wird ausgewählt, zur Glättung gefiltert, und um die anschließende Suche zu erleichtern differenziert. Schließlich werden die Stellen an denen ein Vorzeichenwechsel stattfindet als Kandidaten vorgemerkt. Mit Hilfe der Methode von Otsu[1] erfolgt eine Klassifizierung und die Kandidaten für die Diagonalen zur Refrainsuche stehen fest. Nach der Filterung der tatsächlichen Diagonalen wird ein Grenzwert zur Binarisierung bestimmt und die Binarisierte Distanzmatrix kann erzeugt werden (Abb. 3-24).

[1] *Otsu's method:* Diese Methode wird in der Bildverarbeitung eingesetzt, um ein Bild mit Grauwerten mit Hilfe eines Schwellwertes in schwarz-weiß umzusetzen.

Um kurze Spalten in den so erhaltenen Diagonalen zu schließen erfolgt ein weiterer *enhancement* Prozess (dabei werden Stücke kürzer als 4 Sekunden verworfen und Kandidaten, die einander zu nahe sind entfernt) und liefert schließlich die *Enhanced Binarized Distance Matrix* (Abb. 3-25). Diese nun noch einmal optimierte Matrix dient als Ausgangspunkt für die folgenden Abschnitte.

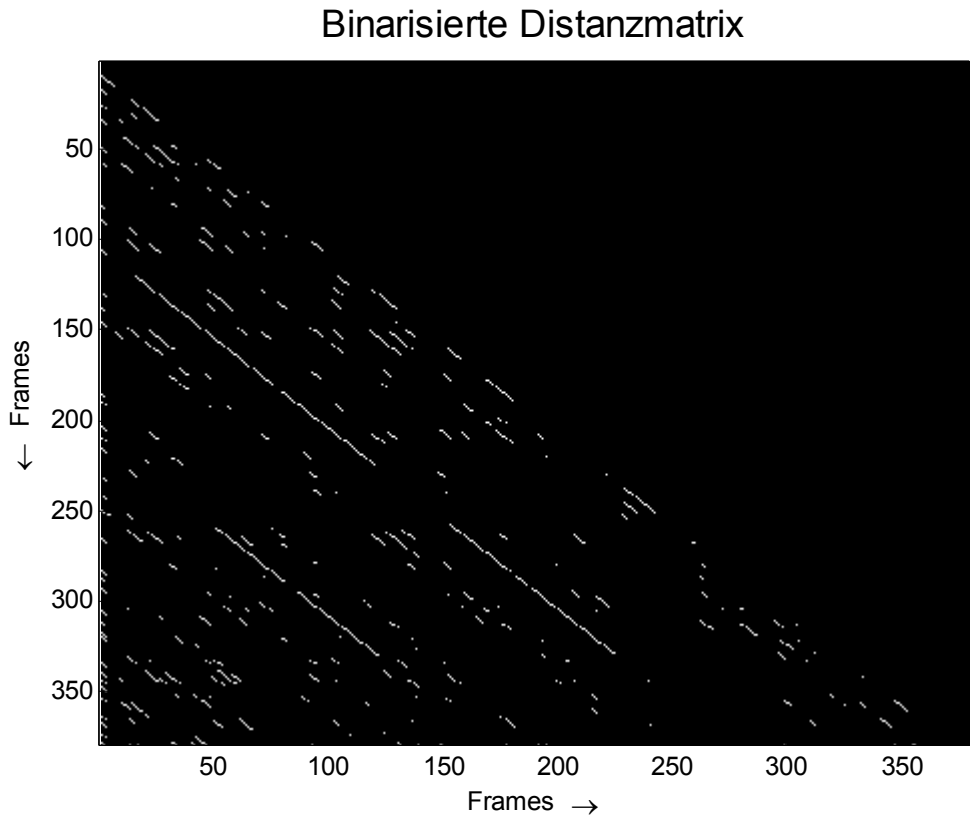

Abb. 3-24: Binarisierte Distanzmatrix (Like a Virgin)

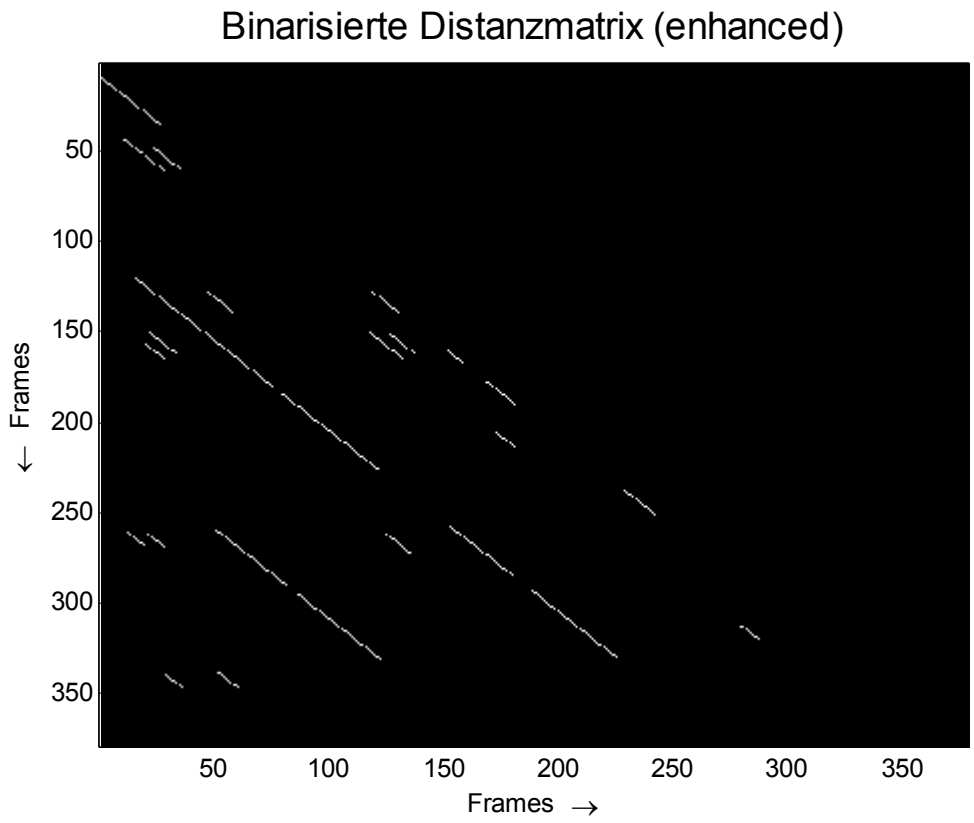

Abb. 3-25: Binarisierte Distanzmatrix, *enhanced* (Like a Virgin)

Diagonalensuche

Die in Abb. 3-25 weiß eingezeichneten Diagonalen sind die Ausgangspunkte der Suche nach dem Refrain.

Es wird nun ein Bewertungsschema eingeführt, das nach unterschiedlichen Gesichtspunkten eine Bewertung der Kandidaten durchführt.

Die Kriterien sind:

- Die Nähe zur erwarteten Position des Refrains im Stück
- Die Abstimmung der Positionen der Kandidaten untereinander
- Die Energie im korrespondierenden Teil des Musikstückes
- Die Distanz innerhalb eines Segments

- Die Anzahl der Wiederholungen

Die Auswertung dieser Kriterien liefert das Ausgangssegment, also den anfänglichen Refrain-Kandidaten.

Filterung

Als nächster Schritt erfolgt eine 2-dimensionale Filterung. Es wird davon ausgegangen, dass viele Stücke im 4/4 Takt vorliegen und dementsprechend der Refrain eine Länge von 8 oder 16 Takten umfasst (das entspricht 32 bzw. 64 *beats*). Meist erfolgt außerdem eine Wiederholung zweier gleicher Teile, ein bestimmtes Muster in der Distanzmatrix ist die Folge. Nach diesem Muster wird nun gesucht. Wird eine passende Sektion in der Matrix gefunden erfolgt ein Abgleich des Refrain-Kandidaten.

Ist ein solches Muster nicht vorhanden, werden die zuvor gefundenen Diagonalen selbst in Hinblick auf innere und äußere Distanzen gefiltert. Auch in diesem Fall erfolgt ein Abgleich der Refrain-Kandidaten und die Position des detektierten Refrains wird ausgegeben.

3.2.3 Ergebnisse

Solange das Schema Strophe–Refrain–Strophe–Refrain–Refrain ungefähr eingehalten wird, funktioniert der Algorithmus gut (vgl. Tab. 3-3).

Beim Titel „Sonnet“ ist die Fehldetektion leicht erklärbar. Im Outro wird ein kurzer Teil 9mal identisch wiederholt. Obwohl hierbei Kriterien wie die Position des Refrains offensichtlich nicht erfüllt sind, ist die große Anzahl der Wiederholungen für den Algorithmus anscheinend ein sicheres Zeichen für einen Refrain.

Musiktitel	Refrain (detektiert)	Refrain (manuell)
Let It Be	1:20-1:48	1:18-1:45
Like a Virgin	1:35-1:51	1:35-1:53
Sonnet	3:21-3:31 (falsch, ist Outro mit sich wiederholdenden Teilen)	z.B.:1:41-2:05
Real World	1:21-1:29	1:15-1:39

Tab. 3-3: Vergleich der detektierten Position des Refrains mit der manuellen Auswertung

Die Suche auf Basis der Ähnlichkeiten zeigt gute Ergebnisse, solange eine einfache Form (z.B. Strophe-Refrain-Refrain) eingehalten wird.

3.3 Vergleich der Modelle

Das erste Modell ist auf Grund der Verknüpfung mit den Harmonien eines Stückes wesentlich komplexer. Eine Auswertung der Formteile wäre gut möglich, durch die große Anzahl an veränderbaren Parametern die es zu optimieren gilt und die angesprochenen Probleme (vgl. Abschnitt 3.1.3) ist in der momentanen Form eine automatisierte Strukturbeschreibung nicht möglich.

Das zweite Modell funktioniert sehr gut, solange sich das Musikstück an die Vorgaben hält, also die Abfolge von Strophe und Refrain standardmäßig ausfällt. Im Bereich Rock/Pop ist das oft der Fall, falls nicht ist das Modell allerdings zu unflexibel.

4 Schlussfolgerung

Das Ziel der untersuchten Modelle ist es, einerseits durch Segmentierung harmonisch gleicher Teile und das Auffinden gleicher Akkordsequenzen die Struktur zu bestimmen, andererseits durch die Bestimmung von Ähnlichkeiten repetitive Teile zu erkennen und unter der Bedingung, dass im Genre Pop/Rock einfache Liedformen vorherrschen, den Refrain zu bestimmen.

Das vorgeschlagene harmonische Modell zeigt das Potential zur Strukturerkennung, es treten allerdings Probleme auf, die es für zukünftige Versionen des Algorithmus auszumerzen gilt.

Verbesserungsvorschläge sind unter anderem:

- Optimierung der *peak* Detektion der HCDF
- Erweiterung der Akkordmatrizen
- Separate Auswertung des Bassbereiches
- Tonartbestimmung zur Verbesserung der Akkorderkennung
- Adaptierung des Suchalgorithmus auf musikalische Notwendigkeiten

Die Suche auf Basis der Ähnlichkeiten zeigt gute Ergebnisse, solange eine einfache Form (z.B. Strophe-Refrain-Refrain) eingehalten wird, was im Bereich Rock/Pop oft der Fall ist. Das Modell ist allerdings zu unflexibel um auf Abweichungen zu reagieren.

Im Bereich Rock/Pop ist das oft der Fall, wenn nicht, ist das Modell allerdings zu unflexibel.

Die Strukturbeschreibung eines Musikstückes ist mit den vorgestellten Methoden eingeschränkt möglich, auf das im Titel der Arbeit angesprochene „Labelling“, also der Bezeichnung der erkannten Teile muss in dieser Phase allerdings noch verzichtet werden, zu ungenau sind die Ergebnisse, zu viele Probleme treten auf.

5 Literaturverzeichnis

[Allamanche] ALLAMANCHE Eric: *Content-based Identification of Audio material Using MPEG-7 Low Level Description.* ISMIR 2001

[Aucouturier] AUCOUTURIER Jean-Julien, PACHET Francois: *Music Similarity Measures: What's the Use?*. IRCAM - Centre Pompidou, 2002

[Bartsch] BARTSCH Mark A., WAKEFIELD Gregory H.: *To catch a Chorus: Using Chroma-based representations for Audio Thumbnailing.* New Paltz, New York, 2001

[Bello1] BELLO Juan P.: *A Tutorial on Onset Detection in Music Signals.* IEEE Transactions on Speech and Audio Processing, 2005

[Bello2] BELLO Juan P., PICKENS Jeremy: *A Robust Mid-level Representation for Harmonic Content in Music Signals.* Queen Mary, University of London, 2006

[Blankertz] BLANKERTZ Benjamin: *The Constant Q Transform.*

[Brown] BROWN Judith C.: *Calculation of a constant Q spectral transform.* Journal of the Acoustical Society of America, vol.89, no.1, 1991

[Chai] CHAI Wei, VERCOE Barry: *Structural Analysis of Musical Signals for Indexing and Thumbnailing.* Proceedings of the 2003 Joint Conference on Digital Libraries (JCDL'03), 2003

[Cooper] COOPER Matthew, FOOTE Jonathan: *Summarizing popular music via Structural Similarity Analysis.* IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2003

[Ellis1] ELLIS Daniel P.W.: *Beat Tracking with Dynamic Programming.* MIREX 2006 Audio Beat Tracking Contest system description, 2006

[Ellis2] ELLIS Daniel P.W., POLINER Graham E.: *Identifying „Cover Songs" with Chroma Features and Dynamic Programming Beat Tracking.* LabROSA, Dept. of Electrical Engineering, Columbia University, New York, USA

[Eronen] ERONEN Antti: *Chorus Detection with combined use of MFCC and Chroma Features and Image Processing Filters.* 10th Int. Conference on Digital Audio Effects (DAFx-07), Bordeaux, France, 2007

[Foote] FOOTE Jonathan T.: *Content-Based Retrieval of Music and Audio.* SPIE 1997

[Goto1] GOTO Masataka, MURAOKA Yoichi: *Issues in Evaluating Beat Tracking Systems.* IJCAI-97 Workshop on Issues in AI and Music

[Goto2] GOTO Masataka: *A Chorus Section Detection Method for Musical Audio Signals and Its Application to a Music Listening Station.* IEEE, 2006

[Harte1] HARTE Christopher, SANDLER Mark B.: *Automatic Chord Identification Using a Quantised Chromagram.* AES Convention Paper 6412, 118th Convention, Barcelona, Spain, 2005

[Harte2] HARTE Christopher, SANDLER Mark, GASSER Martin: *Detecting Harmonic Change in Musical Audio.* AMCMM '06 Santa Barbara, California, USA, 2006

[Herrera] HERRERA Perfecto: *Towards instrument segmentation for music content description: a critical review of instrument classification techniques.* ISMIR 2000

[Klapuri] KLAPURI Anssi P., ERONEN Antti J., ASTOLA Jaako T.: *Analysis of the Meter of Acoustic Musical Signals.* Institute of Signal Processing, Tampere University of Technology, Finland, 2006

[LabROSA] ELLIS Dan: *Music Beat Tracking and Cover Song Identification.* http://labrosa.ee.columbia.edu/projects/coversongs/, 20.3.2008

[Logan] LOGAN Beth: *Mel Frequency Cepstral Coefficients for Music Modeling.* Cambridge Research Laboratory

[McKinney] McKINNEY Martin F., MOELANTS Dirk: *Deviations from the resonance theory of tempo induction.* Proceedings of the Conference on Interdisciplinary Musicology (CIM04), 2004

[Michels] MICHELS Ulrich: *dtv-Atlas Musik.* Deutscher Taschenbuch Verlag GmbH & Co. KG, München, 2005

[MIREX] MIREX-06: *Audio tempo and beat evaluations.* http://www.music-ir.org/mirex/2006/index.php/Audio_Beat_Tracking, 20.3.2008

[Nevill] NEVILL-MANNING Craig G.: *Identifying Hierarchical Structure in Sequences: A linear-time algorithm.* Al Access Foundation and Morgan Kaufmann Publishers, 1997

[Ong] ONG Bee Suan, HERRERA Perfecto: *Semantic Segmentation of Music Audio Contents.* Universitat Pompeu Fabra, Barcelona, Spanien

[Scheirer] SCHEIRER Eric D.: *Tempo and beat analysis of acoustic musical signals.* Journal of the Acoustical Society of America 103 (1), 1998

[Seppänen] SEPPÄNEN Jarno, ERONEN Antti, HIIPAKKA Jarmo: *Joint Beat & Tatum Tracking from Musical Signals.* University of Victoria, 2006

[Shepard] SHEPARD Roger. N.: *Circularity in judgments of relative pitch.* Journal of the Acoustical Society of America 36, 1964

[Tzanetakis] TZANETAKIS George, ESSL Georg, COOK Perry: *Automatic Musical Genre Classification Of Audio Signals.* ISMIR 2001

[Yoshioka] YOSHIOKA Takuya: *Automatic Chord Transcription with concurrent Recognition of Chord Symbols and Boundaries.* Universitat Pompeu Fabra, 2004.

[Zwicker] FASTL Hugo, ZWICKER Eberhard: *Psychoacoustics – Facts and Models.* Springer-Verlag Berlin Heidelberg, 2007

6 Anhang A: Übersicht der Matlab Programme

6.1 Programm zur harmonischen Analyse

Die Funktion „harmonic_analysis“ wird entweder nur mit dem Pfad der Audiodatei aufgerufen (z.B.: harmonic_analysis('sonnet_11m.wav')), es ist allerdings auch möglich folgende Parameter zusätzlich zu übergeben:

smoothing: Glättungswert für die HCDF

threshold: Schwellwert für die *peak-picking* Funktion der HCDF

f0, fmax: untere und obere Grenzfrequenz der *constant-Q* Transformation

buffersize, hopsize: die Fenstergröße und Schrittweite der *constant-Q* Transformation

Die in nachfolgender Darstellung (vgl. Abb. 6-1) gezeigten Module werden nacheinander abgearbeitet. Nach dem *beat-tracker* erfolgt die Berechnung der Chroma-Vektoren mit anschließender Mittelung (*beat*-synchron). Auf diese gemittelten Vektoren wird die HCDF angewandt und die Chroma Werte werden ein weiteres Mal gemittelt, diesmal entsprechend den Zeitpunkten, die die HCDF bestimmt. Das nächste Modul ist der Akkord-Detektor, der außerdem den jeweils vorhergegangenen Akkord zur Gewichtung übergibt. Es folgt sie Mustererkennung und die abschließende grafische Ausgabe.

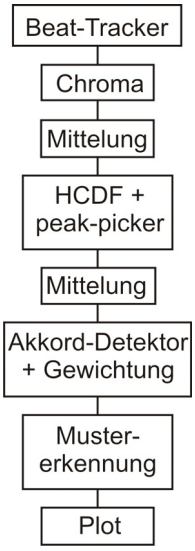

Abb. 6-1: Flussdiagramm zum Programm „Harmonische Analyse“

6.2 Programm zur Ähnlichkeitsbestimmung

Die Funktion „similarity_analysis” wird mit dem Pfad und Dateinamen der Audiodatei aufgerufen, z.B.: similarity_analysis('likeavirgin_22m.wav'). Die einzelnen Module (vgl. Abb. 6-2) werden nacheinander bearbeitet. Nach dem *beat-tracker* erfolgt sie Berechnung der *beat*-synchronen MFCC und der dazugehörigen Distanzmatrix. Ebenso wird danach das Chromagramm berechnet, daraus die Distanzmatrix erstellt, diese allerdings auch noch zur Verbesserung durch ein 2D Filter geschickt (*enhancement*). Als nächster Schritt folgt die Bestimmung der für den Refrain geeigneten Diagonalen und anschließende Binarisierung. Danach werden die Segmente der Refrain-Kandidaten gesucht und mit Hilfe eines Bewertungsschemas (*scoring*) wird die exakte Position des Refrains bestimmt.

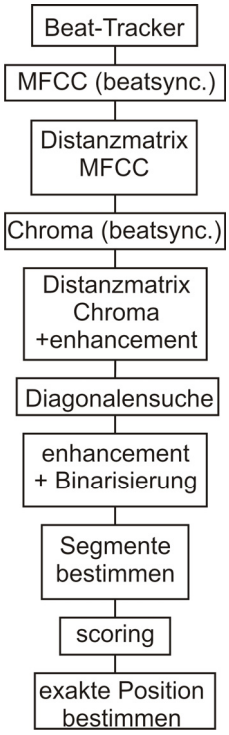

Abb. 6-2: Flussdiagramm zum Programm „Ähnlichkeitsbestimmung“