JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis

Automatische Erkennung von Cover-Versionen und Plagiaten in moderner Pop- und Rockmusik

MAGISTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

in der Studienrichtung

INFORMATIK

Eingereicht von:
Bernhard Niedermayer, Bakk.techn., 0156081

Angefertigt am:
Institut für Computational Perception

Betreuung:
Univ.Prof. Dr. Gerhard Widmer

Linz, September 2007

Kurzfassung

Diese Arbeit befasst sich mit der automatischen Erkennung unterschiedlicher Versionen eines Musikstücks. Für derartige Klassifizierungs-Aufgaben ist es notwendig, aussagekräftige Merkmale aus den vorhandenen Daten zu extrahieren. Neben zwei Ansätzen aus der Literatur, die sich jeweils auf das gesamte Musikstück beziehen, werden zwei weitere vorgestellt, die auf der Idee basieren, einen Song nur durch seinen Refrain zu repräsentieren. Dabei werden in dieser Arbeit alle Arbeitsschritte sowie notwendige Grundlagen zur Extraktion dieser vier Features aus den jeweiligen Audio-Signalen beschrieben.

Auf Basis der berechneten Features erfolgt die Klassifizierung. Dabei geht es darum zu entscheiden, ob zwei verglichene Musikstücke unabhängig voneinander sind oder nicht. Neben dem aus der Sprachverarbeitung bekannten *Dynamic Time Warp*-Algorithmus wird in dieser Arbeit eine zweite Methode vorgestellt, die eine konstante Tempo-Änderung zwischen zwei Versionen eines Musikstücks unterstellt.

Anhand eines experimentellen Prototyps findet die Evaluierung und Diskussion aller Kombinationen aus extrahiertem Feature und Klassifizierungsmethode statt. Den Abschluss dieser Arbeit bildet ein Ausblick auf Verbesserungsmöglichkeiten als Anregung für weiterführende Forschung.

Abstract

This thesis is dedicated to the automatic detection of different versions of one piece of music. In order to carry out such classification tasks it is necessary to extract meaningful characteristic from given data. Besides two known approaches that refer to the whole piece, two new ones are introduced. Those features are based on the idea that a song is represented by its chorus only. This thesis describes all the necessary steps as well as relevant basics in order to extract those four features from arbitrary audio signals.

Based upon the computed features the actual classification task can be performed. It is to decide whether two compared pieces of music are independent from each other or not. As well as the *Dynamic Time Warp* algorithm, which is know from the field of speech processing, another approach is introduced. It is based upon the assumption that the variation in tempo is constant between two versions of on piece of music.

All possible combinations of features and classification methods are evaluated and discussed using an experimental prototype. This thesis is completed by a brief recapitulation and some prospects for future work.

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Motivation 1
1.2 Ziele und Beitrag 2
1.3 Aufbau 3

2 Grundlagen **5**
2.1 Cover-Versionen 5
2.2 Musikalische Grundlagen 8
2.2.1 Skalen 8
2.2.2 Tonsysteme 9
2.2.3 Obertöne, Mehrklänge und Harmonien 15

3 Feature Extraktion **18**
3.1 Arbeitsablauf 19
3.2 Spektralanalyse 21
3.2.1 Diskrete Fourier Transformation 21
3.2.2 Fast Fourier Transformation 22
3.2.3 Fensterung 23
3.2.4 Zeit- und Frequenz-Auflösung bei der Fourier Transformation . . 25
3.2.5 Constant Q Transformation 26
3.2.6 Übergangserkennung 29
3.2.7 Peak Filterung 30
3.2.8 Maskierung 32
3.2.8.1 Absolute Hörschwelle 33
3.2.8.2 Kritische Bänder 33
3.2.8.3 Simultane Maskierung 35
3.2.8.4 Nicht simultane Maskierung 36
3.3 Referenz Frequenz 37
3.3.1 Referenzfrequenz und Obertonreihen 40
3.3.2 Resultate 41
3.4 Harmonic Pitch Class Profile 42
3.4.1 Gewichtungsfunktion 44
3.4.2 HPCP und Obertonreihen 45
3.4.3 Normalisierung 48

3.5 Transposed Harmonic Pitch Class Profile 48
3.5.1 HPCP für Framegruppen 49
3.5.2 Bestimmung der Tonart 49
3.6 Melodieerkennung 55
3.6.1 PreFEst 55
3.6.1.1 Augenblicksfrequenzen und Kandidaten 56
3.6.1.2 Grundtöne 57
3.6.1.3 Trajektorien 60
3.6.1.4 Erweiterungen 61
3.6.2 Weitere Verfahren zur Melodieextraktion 61
3.6.3 Resultate 64
3.7 Strukturanalyse 67
3.7.1 Selbstähnlichkeits-Matrizen 68
3.7.2 Erkennung wiederholter Abschnitte 70
3.7.3 Arrangieren von Wiederholungen 72
3.7.4 Modulierte Wiederholungen 72
3.7.5 Auswahl des Refrains 73
3.7.6 Variationen 76

4 Klassifizierung **78**
4.1 Ähnlichkeits-Matrizen 79
4.2 Dynamic Time Warp 81
4.3 Linearer Zeitzusammenhang 82

5 Resultate und Perspektiven **86**
5.1 Evaluierung 86
5.2 Erkenntnisse 91
5.3 Perspektiven 93

Literaturverzeichnis **95**

Abbildungsverzeichnis

2.1 Zusammenhang zwischen Bark-, Mel- und Frequenz-Skala 10
2.2 C Dur-Tonleiter 11
2.3 a Moll-Tonleiter 11
2.4 Flaches Modell leitereigener Töne für Dur 11
2.5 Flaches Modell leitereigener Töne für natürliches Moll 12
2.6 Flaches Modell leitereigener Töne für harmonisches Moll 12
2.7 Flaches Modell leitereigener Töne für melodisches Moll (aufsteigend) . . 13
2.8 Empirisches Modell nach Krumhansl ([24]) für Dur 13
2.9 Empirisches Modell nach Krumhansl ([24]) für Moll 14
2.10 Empirisches Modell nach Temperley ([45]) für Dur 14
2.11 Empirisches Modell nach Temperley ([45]) für Moll 15
2.12 Modell nach Chai ([9]) für Dur 15
2.13 Modell nach Chai ([9]) für Moll 16
2.14 Dreiklangs-Modell nach Gómez ([18]) für Dur 16
2.15 Dreiklangs-Modell nach Gómez ([18]) für Moll 17

3.1 Arbeitsablauf zur Extraktion der Features 20
3.2 DFT des Rechteckfensters 24
3.3 Vergleich von Rechteck- und Hamming-Fenster (a) sowie von Rechteck- und Blackman-Harris-Fenster (b) 25
3.4 Vergleich verschiedener Frequenzauflösungen 27
3.5 Gefundene Peaks in einem Ausschnitt aus *Yesterday* von den *Beatles* . . 32
3.6 Absolute Hörschwelle des Menschen ([34]) 34
3.7 Asymmetrie der Maskierung zwischen Tönen und Rauschen ([36]) 36
3.8 Maskierung außerhalb des eigenen kritischen Bands ([36]) 36
3.9 Maskierungsschwellwerte ausgehend von Tönen (a) und Rauschen (b) sowie die daraus resultierenden globalen Werte (c) (aus [36]) 37
3.10 Einzelwerte und Gesamt-Histogramm zur Bestimmung der Referenzfrequenz am Beispiel von *Yesterday* von den *Beatles* 43
3.11 Ausschnitt der momentanen HPCP-Vektoren aus *Yesterday* 47
3.12 Ausschnitt der Melodie des Refrains von *Yesterday* 47
3.13 Summe aller einzelnen HPCP-Vektoren von *Yesterday* 48
3.14 Modifiziertes Modell anhand des flachen Dreiklangsmodells für Dur . . . 52

3.15 Modifiziertes Modell anhand des flachen Dreiklangsmodells für Moll . . 52
3.16 Summe aller THPCP-Vektoren des Stücks *Yesterday* von den *Beatles* . . 54
3.17 Ermittelte und tatsächliche Melodie eines Stücks aus dem *ISMIR 2004 Melody Extraction Contest* Testset 65
3.18 Ermittelte Melodie des Refrains von *Yesterday* 66
3.19 Ausschnitt der Melodie des Refrains von *Yesterday* 66
3.20 Selbstähnlichkeits-Matrix des Stücks *Yesterday* von den *Beatles* 69
3.21 Selbstähnlichkeits-Matrix aus Abbildung 3.20 in der Time-Lag-Darstellung 70
3.22 Time-Lag Darstellung der Ähnlichkeits-Matrix mitsamt den berechneten Werten R_{all} für jede Zeile 71
3.23 Wiederholte Passagen in *Yesterday* 75

4.1 Hough-Transformierte der Ähnlichkeitsmatrizen 84

5.1 Gegenüberstellung der Precision-Recall-Kurven der besten Klassifizierungsmethoden pro Feature 91

Tabellenverzeichnis

2.1 Obertöne von c 16

3.1 Bekannte Fensterfunktionen 25
3.2 Verschiebung der Obertöne 41
3.3 Ermittelte Referenzfrequenzen bekannter Titel der *Beatles* 42
3.4 Lage der Obertöne im HPCP-Profil 46
3.5 Ermittelte Tonarten bekannter Titel der *Beatles* 54
3.6 Lage der Refrain-Stellen in *Yesterday* 74

5.1 Zur Evaluierung verwendete Musikstücke 89
5.2 Ergebnisse der Coverversions-Erkennung bei einem konstanten $fallout$-Wert von 0.1 90
5.3 Maximale f-Werte der besten Klassifizierungsmethoden pro Feature . . . 90

1 Einleitung

Die vorliegende Arbeit widmet sich der automatischen Erkennung verschiedener Versionen eines Musikstücks. In diesem Kapitel wird der Nutzen, den Systeme mit derartiger Funktionalität bringen, erläutert und die genauen Ziele sowie der Aufbau dieser Arbeit dargelegt.

1.1 Motivation

Die Menge der zur Zeit erhältlichen Musikstücke ist in unüberschaubare Größen gewachsen. Allein der Online-Shop *iTunes*[1] bietet bereits über 5 Millionen Titel zum Verkauf an. Aber auch bei privaten Musiksammlungen ist ein Umfang von mehreren tausend Songs schon lange keine Seltenheit mehr. In diesem Kontext gewinnen Systeme zur automatischen Verwaltung und Organisation derartiger Musik-Sammlungen zunehmend an Bedeutung, zumal die Benutzer manchmal gar nicht mehr wissen, welche Musikstücke sie im einzelnen tatsächlich besitzen.

Vor diesem Hintergrund entwickelte sich das Gebiet des *Music Information Retrieval*, dem auch diese Arbeit zuzurechnen ist. Es beschäftigt sich damit, aus den unterschiedlichen Formen, in denen Musik vorliegen kann – angefangen von Audio-Signalen in verschiedenen Formaten, über symbolische Repräsentationen wie MIDI-Dateien bis hin zu einfachen Notenblättern – möglichst vielseitige Metadaten zu extrahieren. In der Literatur oft behandelte Aufgabenstellungen sind etwa die automatische Zuordnung eines Songs zu einem Genre, das Erkennen der Melodiestimme in polyphonen Stücken oder die Empfehlung bestimmter Songs aufgrund von Ähnlichkeiten. Die Methoden, die dazu verwendet werden, stammen aus den Bereichen der Signalverarbeitung, der Mustererkennung, dem Data- sowie Web-Mining, dem Maschinellen Lernen und der Psychoakustik.

Die spezielle Aufgabenstellung im Rahmen dieser Arbeit ist das Erkennen unterschiedlicher Versionen eines gleichen Musikstücks. Der Nutzen, der einem Benutzer daraus

[1] http://www.apple.com/at/itunes/

entsteht, kann anhand zweier Anwendungen dargestellt werden. Zum einen dient ein System, das automatische Versions-Erkennung beherrscht, der besseren Verwaltung von Musiksammlungen. Im Trivialfall können Duplikate, die aufgrund von unterschiedlichen Dateiformaten und Größen nicht sofort als solche ersichtlich sind, erkannt und eliminiert werden. Der eigentliche Anwendungsfall bezieht sich jedoch auf die Navigation innerhalb großer Musiksammlungen – etwa die Suche nach Disco- oder Karaoke-Versionen eines bestimmten Songs.

Zum anderen existiert ein zweiter bedeutender Anwendungsfall der einen juristischen Hintergrund hat. Da Musik als geistiges Eigentum grundsätzlich geschützt ist, besteht für Künstler und Produzenten ein Interesse daran, ob andere Personen Versionen ihrer Musikstücke veröffentlichen. Gerade bei weniger bekannten Interpreten und Titeln, bei denen Plagiate nicht sofort einer breiten Masse an Hörern auffallen, ist ein System, das diese Aufgabe übernimmt von Interesse. Ab welchem Veränderungsgrad ein Song jedoch als eigenständig zu bewerten ist, kann nicht eindeutig beschrieben werden, was die Entwicklung derartiger Systeme zusätzlich erschwert.

Im Rahmen dieser Arbeit wurde ein experimenteller Prototyp eines Systems zur Erkennung von Versionen gleicher Musikstücke entwickelt. Dieser ist so konzipiert, dass sowohl die Methode der Entscheidungsfindung, ob es sich bei zwei Songs um unabhängige handelt oder nicht, als auch die Features, die als Grundlage dieser Entscheidung dienen, austauschbar sind.

1.2 Ziele und Beitrag

Ziel dieser Arbeit ist die Untersuchung von Methoden zur Erkennung von Cover-Versionen bzw. Plagiaten in moderner Pop- und Rockmusik. In der Literatur sind dazu im Wesentlichen zwei Features beschrieben anhand derer eine diesbezügliche Klassifizierung erfolgen kann. Das eine ist die Melodie des Musikstücks, die aus den vorhandenen Daten extrahiert werden muss, das andere das *Transposed Harmonic Pitch Class Profile*. Dabei handelt es sich um eine Reihe an Chroma-Vektoren, die zu jedem Zeitpunkt die relativen Präsenzen aller Töne einer Oktave beschreiben. Das Merkmal ist zudem robust gegenüber Modulationen.

Neben dem Vergleich dieser beiden Features besteht ein Beitrag dieser Arbeit darin, zwei weitere Features zur Versions-Erkennung vorzustellen. Diese basieren auf den bereits genannten, unterscheiden sich jedoch insofern, als dass jedes Stück nur durch seinen Refrain repräsentiert wird. Die relevanten Passagen innerhalb der Songs werden dabei durch eine zusätzlich durchzuführende Strukturanalyse bestimmt. Obwohl durch

diese Vorgehensweise eine erhebliche Datenreduktion stattfindet, kann vermutet werden, dass die Ergebnisse zumindest gleich gut wie bei Betrachtung des gesamten Stücks bleiben. Dies liegt daran, dass der Refrain durch seine oftmaligen Wiederholungen das dominante und somit repräsentativste Motiv innerhalb eines Musikstücks ist. Weiters können bessere Ergebnisse beim Erkennen von Disco-Versionen, bei denen die Strophen im Extremfall komplett durch reine Rhythmus-Passagen ersetzt werden, erwartet werden.

Ausgehend von diesen Features erfolgt die Entscheidung, ob zwei Songs unabhängig voneinander sind oder nicht. Dazu wird in der Literatur der, aus der Sprachverarbeitung bekannte *Dynamic Time Warp*-Algorithmus vorgeschlagen. Dieser erlaubt beliebige Tempovariationen innerhalb des Stücks und auch Einfügungen oder Auslassungen kompletter Abschnitte.

Als weiterer Beitrag wird in dieser Arbeit eine alternative Methode zur Klassifizierung vorgestellt und untersucht. Dabei wird unterstellt, dass aufgrund der gleichbleibenden Notierung der Interpretationsfreiraum bei Versionen desselben Musikstücks eingeschränkt ist. Die Tempoänderung zwischen dem Original und der Cover-Version wird als konstant angenommen. Als mathematische Grundlage zum Bestimmen der Ähnlichkeit zweier Stücke unter dieser Voraussetzung eines linearen Zeitzusammenhangs dienen Ähnlichkeitsmatrizen sowie die Hough-Transformation zum Finden des dominantesten Pfades durch diese Matrizen entlang einer Geraden.

1.3 Aufbau

Zu Beginn dieser Arbeit werden in Kapitel 2 einige wichtige Grundlagen behandelt. Zum einen wird dabei beschrieben, auf welche unterschiedliche Arten Versionen eines Musikstücks entstehen und wie sich diese auf das jeweilige Audio-Signal auswirken, was vor allem der Verdeutlichung der Aufgabenstellung dient. Zum anderen werden wichtige Grundlagen der Musik vorgestellt und deren Repräsentation in maschinenverarbeitbarer Form erläutert.

Die weitere Arbeit orientiert sich am eigentlichen Arbeitsablauf zur Versions-Erkennung. In Kapitel 3 wird der Schritt der Feature Extraktion beschrieben. Begonnen wird dabei mit allgemeinen Methoden der Spektralanalyse (Abschnitt 3.2), die als Resultat die dominierenden Frequenzen zu jedem Zeitpunkt in einem Musikstück liefern. Diese werden dann bezüglich ihrer Referenzfrequenz normiert um einheitliche Stimmungen zu erreichen (Abschnitt 3.3). Nach dieser allgemeinen Vorverarbeitung erfolgt die Berechnung der eigentlichen Features – des THPCPs (Abschnitt 3.4 und 3.5) sowie der

Melodie (Abschnitt 3.6). Den Abschluss bildet die Strukturanalyse, die der Extraktion der Refrain-Stellen innerhalb eines Songs dient (Abschnitt 3.7).

Kapitel 4 erklärt die verschiedenen untersuchten Klassifizierungsmethoden. Ausgehend von einer Beschreibung der Ähnlichkeitsmatrizen, die in beiden Fällen als Grundlagen dienen, werden der *Dynamic Time Warp* Algorithmus sowie die Methode des linearen Zeitzusammenhangs vorgestellt.

Den Abschluss dieser Arbeit bilden die in Kapitel 5 dargestellten Resultate der Vergleiche aller möglichen Kombinationen aus Feature und Klassifizierungsmethode anhand des experimentellen Prototyps sowie ein Resumé über die Arbeit.

2 Grundlagen

Dieses Kapitel beschäftigt sich mit Grundlagen die für die, in dieser Arbeit vorgestellten Methoden von Bedeutung sind. Abschnitt 2.1 geht dabei auf verschiedene Arten von Cover-Versionen ein und dient der Verdeutlichung der Aufgabenstellung. Weiters werden grundlegende Konzepte der Musik – wie Tonsysteme – vorgestellt und Querbezüge zu Messgrößen der Physik sowie zu Modellen, die der maschinellen Verarbeitung dienen, hergestellt.

2.1 Cover-Versionen

In großen Musik-Sammlungen kommt es häufig vor, dass ein Musiktitel in mehreren Versionen enthalten ist. Als Gründe für solche abweichende Versionen nennt Gómez [19] im Kontext von Popularmusik folgende.

Re-Mastered Tracks entstehen durch die analog-digital Wandlung und digitale Nachbearbeitungsschritte. Vor allem älteren Songs, die zu einer Zeit aufgenommen wurden, in der die Digitaltechnik noch weniger weit fortgeschritten war, werden häufig re-mastered, was zu leichten Unterschieden im Klangbild führt.

Karaoke Versionen werden häufig zusammen mit dem Original veröffentlicht. Instrumental-Spuren bleiben komplett erhalten. Der wesentliche Unterschied ist jedoch das Fehlen der Hauptgesangsstimme oder eine zusätzliche Instrumentalstimme, die diesen Part übernimmt.

Live Tracks sind Konzertmitschnitte. Sie unterscheiden sich vom Original vor allem durch die geänderten Aufnahmebedingungen und dadurch verursachtes Rauschen bzw. Hintergrundgeräusche vom Publikum. Während die Besetzung sowie die Stimmlage in der Regel unverändert bleiben, ändern sich teilweise Tempo und Struktur – etwa durch eingefügte Improvisationsteile.

Akustik Tracks sind meistens vom selben Interpreten wie das Original, jedoch werden andere Instrumente und damit verbunden manchmal auch eine andere Stimmlage verwendet. Der Extremfall sind hier Disco-Versionen bei denen Rhythmus-Spuren weit stärker betont sind und sehr oft auch reine Rhythmus-Teile in die Struktur eines Songs eingefügt werden.

Cover Versionen entstehen, wenn andere Künstler einen Musiktitel neu interpretieren. Die möglichen Veränderungen sind hier sehr vielfältig – angefangen vom Singen eines anderen Texts (evt. in einer anderen Sprache) bis hin zum Verändern von Stimmlage, Tempo und Struktur sind dabei kaum Grenzen gesetzt. Die Frage, die sich hier – vor allem in Hinblick auf Copyright Aspekte – stellt ist, bis zu welchem Grad an Veränderungen es sich bei einem Musikstück noch um eine Cover-Version handelt und ab wann man es als eigenständigen Song werten kann. Die Internetplattform *Second Hand Songs*[1] beschäftigt sich mit dem Finden von Cover-Versionen. Im Juli 2007 waren bereits über 14.000 Titel mit insgesamt knapp 42.500 Covers in der Datenbank erfasst.

Da im Kontext der maschinellen Verarbeitung Metadaten über die Entstehung eines Songs in den seltensten Fällen vorhanden sind, gilt es, sich auf das Audio-Signal selbst zu konzentrieren und dieses mit jenem des potenziellen Original zu vergleichen. Gómez [19] gibt dazu einen Überblick über die möglichen Veränderungen, die sich auf das Audio-Signal auswirken und die ein Algorithmus zur Cover-Versions Erkennung berücksichtigen muss.

Typ 0: Unterschiedliche Musikstücke Der Vollständigkeit halber sei auch die Möglichkeit aufgelistet, dass zwei Stücke unabhängig voneinander sind und keine signifikanten Gemeinsamkeiten aufweisen.

Typ I: Identische digitale Kopie Dies ist der Trivialfall in dem es sich bei den beiden verglichenen Audiosignalen um eine Kopie handelt, bei der sämtliche Werte übereinstimmen.

Typ II: Rauschen Dieser Fall tritt ein, wenn es sich zwar um die selbe analoge Aufnahme handelt, die digitalen Signale jedoch durch Unterschiede in der A/D-Wandlung oder der digitalen Nachbearbeitung verschiedene Rausch-Anteile beinhalten. Eine weitere Möglichkeit für das Auftreten dieses Typs ist das Verwenden verlustbehafteter Komprimierungsverfahren (Bsp.: mp3-Format) mit verschieden starken Komprimierungsraten oder auch schon einfaches Quantisierungsrauschen beim Speichern des Signals mit unterschiedlichen Bit-Raten.

[1] http://www.secondhandsongs.com

Typ III: Stimmen Bei diesem Typ bleibt das Stück an sich, mitsamt dem Tempo in dem es gespielt ist, erhalten. Unterschiede bestehen jedoch bei den verwendeten Instrumenten bzw. den Gesangstimmen. Beispiele hierfür sind das Ersetzen der Gesangstimme durch ein Instrument für eine Karaoke-Version bzw. der Einsatz einer Frau anstatt eines Mannes als Vokal-Stimme. Die Modifikationen schlagen sich also im Wesentlichen auf die Klangfarbe nieder.

Typ IV: Tempo Während die anderen Aspekte gleich bleiben, werden die beiden Songs in einem unterschiedlichen Tempo interpretiert.

Typ V: Gleiche Melodie Dieser Typ ist eine Kombination aus den vorherigen drei. Die zugrunde liegende Melodie bleibt zwar die selbe; Unterschiede können jedoch in der Stimmbelegung, dem Tempo und dem Rausch-Anteil bestehen. Dies ist in der Regel der Fall, wenn ein Stück in einem anderen Stil interpretiert wird.

Typ VI: Transposition Im Unterschied zu Typ V wird nun auch die Melodie verändert, indem sie in eine andere Tonart transponiert wird. Häufiger Grund dafür ist, dass sie bezüglich ihrer Stimmlage an einen anderen Sänger bzw. eine andere Gruppe von Instrumenten angepasst wird.

Typ VII: Struktur Dieser Typ tritt sehr häufig auf und berücksichtigt Veränderungen in der Liedform. Beispiele dafür sind zusätzliche Wiederholungen von Refrain bzw. Strophen oder das Einfügen eines neuen Instrumentalparts (Instrumental-Solos).

Der Mensch nimmt beim Hören von Musik die Melodie sehr stark wahr und erkennt diese trotz Einfluss der oben genannten Abweichungen relativ leicht wieder. Eine Studie von Peretz et al. [38] etwa zeigt, dass Menschen, denen eine bekannte Melodie vorgespielt wird, diese sofort mit dem Text verbinden. Sie reicht somit aus, um ein Musikstück quasi zu identifizieren. Im Rahmen der maschinellen Verarbeitung ist jedoch nicht die gefilterte Melodie sondern nur das gesamte Audiosignal verfügbar. Während sich die Typen II und III relativ schwach auf dieses auswirken, wird es von den Typen IV bis VII maßgeblich beeinflusst. Die Herausforderung liegt nun darin, Features zu finden, die der menschlichen Wahrnehmung ähnlich sind und so eine erwartungskonforme Klassifizierung ermöglichen.

2.2 Musikalische Grundlagen

Im Weiteren wird auf einige Grundlagen der Musik zurückgegriffen die hier erläutert werden sollen. Darunter fallen Skalen und Tonsysteme sowie der Zusammenhang zwischen Obertönen, Mehrklängen und Harmonien.

2.2.1 Skalen

Da es sich bei Audiosignalen um Schwingungen und somit physikalische Phänomene handelt, kommen zu deren Beschreibung in erster Linie auch Messgrößen und Einheiten der Physik zum Einsatz. Von besonderer Bedeutung sind dabei die Frequenz $[Hz]$ – ein Maß für die Höhe des jeweiligen Tons – sowie die Amplitude $[dB]$ – ein Maß für die Lautstärke.

In der Musik wird die Tonhöhe nicht direkt in Hertz gemessen, sondern anhand einer logarithmischen Skala. Erhöht man einen Ton um ein bestimmtes Intervall, so kommt dies einer Multiplikation der Frequenz mit dem entsprechenden Faktor gleich. Der zweite grundlegende Unterschied der in der Musik verwendeten Skala im Gegensatz zur physikalischen ist jener, dass sie diskret ist und nur bestimmte Werte (Töne) zulässt, die schriftlich durch Noten aufgezeichnet werden. Der minimale Abstand zwischen zwei Tönen entspricht einem Halbton bzw. einem Frequenzverhältnis von $1 : 2^{\frac{1}{12}}$. Da die musikalische Skala keinen absoluten Nullpunkt kennt, muss sie zudem anhand einer Normierungskonstante an der physikalischen ausgerichtet werden. Diese Konstante ist laut ISO 16:1075[2] mit genau 440Hz festgelegt die dem Kammerton a' entsprechen.

Die Beschränkung auf Halbtonschritte ist sehr restriktiv. Daher hat sich zur Bestimmung von musikalischen Intervallen zusätzlich die von Alexander John Ellis vorgeschlagene Einheit *cent* etabliert. Hundert cent entsprechen dabei einem Halbton – ein Intervall von einem cent somit einem Frequenzverhältnis von $1 : 2^{\frac{1}{1200}}$

Neben den Skalen der Physik und der Musik existieren noch weitere, die sich an der psychoakustischen Wahrnehmung des Menschen orientieren. Grundlage dafür sind zwei Phänomene.

Tonhöhenwahrnehmung Die menschliche Wahrnehmung der Tonhöhe entspricht weder exakt der linearen noch der logarithmischen Frequenzskala der Musik, sondern einer Mischform. Im unteren Frequenzbereich (bis etwa 500Hz) besteht ein linearer Zusammenhang, der dann in einen logarithmischen übergeht.

[2] http://www.iso.org

Kritische Bänder Im menschlichen Ohr ist die Cochlea (Gehörschnecke) verantwortlich für die Umwandlung von Schallwellen in Nervenimpulse. Die Basilarmembran, die dabei auf unterschiedliche Frequenzen reagiert, ist in 24 Sektoren unterteilt von denen jeder die Auswertung aller Frequenzen eines bestimmten Bands übernimmt.

Die daraus resultierende Skala ist die *Bark-Skala* (nach Heinrich Barkhausen). Ein Bark entspricht dabei einem kritischen Band und ist definiert durch

$$z = 13 \arctan(0.00076 f) + 3.5 \arctan\left(\left(\frac{f}{7500}\right)^2\right) \tag{2.1}$$

was einer Normierung von $1,31\text{Bark} \equiv 131\text{Hz} \equiv$ Ton c entspricht.

Auf den selben Prinzipien wie die Bark-Skala basiert die *Mel-Skala*, die durch

$$m = 1127 \ln\left(1 + \frac{f}{700}\right) \tag{2.2}$$

gegeben ist. Abbildung 2.1 (aus Wikipedia[3]) verdeutlicht den Zusammenhang von Frequenz- Bark- und Mel-Skala sowie der Ursache der kritischen Bänder – der Wahrnehmung durch die Basilarmembran im menschlichen Ohr.

Ähnlich wie für die Tonhöhe existieren auch für die absolute und empfundene Lautstärke eines Tons unterschiedliche Skalen. Da diese im aktuellen Kontext jedoch eine untergeordnete Rolle spielen wird auf eine Darstellung verzichtet.

2.2.2 Tonsysteme

Jedes Musikstück wird in einer bestimmten Tonart gespielt bzw. gesungen. Diese ist bei Coverversionen oft Gegenstand von Variationen, um das Stück einem bestimmten Sänger bzw. einer Gruppe von Instrumenten anzupassen. Das Transponieren in eine andere Tonart verändert zwar alle absoluten Tonhöhen, dennoch erkennen Menschen die Melodien als gleich. Soll diese Erkennung nun von einem Computerprogramm durchgeführt werden, so muss der Algorithmus ähnlich robust gegenüber Transpositionen sein wie die menschliche Wahrnehmung.

[3] http://de.wikipedia.org

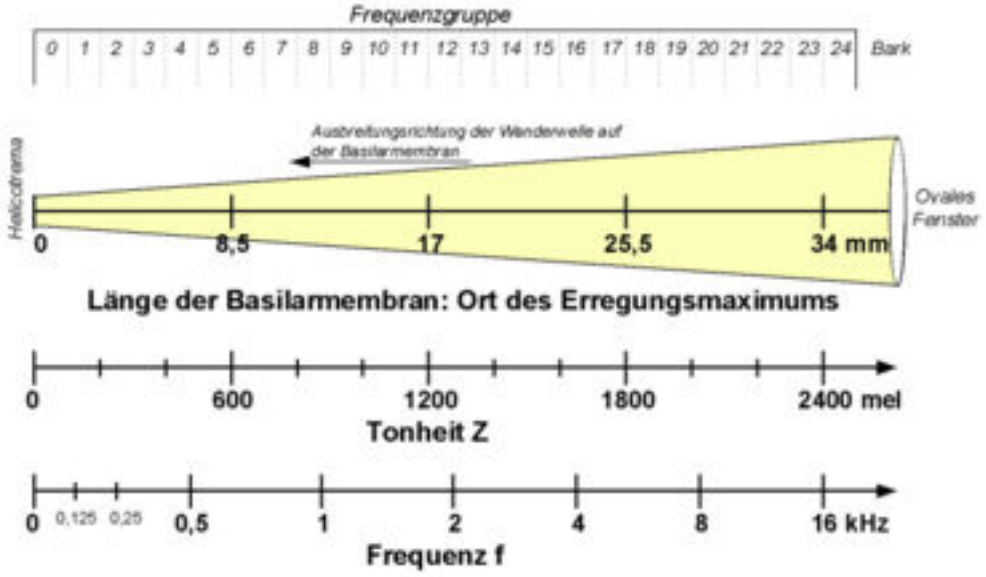


Abbildung 2.1: Zusammenhang zwischen Bark-, Mel- und Frequenz-Skala

Eine Möglichkeit wäre, nicht absolute Tonhöhen zu betrachten, sondern nur relative Veränderungen im Vergleich zu den vorhergehenden Noten. Dies würde bei einstimmigen Melodien relativ gut funktionieren, versagt jedoch bei polyphonen Stücken und erst recht bei der Analyse gesamter Spektren.

Eine Alternative, die diese Schwächen umgeht, ist die Tonart eines Stücks automatisch zu bestimmen und das Musikstück dann in eine (beliebige, aber fixe) Standard-Tonart zu transponieren. Um dies zu ermöglichen, muss ein Konzept für Tonarten in maschinenverarbeitbarer Form vorliegen. Im Folgenden wird nun eine Reihe an Modellen vorgestellt, die als Grundlage dafür dienen.

In westlicher Musik dominieren zur Zeit die beiden Tonsysteme Dur und Moll, die sich beide aus den Kirchentonarten ableiten. Sie bestehen jeweils aus sieben Stufen – die achte Stufe ist wieder der – um eine Oktave erhöhte – Grundton. Bei sieben Stufen mit einem Umfang von insgesamt zwölf Halbtönen können die Intervalle zwischen den einzelnen Stufen nicht äquidistant sein. Daraus folgt, dass zwei dieser Intervalle jeweils nur aus einem Halbton bestehen, während die anderen einen Ganzton betragen. Der Unterschied zwischen den Tonsystemen ist nun die Lage dieser Halbtonschritte. Beiden gemeinsam ist jedoch die große Bedeutung der ersten Stufe (Tonika) sowie der fünften (Dominante) und dahinter folgend der vierten (Subdominante).

Betrachtet man die musikalischen Noten ohne Vorzeichen, so gibt es wieder sieben Noten (. . . -c-d-e-f-g-a-h-c-. . .), die zwölf Halbtönen entsprechen. Es muss also zwei *natürliche* Halbtonschritte geben. Diese liegen zwischen e und f sowie zwischen h und c. Bildet man eine Tonreihe beginnend mit einem c, so entspricht dies der Dur-Tonleiter mit den Halbtonschritten zwischen 3. und 4. Stufe sowie zwischen 7. und 8. (= 1.) Stufe – beginnt man mit a, so erhält man eine Moll-Tonleiter mit den Halbtonschritten zwischen 2. und 3. sowie 5. und 6. Stufe. Wählt man andere Grundtöne, so sind Vorzeichen zu setzten um die Halbtonschritte von ihren natürlichen Lagen weg an die erforderlichen Positionen des jeweiligen Tonsystems zu verschieben. Das folgende Notenbeispiel zeigt das C-Dur sowie das a-Moll System, bei denen die Halbtonschritte in ihren natürlichen Lagen bereits an den richtigen Positionen liegen.

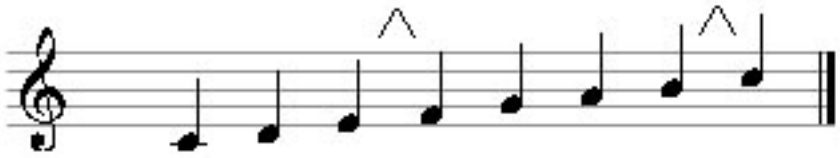



Abbildung 2.2: C Dur-Tonleiter

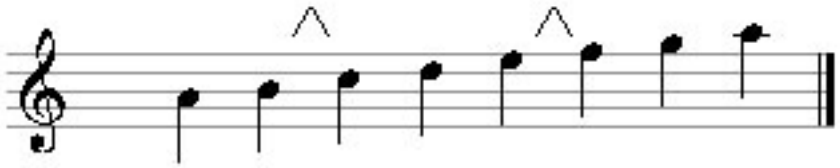



Abbildung 2.3: a Moll-Tonleiter

Das einfachste Modell für die beiden Tonsysteme ist also jenes das ihre leitereigenen Töne in gleichem Maße berücksichtigt. Die Abbildungen 2.4 und 2.5 stellen diese für die Dur- sowie die Moll-Tonleiter dar.

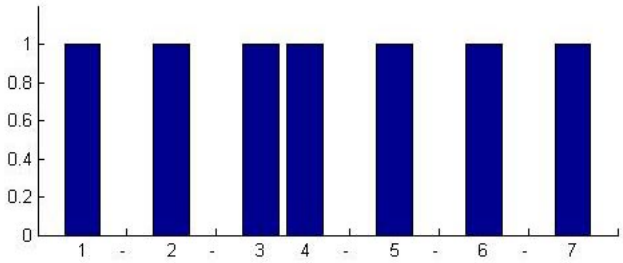


Abbildung 2.4: Flaches Modell leitereigener Töne für Dur

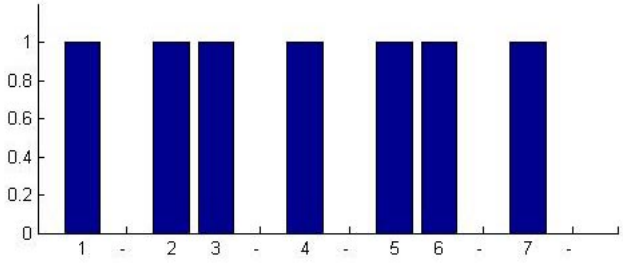


Abbildung 2.5: Flaches Modell leitereigener Töne für natürliches Moll

Ein häufiges Muster in der Musik ist, dass man sich in Melodien gerne auf den Grundton zubewegt. In Dur-Tonleitern ist dabei die siebte Stufe, die nur einen Halbton unter dem Grundton liegt, von besonderer Bedeutung. Um diesen so genannten Leitton auch in Moll-Tonarten zu verwenden, wird hier oft die siebte Stufe generell um einen Halbton erhöht. Diese neue Skala, bei der nun ein dritter Halbtonschritt und ausgleichend dazu an anderer Stelle ein Schritt von drei Halbtönen entstanden ist, wird auch *harmonisches Moll* genannt. Das entsprechende Modell ist in Abbildung 2.6 dargestellt.

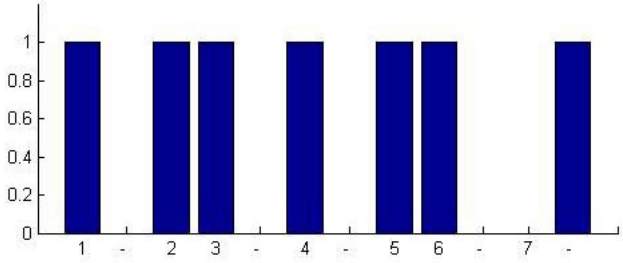


Abbildung 2.6: Flaches Modell leitereigener Töne für harmonisches Moll

Um diesen Drei-Halbtonschritt – der unschön klingt und auch schwer zu singen ist – zu vermeiden, existiert eine dritte weit verbreitete Art der Moll-Tonleiter – das *melodische Moll*. Dabei wird vom harmonischen Moll ausgehend zusätzlich zur siebten auch noch die sechste Stufe erhöht. So wird das übergroße Intervall umgangen und der Halbtonschritt, der eigentlich zwischen 5. und 6. Stufe liegen sollte, zwischen die Stufen 7 und 8 (= 1) geschoben, sodass der Leitton erhalten bleibt und trotzdem keine unschönen Melodien entsteht. Dieser Leitton macht jedoch nur im Kontext von aufsteigenden Melodien Sinn. Bei absteigenden Tonfolgen wird somit auf alle Änderungen verzichtet und ein natürliches Moll verwendet (Abbildung 2.7).

Dieses einfache Modell einer Tonart geht auf Temperley [45] zurück und wird *flat key profile* genannt.

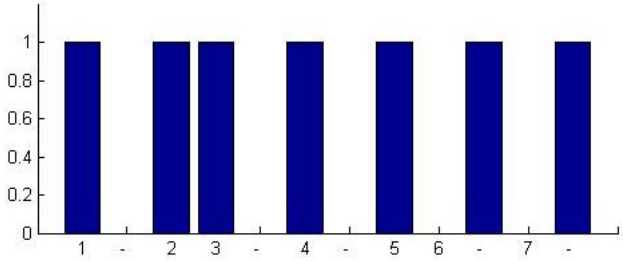


Abbildung 2.7: Flaches Modell leitereigener Töne für melodisches Moll (aufsteigend)

Ein komplexeres Modell wird von Krumhansl [24] vorgeschlagen. Dieses basiert auf der menschlichen Wahrnehmung. Bei seiner Konstruktion wird einer Reihe von Versuchspersonen eine Skala sowohl in aufsteigender als auch in absteigender Reihenfolge vorgespielt. Danach folgt einer der zwölf Töne der chromatischen Skala und für die Probanden gilt es zu bewerten, wie gut der Ton in die zu gehörte Skala passt. Die Ergebnisse stimmen mit der generellen Wissensbasis der Musiktheorie überein. Die beste Übereinstimmung wird beim Grundton erzielt, gefolgt von den beiden anderen Tönen, die den Dreiklang auf Basis des Grundtons bilden (3. und 5. Stufe). Danach kommen die übrigen leitereigenen Töne und – wie erwartet – erzielen die leiterfremden Töne die schlechteste Übereinstimmung. Beim Modell handelt es sich um ein *weighted key profile*. Es ist in den Abbildungen 2.8 für Dur und 2.9 für Moll dargestellt.

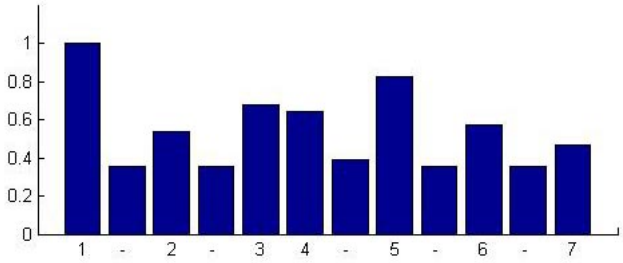


Abbildung 2.8: Empirisches Modell nach Krumhansl ([24]) für Dur

Ein weiteres Modell, das abermals von Temperley [45] stammt, basiert auf jenem von Krumhansl [24]. Die wesentlichste Modifikation ist die stärkere Berücksichtigung der 7. Stufe in Dur und harmonischem sowie melodischem Moll (Abb. 2.10 und 2.11).

Einen komplett anderen Weg geht Chai [9]. Im Unterschied zu den zuvor vorgestellten Modellen ist ihres anhand der Notationen zahlreicher Musikstücke selbst und nicht aufgrund menschlicher Wahrnehmung bestimmt. In jedem Stück wird die Anzahl der Noten aller Stufen gezählt. Der Durchschnitt der Gesamtheit der betrachteten Musik-

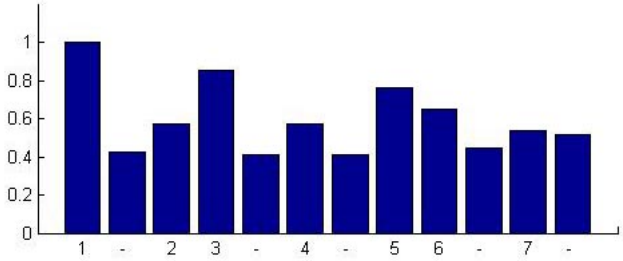


Abbildung 2.9: Empirisches Modell nach Krumhansl ([24]) für Moll

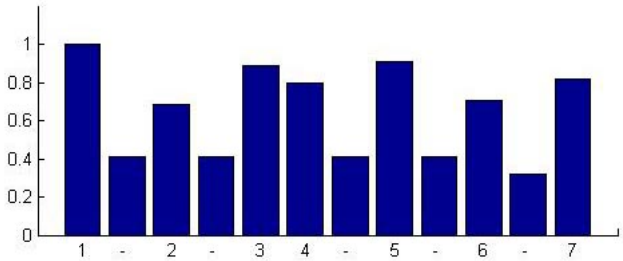


Abbildung 2.10: Empirisches Modell nach Temperley ([45]) für Dur

stücke ergibt dann das errechnete Profil. Dieses berücksichtigt jedoch das Tongeschlecht nicht.

Hat man jedoch mittels des allgemeinen Profils die Lage der Halbtonschritte gefunden, (Bsp.: Halbtöne zwischen e und f sowie h und c. Daraus folgt: C-Dur oder a-Moll möglich) so lässt sich das Tongeschlecht daran bestimmen, ob die 5. Stufe (Dur) oder die 6. Stufe (Moll) dominanter ist (Abb. 2.12 und 2.13).

Einen abermals anderen Weg als bei allen anderen vorgestellten geht Gómez [18] mit einem flachen Modell, das – so wie in den Abbildungen 2.14 und 2.15 gezeigt – nur den Dreiklang auf der ersten Stufe berücksichtigt. Trotz der Einfachheit des Modells erzielt es bei der Anwendung zum Bestimmen der Tonart eines Stück sehr gute Ergebnisse – vor allem bei den Genres Pop und Rock aber auch Jazz und Blues. Dies ist wichtig, da die anderen Modelle primär auf klassische Musik ausgelegt sind und bei anderen Genres bei der Anwendung zur Tonart-Bestimmung verhältnismäßig schlecht abschneiden.

Neben diesen Modellen existieren noch viele weitere, die komplexere Zusammenhänge beim Erkennen von Tonarten berücksichtigen. Darunter fallen etwa die zeitliche Abfolge einzelner Tonklassen oder bestimmte Muster von zusammen klingenden Tönen bzw. Intervallen (vgl. [27], [14], [47], [46], [25]). Ebenfalls interessant ist das auf einem Neuronalen Netz aufbauende Modell aus [4] sowie die Regel- bzw. Entscheidungsbaum-basierten Systeme aus [26] oder [31].

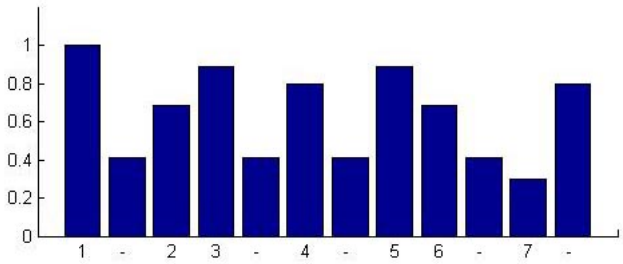


Abbildung 2.11: Empirisches Modell nach Temperley ([45]) für Moll

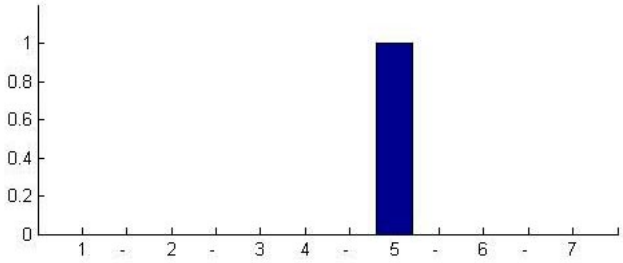


Abbildung 2.12: Modell nach Chai ([9]) für Dur

2.2.3 Obertöne, Mehrklänge und Harmonien

Eine weitere Eigenheit des menschlichen Gehörs – neben der vorwiegend logarithmischen Tonhöhenwahrnehmung – ist, dass Frequenzen in bestimmten, sich wiederholenden Zyklen sehr ähnlich wahrgenommen werden. Dabei stehen die korrespondierenden Frequenzen immer im Verhältnis einer Zweierpotenz zueinander. In der Musik entspricht dieses Verhältnis dem Intervall einer Oktave, die jeweils in zwölf Halbtöne gegliedert ist. Der Tonhöhenunterschied zwischen zwei Frequenzen f_0 und f_1 gemessen in Halbtönen errechnet sich daher als

$$n = 12 \cdot \log_2 \left(\frac{f_1}{f_0} \right) \tag{2.3}$$

Wird ein Ton gespielt oder gesungen, so erklingt jedoch nicht nur dieser mit seiner Grundfrequenz f_0, sondern auch die seiner Obertonreihe. Das sind jene Frequenzen, die einem ganzzahligen Vielfachen von f_0 entsprechen. Am Beispiel vom Ton c wird in Tabelle 2.1 die Obertonreihe bis zur zehnten mitklingenden Komponente dargestellt. Die angegebenen Töne sind dabei zumeist nicht exakt sondern, stellen nur eine Näherung dar um die Einordnung zu erleichtern.

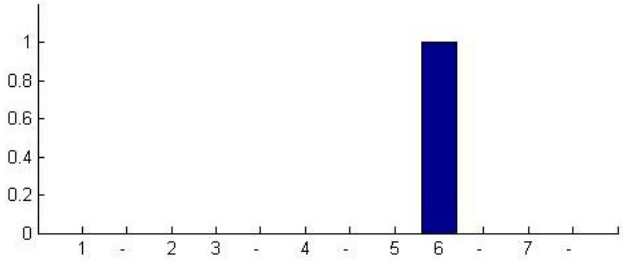


Abbildung 2.13: Modell nach Chai ([9]) für Moll

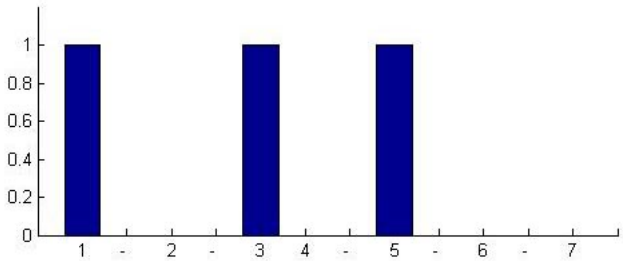


Abbildung 2.14: Dreiklangs-Modell nach Gómez ([18]) für Dur

Die Stärke mit der einzelne Komponenten im gesamten Klangbild enthalten sind sowie deren zeitliche Verteilung hängen in erster Linie vom Instrument ab und bestimmen so die Klangfarbe.

Von Mehrklängen spricht man, wenn nicht nur mehrere Obertöne zusammen klingen, sondern mehrere musikalische Noten gleichzeitig gespielt werden – also mehrere Grundfrequenzen simultan erklingen. Besondere Bedeutung in der Musik haben dabei Dreiklänge, die aus einem Grundton und den zwei jeweils im Intervall einer (großen bzw.

Oberton	**f**	**Frequenz**	**Ton (näherungsweise)**
0	f_0	130,8Hz	c
1	$2 \cdot f_0$	261,6Hz	c’
2	$3 \cdot f_0$	392,4Hz	g’
3	$4 \cdot f_0$	523,3Hz	c”
4	$5 \cdot f_0$	654,1Hz	e”
5	$6 \cdot f_0$	784,9Hz	g”
6	$7 \cdot f_0$	915,7Hz	h”
7	$8 \cdot f_0$	1046,5Hz	c”’
8	$9 \cdot f_0$	1177,3Hz	d”’
9	$10 \cdot f_0$	1308,1Hz	e”’

Tabelle 2.1: Obertöne von c

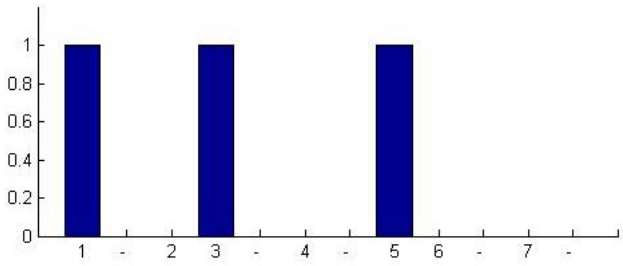


Abbildung 2.15: Dreiklangs-Modell nach Gómez ([18]) für Moll

kleinen) Terz folgenden Tönen besteht. Wird über die Zeit eine Folge solcher Mehrklänge gespielt, bezeichnet man dies als Harmonie.

3 Feature Extraktion

Die Feature Extraktion ist ein elementarer Schritt aller Klassifizierungsaufgaben. Es geht dabei darum, anhand der gegeben Eingangsdaten bzw. der daraus ableitbaren oder messbaren Größen Metadaten (*Features*) zu extrahieren, die die Zuordnung zu einzelnen Klassen erlauben. Erstrebenswert sind daher solche Merkmale, die bei zwei Instanzen der selben Klasse möglichst ähnlich und bei zwei Instanzen unterschiedlicher Klassen möglichst verschieden sind.

Im Kontext von (Cover-)Versions-Erkennung beschreibt eine Klasse ein bestimmtes Musikstück. Instanzen sind demnach verschiedene Versionen eines gleichen Stücks. Da sich die Anzahl voneinander unabhängiger Songs nach oben hin nicht eingrenzen lässt, auf jeden Fall aber sehr groß ist, sind einfach gehaltene Features nicht zweckmäßig, da sie nicht ausreichend zwischen derart vielen Klassen diskriminieren können. Die hier vorgestellten und verglichenen Merkmale sind einerseits die Melodie – ein Vektor mit einer Dimensionalität proportional zur Länge des Musikstücks – und andererseits das *Transposed Harmonic Pitch Class Profile* (THPCP) nach Gómez [18]. Eine *Pitch-Class* ist dabei jeweils einer der zwölf Halbtöne (evt. auch noch feiner: Vierteltöne,...) der chromatischen Skala wobei die genaue Oktavenlage vernachlässigt wird. Das entsprechende Profil ist mit einem Histogramm vergleichbar und misst, wie stark jeder der zwölf Halbtöne zu einem Zeitpunkt präsent ist. Das auf diesem Prinzip aufbauende THPCP-Feature berücksichtigt zudem vorhandene Obertöne und ist robust gegenüber Transpositionen. Zusätzlich zu den Varianten, in denen die Features jeweils in Bezug auf das gesamte Musikstück berechnet werden, untersucht diese Arbeit auch jene Versionen, die sich nur auf den Refrain des Stück beziehen.

Dieses Kapitel widmet sich nun den Arbeitsschritten, die notwendig sind, um die beiden genannten Features aus einem Audiosignal zu extrahieren. Effekte, die durch Stereo oder Raumklang hervortreten, werden dabei vernachlässigt. Als Eingangsdaten sind immer Audiosignale in Mono – jedoch mit beliebigen aber gleichen Sample- und Bit-Raten – angenommen. Davon ausgehend ist der Arbeitsablauf zur Extraktion der Features in Abbildung 3.1 dargestellt. An diesem Ablauf orientiert sich auch der Aufbau des vorliegenden Kapitels.

3.1 Arbeitsablauf

Ausgehend von einem Audio-Signal wird im Schritt der *Spektralanalyse* das Spektrum des Musikstücks berechnet und aufbereitet. Die Aufbereitungsschritte sind dabei die Übergangserkennung, die mögliches Rauschen vermindern soll, die Peak-Erkennung, die zu jedem Zeitpunkt die dominierenden Frequenzen herausfiltert, und die Maskierung, die all jene der verbliebenen Frequenzen entfernt, die vom Menschen aufgrund von Überdeckung mit lauteren Tönen ohnehin nicht wahrgenommen werden können. Resultat dieses Schritts ist eine Liste von dominanten Frequenzen zu jedem Zeitpunkt im Song.

In einem zweiten allgemeinen Schritt werden diese Frequenzen auf einen einheitlichen Stimmton normiert. Dieser ist zwar laut ISO 16:1075[1] als Kammerton a' mit 440 Hertz festgelegt; was aber nicht bedeutet, dass Orchester bzw. Bands auch tatsächlich ihre Instrumente auf diese Höhe stimmen. Ergebnis dieses Schritts sind somit normierte Frequenzen, die in musikalische Noten umgerechnet werden können.

Ausgehend von den nun vorliegenden Daten werden die beiden angesprochenen Features unabhängig voneinander berechnet. Zu dem harmonie-basierten Feature werden zuerst als Vorstufe einfache Chroma-Vektoren berechnet (HPCP), die noch nicht gegenüber Modulationen robust sind. Erst anhand eines weiteren Schritts – der Tonart-Bestimmung – kann das tatsächliche THPCP-Feature abgeleitet werden.

Bei der Berechnung des zweiten Features durch die Melodieerkennung ist ein solcher Schritt zur Tonart-Erkennung nicht notwendig, da Melodien nicht nur absolut, sondern auch anhand der Intervalle zwischen ihren einzelnen Tönen beschrieben werden können.

Neben der Eignung der so berechneten fertigen Features, soll in dieser Arbeit auch untersucht werden, ob die Beschränkung auf den Refrain einen Gewinn bringt. Unabhängig von der Art des Features wird somit in einem letzten Schritt die zeitliche Lage der Refrain-Passagen ermittelt. Mit dieser Information lassen sich dann die relevanten Abschnitte innerhalb der fertigen Features extrahieren. Resultat sind somit zwei weitere Features, die (repräsentativen) Ausschnitten der zuvor vorgestellten entsprechen.

[1] http://www.iso.org

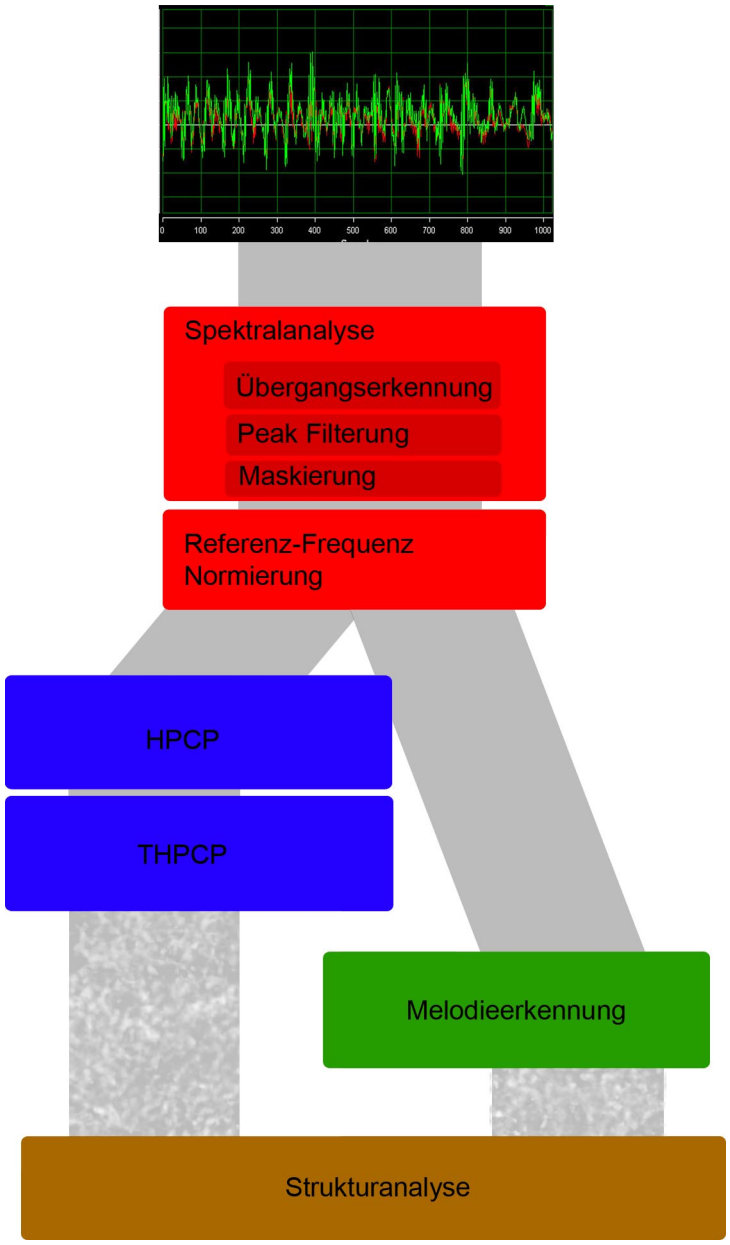


Abbildung 3.1: Arbeitsablauf zur Extraktion der Features

3.2 Spektralanalyse

Als Basis für alle weiteren Analyseschritte dient die Transformation des Audio-Signals vom Zeitbereich in den Frequenzbereich. Während im ersten die Amplitude des Signals als Funktion der Zeit dargestellt wird, ist das Signal im Frequenzbereich als Überlagerung einzelner Sinus- bzw. Cosinus-Wellen repräsentiert – die Amplitude also als Funktion der Frequenz. Dieser Abschnitt beschreibt in Anlehnung an [32], [33] und [44] die wichtigsten Grundlagen der Signalverarbeitung im Audiobereich.

3.2.1 Diskrete Fourier Transformation

Mathematische Grundlage für die Überführung ist die (Diskrete) Fourier Transformation (DFT). Sie ist eine lineare Abbildung eines Vektorraums in einen anderen, wobei beide durch Orthonormal-Basen beschrieben sind.

Sei $x(n)$ ein (periodisches) diskretes Signal im Zeitbereich, bestehend aus N Samples, so wird das Frequenzspektrum $X(k)$ (mit $0 \leq k \leq N-1$) durch die DFT wie folgt berechnet:

$$X(k) = \sum_{n=0}^{N-1} e^{-jk\frac{2\pi n}{N}} x(n) \tag{3.1}$$

Die einzelnen Koeffizienten k entsprechen den jeweiligen Frequenzen, die von 0Hz bis zur Sampling-Frequenz (f_s) in äquidistanten Abständen von $\Delta f = f_s/N$ verteilt liegen. Die Werte $X(k)$ selbst sind dabei komplex. Aus ihnen können mithilfe der nachfolgenden Formeln der Betrag (bzw. die Amplitude) sowie die Phase jedes Frequenzanteils berechnet werden.

$$A(k) = \mid X(k) \mid = \sqrt{X_r(k)^2 + X_i(k)^2} \tag{3.2}$$

$$\phi(k) = \arctan\left(\frac{X_i(k)}{X_r(k)}\right) \tag{3.3}$$

Es ist jedoch nicht möglich, aus den N reellen Eingangswerten des Signals $x(n)$, doppelt so viele – nämlich N komplexe und somit $2N$ reelle – unabhängige Werte zu erhalten. Der Grund für dies Anzahl an Werten ist, dass die Koeffizienten 0 bis $\lceil N/2 \rceil$ die Frequenzen von 0Hz bis zur Nyquist-Frequenz ($f_s/2$) repräsentieren. Alle Frequenzen dar-

über sind laut Abtasttheorem mit den periodisch fortgeführten Amplituden enthalten - die Koeffizienten entsprechen somit den komplex Konjugierten jener des unteren Halbbandes. Sie enthalten keine zusätzliche Information und können daher vernachlässigt werden. Dass dies durchaus Sinn macht, wird auch klar, wenn man sich das Abtasttheorem an sich vor Augen hält. Es besagt, dass man eine Abtastrate von mindestens f_s benötigt, um Frequenzen bis zu $f_s/2$ darstellen zu können. Somit ist klar, dass alle im Audio-Signal vorhandenen Frequenzen im unteren Halbband liegen müssen. Von diesem kann somit Betrags- sowie Phasenspektrum berechnet werden.

3.2.2 Fast Fourier Transformation

Aufgrund der großen Bedeutung der Diskreten Fourier Transformation in der Signalverarbeitung stellt sich die Frage nach einem effizienten Algorithmus zur computergestützten Berechnung. Prinzipiell stellt eine Basistransformation – wie die Fourier Transformation eine ist – eine Multiplikation eines Eingangsvektors der Länge N (hier das Signal) mit einer Transformations-Matrix der Größe NxN dar. Eine derartige Berechnung hat eine asymptotische Komplexität von $O(N^2)$.

Um die Komplexität zu reduzieren wird bei der so genannten *Fast Fourier Transformation* (FFT) das Divide&Conquer-Prinzip angewandt. Anstatt die Transformation auf das gesamte Signal anzuwenden, wird jeweils nur die Hälfte der Eingangswerte betrachtet – einmal jene mit geraden und einmal jene mit ungeraden Indizes. Diese beiden wesentlich einfacher zu berechnenden Teil-DFTs werden im Anschluss wieder zusammengefügt. Dass sie nur eine Länge von $N/2$ Ausgangswerten besitzen, während das Gesamtergebnis doppelt so lang ist, stellt kein Problem dar, da die Koeffizienten im Frequenzbereich periodisch fortgeschrieben werden können. Bei rekursiver Fortführung dieses Verfahrens – das auch als Dezimation im Zeitbereich bezeichnet wird – lässt sich die Komplexität auf $O(N \log N)$ reduzieren.

Weitere Performanzgewinne lassen sich durch eine iterative Implementierung des Algorithmus sowie geeignete Speicherverwaltung erzielen. Hintergedanke der *In Place FFT* ist es, die Daten so im Speicher zu halten, dass bei der Verschmelzung der Teilergebnisse keine Datenumschichtungen notwendig sind. Die dazu notwendige Reihenfolge der einzelnen Teilergebnisse im Speicher lässt sich sehr einfach durch Umkehr der Bits der eigentlichen Indizes ermitteln (vgl. [32], [33], [44]).

3.2.3 Fensterung

Die Fourier Transformation – wie sie oben beschrieben ist – dient dazu ein Signal in seiner Gesamtheit zu analysieren. Dies ist wenig informativ – schließlich möchte man nicht wissen welche Frequenzen generell in einem Musikstück vorkommen, sondern wann und wie lange sie gespielt werden. Um diese Fragen beantworten zu können, genügt es, das Signal in viele sehr kurze Abschnitte (*Fenster* oder *Frames*) zu teilen und diese dann für sich zu analysieren. Dieses Vorgehen heißt Short Time Fourier Transformation (STFT).

Das Aufteilen des Signals geschieht durch Multiplikation mit einer Fensterfunktion $w(n)$ die überall 0 ist, außer im Bereich des Fensters. Dieses wird schrittweise verschoben, um das ganze Signal abzudecken. Formel 3.5 beschreibt die Berechnung des Spektrums eines Frames, wobei $x(n)$ das Eingangssignal und λ die Nummer des Abtastwerts in der Mitte des Fensters ist.

$$x_{STFT}(\lambda, n) = x(n)w(n-\lambda) \tag{3.4}$$

$$X_{STFT}(\lambda, w) = \sum_{n} x(n)w(n-\lambda)e^{-j\omega n} \tag{3.5}$$

Die einfachste mögliche Fensterfunktion $w(n)$ ist die Rechtecksfunktion wie in Formel 3.6 dargestellt.

$$R(t) = \begin{cases} 1 & : \text{ wenn } t \epsilon Fenster \\ 0 & : \text{ sonst} \end{cases} \tag{3.6}$$

Ihre Vorteile liegen einerseits in der Einfachheit und andererseits in der Exaktheit im Zeitbereich. Die DFT geht allerdings von periodischen bzw. periodisch fortsetzbaren Signalen aus. Schneidet man das Signal jedoch an den Fensterrändern hart ab, so ist der letzte Abtastwert mit ziemlicher Sicherheit nicht wieder mit dem ersten fortsetzbar, ohne dass ein Sprung entsteht. Durch diese Unstetigkeit entstehen neue Frequenzen im Spektrum, die das Ergebnis verfälschen. Es ist also intuitiv besser, die Werte an den Rändern des Fensters sanft gegen Null gehen zu lassen und so die kritischen Stellen quasi auszublenden.

Genau zeigen lässt sich diese Schwäche, wenn man die Fensterfunktion selbst – wie in Abbildung 3.2 dargestellt – in den Frequenzbereich transformiert. Aus der Multiplika-

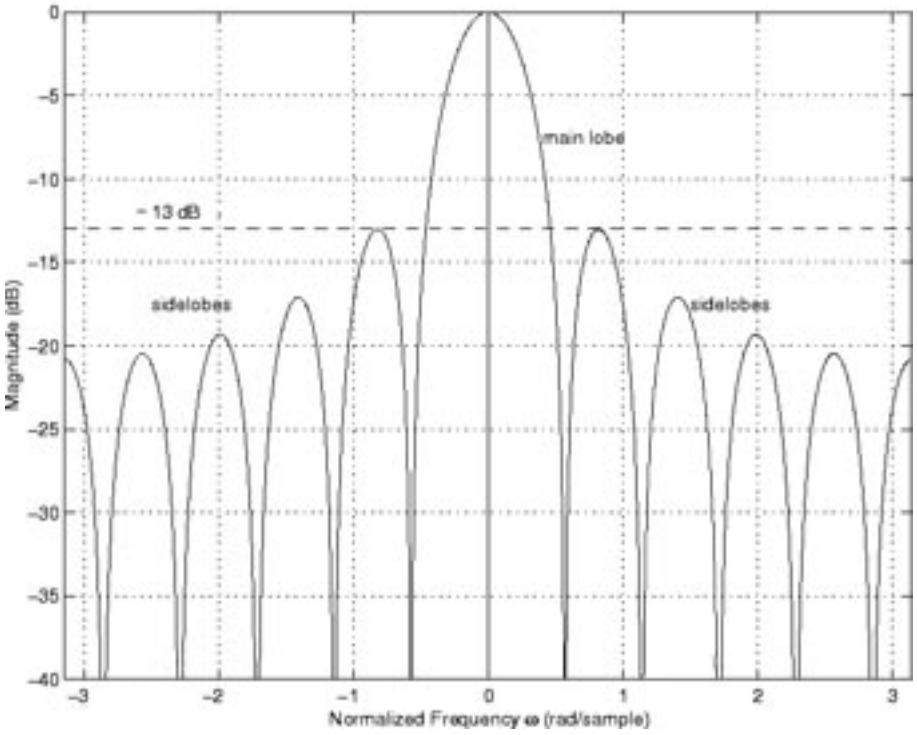


Abbildung 3.2: DFT des Rechteckfensters

tion mit dem Eingangssignal im Zeitbereich (siehe Formeln 3.4 und 3.5) wird dann eine Faltung mit dem Spektrum des Signals.

$$X_{STFT}(k) = X(k) * W(k) \tag{3.7}$$

Je stärker $W(k)$ somit vom idealen Impuls abweicht, desto mehr wird auch das Ergebnis verfälscht. Dabei wird zwischen zwei Arten von Fehlern unterschieden. Einerseits entsteht eine Streuung – der Betrag der Hauptfrequenz beeinflusst die berechneten Stärken anderer Frequenzen. Dieser Effekt wird durch die zahlreichen Neben-Peaks im Spektogramm der Fensterfunktion hervorgerufen und ist umso größer je weniger die Amplituden der Neben-Peaks von jener des Haupt-Peaks abweichen. Andererseits ist die Frequenzauflösung fehlerbehaftet. Sie ist umso besser je schmäler der Haupt-Peak ist. Bei sehr breiten Peaks nimmt sie entsprechend ab.

Tabelle 3.1 listet eine Auswahl an bekannten Fensterfunktionen auf. Die Spalte b gibt dabei die Breite des Haupt-Peaks im Frequenzbereich an. Die Einheit ist die Anzahl der Frequenzbänder. *SNR* steht für das Verhältnis des stärksten Neben-Peaks zum Haupt-Peak. Beim Gauss- und dem Kaiser-Fenster sind die Werte nicht absolut, sondern hängen von den gewählten Parametern ab. Die Abbildung 3.3 verdeutlicht zudem

Name	b	SNR	Definition im Fensterbereich
Rechteckfenster	2	-13dB	$w(n) = 1$
Hanning	4	-23dB	$w(n) = 0.5 - 0.5\cos\left(\frac{2n\pi}{N-1}\right)$
Hamming	4	-43dB	$w(n) = 0.53836 - 0.46164\cos\left(\frac{2n\pi}{N-1}\right)$
BlackmanHarris 62	4	-62dB	$w(n) = \frac{1}{N}\left(0.45 + 0.49\cos\left(\frac{2n\pi}{N}\right) + 0.06\cos\left(\frac{4n\pi}{N}\right)\right)$
Blackman	6	-59dB	$w(n) = \frac{1}{N}\left(0.42 + 0.5\cos\left(\frac{2n\pi}{N}\right) + 0.08\cos\left(\frac{4n\pi}{N}\right)\right)$
Gauss	-	-	$w(n) = e^{-\frac{1}{2}\left(\frac{n-(N-1)/2}{\sigma(N-1)/2}\right)^2}$
Kaiser	-	-	$w(n) = I_0\left(\pi\alpha\sqrt{1-\left(\frac{2n}{N-1}-1\right)^2}\right)/I_0(\pi\alpha)$

Tabelle 3.1: Bekannte Fensterfunktionen

den Unterschied einzelner Fensterarten im Frequenzbereich. In dieser Arbeit wird im Folgenden in Anlehnung an [18] von der Blackman-Harris 62dB Funktion ausgegangen.

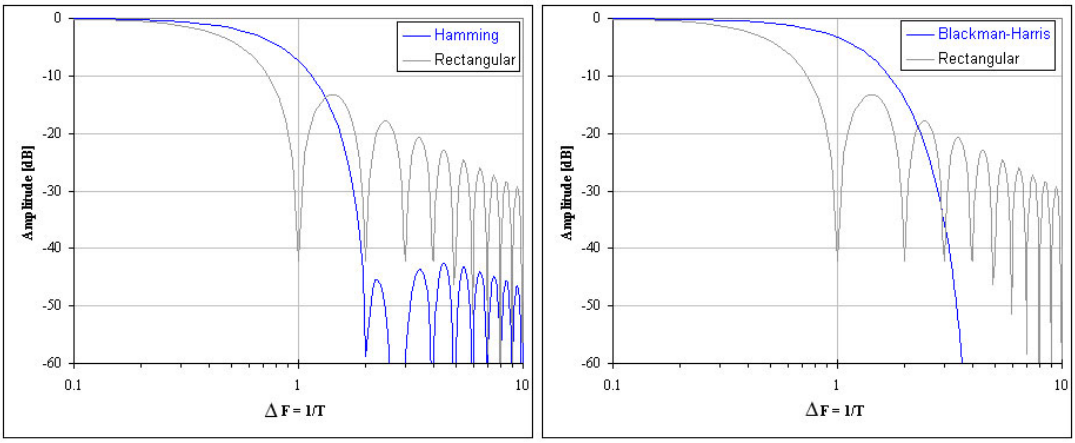


Abbildung 3.3: Vergleich von Rechteck- und Hamming-Fenster (a) sowie von Rechteck- und Blackman-Harris-Fenster (b)

3.2.4 Zeit- und Frequenz-Auflösung bei der Fourier Transformation

Betrachtet man das Gesamtergebnis der STFT, so hat dies eine bestimmte Zeit- sowie eine Frequenzauflösung. Die Zeitauflösung wird durch die Länge der Fenster bestimmt. Je geringer die Länge, desto genauer entspricht das Teilspektrum exakt dem tatsächlichen zum jeweiligen Zeitpunkt. Aber auch die Frequenzauflösung ist direkt von der Länge des Fensters abhängig. Je größer die Länge, desto schmäler werden die resultierenden Frequenzbänder.

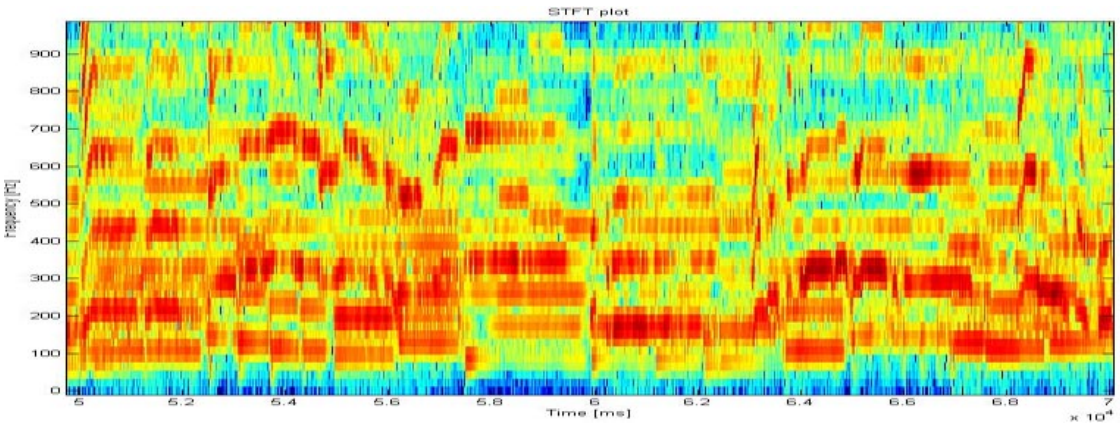
Daraus ergibt sich, dass eine genauere Auflösung in der einen Dimension immer auf Kosten der Genauigkeit in der jeweils anderen Dimension geht. Es gilt also einen vernünftigen Mittelweg zu finden. In [18] wird im gegebenen Kontext eine relativ große Fensterlänge von $N_{frame} = 4096$ Samples vorgeschlagen. Dies entspricht bei einer Abtastrate von $f_s = 44.1$kHz einer Zeitspanne von 93ms. Andere Autoren – wie etwa Fujishima [17] – gehen von noch längeren Frames aus.

Ein Ausweg aus dem beschriebenen Dilemma ist das so genannte *Zero-Padding*. Dabei werden am Beginn sowie am Ende des gefensterten Signals Nullen eingefügt und so die Fensterlängen künstlich vergrößert, ohne dabei zusätzliche Bereiche des Eingangssignals zu berühren. Abbildung 3.4 verdeutlicht jedoch, dass durch eine Verdoppelung der tatsächlichen Fensterlänge eine bessere Frequenzauflösung erreicht wird. Das zusätzliche Zero-Padding mit einem Faktor von 2 (nochmalige Verdoppelung der Eingangsdaten für jede DFT) wirkt sich optisch nicht auf das Spektrum aus. Das Ergebnis wird auch nicht genauer in dem Sinn, dass die Frequenzen der Stimmen selbst schärfer extrahiert werden, sondern dass die einzelnen Bänder schmäler sind.

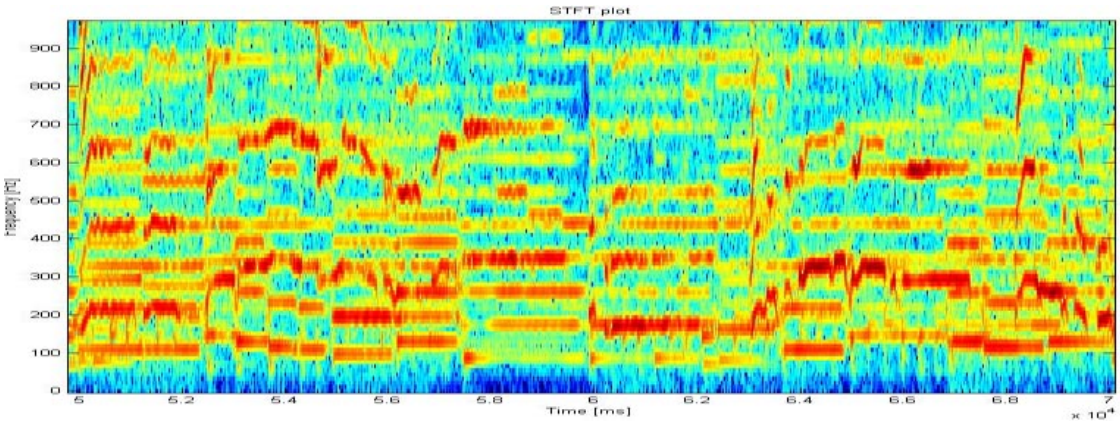
Ein weiterer Punkt der die Zeitauflösung betrifft, ist die Anzahl an Samples N_{hop} um die die Fensterfunktion in jedem Schritt verschoben wird. Ist sie größer als N_{frame} geht Information verloren, da bestimmte Bereiche nie innerhalb eines Fensters zu liegen kommen. Da außerdem die Abtastpunkte an den Rändern des Fensters zunehmend ausgeblendet werden, empfehlen sich Schrittweiten von maximal der Hälfte der Fensterlängen. Üblich sind Verhältnisse $N_{hop} : N_{frame}$ zwischen $1 : 2$ und $1 : 8$ um zu erreichen, dass alle Daten möglichst gleichmäßig gewichtet sind (vgl. [15], [18], [30], [43]). Allen [2] schlägt konkret vor, eine Sprungweite N_{hop} zu verwenden, die der Framegröße dividiert durch die Breite des Haupt-Peaks der Fensterfunktion im Frequenzbereich (vgl. Tabelle 3.1) $\frac{N_{frame}}{b}$ entspricht.

3.2.5 Constant Q Transformation

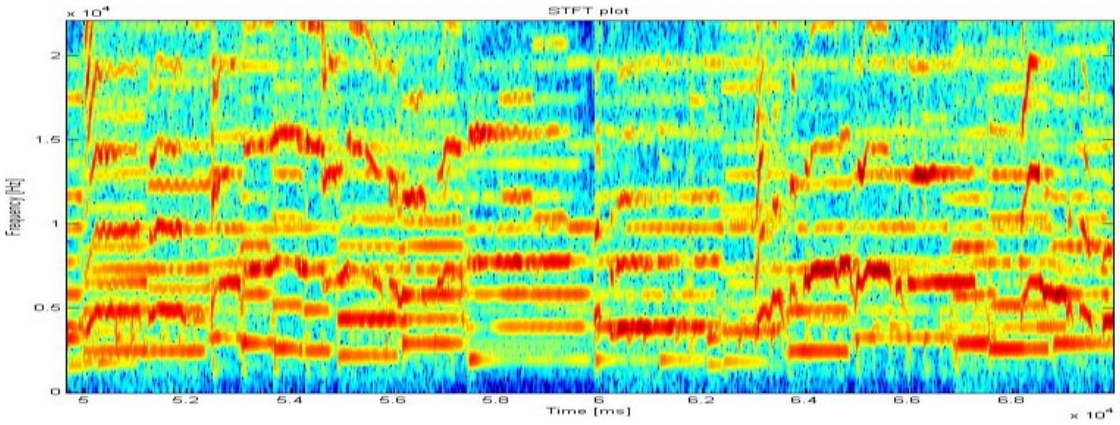
Eine andere Herangehensweise an das Problem der Wahl zwischen Zeit- und Frequenzauflösung bietet die Constant-Q Transformation [7]. Anders als bei der Fourier Transformation werden die Mitten der Frequenzbänder nicht in gleichen Abständen gewählt, sondern entsprechend einer geometrischen Folge mit $f_k = f_0 \cdot 2^{\frac{k}{b}}$. f_0 steht dabei für die Grundfrequenz und b für die Anzahl der Frequenzbänder pro Oktave. Der Name ergibt sich aus dem konstanten Verhältnis Q zwischen Frequenz und Auflösung, das unabhängig vom Frequenzband ist und allein vom Parameter b abhängt.



(a)



(b)



(c)

Abbildung 3.4: Vergleich verschiedener Frequenzauflösungen anhand eines Ausschnitts aus dem Spektrum von *Yesterday* von den *Beatles*. Zwischen dem Spektrum anhand einer Fensterbreite von 2048 Samples (a) und 4096 Samples (b) ist ein deutlicher Unterschied zu sehen ist. Für (c) wurde ein Fenster von 4096 Samples und zusätzlich ein Zero-Padding Faktor von 2 verwendet.

$$\begin{aligned} Q &= \frac{f_k}{f_{k+1} - f_k} \\ &= \frac{f_k}{f_k(2^{\frac{1}{b}} - 1)} \\ &= \frac{1}{2^{\frac{1}{b}} - 1} \end{aligned} \tag{3.8}$$

Neben der offensichtlichen Stärke, dass die Resultate dieser Art der Transformation bei entsprechender Wahl der Parameter direkt mit musikalischen Notensystemen korrespondieren, ist sie auch besser an das menschliche Gehör angepasst. Dieses kann bei hohen Frequenzen die exakte Tonhöhe wesentlich unschärfer erkennen als bei niedrigen. Dafür benötigt es bei tiefen Frequenzen mehr Zeit um sie zu bestimmen [5].

Die Formel zur Berechnung der Transformation ist jener der Fourier Transformation ähnlich.

$$X_{cq}(k) = \frac{1}{N_k} \sum_{n<N_k} x(n) e^{-j2\pi Q \frac{n}{N_k}} \tag{3.9}$$

mit Q laut Formel 3.8 und $N_k = Q\lceil \frac{f_s}{f_k} \rceil$

Schreibt man die Transformation als Matrix-Multiplikation, so erhält man

$$X_{cq} = x \cdot T^*_{nk} \qquad \text{mit} \tag{3.10}$$

$$T_{nk} = \begin{cases} \frac{1}{N_k} e^{j2\pi Q \frac{n}{N_k}} & \text{wenn} \quad n < N_k \\ 0 & \text{sonst} \end{cases} \tag{3.11}$$

Um Aufwand bei der Matrix-Multiplikation einzusparen, führt man sie bei der effizienten Berechnung laut [8] im Frequenzbereich durch. Dazu wird der Kernel T_{nk} spaltenweise fourier-transformiert ($T_{nk} \overset{Fourier}{\Rightarrow} S_{nk}$). Eliminiert man in der so erhaltenen Matrix alle Werte unter einer gewissen Schwelle, so bleiben nur mehr wenige Werte ungleich Null übrig und die Berechnung der Constant Q Transformierten – entsprechend Formel 3.12 – erfordert nicht mehr viel Rechenaufwand. Ein effizienter Algorithmus

zur Berechnung der jeweiligen Fourier Transformationen wurde mit der FFT bereits vorgestellt (siehe Seite 22).

$$X_{cq} = \frac{1}{N} X_{fourier} \cdot S^*_{nk} \tag{3.12}$$

Im Vergleich zur (Diskreten) Fourier Transformation hat die Constant-Q Transformation im Bereich der Musikverarbeitung den Vorteil, die Output-Datenrate bei gleichem Nutzen der Ergebnisse zu reduzieren. Der Grund dafür ist die niedrigere Frequenzauflösung in Bereichen, in denen das menschliche Gehör Frequenzen ohnehin schlechter unterscheiden kann. Weiters fällt das Arbeiten auf den Resultaten leichter, da musikalische Intervalle bzw. ein bestimmter Oberton immer derselben Differenz der Band-Indizes entsprechen. Dem gegenüber steht der Nachteil der komplexeren Berechnung.

Weiters sei die (Diskrete) Wavelet-Transformation erwähnt, die ebenfalls Signale vom Zeit- in den Frequenzbereich transformiert. Grundlage der Transformation sind jedoch nicht Phasoren bzw. Sinusschwingungen wie bei den beiden vorgestellten, sondern Wavelets.

3.2.6 Übergangserkennung

Bevor weitere Verarbeitungsschritte folgen, werden die Audiodaten einer zusätzlichen Vorverarbeitung unterzogen. Ziel ist es, jenen Teil der Daten zu extrahieren, der tatsächlich verwertbare Informationen enthält.

Bei Musikstücken werden je nach Takt und Tempo Töne zu bestimmten Zeiten (Taktzeiten) angespielt, halten dann eine gewisse Zeit an und klingen schließlich ab. Ein Klangbild wird also in sehr kurzer Zeit erzeugt und bleibt dann relativ lange stabil. Diese stabile Phase ist sehr informativ, während die instabilen Abschnitte, in denen sich das Klangbild verändert wenig Informationen enthalten. Dazu kommt weiters, dass die Taktzeiten von Schlagwerken und perkussiven Instrumenten dominiert werden. Der Anteil an Rauschen ist daher sehr hoch.

Zur Bestimmung des Klangbilds sind die stabilen Phasen relevant. Instabile Perioden – insbesondere die Taktzeiten – müssen also erkannt und entfernt werden. Methoden dazu liefern unter anderem Bonada [6] oder Roebel [41].

Der in [6] beschriebene Ansatz besteht im Wesentlichen darin, Intensitätsänderungen in einzelnen Frequenzbändern des Signals zu betrachten und lokale Maxima zu finden.

Features die dazu verwendet werden sind unter anderen Mel Cepstrum Koeffizienten und deren Ableitungen.

Roebel [41] hingegen bedient sich des in [11] als

$$\bar{t} = \frac{\int -\frac{\partial\phi(\omega)}{\partial\omega} A(\omega)^2 d\omega}{\int A(\omega)^2 d\omega} \tag{3.13}$$

definierten Schwerpunkts der momentanen Energie eines Signals, wobei ω die Frequenz darstellt. Die Definition bezieht sich zwar auf stetige Signale, kann jedoch leicht für diskrete Signale abgewandelt werden.

Grundlage des Verfahrens ist, dass sich nach einer Fourier Transformation die Phase von stabilen Frequenzanteilen nicht ändert. Somit ergibt $\frac{\partial\phi(\omega)}{\partial\omega} = 0$. In der Folge bleibt somit auch der Schwerpunkt $\bar{t} = 0$ – d.h. in der Mitte des Frames. Dementsprechend können instabile Perioden daran erkannt werden, dass der Schwerpunkt dezentral liegt. Kommt ein Frequenzanteil neu hinzu, erscheint er zuerst im rechten Teil des Fensters und der Schwerpunkt wird dorthin verschoben. Verschiebt man den Frame nun weiter, so rückt der Schwerpunkt entgegengesetzt immer weiter zurück zur Mitte. Der umgekehrte Fall tritt auf, wenn ein Frequenzanteil ausklingt.

3.2.7 Peak Filterung

Weiterer Teil der Spektralanalyse ist die Erkennung von besonders stark ausgeprägten Frequenzanteilen, da diese die vom Menschen wahrgenommenen Harmonien und Melodien ausmachen. Dabei orientiert sich diese Arbeit am Modell von Serra [43], das ein Signal als Überlagerung eines deterministischen Anteils und eines stochastischen Anteils interpretiert. Der deterministische Signalanteil besteht dabei aus Sinusoiden – stabilen Frequenzen, deren Amplituden und Phasenverschiebungen sich über einen gewissen Zeitraum nicht wesentlich verändern. Die stochastische Komponente beschreibt ein zusätzliches Rauschen als Wahrscheinlichkeitsfunktion möglicher Amplituden für jede Frequenz. Das gesamte Klangbild wird somit entsprechend folgender Formel modelliert.

$$s(t) = \sum_{r=0}^{R-1} A_r(t) cos(\theta_r(t)) + e(t) \tag{3.14}$$

$e(t)$ ist dabei das Rauschen zum entsprechenden Zeitpunkt, R die Anzahl der mitschwingenden Sinusoide und A_r sowie θ_r die jeweiligen momentanen Amplituden sowie Phasen dieser Komponenten. Die momentane Phase lässt sich leicht aus der Frequenz sowie der generellen Phasenverschiebung einer Schwingung herleiten.

$$\theta_r(t) = \int_0^t \omega_r(\tau)d\tau + \theta_{r_0} \tag{3.15}$$

Dieses Modell ist insofern wertvoll, als dass man bei der Analyse im gegebenen Kontext den Rausch-Anteil vernachlässigen und sich auf die deterministischen Komponenten des Signals konzentrieren kann. Diese werden ausgehend von dem nach Formel 3.2 berechneten Amplitudenspektrum bestimmt. Kandidaten für mögliche Sinusoide sind *Peaks* – lokale Maxima innerhalb eines Frames – im Spektrum. Zusätzliche Kriterien sind eine gewisse Mindest-Amplitude, sowie das Verhältnis der Amplitude am Maximum im Vergleich zu den beiden benachbarten Minima (vgl. [43]).

Mit dieser Methode können Frequenzbänder, die einen Peak enthalten, bestimmt werden. Da jedes Frequenzband jedoch eine Breite Δf besitzt, kann die Peak-Frequenz mit einem Fehler von bis zu $\pm\frac{\Delta f}{2}$ behaftet sein. Eine Möglichkeit, diesen Fehler einzuschränken, ist eine Erhöhung der Frequenzauflösung etwa durch das bereits beschriebene Zero-Padding. Alternativ dazu kann die Lage des Peaks durch Interpolation mit Hilfe der benachbarten Werte angenähert werden. In [18] und [43] ist dazu die Methode der Quadratischen Interpolation des Spektrums beschrieben bei der davon ausgegangen wird, dass die drei dem Maximum am nächsten liegenden Punkte auf einer Parabel liegen, welche durch die Formel

$$y(x) = a(x-p)^2 + b \tag{3.16}$$

festgelegt ist. a ist dabei der Krümmungsfaktor, b die Verschiebung entlang der Ordinate und p das gesuchte Zentrum der Parabel auf der Abszisse. Dieses kann unter Verwendung der Amplituden (in dB) im Spektrum (α, β, γ), die dem Maximum am nächsten liegen ($\alpha \leq \beta \geq \gamma$) errechnet werden.

$$p = \frac{1}{2} \cdot \frac{\alpha - \gamma}{\alpha - 2\beta + \gamma} \tag{3.17}$$

Somit ergeben sich für die approximierte Frequenz und Amplitude des Peaks

$$f_p = f_{bin} + p\Delta f \tag{3.18}$$

$$A_p = \beta + \frac{1}{4}(\alpha - \gamma)p \tag{3.19}$$

Für viele der weiteren Anwendungen reicht das Finden der Peaks als Kandidaten für stabile Komponenten. Es stellt eine Beschränkung auf die wesentlichen Frequenzanteile eines Signals dar. Ein Beispiel dafür ist in Abbildung 3.5 dargestellt. Das Gesamtspektrum dieses Ausschnitts wurde dabei bereits in Abbildung 3.4 gezeigt. Ein weiterer Vorteil liegt natürlich in der Datenreduktion und der damit verbundenen Effizienzsteigerung aller folgenden Verarbeitungsschritte. Zusätzliche Datenreduktion kann durch Berücksichtigung von Maskierungseffekten erreicht werden (vgl. Abschnitt 3.2.8).

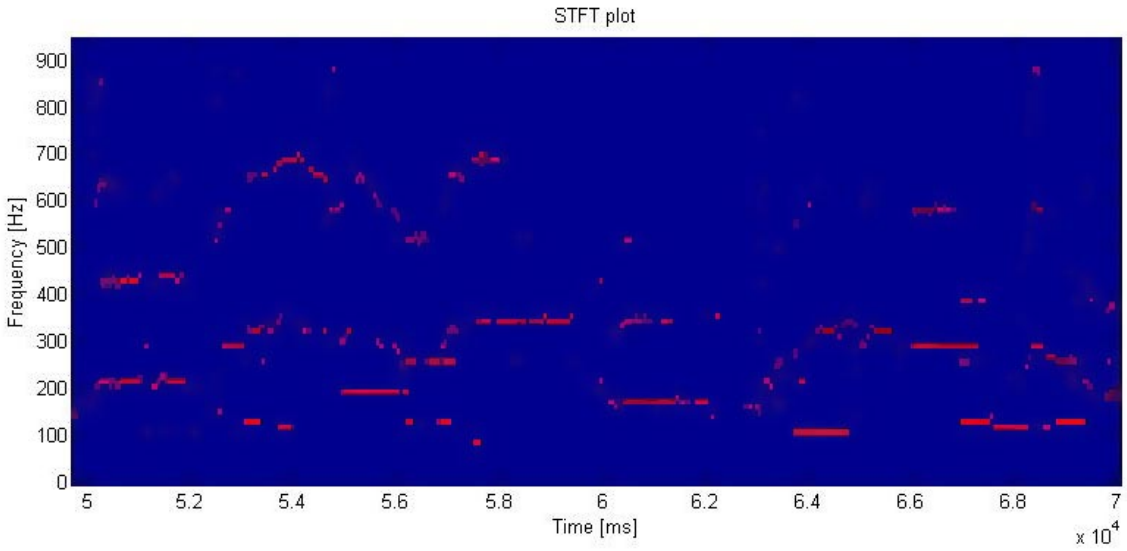


Abbildung 3.5: Gefundene Peaks in einem Ausschnitt aus *Yesterday* von den *Beatles*

Es gibt jedoch auch Anwendungen, die das Verbinden einzelner Peaks zu Trajektorien erfordern. Neben der ursprünglichen Anwendung, in dem in [43] beschriebenen *Digital Phase Vocoder*, ist die Melodieerkennung (vgl. Abschnitt 3.6) ein weiteres Beispiel.

3.2.8 Maskierung

Da die Bewertung aller Ergebnisse von Audioverarbeitung im gegebenen Kontext in letzter Instanz vom Zuhörer – also vom Menschen – vorgenommen wird, ist es hilfreich,

sich am menschlichen Gehör zu orientieren. Der Forschungsbereich der Psycho-Akustik liefert dazu Modelle, die die Audio-Wahrnehmung des Menschen beschreiben. Diese Wahrnehmung unterliegt bestimmten Einschränkungen bzw. Unschärfen. Berücksichtigt man diese, so können einerseits Algorithmen besser an die perzeptuellen Vorgänge beim Menschen angepasst werden. Andererseits ist es möglich durch Vernachlässigung von Daten, die vom Menschen ohnehin nicht wahrgenommen werden, eine Reduktion des zu verarbeitenden Datenvolumens zu erreichen. Dieses Prinzip machen sich beispielsweise verlustbehaftete Komprimierungsverfahren zu Nutzen.

Die wichtigsten der angesprochenen Einschränkungen der Wahrnehmung sind die absolute Hörschwelle, simultane Maskierung, temporale Maskierung und Eigenschaften der kritischen Bänder (vgl. [36] und [48]). Ihr Einfluss auf die Audioverarbeitung wird im Folgenden beschrieben.

3.2.8.1 Absolute Hörschwelle

Die absolute Hörschwelle beschreibt den minimalen Schalldruck (in dB) der notwendig ist, damit ein Ton ohne jegliche Störgeräusche von einem Menschen wahrgenommen werden kann. Dieser ist abhängig von der Frequenz und kann verallgemeinernd für Menschen mit gutem Gehör durch die Funktion

$$T_q(f) = 3.64\left(\frac{f}{1000}\right)^{-0.8} - 6.5e^{-0.6\left(\frac{f}{1000}-3.3\right)^2} + 10^{-3}\left(\frac{f}{1000}\right)^4 \quad [dB] \tag{3.20}$$

angenähert werden. Entsprechend dieser Formel ist das menschliche Gehör bei Frequenzen rund um 4kHz am empfindlichsten. Abbildung 3.6 verdeutlicht den Zusammenhang zwischen Frequenz und Hörschwelle.

Diese Erkenntnis kann man sich insofern zu Nutze machen, als dass man alle durch die bisherigen Arbeitsschritte gefundenen Peaks vernachlässigt, deren Intensität unter der für ihre Frequenz zutreffende Hörschwelle fällt.

3.2.8.2 Kritische Bänder

Die absolute Hörschwelle ist jedoch aus zweierlei Gründen von untergeordnetem Wert. Einerseits ist ein Ton in Musikstücken nie allein zu hören. Aufgrund des Mitschwingens von Obertönen sind immer mehrere Frequenzen gleichzeitig präsent. Andererseits spielt

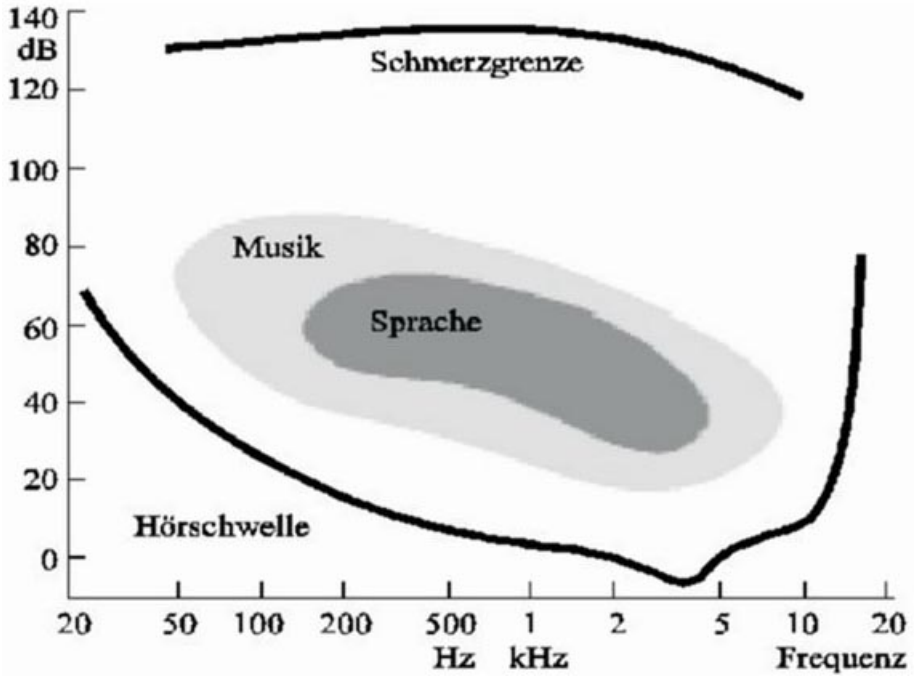


Abbildung 3.6: Absolute Hörschwelle des Menschen ([34])

auch die zeitliche Auflösung eine Rolle bei der Wahrnehmung. Töne die zu kurz klingen werden ebenfalls vom Menschen nicht gehört.

Bessere Aussagen lassen sich treffen, wenn man das Phänomen der kritischen Bänder berücksichtigt. Ein derartiges Band ist ein Wahrnehmungskanal in der Hörschnecke (Cochlea) – die man sich als Bank von überlappenden Bandpassfiltern vorstellen kann. Experimente zeigten, dass eine Frequenz mit einer bestimmten Intensität genauso laut wahrgenommen wird, wie mehrere Frequenzen mit derselben Intensität, wenn diese innerhalb des gleichen kritischen Bands liegen. Erst wenn eine Frequenz außerhalb des Bands mitschwingt wird die Überlagerung als lauter empfunden. Frequenzen innerhalb eines kritischen Bands werden also stark gekoppelt wahrgenommen und können daher auch nur schwer unterschieden werden. Die Breite eines solchen Bandes ist abhängig von der Frequenz im Zentrum und entspricht genau einem Bark (vgl. Abschnitt 2.2.1).

Sowohl simultane als auch temporale Maskierungseffekte sind somit von der Existenz der kritischen Bänder betroffen. Das grundlegende Schema ist jedoch das gleiche: Liegen zwei Frequenzen nahe beisammen, so sind sie nur schwer zu unterscheiden. Diese Schwierigkeit nimmt ab, je weiter sie sich voneinander entfernen.

3.2.8.3 Simultane Maskierung

Unter Maskierung versteht man den Effekt, dass ein Ton nicht hörbar ist, weil er von einem anderen übertönt wird. Simultane Maskierung tritt auf, wenn diese beiden Töne gleichzeitig erklingen. Aus biologischer Sicht lässt sich das so erklären, dass die Basilarmembran in der Cochlea durch einen lauten Ton so stark erregt wird, dass die Schwingung, die durch einen leiseren Ton mit ähnlicher Frequenz auf der Membran erzeugt wird nicht, als eigener Stimulus erkannt wird.

Eine weitere Besonderheit ist, dass es einen Unterschied macht, ob es sich bei den jeweiligen Schwingungen um einzelne Töne oder um Rauschen innerhalb eines relativ schmalen Bands (1 Bark) handelt. Der Begriff Asymmetrie von Maskierung beschreibt, dass ein Ton, der Rauschen maskiert, wesentlich stärker sein muss (Signal-Masked-Ratio SMR = 24dB) als Rauschen, das einen Ton maskiert (SMR = 4dB). Mathematisch beschrieben wird diese Asymmetrie durch die beiden Formeln für die Schwellwerte. Diese drücken aus, ab welcher Intensität TH_T ein Ton trotz Rausches mit einer Intensität von E_N innerhalb des selben kritischen Bands noch zu hören ist bzw. umgekehrt:

$$TH_N = E_T - 14.5 - B \tag{3.21}$$

bzw.

$$TH_T = E_N - K \tag{3.22}$$

B steht dabei für die Nummer des kritischen Bands und K ist ein individueller Parameter, der in der Regel zwischen 3 und 5 dB liegt. Abbildung 3.7 verdeutlicht nochmals das Phänomen der Asymmetrie der Maskierung.

Weiters zu berücksichtigen ist die Ausbreitung des Einflusses eines maskierenden Tons außerhalb des eigenen kritischen Bands. Der Maskierungseffekt auf niedrigere Frequenzen ist dabei wesentlich geringer als jener auf höhere. Näherungsweise ist der Maskierungsschwellwert um 25dB pro Bark in absteigende und um 10dB pro Bark in aufsteigende Richtung verringert. Ausgedrückt wird dies durch die analytische Funktion

$$SF(x) = 15.81 + 7.5(x + 0.474) - 17.5\sqrt{1 + (x + 0.474)^2} \quad [dB] \tag{3.23}$$

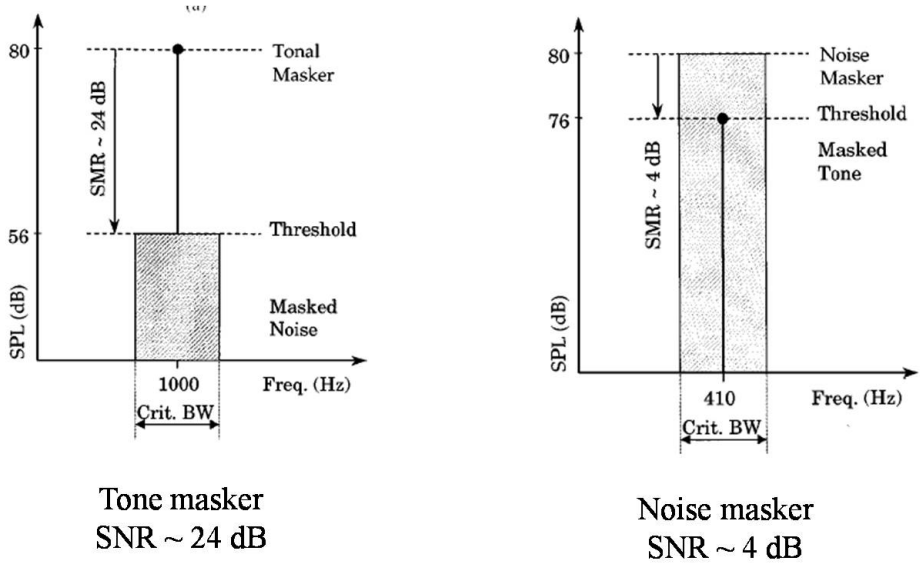


Abbildung 3.7: Asymmetrie der Maskierung zwischen Tönen und Rauschen ([36])

Die Abszisse x ist dabei in der Einheit Bark. Gezeigt wird der Sachverhalt in Abbildung 3.8.

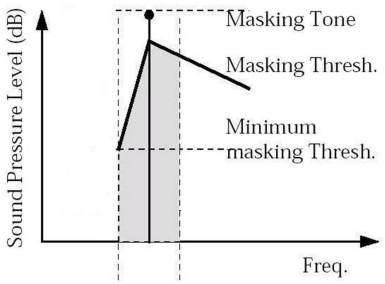


Abbildung 3.8: Maskierung außerhalb des eigenen kritischen Bands ([36])

Sind alle einzelnen Schwellwerte bekannt, so kann daraus ein globaler Maskierungsschwellwert abgeleitet und auf die erkannten Peaks angewandt werden. In Abbildung 3.9 sind dabei in (a) die Maskierungsschwellwerte ausgehend von den Peaks die Tönen zugeordnet werden können, in (b) jene, die von Rauschen stammen und in (c) der globale Maskierungsschwellwert unter Miteinbeziehung der absoluten Hörschwelle dargestellt.

3.2.8.4 Nicht simultane Maskierung

Nicht simultane oder temporale Maskierung beschreibt den Effekt, dass ein Ton nicht nur gleichzeitig erklingende Töne maskieren, sondern auch davor oder noch danach

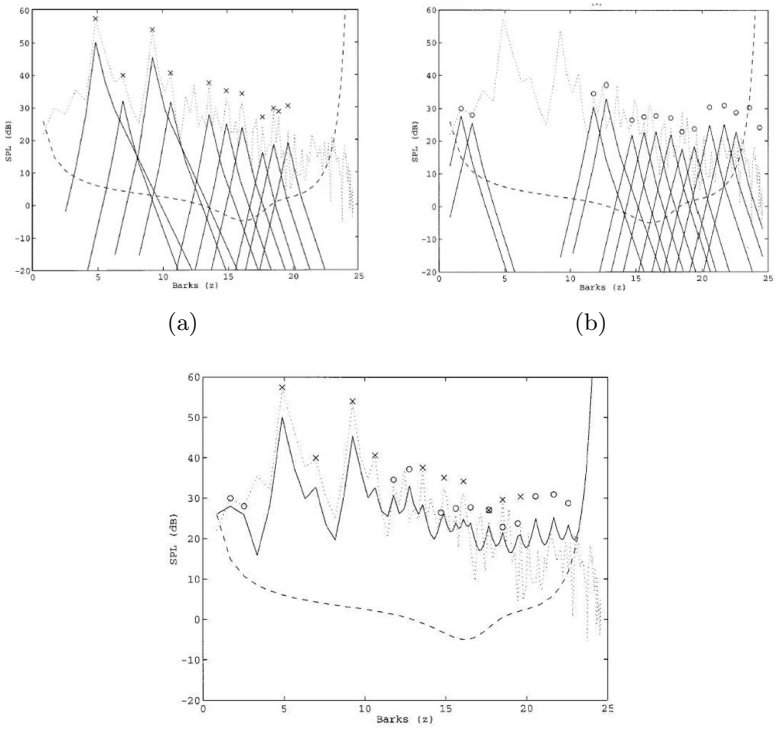


Abbildung 3.9: Maskierungsschwellwerte ausgehend von Tönen (a) und Rauschen (b) sowie die daraus resultierenden globalen Werte (c) (aus [36])

erklingende beeinflussen kann. Dabei ist die Zeitspanne des Maskierungseffekts vor dem Ton mit $1-2$ ms wesentlich kürzer als nach ihm, wo sie $50-300$ ms beträgt. Die genaue Dauer ist abhängig von der Intensität und der Dauer des maskierenden Tons selbst.

3.3 Referenz Frequenz

Die zuvor beschriebenen Verarbeitungsschritte waren alle auf Merkmale unterster Ebene ausgerichtet. Darauf aufbauend werden solche auf höheren Abstraktionsniveaus abgeleitet. Im Rahmen der Musikverarbeitung ist es dabei sinnvoll, ein Stück auch durch Begriffe und Konzepte der Musik zu beschreiben. Der wesentlichste Schritt ist dabei jener weg von Frequenzen als Eigenschaft von Tönen hin zu musikalischen Tonwerten und -systemen.

Laut ISO 16:1075[2] ist der Kammerton a' mit 440Hz definiert. Die exakten Frequenzen für alle anderen Töne können leicht berechnet werden, da sie einer logarithmischen Skala folgen. Die Verdoppelung der Frequenz entspricht einem Anheben der Tonhöhe um eine

[2]http://www.iso.org

Oktave. Bei 12 Halbtönen pro Oktave hat der n-te Halbton über dem Kammerton die Frequenz

$$f_n = 440Hz \cdot 2^{\frac{n}{12}} \tag{3.24}$$

Die Referenzfrequenz f_{ref} für die Ausrichtung des Tonsystems lag jedoch nicht immer bei diesen 440Hz und auch heute werden von Orchestern und Bands sehr oft andere Stimmungen verwendet. Ein genaues System zum Umlegen der Frequenzen aus der Spektralanalyse auf musikalische Tonhöhen muss dies berücksichtigen und die Referenzfrequenz selbständig aus dem Audiosignal ableiten können.

Dabei sind zwei prinzipielle Herangehensweisen denkbar: Eine ist, die Bestimmung der Referenzfrequenz bereits vor der Zuordnung zu Tonhöhen bzw. Pitch-Klassen durchzuführen – die andere, dies erst danach in Form einer Korrektur zu erledigen. Ein Beispiel für letztere ist das in [17] beschriebene Verfahren. Dabei wird von einer sehr hohen Frequenzauflösung ausgegangen. Die feinen Bänder werden mit einer Genauigkeit von 100 Teilbändern pro Halbton auf eine Oktave gefaltet. Von diesen 100 Bändern wird dann jeweils die Varianz ermittelt. Die Bestimmung der Referenzfrequenz erfolgt indirekt dadurch, dass das Halbtonprofil nun ringförmig über die Frequenzbänder geshiftet wird, bis ein Minimum in der Varianz innerhalb der Bändergruppen gefunden ist. Gegen dieses Verfahren sprechen jedoch die sehr hohe nötige Frequenzauflösung, sowie die Tatsache, dass es patentrechtlich[3] geschützt ist.

Der andere Weg – die Referenzfrequenz bereits vor der Zuordnung zu musikalischen Noten zu bestimmen – ist in [18] und [49] beschrieben. Der Grundgedanke dabei ist, alle im Spektrum gefundenen Peaks mit den jeweiligen Frequenzen f_i auf Übereinstimmung mit den Lagen der Halbtöne zu prüfen. Geht man vom Kammerton (a' = 440Hz) aus, so ist das Intervall – gemessen in Halbtönen – zwischen der Referenzfrequenz und dem entsprechenden Peak

$$\Delta_i = 12 \cdot \log_2 \left(\frac{f_i}{440} \right) \tag{3.25}$$

Ist Δ_i nicht ganzzahlig, so ist der Ton i verstimmt, und zwar um einen Faktor

[3] US Patent 6057502

$$d_i = \Delta_i - round(\Delta_i) \quad \text{mit } d_i \epsilon [-0.5, 0.5[\tag{3.26}$$

Um nun den Verstimmungsfaktor aller einzelnen Peaks eines Frames zu einer Zahl zu akkumulieren, wird ein Histogramm mit einer, der nötigen Genauigkeit entsprechenden Auflösung r über das Intervall $[-0.5, 0.5[$ gebildet. Aus dem Modalwert $argmax_n(hist(n))$ kann somit der Verstimmungsfaktor als

$$d = -0.5 + r \cdot argmax_n(hist(n)) \tag{3.27}$$

abgeleitet werden. Daraus ergibt sich eine Referenzfrequenz von

$$f_{ref} = 440 \cdot 2^{\frac{d}{12}} \tag{3.28}$$

Der Grund, wieso es nicht zielführend ist, den Mittelwert μ_d aller Einzelwerte d_i zur Akkumulierung heranzuziehen ist, dass er in bestimmten Fällen falsche Werte liefern würde. Liegt die tatsächliche Verstimmung relativ knapp bei einem halben Halbton, so würden die errechneten Werte aufgrund der Ungenauigkeiten der STFT und den auf ihr aufbauenden Berechnungen (nur interpolierte und keine exakten Frequenzen) teils dem Halbton darüber und teils dem Halbton darunter zugerechnet werden. Die Werte d_i wären somit entweder ganz am unteren oder ganz am oberen Ende des Intervalls und würden sich bei der Bildung des arithmetischen Mittels auf einen Wert nahe 0 ausgleichen. Dies wäre jedoch genau das Gegenteil des eigentlichen Sachverhalts.

Um von diesen Werten für die Referenzfrequenz, die sich auf jeweils einen Frame beziehen, auf das gesamte Stück schließen zu können, ist ein weiterer Schritt nötig. Aus denselben Gründen, wie bereits oben genannt, wird auch hier nicht das arithmetische Mittel zur Akkumulierung herangezogen, sondern wieder ein Histogramm. Dieses hat ebenfalls einen Wertebereich von $[-0.5, 0.5[$ und eine beliebig wählbare Auflösung r. Das Resultat d wird analog zu den Teilergebnissen gemäß Formel 3.27 ermittelt. Dabei ist es nicht einmal nötig alle Frames in die Berechnung des Histogramms einzubeziehen, sondern es macht durchaus Sinn nur einen bestimmten Abschnitt– etwa 30 bis 60 Sekunden – aus der Mitte des Stücks herauszunehmen. Der Grund dafür ist, dass dort möglichst viele Instrumente und Stimmlagen vertreten sind, im Gegensatz zu Intro sowie Schlussteil, die oft nur von wenigen Stimmen gespielt werden. Die endgültige

Referenzfrequenz ergibt sich unabhängig davon laut Formel 3.28, wobei d wiederum der Modalwert der Verteilung ist.

3.3.1 Referenzfrequenz und Obertonreihen

Wie in Abschnitt 2.2.3 beschrieben, schwingt mit einem Ton immer eine ganze Reihe an Obertönen mit. Diese entsprechen den ganzzahligen Vielfachen der Grundfrequenz. Das bedeutet, dass alle jene Obertöne mit einer Zweierpotenz als Faktor genau um eine oder mehrere Oktaven verschoben sind. Alle anderen fallen jedoch nicht mit einer Stufe des musikalischen Tonsystems zusammen. Zusätzlich zu den stochastischen Anteilen des Signals wird dadurch auch die Referenzfrequenz-Bestimmung beeinflusst. Denn schließlich kann nicht unterschieden werden, ob eine Frequenz nicht mit der eines exakten Tons übereinstimmt weil die Referenzfrequenz eine andere ist, oder weil es sich bei der Frequenz nicht um einen Grund- sondern einen Oberton handelt.

Bei einem tatsächlichen Verstimmungsfaktor von δ haben die Obertöne somit folgende Frequenzen f'_n.

$$f'_n = n \cdot \delta \cdot f_0 \tag{3.29}$$

Daraus ergibt sich als Abweichung des Obertons von der Referenzfrequenz ein Δ_n von

$$\begin{aligned} \Delta_n &= 12 \cdot \log_2 \left(\frac{n \cdot \delta \cdot f_0}{440} \right) && (3.30) \\ &= 12 \cdot \log_2 \left(\frac{f_0}{440} \right) + 12 \cdot \log_2(n) + 12 \cdot \log_2(\delta) \\ &= dev_{f_0} + dev_n + \Delta && (3.31) \end{aligned}$$

dev_{f_0} beschreibt das Intervall zwischen dem Grundton und dem Kammerton ohne Einfluss einer Verstimmung. Δ drückt eben diese Verstimmung (siehe Formel 3.25) in Bezug auf den Grundton aus. $dev_n = 12 \cdot \log_2(n)$ hingegen ist die neu hinzukommende Verschiebung aufgrund der Lage des n-ten Obertons. Berechnet man d laut Formel 3.26 so ergibt sich ein Einfluss von dev_n auf d als

$$d_n = dev_n - round(dev_n) = 12 \cdot \log_2(n) - round(12 \cdot \log_2(n)) \tag{3.32}$$

Frequenz	dev_n	d_n	**Frequenz**	dev_n	d_n
f_0	0	0	$11 \cdot f_0$	41.5132	-0.4868
$2 \cdot f_0$	12.0000	0	$12 \cdot f_0$	43.0196	0.0196
$3 \cdot f_0$	19.0196	0.0196	$13 \cdot f_0$	44.4053	0.4053
$4 \cdot f_0$	24.0000	0	$14 \cdot f_0$	45.6883	-0.3117
$5 \cdot f_0$	27.8631	-0.1396	$15 \cdot f_0$	46.8827	-0.1173
$6 \cdot f_0$	31.0196	0.0196	$16 \cdot f_0$	48.0000	0
$7 \cdot f_0$	33.6883	-0.3117	$17 \cdot f_0$	49.0496	0.0496
$8 \cdot f_0$	36.0000	0	$18 \cdot f_0$	50.0391	0.0391
$9 \cdot f_0$	38.0391	0.0391	$19 \cdot f_0$	50.9751	-0.0249
$10 \cdot f_0$	39.8631	-0.1369	$20 \cdot f_0$	51.8631	-0.1369

Tabelle 3.2: Verschiebung der Obertöne

Aus Tabelle 3.2 wird ersichtlich, dass das Gesamtergebnis für die Referenzfrequenz f_{ref} durch die Obertöne nicht maßgeblich verfälscht wird. Die Werte nahe bei 0 dominieren die obertonabhängigen Verschiebungen – zwölf von zwanzig Werten weichen um weniger als 0.1 von 0 ab. Da weiters nicht der Mittelwert, sondern der Modalwert ausschlaggebend für das Resultat ist, fallen die zwangsläufigen Ausreißer nicht ins Gewicht und können somit vernachlässigt werden.

3.3.2 Resultate

Das Ergebnis der Referenz-Frequenz Analyse des Titels *Yesterday* der *Beatles* ist in Abbildung 3.10 ersichtlich. Obwohl die Werte im Plot bei diesem Beispiel relativ ungeordnet liegen, kann man doch anhand des Histogramms eine eindeutige Spitze erkennen. In diesem Fall wurde eine Frequenz f_{ref} von 434.70Hz bestimmt. Tabelle 3.3 listet zudem einige weitere bekannte Titel der *Beatles* und die in der Implementierung erkannten Referenzfrequenzen f_{ref} sowie die Signifikanz der jeweiligen Schätzungen auf, die hier zum Einen in Anlehnung an die Varianz ausgedrückt werden soll als

$$s_1 = \frac{1000}{\sqrt{\frac{\sum_{i=0}^{F}(f_i - f_{ref})^2}{F}}} \tag{3.33}$$

Neben diesem Maß, das besagt, wie nahe die übrigen Werte bei dem tatsächlich geschätzten liegen, drückt s_2 aus, wie stark sich f_{ref} vom zweitstärksten Kandidaten für die Referenzfrequenz f_{second} abhebt.

$$s_2 = 1 - \frac{hist(f_{second})}{hist(f_{ref})} \tag{3.34}$$

Titel	f_{ref}	**Signifikanz** s_1	**Signifikanz** s_2
Baby you can drive my Car	435.20Hz	6.3981	0.6367
Help	435.20Hz	7.2809	0.6387
I wanna hold your Hand	**441.53Hz**	7.4037	**0.0273**
Lady Madonna	435.20Hz	7.8685	0.5987
Let it be	435.20Hz	8.4871	0.4835
Ob-La-Di Ob-La-Da	435.20Hz	6.1399	0.7151
Penny Lane	435.20Hz	8.9651	0.3654
She loves you	435.20Hz	7.1442	0.5436
Yellow Submarine	435.20Hz	7.7320	0.8597
Yesterday	**434.70Hz**	8.2600	**0.0368**

Tabelle 3.3: Ermittelte Referenzfrequenzen bekannter Titel der *Beatles*

Zu der Übersicht ist anzumerken, dass bei der in der Implementierung gewählten Auflösung mit 100 Klassen der Unterschied zwischen einer Referenzfrequenz von 434.70Hz und 435.20Hz genau zwei Klassen betrifft. Da in Abbildung 3.10 zudem zu sehen ist, dass der zweithöchste Peak bei genau diesen 435.20Hz liegt, kann man vermuten, dass dies die von den *Beatles* verwendete Stimm-Frequenz ist. Beim Beispiel von *Yesterday* handelt es sich also um einen Ausreißer, bei dem die korrekte Frequenz zwar deutlich vorhanden ist, jedoch von einer – zwar ähnlichen, aber trotzdem anderen – überstimmt wird. Dass das Ergebnis zudem wenig signifikant ist, zeigt der äußerst niedrige Wert von s_2.

Ein weiterer Ausreißer ist bei *I wanna hold your Hand* zu erkennen. Auch hier sind im Histogramm zwei fast gleich starke Klassen zu erkennen, die dominieren. Dies drückt der wiederum sehr niedrige Wert s_2 aus. Die etwas weniger präsente Klasse entspricht genau den 435.20Hz der übrigen Stücke – das gelieferte Ergebnis kann somit ebenfalls als Fehler aufgefasst werden. Dass der Wert für die Signifikanz s_1 trotz zweier konkurrierender Peaks nicht deutlich geringer ist als bei anderen Songs liegt daran, dass diese Peaks einerseits nicht sehr weit auseinander liegen und andererseits die übrigen Klassen nur relativ schwach ausgeprägt sind.

3.4 Harmonic Pitch Class Profile

Das Pitch Class Profile wurde von Fujishima in [16] als Feature zur Akkord-Erkennung vorgestellt und von Gómez in [18] zur Beschreibung der Tonalität aufgegriffen. Prinzipiell handelt es sich dabei um einen Chroma-Vektor, der die zwölf Halbtöne der Oktave umfasst und für jeden beschreibt, wie stark dieser Ton im momentanen Signal präsent ist. Berechnet wird er indem alle Peaks des Spektrums auf eine Oktave zusammengefaltet werden.

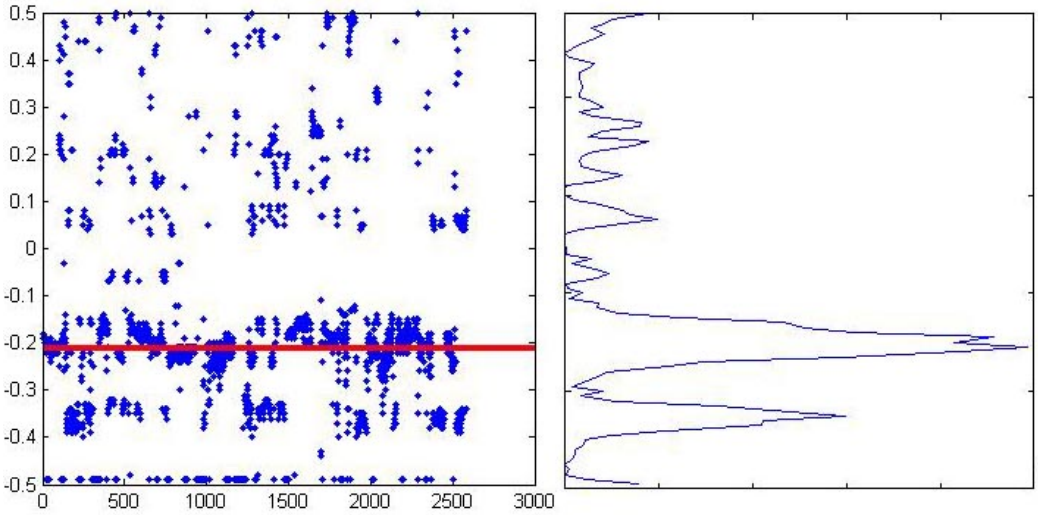


Abbildung 3.10: Einzelwerte und Gesamt-Histogramm zur Bestimmung der Referenzfrequenz am Beispiel von *Yesterday* von den *Beatles*

Als Verbesserung des ursprünglichen Features aus [16] wurden von Gómez [18] folgende Modifikationen vorgeschlagen:

Gewichtung Es gehen nicht alle Peaks mit gleichem Einfluss in die Berechnung des Feature-Vektors ein, sondern eine Gewichtungsfunktion wird auf die einzelnen Eingangswerte angewandt.

Obertöne Wird ein Ton gespielt, so klingt, wie in Abschnitt 2.2.3 beschrieben, eine ganze Reihe an Obertönen mit. Diese entsprechen zum Teil anderen Pitch-Klassen als der Grundton und verfälschen so das Ergebnis. Dieser Effekt wird bei der Berechnung berücksichtigt.

Auflösung Die Auflösung kann dahingehend verändert werden, dass man anstatt der zwölf Klassen, die einer Klasse pro Halbton entsprechen, auch mehrere Klassen auf einen Halbton kommen lassen kann. Eine Reduzierung auf unter zwölf macht hingegen keinen Sinn, da die Konzepte der Musik nicht mehr repräsentiert werden könnten.

Die angesprochene Berücksichtigung der Obertöne wird durch die geänderte Bezeichnung *Harmonic Pitch Class Profile* (HPCP) ausgedrückt. Der HPCP-Vektor ist dann definiert als

$$HPCP(n) = \sum_{i=0}^{nPeaks} w(n, f_i) \cdot a_i^2 \qquad \text{mit } n = 0...N-1 \tag{3.35}$$

Die Größe des Vektors N ist variabel, sinnvollerweise jedoch ein ganzzahliges Vielfaches von 12. Der Index n bezeichnet die einzelnen Dimensionen des Vektors und somit das jeweilige Band innerhalb der Oktave. f_i und a_i beschreiben Frequenz und Amplitude des betrachteten Peaks i. Der Term $w(n, f_i)$ stellt das Gewicht mit dem eine Frequenz f_i zum entsprechenden Band n beiträgt dar. Diese Gewichtungsfunktion soll im nächsten Abschnitt näher erläutert werden.

3.4.1 Gewichtungsfunktion

Die Gewichtungsfunktion hat zwei grundlegende Aufgaben. Einerseits ist sie dafür zuständig, die Frequenzen aller Peaks – die ja über mehrere Oktaven verstreut liegen – auf eine einzige Oktave zu falten. Andererseits sieht das HPCP-Feature aus [18] im Unterschied zu seinem Vorgänger (vgl. [16]) vor, dass Peaks nicht nur zu genau der Pitch-Klasse beitragen in die sie exakt fallen, sondern auch die benachbarten Klassen innerhalb eines bestimmten Fensters beeinflussen. Der Grund dafür ist jener, dass Peaks auch sehr nahe an der Grenze zwischen zwei Klassen liegen können. In solchen Fällen wird ihr Einfluss dann nicht auf eine Klasse beschränkt, sondern relativ gleichmäßig aufgeteilt.

Der Beitrag, den ein Peak nun zu einem Band leistet, entspricht dem Quadrat der Amplitude (in dB) und wird mit einer $\cos^2$-Funktion gewichtet. Relevant für die Gewichtung ist somit der Abstand der tatsächlichen Frequenz eines Peaks f_i (übertragen auf eine einheitliche Oktavenlage) von der Mitte des jeweiligen Chroma-Bands gemessen in Halbtönen. Es sei die Mitte des n-ten Bands

$$f_n = f_{ref} \cdot 2^{\frac{n}{N}} \tag{3.36}$$

und deren Abstand zur Frequenz des i-ten Peaks

$$d = 12 \left[\log_2 \left(\frac{f_i}{f_n} \right) + m \right] \tag{3.37}$$

Der Summand m bewirkt die Faltung auf jene Oktave, in der die Referenzfrequenz und entsprechend Formel 3.36 auch die Bänder des Feature-Vektors liegen. Es handelt sich dabei um eine ganze Zahl, die für jede Frequenz f_i so zu wählen ist, dass $\mid d \mid$ minimal

wird. Alternativ dazu kann d über Formel 3.38 berechnet werden, ohne dass ein Faktor m bestimmt werden muss.

$$d = 2\left(\left[12\log_2\left(\frac{f_i}{f_n}\right)\right] \bmod 6\right) - \left(\left[12\log_2\left(\frac{f_i}{f_n}\right)\right] \bmod 12\right) \tag{3.38}$$

Ist l die Länge des Fensters, innerhalb dessen ein Gewicht größer 0 vorhanden sein soll, so ergibt sich die vollständige Gewichtungsfunktion als

$$w(n, f_i) = \begin{cases} cos^2\left(\frac{\pi d}{l}\right) & : \text{ für } \mid d \mid \leq \frac{l}{2} \\ 0 & : \text{ für } \mid d \mid > \frac{l}{2} \end{cases} \tag{3.39}$$

In [18] wird für den Parameter l ein Wert von $\frac{4}{3}$ Halbtönen vorgeschlagen und für N – also die Größe des Featurevektors – ein Wert von 36, was einer Auflösung von 3 Bändern pro Halbton entspricht. Daraus ergibt sich, dass jeder Peak zu genau 4 verschiedenen HPCP-Bändern aber nur zur Repräsentation zweier Halbtöne beiträgt.

3.4.2 HPCP und Obertonreihen

Bei der Berechnung des HPCP-Feature wird bewusst auf das Vorhandensein von Obertönen eingegangen. Diese sind, wie bereits beschrieben, die ganzzahligen Vielfachen der Frequenz des Grundtons. Jede Zweierpotenz als Faktor bewirkt, dass der Oberton genau um eine oder mehrere Oktaven über dem Grundton liegt und somit in die selbe Pitch-Klasse fällt. Die anderen jedoch kommen in abweichenden Bändern zu liegen und beeinflussen so das Ergebnis. Die Pitch-Klasse der n-ten Oberschwingung befinden sich

$$i(n) = 2(12 \,.\, \log_2(n)) \bmod 6 - (12 \,.\, \log_2(n)) \bmod 12 \tag{3.40}$$

Bänder von jener des Grundtons entfernt. Für die ersten zwanzig Obertöne liegen die Frequenzen dabei wie in Tabelle 3.4 gezeigt um jeweils d Klassen in Bezug zur Grundfrequenz verschoben. Zur besseren Darstellung wird auch der Ton der jeweiligen Pitch-Klasse angegeben, wobei von einem c als Grundton ausgegangen wird. Weiters ist zu beachten, dass die Obertöne mit zunehmender Ordnung an Bedeutung verlieren.

Um die Auswirkungen auf die Verteilung des Feature-Vektors zu minimieren, schlägt Gómez in [18] vor, jeden Peak nicht nur für die Pitch-Klasse seiner eigene Frequenz

Frequenz	**Ton**	**d**	**Frequenz**	**Ton**	**d**
f_0	c	0	$11 \cdot f_0$	fis, ges	6
$2 \cdot f_0$	c	0	$12 \cdot f_0$	g	-5
$3 \cdot f_0$	g	-5	$13 \cdot f_0$	gis, ais	-4
$4 \cdot f_0$	c	0	$14 \cdot f_0$	ais, b	-2
$5 \cdot f_0$	e	4	$15 \cdot f_0$	h	-1
$6 \cdot f_0$	g	-5	$16 \cdot f_0$	c	0
$7 \cdot f_0$	ais, b	-2	$17 \cdot f_0$	cis, des	1
$8 \cdot f_0$	c	0	$18 \cdot f_0$	d	2
$9 \cdot f_0$	d	2	$19 \cdot f_0$	dis, es	3
$10 \cdot f_0$	e	4	$20 \cdot f_0$	e	4

Tabelle 3.4: Lage der Obertöne im HPCP-Profil

f_i einen Beitrag erzeugen zu lassen, sondern auch für all jene, deren Oberton er ist. Das sind unter Berücksichtigung von N Obertönen die Klassen der Frequenzen $\frac{f_i}{2}, \frac{f_i}{3}, \frac{f_i}{4}, \frac{f_i}{5}, ..., \frac{f_i}{N}$. Diese werden jedoch nicht mit demselben Gewicht beeinflusst, sondern umso schwächer je höher die Ordnung der Oberschwingung ist. Eine Gewichtungsfunktion kann in Anlehnung an [42] wie folgt aussehen.

$$w_{harm}(n) = s^{n-1} \tag{3.41}$$

Dabei ist sinnvollerweise $s < 1$ und abhängig von der Klangfarbe der verwendeten Instrumente zu wählen.

Das Schließen von Peak-Frequenzen auf Grundtöne ist effektiver als umgekehrt das Schwächen von Frequenzen die Obertönen entsprechen, sobald ein Grundton gefunden wurde. Im ersten Fall wird nur einmal ein Fehler begangen und zwar, wenn die betrachtete Frequenz tatsächlich die Grundschwingung war. Aber selbst in diesem Fall fällt der Fehler nicht sehr stark ins Gewicht und ist außerdem für alle Töne der gleiche. Bei der zweiten Alternative hingegen begeht man Fehler die für verschiedene Töne anders ausfallen. Denn weder weiß man, welche Obertöne genau mit einem Grundton mitschwingen, noch wie stark diese zu einem gegebenen Zeitpunkt noch klingen. Man muss also beides schätzen und bewirkt so nicht abwägbare Ungenauigkeiten im Ergebnis.

Das Resultat der Berechnungen ist für den Song *Yesterday* der *Beatles* in den Abbildungen 3.11 sowie 3.13 dargestellt. Erstere zeigt die HPCP-Vektoren des bereits in den Abbildungen 3.4 (Gesamt-Spektrum) und 3.5 (Gefilterte Peaks) dargestellten Ausschnitts. Der zweite längere kontinuierliche Block (auf Ton f) in der Mitte des Ausschnitts, bei Sekunde 61 beginnend, ist der erste Takt des Refrains. Die in den Noten

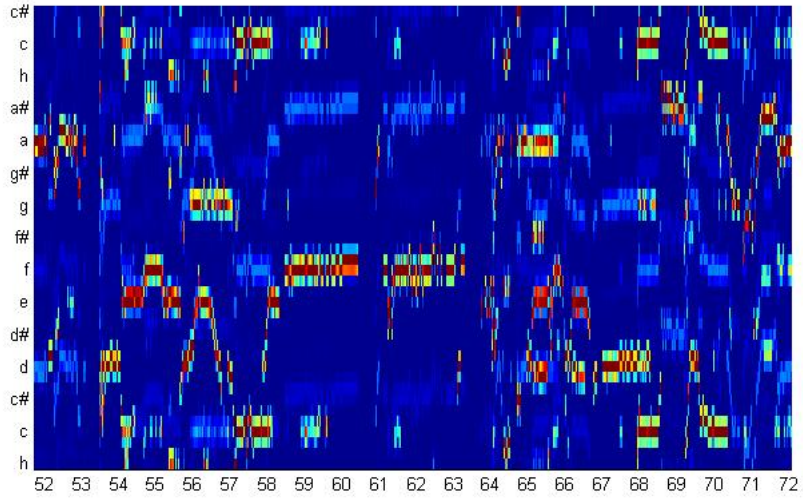


Abbildung 3.11: Ausschnitt der momentanen HPCP-Vektoren aus *Yesterday*

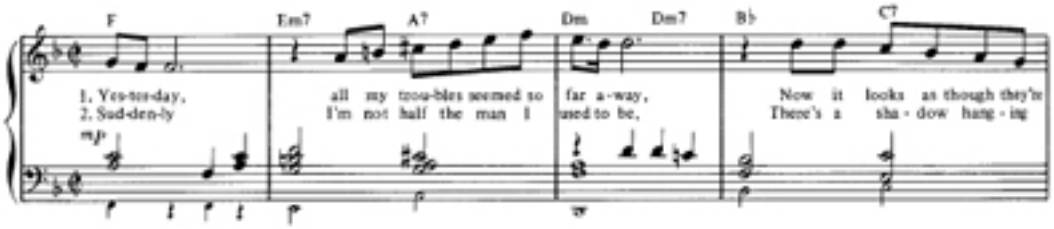



Abbildung 3.12: Ausschnitt der Melodie des Refrains von *Yesterday*

in Abbildung 3.12 erkennbare aufsteigende Melodie ist in den HPCP-Vektoren ab Sekunde 64 zu sehen und führt zur punktierten halben Note, die dem lang anhaltenden d in Sekunde 67 entspricht. Das c im vierten Takt ist ebenfalls deutlich zu sehen und genauso die absteigende Folge zurück zum g. Zudem sind in den HPCP-Vektoren jeweils die Grundtöne der Begleitakkorde – die auch in der Notenzeile aus Abbildung 3.12 vermerkt sind – gut zu erkennen.

Abbildung 3.13 zeigt die Summe aller HPCP-Vektoren. Besonders deutlich sieht man die Dominanz des Tons f sowie in weiterer Folge des F-Dur Dreiklangs. Als einziger natürlicher Ton ist h schwächer ausgeprägt als das benachbarte b. Anhand der HPCP-Summe kann man somit auf die Tonart F-Dur, die genau dieses eine Vorzeichen (H → B) besitzt schließen. Da dies die tatsächliche Tonart des Stückes ist, kann vom HPCP-Feature eine gewisse Signifikanz angenommen werden. Eine genaue Evaluierung wäre nur mit äußerst großem Aufwand möglich, wenn man Referenzwerte von einem genauen Notenblatt oder einem der Aufnahme exakt entsprechenden Midi-File herleitet.

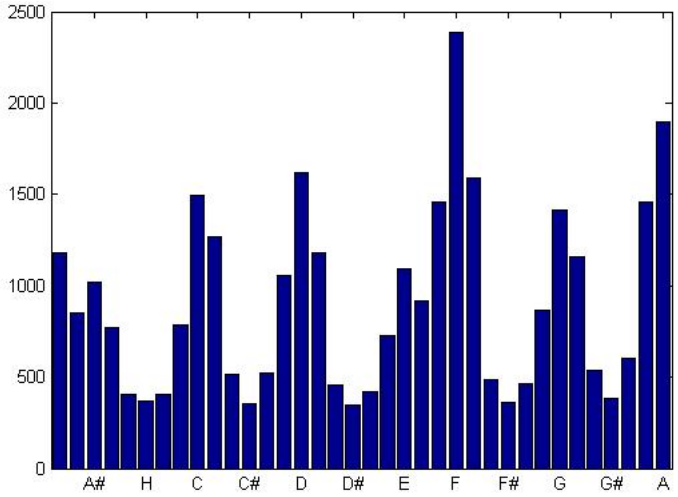


Abbildung 3.13: Summe aller einzelnen HPCP-Vektoren von *Yesterday*

3.4.3 Normalisierung

Um das Feature unabhängig von der absoluten Lautstärke des Musikstücks zu machen, werden zwei Normalisierungsschritte angewandt. Zum einen das so genannte *spectral whitening*, bei dem das gesamte Spektrum in einem Vorverarbeitungsschritt entsprechend seiner spektralen Hülle in ein flaches Spektrum konvertiert wird. Dies hat zur Folge, dass Töne in hohen Oktaven gleich stark zum HPCP-Vektor beitragen wie solche in den tieferen Tonlagen. Das Resultat hängt also nicht davon ab, welche Frequenzbänder aufgrund der Musikrichtung besonders betont werden (vgl. [18]).

Der zweiter Normalisierungsschritt erfolgt in der Nachbearbeitung. Dabei werden alle Vektoren der einzelnen Frames so skaliert, dass der Maximalwert der einzelnen Pitch-Klassen jeweils 1 beträgt.

$$HPCP_{norm}(n) = \frac{HPCP(n)}{\max(HPCP)} \tag{3.42}$$

3.5 Transposed Harmonic Pitch Class Profile

Das HPCP-Feature beschreibt, welche Töne des musikalischen Tonsystems welchen Beitrag zum gesamten Klangbild eines Stücks leisten. Dieses Merkmal in Form eines Vektors ist jedoch nicht robust gegenüber Transpositionen in andere Tonarten. Dies würde

einem ringförmigen Shiften der einzelnen Werte entsprechen. Derartige Transpositionen sind in der Musik – speziell bei Cover-Versionen von Musikstücken – sehr verbreitet, um das entsprechende Stück an den Tonumfang der verwendeten Instrumente bzw. die Stimmlagen der Sänger anzupassen. Dabei bleiben alle Intervalle und somit eine sehr starke Korrelation der Wahrnehmung erhalten.

Dies soll sich auch im Feature niederschlagen. Dazu wird in [18] vorgeschlagen, alle HPCP-Vektoren in eine quasi einheitliche Tonart zu transponieren und erst diese modifizierten Feature-Werte zur Analyse heranzuziehen. Das daraus resultierende *Transposed Harmonic Pitch Class Profile* (THPCP) ist definiert als

$$THPCP(n) = HPCP([n - n_{shift}] \bmod N) \tag{3.43}$$

Dabei ist N die Dimensionalität des Feature-Vektors und n_{shift} die Anzahl an HPCP-Bändern, um die die jeweilige Tonart des Stücks von der (beliebigen, aber fixen) Standardtonart abweicht.

3.5.1 HPCP für Framegruppen

Im Gegensatz zum HPCP-Feature, das, wie in Abschnitt 3.4 erläutert wurde, jeweils einen einzelnen Frame beschreibt, ist oft erwünscht, größere Segmente bzw. das ganze Musikstück zu charakterisieren. In diesem Fall müssen statistische Verfahren angewandt werden, um die Einzelwerte zu akkumulieren. Von besonderer Bedeutung ist hier das arithmetische Mittel

$$\overline{HPCP(n)} = \frac{1}{i_{end} - i_{start} + 1} \cdot \sum_{i=i_{start}}^{i_{end}} HPCP_i(n) \tag{3.44}$$

Je nachdem wie groß die Länge $l = i_{end} - i_{start} + 1$ des Segments gewählt wird, kann entweder der aktuell gespielte Akkord oder die lokale Tonart (bei Modulationen innerhalb des Stücks) bzw. die globale Haupttonart des Stücks abgeschätzt werden.

3.5.2 Bestimmung der Tonart

In Formel 3.43 ist der Term n_{shift} der Abstand der Tonart des Stücks zur gewählten Standardtonart. Geht man nicht von einem Notenblatt, sondern einer Audiodatei als

Eingabe aus, so ist ohne das Heranziehen zusätzlicher Metadaten die Tonart des vorliegenden Musikstücks nicht bekannt. In diesen Fällen muss sie, um n_{shift} zu berechnen, erst aus den vorhandenen Daten bestimmt werden.

Um die Tonart zu abzuschätzen, wird im ersten Schritt ein globaler $\overline{HPCP}$-Vektor aufgestellt. Dieser wird dann mit einer Matrix an möglichen Tonart-Profilen verglichen. Die beste Übereinstimmung ergibt die geschätzte Tonart. Zum Vergleich werden die in Abschnitt 2.2.2 beschriebenen Profile für Dur- bzw. Moll-Tonarten, jeweils aufbauend auf allen der zwölf Halbtöne, herangezogen. Weniger weit verbreitete Systeme, wie etwa Kirchentonarten oder gar Tonleitern mit feineren Abstufungen als Halbtonschritten, wie sie in gewissen Musikrichtungen vorkommen, werden nicht berücksichtigt. Im Kontext aktueller Main-Stream Pop- und Rock-Musik bzw. ähnlicher Stile stellt dies jedoch kein Problem dar, da derartige Skalen so gut wie nie verwendet werden.

Als Maß für die Ähnlichkeit wird laut [18] die Korrelation zwischen dem Referenzprofil und dem tatsächlichen $\overline{HPCP}$-Vektor berechnet. Sowohl für Dur als auch für Moll erhält man dabei N Werte. Diese entsprechen einem Vergleich pro HPCP-Band als Grundton. Bezeichnet $K(i)$ das um i Bänder geshiftete Tonartenprofil, so errechnen sich die Werte als

$$\begin{aligned} R_{Dur}(i) &= r(\overline{HPCP}, K_{Dur}(i)) \\ R_{Moll}(i) &= r(\overline{HPCP}, K_{Moll}(i)) \end{aligned}$$

Die Korrelation $r(x, y)$ ist dabei gegeben als

$$r(x,y) = \frac{E[(x-\mu_x) \cdot (y-\mu_y)]}{\sigma_x \cdot \sigma_y} = \frac{cov(x,y)}{\sigma_x \cdot \sigma_y} \tag{3.45}$$

Neben seiner Bedeutung zum Schätzen der tatsächlichen Tonart gibt der jeweilige Maximalwert aus R zudem an, wie stark die entsprechende Tonart präsent ist. Ein sehr hoher Wert deutet darauf hin, dass sie das ganze Stück über dominant ist. Ein Wert, der sich nicht wesentlich vom zweitgrößten abhebt, ist hingegen ein Indiz dafür, dass im Stück Modulationen vorkommen und Teile in anderen Tonarten gespielt sind. Dieses Verfahren ist auch unter dem Namen *Krumhansl-Schmuckler Algorithmus* bekannt. Es basiert auf dem Tonart-Bestimmungs Algorithmus aus [45] und wurde in [24] bezüglich der verwendeten Tonleiter-Modelle angepasst. Diese wurden bereits in Abschnitt 2.2.2 vorgestellt.

Die Methode geht ursprünglich von unbegleiteten Melodien als Eingangsdaten aus. Die Annahme, dass ein Musikstück nur aus einer bzw. mehreren Melodiestimmen besteht, ist jedoch eine zu starke Vereinfachung. Zum einen sind in den meisten Musikstücken Melodien mit Akkorden hinterlegt – zum anderen spielen auch Schlagwerke und perkussive Instrumente eine Rolle, wenn nicht Notenblätter sondern die tatsächlichen Spektren betrachtet werden.

Um dies zu berücksichtigen schlägt Gómez [18] vor, die ursprünglichen Tonsystem-Modelle dahingehend zu modifizieren, dass sie das Vorhandensein von Akkordfolgen berücksichtigen. Dazu wird ein beliebiges melodie-basiertes Modell betrachtet und unter der Annahme, dass die Töne nicht alleine angespielt werden sondern immer als Dreiklang, wird das neue Modell konstruiert. Die 5. Stufe einer Tonart ist beispielsweise in den Dreiklängen mit den Grundtönen auf 1., 3. und 5. Stufe enthalten. Wenn $T(i)$ die relative Dominanz der i-ten Stufe einer Tonart beschreibt, so ist gemäß obigem Beispiel der Wert für die 5. Stufe im modifizierten Modell

$$T_{polyphon}(5) = T(1) + T(3) + T(5)$$

Verallgemeinernd kann man sagen, dass alle Pitch-Klassen aus dem ursprünglichen Modell mit einem bestimmten Gewicht α zu jenen des polyphonie-basierten Modells beitragen.

$$T_{polyphon}(i) = \sum_{j=0}^{11} \alpha(i,j) \cdot T(j) \tag{3.46}$$

bzw. in Matrixschreibweise

$$T_{polyphon} = \alpha \cdot T^t \tag{3.47}$$

Laut den Versuchen in [18] bewährt es sich, nur die Dreiklänge auf den Hauptstufen zu berücksichtigen. Das sind die 1. (Tonika), die 4. (Subdominante) und die 5. Stufe (Dominante). Die entsprechenden Matrizen α_{Dur} und α_{moll} können auf dieser Annahme basierend leicht aufgestellt werden.

Ähnlich zum Vorliegen von Akkorden anstatt einzelner Töne, kann auch das Vorhandensein von Oberschwingungen in das Profil miteinbezogen werden. In Formel 3.40 wird berechnet in welcher Pitch-Klasse die n-te Oberschwingung liegt. Diese Obertöne

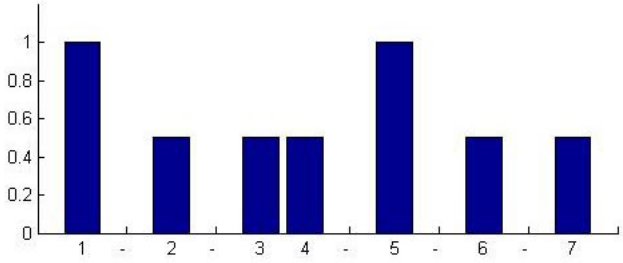


Abbildung 3.14: Modifiziertes Modell anhand des flachen Dreiklangsmodells für Dur

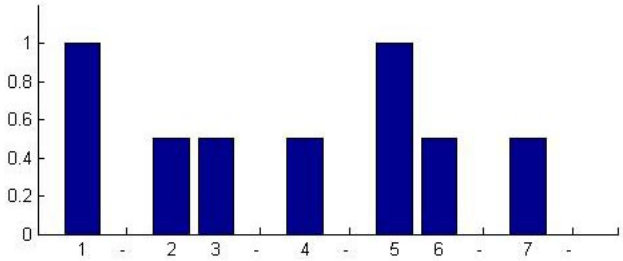


Abbildung 3.15: Modifiziertes Modell anhand des flachen Dreiklangsmodells für Moll

können somit in die Matrizen α_{Dur} und α_{moll} mit eingetragen werden. Sinn macht es jedoch, sie nicht gleich stark wie die Grundtöne zu gewichten, sondern das Gewicht entsprechend der Formel $w_{harm}(n) = s^{(n-1)}$ mit $s < 1$ zu wählen.

Die Frage, ob es ausreicht, nur einen kurzen Ausschnitt eines Stücks zu betrachten, oder ob es besser ist, die gesamte vorhandene Information zu verwenden, wird ebenfalls in [18] beantwortet. Laut diesen Untersuchungen wird das beste Ergebnis bei Betrachtung des gesamten Stücks erzielt. Nicht allzu weit dahinter liegt die Genauigkeit, bei Berücksichtigung der ersten oder letzten 30 Sekunden. Wählt man jedoch Ausschnitte aus der Mitte, so ist das durchschnittliche Ergebnis deutlich schlechter. Dies steht im Einklang mit der Musiktheorie, die davon ausgeht, dass ein Stück mit der Haupttonart beginnt und endet, während Variationen, und damit oft auch Modulationen, vorwiegend in den Mittelteilen zu finden sind.

Eine andere Modifikation des Krumhansl-Schmuckler Algorithmus stellen Madsen und Widmer [27] mit ihren Intervall-Profilen vor. Der Grundgedanke dabei ist, die zeitliche Abfolge von Noten – die bei dem HPCP-Feature wegabstrahiert ist – mitzuberücksichtigen. Das daraus abgeleitete Modell besteht aus zwei 12x12-Matrizen für Dur und Moll, die für die Pitch-Klasse jedes Tons die Wahrscheinlichkeiten der Klassen aller möglichen Folgetöne angeben. Zudem macht es insbesondere für Moll Sinn, unterschiedliche Profile für auf- bzw. absteigende Tonfolgen zu verwenden (melodisches Moll, vgl. Abschnitt 2.2.2).

Gewonnen wird das Modell durch Zählen aller Intervalle anhand von Trainingsdaten. Diese werden dabei auf einen Grundton c normiert um eine einheitliche Basis zu erhalten. Auf- bzw. absteigende Intervalle werden getrennt voneinander gezählt. Das gleiche Verfahren wird in weiterer Folge auf ein zu bewertendes Stück angewandt. Das erhaltene Eingabe-Profil wird dann mit den aus den Trainingsdaten erhaltenen Profilen für Dur und Moll bzw. allen durch Shiften des Grundtons daraus ableitbaren verglichen. Ausschlaggebend für die geschätzte Tonart ist die höchste Korrelation laut

$$cor(P_i, D_i) = \sum_j \sum_k P_i(j,k) D_i(j,k) \tag{3.48}$$

wobei P_i die gelernte Intervall-Verteilung und D_i das Profil des aktuellen Eingabestücks ist (vgl. [27]).

Die dabei erzielten Resultate sind mit jenen die bei Anwendung verschiedener PCP-basierter Modelle erreicht werden vergleichbar. Zusätzliches Potenzial liegt jedoch in der Kombination beider Ansätze. Der beschriebene Algorithmus hat aber den Nachteil, dass einzelne Stimmen bekannt und voneinander getrennt sein müssen. Davon kann man etwa bei MIDI-Dateien ausgehen, nicht jedoch bei polyphonen Audiosignalen. Aus diesem Grund wird in dieser Arbeit zur Tonartbestimmung auf den Krumhansl-Schmuckler Algorithmus alleine zurückgegriffen. Abbildung 3.16 zeigt dabei die Tonart-normalisierte Version des in Abbildung 3.13 dargestellten Features von *Yesterday* von den *Beatles*. Gut zu erkennen ist die Dominanz des Grundtons sowie der leitereigenen Töne gegenüber den Zwischenstufen.

Weitere Ergebnisse der Tonarterkennung sind in Tabelle 3.5 aufgelistet. Als Referenz wird dazu die Analyse vieler *Beatles*-Songs in [39] herangezogen. Die angegebene Signifikanz eines Ergebnisses wurde im Vergleich zum zweit-wahrscheinlichsten berechnet als

$$s = 100 \cdot \left(1 - \frac{corr(key_2, hpcp)}{corr(key_1, hpcp}\right) \tag{3.49}$$

Aus Tabelle 3.5 wird ersichtlich, dass bei 5 von 10 Titeln die Tonart exakt ermittelt wurde. Bei insgesamt 24 möglichen Tonarten liegt diese Erkennungsrate von 50% weit über jener von rund 4%, die durch reines Raten zu erwarten wäre. Dazu kommt, dass bei zwei weiteren Stücken zwar der Grundton richtig erkannt, jedoch jeweils Moll anstatt Dur geschätzt wurde. Auch bei *Drive my Car* ist der Fehler sehr klein. Hier wurde G-Dur anstatt D-Dur berechnet. Der Unterschied zwischen diesen beiden Tonarten ist im Wesentlichen, dass D-Dur ein Vorzeichen mehr besitzt – ein zusätzliches cis. Betrachtet

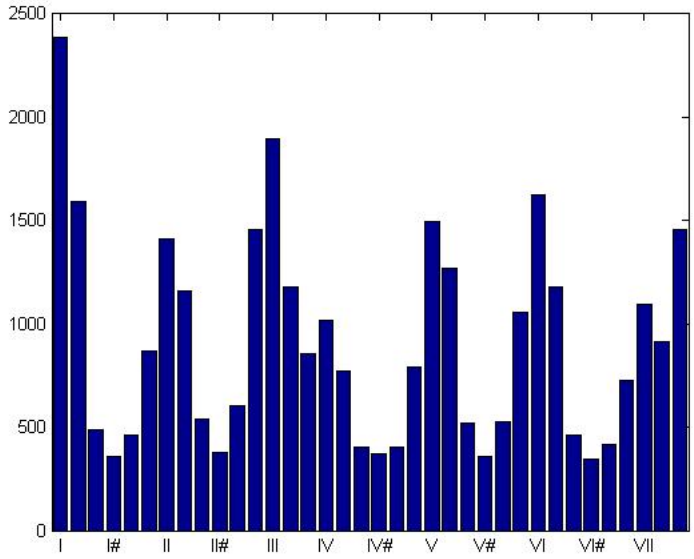


Abbildung 3.16: Summe aller THPCP-Vektoren des Stücks *Yesterday* von den *Beatles*

man die Noten des Stücks, so sieht man, dass dieses cis so gut wie immer aufgelöst und stattdessen ein c gespielt wird. Der verwendete Tonumfang entspricht also tatsächlich eher der geschätzten G-Dur.

Obwohl sie nur auf einer sehr kleinen Musiksammlung getestet wurde, lässt sich feststellen, dass die im Rahmen dieser Arbeit entstandene Implementierung nicht die 83% exakt erkannter Tonarten, die etwa von Gómez ([18]) erreicht wurden, liefert. Da die verwendeten Methoden jedoch sehr ähnlich sind kann dieser Unterschied teilweise darin

Titel	Tonart		**Signifikanz**
	geschätzt	**tatsächlich**	
Baby you can drive my Car	D-Dur	G-Dur	0.4459
Help	A-Dur	A-Dur	0.6152
I wanna hold your Hand	G-Dur	G-Dur	1.2999
Lady Madonna	a-Moll	A-Dur	2.2359
Let it be	C-Dur	C-Dur	10.0459
Ob-La-Di Ob-La-Da	c-Moll	B-Dur	3.2018
Penny Lane	h-Moll	H-Dur	12.0761
She loves you	G-Dur	G-Dur	2.6250
Yellow Submarine	C# -Dur	G-Dur	0.2249
Yesterday	F-Dur	F-Dur	2.9575

Tabelle 3.5: Ermittelte Tonarten bekannter Titel der *Beatles*

begründet liegen, dass hier Originalaufnahmen und nicht gesampelte MIDI-Dateien als Grundlage dienten, wodurch das Klangbild unregelmäßiger wird.

3.6 Melodieerkennung

Als zweites Feature neben den THPCPs soll die Melodie selbst und ihre Eignung zur Coverversions-Bestimmung untersucht werden. Sie ist ein charakteristisches Merkmal jedes Musikstücks und bleibt beim Zuhörer meist leicht in Erinnerung. Da zwei Versionen eines Songs immer auf der gleichen zugrunde liegende Melodie aufbauen, liegt es nahe, sie zur Klassifizierung heranzuziehen.

Ein ähnliches Forschungsthema, das sich mit Melodien und deren Reproduktion durch den Hörer befasst ist *Query-By-Humming* (vgl. beispielsweise [12]). Dabei geht es um Systeme, die ein Musikstück – von dem Interpret und Titel dem Benutzer nicht bekannt sind – anhand eines gesummten Melodieausschnitts aus einer Datenbank abfragen. Die gesummte Melodie ist dabei im weitesten Sinne auch eine eigene Version des Songs.

Während es bei MIDI-Dateien noch relativ einfach ist, einzelne Stimmen zu unterscheiden und Hauptstimmen, die die Melodie enthalten zu identifizieren, ist dies bei reinen Audio-Daten weitaus komplexer. Neben den Frequenzen, die die eigentlichen Stimmen darstellen, sind im Spektrum noch solche von Schlagwerken und perkussiven Instrumenten, Obertonreihen sowie Rauschen vorhanden. Dennoch existieren zahlreiche Ansätze, die sich mit dem Extrahieren der Melodie aus Audiosignalen beschäftigen (vgl. [15], [20], [28] und [30]).

Ausgangsbasis sind dazu in der Regel die in Abschnitt 3.2 vorgestellten Methoden zur Berechnung des Spektrums eines Signals sowie Vor- und Nachbearbeitungsschritte wie Übergangserkennung, Peak-Filterung oder Normierungen.

3.6.1 PreFEst

Einen viel zitierten Ansatz zur Extraktion der Basisfrequenz von Melodie- und Basstimme liefert Goto [20]. Anders als frühere Methoden, die sich hauptsächlich auf einstimmige Musikstücke bzw. solche, die von maximal drei Instrumenten gespielt werden, konzentrierten, geht das Verfahren bewusst auf den Kontext der Popularmusik und handelsüblicher CD-Aufnahmen ein. Der Name *PreFEst* steht dabei für *Predominant F0 Estimation*. *F0* bezeichnet dabei eine Grundschwingung und soll ausdrücken, dass diese gefunden und Obertöne eliminiert werden sollen. Wesentlich sind dabei folgen-

de drei Annahmen, die für die meisten Musikrichtungen jedoch keine Einschränkung darstellen.

- Melodie- und Bass-Töne werden – wie in Abschnitt 2.2.3 beschrieben – jeweils von einer Reihe an Obertönen begleitet.

- Die Bassstimme ist in den tiefen Frequenzbereichen zu finden; dagegen bewegt sich die Melodie in den mittleren bis hohen Bändern. Die Grenze ist unabhängig von den angewandten Algorithmen und durch einen entsprechenden Parameter festzulegen.

- Sowohl Melodie- als auch Bassstimme haben vorwiegend kontinuierliche Verläufe über die Zeit. An eine Melodie-Note knüpft eine andere an – ebenso folgt im Signal ein Melodie-Ton direkt auf den vorherigen. Pausen sind die Ausnahme.

3.6.1.1 Augenblicksfrequenzen und Kandidaten

Grundlage für alle weiteren Schritte der Methode ist die Transformation des Audiosignals in den Frequenzbereich anhand der STFT und einer Filterbank mit unterschiedlichen Frequenz-/Zeit-Auflösungen. Der dadurch erzielte Effekt ist mit jenem, der in Abschnitt 3.2.5 beschriebenen, Constant Q Transformation zu vergleichen. Die Frequenzauflösung nimmt nach oben hin ab – so werden Bereiche, die das menschliche Gehör unschärfer wahrnimmt weniger genau erfasst als die unteren Frequenzbänder, bei denen der Mensch auch geringe Unterschiede gut erkennen kann. Der eigentliche Vorteil liegt in der Minimierung der Daten bei ausreichender Frequenzauflösung in allen Bereichen.

Anhand des Spektrums werden im nächsten Schritt die von Abe et al. in [1] vorgeschlagenen Augenblicksfrequenzen $\lambda(\omega, t)$ gemäß Formel 3.50 errechnet. Das Spektrum als Ergebnis der STFT ist aufgrund von Unschärfen durch die verwendete Fensterfunktion (vgl. Abschnitt 3.2.3) verschmiert. Gegen die somit relativ grobe Zeit-/Frequenzauflösung soll das IF *(Instataneous Frequency)* Spektrum Abhilfe schaffen, um Obertöne genau abgrenzen und identifizieren zu können.

$$\begin{aligned} \lambda(\omega, t) &= \frac{\partial}{\partial t} \arctan\left(X(\omega, t) e^{j\omega t}\right) \\ &= \omega + \frac{a\frac{\partial b}{\partial t} - b\frac{\partial a}{\partial t}}{a^2 + b^2} \end{aligned} \tag{3.50}$$

a und b stehen dabei für den Real- bzw. Imaginärteil $(\Re(X), \Im(X))$ des komplexen Spektrums X des Signals $x(t)$. Die Augenblicksfrequenzen drücken somit die momentanen Änderungsraten der Phase aus. Diese ist bei stabilen Sinusoiden gleich der Frequenz ω der Bandmitte und vergrößert sich in ihrer Abweichung umso mehr, je schwankender das Signal ist.

Mit Hilfe der Augenblicksfrequenzen werden in weiterer Folge stabile Kandidaten für Melodie und Bass aus dem Spektrum gefiltert. Die entsprechende Menge $\Psi_f^{(t)}$ ergibt sich als

$$\Psi_f^{(t)} = \{\psi \mid \lambda(\psi, t) - \psi = 0, \frac{\partial(\lambda(\psi, t) - \psi)}{\partial\psi} < 0\} \tag{3.51}$$

Die Energie Verteilungsfunktion der Frequenzen folgt daraus entsprechend folgender Formel.

$$\Psi_p^{(t)}(\omega) = \begin{cases} \mid X(\omega, t) \mid & : \quad \text{wenn } \omega \epsilon \Psi_f^{(t)} \\ 0 & : \quad \text{sonst} \end{cases} \tag{3.52}$$

Sind alle möglichen Kandidaten ermittelt, werden sie in zwei Gruppen getrennt, je nachdem ob sie für die Melodie- oder die Bass-Stimme in Frage kommen. Im ursprünglichen Verfahren aus [20] wird eine Frequenz von 261.6Hz, was dem Ton c' entspricht, als Schwellwert angenommen. Die Energieverteilungsfunktion wird nun für beide Gruppen einerseits auf die cent-Skala übertragen und andererseits so normiert, dass die Summe der Energien aller Frequenzen eines Frames stets 1 beträgt. Das Resultat zu einem bestimmten Zeitpunkt t wird in der Folge mit $p_\Psi^{(t)}(x)$ bezeichnet.

3.6.1.2 Grundtöne

Die zentrale Annahme ist nun jene, dass diese beobachtete Energieverteilung von einer Menge an gewichteten Grundfrequenzen und deren Obertönen erzeugt wurde. Ziel ist es, alle Oberschwingungen zu eliminieren und lediglich die Grundtöne zu erhalten. Als Ausgangspunkt dazu dient ein statistisches Tonmodell. Die Wahrscheinlichkeit, dass eine Frequenz x durch den Grundton F erzeugt wird, entspricht $p(x \mid F)$. Das Gesamtsi-

gnal besteht aus einer gewichteten Überlagerung an Grundtönen aus einem bestimmten Intervall

$$\theta^{(t)} = \{w^{(t)}(F) \mid F_{low} \leq F \leq F_{high}\} \tag{3.53}$$

$w^{(t)}(F)$ drückt dabei die momentane Gewichtung einer Grundfrequenz zum Zeitpunkt t aus. Sie ist derart normiert, dass das Integral über $\theta^{(t)}$ stets 1 ergibt. Daraus folgt die Dichtefunktion für das Gesamtmodell laut Formel 3.54

$$p(x; \theta^{(t)}) = \int_{F_{low}}^{F_{high}} w^{(t)}(F) p(x \mid F) dF \tag{3.54}$$

Da weder die Anzahl der simultan klingenden Grundtöne noch deren Lage bzw. Intensität a priori bekannt sind, muss der Parameter $\theta^{(t)}$ für jeden Frame t so geschätzt werden, dass das daraus generierte Modell die beobachteten Daten $\Psi_{\Psi}^{(t)}(x)$ möglichst gut beschreibt.

Der verwendete Maximum-Likelihood Schätzer wird dabei durch Maximierung der *Mean Log-Likelihood* bestimmt. In diesem Fall entspricht dies

$$\int_{-\infty}^{\infty} p_{\Psi}^{(t)}(x) \log p(x; \theta^{(t)}) dx \tag{3.55}$$

Da eine analytische Lösung zu komplex wäre, bedient sich PreFEst eines *Expectation Maximization (EM)* Algorithmus nach [13]. Dieser durchläuft iterativ jeweils einen E- (expectation) und einen M-Schritt (maximization) bis das Ergebnis $\overline{\theta^{(t)}}$ den tatsächlichen Wert ausreichend approximiert. Um das Verfahren zudem zu beschleunigen wird unter der Annahme, dass einzelne Töne längere Zeit stabil bleiben, als Initialwert jeweils das Ergebnis des vorherigen Frames herangezogen.

Als versteckte Variable wird F eingeführt. F beschreibt – wie in Formel 3.54 angedeutet – welcher Grundton und dessen Tonmodell für das Generieren einer beobachteten Frequenz verantwortlich waren. Die beiden Teilschritte jeder Iteration laufen unter diesen Voraussetzungen wie folgt ab.

1. Expectation – E-Schritt:
   Die bedingte Schätzung des Mean Log-Likelihood wird berechnet unter Voraussetzung des entsprechenden Werts aus der vorigen Iteration.

$$Q(\theta^{(t)} \mid \theta'^{(t)}) = \int_{-\infty}^{\infty} \int_{F_{low}}^{F_{high}} p_{\Psi}^{(t)}(x) p(F \mid x; \theta'^{(t)}) \log p(x, F; \theta^{(t)}) dF dx \tag{3.56}$$

2. Maximization – M-Schritt:
   $Q(\theta^{(t)} \mid \theta'^{(t)})$ wird als Funktion von $\theta^{(t)}$ maximiert.

Durch Auflösen der Formeln unter Zuhilfenahme aller bisherigen Gleichungen, der Euler-Lagrangen Differenzialgleichung und des Theorems von Bayes ergibt sich für die neue Schätzung $\overline{\theta^{(t)}}$ und somit für die Gewichtungsfunktion $\overline{w^{(t)}(F)}$ ein Wert von

$$\overline{w^{(t)}(F)} = \int_{-\infty}^{\infty} p_{\Psi}^{(t)}(x) \frac{w'^{(t)}(F) p(x \mid F)}{\int_{F_{low}}^{F_{high}} w'^{(t)}(\eta) p(x \mid \eta) d\eta} dx \tag{3.57}$$

Das einzige, was somit zu klären bleibt, ist das zu verwendende Tonmodell $p(x \mid F)$, das beschreibt, mit welcher Wahrscheinlichkeit ein Grundton F zu einer beliebigen Frequenz x beiträgt. Goto [20] schlägt dazu ein an der Gauss-Funktion $G(x; \mu, \sigma)$ orientiertes Modell vor.

$$p(x \mid F) = \alpha \sum_{h=1}^{N_i} c(h) G(x; F + 1200 \log_2 h, W_i) \quad \text{mit} \tag{3.58}$$

$$c(h) = G(h; 1, H_i) \tag{3.59}$$

α ist dabei eine Normierungskonstante und N beschreibt die Anzahl der Obertöne, die berücksichtigt werden sollen. W_i entspricht der Standardabweichung der Gauss'schen Normalverteilung – je größer dieser Parameter gewählt wird, desto breiter ist der Einfluss einer Frequenz auf das Spektrum. Ähnliches gilt für die Gewichtungsfunktion der Obertöne $c(h)$. Sie wird durch den Parameter H_i bestimmt. Je größer er angesetzt wird, desto mehr fallen Obertöne ins Gewicht. Der Index i deutet die Unterscheidung zwischen Bass- und Melodie-Bereich an. Für beide sind die Parameter einzeln anzupassen.

3.6.1.3 Trajektorien

Um nun tatsächlich Melodie und Bass aus der errechneten Überlagerung gespielter Töne zu extrahieren, werden die zu einem Zeitpunkt jeweils dominantesten und stabilsten Grundton-Trajektorien herangezogen. Derartige Trajektorien erzeugt ein Multi-Agent-System aus den vorliegenden Daten. Jeder einzelne Agent generiert dabei genau eine Trajektorie durch Verbinden mehrerer zeitlich benachbarter hervorstechender Peaks. Dabei werden alle Frames nacheinander laut folgendem Schema abgearbeitet.

1. Ein Peak-Detektor erkennt hervorstechende Peaks, die über einem festgelegten Schwellwert relativ zum Maximalwert liegen. Die aktiven Agenten bestimmen durch Interaktion, welchem ein solcher Peak am nähesten liegt und somit am besten zuzuordnen ist. Erheben mehr als ein Agent Anspruch auf einen Peak, so wird er dem verlässlichsten zugerechnet. Beansprucht keiner der Agenten den Peak, so wird ein neuer erzeugt um ihn zu verfolgen.

2. Jeder Agent hat einen Penalty-Wert, der erhöht wird, sobald in einem Frame kein Peak gefunden wird um seine Trajektorie fortzusetzen. Findet der Agent in einem der darauf folgenden Frames wieder einen passenden Peak, so wird die Penalty auf 0 zurückgesetzt. Erreicht dieser Wert jedoch eine bestimmte Grenze, dann terminiert der Agent und die Trajektorie endet.

3. Jeder Agent bestimmt seine eigene Verlässlichkeit. Dieser Wert ermittelt sich daraus, wie stark sich die Energien der verfolgten Peaks jeweils von jenen ihrer Nachbarfrequenzen abheben. Die Verlässlichkeit einer Trajektorie gemeinsam mit ihrer Gesamtenergie gibt schließlich den Ausschlag dafür, welche als Melodie- bzw. Bass-Stimme zu einem bestimmten Zeitpunkt ausgewählt wird.

Beim schrittweisen Durchlaufen aller Frames bietet es sich an, am Ende des Musikstücks zu beginnen und nach vorne hin weiterzuarbeiten. Der Grund dafür ist, dass zu den Taktzeiten Schlagwerke und perkussive Instrumente sehr viel Rauschen erzeugen. Es sind also tendenziell viele Peaks vorhanden und es ist nicht klar, welche zum Fortsetzen bzw. Beginnen der Trajektorien die geeignetsten sind. Startet man vom Schluss des Songs, so werden zuerst die ausklingenden Töne behandelt. Verfolgt man diese dann rückwärts bis zu ihrem Anstoß, so ist es bedeutend einfacher, sinnvolle Trajektorien zu generieren.

3.6.1.4 Erweiterungen

Zu dem vorgestellten Verfahren entwickelte Goto [21] noch Erweiterungen. Grund dafür sind im Wesentlichen die Einschränkungen, dass einerseits a priori Information über den ungefähren Melodieverlauf nicht genutzt werden konnte und andererseits, dass ein einziges quasi ideales Tonmodell verwendet wurde. Dies entspricht nicht der Realität in der, bedingt durch unterschiedlichen Klangfarben, verschiedene Obertonstrukturen vorherrschen.

Um die angesprochenen Punkte zu verbessern, wurden drei Erweiterungen zum PreFEst Ansatz hinzugefügt.

- Mehrere Tonmodelle
  Für jede Grundfrequenz wird nicht ein einziges Tonmodell sondern mehrere aufgestellt und deren Überlagerung betrachtet.

- Schätzen der Gestalt der Tonmodelle
  Für jedes Tonmodell wird die relative Energie jedes Obertons durch einen EM Algorithmus ebenso geschätzt wie die Gewichtung der einzelnen Tonmodelle untereinander.

- Einführen einer a priori Verteilung
  Anstatt der Maximum-Likelihood Schätzung wird nun eine auf Basis der *Maximum A Posteriori Probability (MAP)* verwendet. Diese erlaubt die Berücksichtigung einer a priori Verteilung der Tonmodelle.

Die Experimente in [21] zeigten eine deutliche Verbesserung gegenüber dem ursprünglichen Verfahren aus [20].

3.6.2 Weitere Verfahren zur Melodieextraktion

Eine Methode, die stark an PreFEst angelehnt ist, ist die von Marolt in [29] vorgestellte zur Melodieextraktion anhand klangfarblicher Ähnlichkeiten. Die Bestimmung und Gewichtung der Grundfrequenzen läuft analog zum PreFEst-Ansatz ab. Beim Verfolgen der Trajektorien wird jedoch, ausgehend von einem gefundenen Peak, immer ein 50 ms langes Fenster im Voraus als Ganzes betrachtet. Die weiterverfolgten Frequenzen werden somit in einem größeren Kontext bewertet und die Methode im gesamten ausreißerunempfindlicher.

Innerhalb solcher 50 ms Abschnitte sucht der Algorithmus dann nach stabilen Fragmenten. Das Extrahieren von Melodiestimmen besteht in der Folge aus dem Gruppieren passender aufeinanderfolgender Fragmente. Dabei wird in mehreren Schritten vorgegangen.

1. Fragmente mit stark ausgeprägter Obertonreihe werden herausgefiltert. Ein zweiter Filter eliminiert alle jene Fragmente, deren relative Lautheit unter einem bestimmten Schwellwert liegt.

2. Auf diese Art ausgewählte signifikante Fragmente werden entsprechend ihrer Ähnlichkeit geclustert. Diese ergibt sich anhand mehrere Features. Unter anderem sind das die Durchschnittsfrequenz, die durchschnittliche Lautstärke, das Verhältnis ungerader zu geraden Obertönen bzw. die Unregelmäßigkeiten in den Obertönen, die Cepstrum Koeffizienten oder die Dominanz der erzeugenden Tonmodelle im Verhältnis zum Gesamtklangbild. Das Ergebnis sind zwei bis fünf Cluster, die anhand des *k-means* Algorithmus gebildet werden.

3. Gerichtete azyklische Graphen (*directed acyclic graphs – DAGs*) werden durch Verbinden benachbarter ähnlicher Fragmente desselben Clusters gebildet. Auf diese Art werden näherungsweise größere Bruchstücke der Melodiestimmen zusammengesetzt. Jedem Fragment wird dabei ein Graphknoten zugeordnet. Die Kanten werden je nach Frequenz- und Lautstärkenunterschied, Länge des zeitlichen Spalts zwischen den Gruppen sowie Klangfarbenabweichung gewichtet.

4. Kurze Lücken zwischen verbundenen signifikanten Fragmenten werden bestmöglich geschlossen. Dabei kommt ein *Shortest Path* Algorithmus zur Anwendung um den besten tatsächlichen Pfad entlang gefundener Fragmente von einem DAG-Knoten zum nächsten zu finden. Die Gewichtung der Kanten entlang eines Pfades ergibt außerdem die Kosten.

5. Konkurrierende Pfade über die Fragmente werden aufgelöst indem kurze Pfade bzw. solche mit geringen Kosten bevorzugt werden.

6. Pfade des dominantesten Clusters formen schließlich die extrahierte Melodie. Die Dominanz ergibt sich dabei aus einer Kombination aus Lautstärke, Cluster-Konsistenz und Verhältnis der Gesamtzeit zu jener Zeit, in der eine Melodiestimme präsent ist.

Anwendung findet dieses Verfahren unter anderem in [30] um die Ähnlichkeit zwischen zwei Musikstücken zu ermitteln. Die ermittelte Melodie wird dabei anhand des durch eine Rhythmuserkennung gewonnenen Tempos normiert. Weiters findet eine Struktur-

analyse der zu vergleichenden Stücke statt und die Gesamt-Übereinstimmung ergibt sich aus jener der korrespondierenden Strophen sowie Refrains.

Ein weiters Verfahren – das zudem beim MIREX[4] Melodieerkennungs-Bewerb 2005[5] die beste Gesamtgenauigkeit ebenso wie die beste Laufzeit erreichte – wird von Dressler in [15] beschrieben. Dabei bildet eine Spektralanalyse mittels STFT die Ausgangsbasis, gefolgt von einer Bestimmung der Augenblicksfrequenzen (wie in Abschnitt 3.6.1.1 beschrieben) zur besseren Unterscheidung der vorhandenen Frequenzen. Diese werden laut des Modells aus [43] in stabile Sinusoide und Rauschen getrennt (vgl. Kapitel 3.2.7).

Im Unterschied zu PreFEst berücksichtigt der Ansatz aus [15] psychoakustische Erkenntnisse – wie etwa die bereits beschriebenen Maskierungseffekte. Entsprechend der menschlichen Wahrnehmung werden die Intensitäten der stabilen Sinusoide neu gewichtet und dann zu *Streams* zusammengefügt. Dabei ist die Ähnlichkeit – die nach Kriterien, die mit den zuvor beschriebenen von Marolt [30] vergleichbar sind – ausschlaggebend. Ein regelbasiertes Entscheidungsverfahren wählt dann die tatsächlich extrahierte Melodie aus den Kandidaten-Streams.

Ein weiterer Ansatz stammt von Madsen und Widmer [28]. Dabei wird davon ausgegangen, dass Melodie durch hohe Komplexität gekennzeichnet ist. Sie ist also weder eintönig noch redundant. Als Maßeinheit dafür bietet sich die bekannte Entropie nach Shannon an.

$$H(X) = -\sum_{x \epsilon X} p(x) \log_2 p(x) \tag{3.60}$$

Innerhalb bestimmter Zeitfenster wird sie für folgende drei Features einer Stimme berechnet. Von jener mit größtem Entropiewert innerhalb eines Fensters nimmt man an, dass sie zu diesem Zeitpunkt die Melodie enthält.

- Pitch Class: Wie bereits beschrieben handelt es sich dabei um die Tonhöhen, gefaltet auf eine einzelne Oktave. Dieses Feature ist vergleichbar mit den PCPs (vgl. Abschnitt 3.4).

- Intervall: Es wird die Häufigkeit aller musikalischer Intervalle gezählt – auf- und absteigende werden dabei gesondert behandelt. Dabei besteht entfernte Ähnlichkeit zu den ebenfalls bereits beschriebenen Intervallprofilen aus [27].

[4] Music Information Retrieval Evaluation eXchange
[5] http://www.music-ir.org/mirex2005/index.php/Audio_Melody_Extraction

- Notenlänge: Berechnet wird ein Histogramm über die Längen aller Noten. Die einzelnen Klassen unterscheiden sich dabei um mindestens 10%.

Alternativ zu den Entropien dieser drei Features bzw. Kombinationen daraus, wird die Komprimierbarkeit als Maß für Komplexität vorgeschlagen. Der Vorteil dieses Maßes besteht in der zusätzlichen Berücksichtigung wiederkehrender Strukturen, die die Komplexität verringern.

Der Ansatz geht jedoch davon aus, dass einzelne Stimmen bekannt und voneinander getrennt sind. Diese Annahme ist bei MIDI-Dateien als Eingabe gegeben, nicht jedoch bei direkter Analyse der Audiosignale. Die Methode ist daher im gegebenen Kontext nicht direkt anwendbar – könnte jedoch dennoch verwendet werden um die Entscheidung zwischen mehreren Kandidaten auf die Melodie – so wie sie in allen zuvor vorgestellten Verfahren zu treffen ist – zu unterstützen.

Daneben existiert noch eine Reihe weiterer Ansätze zur Melodieerkennung, die teilweise auf komplett anderen Ansätzen basieren. Ein Beispiel dafür ist die in [23] vorgeschlagene Methode, die auf *Self-Organizing Neural Networks (SONNETs)* aufbaut. Aufgrund der guten Dokumentation sowie der direkten Anwendbarkeit ist die Implementierung im Rahmen dieser Arbeit jedoch an das PreFEst Verfahren laut [20] angelehnt.

3.6.3 Resultate

Die Test-Implementierung, die im Zuge dieser Arbeit entwickelt wurde, orientiert sich an der vorgestellten PreFEst Methode. Allerdings galt es – wie in Abbildung 3.1 zu Beginn des Kapitel 3 dargestellt – Synergieeffekte zwischen den Berechnungen der beiden Features – THPCP-Vektoren sowie der Melodie – auszunutzen. So fand etwa die Berechnung der STFT oder der Kandidaten-Peaks nicht, wie in Abschnitt 3.6.1 vorgestellt, statt. Alternativ dazu sind es Codesegmente aus der HPCP-Berechnung, die hier wiederverwendet wurden.

Zum Test wurde die annotierte Referenzsammlung des *Melody Extraction Contest*[6] der *ISMIR 2004* herangezogen. Diese besteht aus zehn kurzen Stücken unterschiedlicher Stilrichtungen. In Abbildung 3.17 ist das Ergebnis der ermittelten Melodie im Vergleich zur tatsächlichen dargestellt. Die gelbe Linie ist dabei die tatsächliche – die weiße die geschätzte Melodie. Die Grafik im Hintergrund stellt die jeweilige Stärke der gerade vorhandenen Stimmen dar.

[6] http://ismir2004.ismir.net/melody_contest/results.html

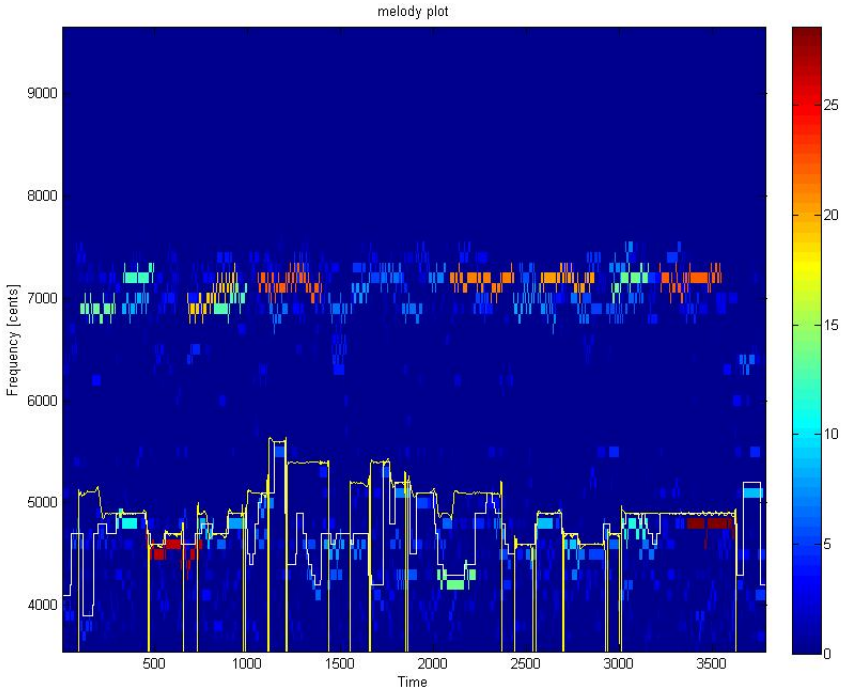


Abbildung 3.17: Ermittelte und tatsächliche Melodie eines Stücks aus dem *ISMIR 2004 Melody Extraction Contest* Testset

Was man sofort erkennt ist, dass keine Pausen erkannt werden, was vom Algorithmus her jedoch auch nicht vorgesehen ist. Weiters ist zu sehen, dass an Stellen, an denen die Berechnung daneben liegt, zwar eine vorhandene Stimme in der richtigen Tonhöhe erkannt, diese jedoch kurzfristig von anderen übertönt wird. Über die gesamte Länge (excl. Pausen) wurden in diesem Beispiel 64% der Melodie-Töne richtig ermittelt. Die gute Performanz am Anfang und vor allem am Ende des Ausschnitts steht dabei einem schlechten Abschneiden im Mittelteil gegenüber. Betrachtet man alle zehn Stücke des Testsets, so betrug die mittlere Trefferquote 56%. Dies liegt zwar sehr deutlich über der *Baseline* – dem Ergebnis, das durch reines Raten entstanden wäre – ob diese Genauigkeit jedoch ausreicht um Versionen eines Musikstücks zu identifizieren, ist fraglich.

Um den Rechenaufwand zur Analyse tatsächlicher Real-World Audioaufnahmen besser bewältigen zu können, wurde in der Test-Implementierung eine weitere Vereinfachung gegenüber dem PreFEst Verfahren eingeführt. Anstatt der relativ komplexen Bestimmung der Grundschwingungen fand die Behandlung der Obertöne wie bei der Berechnung des THPCP-Features (vgl. Abschnitt 3.4) beschrieben statt. Dies verursacht zwar erheblich mehr Rauschen, was aber in Kauf genommen werden kann, da zur Melodie-Extraktion ohnehin nur die stärksten – und somit deutlich über dem Pegel des Rauschens liegenden – Kandidaten-Frequenzen berücksichtigt werden.

Die Abbildungen 3.18 sowie 3.19 zeigen die Melodie der in allen bisherigen Beispielen gezeigten Refrain-Stelle von *Yesterday* von den *Beatles* sowie die tatsächlichen Noten dieses Ausschnitts. Die Note c' entspricht dabei 3600 cent – die Zeitachse hat Hundertstel-Sekunden als Einheit. Die Linie entspricht der berechneten Melodie (ist allerdings in der Grafik einen Ganzton höher eingezeichnet, um die einzelnen Peaks, aus denen sie sich zusammensetzt, im Hintergrund besser erkennen zu können).

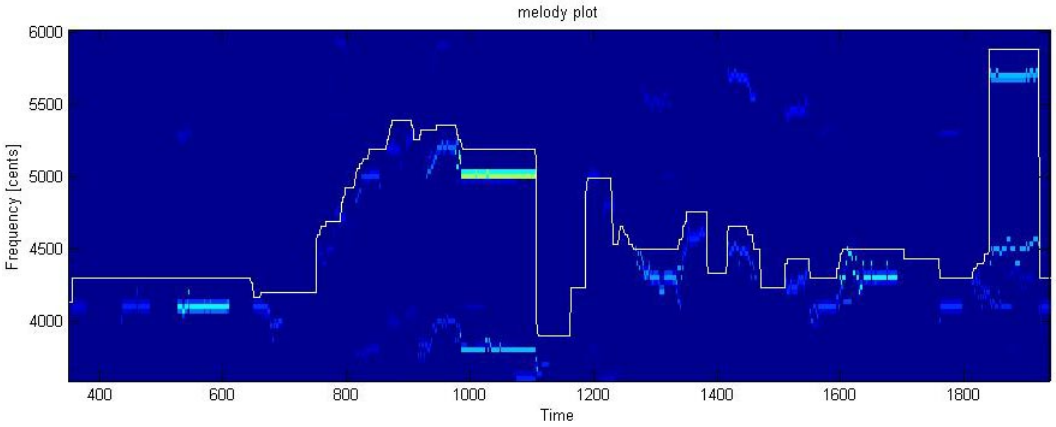


Abbildung 3.18: Ermittelte Melodie des Refrains von *Yesterday*

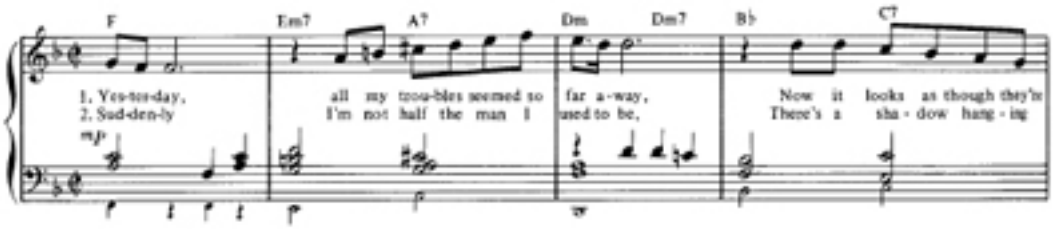



Abbildung 3.19: Ausschnitt der Melodie des Refrains von *Yesterday*

Gut zu erkennen sind die f' des ersten Takts ab etwas mehr als 500 $\frac{sec}{100}$ sowie die Pause (ca. 700 bis 750 $\frac{sec}{100}$). Ebenfalls deutlich zu sehen ist die aufsteigende Passage über sechs Töne bis hin zum f", von denen zwar alle anhand der Peaks zu erkennen, jedoch in der Ausgabe nur recht undeutlich repräsentiert sind. Wieder äußerst markant ist das d", das den dritten Takt dominiert und die darauf folgende Pause. Der vierte und letzte in Abbildung 3.19 gezeigte Takt wurde leider nur sehr ungenau und mit Fehlern erkannt. Der Rest des Refrains, zu dem hier keine Noten mehr dargestellt sind, wurde zufriedenstellend extrahiert, mit der einen Ausnahme, dass der hohe Ausschlag bei etwa 1850 $\frac{sec}{100}$ ein Fehler ist. Richtigerweise wäre die Stelle um genau eine Oktave tiefer – die entsprechenden Peaks sind sogar zu sehen, werden jedoch übertönt.

3.7 Strukturanalyse

Neben dem Vergleich der beiden Features Melodie und THPCP-Vektoren zur Erkennung von Cover-Versionen von Musikstücken, soll in dieser Arbeit auch untersucht werden, wie das Ergebnis durch eine Strukturanalyse beeinflusst werden kann. Der Hintergedanke dabei ist, dass sich die Struktur in vielen Cover-Versionen zumindest geringfügig ändert. Wie in Kapitel 2.1 beschrieben, kann dies etwa durch Einfügen eines zusätzlichen Solos bzw. ganz einfach durch nochmaliges Wiederholen des Refrains am Ende des Stückes geschehen. Ein System, das nur die jeweiligen Refrains miteinander vergleicht, wäre robust gegenüber solchen Veränderungen. Außerdem stellen mehrmals wiederholte Passagen redundante Informationen dar, die vermieden werden können.

Beschränkt man sich nur auf den Refrain, so gehen andere Bereiche wie Intro, Übergänge, Bridge, Solos und vor allem die Strophen verloren. Im Kontext aktueller Mainstream Pop- und Rockmusik stellt dies jedoch keine gravierende Einschränkung dar. Musikalisch werden die meisten dieser Songs vom Refrain dominiert und auch bei den Hörern ist er jener Teil, der am besten in Erinnerung bleibt bzw. über den das Stück identifiziert wird [22]. Es ist also zu erwarten, dass das Vernachlässigen der übrigen Teile zwar einen Informationsverlust darstellt, die verbleibenden Daten jedoch sogar dem näher kommen, was der menschliche Wahrnehmung und dem Gedächtnis entspricht.

Dieser Abschnitt befasst sich daher mit der Analyse der Struktur eines Musikstücks und daraus folgend dem Erkennen der Refrain-Teile. Die Basis aller Verfahren ist dabei das Finden wiederkehrender Elemente. Dazu ist ein Maß für die Ähnlichkeit solcher Stellen nötig. Dieses wiederum greift auf bereits vorgestellte Verfahren zum Berechnen von Spektren, Chroma-Vektoren (HPCP-Vektoren) und ähnlichen Features zurück. Das am öftesten wiederholte musikalische Motiv (mit einer gewissen Mindestlänge) ist in der Regel der Refrain.

Probleme ergeben sich jedoch daraus, dass die Spektren und somit alle extrahierten Features einzelner Wiederholungen nie komplett ident sind. Eine Auflistung der Gründe dafür liefert [22].

- Akustische Features und Ähnlichkeitsmaße
  Ob ein Teil eines Songs die Wiederholung eines anderen ist, muss anhand akustischer Features und deren Ähnlichkeit entschieden werden. Geht man von Audio-Signalen als Eingangsdaten aus, so sind schließlich keine Informationen auf höherem Abstraktionsniveau vorhanden. Das daraus entstehende Problem ist jedoch, dass diese Features bei gleicher Melodie erheblich abweichen können – etwa, wenn andere Begleitstimmen präsent sind.

- Kriterium zum Erkennen von Wiederholungen
  Der Schwellwert des Ähnlichkeitsmaßes, ab dem ein Abschnitt als Wiederholung eines anderen gewertet wird, ist vom Musikstück abhängig. Einerseits kann sein, dass ein Song durch ständig anwesende und gleichbleibende Begleitstimmen als ganzes ein sehr gleichförmiges Spektrum erzeugt. Der Schwellwert ist in diesem Fall relativ hoch anzusetzen. Andererseits ist es auch möglich, dass in einem Stück sowohl Begleitung als auch Tonart wechseln. Bei derartigen Songs muss ein verhältnismäßig niedriger Schwellwert gewählt werden. Da a priori keine Informationen über das Musikstück bekannt sind, muss dieser Wert automatisch angepasst werden.

- Erkennen beider Enden von wiederholten Abschnitten
  Ein exakter Algorithmus muss in der Lage sein, sowohl Start als auch Ende einer Sektion zu erkennen. Dass dies oft nicht selbstverständlich ist, zeigt das Beispiel einer Liedform *[A B C B C C]*. In diesem Fall würde zunächst *[B C]* als zusammengehörig klassifiziert. Erst mit Hilfe der Information, die durch das letzte *[C]* hinzukommt, kann eine Trennung von *[B]* und *[C]* erreicht werden.

- Erkennen von modulierten Wiederholungen
  Zusätzliche Schwierigkeiten entstehen durch Modulationen einzelner Abschnitte, wie sie oft am Ende eines Songs auftreten, um eine nochmalige Wiederholung des Refrains nicht eintönig erscheinen zu lassen. Dass es jedoch wichtig ist, auch derartige Passagen zuverlässig zu erkennen, zeigt das vorher gebrachte Beispiel, in dem erst durch die letzte Wiederholung ein korrektes Abgrenzen der vorigen möglich wurde.

Im Folgenden werden nun die einzelnen Schritte zum Finden des Refrains und seiner Wiederholungen (vorwiegend in Anlehnung an [22]) beschrieben.

3.7.1 Selbstähnlichkeits-Matrizen

Ausgangspunkt zum Erkennen sich wiederholender Abschnitte innerhalb eines Stücks sind in vielen Ansätzen (vgl. beispielsweise [10] bzw. [22]) die Ähnlichkeiten einzelner Frames zu anderen innerhalb desselben Stücks. Eine sich anbietende Darstellungsform dieser Maße sind die Selbstähnlichkeits-Matrizen, in denen jede Zelle die Übereinstimmung eines Frame-Paars darstellt.

Die Ähnlichkeit wird ausgehend von Chroma-Vektoren (HPCP-Vektoren, siehe Kapitel 3.4) errechnet. In [22] wird dazu das Maß $r(t,l)$ laut der Formel

$$r(t,l) = 1 - \frac{\left| \frac{\overrightarrow{v}(t)}{\max_c v_c(t)} - \frac{\overrightarrow{v}(t-l)}{\max_c v_c(t-l)} \right|}{\sqrt{12}} \tag{3.61}$$

vorgeschlagen. Die Vektoren $\overrightarrow{v}(t)$ und $\overrightarrow{v}(t-l)$ stellen dabei die Chromata zum Zeitpunkt t sowie des um l (Lag) Schritte versetzten Frames dar. Der Offset l muss somit $0 \leq l \leq t$ genügen. Die Normierungen – einerseits in Bezug auf die Vektoren, sodass ihr maximaler Wert 1 beträgt und andererseits des Gesamtterms mittels der Division durch die Länge der Diagonale des 12-dimensionalen Hypercubes ($\sqrt{12}$) – bewirkt, dass $r(t,l)$ immer $0 \leq r(t,l) \leq 1$ entspricht.

Um redundante Informationen zu vermeiden, kann die Selbstähnlichkeits-Matrix nun als Dreiecksmatrix berechnet werden, wobei die eine Dimension t und die andere l entspricht. Eine solche Matrix ist in Abbildung 3.21 gezeigt – dabei wurden zusätzlich alle Werte die kleiner als der Durchschnitt waren entfernt. Die ursprüngliche Selbstähnlichkeits-Matrix ist in Abbildung 3.20 zu sehen. Man erkennt deutlich den Vorteil der Umrechnung – wiederholte Passagen sind nun nicht als Diagonalen mit besonders hohen Werten, sondern als horizontale Linien repräsentiert. Diese sind – wie im Folgenden beschrieben – relativ leicht zu finden.

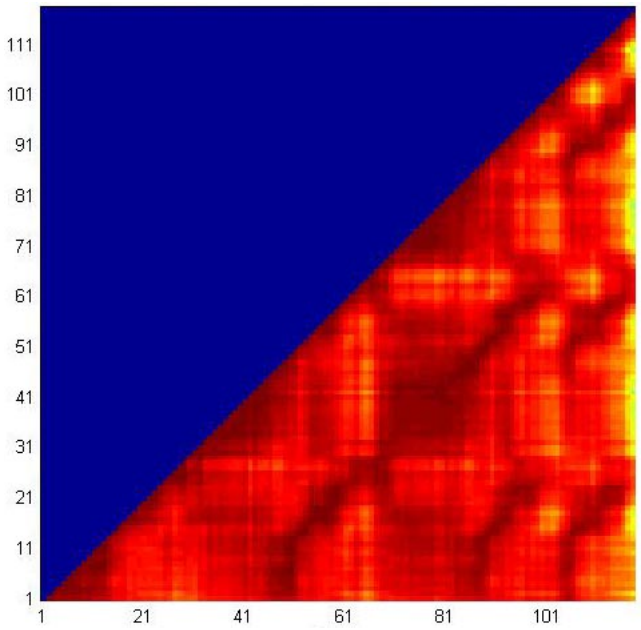


Abbildung 3.20: Selbstähnlichkeits-Matrix des Stücks *Yesterday* von den *Beatles*

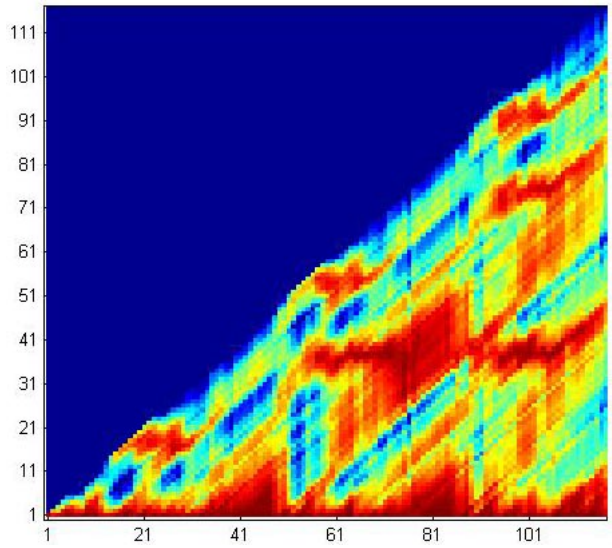


Abbildung 3.21: Selbstähnlichkeits-Matrix aus Abbildung 3.20 in der Time-Lag-Darstellung

3.7.2 Erkennung wiederholter Abschnitte

Die Erkennung wiederholter Abschnitte geht vom Maß $r(t, l)$ aus, das in einem Koordinatensystem mit der Zeit t als Abszisse und dem Zeitversatz l als Ordinate aufgetragen wird. Gesucht wird dann nach horizontalen Linien entlang derer hohe $r(t, l)$-Werte vorherrschen. Die horizontale Lage bedeutet einen gleichen Offset l und somit eine Parallelität über die Zeit; die hohen Werte deuten auf große Ähnlichkeit hin.

Um derartige Liniensegmente zu finden, wird für jeden Zeitversatz l berechnet, wie groß die Erfolgs-Wahrscheinlichkeit ist, solche in ihm zu finden. Der dafür berechnete Wert R_{all} ergibt sich laut folgender Formel.

$$R_{all}(t, l) = \int_{l}^{t} \frac{r(\tau, l)}{t - l} d\tau \tag{3.62}$$

Der Zeitpunkt, bis zu dem gesucht wird, kann dabei über t eingegrenzt werden, ist jedoch sinnvollerweise mit dem Ende des Songs gleichzusetzen. Vor dieser Berechnung wird $r(t, l)$ zudem durch Subtraktion des lokalen Mittelwerts normalisiert. Das Ergebnis dieser Berechnung ist in Abbildung 3.22 zu sehen. Ausgegangen wurde dabei wiederum vom Song *Yesterday* von den *Beatles*. Die Werte in der Ähnlichkeits-Matrix entsprechen jenen aus Grafik 3.21 – es wurde jedoch zur Verdeutlichung eine höhere Auflösung gewählt. Die Peaks in R_{all} zeigen, dass in den jeweils entsprechenden Zeilen verhältnismäßig hohe Werte von $r(t, l)$ zu finden sind.

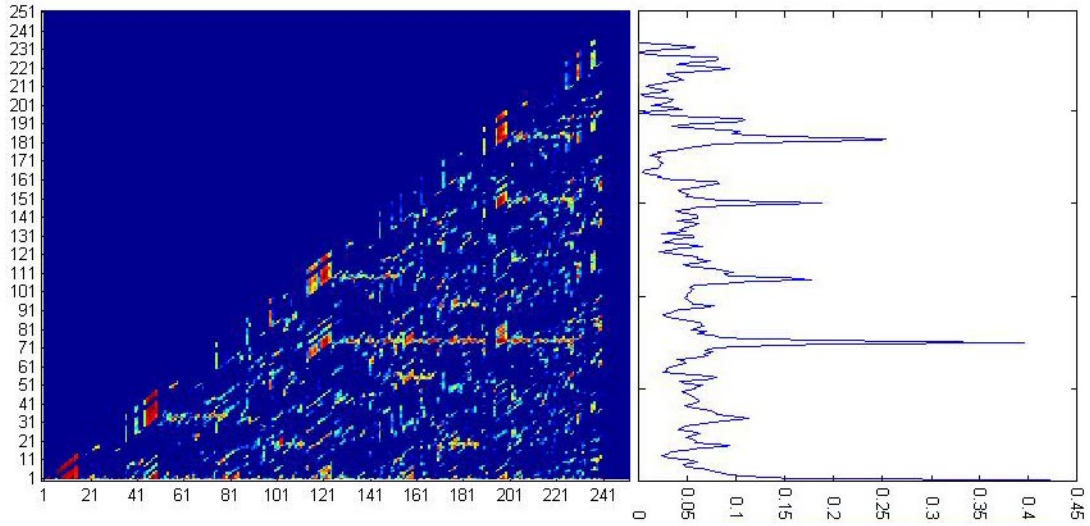


Abbildung 3.22: Time-Lag Darstellung der Ähnlichkeits-Matrix mitsamt den berechneten Werten R_{all} für jede Zeile

Weiterverfolgt werden dann Offsets, die vielversprechenden Peaks von R_{all} entsprechen. Als Auswahlkriterium dient dabei ein Schwellwert Th_R. Da der Wert Th_R eng mit dem Kriterium zum Bestimmen von Wiederholungen zusammenhängt, ist er wie oben erläutert an den entsprechenden Song anzupassen. Die Adaption wird durch Maximierung des in [35] vorgestellten Maßes σ_B^2 (Formel 3.63) zur Unterscheidungsfähigkeit zweier Klassen erreicht. Es zielt einerseits auf zwei möglichst gleich große Klassen ab und andererseits auf Klassen, die sich untereinander möglichst stark unterscheiden.

$$\sigma_B^2 = \omega_1\omega_2(\mu_1 - \mu_2)^2 \tag{3.63}$$

Dabei sind ω_1 und ω_2 die relativen Häufigkeiten der Klassen ($\omega_1 + \omega_2 = 1$) und μ_1 sowie μ_2 die Mittelwerte der jeweiligen Instanzen, die in diesem Fall den Werten der Peaks von R_{all} entsprechen.

An den Stellen, die den auf diese Weise selektierten Peaks entsprechen, wird nun in horizontaler Richtung nach Segmenten gesucht, deren einzelne Werte ebenfalls über einem Schwellwert liegen. Dieser wird analog zum Wert zur Peakfilterung anhand der Maximierung des Maßes aus Formel 3.63 ermittelt. Außerdem werden die Eingangswerte durch Berechnen des gleitenden Durchschnitts robuster gegenüber Ausreißern gemacht.

3.7.3 Arrangieren von Wiederholungen

Die gefundenen Liniensegmente entsprechen immer nur einem Paar von wiederholten Passagen. In der Realität muss jedoch auch mit Gruppen von drei oder noch mehr Wiederholungen gerechnet werden. Daher werden zunächst Cluster anhand jener Paare gebildet, bei denen eine ungefähre Übereinstimmung zwischen jeweils einem der Segmente besteht.

In einem weiteren Schritt wird versucht, noch weitere Wiederholungen zu finden, die im ersten Anlauf übersehen wurden. Dazu wird aus jeder Gruppe ein repräsentatives Segment mit Startpunkt T_{S_i} und Endpunkt T_{E_i} gewählt und nach weiteren dazu ähnlichen Stellen gesucht. Dabei geht man analog zur zuvor beschriebenen Peakerkennungs-Methode vor, mit dem Unterschied, dass nicht von R_{all} als Basis dient, sondern $R_{[T_{S_i},T_{E_i}]}$, das definiert ist als

$$R_{[T_{S_i},T_{E_i}]}(l) = \int_{T_{S_i}}^{T_{E_i}} \frac{r(\tau,l)}{T_{E_i} - T_{S_i}} d\tau \tag{3.64}$$

Aus der Menge der gefundenen Peaks werden im Wesentlichen all jene aussortiert, die in sich zu variabel sind und solche, die mit einem stärkeren Peak zeitlich überlappen. Die verbleibenden werden dann in die bestehenden Gruppen eingefügt. Abschließend müssen Gruppen zusammengelegt werden, falls sich herausstellt, dass sie überlappen.

3.7.4 Modulierte Wiederholungen

Bereits mehrfach wurde die Problematik modulierter Wiederholungen, wie sie vorwiegend am Ende eines Stücks vorkommen, erwähnt. Die Modulation kann dabei einfach durch die jeweilige Verschiebung der Grundfrequenzen in Halbtönen quantifiziert werden. Äquivalent zum Shiften der Tonmodelle zur Tonartbestimmung (vgl. Abschnitt 3.5) können auch die Chroma-Vektoren ringweise verschoben und in allen Versionen mit-

einander verglichen werden. Mathematisch wird das durch die einfache Verschiebungs-Matrix S ausgedrückt.

$$S = \begin{pmatrix} 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & 1 & 0 \\ 0 & \dots & \dots & 0 & 0 & 1 \\ 1 & 0 & \dots & \dots & \dots & 0 \end{pmatrix} \tag{3.65}$$

Gleichung 3.61 wird demnach modifiziert und man erhält unter Berücksichtigung einer Modulation um m Halbtöne folgenden Zusammenhang.

$$r_m(t,l) = 1 - \frac{\left| \frac{S^m \cdot \vec{v}(t)}{\max_c v_c(t)} - \frac{\vec{v}(t-l)}{\max_c v_c(t-l)} \right|}{\sqrt{12}} \tag{3.66}$$

Das beschriebene Verfahren zur Berechnung der Ähnlichkeiten wird nicht nur auf $m = 0$ sondern auf $m = 0, 1, ..., 11$ angewandt. Das Resultat besteht aus zwölf Mengen von Liniensegmenten und deren Gruppierungen. Diese werden schließlich zusammengefügt, sodass modulierte und nicht-modulierte Wiederholungen in den gleichen Sets enthalten sind.

3.7.5 Auswahl des Refrains

Das Ergebnis der bisherigen Schritte besteht aus einer Menge an Gruppen sich wiederholender Passagen innerhalb eines Musikstücks. Es bleibt noch zu bestimmen, welche dieser Gruppen nun die Refrain-Stellen beschreibt. Dazu wird ein weiteres Maß ν_i, für die Wahrscheinlichkeit eines Clusters, den Refrain zu enthalten, definiert.

$$\nu_i = \left(\sum_{j=1}^{M_j+1} \lambda_{ij} \right) \log \frac{T_{E_i} - T_{S_i}}{D_{len}} \tag{3.67}$$

Dabei ist λ_{ij} der Durchschnitt der Ähnlichkeiten $r(t,l)$ innerhalb eines Segments – die Summe somit der akkumulierte Wert über alle Elemente der Gruppe i. Die Konstante D_{len} ist ein frei wählbarer Parameter zur Normierung. Zudem wird die letztendliche Auswahl der Refrain-Gruppe wie folgt beeinflusst.

	Beginn [sek]	**Ende [sek]**
Verse 1	5	22
Verse 2	23	40
Verse 3	60	77
Verse 4	97	114

Tabelle 3.6: Lage der Refrain-Stellen in *Yesterday*

- Alle Gruppen, die Sektionen mit einer für einen Refrain untypischen Länge repräsentieren, werden aussortiert. Goto [22] schlägt dazu Grenzen von 7.7 und 40 Sekunden vor.

- Segmente, die sehr lang sind, können auch zusammengesetzte Song-Teile, etwa eine Strophe-Refrain-Kombination enthalten. Kürzere Fragmente, die am Ende eines langen liegen, werden daher stärker gewichtet.

- Refrains neigen dazu, aus zwei gleichen Teilabschnitten zu bestehen. Beinhalten Segmente jeweils zwei solche Passagen aus einer gleichen Gruppe, so werden sie ebenfalls stärker gewichtet.

In [22] wurde gezeigt, dass diese Methode dazu in der Lage ist, den Refrain von 80% der Songs aus dem Test-Set[7] korrekt zu erkennen. Die Implementierung im Rahmen dieser Arbeit ist sehr stark am Ansatz aus [22] angelehnt. Versuche zeigten, dass die Strukturanalyse in einer großen Anzahl verschiedener Musikstücke tatsächlich den Refrain zurückliefert. Abbildung 3.23 stellt mehrere Gruppen von wiederholten Passagen, die im Stück *Yesterday* von den *Beatles* identifiziert wurden, dar. Die Einheit der Abszisse sind dabei halbe Sekunden. Die tatsächliche Struktur des Stück entspricht laut [39]

Intro | *Refrain* | *Refrain* | *Bridge* | *Refrain* | *Bridge* | *Refrain* | *Outro*

Bei der zugrundeliegenden Aufnahme wurden die Lagen der einzelnen Refrain-Teile manuell wie in Tabelle 3.6 bestimmt. Obwohl der gesungene Text dieser Passagen nur bei der dritten und vierten Wiederholung der gleiche ist, und außerdem bei der ersten nicht alle Begleitstimmen präsent sind, wurden in Spur 6 dennoch alle mit ziemlich hoher Genauigkeit erkannt (der erste Balken entspricht zwei erkannten Refrain-Teilen, die sich minimal überschneiden) und tatsächlich als Refrain gekennzeichnet. Auch die beiden gleichen Bridges dazwischen konnte der Algorithmus richtig als Wiederholungen herausfiltern (Spur 5).

[7] RWC Music Database: Popular Music

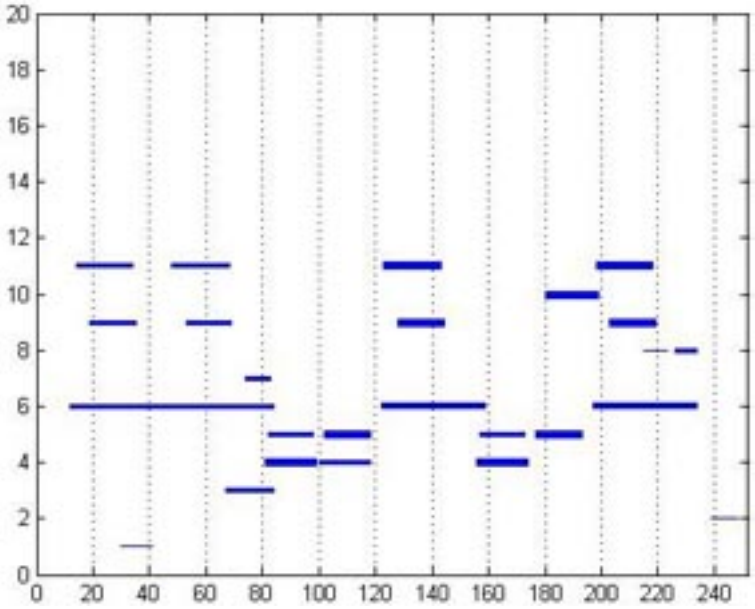


Abbildung 3.23: Wiederholte Passagen in *Yesterday*

Ein Problem, das in der Darstellung des Beispiels von *Yesterday* deutlich wird ist, dass die Start- sowie End-Zeiten der wiederholten Passagen zwar ungefähr richtig liegen, jedoch nicht auf die Sekunde genau bestimmt werden konnten. Dies liegt daran, dass zum Berechnen der Selbstähnlichkeits-Matrix die Auflösung reduziert wurde. Bei einer Sample-Frequenz von 44.1kHz und einer gewählten Schrittweite der Fensterung (vgl. Abschnitt 3.2.3) von 512 Samples kommt man auf etwas mehr als 86 Frames pro Sekunde – von denen jeder durch 36 Einzelwerte des THPCP-Vektors repräsentiert wird. Bei der zugrundeliegenden quadratischen Komplexität kommt man selbst bei relativ kurzen Songs – wie *Yesterday* mit 127 Sekunden – zu enorm vielen zu berechnenden und zu speichernden Werten. Daher wird durch Heranziehen des Durchschnitts aus jeweils über 100 Vektoren der tatsächliche Rechenaufwand erheblich verringert. Dafür wird eine gewisse Unschärfe beim Erkennen der Refrain-Abschnitte in Kauf genommen. Im gegebenen Kontext ist dies jedoch akzeptabel. Auch leicht verschobene Refrain-Stellen zweier Versionen eines Songs müssen dennoch genug Übereinstimmung aufweisen, um als solche erkannt zu werden.

Ein größeres Problem ist jedoch bei Disco-Versionen unterschiedlicher Musikstücke zu erkennen. In derartigen Versionen werden oft zusätzlich oder an Stelle der Strophen Passagen eingefügt, in denen Schlagwerke und elektronische Bässe dominieren. Diese Stellen sind sowohl in sich als auch untereinander sehr ähnlich und werden aufgrund ihrer herausragenden Präsenz meist fälschlicher Weise als Refrain gewertet. Um diese Fehler zu beseitigen, könnte weiterführend – in Anlehnung an [28] – versucht werden, Stimmen mit geringer Komplexität, wie sie in derartigen Rhythmus- bzw. Bass-Spuren vermutet werden kann, bei der Bestimmung des Refrains auszuschließen.

3.7.6 Variationen

In [10] werden Variationen zum beschriebenen Verfahren vorgestellt. In erster Linie geht es dabei um das Ähnlichkeitsmaß, das zur Berechnung der Selbstähnlichkeits-Matrix verwendet wird. Erläutert werden dabei folgende.

- Distanzmaß
  Anstatt der Ähnlichkeit wird die Distanz zwischen zwei Chroma-Vektoren auf Winkelbasis als
  $$d = 0.5 - 0.5 \cdot \frac{\nu_i \cdot \nu_j}{\mid \nu_i \mid\mid \nu_j \mid} \tag{3.68}$$
  berechnet.

- Tonartberichtigtes Distanzmaß
  Dieses Maß basiert auf der Annahme, dass Modulationen a priori bekannt sind und bereits beim Berechnen der Selbstähnlichkeits-Matrix berücksichtigt werden können. Da diese Annahme im gegebenen Kontext nicht haltbar ist, wird auf eine nähere Darstellung verzichtet.

- Intervall-basierte Distanz
  Intervalle sind unabhängig vom Grundton der jeweiligen Tonart und daher von obiger Annahme losgelöst. Die Chroma-Vektoren ν_i werden dazu in Vektoren u_i wie folgt umgerechnet.
  $$u_i[j] = \mid shift(\nu_{i+1}, j) - shift(\nu_i, 0) \mid \tag{3.69}$$
  Die Vektoren u_i enthalten somit die Distanz zwischen ν_i und dem um j geshifteten ν_{i+1}. Aus zwei derartigen Intervall-Vektoren kann nun wiederum ein beliebiges Distanzmaß berechnet werden.

Als zusätzliche Erweiterung schlägt [10] vor, anstatt eines linearen zeitlichen Zusammenhangs einzelner Wiederholungen einen beliebigen zu erlauben. Ersterer unterstellt, dass keine Tempo-Variationen zwischen einzelnen Wiederholungen auftreten. Dass diese eine nicht immer zutreffende Annahme ist, verdeutlichen zahlreiche Schlussteile von Songs, in denen das Tempo abnimmt (*rallentando, allargando*) bzw. auch unterschiedliche Phrasierungen wiederholter Motive um die Spannung zu halten.

Die Methode, die es erlaubt auch Wiederholungen trotz beliebiger Tempo-Variationen zu erkennen, ist der aus der Spracherkennung bekannte *Dynamic Time Warp*-Algorithmus. Grundlage dazu ist die komplette Selbstähnlichkeits-Matrix. Anhand dieser Daten wird eine neue Matrix DTW aufgestellt, deren Zellen DTW_{ij} genau eine

Ausrichtung zweier Fragmente des Songs repräsentiert, bei der Frame i des einen auf Frame j des anderen zu liegen kommt. Der Wert der Zelle beschreibt, wie ähnlich die beiden aneinander ausgerichteten Abschnitte bis zu dem aktuell betrachteten Paar bestenfalls sein können. Anhand dieser Information wird dann in einer weiteren Iteration die optimale Ausrichtung bis zum nächsten Paar bestimmt. Unterschiedliche Tempi werden insofern berücksichtigt, als dass bei der Ausrichtung der Song-Segmente aneinander, auch ausgelassene bzw. neu eingefügte Frames erlaubt sind. Gesucht werden im gegebenen Kontext dann besonders gut zusammenpassende Abschnitte. Der Algorithmus wird in Kapitel 4.2 noch näher beschrieben, da er auch im Klassifizierungsschritt Anwendung findet.

Die hier genannten Variationen wurden jedoch im Rahmen dieser Arbeit nicht implementiert. Dies liegt einerseits daran, dass die zusätzlich vorgestellten Ähnlichkeitsmaße keine dokumentierten Verbesserungen mit sich ziehen und andererseits die Unterstellung eines gleichbleibenden Tempos innerhalb der Refrain-Stellen eines Songs – obwohl nicht immer zutreffend – angesichts der Ergebnisse aus [22] eine sinnvolle Vereinfachung zur Reduzierung des Rechenaufwands darstellt.

4 Klassifizierung

Nachdem beschrieben wurde, wie signifikante Features aus den Audio-Signalen extrahiert werden können, gilt es nun, anhand dieser Merkmale zu entscheiden, ob ein Musikstück eine Version eines anderen ist oder nicht. Im Rahmen der Mustererkennung bezeichnet man diesen Schritt als Klassifizierung – eine Instanz wird genau einer Klasse zugeordnet. Im gegebenen Kontext entspricht ein solcher Cluster einem eigenständigen Song. Die perfekte Instanz einer jeden Klasse ist somit die Original-Interpretation – alle weiteren Instanzen sind (Cover-)Versionen.

In vielen Fällen kann eine Instanz einer bestimmten Klasse zugeordnet werden, indem ihre Übereinstimmung mit allen möglichen berechnet und die ähnlichste ausgewählt wird. Diese Vorgehensweise ist hier nicht möglich, da die Anzahl der Klassen nach oben hin nicht sinnvoll beschränkt werden kann, auf jeden Fall aber enorm groß ist.

Alle geeigneten Methoden der Klassifizierung laufen somit darauf hinaus, ein Maß für die Ähnlichkeit eines Songs zu allen bisher bekannten Klassen zu finden und dann zu entscheiden, ob die aktuelle Instanz einer von ihnen zuzurechnen ist, oder ob ein neuer Cluster gebildet werden soll. Für diese Entscheidung muss zusätzlich zur Ähnlichkeits-Metrik auch noch ein geeigneter Schwellwert gefunden werden.

Ein eigenes Forschungsgebiet befasst sich mit dem Ermitteln von Ähnlichkeiten zwischen Musikstücken (vgl. beispielsweise [37]). Die dort vorwiegend verwendeten Merkmale wie Klangfarbe oder Rhythmus sind zur Cover-Versions-Erkennung wenig geeignet, da sie wie in Kapitel 2.1 von einer Version zur anderen variieren können. Weiters geht es nicht darum zu bestimmen, ob zwei Stücke ähnlich klingen, sondern ob die zugrunde liegende Melodie die gleiche ist. Obwohl das Feld der Musik-Ähnlichkeit sehr naheliegend ist, sind seine Methoden (zur Klassifizierung) nur sehr bedingt übertragbar.

Auch bei Anwendung adaptierter Methoden zur Ermittlung des Übereinstimmungsgrads zweier Musikstücke bleibt die Frage offen, ab welchem Schwellwert sie als Version desselben Stücks einzustufen sind. Diese Frage ist nicht eindeutig beantwortbar und insbesondere im Hinblick auf Copyright- und Plagiats-Aspekte ein Streitpunkt. Geht man

jedoch von solchen Versionen als Grundwahrheit aus, die offiziell als solche bezeichnet werden, so kann anhand von Trainingsdaten ein sinnvoller Schwellwert für jede geeignete Kombination aus Feature und Klassifizierungsmethode ermittelt werden. Im Folgenden werden nun zwei Ansätze zur Klassifizierung vorgestellt.

Der eine stammt aus [18] und beschreibt direkt die Verwendung des in Kapitel 3.4 vorgestellten HPCP-Features sowie des um Transpositions-Robustheit erweiterten THPCP-Features aus Abschnitt 3.5 zur Cover-Versions-Erkennung basierend auf dem *Dynamic Time Warp*-Algorithmus. Dieser findet häufig in der Spracherkennung Anwendung und wird auch in [10] zur Strukturanalyse verwendet. Im Rahmen dieser Arbeit erfolgt zudem eine Adaption zur Cover-Versions-Erkennung anhand der extrahierten Melodie.

Der zweite – in dieser Arbeit entwickelte – Ansatz unterstellt eine konstante globale Tempo-Variation zwischen Original und Cover-Version. Die Annahme erscheint sinnvoll, da Tempoänderungen oft mit der Interpretation eines Songs in einem anderen Stil verbunden sind. Dieser Stil hat dann jedoch in der Regel ein charakteristisches Tempo, das innerhalb eines Stücks beibehalten wird. Gegenüber lokal sehr beschränkter Änderungen des Tempos, wie sie etwa bei abweichender Phrasierung auftreten, ist die Methode dennoch robust. Sie wird im Weiteren als die des *linearen Zeitzusammenhangs* bezeichnet.

4.1 Ähnlichkeits-Matrizen

Ausgangsbasis für alle hier vorgestellten Klassifizierungsmethoden ist die Ähnlichkeits-Matrix SM der beiden zu vergleichenden Stücke. Jede Zelle $SM_{i,j}$ enthält dabei das Maß an Übereinstimmung des i-ten Frames aus einem Song mit dem j-ten des anderen. In Bezug auf das (T)HPCP-Feature wurden dazu bereits in Kapitel 3.7.1 sowie Abschnitt 3.7.6 folgende Metriken vorgestellt, die alle einen möglichen Wertebereich von $[0, 1]$ besitzen. Die Distanzen aus [10] wurden dabei ebenfalls zu Ähnlichkeitsmaßen umgeformt.

- Normierte Differenz nach [22]
  Die Ähnlichkeit entspricht der Kürze des normierten Differenzvektors.

$$r(i,j) = 1 - \frac{\left| \frac{\vec{u}(i)}{\max_c u_c(i)} - \frac{\vec{v}(j)}{\max_c v_c(j)} \right|}{\sqrt{12}} \tag{4.1}$$

  Aufgrund der hohen Geschwindigkeit mit der dieses Maß berechnet werden kann, ist es jenes, auf das in dieser Arbeit zurückgegriffen wird. Alle folgenden Schritte bauen somit darauf auf.

- Normierter Winkel
  Die Ähnlichkeit entspricht jener der Winkel zwischen den beiden Vektoren.

$$d(i,j) = 0.5 + 0.5 \cdot \frac{u(i) \cdot v(j)}{\mid u(i) \mid\mid v(j) \mid} \tag{4.2}$$

- Korrelation
  Für die Korrelation zweier Variablen gibt es mehrere Maße – das bekannteste ist jenes nach *Pearson* und ist definiert als

$$r_{XY} = \frac{Cov(X,Y)}{\sqrt{Var(X)} \cdot \sqrt{Var(Y)}} = \frac{E((X - E(X))(Y - E(Y)))}{\sqrt{E((X - E(X))^2)} \cdot \sqrt{E((Y - E(Y))^2)}} \tag{4.3}$$

  mit der Erwartungswertfunktion E. Die Werte reichen dabei von -1 bei stärkst möglichem inversem Zusammenhang bis 1 bei Übereinstimmung. Um sie ebenfalls auf einen Bereich von $[0, 1]$ zu normieren, erfolgt die Umrechnung nach

$$r'_{XY} = 0.5 + \frac{r_{XY}}{2} \tag{4.4}$$

- Intervall-basierte Distanz nach [10]
  Nötig ist vorab die Berechnung der Intervallschritte innerhalb von u bzw. analog dazu für v als

$$\overline{u_i}[s] = 1 - \frac{\mid \frac{shift(u(i+1),s)}{\max_c u_c(i)} - \frac{shift(u(i),0)}{\max_c v_c(j)} \mid}{\sqrt{12}} \tag{4.5}$$

  Die Vektoren $\overline{u_i}$ enthalten somit die Ähnlichkeiten zwischen $u(i)$ sowie dem um s geshifteten $u(i+1)$, was die Dominanz des jeweiligen Intervalls ausdrückt. Diese Intervall-Wahrscheinlichkeits-Vektoren können nun anhand der zuvor beschriebenen Maßzahlen verglichen werden.

Eine mögliche Visualisierung ist jene, eine Ebene zwischen zwei Koordinatenachsen – die dem Zeitfortschritt im jeweiligen Song entsprechen – aufzuspannen. Werden Ähnlichkeiten durch Farben dargestellt, können Verfahren der Bildverarbeitung zum Finden von Ähnlichkeiten angewandt werden. Diese entsprechen nämlich Diagonalen entlang derer besonders hohe Ähnlichkeitswerte liegen. Der im folgenden Kapitel vorgestellte Dynamic Time Warp-Algorithmus wird in der Bildverarbeitung etwa zum Identifizieren gleicher Objekte in mehreren Bildern angewandt.

4.2 Dynamic Time Warp

Unbedingt notwendig zur Klassifizierung zweier Stücke als eigenständig bzw. als Versionen des selben Songs ist es, deren Merkmale im zeitlichen Verlauf zu beobachten. Anders als bei allgemeiner Ähnlichkeit reicht es nicht, gleiche Akkorde, Harmonien oder Klangfarben zu identifizieren, sondern es gilt übereinstimmende Akkord- oder Harmonie-Folgen zu finden.

Gesucht ist somit eine Ausrichtung der beiden Musikstücke aneinander, sodass jeder Frame des einen mit einem möglichst ähnlichen des anderen korrespondiert. Da beide Stücke in der Regel nicht auf den Frame genau gleich lang sind, ist es nötig, Lücken in die Anordnung einfügen zu können. Wird etwa der C-Dur Dreiklang in einem Tempo gespielt, bei dem jeder Ton zwei Frames anhält und in der zweiter Version in einem langsameren Tempo, sodass derselbe Ton jeweils in drei Frames aufscheint, so ergibt dies folgende vereinfachte Ausrichtung.

$$\begin{array}{cccccccccc} c & c & - & e & e & - & g & g & - \\ c & c & c & e & e & e & g & g & g \end{array}$$

Ein weiteres Beispiel für die Notwendigkeit von Lücken in einem größeren Rahmen sind neue Abschnitte in einem Song, die im anderen nicht enthalten sind. Geht man etwa von einer ursprünglichen Liedform *[Intro-Vers-Chorus-Vers-Solo-Vers-Chorus]* aus, und davon, dass in der Cover-Version das Solo fehlt, dafür aber eine zusätzliche Wiederholung des Refrains am Ende hinzugefügt wurde, so entspricht dies folgender Ausrichtung.

$$\begin{array}{cccccccc} Intro & Vers & Chorus & Vers & Solo & Vers & Chorus & - \\ Intro & Vers & Chorus & Vers & - & Vers & Chorus & Chorus \end{array}$$

Der Dynamic Time Warp (DTW) Algorithmus berechnet anhand eines Maßes für die Ähnlichkeit zweier Elemente (Frames) der Eingabedaten sowie einer festgelegten Strafe für nötige Lücken die optimale Ausrichtung. Diese Ähnlichkeiten wurden bereits quantifiziert und sind aus der Ähnlichkeits-Matrix ersichtlich. Jede Zelle $SM_{i,j}$ entspricht der Gleichartigkeit zwischen Frame i des eine und Frame j des anderen Songs. In der Visualisierung kann das Ergebnis des DTW-Algorithmus als optimaler Pfad diagonal durch die aufgespannte Ebene (vgl. Kapitel 4.1) interpretiert werden.

Die einfachste Möglichkeit das Verfahren durchzuführen, besteht in der Methode des *Dynamic Programming*. Dabei wird eine $(n+1) \times (m+1)$- Matrix DTW aufgestellt, wobei n und m die Längen der beiden Input-Ketten – bestehend aus den Frames der

betrachteten Songs – bezeichnen. Die Zellen der ersten Zeile sowie der erste Spalte werden mit $-\infty$ initialisiert. Einzige Ausnahme ist die Zelle $DTW_{0,0}$ die den Wert 0 erhält. Davon ausgehend wird iterativ jede weitere Zelle, für die alle Voraussetzungen bekannt sind, berechnet als

$$DTW_{i,j} = SM_{i,j} + \max \begin{cases} DTW_{i-1,j-1} & \text{Gute Übereinstimmung} \\ DTW_{i-1,j} & \text{Einfügung} \\ DTW_{i,j-1} & \text{Auslassung} \end{cases} \tag{4.6}$$

Jeder Eintrag in $DTW_{i,j}$ bezeichnet somit den maximalen Gewinn einer Ausrichtung der beiden Eingabe-Ströme bis zum $(i-1)$-ten und $(j-1)$-ten Element, wobei diese in der gegenseitigen Ausrichtung miteinander korrespondieren. Die Werte $SM_{i,j}$ sind der Ähnlichkeits-Matrix zu entnehmen. Der optimale Gesamtgewinn als Resultat ist immer in der Zelle $DTW_{n,m}$ enthalten.

Ist man an der optimalen Ausrichtung selbst interessiert, so ist es weiters nötig, in einer zweiten Matrix festzuhalten, welche der drei in Formel 4.6 beschriebenen Alternativen das Maximum ergab. In einem abschließenden *Back-Tracking*-Schritt wird dann, ausgehend vom Endergebnis in Zelle $DTW_{n,m}$, der Pfad zurückverfolgt. Dies kann im gegebenen Kontext jedoch vernachlässigt werden, da nicht die exakte Ausrichtung der beiden Songs aneinander von Interesse ist, sondern lediglich der erzielte Gewinn als Maß für die Übereinstimmung.

Der errechnete Wert ist abhängig von der Länge der beiden verglichenen Musikstücke. Um ihn dennoch mit einem einheitlichen Schwellwert vergleichen zu können, ist eine Normierung notwendig. Dazu wird der Gewinn mit der Länge des ermittelten Pfades in Relation gesetzt.

4.3 Linearer Zeitzusammenhang

Ein weiterer Beitrag dieser Arbeit ist, neben der Gegenüberstellung verschiedener Features zum Erkennen von Versionen eines gleichen Musikstücks, der Vergleich unterschiedlicher Methoden der Klassifizierung. Zusätzlich zum in der Literatur vorgestellten Dynamic Time Warp Verfahren – das im Audio-Bereich vor allem in der Sprachverarbeitung Anwendung findet – soll hier ein Ansatz vorgestellt werden, der auf der Annahme eines linearen Zeitzusammenhangs beruht. Anders als bei reiner Sprache, wo die Geschwindigkeit von Wort zu Wort variieren kann, ist die Interpretationsfreiheit bei Musik aufgrund gleichbleibender Notierung bei Versionen eines Songs eingeschränkt. Das glo-

bale Tempo kann sich von einer Version auf die andere zwar sehr wohl ändern; doch ist anzunehmen, dass die Änderungsrate konstant bleibt. Unterschiedliche Phrasierungen sind zwar auch gewissermaßen Tempoänderungen, jedoch sind diese derart lokal beschränkt, dass sie durch eine entsprechende Verringerung der Zeitauflösung eliminiert werden können.

Als einfache Variante kann der Durchschnitt der Ähnlichkeitswerte entlang der Hauptdiagonale der Ähnlichkeits-Matrix als Maß für die Übereinstimmung zweier Songs herangezogen werden. Probleme treten jedoch auf, sobald entweder die Struktur verändert, oder einfach eine kurze Stille am Beginn bzw. am Ende einer der Aufnahmen hinzugefügt wurde.

Um dieses Problem zu lösen, wird stattdessen nicht die Diagonale sondern jene Gerade durch die Ähnlichkeits-Matrix, entlang derer sich die größten Werte befinden, zur Klassifikation herangezogen. Eine notwendige Einschränkung dabei ist, dass die Steigung grob 45 Grad entspricht, wie es bei unverändertem Tempo der Fall wäre. Einen zusätzlichen Nutzen dieser Einschränkung des Suchraums stellt außerdem der verringerte Rechenaufwand dar.

Zur Identifizierung der dominantesten Gerade wird die Hough-Transformation (vgl. [3] und [40]) herangezogen. Dabei werden die einzelnen Punkte vom herkömmlichen räumlichen Koordinatensystem (Bildbereich) in einen Raum transformiert, dessen Punkte Geraden entsprechen. Die Koordinaten repräsentieren dabei die Steigung θ sowie dem Abstand zum Nullpunkt ρ. Jede Gerade durch den Punkt (x, y) entspricht somit

$$\rho = x \cos\theta + y \cos\theta \tag{4.7}$$

Da unendlich viele Geraden durch einen beliebigen Punkt gelegt werden können, besitzt jeder Punkt im Bildbereich eine Vielzahl an Darstellungen im Hough-Raum. Um diese dennoch handhaben zu können, wird der Hough-Raum entsprechend einer gewählten Auflösung in äquidistante Abschnitte der Längen $\Delta\theta$ und $\Delta\rho$ eingeteilt und so diskretisiert. Im Unterschied zu ρ, das theoretisch ins Unendliche wachsen kann, ist θ – da die Richtung der Geraden irrelevant ist – auf einen Bereich zwischen 0 und 180 Grad eingeschränkt. Jeder Punkt (x, y) hat somit $\frac{180}{\Delta\theta}$ Repräsentationen im Hough-Raum, entsprechend folgenden Formeln

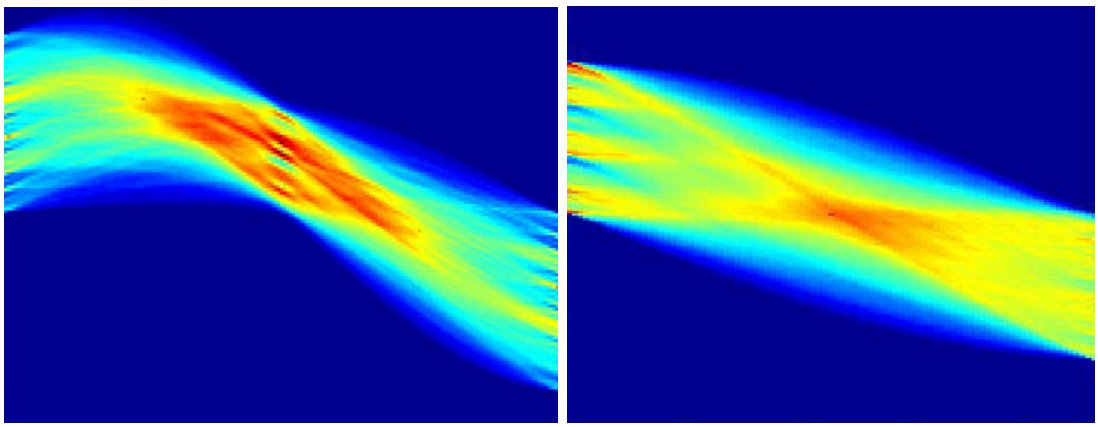
$$\theta(i) = i\frac{180}{\Delta\theta} \tag{4.8}$$

$$\rho(i) = x \cos\theta(i) + y \cos\theta(i) \tag{4.9}$$

Das Errechnete $\rho(i)$ kann wiederum einer der Klassen des diskretisierten Hough-Raums zugerechnet werden. Auf diese Art erhält man die Koordinaten aller Hough-Transformierten eines Punktes.

Wird ein ganzes Bild hough-transformiert, so entstehen durch die Kumulierung in diesem Raum Peaks an jenen Stellen, die tatsächlichen Geraden im Bild entsprechen. Im gegebenen Kontext kann die dominanteste Gerade somit als Modalwert der Hough-Transformierten bestimmt werden.

In Abbildung 4.1 sind die Hough-Transformierten zweier Ähnlichkeitsmatrizen dargestellt. Die Grafik (a) entstand aus den Ähnlichkeiten zweier Versionen des gleichen Musikstücks. In x-Richtung sind dabei die Winkel von 0 bis 90 Grad dargestellt. Deutlich zu sehen sind die hohen Werte in der Mitte – also bei annähern 45 Grad. Die y-Richtung stellt die Verschiebung der Geraden zum Nullpunkt dar. Hier ist die leichte Verschiebung in die oberen Quadranten zu erkennen, was bedeutet, dass die dominanten Geraden durch die Ähnlichkeitsmatrix im Bildbereich in x-Richtung verschoben liegen. Der Grund dafür kann ein längeres Intro jenes Songs sein, der entlang eben dieser x-Richtung aufgetragen ist.



(a) (b)

Abbildung 4.1: Hough-Transformierte der Ähnlichkeitsmatrizen zweier Versionen des gleichen (a) und zweier unabhängiger Musikstücke (b)

In Abbildung 4.1(b) ist die Hough-Transformierte der Ähnlichkeitsmatrix zweier unabhängiger Songs zu sehen. Markant sind die insgesamt niedrigeren Werte – dennoch relativ hohe Werte sind neutral in der Mitte gelegen. Diese entsprechen Geraden durch den Ursprung mit einer Steigung von ungefähr 45 Grad. Einige Ausreißer sind weiters im Bereich extrem flacher Geraden zu erkennen, diese stammen von sehr neutralen Frames innerhalb eines Songs die zu vielen des anderen Songs relativ ähnlich sind.

Eine Änderung der Struktur bewirkt jedoch, dass übereinstimmende Passagen vor und nach dieser Änderung auf parallelen Geraden liegen. Die Beschränkung auf eine einzige Gerade, entlang derer die Ähnlichkeit gemessen wird, kann Variationen in der Struktur eines Musikstücks somit nicht berücksichtigen. Dennoch korrespondieren die dominantesten Geraden mit übereinstimmenden Abschnitten der verglichenen Songs.

Trotz der angesprochenen Schwäche wird im Rahmen dieser Arbeit auf eine Erweiterung des Prinzips zur besseren Berücksichtigung von Struktur-Änderungen verzichtet. Grund dafür ist, dass das Verfahren auch bei nur teilweise korrespondierenden Abschnitten höhere Werte als bei komplett unabhängigen Stücken liefert und somit eine Entscheidung ermöglicht.

Ähnlich wie bei dem durch den Dynamic Time Warp Algorithmus ermittelten Wert, ist auch hier eine Normierung notwendig. Diese geschieht jedoch implizit durch Heranziehen der durchschnittlichen Ähnlichkeit entlang der ermittelten Geraden anstatt derer Summe.

Beiden Verfahren gemeinsam ist, dass der Schwellwert, ab dem zwei Songs als Versionen desselben Musikstücks gewertet werden, anhand von Trainingsdaten bestimmt werden muss.

5 Resultate und Perspektiven

Nachdem in den vorhergehenden Kapiteln die Methodik der Versions-Erkennung bei Musikstücken dargelegt wurde, folgt nun eine Übersicht über die Resultate, die die Implementierung des experimentellen Prototyps ergab.

5.1 Evaluierung

Ziel dieser Arbeit ist der Vergleich unterschiedlicher Features sowie Klassifizierungsmethoden zur Erkennung von Coverversionen. In dieser Arbeit wurden dabei folgende Features beschrieben.

- Transposed Harmonic Pitch Class Profile
- Transposed Harmonic Pitch Class Profile des Refrains
- Melodie
- Melodie des Refrains

Da diese Merkmale alle zu komplex sind, um eine direkte Klassifizierung zu ermöglichen, werden im Klassifizierungsschritt jeweils die Features zweier Songs gegenübergestellt und aus diesem Vergleich eine einfache Maßzahl für deren Gleichartigkeit ermittelt. Auch hierfür wurden bereits Methoden vorgestellt. Im Rahmen der Evaluierung fand zudem eine Erweiterung um zwei zusätzliche, sehr einfache, Varianten statt. Bei den vier verglichenen Klassifizierungsmethoden handelt es sich um

- Dynamic Time Warp
- Linearer Zeitzusammenhang

- Hauptdiagonale
  Diese Methode berechnet die Gleichartigkeit zweier Stücke anhand der durchschnittlichen Werte entlang der Hauptdiagonale der Ähnlichkeitsmatrix. Dadurch werden zwar Variationen im Tempo berücksichtigt, die Methode ist jedoch nicht robust gegenüber Strukturveränderungen.

- 45 Grad Diagonale
  Eine weitere Variante stellt das Heranziehen des Durchschnittswerte der Ähnlichkeiten entlang einer Gerade aus dem Ursprung der Ähnlichkeitsmatrix mit einer Steigung von 45 Grad dar. Dieses Vorgehen unterstellt nicht nur eine gleiche Struktur der beiden verglichenen Songs, sondern auch ein gleiches Tempo.

Es liegt auf der Hand, dass die beiden zuletzt genannten Methoden unzulässige Annahmen treffen. Sie wurden jedoch aus dem Grund in die Evaluierung miteinbezogen, um zu sehen, wie stark sich die Ergebnisse durch Berücksichtigung der jeweiligen Strukturen und Tempi der Musikstücke verbessern lassen.

Die Bewertung der einzelnen Kombinationen erfolgt anhand der *recall*, *precision* und *fallout* Maße. Diese sind im Kontext von Suchaufgaben gebräuchlich und messen die Güte des Ergebnisses unter der Annahme, dass gewisse Informationen über den Suchraum vorhanden sind.

Der *recall* misst, wie viele der relevanten Instanzen p tatsächlich gefunden wurden und ist definiert als

$$recall = \frac{pos_{found}}{pos_{found} + pos_{missed}} \tag{5.1}$$

Als relevante (positive) Instanz wird in diesem Kontext ein Paar an Songs gewertet, das aus zwei Versionen des gleichen Musikstücks besteht. Durch ein Anpassen des Schwellwerts, ab dem Versionen als solche erkannt werden, kann dieser Wert natürlich beliebig erhöht werden. Dies geht jedoch insofern zu Lasten der Qualität, als dass immer mehr Instanzen fälschlicherweise im Ergebnis aufscheinen. Die *precision* misst daher das Verhältnis der richtigen (positiven) Treffer in der Ergebnismenge zu den inkorrekten (negativen) und ist festgelegt als

$$precision = \frac{pos_{found}}{pos_{found} + neg_{found}} \tag{5.2}$$

Ähnlich zur *precision* ist auch der *fallout* ein Maß für den Anteil an falschen Instanzen im Ergebnis. Der Unterschied ist jedoch, dass diese Menge nicht zum Gesamtumfang des Ergebnisses, sondern zur Zahl der negativen Instanzen im Suchrauch selbst in Beziehung gesetzt wird.

$$fallout = \frac{neg_{found}}{neg_{found} + neg_{missed}} \tag{5.3}$$

Da diese drei Maßzahlen sehr stark miteinander korrelieren, ist es im Sinne einer leichteren Vergleichbarkeit von Resultaten nützlich, sie zu einem einzelnen Wert zu verknüpfen. Dieser so genannte *f-Wert* ergibt sich als

$$f = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5.4}$$

Um Ergebnisse zudem besser beurteilen zu können, wird in Anlehnung an [18] eine Baseline für diese Maße hergeleitet. Sie beschreibt des Ergebnis, das bei zufälliger Auswahl einer bestimmten Anzahl n an Paaren zustande kommt. Ausgegangen wird dabei von einer Sammlung mit N Songs und somit $N_{pairs} = \frac{N \cdot (N-1)}{2}$ möglichen Paarungen, von denen N_{cover} tatsächlich Versionen des jeweils selben Musikstücks sind. Die Menge der richtigen Klassifizierungen innerhalb dieser Ergebnismenge ergibt sich als

$$\begin{aligned} pos_{found}^{rand} &= \sum_{k=1}^{min(n, N_{cover}-1)} k \cdot \frac{\binom{N_{pairs}-N_{cover}}{n-k}}{\binom{N_{cover}-1}{k} \cdot \binom{N_{pairs}-1}{n}} \\ &= \frac{n \cdot (N_{cover} - 1)}{N_{pairs} - 1} \end{aligned} \tag{5.5}$$

Setzt man pos_{found}^{rand} in die Formeln für *recall* und *precision* ein, so ergeben sich

$$recall^{rand} = \frac{pos_{found}^{rand}}{N_{cover}-1} = \frac{n}{N_{pairs} - 1} \tag{5.6}$$

$$precision^{rand} = \frac{pos_{found}^{rand}}{n} = \frac{N_{cover} - 1}{N_{pairs} - 1} \tag{5.7}$$

$$f^{rand} = 2 \cdot \frac{N_{cover} - 1}{N_{cover} - 1 + n} \cdot \frac{n}{N_{pairs} - 1} \tag{5.8}$$

Titel	**Anzahl**
Imagine	9
Yesterday	9
Suspicious Minds	8
Lady Marmelade	6
Another Brick in the Wall	5
Everything I do	5
Irgendwie, irgendwo, irgendwann	5
Seasons in the Sun	5
Something stupid	5
Funky Town	4
November Rain	4
Whish you were here	4
Ein Kompliment	3
Mad World	3
Smoke on the Water	3
Viva Colonia	3
All good Things	2
Blowing in the Wind	2
Elenor Rigby	2
Satisfaction	2

Tabelle 5.1: Zur Evaluierung verwendete Musikstücke

Unter diesen Voraussetzungen wurden alle möglichen Paarungen an Features und Klassifizierungsmethoden evaluiert. Als Testmenge diente dabei eine kleine Sammlung mit einem Umfang von 89 Songs. Bei der Zusammenstellung wurde einerseits darauf geachtet, unterschiedliche Richtungen innerhalb der Genres Pop und Rock zu repräsentieren und andererseits auch möglichst verschiedene Arten von Versionen abzudecken. Diese reichen von einfachen Live-, über unterschiedlichste Cover- im eigentlichen Sinn bis hin zu Disco-Versionen, in denen das ursprüngliche Musikstück nur mehr rudimentär enthalten ist. Die genaue Zusammensetzung ist in Tabelle 5.1 aufgelistet.

Daraus ergibt sich die Anzahl N_{cover} von 199 sowie N_{pairs} von 3916. Somit folgen unter der Annahme, dass $n = N_{cover} = 199$ Paarungen zufällig aus der Testmenge ausgewählt werden, die Werte $recall^{rand} = 0.05$, $precision^{rand} = 0.05$ und f^{rand} von ebenfalls 0.05.

Allen untersuchten Methoden ist gemeinsam, dass sie eine einzige Maßzahl für die Gleichartigkeit zweier Songs berechnen und diese dann mit einem Schwellwert vergleichen. Liegt die Ähnlichkeit darüber, so werden die Titel als Versionen desselben Musikstücks gewertet – liegt die Ähnlichkeit unter dem Grenzwert, gelten sie als unabhängig. Um einen groben Überblick über alle Varianten der Bewertung zu bekommen, wurde deren Genauigkeit unter Annahme eines einheitlichen *fallout*-Werts von 0.1 gegenübergestellt. Die Resultate sind in Tabelle 5.2 aufgelistet.

THPCP		
Klassifizierung	**recall**	p_{found}
DTW	26%	49
Linear	**44%**	**88**
Diagonale	31%	61
45 Grad	35%	70

THPCP – Refrain		
Klassifizierung	**recall**	p_{found}
DTW	27%	54
Linear	15%	29
Diagonale	29%	58
45 Grad	**30%**	**60**

Melodie		
Klassifizierung	**recall**	p_{found}
DTW	11%	22
Linear	11%	21
Diagonale	**14%**	**27**
45 Grad	8%	15

Melodie – Refrain		
Klassifizierung	**recall**	p_{found}
DTW	4%	8
Linear	13%	25
Diagonale	**21%**	**41**
45 Grad	16%	32

Tabelle 5.2: Ergebnisse der Coverversions-Erkennung bei einem konstanten $fallout$-Wert von 0.1

Feature	**Klassifizierung**	**f-Wert**
THPCP	Linearer Zeitzusammenhang	0.31
THPCP-Refrain	45 Grad Diagonale	0.23
Melodie	Hauptdiagonale	0.16
Melodie-Refrain	Hauptdiagonale	0.14
Zufall		0.05

Tabelle 5.3: Maximale f-Werte der besten Klassifizierungsmethoden pro Feature

Für einen genaueren Vergleich wurde für die beste Klassifizierungsmethode jedes Features eine genaue Precision-Recall-Kurve aufgestellt. Abbildung 5.1 zeigt die Ergebnisse. Daraus wird ersichtlich, dass das ursprünglich bessere Abschneiden des refrain-basierten Melodie-Features Zufall war. Dies wird auch durch den maximal erreichbaren f-Wert bestätigt. Diese Werte sind vergleichend in Tabelle 5.3 aufgelistet.

Bei der Betrachtung der f-Werte fällt auf, dass alle deutlich über der errechneten Baseline von 0.05 liegen. Das einzige der vier Features für das Vergleichswerte vorliegen ist das THPCP-Feature, das in [18] in Verbindung mit einer Dynamic Time Warp-Klassifizierung evaluiert wurde. Dabei ergab sich ein f-Wert von 0.39. Die f-Werte sind jedoch nicht unmittelbar miteinander vergleichbar, da die Musik-Sammlungen auf denen getestet wurde, andere waren. Die Zusammensetzung jener aus [18] ergab eine um 20% höhere Baseline von 0.06, wodurch bereits einen Großteil des Abweichens des Ergebnisses erklärt werden kann. Weiters ist nicht bekannt, wie stark die Variationen zwischen einzelnen Versionen der Songs des Testsets aus [18] waren.

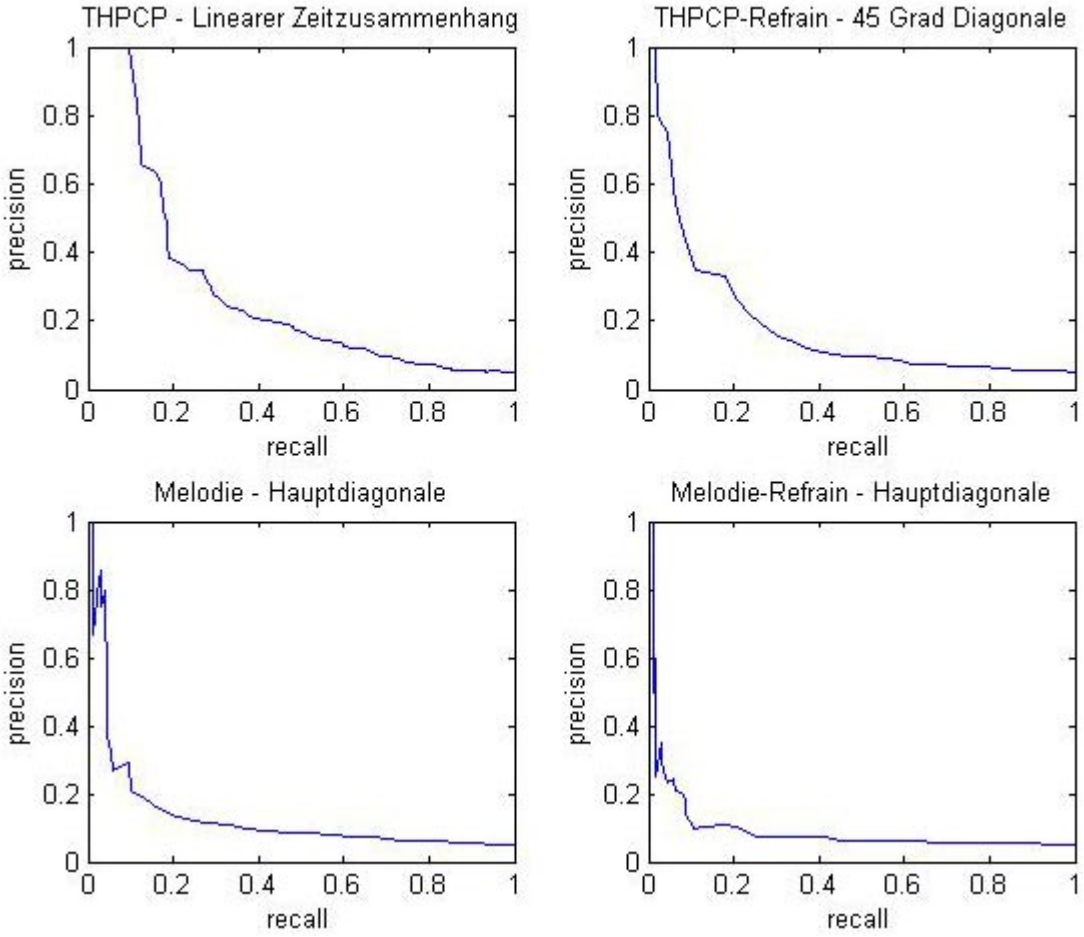


Abbildung 5.1: Gegenüberstellung der Precision-Recall-Kurven der besten Klassifizierungsmethoden pro Feature

5.2 Erkenntnisse

Als wichtigste Erkenntnis kann man sehen, dass eine Cover-Versions- bzw. Plagiats-Erkennung anhand der hier vorgestellten Methoden aus zwei Gründen noch weit von der Praxistauglichkeit entfernt ist.

Zum einen lässt die Zuverlässigkeit aller möglichen Paarungen an Features und Klassifizierungsmethoden zu wünschen übrig. Obwohl alle Maßzahlen deutlich über den jeweiligen Baselines lagen, sind sie für den Einsatz in realen Musiksammlungen mit mehreren tausend im privaten bzw. mehreren Millionen Songs im kommerziellen Bereich unzulänglich. Geht man von dem in der Einleitung genannten Beispiel – dem iTunes Online-Shop[1] mit einem Angebot von über 5 Millionen Musikstücken – aus, so würde ein *fallout* von nur 0.01 mehr als 10^{10} fälschlicherweise als Cover klassifizierte Paarungen zur Folge haben. Dies gilt selbst unter der Annahme, dass zu jedem Ori-

[1] http://www.apple.com/at/itunes/

ginal durchschnittlich fünf Cover-Versionen erhältlich sind. Eine Suche nach Plagiaten beispielsweise kann auf diese Weise nicht unterstützt werden.

Zum anderen stellt die Laufzeit ein Problem dar. Viele der beschriebenen Berechnungen besitzen quadratische Komplexität. Beispiele hierfür sind im Bereich der Feature Extraktion die Strukturanalyse (durch Aufstellen der Selbstähnlichkeits-Matrizen) oder im Bereich der Klassifizierung der Vergleich aller Songs miteinander. Insgesamt betrachtet reichte die Performanz trotz des kleinen Testsets nicht aus, um das Melodie-Feature in seiner ursprünglichen Zeitauflösung zu belassen. Selbst mit der verringerten Auflösung, von nur zwei Frames pro, Sekunde dauerte die Abarbeitung der Testmenge mehrere Stunden.

Ziel dieser Arbeit war jedoch nicht die Entwicklung eines ausgereiften Software-Produkts, sondern der Vergleich unterschiedlicher Herangehensweisen an die Problemstellung. Diesbezüglich ist das schlechte Abschneiden der Klassifizierung anhand des Dynamic Time Warp-Algorithmus überraschend – schließlich ist sie jene Methode, die in der Literatur (vgl. [10] und [18]) empfohlen wird. Die Annahme, dass Tempovariationen über ein gesamtes Stück konstant bleiben, wurde durch das gute Abschneiden der Methode des linearen Zeitzusammenhangs bei Anwendung auf die Gesamt-Features bekräftigt. Sie erlaubt weniger Spielraum als das Dynamic Time Warp-Verfahren, was sich als positiv erwiesen hat.

Eine weitere Überraschung stellt das gute Abschneiden der Klassifizierung unter Verwendung der Hauptdiagonalen bzw. der Geraden mit exakt 45 Grad Steigung innerhalb der Ähnlichkeits-Matrizen dar. Diese wurden nur in die Evaluierung miteinbezogen, um die Bedeutung der Berücksichtigung von Struktur- und Tempo-Variationen abschätzen zu können. Das Heranziehen der Hauptdiagonalen hat sich jedoch als zuverlässiger herausgestellt, als dies erwartet wurde. Das kann daran liegen, dass auch bei Versionen eines Musikstücks die nicht die gleiche Struktur besitzen, dennoch gleiche Fragmente erkannt werden, und diese den Ausschlag für die guten Bewertungen geben.

Vergleicht man die Features, so ist eindeutig festzustellen, dass das THPCP-Feature – sowohl als gesamtes, als auch auf den Refrain beschränkt – das zuverlässigere ist. Der Hauptgrund dafür liegt überwiegend an der Ungenauigkeit der Melodieerkennung. Eine durchschnittliche Erkennungsrate von unter 60% (vgl. Abschnitt 3.6) pro einzelnem Stück ist offensichtlich nicht genau genug um verlässliche Vergleiche durchführen zu können.

Eine Verbesserung des auf den Refrain beschränkten Features im Vergleich zum Feature als Ganzes, konnte nicht erreicht werden. Diese Steigerung der Genauigkeit, gemessen am f-Wert, ist weder beim THPCP-Ansatz noch bei der Melodie zu beobachten.

Grund dafür kann einerseits die Kürze des Features sein. Einzelne Takte, die beispielsweise beim THPCP-Feature aufgrund von gleichen Begleitakkorden ähnliche Chroma-Vektoren ergeben, fallen so mehr ins Gewicht und die Aussagekraft nimmt ab. Andererseits kann auch das fehlerhafte Erkennen von Refrain-Stellen bei Disco-Versionen mit langen Rhythmus-Passagen – wovon einige im Testset vorhanden waren – ein Grund sein.

Als Resumee kann man zusammenfassen, dass sich, von den beiden im Rahmen dieser Arbeit eingeführten Methoden, eine (die Beschreibung eines Musikstücks nur anhand des Refrains) nicht bewähren, die andere (die Klassifizierung unter Annahme einer konstanten Tempoänderungen zwischen zwei Songs) jedoch sehr wohl Verbesserungen bringen konnte. Somit geht das Gesamt-THPCP-Feature in Verbindung mit einer Klassifizierung, die von einem linearen Zeitzusammenhang ausgeht, in dieser Arbeit als die beste Möglichkeit zur Erkennung von Versionen eines Musikstücks hervor.

5.3 Perspektiven

Nicht untersucht wurde hier die Kombination der beschriebenen Features. Möglicherweise lassen sich die Ergebnisse der Versions-Erkennung durch Verbinden von melodie- und harmonie-basierten Merkmalen verbessern.

Von Interesse für zukünftige Arbeiten in diesem Bereich wäre es jedoch, die Feature Extraktion zu verbessern. Alle der vorgestellten Schritte sind mit Ungenauigkeiten behaftet. Eine tiefgreifende Auseinandersetzung und die Suche nach möglichen Verbesserungen für alle von ihnen hätte den Rahmen dieser Arbeit gesprengt. Sie ist jedoch Gegenstand unterschiedlicher Schwerpunkte im Rahmen des *Music Information Retrievals*, wie etwa der Tonartbestimmung oder der Melodieerkennung. Es bleibt abzuwarten, welche Fortschritte in diesen Forschungsbereichen in Zukunft erzielt werden und welche Möglichkeiten sich daraus für die Versions-Erkennung ergeben.

Auch im Finden neuer repräsentativer Features ist Verbesserungspotential zu erwarten. Die automatische Transkription, die sich mit dem Rekonstruieren der Notenblätter anhand von Audio-Signalen beschäftigt, könnte hier einen wertvollen Beitrag leisten. Kann man von einer zuverlässigen symbolischen Repräsentation der Musikstücke ausgehen, ist es wesentlich leichter, aussagekräftige Merkmale zu finden, als rein mit Audio-Signalen als Ausgangspunkt.

Einen ganz anderen Ansatz stellt das Hinzuziehen vorhandener Metadaten, wie etwa der Namen der Stücke, dar. Sind diese bei zwei Songs gleich, so kann mit relativ großer

Wahrscheinlichkeit davon ausgegangen werden, dass es sich um Versionen desselben Musikstücks handelt. Probleme ergeben sich bei dieser Methode jedoch zum einen, wenn die Metadaten unzuverlässig sind. Zum anderen ist in Fällen von Plagiaten nicht anzunehmen, dass ein Musikstück als ganzes mitsamt Liedtext und Titel übernommen wurde. Für solche Fälle kann auch die Berücksichtigung von Metadaten keinen zusätzlichen Nutzen bringen.

Insgesamt ist die Versions-Erkennung ein interessanter Bereich innerhalb des *Music Information Retrievals* in dem noch viel Spielraum für Verbesserungen vorhanden ist. Man darf somit auf zukünftige Entwicklungen gespannt sein.

Literaturverzeichnis

[1] Toshihiko Abe, Takao Kobayashi, and Satoshi Imai. The if spectrogram: A new spectral representation. In *Proceeding of the International Symposium on Simulation, Visualization, and Auralization for Acoustic Research and Education*, pages 423—430, Tokyo, April 1997.

[2] J. Allen. Short term spectral analysis, synthesis, and modification by discrete fourier transform. *Acoustics, Speech, and Signal Processing*, 25(3):235—238, 1977.

[3] M. Atiquzzaman and M. Akhtar. Complete line segment description using the hough transform, 1994.

[4] J.J. Bharucha. Neural nets, temporal composities and tonality. In D. Deutsch, editor, *The psychology of music*, Cognition and Perception, pages 413—440. Adcademic Press, second edition edition, 1999.

[5] B. Blankertz. The constant q transform. draft, 1999.

[6] J. Bonada. Automatic technique in frequency domain for near-lossless time-scale modification of audio. In ICMA: International Computer Music Conference, Berlin, 2000.

[7] J. Brown. Calculation of a constant q spectral transformation, September 1990.

[8] M. Brown, J. und Puckette. An efficient algorithm for the calculation of a constant q transform, 1992.

[9] W. Chai and B. Vercoe. Detection of key change in classical piano music. In *Proceedings: International Conference on Music Informaiton Retrieval*, 2005.

[10] Wai Chai. *Automated Analysis of Musical Structure*. Phd-thesis, Massachusetts Institute of Technology (MIT), 2005.

[11] L. Cohen. *Time-Frequency Analysis*. Prentice Hall, 1995.

[12] R. Dannenberg, W. Birmingham, B. Pardo, N. Hu, C. Meek, and G Tzanetakis. A comparativ evaluation of search techniques for query-by-humming using the musart testbed. *Journal of the American Socieaty for Information Science and Technology*, 58(3), February 2007.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1—38, 1977.

[14] D. Deutsch. The processing of pitch combinations. In D. Deutsch, editor, *The psychology of music*, pages 349—411. Academic Press, second edition edition, 1999.

[15] Karin Dressler. Extraction of the melody pitch contour from polyphonic audio. In *MIREX*, London, September 2006.

[16] T. Fujishima. Realtime chord recognition of musical sound: A system using common lisp music. In *ICMA: International Computer Music Conference*, pages 464–467, Beijing, 1999.

[17] T. Fujishima. Apparatus and method for recognizing musical chords, 2000.

[18] E. Gómez Gutiérrez. *Tonal Description of Music Audio Signals*. Phd thesis, University Pompeu Fabra, 2006.

[19] E. Gómez Gutiérrez. Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing*, 18(3), 2006.

[20] Masataka Goto. A robust predominant-f0 estimation method for real-rime detection of melody and bass lines in cd recordings. In *Proceedings of the IEEE International Converence on Acoustics, Speech and Signal Processing*, pages 757—760, June 2000.

[21] Masataka Goto. A predominant-f0 estimation method for cd recordings: Map estimation using em algorithm for adaptive tone models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3365—3368, May 2001.

[22] Masataka Goto. A chorus-section detection method for musical audio signals. In *Proceedings of the International Conference on Acoustics, Speech, and Signal*

Processing, pages 437—440. IEEE Computer Society, April 2003.

[23] Steven Harford. Automatic segmentation. learning and retrieval of melodies using a self-organizing neural network. In *Proceedings of the ISMIR*, 2003.

[24] C. L. Krumhansl. The cognition of tonality – as we know it today. Hournal of New Music Research, 33(3), 2004.

[25] M. Leman. An auditory model of the role of short term memory in probe-tone ratings. *Music Perception*, 14(4):481—509, 2000.

[26] F. Lerdahl. *Tonal ptich space*. Oxford University Press, Ney York, 2001.

[27] Soren Tjagvad Madsen and Gerhard Widmer. Key-finding with interval profiles. In Proceedings of the International Computer Music Conference (ICMC 2007), Copenhagen, Denmark., 2007.

[28] Soren Tjagvad Madsen and Gerhard Widmer. Towards a computational model of melody identification in polyphonic music. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, Hyderabad, India, 2007.

[29] Matija Marolt. Audio melody extraction based on timbral similarity of meldoic fragments. In *Proceedings of the EUROCON*, November 2005.

[30] Matija Marolt. A mid-level melody-based representation for calculating audia similarity. In *Proceedings of the ISMIR*, pages 280—285, 2006.

[31] G. Martens, H.D. Meyer, B.D. Beats, and M. Leman. Distance-based versus tree-based key recognition in musical audio, 2004.

[32] M. Meyer. *Signalverarbeitung - Analoge und digitale Signale, Systeme und Filter*. Vieweg, 3. aufl. edition, 2003.

[33] A.V. et.al. Oppenheim. *Zeitdiskrete Signalverarbeitung*. Pearson Studium, 2. aufl. edition, 2004.

[34] Universität Osnabrück. Fachbereich musik. Online: http://www.musik.uni-osnabrueck.de, Stand 2007.

[35] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62—66, March 1979. minimize inter class variance.

[36] T. Painter and A. Spanias. Perceptual coding of digital audio. In *Proceedings of the IEEE*, volume 88, pages 451—513. IEEE, IEEE Computer Society, April 2000.

[37] Elias Pampalk. *Computational Models of Music Similarity and their Application in Music Information Retrieval*. PhD thesis, Technische Universität Wien, 2006.

[38] Isabelle Peretz, Lise Gagnon, Sylvie Hébert, and Joël Macoir. Singing in the brain - insights from cognitive neuropsychology. *Music Perception*, 21(3):373—390, 2004.

[39] Alan W. Pollak. A musicological analysis of all beatles' songs. (Online: http://www.icce.rug.nl/ soundscapes/), 1989—2000 (Stand: August 2007).

[40] J. Princen, J. Illingworth, and J. Kittler. A formal definition of the hough transform: properties and relationships. *Journal of Mathematical Imaging and Vision*, Vol. 1:153—168, 1992.

[41] A. Röbel. Transient detection and preservation in the phase vocoder. Proc. Int. Computer Music Conference (ICMC'03), pp.247-250, Singapore, 2003.

[42] Thomas D. Rossing. *The science of sound*. Addison-Wesley, 2nd edition edition, 1990.

[43] X. und Smith J. Serra. Spectral modeling synthesis: A sound analysis/synthesis based on deterministic plus stochastic decomposition. *Computer Music Journal*, Vol. 14(Nr. 4):12–24, Winter 1990.

[44] K. Steiglitz. *Digital Signal Processing Primer - With Applications to Digital Audio and Computer Music*. Addison-Wesley, 1996.

[45] D. Temperley. What's key for key? – the krumhansl-schmuckler key finding algorithm reconsidered. Music Perception, 17(1), 1999.

[46] E. Terhardt. Calculating virtual pitch. *Hearing Research*, 1:155—182, 1979.

[47] P. Toiviainen and C.L. Krumhansl. Measuring and modeling real-time responses to music: The dynamics of tonality induction. Perception, 2003.

[48] Steven van de Par, Armin Kohlrausch, Richard Heusdens, Jesper Jensen, and Søren Holdt Jensen. A perceptual model for sinusoidal audio coding based on spectral integration. *EURASIP Journal on Applied Signal Processing*, 2005(9):1292–1304, 2005. doi:10.1155/ASP.2005.1292.

[49] Yongwei Zhu, Mohan S. Kankanhalli, and Sheng Gao. Music key detection for musical audio. In *MMM '05: Proceedings of the 11th International Multimedia Modelling Conference (MMM'05)*, pages 30–37, Washington, DC, USA, 2005. IEEE Computer Society.

curriculum vitæ

Bernhard Niedermayer, BSc.

b.niedermayer@inode.at

Geboren am	28. Juli 1983
Familienstand	Ledig
Staatsbürgerschaft	Österreich
Adresse	Prennlehnerweg 20
	A-4060 Leonding

Ausbildung und Studium

1992–2001	BRG Landwiedstraße (Linz)
2001–2005	Bakkalaureatsstudim Informatik an der Johannes Kepler Universität Linz (JKU)
2005	Auslandssemester an der Oxford Brookes University (England)
2005–	Studium der Wirtschaftsinformatik an der Johannes Kepler Universität Linz
2006–	Master-Studium Informatik an der Johannes Kepler Universität Linz

Praxis

2002–2006	Unterstützung des oberösterreichischen Leichtathletikverbands im EDV-Bereich bei Veranstaltungen
2004–2005	Entwicklung von eLearning-Materialien für das Institut für Informationsverarbeitung und Mikroprozessortechnik an der JKU
2004–2006	Arbeit als Tutor am Institut für Angewandte Wissensverarbeitung an der JKU
2006	Entwicklung von eLearning-Materialien für die Oxford Brookes University
2006–2007	Abhaltung von (Programmier-)Schulungen für die Firma Integanet

Fremdsprachenkenntnisse

Englisch	Sehr gut in Wort und Schrift
Spanisch	Grundkenntnisse

Auszeichnungen

Maturiert mit ausgezeichnetem Erfolg

Abschluss des Bakkalaureatsstudiums mit ausgezeichnetem Erfolg

Mehrmalige Auszeichnung mit einem Leistungsstipendiumg während des Studiums

Viermalige Teilnahme an der Österreichischen Chemie-Olympiade, bestes Resultat: 9. Rang (2001)

Linz, 15. September 2007

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Linz, am 17. September 2007

Bernhard Niedermayer