IEM – Institut für Elektronische Musik und Akustik

Technische Universität Graz

Projekt Toningenieur WS-2007

# Automatische Melodietranskription

Autor: Amir Rahimzadeh

Betreuer: Dr. Alois Sontacchi

# Inhaltsverzeichnis

# Kapitel 0 – Einleitung

Trankription von Musik ist der Vorgang des Hörens einer muskalischen Darbietung und des Extrahieren eines Notentextes aus dem Gehörten.

Die Fähigkeit das Gehörte in einem Notentext zu übersetzen setzt musikalisches Wissen und ein geschultes Gehör voraus und ist deshalb meist erfahrenen Musikern vorbehalten. Grundlegend für eine Transkription ist das Erkennen der Töne (Anfang/Ende, Engl. *„note onset“* / *„note offset“)* und deren Tonhöhe (Engl. *„pitch“* oft auch *„F0“*).

Zum Auffinden der Noten bedient man sich bei Automatischen Musiktranskriptions-Systemen sogenannter Detektionsfunktionen, welche zum Erkennen transienter Anteile im digital vorliegenden Audiosignal dienen.

Die Tonhöhe (Engl.: *„pitch“*) ist eine perzeptive Größe und hängt vom Grundton „F0“ und von der Obertonstruktur eines Klanges ab. Sie ist definiert, als die Frequenz eines Sinustons der den gleichen Tonhöheneindruck vermittelt, wie der zu analysierende Klang. Die Tonhöhebestimmung kann mittels unterschiedlicher Methoden im Zeitbereich wie auch im Frequenzbereich erfolgen, welche sich vor allem in der Fehleranfälligkeit und im Rechenaufwand unterscheiden.

Sind Notenpositionen und Tonhöhen bekannt, so kann aus der vorliegenden Information ein Notentext generiert werden.

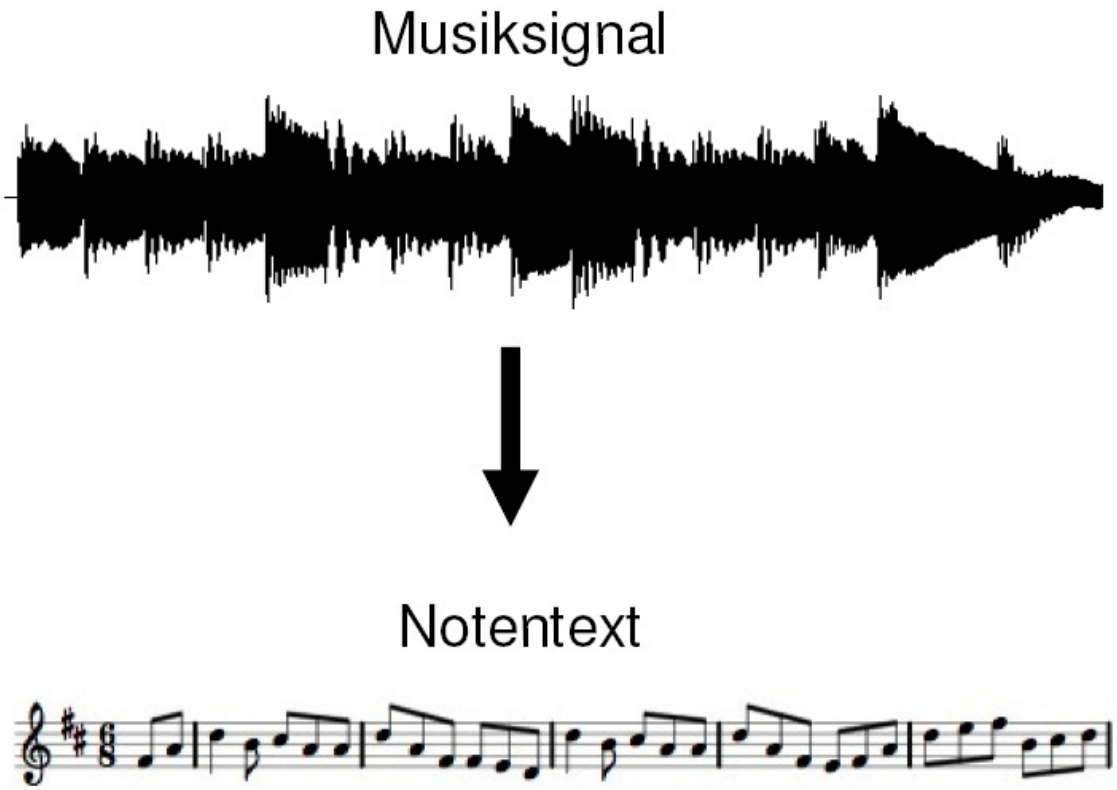



Abb. 1: Prinzip Musiktranskription

# Kapitel 1 – Das System

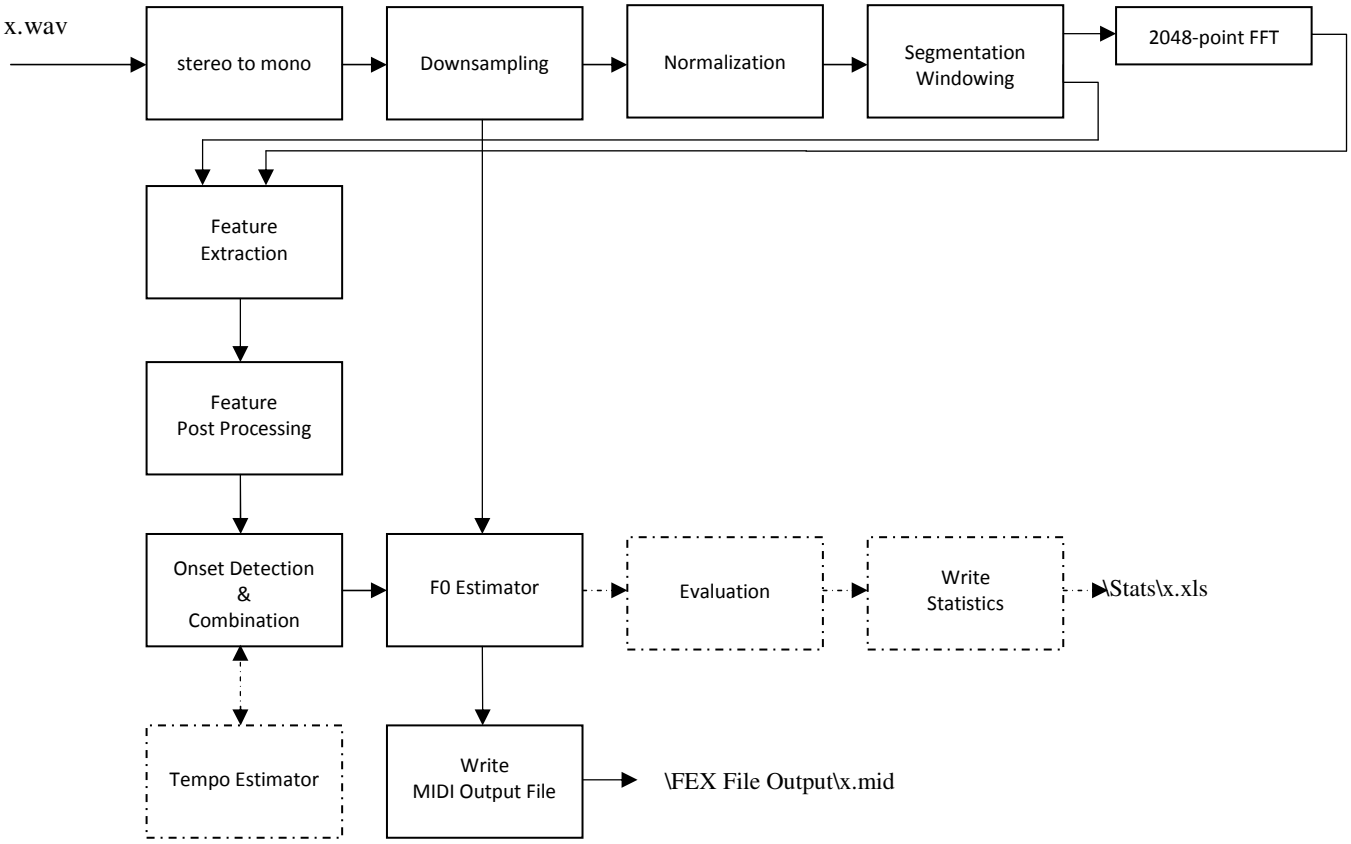



Abb. 2: Blockschaltbild des Systems

Das System erhält am Eingang eine vom Benutzer gewählte pcm-kodierte Audiodatei. Das Audiosignal wird in überlappende Segmente entsprechend der gewählten Segmentlänge (engl. „*frame size*“) u. Sprunggröße (engl. „*hop size*“) unterteilt. Um „leakage“ Effekte bei der darauffolgenden FFT zu vermeiden, werden die Segmente mit einem ungeradzahligem *hanning*-Fenster multipliziert. Die gefensterten Datensegmente werden zur Berechnung von 4 Audiodeskriptoren (engl. „*Features*“) bzw. Detektionsfunktionen herangezogen, welche zum Auffinden der Onsetzeitpunkte dienen.

Die Detektionsfunktionen schlagen zu Beginn eines Klangs je nach Klangqualität (Lautstärke, Klangfarbe, Spielweise) unterschiedlich stark aus (engl. „*peaks*“). Es ist daher eine Nachbearbeitung der Detektionsfunktionen nötig. Dies geschieht in der Stufe „Feature Post Processing“, wo mittles „dynamischen treshholdings“ und Normalisierens die relevanten „*peaks*“ für die nächste Stufe aufbereitet werden.

Die Stufe „Onset Detection & Combination“ liefert die Zeitpunkte der geschätzten Notenanfänge und kombiniert Onsets unterschiedlicher Detektionsfunktionen, die in ein selbes vordefiniertes zeitliches Intervall fallen, miteinander, um die Robustheit gegenüber Fehldetektion zu erhöhen.

Des weiteren wurde ein Tempo-Schätzer implementiert, dessen Schätzung herangezogen werden kann um Onsetzeitpunkte auszuschließen, die zu weit von einem idealen Zeitraster abweichen. Dies funktioniert nur bedingt, da sich menschliche Darbietungen meist gerade durch das Abweichen von den exakten Zählzeiten auszeichnen.

Der „F0-Estimator“ dient zur Bestimmung der Frequenz des Grundtons einer gespielten Note. Er arbeitet auf Basis von Autodifferenz-Funktionen die starke Verwandschaft zur Autokorellations-Funktion aufweisen. Am Eingang erhält der „F0-Estimator“ das Audiosignal „x.wav“, die Onsetzeitpunkte sowie das geschätzte Tempo in „Beats per minute“ (kurz: BPM) des „Tempo Estimators“. Das zwischen zwei Onsetzeitpunkten liegende Signal wird nicht direkt zur F0-Bestimmung verwendet, sondern in Segmente der Länge von 1/32-Noten unterteilt. Dadurch sollen zusätzliche Noten erkannt werden, die von der Onset Detection nicht gefunden wurden.

Aus den vorliegenden Informationen wird in Folge eine MIDI-Datei generiert, da MIDI unseres Erachtens die universellste Representation musikalischer Daten darstellt und ein Notentext mittels Sequencer Software daraus leicht erstellt werden kann.

# Kapitel 2 – Onset Detection

## 2.1.: Allgemeines

Die Analyse der Onsets ist Ausgangspunkt vieler Algorithmen zur automatischen Analyse digitaler Audiodaten, wie Tempo-, Beat- und Rhythm-Estimation [5] und -Tracking, automatischer Segmentierung von Audiodaten, Musiktrankription [3],[4] sowie score-following (Anzeigen der gespielten Note im Notentext). Die grundlegende Schwierigkeit der Aufgabe besteht darin tatsächliche Onsets in der Detektionsfunktion zu erkennen und von anderen Signalschwankungen, die aufgrund von Vibratospielweise bzw. Modulationen beim Ausklang eines Tons entstehen können, zu unterscheiden.

Im Folgenden sind die notwendigen Schritte zur Detektion und Lokalisation der Onsetzeitpunkte schematisch dargestellt.

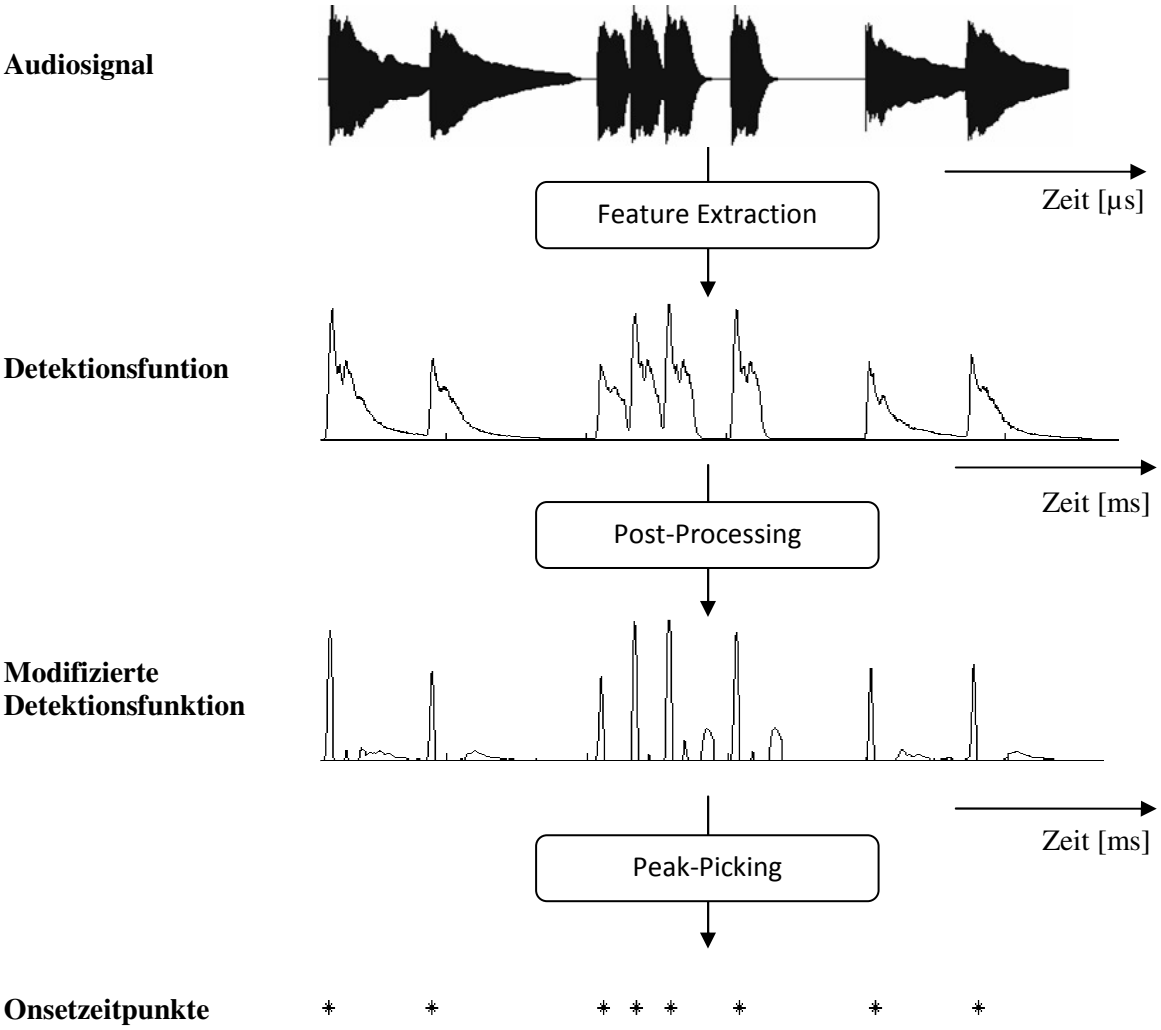



Abb. 3: Arbeitsschritte des Detektionsalgorithmus – schematisch

Audiodaten enthalten im Allgemeinen transiente und stabile Anteile (siehe Abb. 4). Transienten sind Folge des Einschwingvorgangs bei Anspielen eines Tons auf einem Instrument. Bei Instrumenten die Klänge mit harmonischer Obertonstruktur erzeugen, beschränken sie sich auf die ersten 10-30 ms. Nachdem alle Harmonischen eingeschwungen sind, folgt der stabile Anteil. Dieser ist charakterisiert durch waagrechte Linien im Spektrogramm und wird oft auch als quasi-stationär bezeichnet, da sich die Sinuskomponenten in Frequenz und Amplitude von frame zu frame nur geringfügig ändern.

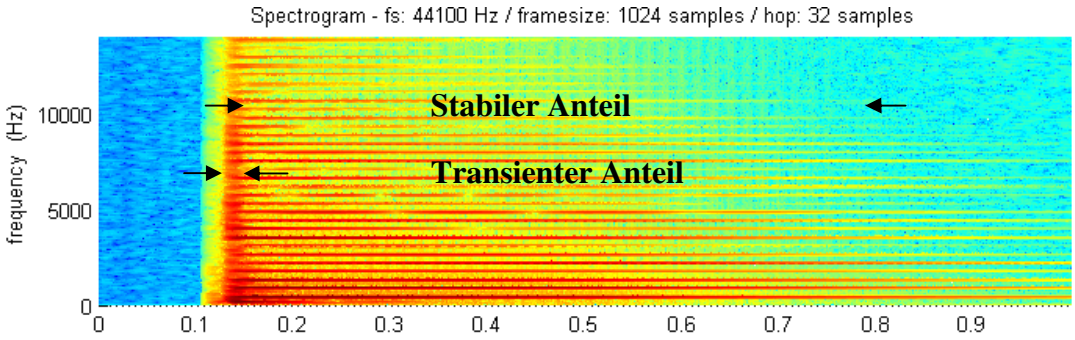



Abb.: 4 – Spektrale Charakteristika transienter und stabiler Signalanteile am Beispiel eines Pianoklangs

Zum Erkennen transienter Anteile im Audiosignal und somit höchstwahrscheinlicher Notenanfänge bedient man sich der bereits erwähnten Detektionsfunktionen. Diese gewinnt man durch Anwendung mathematischer Transformationen auf das segmentierte Eingangssignals oder einer spektralen Repräsentation desselbigen. Ein Beispiel einer Detektionsfunktion (hier: Spectral Distance) ist in Abb. 2 zu sehen. Die Onsetzeitpunkte werden mittels „Peak-Picking“ (Lokalisation der Maxima) gefunden.

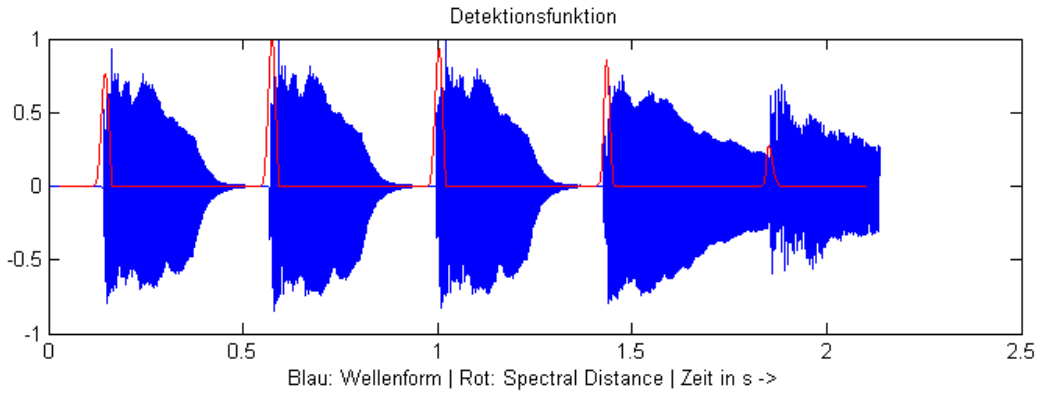



Abb.: 5 Wellenform (blau) und für das Peak-Picking aufbereitete Detektionsfunktion (rot)

Die Segmentierung stellt einen wichtigen Informationsreduktionsschritt dar. Dieser ist nötig, da die hoch korellierten Audiodaten ihre statistischen Eigenschaften nur langsam im Vergleich zur Sampling Rate (Bereich: µs) ändern und perzeptiv relevante Änderungen auf einer anderen Zeitbasis nämlich im ms-Bereich stattfinden. Außerdem liegt die zeitliche Auflösung des menschlichen Gehörs laut [7] bei ca. 10 ms, d.h. 2 aufeinanderfolgende Töne müssen mindestens diesen zeitlichen Abstand zueinander aufweisen, um als 2 getrennte akustische Ereignisse wahrgenommen zu werden.

Detektionsfunktionen unterscheiden sich darin unterchiedliche Tonqualitäten (abhängig von Instrument, Spielweise, Lautstärke) verschieden stark aufzuzeigen. Im folgenden sieht man vier mal den selben Ton mit unterschiedlichen Instrumenten (Piano, Trompete, Klarinette, Synthesizer) gespielt und darunter die von uns verwendeten Features. Offensichtlich gibt es kein Feature, welches bei allen Instrumenten gleich ausschlägt.

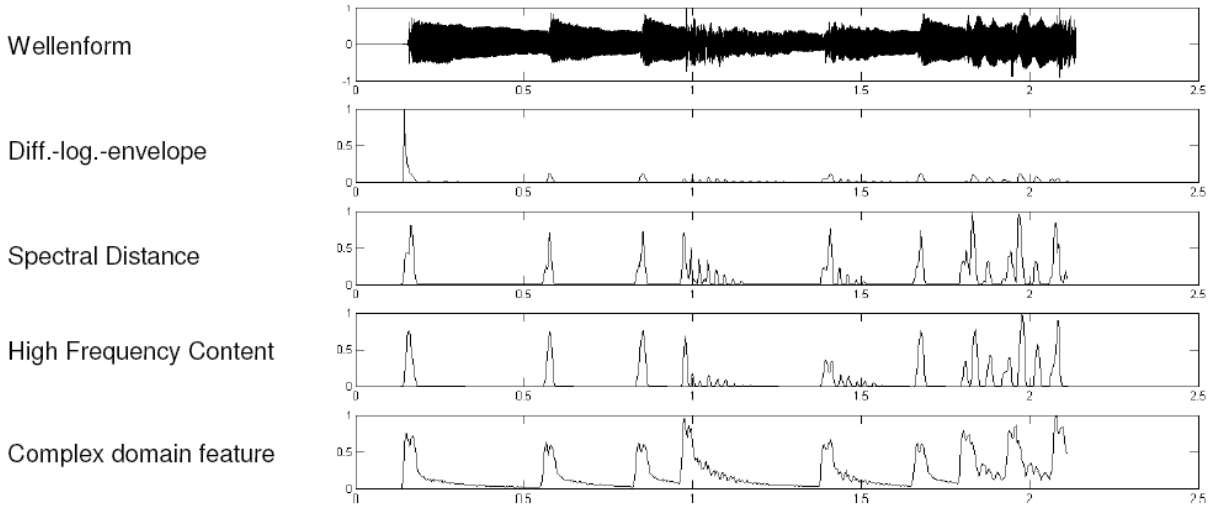



Abb.: 6: Qualitativer Vergleich der Onsetdetektionsfunktionen am Beispiel eines Pianoklangs

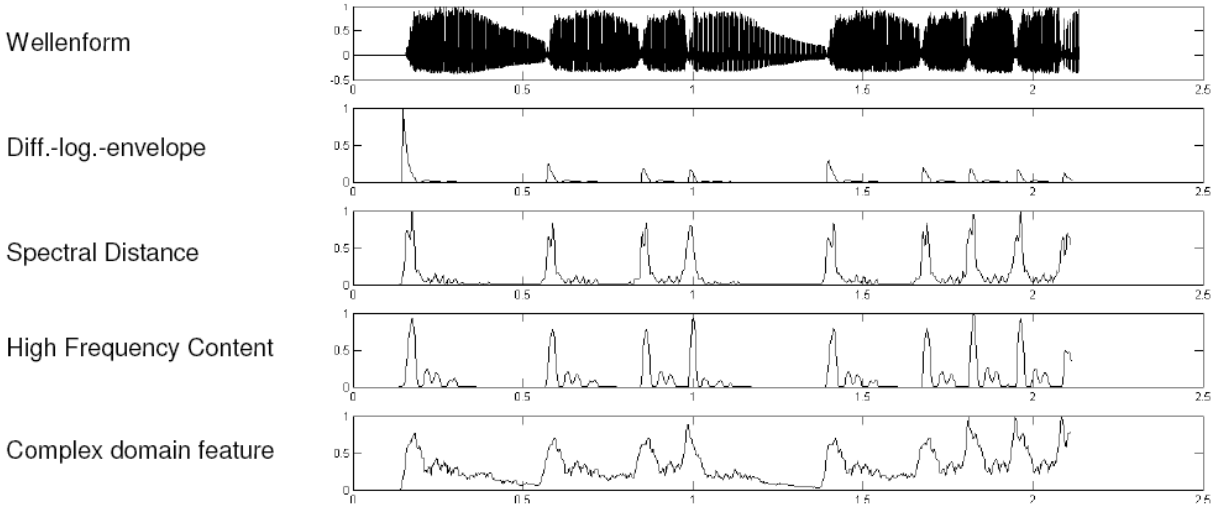



Abb.: 7: Qualitativer Vergleich der Onsetdetektionsfunktionen am Beispiel eines Trompetenklangs

Um die Lokalisation der Onsetzeitpunkte unabhängig von der Klangqualität zu machen, haben wir uns entschieden, parallel 4 Detektionsfunktionen zu verwenden und die daraus abgeleiteten Onsets zu kombinieren. Dies erhöht die Zuverlässigkeit der extrahierten Onsetzeitpunkte und bietet die Möglichkeit unwahrscheinliche Onsets, welche z.B. nur in einer Funktion auftreten von der Transkription auszuschließen.

Die Vorteile gegenüber Systemen ohne explizite Onset Detection wie in [1] , wo ausschließlich das Ergebnis des F0-Estimators und die Einhüllende des Signals herangezogen wird um das Audiosignal in eine Folge von Noten zu unterteilen, sind folgende. Sind Notenpositionen bekannt, so kann einerseits das Datensegment, das zur Tonhöhenschätzung verwendet wird, vergrößert werden. Andererseits können so Transiente Signalanteile (die ersten 10-30 ms zu Notenbeginn), welche keine glaubwürdigen F0-Schätzungen zulassen, von der Tonhöhenschätzung ausgenommen werden. Beide Maßnahmen dienen dazu die Robustheit und Zuverlässigkeit der Tonhöhenschätzung zu erhöhen.

## 2.2.: Verwendete Audiodeskriptoren

Energie basierte Audiodeskriptoren:

Bei der Analyse von Audiosignalen in Bezug auf ihre Einhüllende ist augenscheinlich, dass Notenanfänge mit einem aprupten Anstieg der Einhüllenden einhergehen. Frühere Methoden zur Onsetbestimmung [6] beruhten ausschließlich auf dem Prinzip starke positive Änderungen der Einhüllenden des Audiosignals zu detektieren. Diese Methode funktioniert sehr gut für perkussive- , klar seperierte Töne, weniger jedoch für Klänge mit „langsamem“ Einschwingvorgang wie bei Streich- und Holzblasinstrumenten.

Zur Berechnung eines Wertes der Einhüllenden bzw. der lokalen Energie zum Zeitpunkt n werden die Samples im symmetrisch um den Analysezeitpunkt liegenden Segment quadriert, mit einem Fenster w(m) multipliziert und aufsummiert (Eq. 1).

$$E(n) = \frac{1}{N} \sum_{m=-N/2}^{m=+N/2} x(n+m)^2 \cdot w(m)$$

Eq. 1: „Envelope Follower“

Das resultierende Signal (Abb.: 4.b) ist nicht unmittelbar für die „Onset Detection“ mittels Peak-Picking geeignet. Eine Verbesserung erfolgt durch Ableiten der Funktion bzw. Differenzbildung der zeitdisskreten Folge (Abb.: 4.c). Stellen stärkster Steigung werden dadurch zu lokalen Maxima und Minima in der resultierenden Funktion.

Psychoakustische Studien sprechen dafür, dass die Wahrnehmung von Lautheit logarithmischer Bewertung unterliegt. Außerdem werden Lautheitsunterschiede $\Delta E$ laut [8] relativ zur Gesamtlautheit $\Delta E/E$ des Signals wahrgenommen.

Entsprechend der Regel für „logarithmische Ableitung“ kann man den Quotient wie folgt schreiben:

$$\frac{\partial E/\partial n}{E} = \frac{\partial(\log E)}{\partial n}$$

Eq. 2: Umformung des Quotienten mittels
Regel für „logarithmische Ableitung“

Laut [9] simuliert diese erste Ableitung der log-Energiefunktion nach der Zeit bzw. die erste Differenz im Zeitdiskreten (Abb.: 4.d) die Beurteilung von Lautheit durch das menschlichen Gehörs.

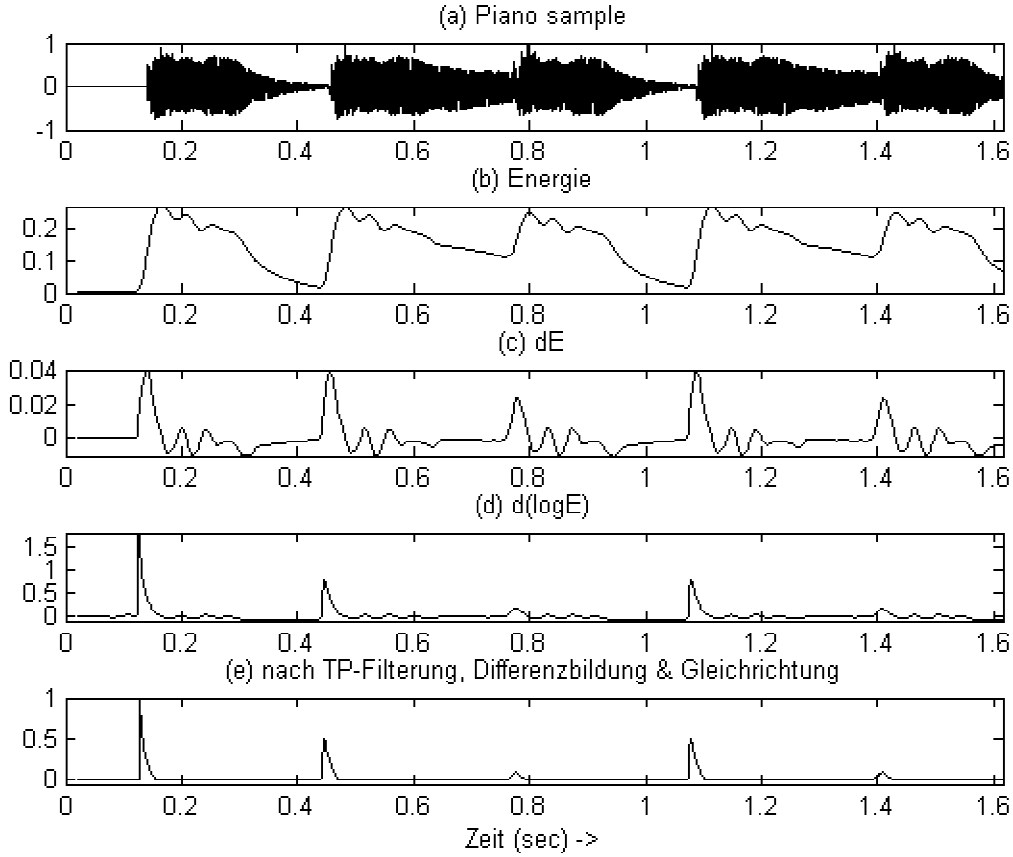



Abb. 8: (a) Wellenform, (b) lokale Energiefunktion,
(c) 1. Ableitung bzw. 1. Differenz der Energiefunktion,
(d) 1. Differenz der logarithmierten Energiefunktion,
(e) für das „Peak-Picking“ aufbereitete Detektionsfunktion

High Frequency Content – HFC

Transienten sind spektral gesehen breitbandig, d.h. die Energie ist über alle Frequenzen annähhernd gleich verteilt. Da normalerweise die Energie in Musiksignalen im tieffrequenten Bereich konzentriert ist, lassen sich Transienten vor allem im hochfrequenten Bereich leicht detektieren. HFC ist ein Feature welches genau diese Eigenschaft des Signals nützt. Es wird die lokale spektrale Energie berechnet, mit der Verfeinerung, dass vor dem Aufsummieren jeder bin mit einem frequenzabhängigen Gewicht multipliziert wird. Wir verwenden lineare Gewichtung der bins proportional ihrer Frequenz, wodurch der hochfrequente Teil des Signals überbetont wird.

$$HFC(n) = \frac{2}{N}\sum_{k=0}^{\frac{N}{2}-1} X(n,k)^2 W(k) \qquad \text{mit } W(k) = |k|$$

Eq. 3: High Frequency Content mit frequenzproportionaler linearer Gewichtung W(k) der bins X(k) aus der segmentweisen STFT

Spektraler Fluss (Engl.: „Spectral Flux“ bzw. „ Spectral Distance“)

Der spektrale Fluss ist ein Feature, welches spektrale Änderungen von einem zum nächstem Analysezeitpunkt erfasst. Die Änderungen werden zwischen zeitlich aufeinanderfolgenden bins berechnet und aufsummiert. Sind im zu analysierenden Datensegment hauptsächlich Harmonische enthalten und ist somit die Energie auf ein paar Sinuskomponenten beschränkt, so weist die Funktion einen sehr kleinen Wert auf, da die meisten Bins und somit auch die Differenz zwischen aufeinanderfolgenden 0 ergibt. Transienten hingegen sind breitbandig, d.h. die Energie ist annähernd gleich auf alle Bins verteilt, was einen hohen Wert der Detektionsfunktion zu Folge hat.

$$SD(n) = \frac{1}{K}\sum_{k=1}^{K} X(n+1,k) - X(n,k)$$

K…number of bins

Eq. 4: Spectral Flux

Phase deviation:

Eine alternative zu rein energiebasierter „Onset Detection“ bietet die Verwendung der Phaseinformation, die aus der „Short Time Fourrier Transform“ (STFT) der Signalsegmente zu Verfügung steht. Harmonische Klänge bestehen aus Sinuskomponenten, deren Frequenz während des stationären Teils eines Klangs mehr oder weniger konstant ist. D.h. für eine Sinuskomponente der Frequenz k bleibt auch die Differenz der Phasenwerte $\Delta\,\varphi_k[m] = \varphi_k[m] - \varphi_k[m\text{-}1]$ der STFT zweier aufeinanderfolgender Segmente näherungsweise konstant (m...frame Index). Es gilt also:

$$\Delta\,\varphi_k[m\text{-}1] \approx \Delta\,\varphi_k[m]$$

Eq. 5: Frame-to-frame Phasenfortschreitung von quasi-stabilen Sinuskomponenten

Diese Eigenschaft kann zur Bildung einer Detektionsfunktion verwendet werden [10]. Man definiert eine Funktion, welche die Abweichungen der Phasenänderung $\Delta\ \varphi_k[m]$ von der prädizierten Phasenänderung $\Delta\ \varphi_k[m-1]$ erfasst:

$$d_{\varphi,k} = \text{princarg}(\Delta\ \varphi_k[m] - \Delta\ \varphi_k[m-1]) = \text{princarg}(\varphi_k[m] - 2*\varphi_k[m-1] + \varphi_k[m-2])$$

Eq. 6: Erfassen der Änderungen der Phaseninkremente
über den gesamten Frequenzbereich

Für die Berechnung des Phaseninkrements muss die „unwrapped phase“ verwendet werden, da die Phase zwischen 2 frames um mehr als $2\pi$ steigen kann und das Ergebnis sonst mehrdeutig wäre.

Um eine Aussage über den gesamten Frequenzbereich treffen zu können, werden die Phaseninkremente für alle Frequenzen k in einem Histogramm $h(|d_\varphi|)$ erfasst. Während des stationären Teils eines Klangs wird die Funktion $d_{\varphi,k}$ viele kleine Wert aufweisen und die Verteilung somit schmal und spitz um den Wert 0 ausfallen, während transiente Bereiche unterschiedlichste Werte von $d_{\varphi,k}$ und somit eine breite und flache Verteilung zu Folge haben. Dieser Unterschied kann sehr gut durch Berechnung des Mittelwerts der Verteilung abgebildet werden.

$$\eta(m) = \text{mean}\{h(|d_\varphi|)\}$$

Eq. 7: Der Mittelwert der Verteilung h wird segmentweise
berechnet und dient als Detektionsfunktion

Auf Phaseninformationen basierende „Onset Detection” hat den Vorteil vor allem langsame, tonale Onsets (wie von Holzblas- und Streichinstrumenten) zu erfassen, bei denen energiebasierte Features oft versagen. Nachteilig ist die hohe Anfälligkeit auf Phasenverzerrungen die ebenfalls zu hohen Werten der Detektionsfunktion führen. Diese treten vor allem bei Komponenten sehr geringer Energie verursacht durch „FFT-channel crosstalk“ von benachbarten Kanälen auf. Eine Verbesserung der Situatuion erreicht man durch Gewichtung der Phaseninkremente mit der Amplitude des entsprechenden bins der STFT.

Des weiteren neigt dieses Feature dazu, Notenenden die ebenfalls Transiente Bereiche darstellen, genauso stark aufzuzeigen wie Notenanfänge, was eine Unterscheidung der zwei Bereiche – Notenanfang oder Notenende– sehr schwierig macht.

Aufgrund der hohen Fehleranfälligkeit wurde dieses Feature von der kombinierten Onset Detection ausgenommen.

Eine Verbesserung der Problematik ergibt sich durch gemeinsame Verwendung von Amplituden- und Phaseninformationen, was im folgenden erklärt wird.

Complex domain feature

„Features“ basierend auf der Analyse der lokalen Energie liefern gute Ergebnisse bei perkussiven Onsets, während „Features“ die auf der Analyse der Phaseninformation beruhen, dazu neigen, langsame, tonale Onsets besser abzubilden. Es ist daher naheliegend, zu versuchen, die positiven Eigenschaften beider Methoden zu kombinieren. Eine gemeinsame und gleichzeitige Auswertung der Amplituden- und Phaseninformation kann laut [10] nur im Komplexen erfolgen. Für stationäre Signalanteile kann angenommen werden, dass Amplitude und Frequenz annähernd konstant bleiben.

Der prädizierte Wert des k-ten STFT bin’s in Polarform kann somit folgendermaßen geschrieben werden:

$$\underline{\hat{X}}(m,k) = \left|\hat{X}(m,k)\right| e^{j\hat{\varphi}_k(m)}$$

Mit $\hat{X}_k(m) = X_k(m-1)$ und $\varphi_k(m) = princarg\{2\varphi_k(m-1) - \varphi_k(m-2)\}$

Eq. 8: Prädizierter Wert des k-ten STFT bin’s des m-ten Segments in Polarform

Der prädizierte Amplitudenwert $\hat{X}(m,k)$ zum Analysezeitpunkt m wird entsprechend dem Wert $X(m-1,k)$ des vorherigen Segments gewählt. Den prädizierten Phasenwert erhält man durch Summation des Phasenwerts des vorherigen Frames und der Phasendifferenz der vorangehenden Frames des entsprechenden bins.

Die Abweichung der gemessenen Werten $\underline{X}(m,k)$ der STFT und den prädizierten Werten $\underline{\hat{X}}(m,k)$ kann mittels Euklidischer Distanz berechnet werden und dient als Maß für die Stationarität des k-ten bins.

$$\Gamma_k(m) = \left\{\left[\Re\left(\underline{\hat{X}_k}(m)\right) - \Re\left(\underline{X_k}(m)\right)\right]^2 + \left[\Im\left(\underline{\hat{X}_k}(m)\right) - \Im\left(\underline{X_k}(m)\right)\right]^2\right\}^{1/2}$$

Eq. 9: Euklidische Distanz zweier komplexer Zeiger

Durch Multiplikation der komplexer Zeiger mit einem komplexen Phasenterm $e^{j\varphi_{rot}}$ kann man die Zeiger so rotieren, dass $\underline{\hat{X}_k}(m)$auf der reellen Achse zu liegen kommt und sich die Gleichung wie folgt vereinfacht:

$$\Gamma_k(m) = \left\{\left|\hat{X}_k(m)\right|^2 + |X_k(m)|^2 - 2\ \left|\hat{X}_k(m)\right|\ |X_k(m)|\cos(\mathrm{d}\varphi,\mathrm{k})\right\}^{1/2}$$

Eq. 10: Vereinfachte Gleichung nach Mapping von $\hat{X}_k(m)$ auf die reelle Achse

Durch Aufsummieren der Ergebnisse der Funktion für alle bins erhält man die Detektionsfunktion η(m).

$$\eta(m) = \sum_{k=0}^{K} \Gamma_k(m)$$

Eq. 11: Resultierende Detektionsfunktion

## 2.3.: Feature Post-Processing

Die Detektionsfuntionen sind im unbearbeiteten Form nicht unmittelbar für das Peak-Picking geeignet. Zu den gängigen Aufbereitungsmöglichkeiten von Detektionsfunktionen zählen Tiefpassfilterung, Normalisieren, Differenzbildung sowie Entfernen des Gleichanteils, wodurch ein großer Teil irrelevanter Information entfernt werden kann (siehe Abb. 8).

## 2.3.1: Thresholding

Je nach Typ der Detektionsfuntkion können trotz vorangehenden „Post-Processings“ „Peaks“ auftreten, die keinen Onsetzeitpunkten entsprechen. Üblicherweise führt man deshalb eine Entscheidungsschwelle (engl. „threshold“) ein, wonach „Peaks“ die darüberliegen als tatsächliche „Onsets“ gewertet werden und darunterliegende ausgeschlossen werden.

Dies kann für Signale geringer Dynamik gut funktionieren, im Allgemeinen enthalten Musikstücke jedoch lautere und leisere Noten bzw. Passagen, da die Dynamik eines der wichtigsten Ausdrucksmittel in der Musik ist. Dies hat allerdings auch höhere und niedrigere Peaks in der Detektionsfunktion zu Folge. Es ist daher schwer einen fixen Schwellwert (Abb. 9) zu bestimmen der das oben genannte Kriterium, nämlich die Unterscheidung in Onset-relevante und irrelevante Peaks, für einen gesamten Song bzw. mehrere Songs erfüllt.

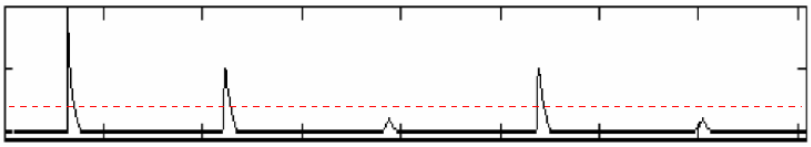


Abb. 9: Statischer Schwellwert

Zielführender ist es daher einen Signal-adaptiven Schwellwert (Abb. 10) zu verwenden. Dieser wird oft als tiefpassgefilterte Version der Detektionsfunktion berechnet. Nachteilig dabei ist, dass einzelne große Peaks hohe Schwellwerte bewirken und dazu neigen daraufolgende kleinere Peaks „zu maskieren“. Andere Methoden, die Schwellwerte auf Basis von „Percentiles“ (wie z.B. der lokale Medianwert) berechnen, sind weniger anfällig auf diese Art von Fehlern ist.

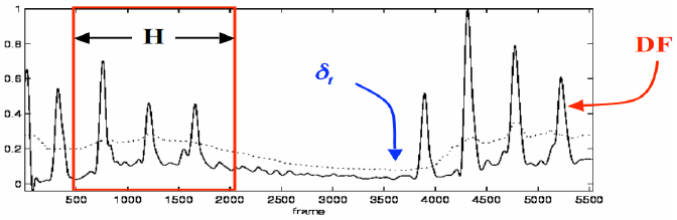



Abb. 10: Adaptiver Schwellwert

Der von uns verwendete Schwellwert verwendet einen simplen Integrator, der die lokale Standardabweichung der Detektionsfunktion als Eingangssignal erhält.. Peaks die über diesem Schwellwert liegen werden als Onsetzeitpunkte gewertet.

$$threshold(m) = a * threshold(m-1) + b * std(f(M))$$

für M = -H/2 bis +H/2

Eq. 12: Schwellwertberechnung

## 2.4.: Onset Combination

Detektionsfunktionen bilden verschiedene Signaleigenschaften unterschiedlich gut ab und sind meist anfällig auf einen speziellen Fehlertyp. Wir haben uns daher entschieden, gleichzeitig vier verschiedene Detektionsfunktionen (siehe 2.2: Verwendete Audiodeskriptoren) zu verwenden und die daraus abgeleiteten Onsetzeitpunkte zu kombinieren.

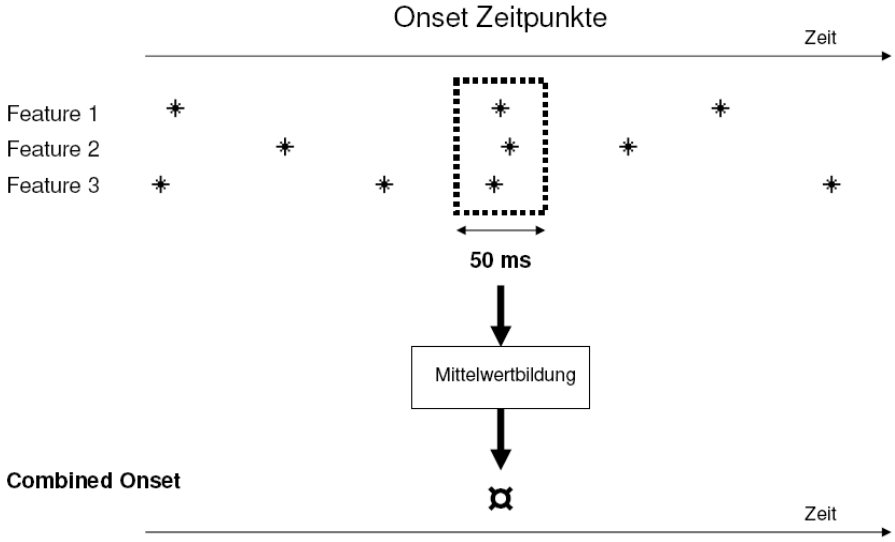



Abb. 11: Kombination der Onsetzeitpunkte verschiedener Features

Dies hat den Vorteil, dass man mit hoher Wahrscheinlichkeit mehr Onsetzeitpunkte findet, als bei Verwendung einer einzelenen Detektionsfunktion. Natürlich steigt damit auch die Zahl der „False Positives“, der fälschlicherweise als Onsets gewertete irrelevante Ausschläge der Funktion.

Um dem entgegen zu wirken weisen wir den Onsets einen Zuverlässigkeitswert zu. Dieser entspricht der Anzahl an Funktionen in denen ein Onsetzeitpunkt erkannt wurde und kann entsprechend den Wert 1-4 annehmen. Onsets können dann nach ihrem Zuverlässigkeitswert gefiltert werden. Momentan werden vom System ausschließlich Onsets akzeptiert, die in mindestens 2 Detektionsfunktionen erkannt wurden.

# Kapitel 3 – F0 Estimator

Der „F0-Estimator“ dient zur Bestimmung der Frequenz des Grundtons einer gespielten Note.

Es gibt zahlreiche Algorithmen, die für diese Aufgabe entwickelt wurden, welche sich vor allem in Fehleranfälligkeit, Rechenaufwand und der Art der Informationsverarbeitung unterscheiden.
Die Gemeinsamkeit die alle Algorithmen aufweisen, ist die Suche nach Periodizitäten, welche Eigenschaft harmonischer Klänge sind.

Die Algorithmen lassen sich in 3 Kategorien einteilen:

1.) „Spectral location type algorithms“ suchen nach Obertönen, die in harmonischen Verhältnis, nämlich ganzzahligen Vielfachen zur Grundfrequenz stehen. Dazu zählen Algorithmen die auf der Autokorrelation des Signals oder „Harmonic Pattern matching“ beruhen. Die letztere Methode verwendet Spektren mit logarithmierter Frequenzachse, wodurch die Abstände zwischen Teiltönen unabhängig von F0 werden. Dieses log-Spektrum wird mit einem idealen Spektralen Muster (1 bei harmonischen Positionen, sonst 0) kreuz-korreliert und somit entspricht die Position höchster Korrelation der Grundfrequenz F0.

2.) „Spectral interval type algorithms“ basieren auf der Beobachtung, dass periodische, aber nicht-sinusförmige Signale periodische Amplitudenspektren aufweisen, deren Peaks in periodischem Abstand F0 auftreten. Bildet man die Autokorrelation des Amplitudenspektrums, so entspricht die Position höchster Korrelation der gesuchten Frequenz des Grundtons F0. Es wird berichtet, dass diese Methode besser für leicht inharmonische Klänge funktioniert, da Frequenzintervalle zwischen Partialtönen stabiler sind als absolute spektrale Positionen.

3.) „Auditory model type algorithm“ machen sich psychoakustisches Wissen zunutze, um gewisse Vorgänge bei der physiologischen Verabeitung des Gehörs von akustischen Signalen zu simulieren. Meist wird das Signal in Subbänder unterteilt. Analyse auf Periodizitäten erfolgt in jedem Subband, was eine gegenseitige Auslöschung von Information benachbarter Subbänder verhindert. Diese Sub-band Korrelationsfunktionen können schließlich zu einer Summen-Korrelationsfunktion durch Integration der Information aller Kanäle zusammengefasst werden. Vorteil dieser Methode ist, dass die Tonhöhe anhand von Periodizitäten vieler Subbandsignale ermittelt wird und somit robust gegen bandbeschränktes Rauschen ist. Nachteilig ist der erhöhte Berechnungsaufwand.

Der von uns verwendete Algorithmus gehört der ersten Kategorie von F0-Schätzern an und wurde auf Basis des YIN-Algorithmus („YIN – A fundamental frequency estimator for speech and musical signals”) [11] implementiert. Er arbeitet anhand von Auto-Differenz-Funktionen, welche starke Verwandtschaft mit der Autokorrelationsfunktion aufweisen. Der Hauptunterschied zwischen Autokorrelationsfunktion und Autodifferenzfunktion liegt darin, dass bei höchster Korrelation der Signale die Funktion anstatt eines „Peaks“ ein „Valley“ (einen Funktionseinbruch) aufweist. Des weiteren zeigt die Autodifferenzfunktion geringere Empfindlichkeit gegenüber Amplituden-Schwankungen die oft zu Verdopplung oder Halbierung der Grundfrequenz führen.

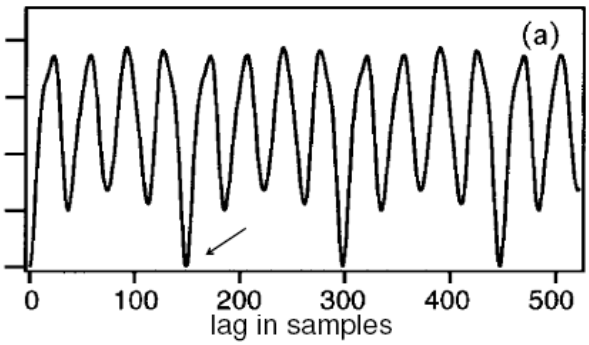



Abb. 12: Autodifferenzfunktion eines Sprachsamples

Der kleinste Zeitversatz bzw. das kleinste „lag“ in Samples (Abb. 12) das die stärkste Korellation aufweist wird zur F0 Bestimmung entsprechend folgender Formel verwendet.

$$F0(m) = \frac{1}{\frac{lag}{fs}} = \frac{fs}{lag} \text{ in Hz}$$

Eq. 13: Berechnung der Frequenz in Hz aus dem Versatz bzw. der Periodendauer in samples

Da das tatsächliche Minimum auch zwischen 2 Samples liegen kann, wird Parabolische Interpolation verwendet um Sub-sample Genauigkeit zu erreichen.

Eine in YIN eingeführte spezielle Normierung der Autodifferenzfunktion bewirkt, dass keine obere Grenzfrequenz nötig ist (Abb. 13). Des weiteren waren für die Wahl des YIN-Algorithmus der relativ geringe Berechnungsaufwand und die und die sehr niedrigen Error-rates ausschlaggebend.

$$d'_t(\tau) = \begin{cases} 1, & \text{if } \tau = 0, \\ d_t(\tau) \Big/ \left[ (1/\tau) \sum_{j=1}^{\tau} d_t(j) \right] \end{cases}$$

Eq. 14: Spezielle Normierung der Autodifferenzfunktion

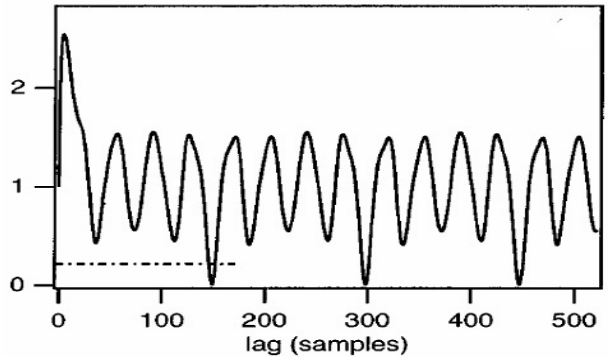



Abb. 13: Cummulative Mean Normalized Difference Function

Die Wahl der Göße des Intervalls, das zur F0 Bestimmung verwendet wird, wird in Abhängigkeit von den Inter Onset Intervall Größe entsprechend dem kleinsten IOI gewählt.

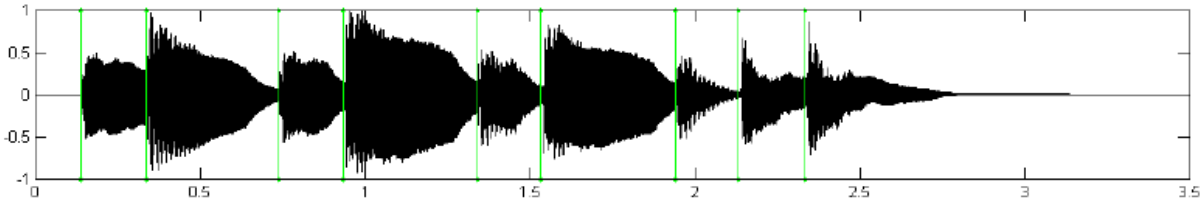



Abb. 14: Darstellung der Inter Onset Intervalle

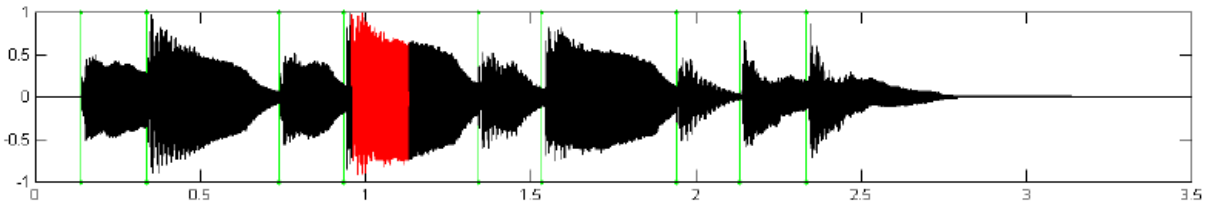



Abb. 15: F0 Analyse auf Basis des kleinsten, häufigsten IOI, framesize in rot dargestellt

Liegt nun eine Frequenzschätzung vor so muss diese in absolute Notenwerte umwandeln um im musikalischen Kontext verstanden zu werden. Dies erfolgt mittels folgender Formel die einen Umrechnung von Frequenz in Hz in MIDI Noten Nummer darstellt.

$$MIDI = 69 + 12 \cdot \log_2\left(\frac{f_0}{440}\right)$$

Eq. 15: Umrechnung von Frequenz in Hz nach MIDI Noten Nummer

# Kapitel 4 – Tempo Estimator

Es wurde ein Temposchätzer realisiert, der auf Basis der Inter Onset Intervalle IOI arbeitet. Das häufigst auftretende IOI wird herangezogen um ein ideales Zeitraster aufzustellen. Um die idealen Zeitpunkte wird ein 50 ms Fenster gelegt. Dasselbe geschieht mit einem Vektor, der die Onsetzeitpunkte enthält. Dann werden die beiden Folgen miteinander korelliert und das Zeitraster variiert. Der Variationswert der die größte Korellation liefert wird zur Temposchätzung und in weiterer Folge zu Bestimmung der Größe des Intervalls zu F0 Bestimmung herangezogen.

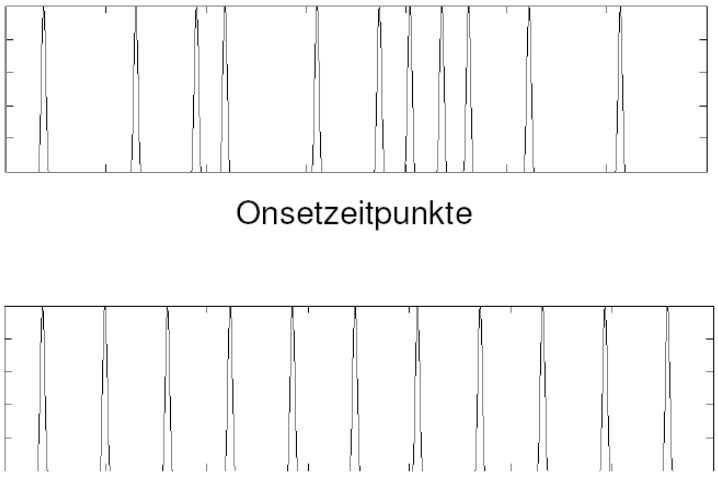
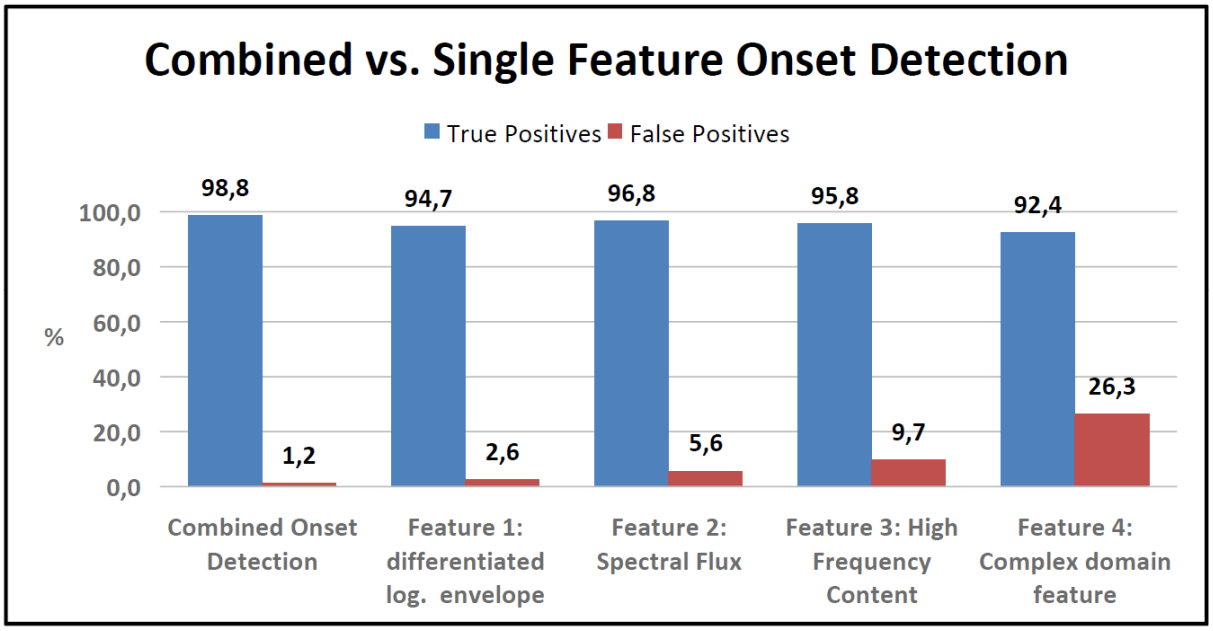


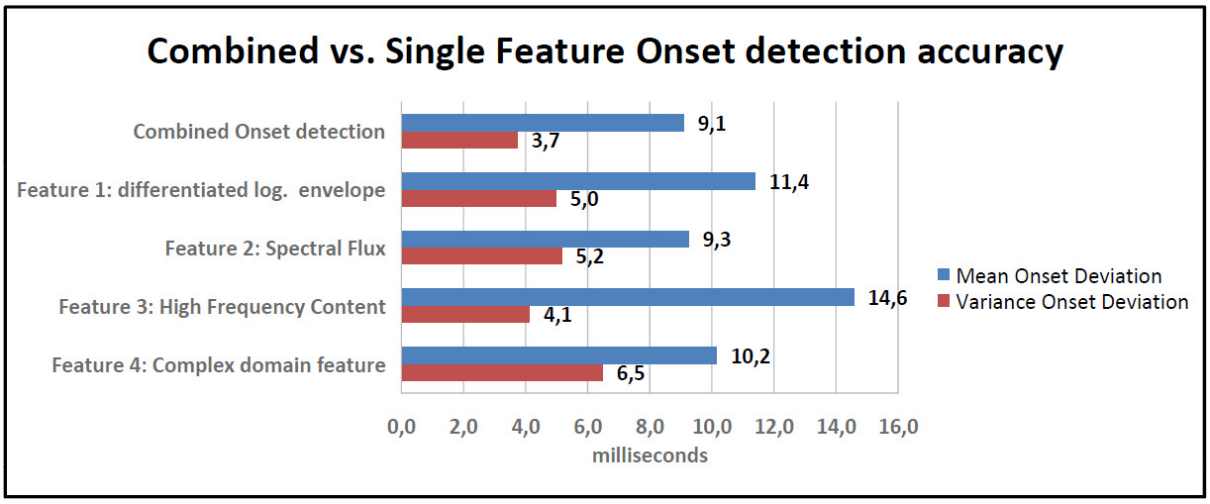

Abb. 16: Temposchätzung durch Korellation des Vektors der Onsetzeitpunkte mit einem idealen Zeitraster

# Kapitel 5 – Evaluation

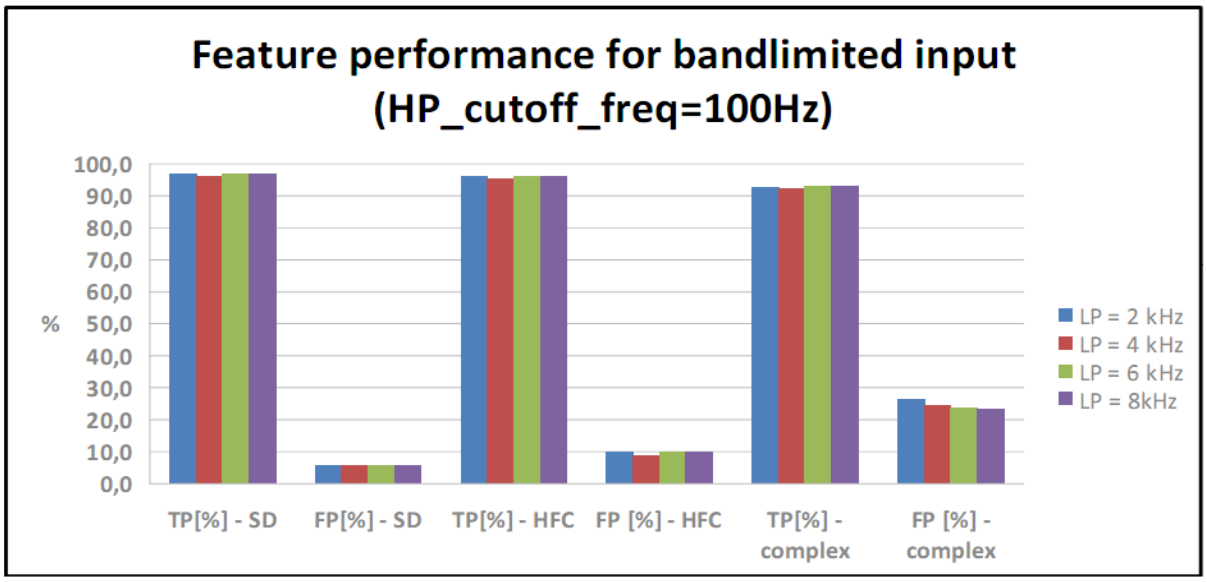
Um ein Transkriptionsystem testen zu können bedarf es einer Referenz zu der man die Ergebnisse des Algorithmus vergleicht. Eine Möglichkeit besteh darin, den erstellten Notentext mit dem Original zu vergleichen, was allerdings sehr zeitaufwendig wäre. Aus diesem Grund haben wir uns entschieden mit MIDI Dateien zu arbeiten. Die MIDI Dateien wurden von diversen Internetseiten kostenlos heruntergeladen. Die Dateien wurden in Folge in ein Sequencer Programm geladen, einzelne Melodiespuren extrahiert, MIDI Dateien der Einzelspuren erzeugt und schließlich mittels Sampler Audiodateien erzeugt. Für jede MIDI Datei wurden bis zu 5 Audiodateien unter Verwendung unterschiedlicher Instrumente generiert. Dies hat den Vorteil, dass Ergebnisse unabhängig vom gespielten Stück verglichen werden können. Die verwendeten Instrumente sind Piano, Trompete, Klarinette, Synthesizer, Synthbass. Instrumente die den Notenumfang der Melodie nicht abdecken konnten wurden ausgespart und nur mit den übrigen Instrumenten synthetisiert. Für den Synthbass wurden die Noten um 2 Oktaven nach unten verschoben. Insgesamt wurden aus 41 Lieder bzw. MIDI Dateien (davon 21 Klassik / 20 Pop) auf diese Weise Audiodateien generiert.
Für die Evaluierung des Systems wurden je Lied 50 Sekunden Audio extrahiert und analysiert.



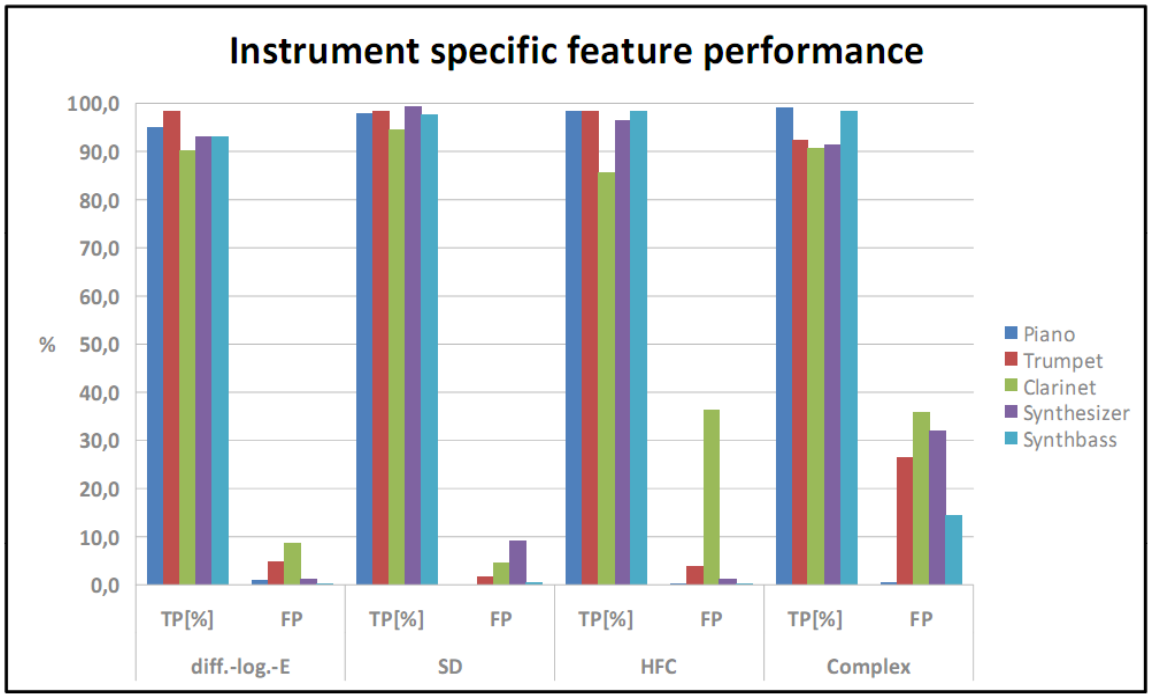

Es ist klar ersichtlich, dass eine gemeinsame Auswertung der Onsetzeitpunkte Vorteile in zweierlei Hinsicht mit sich bringt. Einerseits steigt die Anzahl der erkannten Onsets, zum anderen sinkt die Anzahl an falsch erkannten Onsets, bedingt durch die Verwendung des Zuverlässigkeitswertes für Onsets.



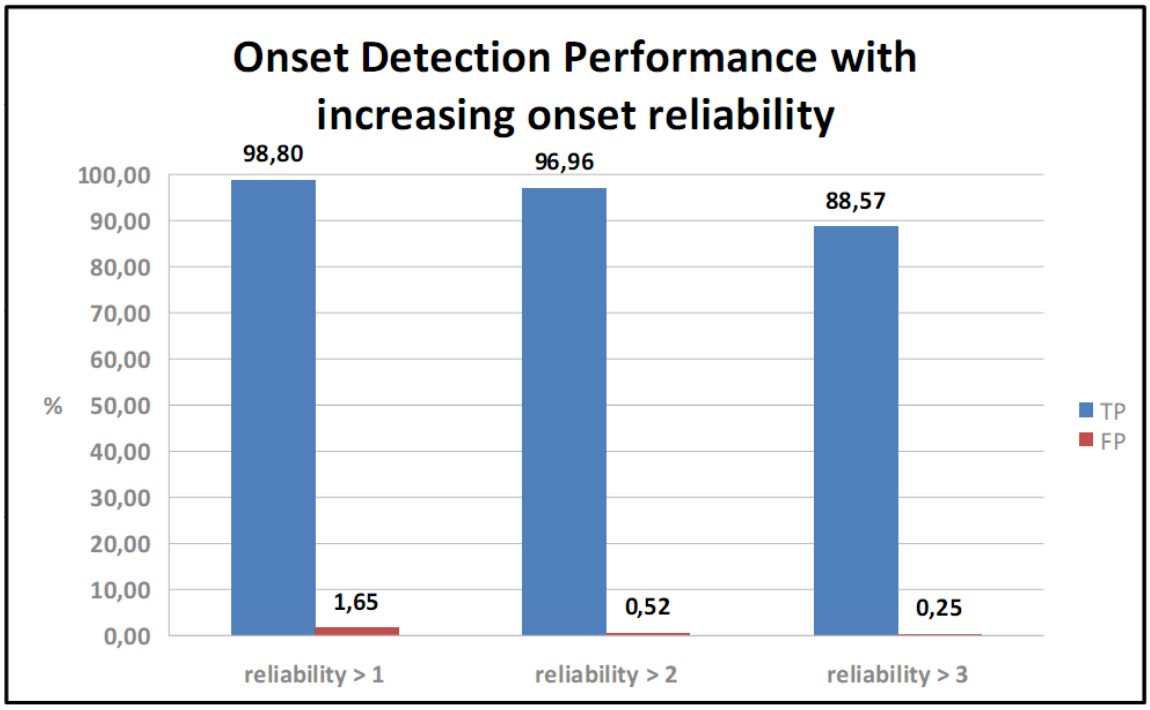

Weiters zeigt sich, dass die Genauigkeit der Onsetszeitpunkte der kombinierten Analyse im Mittel höher ist, als die der aus einzelnen Detektionsfunktionen abgeleiteten Onsets.



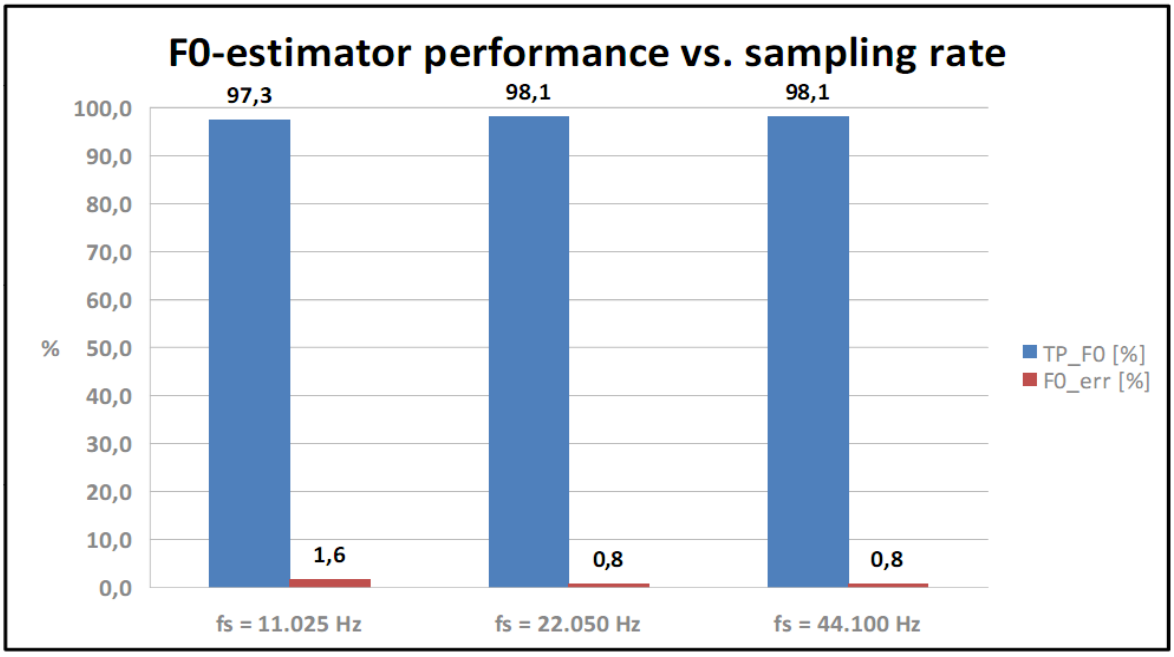

Da sich keine relevanten Performance-Unterschiede bei Einschränken der zur Berechnung der Features verwendeten Bandbreite gezeigt haben, wurde diese mit F _low=100Hz auf f_high=2000Hz beschränkt.



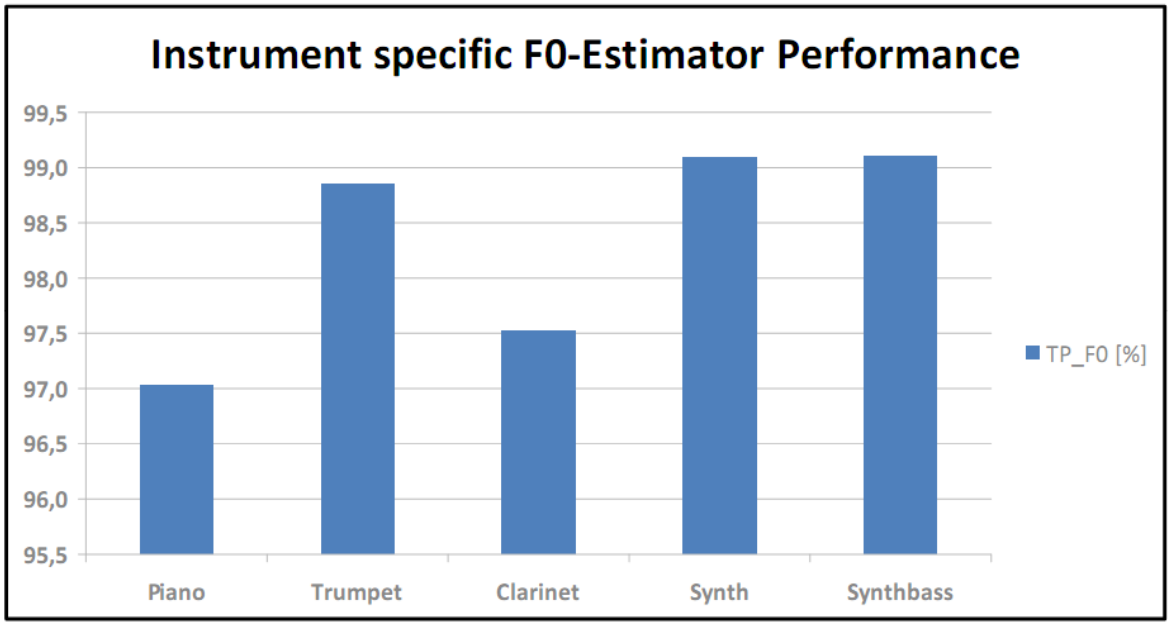

Die Performance der Features für einzelne Instrumente gibt Aufschluss darüber, welche Features sich dazu eignen was für Arten von Klängen abzubilden. Die hohe False Positive Rate des „Complex domain features“ mag sich darin begründen, dass die Kombination der Amplituden und Phaseninformation auch eine Anfälligkeit auf beide Fehlertypen bedingt.



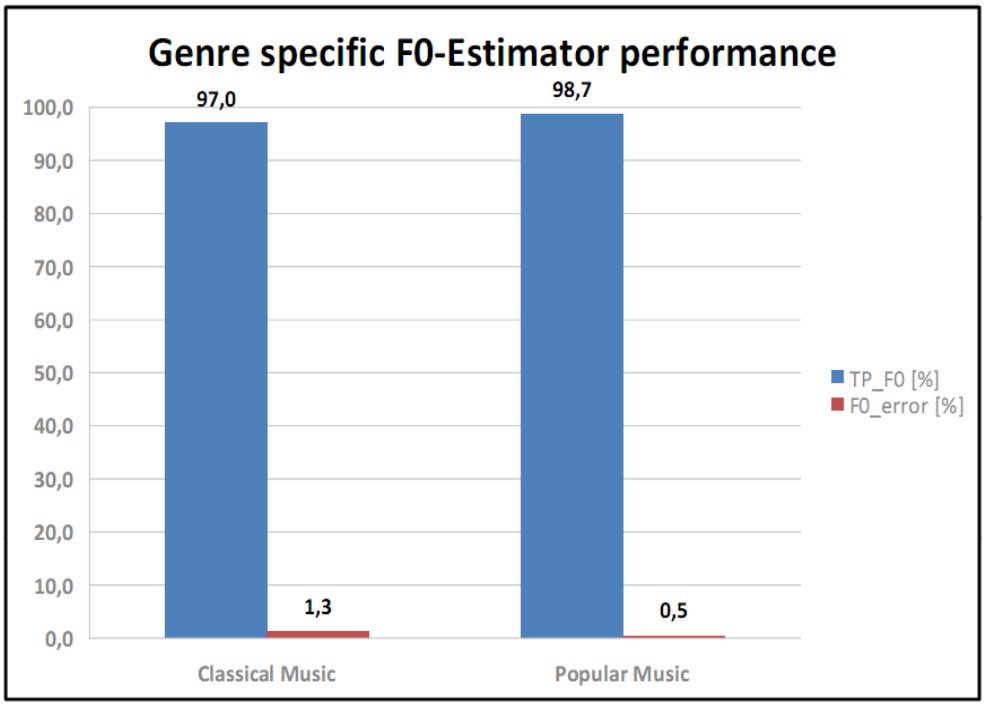

Durch Steigern des notwendigen Zuverlässigkeitswertes (reliability), die ein Onset aufweisen muss, um als richtig gewertet zu werden, kann die Anzahl der falsch erkannten Onsets weiter verringert werden. Dies hat natürlich auch eine geringere Detektionsrate für korrekte Onsets zu Folge.




Die Verschlechterung der Ergebnisse bei 11.025 kHz sampling rate für die F0 Bestimmung ist dadurch zu erklären, dass bei so geringer sampling rate die Genauigkeit der Lokalisation des Minimums durch Interpolation nicht mehr ausreicht.




Es zeigt sich, dass der F0 Estimator auf Basis des YIN Algorithmus sich sehr gut für die F0-Bestimmung von Melodien eignet.




Die etwas besseren Ergebnisse für Pop Musik lassen sich dadurch erklären, dass einerseits Pop Musik im allgemeinen weniger Noten pro Zeiteinheit bei gleichem Tempo beinhaltet wie klassische Musik, zum anderen Klassik oft sehr schnelle Notenwechsel (z.B. Triller) beinhaltet, die für das System schwieriger auf zu lösen sind.

# Referenzen


[1] Monophonic Transcription with autocorrelation
Giuliano Monti, Mark Sandler, Department of Electronic Engineering, King's College London, Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy, December 7-9, 2000

[2] A Tutorial on Onset Detection in Music Signals
Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler, IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 13, NO. 5, SEPTEMBER 2005

[3] Automatic Transcription of Music
Anssi Klapuri, Master of Science Thesis, TAMPERE UNIVERSITY OF TECHNOLOGY Department of Information Technology

[4] CUEX: An Algorithm for Automatic Extraction of Expressive Tone Parameters in Music Performance from Acoustic Signals
Anders Friberg, Erwin Schoonderwaldt, Royal Institute of Technology (KTH), Speech, Music, and Hearing, Patrik N. Juslin, Uppsala University, Sweden, Department of Psychology,

[5] Tempo and Beat Analysis of Acoustic Musical Signals
Scheirer (1996), Machine Listening, Group, MIT Media Laboratory, 1996.

[6] On the Automatic Transcription of Percussive Music –
From Acoustic Signal to High-Level Analysis
A. W. Schloss, Ph.D. dissertation, Dept. Hearing and Speech, University of Stanford, CA, 1985.

[7] An Introduction to the Psychology of Hearing
B. C. J. Moore, New York: Academic, 1997.

[8] Sound Onset Detection by applying psychoacoustic knowledge
Anssi Klapuri, Signal Processing Laboratory, Tampere University of Technology, FINLAND

[9] "Hearing" – Handbook of Perception and Cognition

[10] On the Use of Phase and Energy for Musical Onset - Detection in the Complex Domain
Juan P. Bello, Chris Duxbury, Mike Davies, and Mark Sandler, IEEE Signal Processing Letters, Vol. 11, No.6, June 2004

[11] YIN - A fundamental frequency estimator for speech and music
Alain de Cheveigne, Ircam-CNRS, Paris, France