

Ein automatisches Verfahren zur Notenmitverfolgung polyphoner Gitarrenmusik



Bachelorarbeit

im Studiengang Angewandte Informatik der Fakultät Wirtschaftsinformatik
und Angewandte Informatik der Otto-Friedrich-Universität Bamberg

Themensteller: Prof. Dr. Diedrich Wolter

vorgelegt von: Moritz Dück
Matrikelnummer: 1860589

Zusammenfassung

Ein Notenmitverfolgungssystem schätzt simultan zur menschlichen Performance eines Musikstücks die aktuelle Position in den dazugehörigen Noten. Im Gegensatz zu anderen Notenmitverfolgungssystemen wird in dieser Arbeit die für Gitarrenbegleitungen übliche Akkordnotation statt der sonst gebräuchlichen Notenschrift als Notationsgrundlage verwendet. Diese enthält nur grobe zeitliche Informationen zu den Tonhöhen, sodass das System verstärkt metrische Aspekte berücksichtigt. Es wird ein Tempomodell mithilfe eines Kalman-Filters modelliert, das den zugrundeliegenden Puls der Musik erkennen soll. Zur Notenmitverfolgung wird ein Hidden Markov Model auf Basis einer Akkorderkennung genutzt, dessen zeit-diskrete Update-Zeitpunkte von dem Puls der Temposchätzung vorgegeben werden. Die durchschnittliche Erkennungsrate des formulierten Modells bleibt hinter den Ergebnissen von State-Of-The-Art-Systemen zurück. Aus diesen Resultaten können entscheidende Herausforderungen bei der Quantisierung von Notenlängen und der zeitlich kohärenten Modellierung eines Zustandsmodells gefolgert werden.

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 1 |
| 2 | Musikalische Grundlagen | 3 |
| 2.1 | Zeitlicher Aufbau von Musik | 3 |
| 2.2 | Tonale Grundlagen — Der Aufbau von Akkorden | 5 |
| 2.3 | Gitarrenspezifische Tonentwicklung | 6 |
| 3 | Hintergrund | 8 |
| 3.1 | Hidden Markov Models als Zustandsschätzer | 8 |
| 3.2 | Funktionsweise eines Kalman-Filters | 9 |
| 3.3 | Der verfolgte Ansatz im Kontext anderer Notenmitverfolgungssysteme | 10 |
| 4 | Rahmenbedingungen der Notenmitverfolgung | 12 |
| 4.1 | Die gegebenen Noten | 12 |
| 4.2 | Aspekte menschlicher Performance | 12 |
| 4.2.1 | Fehler | 13 |
| 4.2.2 | Betonungen und Ausdruck | 13 |
| 4.2.3 | Tempo und Timing | 13 |
| 5 | Aufbau des Notenmitverfolgungssystems | 14 |
| 5.1 | Feature-Erkennung | 15 |
| 5.1.1 | Ansatzerkennung | 15 |
| 5.1.2 | Akkorderkennung | 18 |
| 5.2 | Geschwindigkeitsschätzung mithilfe eines Kalman-Filters | 20 |
| 5.2.1 | Quantisierung der Notenstellen | 21 |
| 5.2.2 | Modellparameter des Kalman-Filters | 22 |
| 5.3 | Die Mitverfolgung mit einem HMM | 25 |
| 6 | Evaluation | 27 |
| 6.1 | Technische Umsetzung des Modells | 27 |
| 6.2 | Evaluation der Geschwindigkeitsschätzung | 29 |
| 6.3 | Evaluation der Notenmitverfolgung | 31 |
| 6.3.1 | Evaluierungsmethodiken für Notenmitverfolgungssysteme | 31 |
| 6.3.2 | Anpassung des Evaluierungs-Frameworks an die Akkordnotation | 31 |
| 6.3.3 | Testaufbau | 32 |
| 6.3.4 | Ergebnisse | 33 |
| 6.3.5 | Diskussion | 33 |
| 7 | Fazit | 35 |
| A | Ausführliche Ergebnisse der Evaluation | 39 |
| B | Implementierung der Audioverarbeitung mittels Essentia | 46 |

1 Einleitung

Im Internet oder in Liederbüchern sind für viele Musikstücke Noten für die Begleitung auf der Gitarre zu finden. Dargestellt werden dabei der Text, annotiert mit den zu spielenden Akkorden, sowie allgemeine Informationen zu dem Stück, wie etwa dem Tempo oder einem passenden Schlagrhythmus. Ein Beispiel findet sich in Abb. 1. Gedacht ist diese Art der Notation für die spontane Begleitung von Gesang, wobei der Gitarrist¹ auch ohne vorheriges Üben in der Lage ist, den gegebenen Rhythmus und die Akkordabfolge zu spielen. Aktuelle Versionen dieser Noten sind statisch, eine interaktive Version dieser Darstellung könnte automatisch den zu spielenden Akkord hervorheben und die Noten mitverfolgen, und so dem Gitarristen eine bessere Spielerfahrung bieten. Da jedoch das Tempo einer menschlichen Performance variiert, reicht das Abspielen der Noten in der Originalgeschwindigkeit, wie etwa bei Karaoke, nicht aus. Eine praktikable Notenmitverfolgung erfordert deshalb den Abgleich der gespielten Noten mit den vorgegebenen Noten, da nur so die Noten mit dem Spieler synchron gehalten werden können.

Ein anderer Anwendungsfall eines solchen funktionierenden Systems ist eine künstliche Begleitung, die passend zu dem Audiosignal des Gitarristen andere Musikelemente (Schlagzeug, Klavier oder Bass) beifügt, wie es Dannenberg (1984) und Vercoe (1984) in den ersten Arbeiten zur Notenmitverfolgung beschreiben.

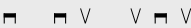
| | |
|--|---|
| $\frac{4}{4}$ 150bpm  [Intro] G C G D G C G D [Verse] G C G D Hey where did we go, days when the rains came G C G D Down in the hollow, playin' a new game G C G D Laughin' and a-runnin' hey, hey, skippin' and a-jumpin' G C G D C In the misty mornin' fog with, ah, our hearts thumpin' with you D G Em C D G D7 My brown-eyed girl, you're my brown-eyed girl | Taktart & Tempo Schlagrhythmus Text und Akkorde |
|--|---|

Abbildung 1: Exemplarische Darstellung des Liedes *Brown Eyed Girl* von *Van Morrison* in üblicher Akkordnotation für die Gitarrenbegleitung von der Internetseite *Ultimate Guitar*.²

¹Im Folgenden sollen bei sämtlichen Bezeichnungen sowohl die männliche als auch die weibliche Form eingeschlossen sein.

²https://tabs.ultimate-guitar.com/tab/van_morrison/brown_eyed_girl_chords_76882 (zuletzt aufgerufen am 07.09.2019).

Innerhalb der vorliegenden Arbeit soll ein Notenmitverfolgungssystem basierend auf einem *Hidden Markov Model*² (HMM) umgesetzt werden. Diese Modelle sind in der Vergangenheit bereits häufig für die Aufgabe der Mitverfolgung genutzt worden und stellen einen flexiblen sowie etablierten Ansatz zur probabilistischen Modellierung dar. Zur Unterstützung des zeitlich-diskreten Modells wird eine Geschwindigkeitsschätzung basierend auf einem *Kalman-Filter* umgesetzt.

Ziel der Arbeit ist es zum einen, Potenziale und Schwierigkeiten des gewählten Ansatzes zu benennen, um daraus andererseits allgemeine Herausforderungen einer robusten Notenmitverfolgung abzuleiten. Dabei steht vor allem die unsichere Notengrundlage der Akkordnotation im Fokus.

Um ein solches Notenmitverfolgungssystem zu entwickeln, wird zunächst notwendiges musikalisches Hintergrundwissen in Abschnitt 2 eingeführt. Darauf folgt ein Überblick über die für die Erkennung benötigten Techniken und eine Einordnung der Arbeit in die wissenschaftliche Disziplin der Notenmitverfolgung (Abschnitt 3). Mit diesen Grundlagen wird die Problemstellung in 4 präzisiert. Das entwickelte Modell sowie die zugehörigen Schritte der Audioverarbeitung werden in Abschnitt 5 dargelegt. In der abschließenden Evaluation wird das entwickelte Notenmitverfolgungssystem getestet und die Ergebnisse diskutiert.

²zu deutsch: *verdecktes Markowmodell*.

2 Musikalische Grundlagen

Notenmitverfolgungssysteme nutzen Hintergrundwissen aus der Musiktheorie, etwa bei der Analyse der Audiodaten oder bei der Erstellung des probabilistischen Modells basierend auf den Noten. Für das vorliegende System relevante musikalische Grundlagen werden hier eingeführt.

Musik ist einerseits geprägt von zeitlichen Beziehungen zwischen musikalischen Einzelereignissen, andererseits von der Wahrnehmung von Tönen. Diese beiden Aspekte werden in der Diskussion nacheinander betrachtet und relevante Aspekte herausgestellt. Abschließend werden Spezifika der Gitarre hervorgehoben, um den Besonderheiten des Instruments in der Erkennung Rechnung zu tragen.

2.1 Zeitlicher Aufbau von Musik

Die zeitliche Anordnung von Noten ist nicht zufällig, sondern folgt in der Musik meist geordneten Strukturen: Noten liegen meist auf einem Raster und Notenzustellungen stehen in einfachen Verhältnissen zueinander (1:1, 1:2). Zusätzlich werden sich wiederholende Muster für die Anordnung verwendet, die eine gewisse Regelmäßigkeit aufbauen.

Das zeitliche Verhältnis zwischen Noten wird im Begriff des *Rhythmus* gefasst, umschließt jedoch eine Vielzahl an Aspekten, sodass eine klare Definition schwer zu finden ist (Deutsch, 2013). In dieser Arbeit werden lediglich für die Erkennung relevante Aspekte von Rhythmen eingeführt.

Ansatz Der *Ansatz* ist der Beginn eines musikalischen Ereignisses und umfasst Eigenschaften wie den Ansatzzeitpunkt oder die Stärke des Ansatzes. Beim Ansatzzeitpunkt unterscheidet man zwischen der Notenansatzzeit, der akustischen sowie der wahrgenommenen Ansatzzeit (Lerch, 2012):

1. *Notenansatzzeit*: Der Zeitpunkt, an dem das Instrument zur Tonerzeugung angeregt wird.
2. *Akustische Ansatzzeit*: Der frühestmögliche Zeitpunkt, an dem ein Signal oder ein akustisches Ereignis messbar wird.
3. *Wahrgenommene Ansatzzeit*: Der frühestmögliche Zeitpunkt, an dem das Signal von einem Hörer wahrgenommen werden kann.

Die Notenansatzzeit ist per Definition nicht messbar und deshalb für die Analyse nicht relevant. Für die spätere Diskussion ist jedoch wichtig, dass die akustische Ansatzzeit nicht der wahrgenommenen Ansatzzeit entspricht. Darüber hinaus diskutiert Lerch (2012) verschiedene Erkenntnisse zu der Genauigkeit des Menschen beim Erkennen von Ansätzen und postuliert zusammenfassend einen minimalen mittleren absoluten Fehler von 5-10 ms. Dieser Wert kann auch als ein möglicher Richtwert für die benötigte Genauigkeit der Notenmitverfolgung verstanden werden, da eine genauere Auflösung keine für den Menschen relevante Verbesserung darstellt.

Tempo, Zählzeit, Takt und Tatum Das musikalische Tempo misst den Fortschritt in den Noten pro Zeiteinheit. Als Messeinheit des Tempos in Musikstücken werden häufig die Anzahl der Schläge pro Minute in *bpm* (*beats per minute*) gemessen. Dieses Tempo entspricht typischerweise der Geschwindigkeit, in der ein Mensch mit seinem Fuß zur Musik wippen würde.

Man kann verschiedene solcher Tempi durch ganzzahlige Vielfache finden, die der Musik einen passenden Puls zuordnen. Jede passende Geschwindigkeit ist ein *Puls-Level*. Abb. 2 zeigt verschiedene Puls-Level im Bereich von sechzehntel bis halben Noten für einen gegebenen Gitarren-Schlagrhythmus.

Die *Zählzeit* bildet den Grundstein für das Zählen in *Takten*. Sie ist das Puls-Level, auf das sich Noten in ihrer Notation beziehen. In diesem Fall entspricht die Zählzeit dem Level der Viertelnoten.

Ein *Takt* ist ebenfalls ein Puls-Level, zeichnet sich aber durch eine regelmäßige Wiederholung von gruppierten Musikelementen aus. Normalerweise umfasst ein Takt 3 bis 7 Zählzeiten (Lerch, 2012). Takt und Zählzeit sind deshalb in der musikalischen Komposition relevant, da hier neben der bisher dargelegten binären Unterteilung beispielsweise auch ternäre Verhältnisse möglich sind.

Die Angabe der *Taktart* eines Musikstücks erfolgt in einer Bruchnotation (z. B. $\frac{3}{4}$), wobei der Zähler die Anzahl der Zählzeiten pro Takt angibt. Der Nenner legt die Dauer einer Zählzeit als Notenwert fest. Die Dauer einer Zählzeit in Abb. 2 entspricht einer Viertelnote, der notierte Schlagrhythmus wiederholt sich alle vier Zählzeiten. Durch diese Regelmäßigkeit — vier Zählzeiten pro Takt, wobei eine Zählzeit eine Viertelnote ist — passt der gezeigte Rhythmus zur Taktart der Viervierteltakte $\frac{4}{4}$.

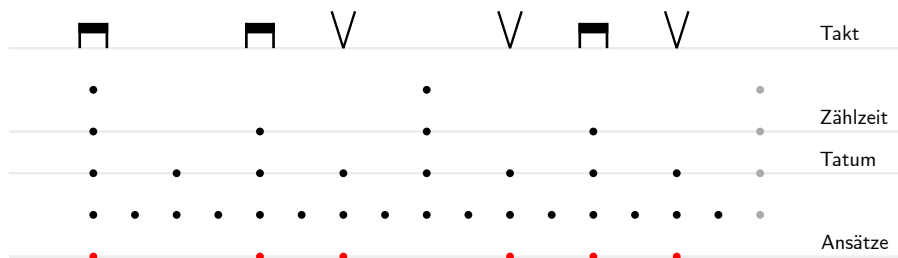


Abbildung 2: Ein Viervierteltakt notiert mit verschiedenen Puls-Levels (Punkte), einem exemplarischen Gitarrenrhythmus (oben) und den hörbaren Ansätzen (rote Punkte) (Uhle and Herre, 2003).

Uhle and Herre (2003) führen neben dem Takt und der Zählzeit noch ein weiteres relevantes Level auf, das *Tatum*. Bilmes (1993) führt den Begriff ein und beschreibt ihn entweder als den kleinsten Abstand zwischen zwei Noten oder allgemeiner als die Zeitunterteilung, die sich am besten mit allen Ansätzen deckt. Das Tatum liegt im Bereich einer binären, ternären oder quaternären Unterteilung der Zählzeit und soll eine atomare zeitliche Einheit bilden. Der Vorteil einer solchen noch feineren zeitlichen Untergliederung besteht darin, dass man das Tatum als rhythmisches Raster oder Quantisierungsgröße verstehen kann (Lerch, 2012).

Einzelne Pulse sind meist nicht in direkter Form in der Musik vorhanden, sodass der globale Puls nur indirekt über andere wahrnehmbare Aspekte der Musik erkennbar ist (Uhle and Herre, 2003). Kann man die Frequenz der zugrundeliegenden Pulse richtig erkennen und erfolgreich einem Puls-Level zuordnen, kennt man die aktuelle Geschwindigkeit der Musik. Diese Zuordnung gilt es in der Aufgabe der Tempoerkennung zu leisten.

2.2 Tonale Grundlagen — Der Aufbau von Akkorden

Die Notation für Gitarrenbegleitung verwendet Akkorde zum Abbilden der tonalen Entwicklung eines Musikstücks. Akkorde entstehen durch die Gruppierung von einzelnen Noten in einen gemeinsamen Klang. Der Aufbau von Akkorden und die Verwendung derselben in der vorliegenden Arbeit sollen im Folgenden erläutert werden.

Wahrnehmung von Tonhöhen — Die Tonleiter Die menschliche Wahrnehmung der Tonhöhe ist abhängig von der Höhe der Frequenz eines Tones. Je höher die Frequenz, desto höher wird der Ton wahrgenommen. Der Zusammenhang ist nicht-linear und wird deshalb durch Modelle wie etwa die *Mel-Skala* abgebildet. Daneben nimmt der Mensch Töne, die in einem Verhältnis von 2^n zueinanderstehen (2,4,8...) als sehr ähnlich wahr. Da sich dadurch nach jeder Verdopplung der Frequenz eine Wiederholung der wahrgenommenen Tonhöhen ergibt, wird jedes dieser Intervalle als eine *Oktave* bezeichnet. Jede Oktave wird in 12 *Noten* aufgeteilt, die zusammen die *Tonleiter* ergeben (Lerch, 2012):

| | | | | | | | | | | | |
|---|-------|---|-------|---|---|-------|---|-------|----|-------|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| C | C#/Db | D | D#/Eb | E | F | F#/Gb | G | G#/Ab | A | A#/Bb | B |

Der Abstand zwischen zwei Noten in der Tabelle entspricht einem *Halbtonschritt*. In dieser Arbeit wird die amerikanische Schreibweise mit der Note B statt H verwendet, da dies für die Gitarrennotation üblich ist.

Akkorde Die meisten grundlegenden Akkorde bestehen aus drei Noten und werden von einer Grundnote ausgehend gebildet. Die beiden wichtigsten Gruppen für die Gitarrenbegleitung sind die *Dur-* (*maj*) und *Moll-* (*min*) Akkorde, die auch für die Erkennung genutzt werden sollen.

Ein C-Dur Akkord ist ein Dur-Akkord mit der Grundnote C. Die beiden anderen Noten, die den Akkord ergänzen, sind bei einem Dur-Akkord vier Halbtonschritte und sieben Halbtonschritte erhöht von der Grundnote zu finden. Für einen C-Dur Akkord ergibt das einen Dreiklang aus den Noten C, E und G.

Bei einem Moll-Akkord sind die beiden anderen Noten drei und sieben Halbtonschritte von der Grundnote entfernt. Damit entspricht ein C-Moll Akkord den Einzelnoten C, Eb und G.

Daneben gibt es noch die selteneren *übermäßigen Akkorde* (*aug*) (von der Grundnote aus vier und acht Halbtonschritte erhöht) und *verminderte Akkorde* (*dim*) (von der Grundnote aus drei und sechs Halbtonschritte erhöht). Ebenso ist es üblich, die Dreiklänge um weitere Noten zu ergänzen, was häufig durch zusätzliche Zahlen in der Notation verdeutlicht wird (Clendinning and Marvin, 2016).

Abbildung 3: Exemplarische Schlagrhythmen der Gitarre.

| | ■ Abschlag | ∨ Aufschlag |
|-----|-----------------|-------------|
| 4/4 | 1 + 2 + 3 + 4 + | |
| | ■ | ■ ∨ |
| | ■ | ■ ∨ |
| | ■ | ■ |
| | ■ | ■ |
| 3/4 | 1 + 2 + 3 + | |
| | ■ | ■ ∨ |
| | ■ | ■ |
| | ■ | ■ |

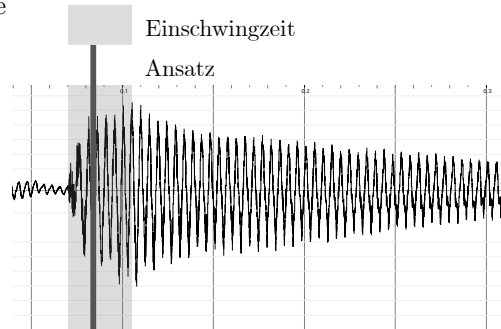


Abbildung 4: Audiosignal eines Anschlags einer Gitarre.

Für die angestrebte Erkennung relevant sind lediglich die zu Beginn eingeführten Dur- und Moll-Akkorde. In Anlehnung an die *MIREX Audio Chord Detection Challenge* aus dem Jahr 2008³, einer Evaluation verschiedener Akkorderkennungssysteme, werden modifizierte Akkorde wie folgt den Basisfällen zugeordnet:

- Dur : maj, dim, aug, maj7, 7, dim7, hdim7, maj6, 9, maj9, sus4, sus2
- Moll : min, min7, minmaj7, min6, min9

Die Notation für die Gitarrenbegleitung notiert immer die Grundnote des Akkords, ohne weitere Ergänzung ist dann ein Dur-Akkord gemeint (C = C-Dur), Moll-Akkorde werden meist mit einem kleinen -m identifiziert (Cm = C-Moll). Für diese Arbeit verwendete Noten werden zunächst nach dem oben genannten Schema normalisiert und in die eben genannte Notation überführt. So entsprechen die Noten in Abb. 1 bereits größtenteils den Vorgaben, lediglich der D7-Akkord zum Ende des Ausschnitts würde für die Erkennung in einen D-Akkord überführt werden.

2.3 Gitarrenspezifische Tonentwicklung

Die Notenmitverfolgung in dieser Arbeit soll ausschließlich Gitarrenmusik verarbeiten, wodurch sich für die erwartete Tonspur einige Charakteristika ergeben, die knapp erläutert werden sollen.

Spielarten der Gitarre Gängige Spielweisen der Gitarre in westlicher Gitarrenmusik sind *Zupfen* und *Streichen*. Beim Zupfen wird jede Saite einzeln durch einen Finger angeschlagen. Beim Streichen werden durch die Bewegung der Finger oder eines Plektrums von oben nach unten oder von unten nach oben über die Saiten mehrere Töne schnell aufeinanderfolgend erzeugt, was als musikalisches Einzelereignis eines Anschlags wahrgenommen wird. Die einfache Gitarrenbegleitung basiert häufig auf dem Streichen der Saiten, weshalb diese Spielweise Grundlage der hier entwickelten Erkennung sein soll.

³https://www.music-ir.org/mirex/wiki/2008:Audio_Chord_Detection (zuletzt aufgerufen am 07.09.2019).

Gitarrenrhythmen Beim Streichen ist es entweder möglich Abschläge — durch das Streichen von oben nach unten — oder Aufschläge — durch das Streichen von unten nach oben — zu spielen. Die Notation eines Gitarrenrhythmus hält fest, zu welchen Zeitpunkten in einem Takt ein Auf- oder Abschlag gespielt werden soll. Exemplarische Rhythmen in verschiedenen Taktarten sind in Abb. 3 dargestellt. Während der genaue Rhythmus variiert, sollen die Beispiele einen Eindruck über zu erwartende musikalische Ereignisse in einer Aufnahme vermitteln.

Tonentwicklung der Gitarre beim Streichen Die Gitarre erzeugt Töne durch das Anschlagen von Saiten. Die *Einschwingzeit* beträgt bei dem Beispiel in Abb. 4 ungefähr 35 ms und zeichnet sich durch einen hohen Anteil an Rauschen und einer hohen Energie im Signal aus, was durch die Mechanik eines Saiteninstruments bedingt ist (Lerch, 2012). Der Ton klingt so lange nach, bis der Spielende das Schwingen der Saiten stoppt. Dies passiert oft erst durch den nächsten Anschlag, sodass nach der Einschwingzeit die gespielte Tonhöhe gemessen werden kann.

3 Hintergrund

Der ausgewählte Ansatz zur Notenmitverfolgung basiert auf einem Hidden Markov Model und einem Kalman-Filter. Beide Techniken werden zunächst eingeführt, bevor die Arbeit in den wissenschaftlichen Kontext eingeordnet wird.

3.1 Hidden Markov Models als Zustandsschätzer

Das Notenmitverfolgungssystem wird in dieser Arbeit mithilfe von Hidden Markov Models umgesetzt. Ein Hidden Markov Model bildet einen Prozess mithilfe von Zuständen $S = \{S_1 \dots S_n\}$ und Beobachtungen O ab. Eine versteckte Abfolge dieser Zustände $x_1 \dots x_T$ generiert die von außen wahrgenommenen Beobachtungen $y_1 \dots y_T$ (siehe Abb. 5). Ein Hidden Markov Model besteht aus zwei stochastischen Modellen: einem *Prozessmodell* und einem *Beobachtungsmodell*. Beide folgen der Markoweigenschaft. Das Prozessmodell formuliert, mit welchen Wahrscheinlichkeiten Zustandsübergänge stattfinden, während das Beobachtungsmodell die Wahrscheinlichkeit beschreibt, eine Beobachtung gegeben eines Zustands zu machen.

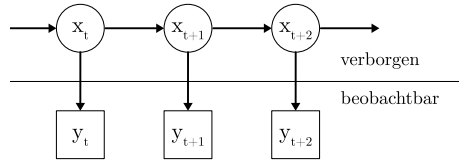


Abbildung 5: Zeitliche Entwicklung eines Hidden Markov Models.

Der Zustandsübergang ist durch eine Wahrscheinlichkeitsverteilung $P(X_{t+1} = S_j | X_t = S_i) \equiv a_{ij}, \forall i, j. \quad 1 \leq i, j \leq N$ definiert, wobei der folgende Zustand nur von dem aktuellen Zustand abhängt (erste Markoweigenschaft). Daneben wird jedem Zustand eine Verteilung über Emissionswahrscheinlichkeiten zugeordnet $p(Y_t = o | X_t = S_j) \equiv b_j(o), \forall j, o. \quad 1 \leq j \leq N \wedge o \in O$, die ebenso nur von dem aktuellen Zustand abhängt (zweite Markoweigenschaft). Zur vollständigen Definition des Modells λ fehlt nur noch eine initiale Zustandsverteilung $P(X_1 = S_j) \equiv \pi_j, \forall j. \quad 1 \leq j \leq N$.

Mithilfe des Vorwärts-Algorithmus kann, gegeben einer Sequenz von Beobachtungen $y_1 \dots y_t = y_1^t$, die Wahrscheinlichkeit jedes Zustands zum Zeitpunkt t berechnet werden. Die *Vorwärtsvariablen* $\alpha_t(j) = P(y_1^t, X_t = S_j | \lambda)$ geben diese Größe an und lassen sich wie folgt berechnen (Rabiner, 1989):

zum Zeitpunkt $t = 1$:

$$\alpha_1(j) = \pi_j * b_j(y_1)$$

Für alle folgenden Zeitpunkte $t = 2 \dots T$:

$$\alpha_t(j) = \left[\sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] * b_j(y_t)$$

3.2 Funktionsweise eines Kalman-Filters

Im weiteren Verlauf der Arbeit wird ein Kalman-Filter genutzt, um eine Geschwindigkeitsschätzung des Spielenden zu errechnen.

Mithilfe eines Kalman-Filters wird versucht, anhand von Messungen den Zustand eines Systems zu schätzen und dabei die Unsicherheit in den Messwerten zu berücksichtigen. Dabei entwickelt sich der modellierte Zustand nach den Gleichungen eines linearen Systems, dem *Bewegungsmodell*, und die Beobachtungen sind durch lineare Transformationen mit den Zuständen verbunden, dem *Beobachtungsmodell*. Unsicherheiten werden durch Gaußsches und weißes Rauschen modelliert.

Das Fortschreiben des Zustands durch das Bewegungsmodell überführt den Zustand x_{t-1} in eine *a-priori-Schätzung* \hat{x}_t zum Zeitpunkt t . Die Varianz $\hat{\Sigma}_t$ kann durch die lineare Transformation ebenso direkt berechnet werden:

$$\begin{aligned}\hat{x}_t &= A_t x_{t-1} \\ \hat{\Sigma}_t &= A_t \Sigma_{t-1} A_t^T + R_t\end{aligned}$$

Durch eine Messung kann die Unsicherheit in der Schätzung verringert werden, dieses Update führt zu der *a-posteriori-Schätzung* x_t .

Das Ziel des Filters ist es, aus der Linearkombination einer a-priori-Schätzung \hat{x}_t und einer gewichteten Differenz zwischen einer Messung z_t und einer erwarteten Messung $C\hat{x}_t$ eine optimale a-posteriori-Schätzung des Zustands x_t zu erhalten, siehe (1) (Welch, 1997).

$$x_t = \hat{x}_t + K(z_t - C\hat{x}_t) \quad (1)$$

$$\Sigma_t = (I - K_t C_t) \hat{\Sigma}_t \quad (2)$$

Dabei entscheidet der *Kalman-Gain* K über die Gewichtung zwischen Vorhersage und Messung. Dieser wird so gewählt, dass er den Fehler der Schätzung statistisch minimiert. Würden bei einer unendlichen Wiederholung des selben Experiments mehrere Filter verglichen werden, lieferte der Kalman-Filter im Mittel die besten Ergebnisse (Maybeck, 1979).

$$K_t = \hat{\Sigma}_t C_t^T (C_t \hat{\Sigma}_t C_t^T + Q_t)^{-1} \quad (3)$$

Mit den obenstehenden Gleichungen ist es möglich, den Zustand eines Systems rekursiv zu schätzen. Dabei ist die Reihenfolge, in der Fortschreibe- oder Korrekturschritte vorgenommen werden, nicht festgelegt und kann nach Bedarf angepasst werden.

3.3 Der verfolgte Ansatz im Kontext anderer Notenmitverfolgungssysteme

Einen Überblick über existierende Ansätze und Modelle zur Notenmitverfolgung liefert Cuvillier (2016) oder zuvor Orio et al. (2003). Die Autoren unterscheiden zwei vorherrschende Lösungsansätze. Die erste Gruppe versucht den Abgleich zwischen Noten und Audiosignal herzustellen, indem beide als Zeitreihen betrachtet werden, die über eine Kostenfunktion bestmöglich abgeglichen werden können. Die zweite Gruppe nutzt probabilistische Modelle, um die Noten abzubilden und die Position in diesen Noten anhand des Audiosignals zu schätzen. Das hier angestrebte Notenmitverfolgungssystem soll auf einem Hidden Markov Model basieren und zählt damit zu den probabilistischen Ansätzen.

Dabei wird angenommen, dass Beobachtungen $y_1 \dots y_T$ (z.B. angeschlagene Akkorde) einem stochastischen Prozess $\{Y_t\}$ entspringen, der von einer Folge an Zuständen $x_0 \dots x_T$ (der aktuellen Position in den Noten) der Zufallsvariable $\{X_t\}$ generiert wird. Das Problem der Notenmitverfolgung ist dann die Umkehrung dieser Hypothese: das Finden der plausibelsten Folge an Zuständen (Notenpositionen), gegeben der Beobachtungen (Akkorde) (Cont, 2009).

Der Großteil bisheriger Arbeiten zu Notenmitverfolgungssystemen basieren auf der *Notenschrift*. Die Besonderheit der vorliegenden Arbeit besteht darin, dass hier die für die Gitarrenbegleitung übliche Akkordnotation zugrunde gelegt wird, welche keine Informationen über die genaue Abfolge von Einzelnoten beinhaltet. Lediglich auf der höheren Abstraktionsebene der Takte, also in einem größeren zeitlichen Raster, sind die erwarteten Akkorde notiert. Diese Akkordangaben bilden die Grundlage für die beabsichtigte Notenmitverfolgung. Mangels einer konkreten Angabe der Einzelnoten können sich verschiedene Interpretationen der selben Noten stark unterscheiden. Ein Beispiel für Notenschrift und einen Eindruck über die im Vergleich zur Akkordnotation unterschiedliche Informationsdichte liefert Abb. 6. Während die Mitverfolgung monophoner Musikstücke die einfachere Aufgabe darstellt, wird in dieser Arbeit direkt von polyphoner Musik ausgegangen.



Abbildung 6: Exemplarischer Vergleich der Informationsdichte der Akkordnotation (oben) und der Notenschrift (unten) am Beispiel eines Ausschnitts aus dem Lied *Here Comes the Sun* von *The Beatles*.⁵

⁵<https://www.musicnotes.com/sheetmusic/mtd.asp?ppn=MN0104281> (zuletzt aufgerufen am 07.09.2019).

Typisch für die Gitarrenbegleitung ist ein hoher Anteil pulsgebender Elemente — einzelne musikalische Ereignisse stehen verstärkt in einem repetitiven und geordneten Bezug zueinander. Die Notenmitverfolgungssysteme anderer Arbeiten setzen ihren Schwerpunkt häufig auf die erkannte Tonhöhe (Orio et al., 2003). In dieser Arbeit sollen im Hinblick auf die Besonderheiten der Akkordnotation die pulsgebenden Strukturen stärker berücksichtigt werden, um die Unsicherheit durch die verringerten Informationen über die Tonhöhen auszugleichen.

Cont (2009) nutzt eine Temposchätzung der aktuellen Performance, um sein probabilistisches Modell bei hoher Unsicherheit in einem korrekten Tempo fortzuschreiben. Seine Arbeit bildet die Vorlage für den hier behandelten Ansatz.

Ein a priori definiertes Hidden Markov Model kann die erwartete Dauer der musikalischen Elemente zwar durch die Wahl der konkreten Modellierung erfassen, jedoch ist es nicht in der Lage, die Variabilität einer menschlichen Performance abzubilden. Die inhärente Temposchwankung eines jeden Spielers erzeugt unterschiedliche Notenlängen in verschiedenen Interpretationen der selben Noten. Die Implikationen für ein HMM zeigen sich vor allem in Passagen mit hoher Unsicherheit, da hier die Beobachtungen alleine nicht genügend Informationen für eine akkurate Fortschreibung liefern. Diese Unsicherheit kann jedoch über eine Temposchätzung ausgeglichen werden: Die Update-Zeitpunkte des Modells werden der aktuellen Geschwindigkeit angepasst, wodurch der Effekt des variablen Tempos vom Hidden Markov Model abstrahiert wird. So schreibt das HMM seinen Zustand trotz Unsicherheit richtig fort — nicht durch Beobachtungen, sondern durch zeitlich korrekte Zustandsübergänge.

Die Temposchätzung basiert auf der wissenschaftlichen Arbeit der *Automatic Rhythm Description* (für einen Überblick siehe Gouyon and Dixon (2005)). Hier wird, meist ohne Wissen über die gespielten Noten, versucht, verschiedene Aspekte der metrischen Struktur eines Musikstücks zu analysieren. Die Geschwindigkeitsschätzung geht dabei einher mit der Bestimmung der Position erkannter musikalischer Ereignisse in den Noten und bildet die Grundlage weiterer Analysen. Im Falle der Notenmitverfolgung sind die gespielten Noten bekannt und damit auch grundlegende Informationen über die metrische Struktur des Stücks. Infolgedessen fußt die Temposchätzung im Kontext der Notenmitverfolgung auf einer modifizierten Informationsgrundlage und kann das bereits vorhandene Kontextwissen über die Noten ausnutzen. Für die Geschwindigkeitsschätzung soll in dieser Arbeit ein *Kalman-Filter* genutzt werden. Im Folgenden werden die beiden benötigten Techniken, Hidden Markov Models und Kalman-Filter, eingeführt.

4 Rahmenbedingungen der Notenmitverfolgung

Die wichtigsten Aspekte für die Erkennung bilden die zu verfolgenden Noten und die menschliche Performance, die als Audiosignal vorliegt. Der folgende Abschnitt definiert, welche Eingrenzungen und Freiheiten für diese Faktoren gelten.

4.1 Die gegebenen Noten

Das Notenmitverfolgungssystem soll in dieser Arbeit auf der Akkordnotation basieren (Abb. 1). Notiert ist der Text des Musikstücks und, oberhalb des Textes, die jeweils zu spielenden Akkorde. Zusätzlich gegeben ist das Originaltempo, die Taktart und ein Beispielrhythmus. Aufschluss über die zeitliche Struktur des Stücks gibt vor allem die Angabe der Taktart, aus der ein Raster der Quantisierung erstellt werden kann (siehe Abb. 7). Informationen über die genaue Melodie des Stücks sind dabei im Gegensatz zur Notenschrift nicht vorhanden.

Es wird angenommen, dass der Gitarrist sich an die zeitliche Abfolge der Akkorde hält und sie beliebig anschlägt, solange die Anschläge zu dem zuvor abgeleiteten Raster passen. Diese Formulierung lässt dem Spieler die Wahl des Rhythmus offen, er muss lediglich zur angegebenen Taktart passen. Die vorgegebene Spielweise ist das Streichen, da die Erkennung immer alle Noten eines Akkords erwartet, das Zupfen einzelner Noten ist nicht vorgesehen.

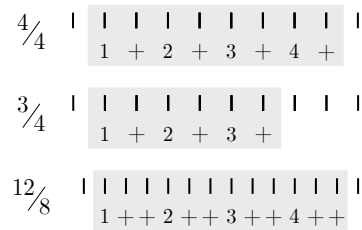


Abbildung 7: Beispiel für verschiedene Taktarten. Während ein $\frac{4}{4}$ - und ein $\frac{3}{4}$ -Takt eine binäre Unterteilung implizieren, führt ein $\frac{12}{8}$ -Takt zu einer ternären Untergliederung.

4.2 Aspekte menschlicher Performance

Die Interpretation eines Musikstücks durch einen Menschen entspricht nie einem genauen Abbild der in den Noten dargestellten Abfolge. Fehler, Betonungen oder Tempoveränderungen werden in den Noten abstrahiert, spielen in der tatsächlichen Performance aber eine wichtige Rolle. Da in dieser Arbeit eine Aufnahme einer menschlichen Interpretation mit Noten in Abgleich gebracht werden soll, ist es wichtig, diese Aspekte zu berücksichtigen.

4.2.1 Fehler

Musiker machen bei der Interpretation von Musikstücken unbeabsichtigte Fehler. Eine Note kann verpasst oder doppelt gespielt werden, eine Note kann mit einer falschen Tonhöhe angespielt werden und Spieler können Sprünge in den Noten vollziehen.

Diese Fehlerquoten zu quantifizieren würde hilfreiche Informationen im Hinblick auf die Erkennung liefern, jedoch ist dies für diese Arbeit aus Mangel an Daten nicht möglich.

4.2.2 Betonungen und Ausdruck

Beabsichtigte Abweichungen von einer direkten Umsetzung der Noten werden genutzt, um ausgewählte Stellen in der Musik hervorzuheben. Eine Übersicht verschiedener Aspekte menschlichen Ausdrucks in der Musik liefert zum Beispiel Juslin (2003). Während Wissen über systematische Abweichungen von den gegebenen Noten helfen kann, ein Notenmitverfolgungssystem zu verbessern, sind diese Aspekte für den hier unternommenen Versuch zu fortgeschritten und werden daher nicht gesondert berücksichtigt.

4.2.3 Tempo und Timing

Musikalisches Tempo entspricht dem Fortschritt in den Noten pro Zeiteinheit. Dieser Fortschritt ist jedoch durch die Variabilität einer menschlichen Performance nicht direkt messbar. Grundlage der Geschwindigkeitsmessung bilden die musikalischen Einzelereignisse, gemessen als Anschläge. Die Zeit zwischen zwei aufeinanderfolgenden Ansätzen wird als *Inter-Onset-Interval*, kurz *IOI*, bezeichnet. Bleiben aufeinanderfolgende IOIs bei gleicher Notenlänge gleich groß, ist das gespielte Tempo konstant. Gouyon and Dixon (2005) kategorisieren Gründe für die Abweichung von dieser Geschwindigkeit in die Kategorien *Tempo* und *Timing*. Abb. 8 zeigt verschiedene Möglichkeiten, wie sich solche Abweichungen ausprägen können. In (A) weicht ein einzelner Anschlag vom Grundpuls ab. In (B) wird ein Anschlag zu spät gespielt, dessen Abweichung sich aber auf alle folgenden Anschläge überträgt. Gemein haben beide Formen, dass die Rate, zu der neue Anschläge folgen, vor und nach der Abweichung gleich ist. Alle Abweichungen dieser Art werden als *Timing* zusammengefasst. (C) zeigt hingegen eine Änderung am Tempo, da sich die Rate der Anschläge auf globaler Ebene verändert. Zu beachten ist, dass zum Zeitpunkt der Beobachtung der ersten Abweichung nicht unterschieden werden kann, ob Fall (A), (B) oder (C) vorliegt.

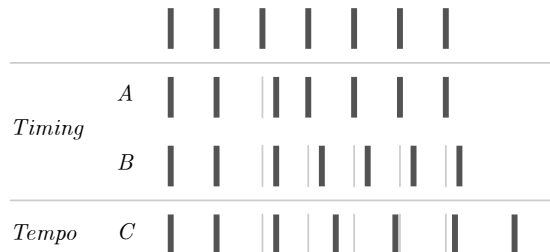


Abbildung 8: Tempo und Timing (Gouyon and Dixon, 2005).

5 Aufbau des Notenmitverfolgungssystems

Basierend auf den Erkenntnissen über menschliche Interpretationen und den gegebenen Noten wird das Problem der Notenmitverfolgung für die Gitarrenbegleitung definiert. Es wird angenommen, dass der Gitarrist die vorher festgelegten Noten auf der Gitarre interpretiert, wobei der Spieler sich an die vorgegebenen Akkorde und die Taktart hält, die genaue Spielweise ist ihm jedoch freigestellt. Bei einer idealen Interpretation wird erwartet, dass alle Ansatzzeitpunkte in das vom Tatum-Level implizierte Raster fallen und immer die in den Noten notierten Akkorde gespielt werden. Jedoch erzeugt der Mensch kein ideales Abbild der in den Noten festgelegten Performance, sondern eine durch die in 4.2 diskutierten Ursachen modifizierte Wiedergabe.

Es werden die einzelnen Ansatzzeitpunkte der Anschläge $t_0 \dots t_n$ und die dazugehörigen Tonhöhenereignisse $p_0 \dots p_n$ gemessen. Diese Messungen sind in der Regel mehrdeutig und liefern deshalb keinen direkten Aufschluss über die genaue Position im Stück. Zusätzlich stammen sie von einer unvollkommenen Erkennung und sind dadurch mit Unsicherheit behaftet. Der Startpunkt des Spielers in den Noten s_0 ist bekannt und es wird angenommen, dass die initiale Geschwindigkeit v_0 der in den Noten notierten Originalgeschwindigkeit des Stücks ähnelt.

Ziel des Modells ist es, die Messwerte mit den Noten in Überdeckung zu bringen. Damit das Modell für die Notenmitverfolgung geeignet ist, soll es die Messdaten sequentiell und in Echtzeit verarbeiten können, während die Noten a priori bekannt sind.

In dieser Arbeit soll die Notenmitverfolgung mithilfe eines Hidden Markov Models realisiert werden. Dabei wird angenommen, dass Beobachtungen $y_1 \dots y_T$ (die erkannten Akkorde) einem stochastischen Prozess $\{Y_t\}$ entspringen, der von einer Folge an Zuständen $x_0 \dots x_T$ (der aktuellen Position in den Noten) der Zufallsvariable $\{X_t\}$ generiert wird. Das Problem der Notenmitverfolgung ist dann die Umkehrung dieser Hypothese: das Finden der plausibelsten Folge an Zuständen (Notenpositionen), gegeben der Beobachtungen (Akkorde) (Cont, 2009).

Perkussive und rhythmische Musik wie die Gitarrenbegleitung eignet sich vergleichsweise gut für eine Geschwindigkeitsverfolgung, also die Messung des Fortschritts in den Noten pro Zeiteinheit (Alonso et al., 2004). Mit einer Geschwindigkeitsschätzung hat man die Möglichkeit, die Diskrepanz zwischen den zeitlich diskreten Hidden Markov Models und dem dynamischen und kontinuierlichen Tempo einer Performance auszugleichen, indem die Temposchätzung die Update-Zeitpunkte des Modells entsprechend der gespielten Geschwindigkeit wählt. Damit muss das Hidden Markov Model idealerweise nur die Fehler des Spielers erkennen und ausgleichen, nicht aber die Variabilität durch das dynamische Tempo berücksichtigen. Solange der Spieler keine Fehler begeht und die Temposchätzung ideal funktioniert, weicht das Modell selbst bei kompletter Unsicherheit der Akkorderkennung nicht von der wahren Position ab, sondern schreibt die Positionsschätzung entsprechend der Temposchätzung fort.

In den nächsten Schritten werden damit zunächst Erkennen für die benötigten Audiofeatures eingeführt, eine Geschwindigkeitsschätzung mittels eines Kalman-Filters entwickelt und anschließend die Modellierung des generativen Modells formuliert.

5.1 Feature-Erkennung

Für die Notenmitverfolgung soll eine Schätzung des gespielten Tempos verwendet werden. In einem Musikstück gibt die Taktart eine zeitliche Struktur vor, deren Regelmäßigkeit ausgenutzt werden kann, um das Tempo zu schätzen. Dafür wird versucht, die einzelnen Ansätze in der Audiospur zu erkennen. Neben der Ansatzerkennung wird eine Akkorderkennung für das generative Modell benötigt, die anhand der erkannten Tonhöhen im Audiosignal eine Wahrscheinlichkeit für die verschiedenen Zustände liefert.

5.1.1 Ansatzerkennung

Die Ansatzerkennung (onset detection) ist eine gut erforschte Aufgabe, bei der es darum geht, mögliche Ansätze im Audiosignal zu erkennen. Die Analyse besteht meist aus zwei Schritten: Zunächst wird eine Ansatzerkennungsfunktion berechnet, die hohe Werte bei potentiellen Ansätzen annehmen soll, aus der im nächsten Schritt mithilfe von Peak-Erkennung die geschätzten Ansätze ausgewählt werden.

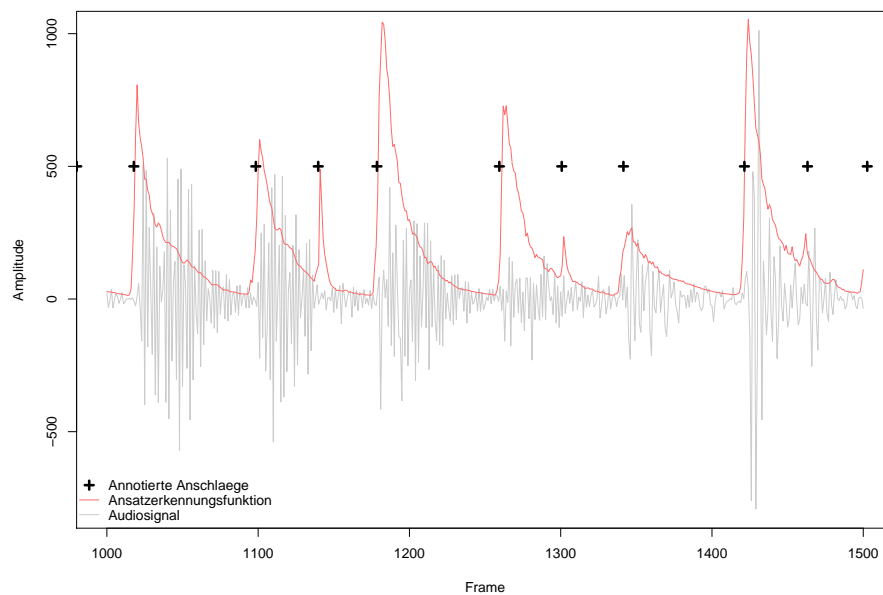


Abbildung 9: Die Amplitude eines Audiosignals mit annotierten Anschlägen des Gitarristen und einer Ansatzerkennungsfunktion (HFC).

Eine Herausforderung ist die Auswahl passender Audiofeatures für die Aufgabe der Ansatzerkennung. Bello et al. (2005) diskutieren mögliche Herangehensweisen. Ein einfacher Ansatz ist es, den zeitlichen Verlauf der Amplitude zu beobachten, da der Beginn einer Note oftmals mit einem Anstieg der Amplitude einhergeht. In Abb. 9 ist ein Ausschnitt der Amplitude eines Audiosignals zu sehen, dazu annotiert die gespielten Anschläge des Gitarristen und die Werte einer Ansatzerkennungsfunktion basierend auf spektralen Informationen, *HFC*

(*High Frequency Content*). Während die Ansatzerkennungsfunktion Maxima an den Punkten der Anschläge aufweist, zeigt die Amplitude des Audiosignals teilweise keine eindeutigen Ausschläge. Bello et al. (2005) nennt überlappende Töne und Amplitudenmodulationen als potentielle Probleme bei einer Erkennung anhand der Amplitude und folgert eine unzureichende Robustheit der Methode für komplexere Audiosignale.

Die Analyse mittels HFC basiert auf der Heuristik, dass Anstiege der Energie aufgrund von Einschwingvorgängen im Spektralbereich als breitbandige Phänomene auftreten, während sich die Energie sonst meist auf die niedrigen Frequenzen konzentriert. Gewichtet man die hohen Frequenzen stärker, erhält man für perkussive Ansätze gute Ergebnisse für die Ansatzerkennung, da der Einschwingvorgang, darunter auch von Gitarrensaiten, viel Rauschen enthält. Durch das Design der Methode ergeben sich Schwierigkeiten bei der Erkennung von niedrigen Tonhöhen, da diese durch die Wertung benachteiligt werden, sowie bei der Erkennung von nicht-perkussiven Ansätzen. Ebenso können starke Energien im hochfrequenten Bereich Ansätze maskieren, beispielsweise wenn ein Becken im Musikstück eingesetzt wird (Bello et al., 2005). In dieser Arbeit liegen lediglich perkussive Ansätze vor und das Audiosignal ist nicht durch andere Instrumente überlagert, die potentielle Ansätze maskieren könnten.

Um diese Schwächen auszugleichen, betrachten andere Techniken die Phaseninformation des Signals. Der Unterschied zwischen zwei aufeinanderfolgenden Phasen einer Frequenzlinie (frequency bin) eines Fensters kann genutzt werden, um die Frequenz zu berechnen, die *Momentanfrequenz* (*instantaneous frequency*). Der Abstand zwischen zwei aufeinanderfolgenden Momentanfrequenzen kann als Indikator der Stabilität des Signals dienen. Hohe Änderungen signalisieren potentielle Ansätze (Duxbury et al., 2003). Rauschen in niedrigerenergetischen Frequenzen oder Verzerrungen durch Nachbearbeitung von Musiksignalen vermindern die Genauigkeit der Erkennung jedoch (Bello et al., 2005).

Um die Erkennung bestmöglich zu gestalten, werden Kombinationen aus beiden Ansätzen gebildet (Bello et al., 2004). Abb. 10 zeigt die HFC-Methode (Masri and Bateman, 1996) im Vergleich mit einer *Complex-Domain Spectral Difference Function* (Bello et al., 2004), die bei der Erkennung auch die Phaseninformation berücksichtigt. Für die Analyse wurde die Softwarebibliothek Essentia genutzt (Bogdanov et al., 2013).

Die Erkenner verhalten sich dabei nicht grundsätzlich unterschiedlich, die Nachteile der HFC-Methode im Vergleich zu einem kombinierten Ansatz fallen durch die stark perkussiven Ansätze nicht ins Gewicht. In dieser Arbeit fällt die Wahl auf eine Erkennung mittels der HFC-Methode, da das Signal für das verwendete Set-Up ein geringeres Rauschen gezeigt hat.

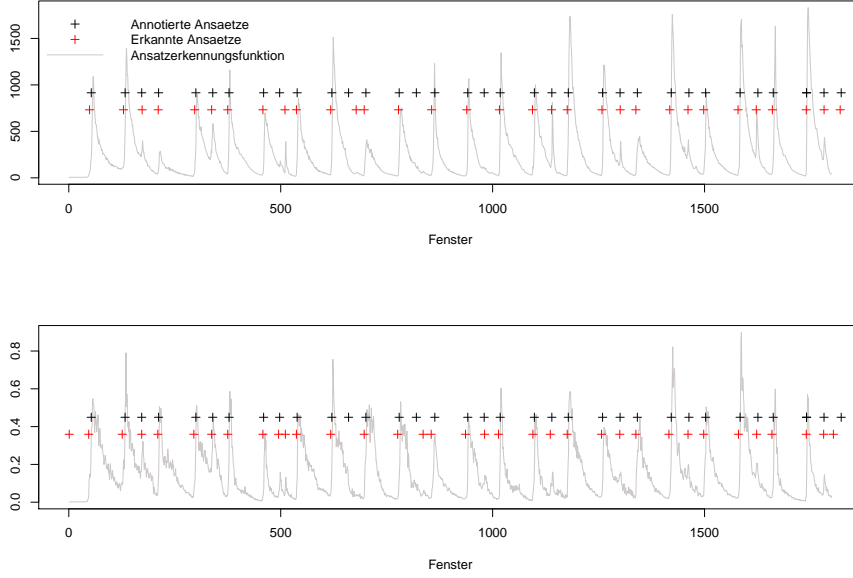


Abbildung 10: Vergleich der Ansatzerkennung anhand von HFC (oben) und einer Complex-Domain Spectral Difference Function (unten).

Peak-Erkennung Um aus der Erkennungsfunktion Ansätze erkennen zu können, müssen die Spitzen des Signals mittels eines Peak-Erkennungs-Algorithmus ausgewählt werden. In dieser Arbeit wird der Ansatz zur Peak-Erkennung von Bock et al. (2012) verwendet, die für eine Online-Erkennung geeignet ist. Dabei wird ein Fenster n als Ansatz akzeptiert, wenn die Ansatzerkennungsfunktion F folgende Bedingungen erfüllt:

1. $F(n) = \max(F(n - w_1 : n))$
2. $F(n) \geq \text{mean}(F(n - w_2 : n)) + \delta$
3. $n - n_{\text{letzterAnsatz}} > w_3$

Bock et al. (2012) evaluieren in ihrer Arbeit darüber hinaus die Parameterwahl: Dabei wird für $w_1 = 3$ empfohlen, für w_2 ein Wert zwischen 4 und 12 und für $w_3 = 3$.

Für das vorliegende System verwendet werden $w_1 = 3$, $w_2 = 10$ sowie ein dynamisch gewähltes w_3 . Um die mehrfache Meldung desselben Ansatzes zu vermeiden, und da der minimale Abstand zwischen zwei Anschlägen eines Rhythmus ein Tatum lang ist, wird die Ansatzerkennungsfunktion mit Hilfe von w_3 für ein halbes Tatum blockiert. Ein passender Schwellenwert δ wurde anhand von Tests ermittelt.

5.1.2 Akkorderkennung

Um Akkorde zu erkennen, wird die Intensität der verschiedenen Noten im Audiosignals gemessen. Dabei sind die erkannten *Tonklassen* (C, C#...B) für die Unterscheidung von Akkorden ausschlaggebend. In welcher Oktave die Noten liegen, ist jedoch nicht von Belang (Müller et al., 2011). Deshalb werden *Harmonic Pitch Class Profiles (HPCP)* als Audiofeatures genutzt, die alle erkannten Noten in eine einzelne Tonleiter abbilden (Fujishima, 1999). Ein Beispiel einer Messung findet sich in Abb. 11. Für die Erkennung wird die Softwarebibliothek Essentia genutzt, die einen Algorithmus zur Erkennung von HPCP-Audiofeatures bereitstellt (Bogdanov et al., 2013).

Die Erkennung von HPCP-Features erfordert relativ lange Analysefenster, Oudre et al. (2009) nutzen beispielsweise 753 ms lange Audioausschnitte. Für das vorliegende System muss ein Kompromiss aus verlässlicher Erkennung und Reaktivität gefunden werden. Für eine Akkordschätzung werden deshalb nach einer Wartezeit von ca. 46 ms für die Notenansatzzeit ca. 255 ms Audiosignal aufgezeichnet und ausgewertet. Für sehr schnelle Tempi ist dies bereits zu lange — bei 140 bpm und einem Viervierteltakt dauert ein Tatum ca 214 ms. Jedoch wird diese Einschränkung zugunsten der Erkennungsrate hingenommen.

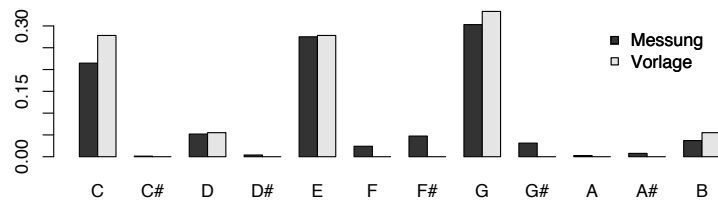


Abbildung 11: Selektierte Beispielmessung bei einem C-Dur Akkord, zusammen mit der Erkennungsvorlage. Während die gezeigte Messung sehr gut zur Erkennung verwendet werden kann, ist ein großer Teil der Messungen weniger eindeutig.

Mit dieser Information kann versucht werden, den gespielten Akkord zu erkennen. Oudre et al. (2009) definieren dafür Akkord-Vorlagen, die bei einer idealen Performance und Erkennung im Audiosignal zu erwarten wären, und versuchen anhand der Ähnlichkeit mit dem erkannten HPCP den tatsächlichen Akkord zu bestimmen. Grundlage der Vorlagen sind die Einzelnoten, aus denen sich ein Akkord zusammensetzt. Dabei ist zu beachten, dass neben der Grundfrequenz f der Note auch Vielfache der Frequenz $2f, 3f, 4f, \dots$, sogenannte *Obertöne*, in dem Signal vorkommen. Dieser Effekt wirkt sich auch auf das erkannte HPCP aus, da nicht alle Obertöne in dieselbe Notenklasse fallen. Gómez (2006) sowie Ryynänen and Klapuri (2008) wählen daher einen Abklingfaktor von 0,6, um den i -ten Oberton mit einem Gewicht von $0,6^i$ in der Erkennung zu berücksichtigen.

In Abb. 12 sieht man eine Erkennungsvorlage für einen C-Dur Akkord, wenn $i \in [1, 3]$ gewählt wird, um die Obertöne $2f, 3f$ und $4f$ mit in die Vorlage aufzu-

nehmen. Während Frequenzen $2f$ und $4f$ per Gesetzmäßigkeit der menschlichen Tonwahrnehmung wieder auf dieselbe Note fallen, taucht der zweite Oberton mit der Frequenz $3f$ sieben Halbtonschritte versetzt auf ($2^{(12+7)/12} \approx 3$) (Müller et al., 2011).

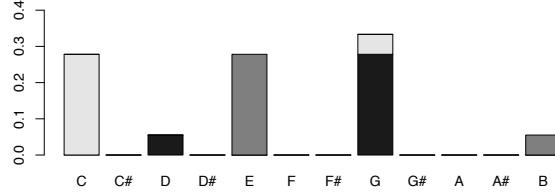


Abbildung 12: Erkennungsvorlage eines C-Dur Akkords.

Um die Distanz $d(h_t, q)$ zwischen dem erkannten HPCP h_t und einer Vorlage q zu bestimmen, können verschiedene Maße verwendet werden. Oudre et al. (2009) untersuchen die Eignung der *euklidischen Distanz*, der *Itakura-Saito-Divergenz* und der *Kullback-Leibler-Divergenz*, die auch als *Information Gain* bekannt ist. Die beiden zuletzt genannten Maße sind aufgrund fehlender Symmetrieeigenschaft keine Distanzmaße und können damit in zwei Varianten $d(h_t, q)$ und $d(q, h_t)$ berechnet werden.

Eine statistische Interpretation der Berechnung der Kullback-Leibler-Divergenz lautet wie folgt: Die KL-Divergenz $d_{kl}(p, q)$ liefert die durchschnittliche Likelihood, unendlich viele Daten mit der Verteilung p zu messen, wenn das Modell $Q = q(x)$ die Daten generiert (Shlens, 2014). Während in der Evaluation von Oudre et al. (2009) die Erkennung mittels der Form $d_{kl}(q, h_t)$ am besten funktioniert hat, wird in dieser Arbeit die Form $d_{kl}(h_t, q)$ benutzt, da die Interpretation gemäß der oberen Intuition besser zu dem angestrebten generativen Modell passt — Der Zustand x_i , dem der Beobachtungsprozess $B_i = Q = q(x)$ zugeordnet ist, generiert eine Beobachtung $o_t = h_t$ (das wahrgenommene HPCP).

Die Kullback-Leibler-Divergenz ist nur definiert, wenn $\forall x. q(x) = 0 \implies p(x) = 0$. Deshalb werden die Erkennungsvorlagen in der Implementierung so angepasst, das auch Noten, die nicht Teil des Akkords sind, ein positives Gewicht bekommen. Anderenfalls würde das Modell Q das Auftreten anderer Noten mit Wahrscheinlichkeit 0 erwarten. Durch Nebengeräusche und eine unsichere Erkennung ist jedoch mit dem Vorkommen anderer Notenklassen in der Messung zu rechnen.

Der Wertebereich der KL-Divergenz liegt zwischen 0 und ∞ . In dieser Arbeit wird jedoch ein Maß $d(h_t, q) \in [0, 1]$ benötigt, um als Wahrscheinlichkeit für das generative Modell interpretiert werden zu können. Nach Shlens (2014) besteht ein direkter Zusammenhang zwischen der *Average Likelihood* \bar{L} und der KL-Divergenz. So ist $\bar{L} = 2^{-d_{kl}(p, q)}$ und nimmt den Wert 1 an, falls die Verteilungen sich exakt entsprechen und nähert sich dem Wert 0, je unwahrscheinlicher Verteilungen aus P der Verteilung Q entspringen. Da somit \bar{L} den gewünschten Wertebereich besitzt, wird dieses Maß in der Erkennung verwendet.

5.2 Geschwindigkeitsschätzung mithilfe eines Kalman-Filters

Um die Geschwindigkeit eines Spielers in einem Stück mitzuverfolgen, wird ein lineares System modelliert, dass in dem Zustandsvektor $x_t = (v_t, a_t)^T$ die Geschwindigkeit und Beschleunigung des Spielers zum Zeitpunkt t enthält. Die Entwicklung des Systems kann wie folgt fortgeschrieben werden:

Bewegungsmodell $p(x_t|x_{t-1})$

$$x_t = \begin{pmatrix} v_t \\ a_t \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_{t-1} \\ a_{t-1} \end{pmatrix} + \epsilon \quad (4)$$

Der Fehler ϵ muss in den Prozess aufgenommen werden, um den in 4.2 beschriebenen Aspekten der menschlichen Performance gerecht zu werden. Ein lineares System ohne einen solchen Fehler vernachlässigt die tatsächliche Unsicherheit der Entwicklung und könnte deshalb den wahren Tempoverlauf über die Zeit nicht abbilden (Cemgil and Kappen, 2003). Der Fehler ϵ wird als normalverteilt um Null angenommen mit $\epsilon \sim N(0, \sigma_\epsilon^2)$.

Da die Werte aus dem Bewegungsmodell nicht verlässlich sind, können Messungen z_t verwendet werden um die Schätzung zu korrigieren. In diesem Fall werden die Messwerte der Ansatzerkennungsfunktion genutzt, um Beobachtungen der Geschwindigkeit und Beschleunigung zu erhalten. Gegeben einer Reihe von Ansatzzeitpunkten $t_1 \dots t_n$ kann folgendes Beobachtungsmodell für Messwerte von $v_1 \dots v_t$ und $a_1 \dots a_t$ formuliert werden:

Beobachtungsmodell $p(z_t|x_t)$

$$\begin{aligned} v_o^{(t)} &= \frac{\Delta s}{\Delta t} &= \frac{(s_t - s_{t-1})}{(t_t - t_{t-1})} &\text{mit } \sigma_{v_o}^2 \\ a_o^{(t)} &= \frac{2\Delta s - v_{t-1} * \Delta t}{\Delta t^2} &= \frac{2(s_t - s_{t-2}) - v_{t-1} * (t_t - t_{t-2})}{(t_t - t_{t-2})^2} &\text{mit } \sigma_{a_o}^2 \end{aligned}$$

Auch diese Größen sind mit Unsicherheit behaftet, da einerseits die Ansatzerkennung unzuverlässig ist und andererseits die Berechnung von Geschwindigkeit und Beschleunigung anhand der IOIs einen Fehler einführt. Letzteres lässt sich mit Referenz auf 4.2.3 erklären: Ein IOI muss anhand zweier Faktoren erklärt werden, dem Tempo und dem Timing. Beide werden jedoch auf die eindimensionale Achse der Zeit projiziert. Da das System unterbestimmt ist, gibt es keine eindeutige Lösung, das Finden einer plausiblen Lösung entspricht dem Problem der *Beat Induction*, deren Ziel es ist, die zum Ansatz gehörige Stelle im Score zu quantisieren. Die Berechnungen in diesem Fall erklären naiv 100% des IOI durch das globale Tempo. Dies führt zu einem gezackten Geschwindigkeitsverlauf (Robertson, 2012).

Das formulierte System ist linear hinsichtlich des inhärenten Zustandsübergangs als auch des Zusammenhangs zwischen dem Zustand und den Beobachtungen, und sowohl die Unsicherheit des Bewegungs-Modells als auch des Beobachtungs-Modells sind normalverteilt und unabhängig. Für solche Systeme stellt der Kalman-Filter einen optimalen rekursiven Schätzer des Zustands dar. Mit den Gleichungen ist es möglich, den Zustand des Systems rekursiv fortzuschreiben.

5.2.1 Quantisierung der Notenstellen

Das bisherige Modell geht davon aus, dass der Fortschritt im Score nach jedem gemessenen Ansatz bekannt ist. Dies gilt jedoch nur für eine ungenaue, aber fehlerfreie Erkennung und eine ungenaue, aber fehlerfreie Umsetzung eines bekannten Rhythmus durch den Spieler. Der genaue Rhythmus ist jedoch unbekannt und Fehler sind zu erwarten. Die Zuordnung ist somit nicht eindeutig. Sobald Anschläge falschen Notenstellen zugeordnet werden, sind die Berechnungen des Filters falsch. Deshalb ist die richtige Quantisierung der Ansätze in den Noten eine entscheidende Herausforderung.

Das Raster für die Quantisierung wird in dieser Arbeit auf Grundlage der angegebenen Taktart aufgebaut und entspricht dem Tatum-Level (siehe Abb. 7). Um die folgende Diskussion verständlich zu gestalten, wird angenommen, dass ein Tatum eine Sekunde lang ist, sodass die möglichen Quantisierungspunkte immer Ganzzahlen sind. Die Aufgabe der Quantisierung ist es, Messungen von Ansatzzeitpunkten diesem Raster zuzuordnen, um die Notenlänge der Ereignisse zu bestimmen.

Cemgil (2004) diskutiert den Einfluss der Korrelation aufeinanderfolgender Ansätze auf die Zuordnung. Bei Annahme von Unabhängigkeit wird die Zuweisung einfach mittels der Wahl des nächsten Quantisierungszeitpunkts getroffen. Die Abweichung eines Ansatzes hat keinen Einfluss auf die erwartete Position anderer Ansätze: Wird ein Ansatz zum Zeitpunkt 2.9 gemessen, so wird er dem Punkt 3 zugeordnet, unabhängig von allen anderen Messungen.

Eine andere Möglichkeit ist es, den Abstand zwischen zwei Ansätzen zu betrachten, den IOI, diesen zu runden und dementsprechend weit in den Noten voranzuschreiten. Nur wenn der IOI größer als 0.5 ist, werden zwei Ansätze unterschiedlichen Punkten zugeordnet. Die Idee ist, dass nach einer besonders großen Abweichung in die eine Richtung, z. B. 0,47 statt 0, nicht direkt eine große Abweichung in die andere Richtung erwartet wird, z. B. 0,52 statt 1. Tatsächlich trifft hier eine Erkennung nach IOI eine plausible Zuordnung von $[0,0]$, da der Abstand zwischen 0,47 und 0,52 kleiner als 0,5 ist. Eine unabhängige Erkennung würde jedoch durch die Wahl des nächsten Punkts die Punkte $[0,1]$ zuordnen, was eine Streckung des gemessenen IOI um den Faktor 20 bedeutet und für die menschliche Wahrnehmung deshalb unplausibel ist (Cemgil, 2004).

Beide Varianten sind in manchen Fällen problematisch. Abb. 13 diskutiert zwei Grenzfälle. Diese Arbeit verwendet aufgrund des schlechten Fehlerverhaltens der IOI-Variante die direkte Quantisierung, ohne die Abhängigkeit aufeinanderfolgender Ansätze zu berücksichtigen.

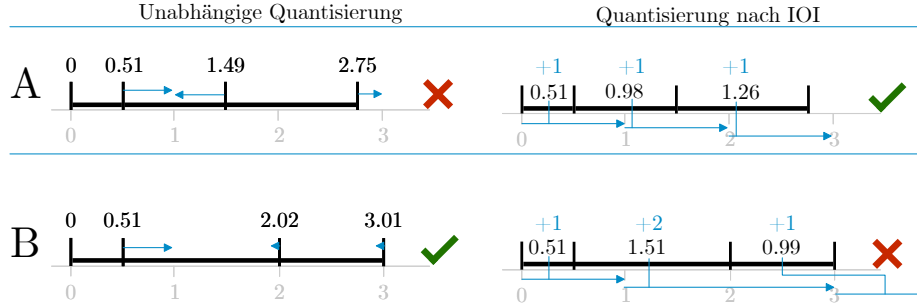


Abbildung 13: Fälle A und B stellen zwei Messreihen von Ansätzen dar. Eine musikalisch plausible Zuordnung ist jeweils $[0,1,2,3]$. In A würde eine unabhängige Erkennung die falsche Zuordnung $[0,1,1,3]$ treffen, eine Zuordnung nach IOI $[0,1,2,3]$. Dafür wäre in Fall B die Zuordnung respektive $[0,1,2,3]$ und $[0,1,3,4]$. Die Fehler der IOI-Erkennung können sich über die Zeit kumulieren: Der Erkenner schätzt, am Ende von B bei Tatum #4 zu sein, tatsächlich ist er aber ziemlich exakt bei #3.

5.2.2 Modellparameter des Kalman-Filters

Der Zustandsvektor des Modells $\vec{x}_t = (v_t, a_t)^T$ besteht aus einer Schätzung der Geschwindigkeit v_t und einer Beschleunigung a_t zum Zeitpunkt t . Um diese Werte in einer Anwendung im Rahmen eines Kalman-Filters erfolgreich fortzuschreiben zu können, werden Werte für die Prozess- und Messunsicherheit ermittelt.

Die Prozessunsicherheit R Die Prozessunsicherheit sollte die gesamte Unsicherheit umfassen, die beim Fortschreiben des Zustandes nicht vom linearen Modell erfasst wird. Dies betrifft vor allem das Abbremsen oder Beschleunigen des Spielers, da das Modell von einer konstanten Beschleunigung ausgeht.

Es gibt dabei unklare Einflüsse auf die erwartete Temposchwankung eines Künstlers für ein gegebenes Stück. Ohne weitere Analyse möglicher Faktoren ist eine Quantifizierung der Unsicherheit im Rahmen dieser Arbeit nicht möglich, hierfür wäre eine Studie nötig. Die Parameter werden deshalb frei gewählt und als möglicher Optimierungsparameter für Trainingsdurchläufe des Modells vermerkt.

Die Messunsicherheit Q Die Messunsicherheit drückt aus, mit welcher Unsicherheit die Messwerte behaftet sind, die der Filter im Korrekturschritt verwendet. Um die Messunsicherheit Q zu bestimmen, benötigt man Informationen über die Varianz in den Messungen σ_v^2 und σ_a^2 sowie über den Korrelationskoeffizienten $\rho_{v,a}$ der beiden Werte. Dazu werden freie Interpretationen von Musikstücken manuell mit Geschwindigkeit und Beschleunigung annotiert und mit den Messwerten verglichen.

Eine wahre Geschwindigkeits- und Beschleunigungskurve gibt es durch die Mehrdeutigkeit von Tempo und Timing nicht. Deshalb wird die Annahme getroffen, dass die vom Spieler absolvierte Tempokurve gleichmäßig verläuft: Der Musiker wechselt das Tempo nicht sprunghaft und die Beschleunigung bleibt für

ausreichend kleine Bereiche konstant. Dann kann eine passende Modellierung gefunden werden, indem abschnittsweise eine initiale Geschwindigkeit festgelegt und für diesen Bereich eine konstante Beschleunigung angenommen wird. Diese beiden Werte können mithilfe einer linearen Regression für die Messwerte der Geschwindigkeit bestimmt werden. Um die gewählte Modellierung anhand der Testdaten zu überprüfen, bietet es sich an, zunächst eine mittlere Geschwindigkeit des Spielers im Stück zu bestimmen und anschließend den Offset der gemessenen Ansätze zur mechanischen Interpretation im Referenztempo über die Zeit zu plotten.

Dieses Vorgehen wird in Abb. 14 verdeutlicht. Die lineare Regression liefert eine Startgeschwindigkeit von $1,4573139 \frac{beats}{sec}$ und eine Beschleunigung von $0,0012231 \frac{beats}{sec^2}$. Die Annahme eines linearen Tempoverlaufs liefert eine gute Annäherung an die tatsächlichen Ansätze.

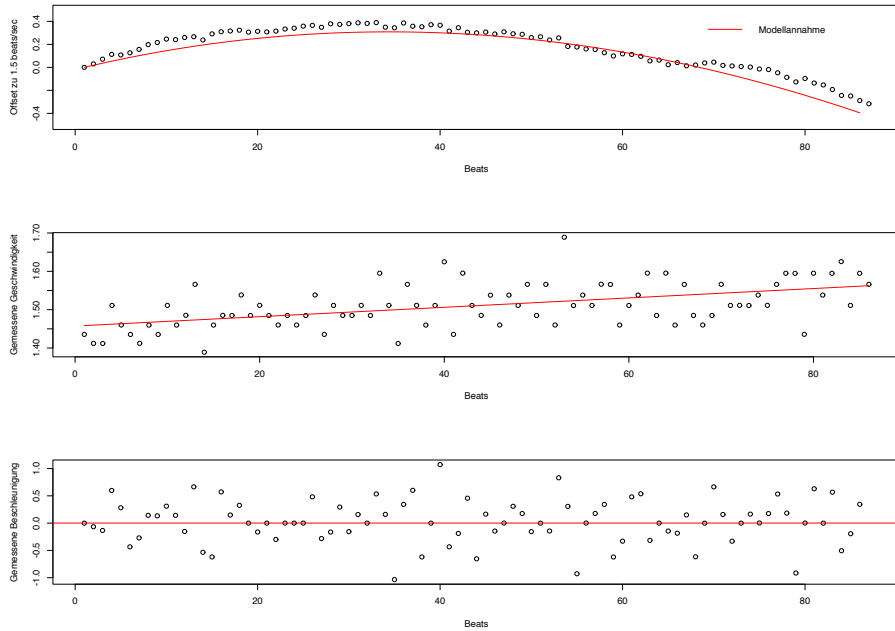


Abbildung 14: Oben: Der Offset in Sekunden der gemessenen Ansätze von einem konstanten Referenztempo zusammen mit dem Offset gegeben der Modellannahme. Sind hier deutliche Abweichungen der tatsächlichen Werte zu dem linearen Modell zu sehen, deutet das auf eine schlechte Anpassung des Modells hin. Nur wenn die Beschleunigung innerhalb eines Abschnitts ungefähr konstant ist, kann eine gute Anpassung erreicht werden. Mitte: Die gemessene Geschwindigkeit zusammen mit dem Ergebnis einer linearen Regression. Unten: Die Beschleunigungsmesswerte mit der Beschleunigung aus der linearen Regression.

Anhand der Schätzungen der linearen Modelle $M_v = \beta_0 + \beta_1 t$ und $M_a = \beta_1$ sowie den Messreihen $v_i = \{v_t\}_{t=1}^T$ und $a_i = \{a_t\}_{t=1}^T$ können Werte für die Varianz der Geschwindigkeits- und Beschleunigungsmesswerte geschätzt werden:

$$\hat{\sigma}_v^2 = \frac{1}{T-1} \sum_{t=1}^T [v_t - (\beta_0 + \beta_1 t)]^2, \quad \hat{\sigma}_a^2 = \frac{1}{T-1} \sum_{t=1}^T [a_t - \beta_1]^2$$

Eine Schätzung des Korrelationskoeffizienten $\hat{\rho}_{v,a}$ lässt sich über die Messreihen v_i und a_i ebenso bestimmen:

$$\hat{\rho}_{v,a} = \frac{\frac{1}{T-1} \sum_{t=1}^T (v_t - \bar{v})(a_t - \bar{a})}{\sqrt{\frac{1}{T-1} \sum_{t=1}^T (v_t - \bar{v})^2} \cdot \sqrt{\frac{1}{T-1} \sum_{t=1}^T (a_t - \bar{a})^2}} = \frac{s_{va}}{s_v s_a}$$

Nun sind alle Werte geschätzt, die für die Messunsicherheit benötigt werden, und können eingesetzt werden:

$$Q = \begin{pmatrix} \sigma_v^2 & \rho_{v,a} \\ \rho_{v,a} & \sigma_a^2 \end{pmatrix} = \begin{pmatrix} \hat{\sigma}_v^2 & \hat{\rho}_{v,a} \\ \hat{\rho}_{v,a} & \hat{\sigma}_a^2 \end{pmatrix}$$

5.3 Die Mitverfolgung mit einem HMM

Die Notenmitverfolgung soll auf einem Hidden Markov Model basieren. Dabei werden die zu spielenden Noten als Zustände $S = \{S_1 \dots S_n\}$ modelliert. Es wird angenommen, dass die beobachtbaren Audiofeatures $y_1 \dots y_T$ von einer verborgenen Folge dieser Zustände $x_1 \dots x_T \in S$ generiert werden. Die Aufgabe der Notenmitverfolgung ist es anschließend, anhand der Beobachtungen Rückschlüsse über die Abfolge der verdeckten Zustände zu ziehen, um die Position in den Noten zu schätzen (Cont, 2009).

Um die diskrete Form des Modells mit der dynamischen und kontinuierlichen Performance eines Menschen in Einklang zu bringen, werden die Zeitpunkte, zu denen Zustandsübergänge stattfinden, anhand der Temposchätzung diskretisiert. Jeder Zustand im Modell repräsentiert ein Tatum in den Noten, sodass potentiell jeder Anschlag des Gitarristen eine Beobachtung für das Modell liefern kann. Die Beobachtungen stammen aus der Akkorderkennung, die eine Likelihood des gemessenen HPCP gegeben dem Zustand berechnet, die direkt als Emissionswahrscheinlichkeit interpretiert wird. Der Aufbau des Modells ist in Abb. 15 grafisch dargestellt. Die initiale Wahrscheinlichkeitsverteilung nimmt die Startposition des Gitarristen in den Noten mit einer Wahrscheinlichkeit von 1 an — Der Spieler startet laut Annahme immer genau an der zuvor spezifizierten Stelle in den Noten. Ähnliche Ansätze der Modellierung finden sich bei Nakamura et al. (2014).

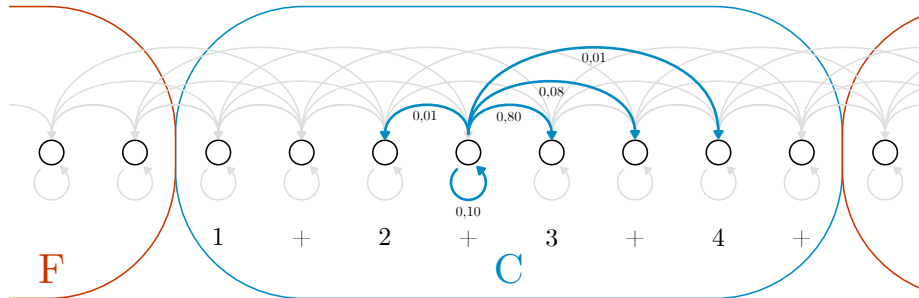


Abbildung 15: Aufbau des verwendeten Hidden Markov Modells. Der zugrundeliegende Viervierteltakt impliziert 8 Zustände pro Takt. Die Übergangswahrscheinlichkeiten wurden frei gewählt.

Im Gegensatz zu Nakamura et al. (2014) wird nicht versucht, beliebige Sprünge in den Noten durch Verbindungen zu allen anderen Zuständen abzubilden. In dem stark mehrdeutigen Fall der Akkordnotation, bei dem viele Passagen des Stücks ähnliche Abläufe zeigen, würde ein solcher Versuch die Mitverfolgung stark erschweren.

Mit einer funktionierenden Temposchätzung ist das Modell in der Lage die Position des Spielers im Stück zu verfolgen. Die Wahl der Zustandsübergangsverteilung sorgt zunächst für eine wachsende Unsicherheit in der Positionsschätzung über die Zeit. Innerhalb eines Akkords sind für eine gegebene Beobachtung alle Zustände gleich wahrscheinlich, sodass die Schätzung der Position durch die Zustandsübergänge für die Dauer des Akkords unsicherer wird, sich jedoch entsprechend dem geschätzten Tempo im Modell nach vorne bewegt. Am Übergang

zwischen zwei Akkorden findet dann eine Korrektur statt: Klingt bereits der neue Akkord, obwohl ein Großteil der Wahrscheinlichkeitsmasse noch in den Zuständen des alten Akkords liegt, wird durch die Gewichtung mit der Emissionswahrscheinlichkeit der neue Akkord um ein Vielfaches wahrscheinlicher, der alte Akkord unwahrscheinlicher. Sollte im anderen Fall die Schätzung in den neuen Akkord übergehen, bevor dieser zu hören ist, bleiben alte Zustände wahrscheinlich, während die Zustände des folgenden Akkords durch die Beobachtung abgestraft werden.

Durch die beschriebene Dynamik können leichte Temposchwankungen, die nicht vom Tempomodell erfasst wurden, sowie verpasste oder doppelt gespielte Noten ausgeglichen werden. Größere Sprünge im Stück oder Pausen durch den Spieler sind im Modell nicht abgebildet. Als Schätzung der Position in den Noten wird nach jedem Update-Schritt durch den Vorwärts-Algorithmus der wahrscheinlichste Zustand ausgegeben.

6 Evaluation

Nach einer kurzen Beschreibung der technischen Umsetzung folgt die Bewertung des vorliegenden Notenmitverfolgungssystems. Dabei wird zunächst der Geschwindigkeitsschätzer gesondert untersucht, wonach das Gesamtsystem mithilfe eines standardisierten Frameworks evaluiert wird.

6.1 Technische Umsetzung des Modells

Die im Rahmen dieser Arbeit umgesetzte Software ist in der Programmiersprache Python geschrieben. Das System basiert auf zwei getrennten Agenten, einem Teil, der das eingehende Audiosignal analysiert und relevante Features berechnet, und einem anderen Teil, der das generative Modell implementiert. Der Grund dieser Trennung ist die zeitliche Unabhängigkeit des Hidden Markov Modells, dessen Update-Zeitpunkte variabel anhand der gemessenen Geschwindigkeit gewählt werden. Technisch gesehen läuft jeder Agent als eigenständiger Prozess, wobei der geteilte Speicher durch *Lock-Objekte* abgesichert ist. Dargestellt wird das System in Abb. 16.

Die Feature-Erkennung (Schritte 1-7 in der Abbildung) besteht aus einer Reihe von Verarbeitungsschritten, die relevante Features aus dem Audiosignal extrahieren, die dann für die Geschwindigkeitsschätzung und die Akkorderkennung genutzt werden können. Zunächst wird das Audiosignal in sich überlappende Fenster gruppiert (Fenstergröße = 1024, Schrittweite = 512) (2), die die Eingabe für die Ansatzerkennungsfunktion bilden (3). Erkennt der Peak-Erkennungs-Algorithmus eine Spitze in der erzeugten Funktion, wird der Zeitpunkt der Messung an den Kalman-Filter weitergeleitet, der daraufhin die Schätzung der Geschwindigkeit aktualisiert (4-5). Ebenso beginnt damit die Erkennung des gespielten Akkords. Dabei wird zunächst die Notenansatzzeit abgewartet und anschließend elf Fenster der Größe 2048 aufgezeichnet und als Eingabe für die HPCP-Erkennung genutzt. Die Akkorderkennung bestimmt aus dieser Messung dann die Wahrscheinlichkeiten für die möglichen Akkorde (6-7).

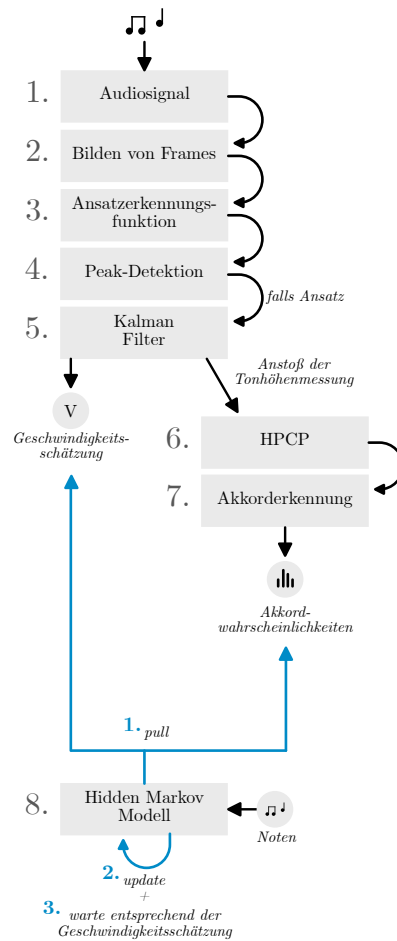


Abbildung 16: Realisierung des Notenmitverfolgungssystems.

Das Hidden Markov Model (8) benötigt die Geschwindigkeitsmessung und die Wahrscheinlichkeiten der Akkorde: Ersteres wird verwendet, um die Update-Zeitpunkte entsprechend dem aktuellen Tempo zu wählen, letzteres dient als Beobachtung zum Fortschreiben des Modells. Dabei holt sich das Modell die Daten bei Bedarf von dem anderen Agenten und wartet anschließend für die restliche Dauer des Tatoms.

Die Softwarebibliotheken für die Audioverarbeitung stellen *Essentia*⁶ (Bogdanov et al., 2013) sowie *SciPy*⁷ und *Numpy*⁸ dar. Zur Audioeingabe per Mikrofon wird *PyAudio*⁹ verwendet. Zur Analyse von Audiodateien werden diese in Echtzeit abgespielt, die Audioausgabe des Computers zur Audioeingabe geschleift, und wie bei der Eingabe über Mikrofon analysiert. Auf diese Weise kann das System einheitlich angesprochen, und die zeitliche Synchronisation der Agenten innerhalb einer konsistenten Implementierung gelöst werden. Für das Abspielen von Audiodateien werden die Softwarebibliotheken *SoundFile*¹⁰ und *Sounddevice*¹¹ genutzt. Das verwendete Audioformat entspricht folgendem Format: `.wav` Header, 32 bit float Codec und Mono-Audio.

Die Peak-Detektion, die vorlagenbasierte Akkorderkennung sowie das Hidden Markov Model und der Kalman-Filter sind entsprechend den Ergebnissen der Arbeit ohne weitere Bibliotheken implementiert worden (Schritte 4,5,7 und 8). Die Umsetzung der zugrundeliegenden Feature-Erkennung durch *Essentia* (Schritte 3 und 6) ist zur Vollständigkeit im Anhang B zu finden.

⁶<https://essentia.upf.edu/documentation/> (zuletzt aufgerufen am 10.09.2019).

⁷<https://www.scipy.org> (zuletzt aufgerufen am 10.09.2019).

⁸<https://numpy.org> (zuletzt aufgerufen am 10.09.2019).

⁹<http://people.csail.mit.edu/hubert/pyaudio/> (zuletzt aufgerufen am 10.09.2019).

¹⁰<https://github.com/bastibe/PySoundFile> (zuletzt aufgerufen am 10.09.2019).

¹¹<https://python-sounddevice.readthedocs.io/en/0.3.13/> (zuletzt aufgerufen am 10.09.2019).

6.2 Evaluation der Geschwindigkeitsschätzung

Die Geschwindigkeitsschätzung soll idealerweise die inhärenten Temposchwankungen einer menschlichen Performance erkennen, und dem Hidden Markov Model auf diese Weise passende Update-Zeitpunkte vorgeben. Da die Schätzung somit einen essentiellen Teil des Modells ausmacht, soll sie hier nochmal gesondert evaluiert werden.

Die Messung der Geschwindigkeit für den Kalman-Filter entspricht der gesuchten Größe direkt, somit kann auch versucht werden, den Zustand mittels eines gleitenden Mittelwerts zu schätzen. Der Vergleich der beiden Ansätze findet sich in Abb. 17 und 18 wieder. Zu sehen ist, dass sich der Mittelwert und der Filter ähnlich verhalten. In Abb. 18 sieht man zudem, dass beide Ansätze einzig von den quantisierten Notenlängen abhängen, die die Grundlage der Geschwindigkeitsmessungen liefern, und so beide bei falscher Quantisierung das wahre Tempo verlieren. Das Problem liegt bei dem rasanten Tempoanstieg von Sekunde 20 bis 30. Während die ersten Messungen noch der Kurve folgen, werden ab einem gewissen Punkt alle Messungen falsch interpretiert. Dieser Abwärtssprung der Messungen ist darauf zurückzuführen, dass die Quantisierung von der aktuellen Geschwindigkeit abhängt, und diese sich nicht schnell genug anpasst. Das führt zu der falschen Quantisierung der Notenlängen (bei einem Viervierteltakt statt zwei Achtelnoten drei Achtelnoten — ein Faktor von $\frac{2}{3}$, der Werte um 1,7 statt 2,55 erklärt). Dies stellt eine fundamentale Herausforderung für den Umgang mit unsicheren Notengrundlagen dar: Nur wenn die erkannten Ansätze richtig interpretiert werden, kann die Geschwindigkeit akkurat geschätzt werden.

Shiu et al. (2008) verwenden deshalb *Probabilistic Data Association*, um die Schätzung des Kalman-Filters zu unterstützen. Dabei werden alle möglichen Interpretationen der Zuordnung der Ansätze mit einer Wahrscheinlichkeit versehen, und anhand dieser Gewichtung die beste Option ausgewählt.

Abbildung 17: Kalman-Filter und gleitender Mittelwert bei synthetisch erzeugten Testdaten. Die Größe der Kreise deutet die Unsicherheit der a-posteriori-Schätzung des Filters an.

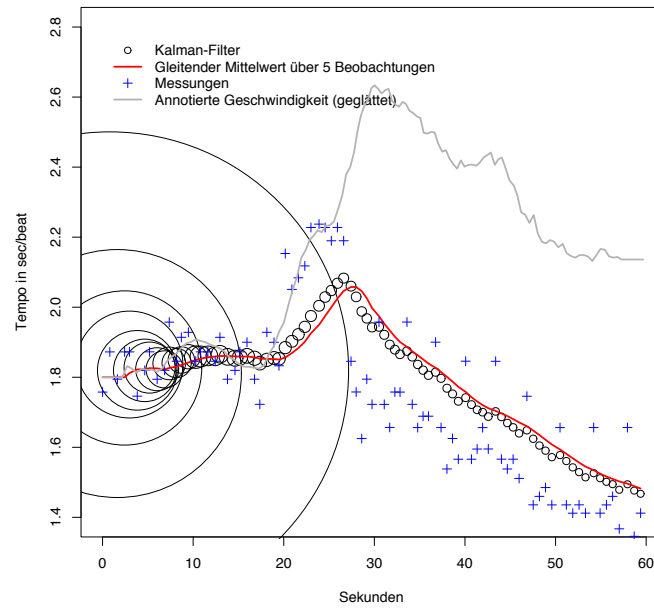
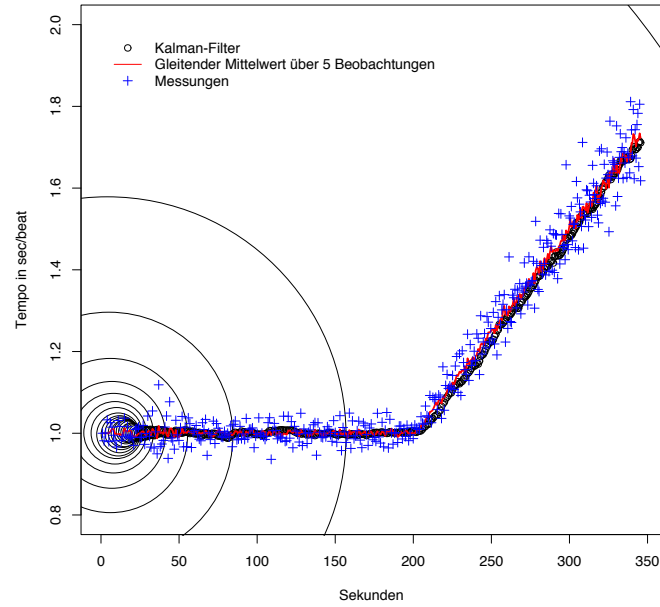


Abbildung 18: Beispiel eines Durchlaufs, bei dem die Quantisierung der Notenlängen fehlschlägt.

6.3 Evaluation der Notenmitverfolgung

Das entwickelte Notenmitverfolgungssystem wird nach dem Standard der MIREX-Challenge evaluiert (Cont et al., 2007), wobei aufgrund der abweichenden Zielsetzung ein anderer Datensatz verwendet wird.

6.3.1 Evaluierungsmethodiken für Notenmitverfolgungssysteme

Viele frühe Arbeiten führen lediglich qualitative Evaluationen ihrer Systeme anhand kleiner Datensätze durch. Cont et al. (2007) schlagen ein Evaluierungs-Framework für Notenmitverfolgungssysteme vor, das in der MIREX-Challenge eingesetzt wird, und von vielen Autoren für die Evaluation übernommen wurde. Abb. 19 verdeutlicht die erhobenen Daten, die zur Evaluation verwendet werden und sich wie folgt berechnen:

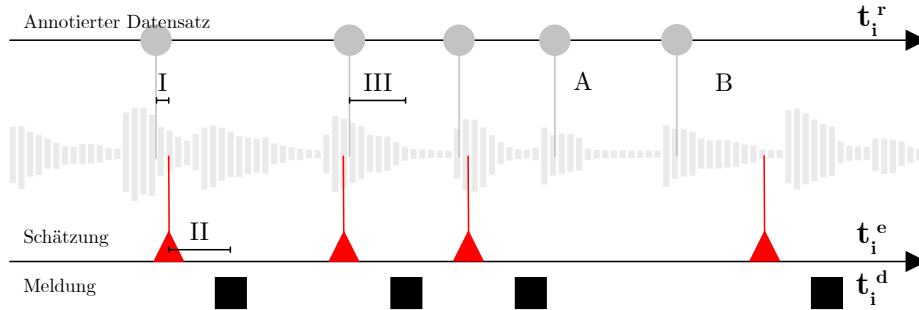


Abbildung 19: Größen der Evaluation nach Cont et al. (2007).

- I Der *Fehler* (error) $e_i = t_i^e - t_i^r$: Zeitlicher Unterschied zwischen dem geschätzten Event (estimate) t_i^e und der Referenz t_i^r .
- II Die *Latenz* $l_i = t_i^d - t_i^e$: der Unterschied zwischen dem Zeitpunkt zu dem die Messung gemeldet wird (detection) t_i^d und dem Zeitpunkt des geschätzten Events.
- III Der *Offset* $o_i = t_i^d - t_i^r$: die Zeitspanne zwischen dem tatsächlichen Auftreten und dem Zeitpunkt der Meldung.

Daneben werden verpasste (A) und falsch angeordnete (B) Noten unterschieden, wobei für (B) ein Schwellenwert $\theta_e = 300ms$ gewählt wird. Da in der klassischen Problemstellung der Notenmitverfolgung maximal so viele Events gemeldet werden, wie auch in den Noten annotiert sind, ist die doppelte Meldung einer Note ausgeschlossen.

Der *durchschnittliche absolute Fehler* μ_e , die *durchschnittliche Latenz* μ_l und der *durchschnittliche absolute Offset* μ_o geben Aufschluss über die Qualität der Notenmitverfolgung. Aus den falsch angeordneten Noten wird eine *Misalign Rate* p_e als Anteil der falsch zugeordneten Noten gebildet.

6.3.2 Anpassung des Evaluierungs-Frameworks an die Akkordnotation

Die Modellgrundlage entspricht entgegen dem Großteil anderer Notenmitverfolgungssystemen der Akkordnotation statt der Notenschrift. Die angestrebte

Positionsschätzung entspricht dadurch wie bei anderen Notenmitverfolgungssystemen der absoluten Position in den Noten (z. B. Beat #67), jedoch können nicht alle Metriken direkt übernommen werden.

Durch die unterschiedliche Informationsgrundlage können verpasste Noten (A) in dieser Arbeit nicht erhoben werden, da dem Gitarristen die konkrete Interpretation des Musikstücks freigestellt ist. Für die Kontrolle der Zuordnung der Notenposition (B) wird der wahre Ansatzzeitpunkt und die zugehörige Notenposition aus dem Datensatz geholt, der zeitlich am naheliegendste Update-Zeitpunkt des Modells bestimmt und dessen Schätzung mit der tatsächlichen Notenposition verglichen. Das Modell liefert jedes Tatum ein Update, womit die naheliegendste Messung höchstens 0,5 Tatum entfernt ist. Dieser Wert darf bis zu 300 ms groß sein, um zeitlich nahe genug zu liegen, um als richtige Zuordnung gewertet zu werden. Solange das Tempo des analysierten Musikstücks also schneller als 36 bpm ist ($=0,6 \frac{\text{beats}}{\text{sec}} = 300 \text{ ms}$ für ein Tatum bei binärer Unterteilung), ist jede richtige Positionsschätzung des Modells auch im Sinne von Cont et al. (2007) richtig zugeordnet.

Leider enthält der von Cont et al. (2007) verwendete Datensatz ausschließlich klassische Musik von nicht-perkussiven Instrumenten, sodass die Daten nicht für das entworfene System zu verwenden sind. Stattdessen wird der *Guitarset-Datensatz* von Xi et al. (2018) für eine beispielhafte Evaluation verwendet.

6.3.3 Testaufbau

Der Guitarset-Datensatz enthält Aufnahmen von sechs Künstlern zu fünf verschiedenen Genres, mit sechs verschiedenen Aufnahmen zu jedem Genre. Die Evaluation in dieser Arbeit nutzt beispielhaft die sechs Stücke des Künstlers #1 aus dem Genre *Singer-Songwriter*. In Abb. 20 sieht man, wie die Daten an das System angebunden und für die Evaluation vorbereitet werden.

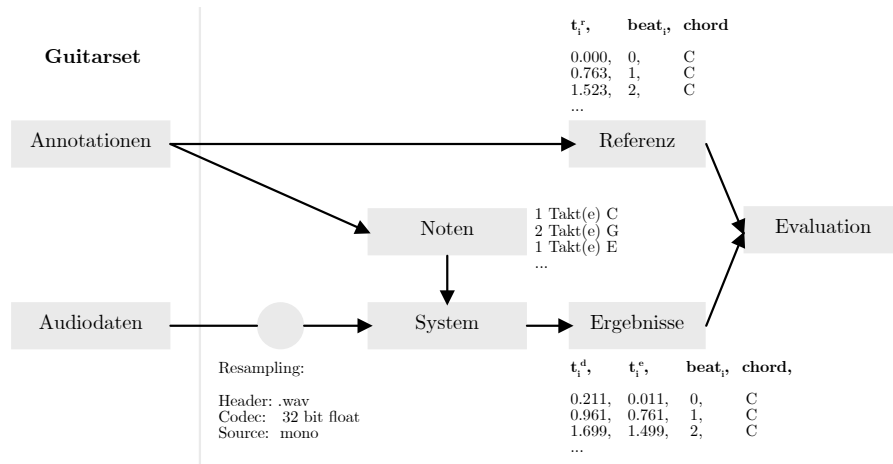


Abbildung 20: Aufbau der Evaluation mit den Audiodaten aus dem Guitarset-Datensatz.

6.3.4 Ergebnisse

Ergebnisse für Stücke aus der Kategorie Singer-Songwriter des Datensatzes sind in Tabelle 6.3.4 dargestellt. Neben den Kennzahlen aus der Tabelle sind ausführliche grafische Ergebnisse der Durchläufe im Anhang A zu finden. Sie sollen das Verhalten des Modells in den verschiedenen Abläufen der Evaluation grafisch verdeutlichen.

| Stück | Events | p_e | p_e^* | μ_e | μ_o | μ_l |
|---------------------|--------|--------------|---------|---------|---------|---------|
| 01_SS1-68-E_comp | 48 | 0,687 | 0,937 | 0,165 | 0,342 | 0,2 |
| 01_SS1-100-C#_comp | 48 | 0,104 | 0,208 | 2,893 | 2,888 | 0,2 |
| 01_SS2-88-F_comp | 64 | 0,313 | 0,563 | 0,820 | 0,909 | 0,2 |
| 01_SS2-107-Ab_comp | 64 | 0,438 | 0,813 | 0,280 | 0,336 | 0,2 |
| 01_SS3-84-Bb_comp | 64 | 0,219 | 0,578 | 0,770 | 0,797 | 0,2 |
| 01_SS3-98-C_comp | 64 | 0,063 | 0,484 | 0,474 | 0,596 | 0,2 |
| Durchschnitt/Gesamt | 352 | 0,296 | 0,599 | 0,843 | 0,920 | 0,2 |

Tabelle 1: Ergebnisse der Evaluation anhand eines Teils der Daten aus dem Guitarset-Datensatz (Xi et al., 2018). Der Wert p_e ist der Anteil der richtig geschätzten Notenpositionen, Wert p_e^* beinhaltet eine Toleranz von 1 Tatum — ist die eigentliche Position der Beat #3, so wird auch 2,5 und 3,5 als richtige Schätzung angesehen. Dies entspricht einem Fehler der Größenordnung 500 ms bei 60 bpm, Viervierteltakt, bis hin zu 214 ms bei 140 bpm. Der Wert soll einen zweiten Anhaltspunkt liefern, wie gut die Mitverfolgung bei höherer Toleranz funktioniert.

6.3.5 Diskussion

Der durchschnittliche Anteil an richtig zugeordneten Notenpositionen $\bar{p}_e = 0,296$ (bzw. $\bar{p}_e^* = 0,599$) entspricht der *Total Precision*, die bei Cont et al. (2007) zum Vergleich zwischen Notensystemen verwendet wird. Das Ergebnis ist durch den abweichenden Datensatz und die angepassten Metriken nicht direkt mit anderen Ergebnissen zu vergleichen, jedoch erzielen State-Of-The-Art-Systeme bei der grundsätzlich ähnlichen Evaluation auf verschiedenen Datensätzen Werte im Bereich von 90%, siehe Cont (2009), Duan and Pardo (2011) oder Carabias-Orti et al. (2015). Dies spricht für Verbesserungspotential des entwickelten Ansatzes.

Auch in der vorliegenden Evaluation kann man feststellen, dass die Temposchätzung noch teils deutliche Defizite aufweist, wie für das Stück in Abb. 25 im Anhang zu sehen ist. Die Grafik oben links zeigt, wie die Geschwindigkeitsschätzung durch die falsche Quantisierung einen Fehler von vier Sekunden gegen Ende des Stücks aufbaut. Anzumerken ist jedoch, dass das Modell trotz der vom Geschwindigkeitsmodell zu kurz gewählten Update-Zeitpunkte sich erneut an die wahre Position annähern kann, wenn ein Akkordübergang stattfindet (siehe Grafik in der Mitte, Beat 36). Eine genauere Analyse des Tempomodells folgt im Abschnitt 6.2.

Im Hidden Markov Model sind die Parameter für den Zustandsübergang nur anhand weniger Versuche gewählt worden. Hier kann mit gezieltem Training eine höhere Performance erreicht werden. Daneben weicht das Modell in

manchen Fällen stark von einem plausiblen Ablauf ab. Abb. 22 zeigt einen Fall, in dem sich die Zustandswahrscheinlichkeitsverteilung auf Zustände an den Übergängen zwischen Akkorden konzentriert und der Zustand mit der maximalen Wahrscheinlichkeit sich sehr sprunghaft ändert — ein solcher Verlauf durch die Zustände des Stücks ist musikalisch nicht plausibel. Eine passende probabilistische Formulierung des Modells sollte eine solche Entwicklung der Zustandswahrscheinlichkeitsverteilung von vorneherein ausschließen. Interessant ist hier ein Blick auf die Arbeit von Cuvillier (2016), der sich mit dem Thema zeitlich kohärenter Modellierung probabilistischer Modelle auseinandersetzt.

Ebenso problematisch ist die Laufzeit des Vorwärts-Algorithmus für längere Stücke, die im Bereich von $O(2TN^2)$ liegt, mit N gleich der Anzahl der Zustände und T der Anzahl an Beobachtungen (Rabiner, 1989). Pro Takt werden bei den gängigen Taktarten sechs bis zwölf Zustände benötigt. Mit wachsender Länge der Stücke führt die quadratische Komplexität zu erheblichem Rechenaufwand: Ein Stück der Länge 30 min bei 120 bpm und Viervierteltakt benötigt bereits 7200 Zustände für die Modellierung.

Störungen ergeben sich aus einer potentiell fehlerhaften Akkorderkennung. Werden systematisch Akkorde durch die Erkennung bevorzugt, so wird die Zustandswahrscheinlichkeitsverteilung an den Übergängen zwischen den Akkorden verharren, obwohl der Akkordwechsel stattgefunden hat. Die Akkorderkennung konnte in dieser Arbeit aus Zeitgründen nicht auf diesen Fehler hin untersucht werden. Die Arbeit nutzt zudem nur sehr wenig Kontextinformation. Sind genauere Daten zu der Fehlerwahrscheinlichkeit von Spielern oder typischen Tempoverläufen von Stücken bekannt, kann dieses Wissen in der Modellierung berücksichtigt werden, um die Erkennung zu verbessern.

Ein interessanter Aspekt wäre die Evaluation des Systems mit einem Datensatz mit Temposchwankungen und Fehlern in der Interpretation. Der verwendete Datensatz besitzt lediglich Stücke mit konstantem Tempo und fehlerfreien Interpretationen, sodass die Modellierung bisher nicht auf die Robustheit hinsichtlich dieser Faktoren getestet werden konnte.

7 Fazit

In dieser Arbeit wurde ein Notenmitverfolgungssystem auf Basis der Akkordnotation umgesetzt und allgemeine Schwierigkeiten bei der Umsetzung robuster Systeme identifiziert. Die Besonderheit dabei ist der Umgang mit der unsicheren Notengrundlage, die dem Spieler Freiheiten in der rhythmischen Interpretation lässt. Dafür wurde die Erkennung in zwei Teile faktorisiert: eine Geschwindigkeitsschätzung basierend auf einem Kalman-Filter, die die inhärenten Temposchwankungen einer menschlichen Performance erkennen soll, und einem Hidden Markov Model, das auf Grundlage der Temposchätzung die Position in den Noten anhand der notierten Akkorde mitverfolgen soll. Das dargestellte System nutzt die im Audiosignal erkannten Notenansatzzeitpunkte und eine vorlagenbasierte Akkorderkennung. Das Modell wurde mithilfe eines angepassten Evaluierungs-Frameworks für Notenmitverfolgungssysteme von Cont et al. (2007) getestet. Das System erreicht übliche Maßstäbe von Notenmitverfolgungssystemen nicht.

Die gewählte Modellierung des HMMs soll menschliche Fehler und Abweichungen abbilden. Dies wird versucht, indem neben dem direkten Übergang zum nächsten Zustand auch Sprünge nach vorne oder hinten in den Noten möglich sind. Durch diese Modellierung verteilt sich die Zustandswahrscheinlichkeit mit der Zeit über das gesamte Stück. Ein großer Anteil der Wahrscheinlichkeitsmasse drückt dann eine Schätzung aus, die eigentlich sehr unwahrscheinlich ist. Hier besteht das Potential, durch eine passendere probabilistische Formulierung unrealistische Abläufe auszuschließen.

Eine offene Herausforderung des vorliegenden Ansatzes ist die Quantisierung der Notenlängen in der Geschwindigkeitsschätzung. Wenn ein Ansatz gemessen wird, muss diesem eine Position in den Noten zugeordnet werden. Je stärker die Variabilität in der menschlichen Performance, desto weniger eindeutig ist diese Zuordnung zu treffen. Verbesserungen im Vergleich zur bisher genutzten Strategie sind durch eine probabilistische Betrachtung der Zuordnung zu erwarten, die idealerweise das Wissen über die Aspekte menschlicher Performance berücksichtigt. Trotz der Schwierigkeiten kann die Analyse metrischer Strukturen eine entscheidende Ergänzung zur Tonhöhenerkennung darstellen, um robuste Systeme zu ermöglichen. Dies gilt insbesondere für Modelle basierend auf unsicheren Notengrundlagen.

Literatur

- Alonso, M., David, B., and Richard, G. (2004). Tempo And Beat Estimation Of Musical Signals.
- Bello, J., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., and Sandler, M. (2005). A Tutorial On Onset Detection In Music Signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047.
- Bello, J., Duxbury, C., Davies, M., and Sandler, M. (2004). On the Use of Phase and Energy for Musical Onset Detection in the Complex Domain. *IEEE Signal Processing Letters*, 11(6):553–556.
- Bilmes, J. A. (1993). *Timing Is of the Essence: Perceptual and Computational Techniques for Representing, Learning, and Reproducing Expressive Timing in Percussive Rhythm*. PhD thesis, Massachusetts Institute of Technology.
- Bock, S., Krebs, F., and Schedl, M. (2012). Evaluating the Online Capabilities of Onset Detection Methods.
- Bogdanov, D., Wack, N., Gomez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., and Serra, X. (2013). Essentia: An Audio Analysis Library for Music Information Retrieval.
- Carabias-Orti, J. J., Rodríguez-Serrano, F. J., Vera-Candeas, P., Ruiz-Reyes, N., and Cañadas-Quesada, F. J. (2015). An Audio to Score Alignment Framework Using Spectral Factorization and Dynamic Time Warping. In *ISMIR*, pages 742–748.
- Cemgil, A. T. (2004). *Bayesian Music Transcription*. PhD thesis, Radboud Universiteit Nijmegen.
- Cemgil, A. T. and Kappen, B. (2003). Monte Carlo Methods for Tempo Tracking and Rhythm Quantization. *Journal of Artificial Intelligence Research*, 18:45–81.
- Clendinning, J. P. and Marvin, E. W. (2016). *The Musician’s Guide to Theory and Analysis: Third Edition*. W. W. Norton & Company.
- Cont, A. (2009). A Coupled Duration-Focused Architecture for Real-Time Music-to-Score Alignment.
- Cont, A., Schwarz, D., Schnell, N., and Raphael, C. (2007). Evaluation of Real-Time Audio-to-Score Alignment.
- Cuvillier, P. (2016). On Temporal Coherency of Probabilistic Models for Audio-To-Score Alignment.
- Dannenberg, R. B. (1984). An On-Line Algorithm For Real-Time Accompaniment. In *ICMC*, volume 84, pages 193–198.
- Deutsch, D. (2013). *Psychology of Music*. Elsevier.

- Duan, Z. and Pardo, B. (2011). A State Space Model for Online Polyphonic Audio-Score Alignment. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 197–200, Prague, Czech Republic. IEEE.
- Duxbury, C., Bello, J. P., Davies, M., and Sandler, M. (2003). A Combined Phase and Amplitude Based Approach to Onset Detection for Audio Segmentation. In *Digital Media Processing for Multimedia Interactive Services*, pages 275–280, Queen Mary, University of London.
- Fujishima, T. (1999). Real-Time Chord Recognition of Musical Sound: A System Using Common Lisp Music. *Proc. ICMC, Oct. 1999*, pages 464–467.
- Gómez, E. (2006). Tonal Description of Polyphonic Audio for Music Content Processing. *INFORMS Journal on Computing*, 18(3):294–304.
- Gouyon, F. and Dixon, S. (2005). A Review of Automatic Rhythm Description Systems. *Computer Music Journal*, 29(1):34–54.
- Juslin, P. N. (2003). Five Facets of Musical Expression: A Psychologist’s Perspective on Music Performance. *Psychology of Music*, 31(3):273–302.
- Lerch, A. (2012). *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. Wiley, Hoboken, NJ.
- Masri, P. and Bateman, A. (1996). Improved Modelling of Attack Transients in Music Analysis-Resynthesis. In *ICMC*.
- Maybeck, P. S. (1979). *Stochastic Models, Estimation and Control*. Number v. 141 in Mathematics in Science and Engineering. Academic Press, New York.
- Müller, M., Ellis, D. P. W., Member, S., Klapuri, A., and Richard, G. (2011). Signal Processing for Music Analysis. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1088–1110.
- Nakamura, E., Nakamura, T., Saito, Y., Ono, N., and Sagayama, S. (2014). Outer-Product Hidden Markov Model and Polyphonic MIDI Score Following. *Journal of New Music Research*, 43(2):183–201.
- Orio, N., Lemouton, S., and Schwarz, D. (2003). Score Following: State of the Art and New Developments.
- Oudre, L., Grenier, Y., and Fevotte, C. (2009). Template-based Chord Recognition: Influence of the Chord Types.
- Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2).
- Robertson, A. (2012). Decoding Tempo and Timing Variations in Music Recordings from Beat Annotations.
- Ryynänen, M. P. and Klapuri, A. P. (2008). Automatic Transcription of Melody, Bass Line, and Chords in Polyphonic Music. *Computer Music Journal*, 32(3):72–86.

- Shiu, Y., Cho, N., Chang, P.-C., and Kuo, C.-C. (2008). Robust On-Line Beat Tracking With Kalman Filtering And Probabilistic Data Association (KF-PDA). *IEEE Transactions on Consumer Electronics*, 54(3):1369–1377.
- Shlens, J. (2014). Notes on Kullback-Leibler Divergence and Likelihood. *arXiv preprint arXiv:1404.2000*.
- Uhle, C. and Herre, J. (2003). Estimation Of Tempo, Micro Time And Time Signature From Percussive Music.
- Vercoe, B. (1984). The Synthetic Performer in the Context of Live Performance. In *Proceedings of International Computer Music Conference*, pages 199–200.
- Welch, G. (1997). An Introduction to the Kalman Filter.
- Xi, Q., Bittner, R. M., Pauwels, J., Ye, X., and Bello, J. P. (2018). GuitarSet: A Dataset for Guitar Transcription.

A Ausführliche Ergebnisse der Evaluation

Die folgende Sektion stellt die einzelnen Durchläufe der Evaluation detailliert anhand von verschiedenen Grafiken dar.

Zu sehen ist jeweils oben links der Verlauf der zeitlichen Differenz zwischen dem vom Tempomodell gemeldeten Zeitpunkt des n -ten Anschlags und dem Zeitpunkt des tatsächlichen Anschlags. Hohe Abweichungen deuten auf eine unzureichende Performance der Geschwindigkeitsschätzung hin.

In der Mitte sieht man die Schätzung des Notenmitverfolgungsmodells über die Zeit: Die ausgefüllten Punkte sind die annotierten Anschläge aus dem Datensatz, die leeren Kreise zeigen die Schätzung des Systems.

Oben rechts ist zudem der Fehler der Schätzung in Sekunden zu sehen. Er berechnet sich aus der Differenz der tatsächlichen Position in den Noten und der geschätzten Position. Durch das Wissen über die tatsächliche Geschwindigkeit lässt sich diese Differenz in Sekunden umrechnen, um den Fehler besser vergleichbar zu machen. Die eingezeichnete horizontale Linie entspricht dem von Cont et al. (2007) vorgeschlagenen Grenzwert von 300 ms, ab dem eine Zuordnung nicht mehr als richtig gewertet wird. Der Fehler wird nach der Zuordnung der Schätzung zu einem Datenpunkt erhoben und enthält somit einen Rundungsfehler von maximal 0,5 mal der Länge eines Tatums.

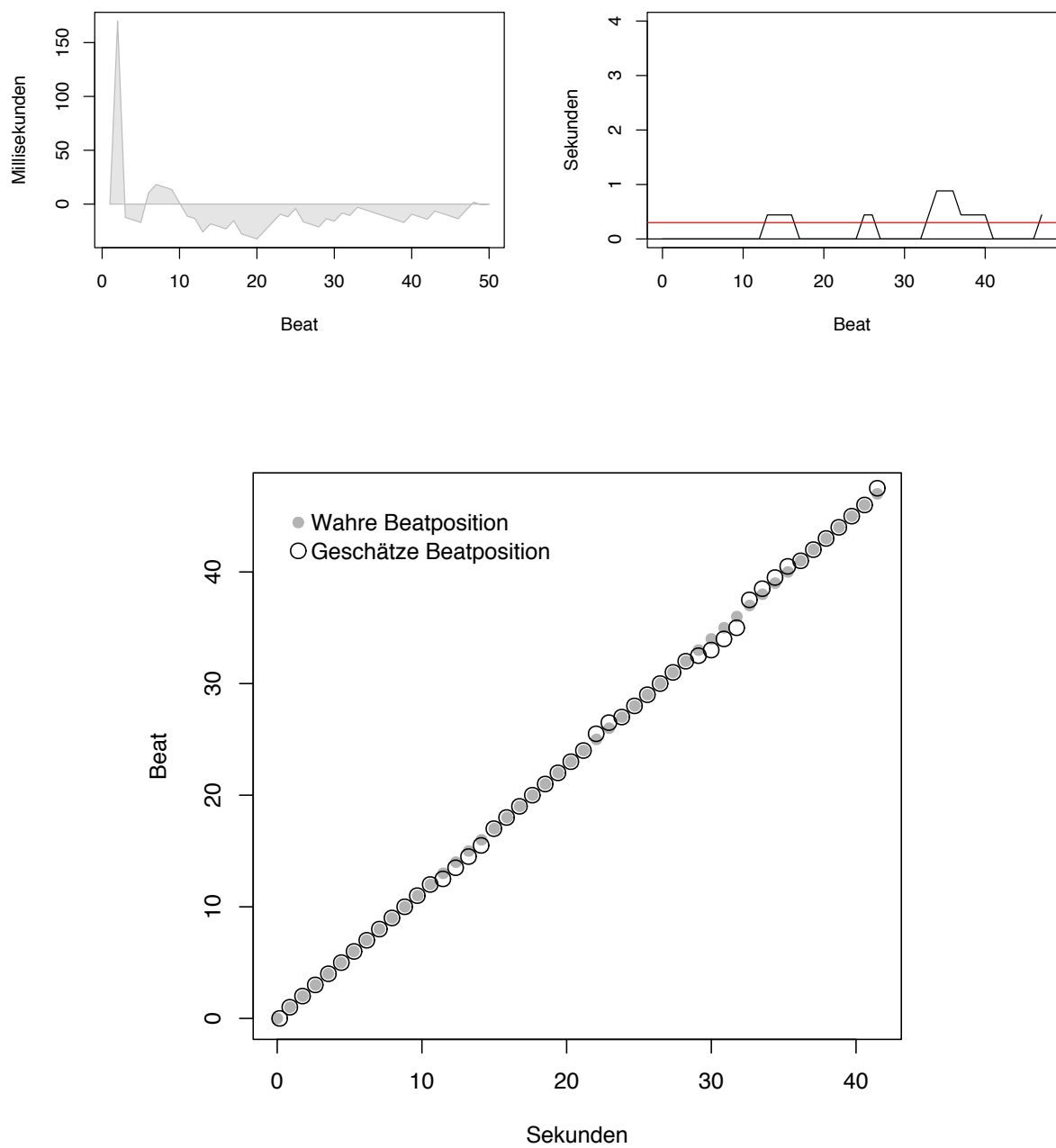


Abbildung 21: Durchlauf des Stücks 01_SS1-68-E_comp.

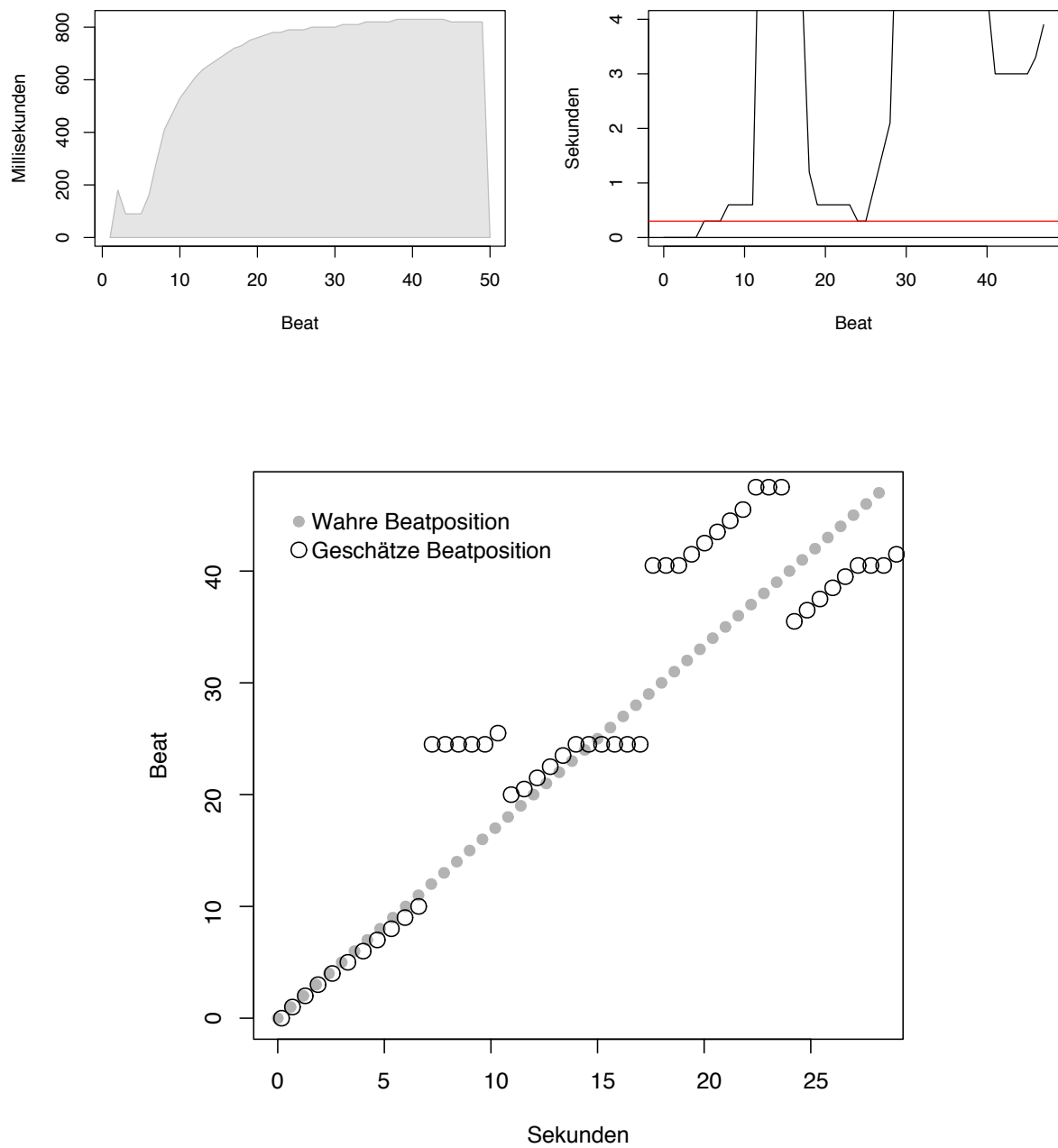


Abbildung 22: Durchlauf des Stücks 01_SS1-100-C#_comp.

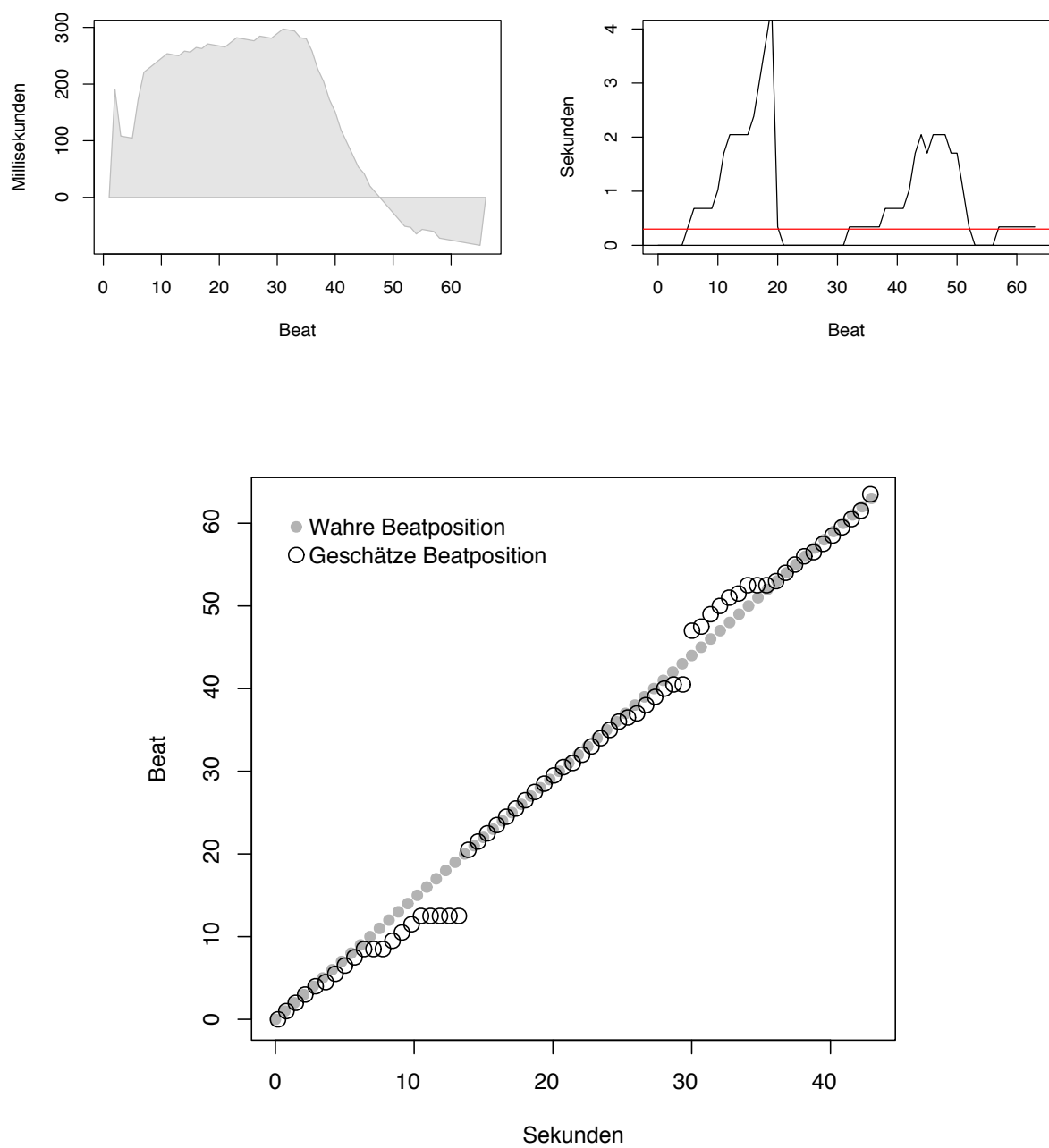


Abbildung 23: Durchlauf des Stücks 01_SS2-88-F_comp.

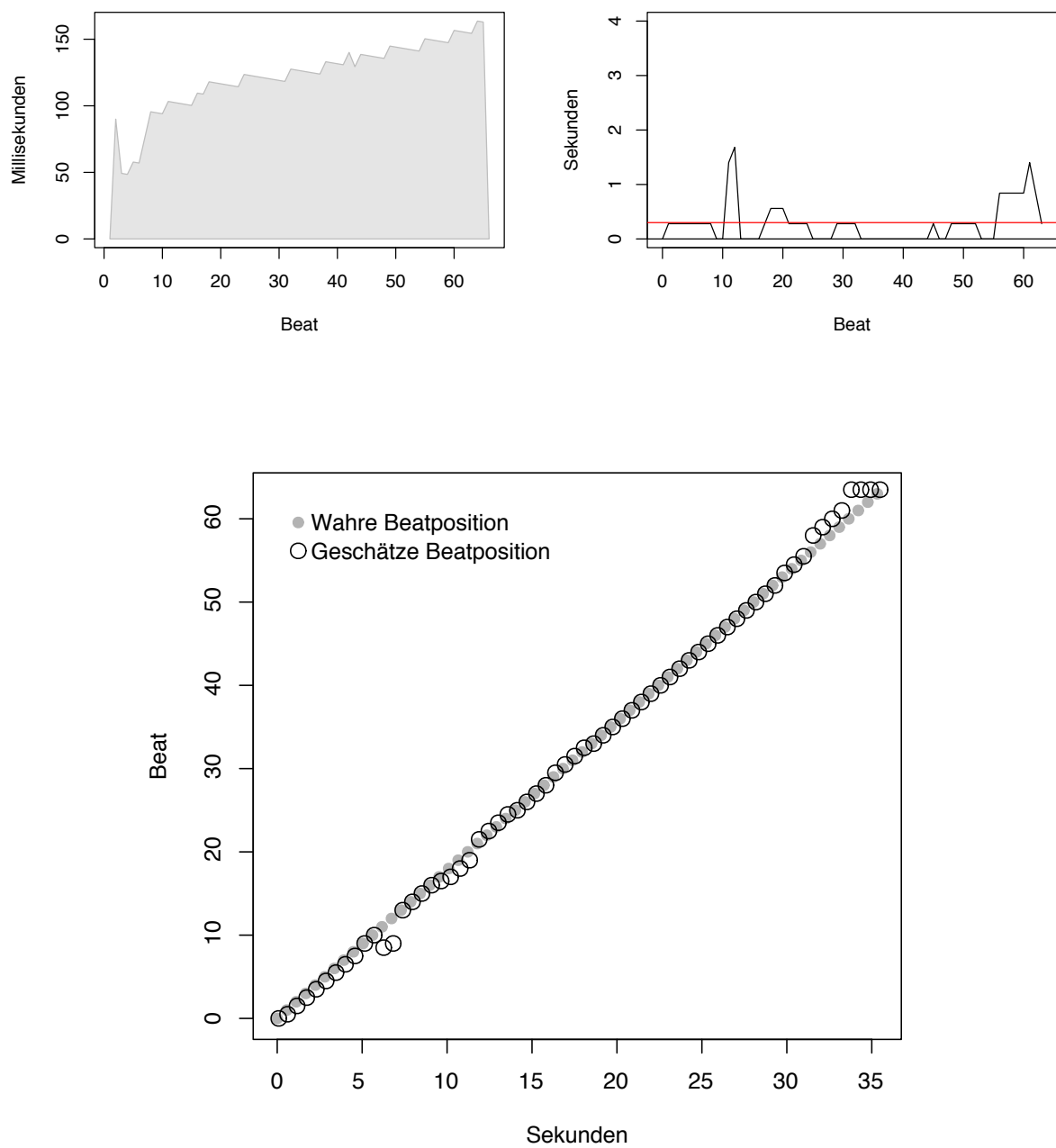


Abbildung 24: Durchlauf des Stücks 01_SS2-107-Ab_comp.

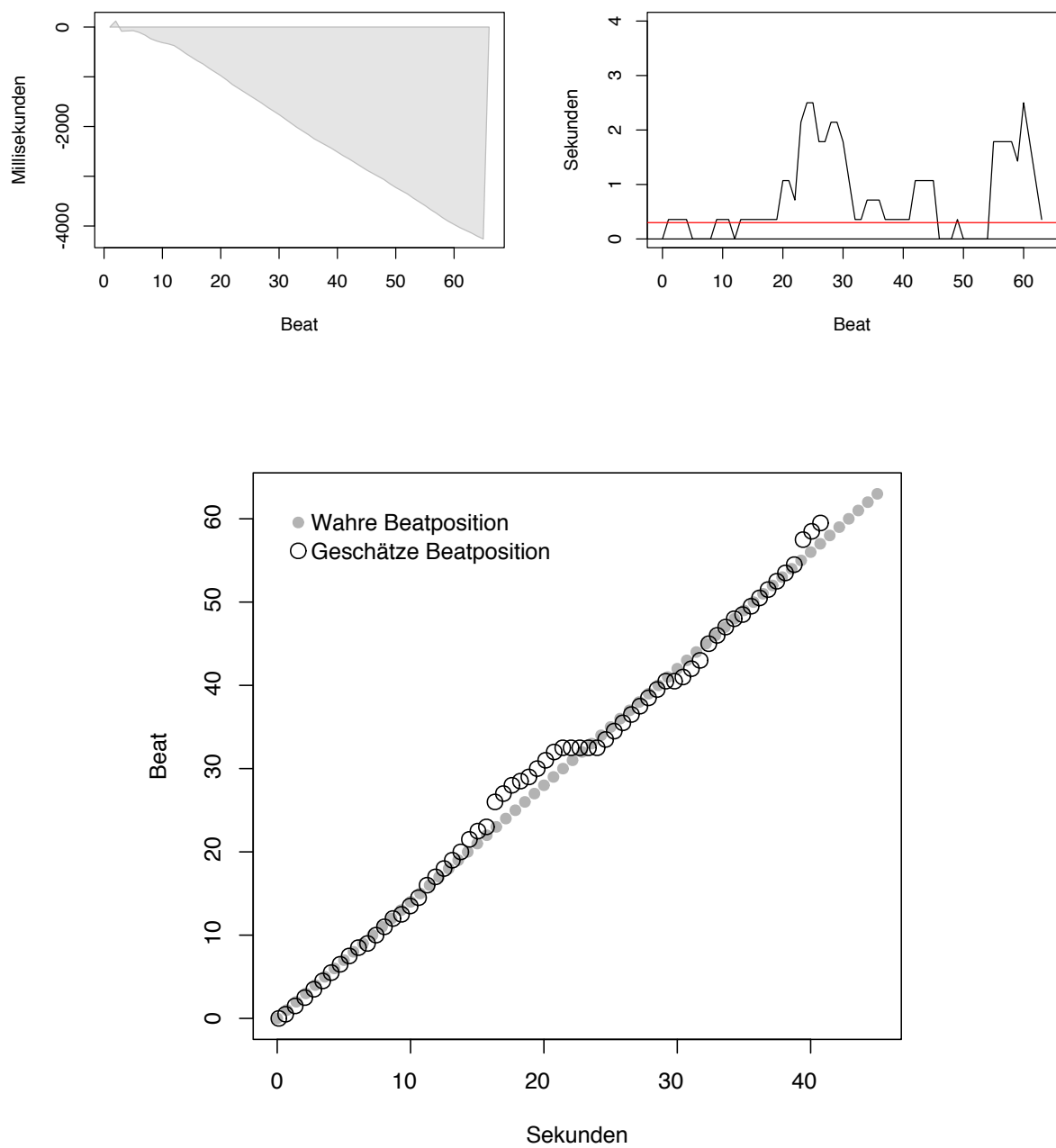


Abbildung 25: Durchlauf des Stücks 01_SS3-84-Bb.comp.

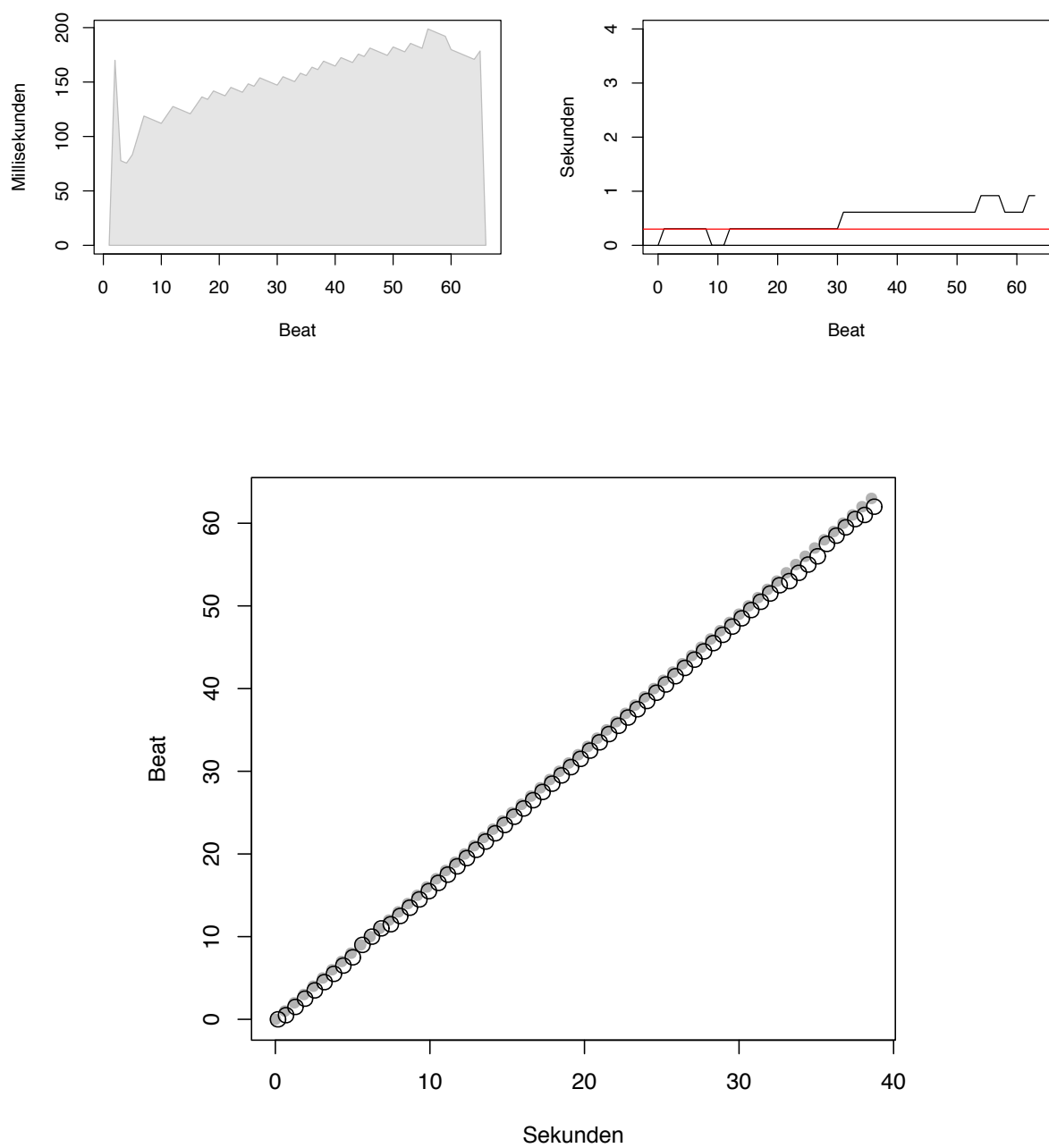


Abbildung 26: Durchlauf des Stücks 01_SS3-98-C_comp.

B Implementierung der Audioverarbeitung mittels Essentia

Für die Erzeugung der Ansatzerkennungsfunktion sowie der HPCP-Features wird die Softwarebibliothek Essentia genutzt. Die konkrete Umsetzung der Funktionalität wird in den untenliegenden Codeabbildungen präsentiert.

Listing 1: Verwendung der Softwarebibliothek Essentia zur Erzeugung der Ansatzerkennungsfunktion.

```
1 from essentia.standard import OnsetDetection, Windowing, FFT,
   CartesianToPolar
2
3 onset_detection_function = OnsetDetection(method='hfc')
4 windowing_function = Windowing(type='hann')
5 fast_fourier_transform = FFT()
6 cartesian_to_polar = CartesianToPolar()
7
8 #frame size = 1024; hop size = 512
9 def onset_function(frame):
10     mag, phase = cartesian_to_polar(fast_fourier_transform(
11         windowing_function(frame)))
12     return onset_detection_function(mag, phase)
```

Listing 2: Verwendung der Softwarebibliothek Essentia zur Berechnung der HPCP-Features.

```
1 from essentia.standard import Windowing, Spectrum, SpectralPeaks, HPCP
2
3 windowing_function = Windowing(type='blackmanharris62', size=2048)
4 spectrum = Spectrum()
5 spectral_peaks = SpectralPeaks(orderBy='magnitude',
6     magnitudeThreshold=0.00001,
7     minFrequency=80,
8     maxFrequency=3500,
9     maxPeaks=60)
10 hpcp = HPCP(size=12,
11     minFrequency=20,
12     maxFrequency=3500,
13     weightType='cosine',
14     nonLinear=False,
15     windowSize=1.)
16
17 # input whole sequence of frames to be estimated
18 def hpcp_function(frame):
19     freq, mag = spectral_peaks(spectrum(windowing_function(frame)))
20     return hpcp(freq, mag)
```

Erklärung

Ich erkläre hiermit gemäß §17 Abs .2 APO, dass ich die vorstehende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort, Datum

Unterschrift