JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

TNF

Technisch-Naturwissenschaftliche Fakultät

Automatic Drum Transcription

BACHELORARBEIT
(Projektpraktikum)

zur Erlangung des akademischen Grades

Bachelor of Science

im Bachelorstudium

INFORMATIK

Eingereicht von:
Herbert Schmöller, 0431047

Angefertigt am:
Institut für Computional Perception

Beurteilung:
Univ. Prof. Gerhard Widmer

Mitwirkung:
Dipl. Ing. Klaus Seyerlehner

Linz, November 2009

Abstract. In dieser Arbeit wird über Sinn, Zweck und Potential von automatischen Drum-Transcription-Systemen berichtet sowie deren derzeitiges und zukünftiges Einsatzgebiet beleuchtet. Es wird auf die Grundlagen perkussiver Instrumente sowie deren Besonderheiten im Vergleich zu tonalen Instrumenten näher eingegangen. Funktionalität und Ablauf verschiedener gängiger Drum-Transcription-Systeme werden im Anschluss daran untersucht und zusammengefasst. Weiters werden die genaue Funktionsweise sowie der Ablauf des im Rahmen dieser Arbeit erstellten einfachen Systems zur Schlagzeug-Erkennung präsentiert. Die erzielten Ergebnisse werden verglichen und etwaige Verbesserungsmöglichkeiten diskutiert.

1. Einführung

1.1 Einsatzgebiete von Drum-Transcription-Systemen

Die Erkennung der rhythmischen Struktur eines Songs ist neben Akkordfolge und tonalem Zusammenhang eines der wichtigsten Kriterien zur Klassifikation eines Musikstückes. Dies trifft vor allem im Rock- und Popbereich zu, wo der rhythmischen Komponente weitaus mehr Aufmerksamkeit geschenkt wird als beispielsweise in der Klassik.

Ein wichtiges Einsatzgebiet für automatische Transkriptionssysteme ist die automatische Kategorisierung von Musik. So können, die durch ein Drum-Transcription-System gewonnenen Informationen dazu verwendet werden, Musik automatisch zu organisieren oder Musikempfehlungen zu generieren. Speziell im Bereich von Online Musikportalen gewinnen Musikempfehlungssysteme immer mehr an Bedeutung. Um Empfehlungen an die Kunden weiterzugeben ist es zunächst unabdingbar gewisse Informationen des Stückes zu erfassen. Solche Informationen, wie Rhythmik und Tonalstruktur dienen unter anderem der Einteilung in musikalische

Genres und andere Aspekte, wie zum Beispiel die emotionale Stimmung des Songs, was dem Kunden die weiterführende Suche nach ähnlichem Material natürlich sehr erleichtert.

Bisher gibt es jedoch nur wenige Plattformen, die wirklich auf den musikalischen Inhalt des Stückes eingehen. Zurzeit werden von vielen Systemen eher lediglich die Bewertungen der User aufgezeichnet und auf Grund derer in weiterer Folge Empfehlungen erstellt. Nur wenige Recommender-Systems basieren tatsächlich auf dem musikalischen Inhalt der Songs, da es immer noch sehr schwierig ist, aus einem gesamten Musikstück zum einen einzelne Informationen zu isolieren und zum anderen diese im Anschluss auch entsprechend zu interpretieren.

Des Weiteren ist die Erfassung musikalischer Zusammenhänge eines Stückes für viele Musiker sehr wichtig, um daraus zu lernen, um kompositorische Aspekte des Songs zu erfassen, oder generell um einfach nur die Notation eines gewissen Instruments zu erhalten.

Zuletzt kommt der automatischen Transkription insofern eine hohe Bedeutung zu, als es viele musikalische Datenbanken mit beispielsweise klassischen Stücken verschiedenster Epochen gibt, von denen keine Partituren in digitaler Form bekannt sind. Die Notation dieser Stücke per Hand ist ein immenser Aufwand, wohingegen die automatisierte Erfassung der Partituren eine hohe Zeit- und Kostenersparnis mit sich bringt.

1.2 Tonale & Perkussive Instrumente

Betrachtet man Musikinstrumente im Allgemeinen, so kann man diese in zwei Gruppen unterteilen:

a) Die Gruppe der harmonischen Instrumente
b) Die Gruppe der Schlaginstrumente

Der Unterschied der beiden Gruppen ist jener, dass harmonische Instrumente in der Lage sind, konstante, tonale Schwingungen zu erzeugen, perkussive Instrumente

hingegen produzieren Geräusche, die keinen tonalen, jedoch einen stochastischen Inhalt aufweisen [Fletcher, 1998].

Spielt man beispielsweise einen Ton auf einem Klavier, der eine gewisse Zeit andauert, so hat dieser eine sinusförmige Grundwelle mit einer bestimmten Frequenz sowie mehrere sinusförmige Oberwellen, deren Frequenzen ganzzahlige Vielfache des Grundtons darstellen. Vernachlässigt man den Ein- und Ausschwingvorgang des Tones, so ist das Spektrum, das aus diesen Grund-und Oberwellen gebildet wird mehr oder weniger konstant. Schlägt man nun einen Akkord, also ein Tongemisch mit der Gitarre an, dessen Dauer ebenfalls eine gewisse Zeit lang konstant ist, so erhält man im Spektrum des Akkords die jeweilige Grundwelle und die entsprechenden Oberwellen eines jeden erklingenden Tons. Lediglich während des Einschwingvorganges, also während dem Anschlag oder auch Onsets des Akkordes und während des Ausschwingvorgangs kann man eine Änderung des Frequenzspektrums über der Zeit ausmachen. Nach dem der Anschlag vorbei ist, und die Töne klingen, bleibt auch hier das Spektrum weitgehend konstant.

Ganz anders bei einem Schlaginstrument wie zum Beispiel einer Snare-Drum. Hier gibt es keinen tonalen Zusammenhang, keine konstanten Töne, die erklingen, sondern lediglich einen Anschlag, den Onset, der eine sehr Rasche Änderung der Amplitude des Signals über der Zeit darstellt. Prinzipiell ist solch ein Anschlag mit einer extremen Variante des Einschwingvorganges eines tonalen Instruments zu vergleichen. Dadurch, dass kein konstanter, harmonisch-tonaler Verlauf entsteht erhält man in der Praxis lediglich einen stochastischen Anteil, der sich in Form eines eher hochfrequenten, unregelmäßigem Frequenzspektrums ausdrückt. Dieses Spektrum variiert stark mit der Zeit und es besteht im Gegensatz zum Akkord- oder Tonspektrum kein ganzzahliger Zusammenhang zwischen den Frequenzen der gleichzeitig vorkommenden Sinus-Schwingungen. Man kann zwar gewissen Schlaginstrumenten, wie Pauke oder Tom-Toms einen Grundton sowie einen tonalen Charakter zu ordnen, beim Groß der perkussiven Instrumente, vor allem in der Pop-Musik ist dies jedoch vernachlässigbar.

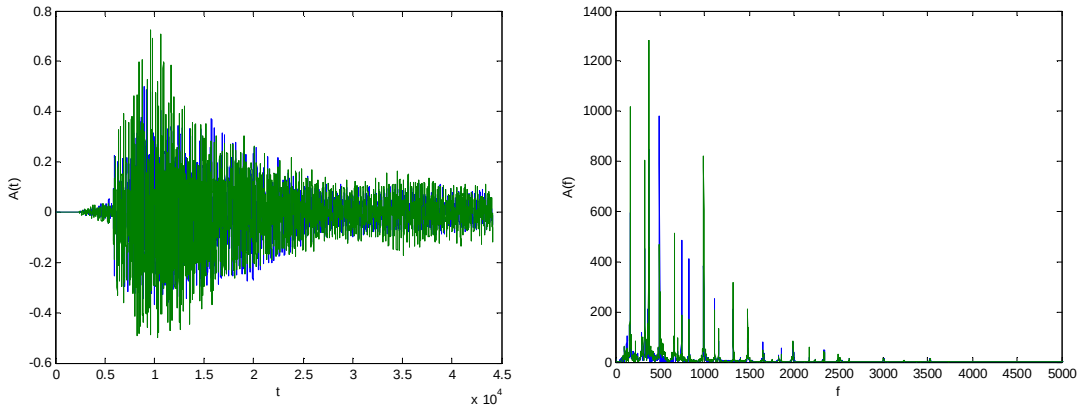

Abb. 1. Klavierakkord in Stereo, links Amplitudenverlauf über der Zeit, rechts Frequenzspektrum nach Durchführung einer FFT.

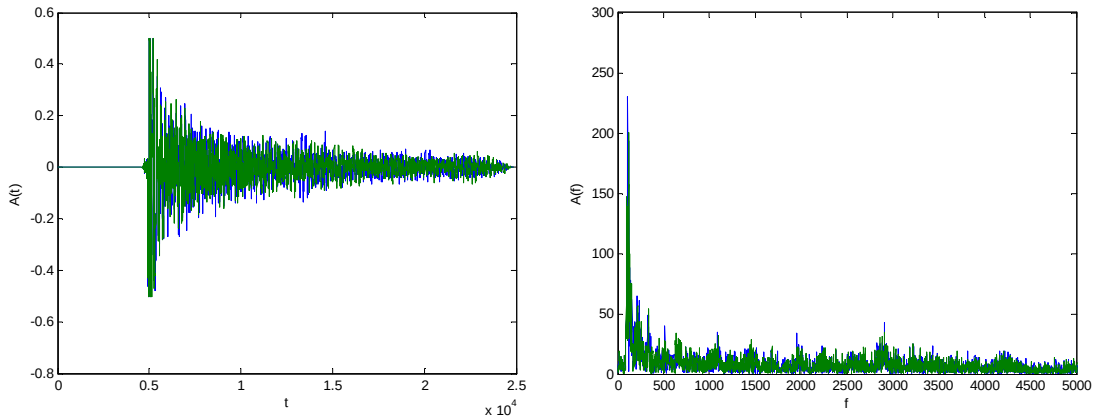

Abb. 2. Snare-Schlag in Stereo, links Amplitudenverlauf über der Zeit, rechts Frequenzspektrum nach Durchführung einer FFT.

Vergleicht man die Spektren eines Klavierakkords und eines Snare-Schlages, wie in Abbildung 1 und 2, so ist ein erheblicher Unterschied im zu erkennen. Der Klavier-Akkord weist einige wenige definierte Oberwellen auf, wohingegen der Anschlag einer Snare-Drum im Spektrum weitaus „chaotischer“ wirkt und hier deutlich die große Menge der in keinem ganzzahligen Verhältnis stehenden Obertöne erkennbar sind.

1.3 Herausforderungen musikalischer Transkription-Systeme

Betrachten wir nun kurz den heutigen Entstehungsprozess eines Songs im Pop- oder Rockbereich: Die Instrumente, sowie auch Gesangslinien und andere akustische Ereignisse werden großteils als je eine einzelne Spur aufgezeichnet. Die Einzelspuren werden dann in einer Vielzahl von Prozessschritten nachbearbeitet, verändert und „feingeschliffen“. Im Anschluss werden all diese Komponenten in einem mehr oder weniger ausgewogenen Verhältnis auf eine Stereospur zusammengemischt, was soviel bedeutet, wie dass alle Signale in einer gewissen Lautsstärke addiert werden.

Auch in der Klassik, wo meist noch weitaus mehr Einzelinstrumente in einem Stück zum Einsatz kommen, werden diese auf ähnlichem Weg auf meist eine einzige Stereospur gemischt.

Dies ist natürlich zielführend, da wir beim Abhören der Musik lediglich ein Stereosystem benötigen und unser Gehör sich trotzdem einen guten Eindruck über die einzelnen in einem Song vorkommenden musikalischen Ereignisse machen kann. Gleichzeitig ist dies auch die größte Herausforderung, die an ein Transkriptions-System gestellt wird: Die Zusammenlegung der einzelnen Ereignisse, der einzelnen Schallquellen, wenn man so will, die in Folge des Aufnahme-und Abmischprozesses durchgeführt wurde muss bis zu einem gewissen Grad rückgängig gemacht werden, um im Anschluss die musikalischen Eigenschaften eines jeden solchen Einzelevents erfassen zu können.

Bisher hat es eine Vielzahl von verschiedenen Ansätzen gegeben, dieses Problem zu überwinden. Welche genau, wird im Folgenden beschrieben. Es ist jedoch bei weitem noch nicht gelungen, die überragende Fähigkeit des menschlichen Gehörs

bzw. des menschlichen Gehirns, einzelne Instrumente, Akkorde, Töne, Intervalle, Rhythmik sowie weitere verschiedenste zum Teil subtile Details aus einem einzigen Musikstück zu erfassen, als einen automatisierten Prozess nachbilden zu können.

2. Aufbau/Struktur

Betrachtet man die nachfolgend vorgestellten Verfahren, so lässt sich der Aufbau eines Drums-Transcription-Systems grob in drei Teile gliedern:

- Signalaufbereitung
- Trennung von harmonischen und stochastischen Anteilen
- Bestimmung & Erfassung der Anschläge

2.1 Signalvorverarbeitung

Im Rahmen der Signalaufbereitung wird versucht das Eingangssignal insofern zu verändern, um den nachfolgenden weiterführenden Schritten ein optimales Ausgangsmaterial zur Verfügung stellen zu können. Beispielsweise wird das Stereo Signal in gewisser Weise auf eine einzige Monospur gemischt oder es werden gewisse Lautstärkeanpassungen durchgeführt. Ebenso kann zum Beispiel ein durchgeführtes Downsampling den Rechenaufwand verringern, eine durchgeführte Interpolation unter anderem die Genauigkeit verbessern. Verschiedene Band-Filter bzw. Transformationen werden ebenfalls im Rahmen der Signalvorverarbeitung durchgeführt.

2.2 Trennung von harmonischen und stochastischen Anteilen

In fast keinem musikalischen Bereich wird jemals ein Schlagzeug oder andere perkussive Instrumente zur Gänze alleine vorkommen. Die rhythmische Sektion einer Band hat vorwiegend begleitenden Charakter, das heißt dass sie fast immer in

Kombination mit anderen Instrumenten sowie Gesang auftritt. Um jedoch perkussive Ereignisse klassifizieren zu können, ist es von Vorteil diese isoliert zu betrachten. Da auch harmonische Instrumente wie zum Beispiel die Gitarre bei Akkordwechseln einen Anschlag produzieren und dieser eben nicht in das Ergebnis eines Drum-Transcription-Systems mit einfließen soll ist eine Trennung zusätzlich von Vorteil. Solch eine Isolation auf verschiedene Ereignisquellen kann mit völlig unterschiedlichen Verfahren durchgeführt werden. Einige dieser Verfahren werden in weiterer Folge vorgestellt. Es gibt auch Systeme, die diese Trennung von harmonischen und stochastischen Anteilen nicht durchführen und daher eher im Umfeld von Solo-Drums Einsatz finden.

2.3 Bestimmung & Erfassung der Anschläge

Um die eigentlichen Anschläge zu detektieren gibt es ebenfalls unterschiedliche Verfahren. Gängig ist die Erkennung/Klassifikation mit Hilfe von Machine Learning Tools, die an Trainingssets trainiert und im Anschluss an Testsets evaluiert werden. Zuvor können noch weitere Schritte durchgeführt werden, wie zum Beispiel die Energie-Erfassung verschiedener Spektralbänder während des Anschlags oder der prinzipiellen Einsatz eines Noise-Gates während vermuteter Anschläge.

3. Drum Transcription Systeme – Ein Überblick

3.1 Drum Transcription nach Gillet und Richard

In [Richard, 2005 - 1] präsentieren Gillet und Richard einen Ansatz, der die Drums aus einem polyphonen musikalischen Umfeld erfasst. Nach Anwendung einer Filterbank wird mit Hilfe der so genannten Noise-Subspace-Projection versucht die harmonischen von den stochastischen Anteilen zu trennen. Im Anschluss daran

werden die Onset-Kandidaten weiter bearbeitet, um eine Klassifikation mit Hilfe eines Support-Vector-Machine-Klassifizierers durchzuführen. Vorteil des Systems ist lt. Gillet und Richard, das beim ganzen Prozess keine Phaseninformation des Eingangssignals verloren geht, weshalb man am Schluss die erkannten Anschläge problemlos zum Eingangssignal addieren oder auch subtrahieren kann.

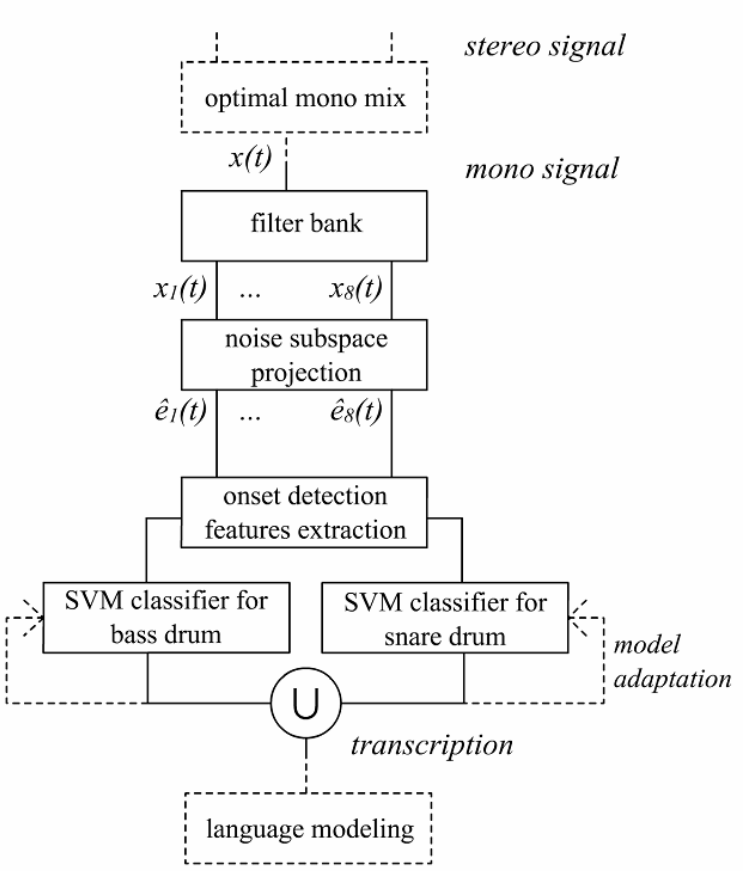

Abb. 3. Ablauf des Drum-Transcription-Systems aus [Richard, 2005 - 1]

3.1.1 Pre-Processing

Handelt es sich beim Eingangssignal um ein Stereosignal, so werden die beiden Kanäle zu einem einzigen zusammengefasst, wobei eine Maximierung nach einem Impuls-Kriterium stattfindet. Da in Musikstücken oftmals Mono aufgenommen wird und das Instrument dann in beiden Kanälen, je einen unterschiedlichen Verstärkungs- oder Gain Wert aufweist, kann dies benutzt werden, um die empfundene Impulsivität des Ereignisses zu erhöhen, indem man den linken und rechten Verstärkungswert des Instrumentes geeignet wählt und die beiden Kanäle dann auf einen Mono-Kanal zusammenmischt. In [Richard, 2005 - 2] wird das erwähnte Impulskriterium wie folgt beschrieben:

Das Verhältnis der beiden Gain-Werte von rechtem und linkem Kanal wird als

$$\beta = \frac{\gamma_2}{\gamma_1} \qquad (1)$$

bezeichnet. Es gilt

$$s_\beta(t) = I(L(t) + \beta R(t)) \qquad (2)$$

zu maximieren. Es wird ein Envelope-Signal *s'(t)* gebildet, indem man *s(t)* halb-gleichrichtet, „dezimiert", tiefpass-filtert sowie differenziert. Danach wird für diesen „Envelope" ein Kontrast-Faktor, wie folgt gebildet:

$$I(s) = \frac{\Sigma_{t=1}^{T} s'(t)}{T\sqrt[T]{\Pi_{t=1}^{T} s'(t)}} \qquad (3)$$

Als Ausgangswert des Pre-Processing-Vorgangs wird schließlich *x(t)* als

$$x(t) = s_{\beta^*}(t) \qquad (4)$$

angenommen, wobei

$$\beta^* = \arg\max I(s_\beta(t)) \qquad (5)$$

gewählt wird.

Der Wert β^* kann auch ermittelt werden, in dem man den Gegenwert zur Kurtosis von *s(t)* als Impulsivitätsmaß benutzt. Beide Ansätze liefern lt. Richard und Gillet vergleichbare Ergebnisse.

3.1.2 Filter-Bank

Im Anschluss wird das Eingangssignal in 8 je eine Oktav umfassende Bänder aufgeteilt. Grund dafür ist, dass die nachfolgende Noise-Subspace-Projection besser in schmalen Bändern funktioniert, wo der Geräuschanteil als Weißes Rauschen angenommen werden kann. Zusätzlich werden die einzelnen Bänder entsprechend ihrer Grenzfrequenzen downgesampled, um den Rechenaufwand zu reduzieren. Als Filter werden einfache FIR-Filter der Ordnung 100 benutzt, was den Geschwindigkeitsansprüchen des Systems ebenfalls entgegenkommt. Folgende Vorgangsweise wurde gewählt:

-Das Spektrum wird rekursiv je an der Mittenfrequenz geteilt.
-Die untere Hälfte wird um den Faktor 2 downgesamplet
-Das ganze wird für die downgesampelte Hälfte wiederholt

Diese Schritte werden so oft durchgeführt, bis man die 8 Oktavbänder erhält. Da die Abtastfrequenz des Eingangssignal mit 44100Hz gewählt wurde, ergibt sich folgende Frequenzaufteilung für die 8 Bänder: (Angaben in Hz)

Band 1: [0, 172]	Band 5: [1378, 2756]
Band 2: [172, 345]	Band 6: [2756, 5512]
Band 3: [345, 689]	Band 7: [5512, 11025]
Band 4: [689, 1378]	Band 8: [11025, 22050]

3.1.3 Noise Subspace Projection

Hier handelt es sich um eine bandweise Dekomposition der harmonischen bzw. stochastischen Anteile, um im Anschluss die Geräuschanteile eines jeden Bands zu erhalten.

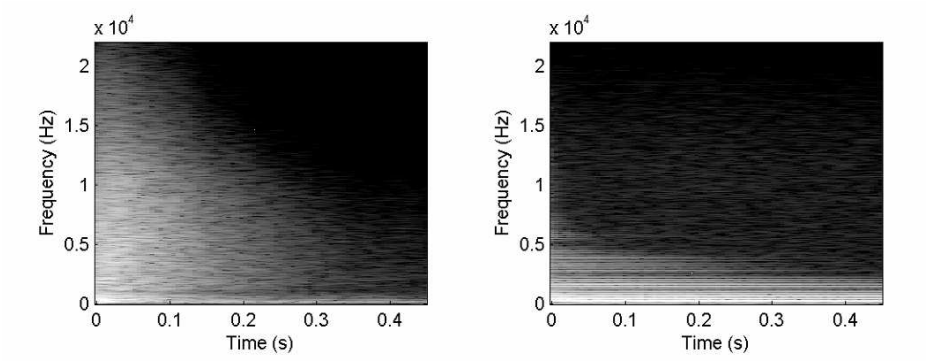

Abb. 4. Spektrogramm eines Snare-Schlags (links) und eines Gitarren-Tones (rechts) [Richard, 2005 - 1]

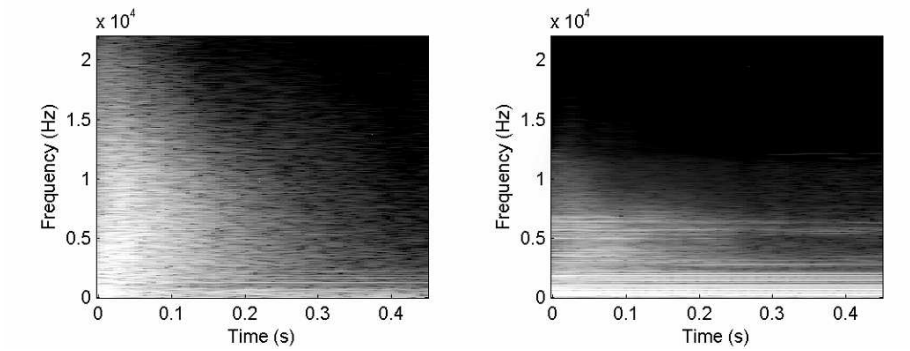

Abb. 5. Stochastische Anteil einer Snare+Gitarren Mischung (links), harmonischer Teil der selben Mischung (rechts) [Richard, 2005 - 1]

Wie aus Abbildung 4 und 5 ersichtlich, entspricht der Snare-Schlag im Spektrogramm weitgehend dem stochastischen Geräuschanteil des Mix-Signales, dem Gitarrenton hingegen kann man weitgehend die harmonischen Komponenten des Spektrogramms zuordnen.

Das Verfahren der Noise-Subspace-Projection beruht im Prinzip auf dem in [Badeau, 2002] vorgestelltem EDS-Modell von Badeau et al. Dieses Modell beruht darauf, dass der harmonische Anteil des Signals als Summe exponential gedämpfter Sinusschwingungen dargestellt werden kann. Subtrahiert man diesen so erhaltenen Anteil dann vom Ausgangssignal, so erhält man den übrig gebliebenen Geräuschanteil, der die Drums repräsentiert.

Es ist zwar möglich zu diesem Zwecke eine Fourier-Analyse wie zum Beispiel die STFT (Short Time Fourier Transformation) zu benutzten, da diese jedoch eine verhältnismäßig geringe Auflösung bietet werden hier Subspace-basierende Methoden verwendet, die im Gegenzug eine höhere Auflösung zu Stande bringen.

Ein Fenster der Länge *L*, welches den Signalvektor *x* definiert wird aus dem ursprünglichen Signal extrahiert. Dieser L-dimensionale Vektor *x* wird aufgeteilt in einen $p = 2*n$ – dimensionalen Raum, der das Signal repräsentiert sowie einen $L-p$ – dimensionalen Raum, der das stochastische Signal darstellt. Die Zahl *n* entspricht hier der Anzahl der exponentiell gedämpften Sinus-Schwingungen. Wenn man nun *x* auf das stochastische Subband projiziert, so erhält man den Geräusch-Vektor des

dazugehörigen Signals *x*. Das ganze Signal kann mit Hilfe einer Overlap-Add-Methode bearbeitet werden.

Das Tracking des eigentlichen Subspace wird mit Hilfe eines klassischen, iterativen EVD-Algorithmus erzielt. Gillet und Richard referenzieren in [Richard, 2005 - 1] zwar auf [Badeau, 2002] - eine Publikation, die sich mit diesem Algorithmus befasst - die genau Vorgehensweise, die in diesem konkreten Fall Anwendung gefunden hat bleibt jedoch im Verborgenen.

In [Badeau, 2002] beschreiben Badeau und Kollegen zwar diesen EVD-Algorithmus auf mathematisch sehr komplexe Weise, jedoch konnte kein Zusammenhang zu [Richard, 2005 - 1] hergestellt werden.

Im Rahmen des EVD-Algorithmus werden 46ms lange Fenster benutzt und die Überlappung der Fenster entspricht einem Faktor von 3/4.

Die Anzahl der benutzten Sinus-Schwingungen pro Band *x(t)* wurde per Hand ausgewählt: 2 Sinusse werden für das unterste Band $x_1(t)$ benutzt, in welchem sich nur Bass-Informationen befinden, je 5 Sinusse für $x_2(t) – x_4(t)$ und je 8 Sinusse für die restlichen Bänder. Falls in diesem Schritt zu wenige Sinus-Schwingungen für ein bestimmtes Band verwendet werden, so besteht die Möglichkeit, dass harmonische Komponenten im Ausgangssignal erhalten bleiben. Nimmt man jedoch zu viele Sinusse für ein Band, so geht leicht Information über das Timbre des Eingangssignals verloren. Es wäre auch möglich, die Anzahl der Sinus-Schwingungen pro Band automatisiert zu errechnen. Dies wurde jedoch auf Grund von zu hoher Computationskosten nicht praktiziert.

Nach der Noise-Subspace-Projection erhält man somit 8 Subbandsignale $e_k(t)$. Aufgrund des unterschiedlichen Downsamplings in einzelnen Bändern, müssen diese zum Schluss zeitlich resynchronisiert und upgesampled werden, in dem man einen Synthese-Filter anwendet.

3.1.4 Onset Detection

Nach der Noise-Subspace-Projection befinden sich auch Anschläge von harmonischen Instrumenten in den Subbändern. Diese werden vorerst belassen und erst im letzten Schritt des Verfahrens, dem Machine-Learning behandelt.

Nun werden die Subbänder weiter bearbeitet. Jedes dieser Subbänder wird halb-gleichgerichtet, womit man nur mehr die positiven Amplitudenwerte erhält. Danach wird ein Tiefpassfilter über jedes der Bänder angewandt. Die daraus erhaltenen 8 Bänder werden $b_k(t)$ genannt. Obwohl zur Onset-Detection oftmals nachfolgende Formel

$$\frac{d}{dt}\log(b_k(t)+A) \qquad (6)$$

benutzt wird, sind Gillet und Richard zu der Erkenntnis gekommen, dass die einfache Ableitung eine höhere Genauigkeit liefert. Somit werden die Spitzen des Signals über

$$\frac{d}{dt}b_k(t) \qquad (7)$$

gefunden.

3.1.5 Feature Extraction

Für jeden Anschlag zur Zeit t werden nachfolgende Features über ein 100ms Zeitfenster errechnet: Die Energie der ersten 6 Sub-Bänder wird gebildet sowie der Durchschnitt der ersten 12 MFCC-Koeffizienten (ohne *c0*) wird über nachfolgende Frames gebildet. Die MFCC werden über dem Rauschsignal

$$\sum_k \hat{e}_k(t) \qquad (8)$$

gebildet.

Es wurden verschiedene Transformationen des Feature-Sets getestet. Die Durchführung einer Principal Component Analysis, kurz PCA konnte die Performance zwar nicht wirklich verbessern, jedoch wurde ersichtlich, dass die ersten 12 Komponenten 96% der totalen Varianz abdecken. Führt man die Klassifikation nun an den ersten 12 Komponenten durch, so reduziert man die Berrechnungkosten um ein Vielfaches, ohne jedoch gravierende Einbußen in der Genauigkeit des Systems in Kauf nehmen zu müssen.

3.1.6 SVM – Klassifikation

Wie vorhin bereits erwähnt, müssen in diesem Schritt auch noch die Anschläge der harmonischen Instrumente mit einbezogen werden bzw. ausgeschieden werden. 2 getrennte Klassifizierer werden verwendet, einer für die Bassdrum-Erkennung (BD, bzw. Non-BD) sowie einer für die Snaredrum-Erkennung (SN, bzw. Non-SN). Spricht keiner der beiden an, so erhält man eine dritte Kategorie, nämlich weder Bassdrum, noch Snaredrum.

Als Klassifizierer selbst werden Support-Vector-Machines benutzt, da diese sehr gut für binäre Klassifizierungs-Aufgaben geeignet sind. Zur Implementierung wird SVM-light verwendet.

Der Ausgangswert der SVM *f(x)* wird nun auf das Intervall] 0, 1[gemapped, was mit folgender Funktion *p(x)* geschieht:

$$p(x) = \frac{1}{1 + e^{Af(x)+B}} \qquad (9)$$

A , B hier werden die beiden Werte mit höchster Wahrscheinlichkeit angenommen, die aus einem Subset der Trainingsdaten ermittelt werden.

Nachdem das allgemeine SVM Modell an den Testdaten trainiert wurde, wird den erkannten Instanzen N ein Rang zugeordnet, indem ein Wahrscheinlichkeitsmaß verwendet wird. In diesem Fall werden als solches Maß die Wahrscheinlichkeitswerte des SVM-Klassifizierers benutzt. Ein Subset mit *k*N* Beispielen, von welchem die besten Erkennungsraten erzielt werden wird ausgewählt. Ein adaptiertes Modell wird nachträglich an diesem kleineren Testset trainiert, um im Anschluss die gesamte Klassifikation mit diesem neu erstellten Modell durchzuführen. Es hat sich gezeigt, dass die besten Ergebnisse mit *k = 0,4* erzielt werden. Das heisst, 40% der erkannten Instanzen werden zum Neu-Trainieren des Systems benutzt.

Um zu guter letzt ein Event zur Zeit *t* zu klassifizieren, wird die Zeitspanne von *t-M* bis *t+M* zusammengefasst. Der Wert *M* wird von Testfile zu Testfile per Hand variiert.

3.1.7 Richard's und Gillet's Alternative zum SVM-Klassifizierer

In [Richard, 2005 - 2] präsentieren Richard und Gillet ihre Drum-Transcription-Methode in etwas abgeänderter Form. Anstatt nach der Noise-Subspace-Projection einem Machine-Learning-Ansatz nachzugehen, wird ein alternativer Weg gewählt, bei dem kein Vorab-Wissen in Form vom Trainings-Daten zur Verfügung stehen muss.

In [Richard, 2005 - 2] werden die nach der Noise-Subspace-Projection erhaltenen Bänder $e_k(t)$ genannt. 3 Drums-Kategorien Bass-Drum, Snare-Drum sowie Cymbals werden definiert, wobei jeder dieser Kategorien ein Index *i(0..2)* zugeordnet wird. Für jedes Subband wird ein Detection-Signal $d_i(t)$ erzeugt.

$$d_i(t) = \sum_k a_{ki}\, \hat{e}_k(t) \qquad (10)$$

a_{ki} wird so gewählt, dass nur charakteristische Frequenzbänder für die jeweilige Drum-Kategorie verwendet werden. Für Bass-Drum werden die ersten beiden Bänder, für Snare-Drum die nächsten beiden Bänder sowie für Cymbals das letzte Band verwendet.

Die Signale $d_i(t)$ wird nun downgesampled, halb-gleichgerichtet und in $d_i'(t)$ umbenannt. Überschreitet $d_i'(t)$ eine gewisse Threshold

$$d'_i(t) > 2\sigma_{d'_i(t)} \qquad (11)$$

so wird ein Anschlag erkannt. σ stellt in diesem Fall die Standard-Abweichung da. Die Länge eines Anschlags wird ebenfalls aus $d_i(t)$ ermittelt, um daraus eine Maskenfunktion $\alpha_i(t)$ zu erstellen, die während der Anschlagdauer den Wert 1 aufweist, ansonsten jedoch auf 0 gesetzt wird. Die Maske verhält sich prinzipiell wie ein Noise-Gate, das ab einer gewissen Threshold eine gewisse Zeit lang "öffnet".

Jedem Noise-Subband wird nun ein Modulationssignal

$$\omega_k(t) = \max_i(r_{ki}\alpha_i(t)) \qquad (12)$$

r_{ki} wird so gewählt, dass nur entsprechend relevanten Frequenzbänder der einzelnen Drum-Klassen benutzt werden:

r_{k0} für Bass-Drum Klasse = [1,1,1,1,1,1,0,0], d.h. die ersten 6 Bänder

r_{k1} für Snare-Drum Klasse = [0,1,1,1,1,1,1,1], also alle Bänder ausser das Erste

r_{k2} für Cymbals-Klasse = [0,0,0,0,1,1,1,1], die oberen Bänder

Die Koeffizienten r_{ki} sind jedoch nicht mit den vorhin erwähnten a_{ki} gleichzusetzen. Während a_{ki} lediglich für die Onset-Detection typische Frequenzbänder der einzelnen Drum-Klassen berücksichtigt, verwendet man bei r_{ki} weitere Bänder, um die Qualität der Modulation bzw. der Synthese zu verbessern.
Man erhält also zuletzt das modulierte bzw. synthetisierte Drumsignal als

$$drums(t) = \sum_k \omega_k(t)\hat{e}_k(t) \qquad (13)$$

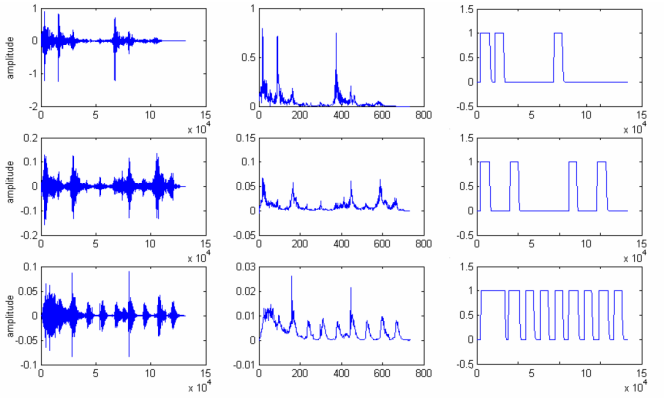

Abb. 6. Von links: $d_i(t)$, $d'_i(t)$, sowie Modulationssignal $\alpha_i(t)$ für die 3 Drums-Klassen Bass-Drum (oben), Snare-Drum (mitte), Cymbals (unten).

3.1.8 Ergebnisse/Evaluation

Beide von Richard und Gillet präsentierte Ansätze wurden auf unterschiedlichem Weg evaluiert. Bei der zuletzt vorgestellten Alternative zur SVM-Klassifikation wurden gängige Populärsongs analysiert und die erkannten, modulierten, reinen Drum-Spuren einmal zum ursprünglichen Signal aufaddiert und einmal vom ursprünglichen Signal subtrahiert. Danach wurden die so erhaltenen Samples von Testpersonen mit den ursprünglichen Songs verglichen, und es wurde die subjektiv wahrgenommene Qualität bewertet. Es stellte sich heraus, dass in vielen Fällen, die aufaddierten Spuren gleichwertig zt. sogar qualitativer klangen, als die Originale, die subtrahierten Signale jedoch eher als mangelhafter eingestuft wurden.

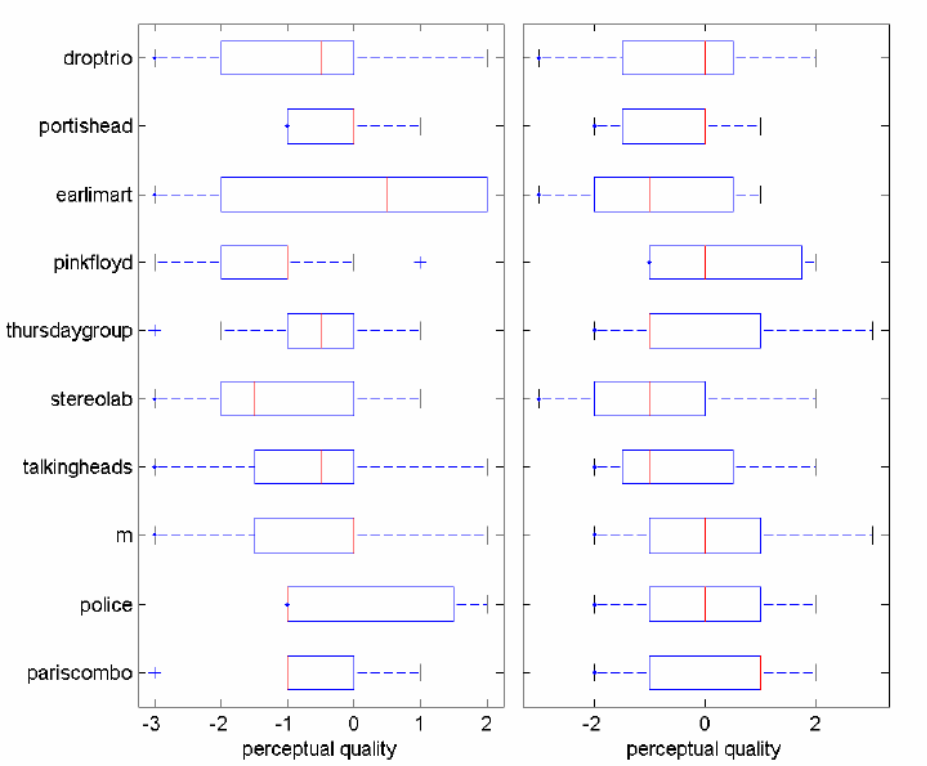

Abb. 7. aus [Richard, 2005 - 2] : Subjektiver Vergleich der subtrahierten Signale (links) sowie der aufaddierten Signale (rechts), durchgeführt an 12 Testpersonen, wobei der Median-Wert jeweils rot markiert ist.

Die Performance des SVM-Ansatzes wurde etwas genauer untersucht. Als Ausgangsbasis diente eine eigens erstellte Sample-Datenbank, die von zwei verschiedenen Drummern eingespielte Test-Stücke aus verschiedenen musikalischen Bereichen enthielt. Snare und Bassdrum-Spuren wurden zuvor extra analysiert um die genauen Anschlagswerte zu erhalten und somit Vergleiche zu den erkannten Anschlägen aus den Gesamtmixes (Drums + Begleitung) zu ermöglichen. Es wurden 2 Evaluationsmaße gebildet und zwar

$$\text{precision} = \frac{N_c}{N_d} \qquad (14)$$

$$\text{recall} = \frac{N_c}{N} \qquad (15)$$

wobei N_d die gesamte Anzahl der erkannten Events darstellt, N_c die Anzahl der korrekt erkannten Events, sowie N die eigentliche Anzahl der zu erkennenden Events. Zusätzliche wurde eine F-Measure als Genauigkeitsangabe des Systems eingeführt, welche sich wiefolgt darstellt:

$$F - measure = \frac{2 * precision * recall}{precistion + recall} \qquad (16)$$

Ein 50ms Toleranzbereich wurde bei der Evaluation ebenfalls angewandt, was soviel heißt, wie das Anschläge als korrekt eingestuft wurden, wenn sie innerhalb von 50ms am jeweiligen Referenzanschlag lagen.

Sequence	Drummer	Bass drum			Snare drum		
		Rec.	Prec.	F-meas.	Rec.	Prec.	F-meas.
Blues	1	72.1%	96.1%	0.82	95.6%	100.0%	0.98
	2	86.1%	82.7%	0.84	87.8%	87.8%	0.88
Blues rock	1	92.2%	92.2%	0.92	100.0%	100.0%	1.00
	2	89.5%	91.7%	0.91	82.2%	80.4%	0.81
Celtic	1	75.4%	94.9%	0.84	77.8%	33.3%	0.47
	2	70.1%	87.8%	0.78	80.3%	68.1%	0.74
Funk	1	79.8%	62.5%	0.70	87.8%	97.0%	0.92
	2	77.6%	90.6%	0.84	81.5%	91.7%	0.86
Jazz funk	1	94.7%	87.3%	0.91	97.9%	85.2%	0.91
	2	78.6%	94.2%	0.86	84.6%	76.7%	0.80
Groove 5/4	1	85.2%	54.1%	0.66	82.8%	96.0%	0.89
	2	91.9%	77.5%	0.84	86.3%	62.9%	0.72
Metal	1	90.3%	77.8%	0.84	83.3%	88.7%	0.86
	2	75.5%	72.1%	0.74	77.9%	75.9%	0.77
Rock	1	90.5%	77.9%	0.84	88.5%	97.7%	0.93
	2	74.1%	88.6%	0.81	88.0%	89.0%	0.88
Shuffle	1	74.4%	81.5%	0.78	85.7%	85.7%	0.86
	2	67.6%	93.1%	0.78	68.9%	81.6%	0.75
Twist	1	97.6%	75.0%	0.85	91.9%	95.8%	0.94
	2	84.8%	98.1%	0.91	78.9%	98.1%	0.87

Abb. 8. Evaluationsergebnisse der SVM-Methode aus [Richard, 2005 - 1]

Abb. 8 zeigt die Ergebnisse des Algorithmus, wobei jeweils an Drummer 1 trainiert und and Drummer 2 evaluiert, oder aber umgekehrt vorgegangen wurde. Es zeigt sich, dass das an Drummer 2 trainierte System bessere Ergebnisse erzielt, da Drummer 2 einen dynamischeren Stil aufweist, was sich zum Beispiel als häufiges Auftreten von Ghostnotes äußert.

3.2 Template Matching nach Yoshii und Kollegen

In [Goto, 2005] und [Goto, 2004] stellen Yoshii und Kollegen ein System vor, das auf einer völlig anderen als der zuvor dargestellten Methode von Richard und Gillet aufbaut. Sie versuchen Drum-Anschläge mit Hilfe von Muster-Erkennungen in der Power-Frequency-Domain des Musikstückes zu klassifizieren. Dazu sind keine große Trainingsdatenbank und auch kein spezielles Vorwissen über die Musikstücke vonnöten. Es werden einzig je ein Seed-Template für Bass-Drum, Snare-Drum (in [Goto, 2004] sowie Cymbals (als Zusatz-Klasse in [Goto, 2005]) benötigt, von dem aus iterativ weitere Muster-Variationen erzeugt werden. Ausgehend von diesen

Templates wird dann pro potentiellem Anschlag ein Distanzmaß errechnet, welches als Basis für die Klassifikation des jeweiligen Onset-Kandidaten dient.

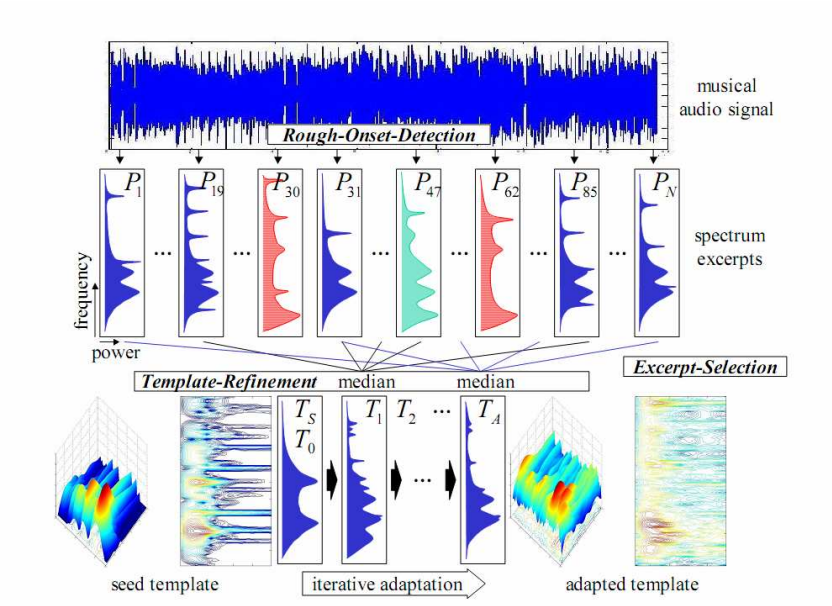

Abb. 9. Überblick der Template-Adaption-Methode aus [Goto, 2004]

3.2.1 Rough Onset-Detection

Um die Rechenkosten der beiden nachfolgenden Bereiche zu reduzieren, wird zu beginn eine grobe Onset-Detection durchgeführt. Die weiter Analyse muss danach nicht mehr von jedem Frame aus durchgeführt werden, sondern kann bei den grob errechneten Onset-Kandidaten beginnen, was den Berrechnungsaufwand natürlich um einiges verringert. Das Eingangssignal wird mit 44.1 khz gesampled und im Anschluss in Zeitframes zu je 441 Samples eingeteilt. Nun wird für jedes dieser Frames die Leistung, notiert als *P(t, f)*, also Power pro Zeit-und Frequenzeinheit

errechnet. *Q(t, f)* stellt die entsprechende zeitliche Änderung, also das Differential von *P(t,f)* dar.

Für jedes Frame wird *P(t,f)* berechnet, indem man eine 4096-Punkte STFT mit Hanning-Fenster anwendet. Falls folgendes Kriterium für 3 aufeinander folgende Frames gilt:

$$\frac{\partial P(t, f)}{\partial t} > 0 \qquad (17)$$

Gilt obiger Zusammenhang also für *t=a-1*, *t=a*, sowie *t=a+1*, wobei *a* die Frame-Nummer darstellt, so wird *Q(a,f)* wiefolgt errechnet:

$$Q(a, f) = \frac{\partial P(t, f)}{\partial t}, t = a \qquad (18)$$

Ansonsten wird *Q(a,f) = 0* gesetzt.

Nun wird für jedes Frame *t*, *Q(t,f)* als gewichtete Summe zu *S(t)* aufaddiert:

$$S(t) = \sum_{f=1}^{2048} F(f) Q(t, f) \qquad (19)$$

F(f) ist die in Abb. 10 dargestellte Tief/Hochpass-Funktion, die der Frequenz-Charakteristik der Drum-Klassen entspricht.

Jede der „groben“ Onset-Times erhält man, indem man nach Spitzen im Signal *S(t)* Ausschau hält.

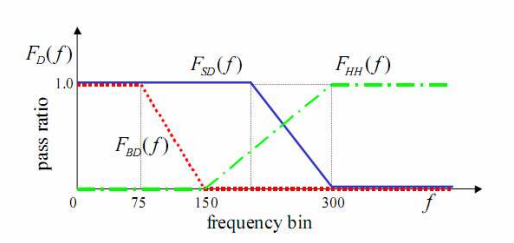

Abb. 10. Tief-bzw. Hochpassfunktionen *F(f)*, jeweils charakteristisch für die einzelnen Drum-Klassen (Bass-Drum, Snare-Drum, HiHat (Cymbals)) aus [Goto, 2005]

3.2.2 Seed-Template(s)

Für jede Drum Klasse wird nun per Hand aus dem Eingangssignal eine Muster-Vorlage, ein sogenanntes Seed-Template *Ts* ermittelt. Ein Exzerpt des Musikstückes wird nun genauer untersucht, um aus den Seed-Templates iterativ angepasste Templates für die einzelnen Drum-Klassen zu erhalten. Dabei wird im Prinzip derselbe Vorgang, wie während der Rough-Onset-Detection durchgeführt. Ab der Onset-Time wird *Ts* vom STFT Power-Spektrum extrahiert. *Ts* liegt dann als Zeit-Frequenz-Matrix *Ts(t,f)* im Bereich
(1<= t <= 15 [frames], 1<= f, <= 2048 [freq-bins]) vor. Nachdem die Templates iterativ verändert werden, werden diese, nach dem *g*-ten Durchlauf des Iterations-Algorithmus *Tg* genannt, wobei dem ursprünglichen Template der Index 0 zugewiesen wird.

Gleichzeitig wird nun aus dem oben genannten Exzerpt das Spektrum *Pi* gebildet. Dies wird für jede Onset-Time *Oi (i = 1, …, N)* [ms] durchgeführt. *Pi* ist ebenfalls als Zeit-Frequenz-Matrix dargestellt und hat die gleiche größe wie *Tg*. *Tg* und *Pi* werden nun mit den Tiefpassfunktionen gewichtet:

$$T'g(t, f) = F(f)Tg(t, f) \qquad (20)$$

$$P'i(t, f) = F(f)Pi(t, f) \qquad (21)$$

Da die Auflösung der Frames lediglich 441 Samples, bzw. 10 ms entspricht und dies nicht genug ist um hochqualitative Vorlagen zu erhalten wird jede Onset-Zeit um +5ms respektive -5ms erweitert um ein eventuell besseres Template zu erhalten. Ein Korrelationsmaß zwischen *Tg* und dem Exzerpt *Pi,j* wird gebildet, wobei *j=-5, 0, 5*.

$$Corr(j) = \sum_{t=1}^{15} \sum_{f=1}^{2048} T'g(t, f)P'i, j(t, f) \qquad (22)$$

Pi,j wird mit der Tiefpass-Funktion gewichtet und wird zu *P'i,j = F(f) Pi,j(t,f)*. Der beste Index *J* wird beim Maximum des Korrelationsmaß bestimmt, also

$$J = argmax(Corr(j)) \qquad (23)$$

Pi wird nun als $P_{i,J}$ bezeichnet.

3.2.3 Exzerpt-Selektionen

Um nun ein Set aus Spektral-Exzerpten zu erhalten, die ähnlich dem Template *Tg* sind, wird ein verbessertes logarithmisches Distanzmaß verwendet. Eine „gewöhnliche“ logarithmische Distanzmessung würde nicht zielführend sein, da eine solche Distanz zu empfindlich auf Unterschiede spektraler Spitzen reagieren würde. Für die verbesserte Distanz werden zwei verschiedene Werte eingeführt. Zum einen Distanz *Di* für den ersten Iterationsschritt *(g=0)*, zum anderen Distanz *Di* für alle

weiteren Iterationen *(g >= 1).* Auch wenn gewisse Frequenzanteile während eines Stückes variieren, kann man mit diesem Distanzmaß immer noch robuste Ergebnisse erzielen.

Tg und *Pi* werden vorher „nieder-quantisiert“ auf eine Zeit-Frequenz-Auflösung von 2 Frames (20ms) sowie 5 Frequency-Bins (54Hz Bandbreite). Die Distanzwerte *Di* zwischen *Tg(Ts)* und *Pi* lassen sich wie folgt darstellen:

$$Di = \sqrt{\sum_{\hat{t}=1}^{15/2} \sum_{\hat{f}=1}^{2048/5} (\hat{T}g(\hat{t}, \hat{f}) - \hat{P}i(\hat{t}, \hat{f}))^2}, (g = 0) \qquad (24)$$

wobei $\hat{T}g(\hat{t}, \hat{f})$ und $\hat{P}i\ (\hat{t}, \hat{f})$ folgend definiert sind:

$$\hat{T}g(\hat{t}, \hat{f}) = \sum_{t=2\hat{t}-1}^{2\hat{t}} \sum_{f=5\hat{f}-4}^{5\hat{f}} T'g(t, f) \qquad (25)$$

$$\hat{P}i(\hat{t}, \hat{f}) = \sum_{t=2\hat{t}-1}^{2\hat{t}} \sum_{f=5\hat{f}-4}^{5\hat{f}} P'i(t, f) \qquad (26)$$

Di für alle weiteren Iteration nach der Ersten sieht folgendermaßen aus:

$$Di = \sqrt{\sum_{t=1}^{15}\sum_{f=1}^{2048}(T'g(t,f) - P'i(t,f))^2}, (g \geq 1) \qquad (27)$$

3.2.4 Template-Verbesserung

Die jeweils verbesserte Version eines Templates T_{g+1} wird berechnet, indem man den Median aller ausgewählten spektralen Exzerpte heranzieht:

$$T_{g+1}(t,f) = \underset{s}{median}(P_s(t,f)) \qquad (28)$$

Wobei $P_{s,s}$ *= (1,…M),* die Spektrum-Exzerpte des vorigen Kapitels darstellt. Der Median wird deswegen zu Neuberrechnung herangezogen, weil dadurch Frequenz-Komponenten von anderen Instrumenten, die nicht zu den Drum-Sounds gehören unterdrückt werden. Die Frequenz-Komponenten von Drums scheinen sich in allen gewählten spektralen Exzerpten an mehr oder weniger derselben Position zu befinden, was nicht der Fall ist bei Frequenzanteilen harmonischer Instrumente. Wird nun der Median für gewisse Zeit *t* und Frequenz *f* berechnet, so werden die „Ausreißer" – Frequenzen der nicht-perkussiven Instrumente unterdrückt, während gleichzeitig die Komponenten der Drum-Sounds erhalten bleiben. Somit können auch Drum-Templates angepasst werden, obwohl in dem zu untersuchenden Musikstück gleichzeitig auch andere Begleitinstrumente vorkommen.

3.2.5 Template-Matching

Es muss nun auch eine spezielle Distanz-Messung erfolgen, da sich im zu untersuchenden Bereich, zwar ein eventuelles Drum-Template, jedoch auch andere

harmonische Komponenten befinden können und dadurch gewöhnliche Distanzmaße einen zu großen Unterschied feststellen würden.

Zu Beginn wird eine Gewichtungsfunktion eingeführt, die charakteristische, spektrale Punkte eines adaptierten Templates repräsentiert. Danach wird die Lautheitsdifferenz zwischen dem Template und den Spektrum-Exzerpten gebildet, wobei hier die Gewichtungsfunktion zum Einsatz kommt. Ist der Lautheitsunterschied größer als eine gewisse Threshold, so wird das jeweilige Exzerpt verworfen, ansonsten wird der Lautheitsunterschied ausgeglichen und das Exzerpt wird einer Distanzmessung unterzogen. In Abb. 11 ist der grobe Ablauf dieser Template-Matching-Phase des Systems zu sehen:

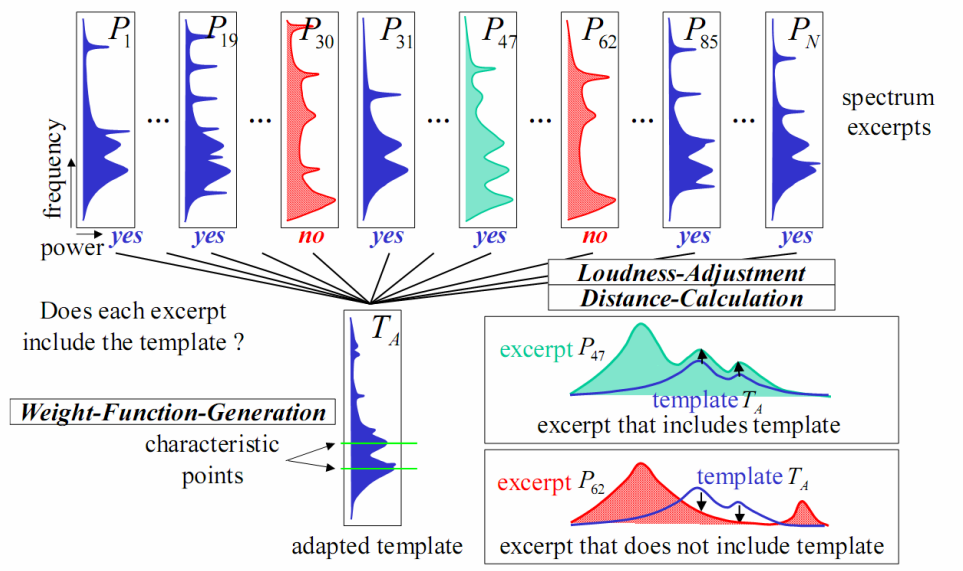

Abb 11. Ablauf des Template-Matching-Systems aus [Goto, 2004]

Die bereits erwähnte Gewichtungsfunktion repräsentiert die spektrale Charakteristik zu jedem Zeitpunkt *t* und zu jeder Frequenz *f* des adaptierten Templates. Sie ist wie folgt definiert:

$$\omega(t, f) = F(f)T_A(t, f) \qquad (29)$$

wobei T_A dem adaptierten Template entspricht und *F(f)* die Tief/Hochpass-Funktion aus Abb. 10 darstellt.

Nun muss die Lautheitsdifferenz zwischen Template T_A und dem Exzerpt P_i errechnet und gegebenenfalls ausgeglichen werden. Dies geschieht, indem gewisse Punkte in der Zeit-Frequenz-Domain eines jeden Frames mit Hilfe der Gewichtungsfunktion ermittelt werden. An diesen Stellen wird dann ein Differenzmaß η_i errechnet. Die gesamte Lautheitsdifferenz eines Frames wird dann wie in Abb. 12 dargestellt als δ_i berechnet. Ist die Lautheit von P_i um einiges geringer als die von T_A, so wird das Exzerpt verworfen und die nachfolgenden Schritte werden nicht durchgeführt. Ist die Lautheit über einem entsprechendem Schwellwert, so werden die beiden Lautstärken angepasst und es das Exzerpt wird in die nächste „Phase“ des Systems weitergeleitet. Die Lautheitsanpassung ist deshalb wichtig, weil nun die Distanz zwischen Exzerpt und Template berechnet wird, und unterschiedliche Lautstärken sich auf diese fehlerhaft auswirken würden.

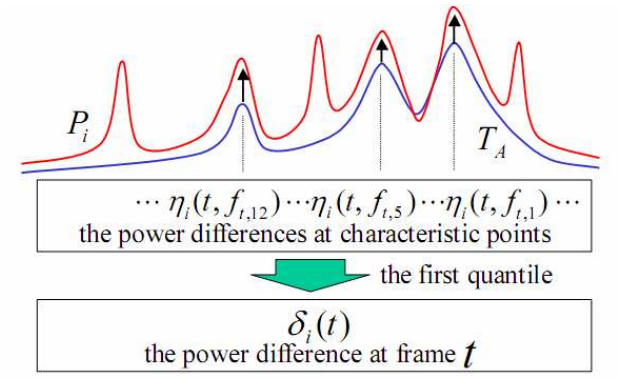

Abb. 12. Lautheitsermittlung an gewissen Punkten im Zeit-Frequenz-Bereich aus [Goto, 2004]

Die Distanz wird folgendermaßen berechnet:

$$\gamma_i(t, f) = 0 \text{ für } P_i'(t, f) - T_A(t, f) \geq \Psi \qquad (30)$$

$$= 1, \text{ ansonsten}$$

wobei $\gamma_i(t, f)$ der lokalen Distanz zwischen T_A und P_i' an der Stelle *t* und *f* entspricht.

Ψ ist eine negative Konstante. Sobald P_i' also größer wird als T_A, so wird $\gamma_i(t, f)$ auf 0 gesetzt. Die gesamte Distanz wird dann berechnet, indem die lokale Distanz über der Zeit-Frequenz-Domain, gewichtet mit der Funktion ω ermittelt wird:

$$\Gamma_i = \sum_{t=1}^{15} \sum_{f=1}^{2048} \omega(t, f) \gamma_i(t, f) \qquad (31)$$

Ist $\Gamma_i <$ einer Threshold Θ_Γ in einem gewissen Bereich P_i', so erkennt das System einen Anschlag der entsprechend untersuchten Drum-Klasse.

3.2.6 Ergebnisse/Evaluation

Es hat sich gezeigt, dass das System von Yoshii und Kollegen die besseren Ergebnisse erzielt, wenn zum Vergleich die bereits adaptierten Templates benutzt werden. Auch hier werden folgende als Qualitätskriterien zur Evaluierung die Recall-Rate, die Precistion-Rate sowie F-Measure herangezogen.

In [Goto, 2004] wurde das System an einer von Goto erstellten Datenbank getestet. Zehn 1-minütige Exzerpte wurden analysiert, wobei es sich um Pop-Songs mit Gesang und Instrumentalbegleitung handelte. In diesem Test wurde lediglich nach Bass- und Snare-Drum gesucht. Folgende Ergebnisse wurden im Durchschnitt erzielt:

	recall	**precision**	**F-measure**
Bass-Drum	90.2%	90.0%	0,90
Snare-Drum	83.4%	92.7%	0.88

Abb. 13 Zusammengefasste Ergebnisse der Evaluierung aus [Goto, 2004]

In [Goto, 2005] sind Testergebnisse eines Contests, der im Rahmen der MIREX2005 durchgeführt wurde zu sehen. Das System von Yoshii und Co. wurde dabei mit Systemen anderer Kollegen verglichen und konnte die besten Ergebnisse erzielen. Klassifiziert wurden dabei 3 Drum-Klassen, nämlich Bass-Drum, Snare-Drum sowie HH-Cymbals. 30 Sekunden Exzerpte von Songs aus allen musikalischen Richtungen wurden analysiert, wobei auch hier „ganze“ Songs inkl. etwaigem Gesang und begleitender Instrumente analysiert wurden. Ein repräsentatives Data-Set von ca. 20% der Testsongs war den Teilnehmern bereits vorab zugänglich, um ihre Systeme vorzubereiten. Abb. 14 zeigt die Ergebnisse des Contests:

Participant	Total	BD	SD	HH	Runtime*
Yoshii, K.	0.670	0.728	0.702	0.574	8534 [s]
Tanghe, K.	0.611	0.688	0.555	0.601	1337 [s]
Dittmar. C.	0.588	0.606	0.581	0.585	673 [s]
Paulus, J.	0.499	0.527	0.430	0.587	1137 [s]
Gillet, O.	0.443	0.598	0.428	0.334	21248 [s]

Abb. 14. Vergleich der F-Measure-und Runtime-Werte im Rahmen des MIREX Contests 2005, aus [Goto, 2005]

3.3 Informationstheoretischer Ansatz

Die bisher betrachteten Systeme zur Drum-Transcription haben folgende Gemeinsamkeit: Sie basieren zur Gänze auf Techniken und Methoden, die speziell im Bereich von Audiosignalen benutzt werden.

Nun gibt es jedoch andere Ansätze, denen in letzter Zeit viel Aufmerksamkeit geschenkt wurde. Dabei wird versucht die Trennung der einzelnen Quellen aus dem Originalsignal mit Hilfe von statistischen bzw. informationstheoretischen Methoden zu erreichen. Derry Fitzgerald und Kollegen haben sich in [Fitzgerald, 2003 - 1], [Fitzgerald, 2003 - 2] bzw. [Fitzgerald, 2004] intensiv mit solchen Methoden, wie der Independent Subspace Analysis (ISA) bzw. der Principal Component Analysis (PCA), sowie der Prior Subspace Analysis (PSA) auseinander gesetzt:

3.3.1 Independent Subspace Analysis (ISA)

Bei der Independent Subspace Analysis handelt es sich um eine Methode zur Aufspaltung eines einzelnen Misch-Signal, das aus mehreren Quellen zusammengemixt wurde. Sie basiert auf einer Redundanz-Verminderung in der Zeit/Frequenz-Ebene des Eingangssignals, wobei die einzelnen Sound-Quellen als nieder-dimensionale Unterräume dieser Ebene repräsentiert werden.

Es wird angenommen, dass sich das Eingangssignal aus einer Summe von p unbekannten, unabhängigen Sound-Quellen zusammensetzt:

$$s(t) = \sum_{q=1}^{p} s_q(t) \qquad (32)$$

Nun wird über dieses Signal eine Short Time Fourier Transformation (STFT) angewandt, wobei nur die Betragswerte der Koeffizienten vewendet werden. Man erhält das Signal Y, das eine Dimension von m x n aufweist. Hierbei stellt n die Anzahl der Frequenz-Kanäle und m die Anzahl der Zeit-Raster dar. Y enthält somit einen Spaltenvektor, der das Frequenzspektrum der Zeit *j* mit *1 <= j <= m* repräsentiert. Der Zeilenvektor von Y ist daher als Änderung eines Frequenzkanals k, mit *1 <= k <= n* über der Zeit anzusehen.

Nun wird angenommen, dass sich das Spektrum *Y* als eine Überlagerung von l statistisch unabhängigen Spektrogrammen Y_j zusammensetzt:

$$Y = \sum_{j=1}^{l} Y_j \qquad (33)$$

Eine weitere Annahme ist jene, dass sich jedes der Spektrogramme Y_j als äußeres Matrizzenprodukt einer gleich bleibenden Frequenz-Basis-Funktion f_j mit einer gleich bleibenden Amplituden-Hüll – oder Gewichtungsfunktion t_j darstellen lässt, also:

$$Y = \sum_{j=1}^{l} f_j t_j^T \qquad (34)$$

Diese unabhängigen Basis-Funktionen repräsentieren Features der individuellen Quellen. Jede Quelle besteht aus einer gewissen Anzahl dieser Basis-Funktionen, welche einen nieder-dimensionalen Unterraum formen, der die Sound-Quelle darstellt.

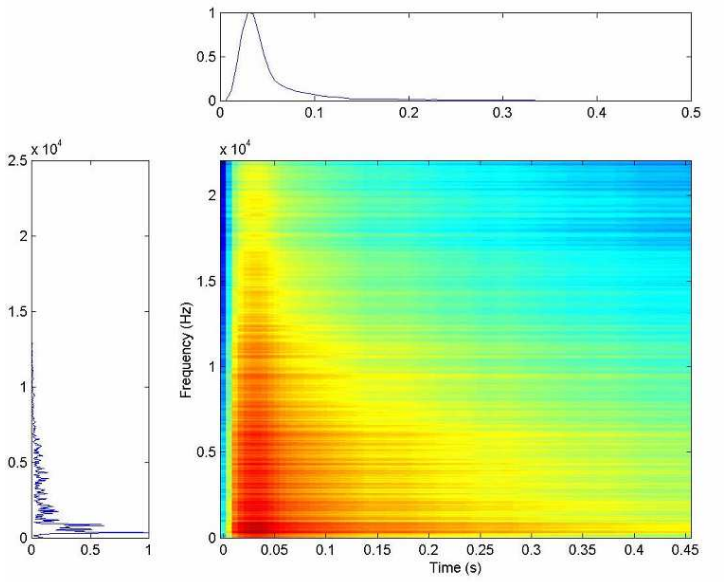

Abb. 15. Spektrogramm eines Snare-Schlages und die dazugehörenden Basis-Funktionen, aus [Fitzgerald, 2004]

Die oben erwähnte Aufspaltung des Spektrums in seine unabhängigen Quellen kann mit Hilfe der Principal Component Analysis, kurz PCA durchgeführt werden. Grundsätzlich wird dabei ein Satz aus korrelierten Variablen linear in einen Satz unkorrelierter bzw. orthogonaler Variablen transformiert. Diese Principal Components sind im Anschluss nach Varianz der ursprünglichen Variablen sortiert. Verwirft man Komponenten mit geringer Varianz, so können Redundanzen vermindert werden, da nur die „wichtigsten“ Komponenten behalten werden. In diesem Fall wird die PCA in Form der Singular Value Decomposition oder SVD durchgeführt. Y wird dadurch aufgespalten in :

$$Y = USV^T \qquad (35)$$

Hier ist U eine *m x m* Matrix, deren Spalten die Principal Components von *Y* im Frequenz-Bereich darstellen, *V* eine *n x n* Matrix, deren Spalten die Principal Componants zeitbasierend repräsentiert. *S* kann als Mixing-Matrix angesehen werden, deren Dimension *m x n* entspricht und die die entsprechenden Singular Values enthält. Wie schon erwähnt, werden nun Komponenten mit niedriger Varianz ausgeschieden. Behält man die ersten *l* Komponenten, so kann obige Formel folgend angeschrieben werden:

$$Y \approx \sum_{j=1}^{l} u_j s_j v_j^T \qquad (36)$$

wobei $u_j s_j$ als h_j und v_j als z_j zu folgender Formel in Matrix-Notation angeschrieben werden kann:

$$\mathbf{Y} \approx \mathbf{h}\mathbf{z}^{\mathbf{T}} \qquad (37)$$

Da PCA jedoch keine statistisch unabhängigen Basis-Funktionen returniert, muss ein weiteres Verfahren, die Independent Component Analysis, kurz ICA durchgeführt werden. Dabei wird versucht, gewisse beobachtete Signale, die sich aus einem Satz unabhängiger nicht-gaußscher Quellen zusammensetzen in einen Satz aus Signalen, die die abhängigen Quellen beinhalten umzuwandeln. Dieser Ansatz ist im Zusammenhang mit musikalischen Quellen sinnvoll, da diese als nicht-gaußisch angesehen werden können. Weiters wird angenommen, dass die unabhängigen Quellen linear gemixt wurden. Folgende Notation wird verwendet:

$$\mathbf{x} = \mathbf{As} \qquad (38)$$

wobei **x** die beobachteten Mix-Signal enthält, **s** die unabhängigen nicht-gaußschen Signale darstellt sowie **A** die entsprechende Mixing-Matrix repräsentiert. Um letztlich die unabhängigen Signalquellen zu erhalten, wird der zentrale Grenzwertsatz herangezogen, der besagt, dass eine gewisse Anzahl von nicht-gaußschen Quellen immer zu einer gaußschen Verteilung tendiert. Aufgrund dessen weisen die Mix-Signale in **x** eher eine gaußsche Wahrscheinlichkeitsverteilung auf, im Vergleich zu den Quell-Signalen in **s**. Also haben die ursprünglichen Quellen eine Wahrscheinlichkeitsverteilung, die eher nicht-gaußisch ist, als jede Mischung der Quellen.

Obwohl ICA weder die genauen Amplitudenwerte der Quell-Signale noch die genaue Reihenfolge von diesen wiederherstellen kann, ist die Wirksamkeit der Analyse in der Praxis trotzdem gegeben. Verschiedene öffentlich zugängliche Algorithmen, wie zum Beispiel FastICA oder Jade können zu diesem Zweck verwendet werden.

Wird ICA an obigem **h** durchgeführt, so erhält man die Basis-Funktionen unabhängig im Frequenz-Bereich:

$$\mathbf{f} = \mathbf{Wh} \qquad (39)$$

wobei **f** die Basis-Funktionen enthält und W die Unmixing-Matrix darstellt, die aus der ICA resultiert. Multipliziert man nun das Spektrogramm Y mit der Pseudoinversen $\mathbf{f}_{pinv}$ der Frequenz-Basis-Funktionen **f**, so erhält man die entsprechenden Amplituden-Basis-Funktionen **t**:

$$\mathbf{t} = \mathbf{f}_{\text{pinv}}\,\mathbf{Y} \qquad (40)$$

Die unabhängigen Spektrogramme erhält man dann wie in Gleichung 34 beschrieben.

Da ISA die Betragswerte der STFT-Koeffizienten benützt und dadurch die Phaseninformation des Eingangssignals verloren geht, ist es nicht möglich die getrennten Quell-Signale zu resynthisieren. Ein unsauberer Weg, diese verlorene Information zurück zu erhalten ist einfach die Phasen-Info der ursprünglichen STFT zu verwenden.

3.3.2 Einschränkungen von ISA

Obwohl ISA eine effektive Methode zur Trennung von gemixten Audio-Signalen darstellt, weist sie trotzdem einige Einschränkungen auf:

Im Rahmen der Independent Subspace Analysis wird die Annahme getroffen, die zu trennenden Quell-Signale seien in ihrem Frequenzverlauf stationär, was jedoch in der Praxis nicht der Fall ist. Durch Aufsplitten des Signals in kleinere Zeitbereiche können quasi-stationäre Zustände erzeugt werden. Jedoch müssen die so erhaltenen Basis-Funktionen der einzelnen Zeitfenster zum Beispiel mit Hilfe von zusätzlichen Algorithmen geclustert werden. Wird ISA eingesetzt, um Drum-Loops zu separieren, kann dieser Schritt übersprungen werden, da Drums im Großen und Ganzen gleich eher bleibende Tonhöhen aufweisen.

Die Qualität der Quell-Trennung hängt bei ISA merklich von der Länge des zu analysierenden Eingangsignals ab. Weist ein Audio-Signal zum Beispiel lediglich einen gleichzeitigen Snare- sowie Hihat-Anschlag auf, so können diese beiden Events mit Hilfe der ISA nicht korrekt separiert werden. Typischerweise sind für eine erfolgreiche Trennung von Hihat und Snare zwei bis vier Anschläge nötig. Diese Zahl variiert jedoch in Abhängigkeit von der Frequenz- und Amplitudencharakteristik der jeweiligen Instrumente.

Einen weiteren kritischen Punkt im Rahmen der Independent Subspace Analysis stellt die Auswahl der richtigen Anzahl an benötigten Komponenten der

Principal Component Analysis dar. Auch hier hängt die Anzahl stark von Frequenz – sowie Amplitudencharakteristik der verwendeten Quell-Signale ab. Zusätzlich kommt es zu einem Trade-Off zwischen der Anzahl an verwendeten Komponenten und der Qualität der resultierenden Basis-Funktionen. Des Weiteren kann sich die Anzahl der zu behaltenden Komponenten mit dem zeitlichen Verlauf des Eingangssignals ändern und die Wahl muss daher mit Vorsicht vorgenommen werden.

Da sich Drum-Sounds in verschiedenen Frequenzbereichen überlappen, ist es nach der ISA möglich, dass sich in einem separierten Kanal auch Anschläge anderer Quellen befinden. Dieses Problem kann, wenn eine gute Trennung vorgenommen wurde, zum Teil mit Hilfe von einfachen Threshold-Kriterien beseitigt werden.

Zuletzt ist die Reihenfolge der einzelnen Komponenten nach Durchführung der Independent Component Analysis nicht genau definiert und die Kanäle können oft nur durch den Einsatz zusätzlicher Frequenz-Kriterien richtig identifiziert werden.

Trotz dieser Limitationen stellt die ISA eine adäquate Möglichkeit zur Trennung unterschiedlicher Source-Quellen aus einem Eingangssignal aus dem Bereich der statistischen bzw. informationstheoretischen Ansätze dar.

3.3.3 Prior Subspace Analysis

Um die Einschränkungen der Independent Subspace Analysis zu umgehen wird von Fitzgerald und Kollegen in [Fitzgerald, 2003 - 1] sowie [Fitzgerald, 2004] die Methode der Prior Subspace Analysis genauer beleuchtet. Hier wird im Vorhinein bekanntes Wissen über die Beschaffenheit der zu trennenden Signale angewandt.

Auch im Rahmen der PSA wird versucht, das Spektrogramm auf ähnliche Weise wie bei der ISA zu trennen. Jedoch wird angenommen, dass bereits im Vorhinein an die eigentlichen Frequenz-Basis-Funktionen *fj* gut angenäherte Funktionen *fp* existieren. Somit kann *fj* durch *fp* substituiert werden und man erhält für das Spektrogramm Y:

$$Y = \sum_{j=1}^{l} f_p t_j^T \qquad (41)$$

Wird nun die Pseudoinverse der Substituts-Frequenz-Funktion *fp* mit dem Spektrogramm multipliziert, so erhält man die geschätzte Amplitunden-Basis-Funktion:

$$\hat{\mathbf{t}} = \mathbf{f}_{\mathrm{pp}}\mathbf{Y} \qquad (42)$$

wobei *fpp* die Pseudoinverse der angenommenen Frequenz-Basis-Funktion *fp* darstellt.

Nachdem die dadurch erhaltenen Amplituden-Funktionen statistisch nicht als unabhängig angesehen werden können, wird in weiterer Folge die ICA durchgeführt, um diese in unabhängiger Form als t zu erhalten.

$$\mathbf{t} = \mathbf{W}\hat{\mathbf{t}} \qquad (43)$$

Eine verbesserte Schätzung der Frequenz-Basis-Funktion erhält man nun durch:

$$\mathbf{f} = \left(\mathbf{Y}\mathbf{t}_{\mathrm{p}}\right)^{\mathrm{T}} \qquad (44)$$

Nachdem man nun die verbesserten Frequenz-Basis-Funktionen erhalten hat, können die einzelnen Spektrogramme der Quell-Signale wie in Kapitel 5.2 beschrieben erfaßt werden.

Der Unterschied von PSA zu ISA ist jener, dass im Rahmen der ISA die Basis-Funktionen im Prinzip „blind“ gewählt werden, um im Anschluss PCA zur Auswahl der wichtigsten Komponenten durchzuführen. PSA hingegen wendet zuvor erfaßtes Wissen über diese Funktionen an, um diese danach genauer bestimmen zu können.
In [Fitzgerald, 2003 - 1] wird dieses nötige Zusatzwissen erzeugt, indem vor dem ISA-Schritt die Einzelquellen, sprich die zu klassifizierenden Drums über einen längeren Zeitraum mittels einer erneuten ISA genau untersucht werden, um danach

die angenäherten Basis-Funktionen zu erhalten. Die mit ISA untersuchten Einzelspuren werden nun mittels PCA weiter bearbeitet. Hier benutzen Fitzgerald und Kollegen pro Solo-Quelle die ersten 3 Komponenten, die aus der PCA rückgegeben werden. Im Anschluss daran wird an diesen Spektrogrammen die ICA durchgeführt, um nun die unabhängigen, angehäherten Unterräume bzw. Basis-Funktionen für die PSA zu erhalten. Es werden nun die Frequenz-Unterräume mit der größten projezierten Varianz verwendet. Im Anschluss daran wird pro Signal-Quelle ein K-Means-Clustering Algorithmus durchgeführt, um pro Drum-Klasse einen Unterraum zu erhalten, der diese am besten charakterisiert.

Abb. 16 zeigt die aus PSA erhaltenen Frequenz-Funktionen für Bass-Drum, Snare-Drum sowie Hihat. Die typische Frequenz-Charakteristik jeder Drum-Klasse ist deutlich zu sehen, beispielsweise den hohen Energieanteil an niederfrequenten Komponenten einer Bass-Drum. Die Snare-Drum weist zwar auch in den niederen Regionen eine hohe Signal-Energie auf, die Resonanz des Anschlags liegt jedoch in Frequenz-Bereichen, die höher als die der Bass-Drum sind. Generell hat eine Snare-Drum eine weite „Streuung“ über die Frequenz-Achse. Bei Hihats verteil sich ebenfalls einen Großteil der Energie in einem breiten Bereich des Spektrums.

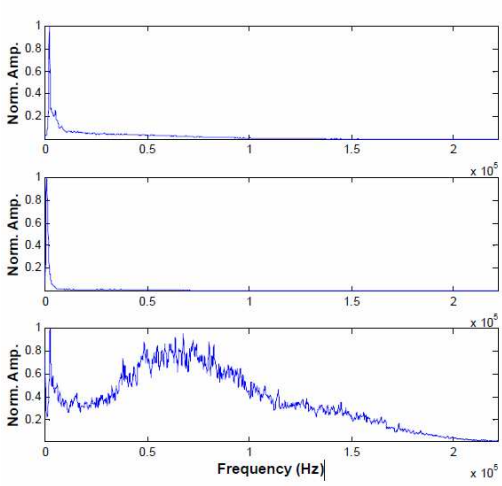

Abb. 16. Subspaces für Snare-Drum (oben), Bass-Drum (mitte) sowie Hihats, aus [Fitzgerald, 2003 - 2]

Abb 17. zeigt die dazugehörigen aus PSA erhaltenen unabhängigen Amplituden-Basis-Funktionen der Sample-Sequenzen. Die gut durchgeführte Trennung der Quellen ist daran ersichtlich, da zum Beispiel Bass-Drum-Anschläge nur als kleine Spitzen in der Basis-Funktion der Snare-Drum auftreten.

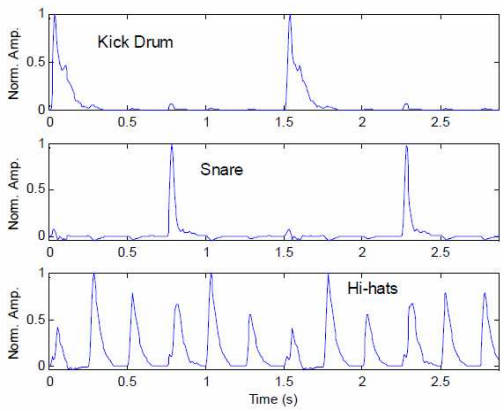

Abb. 17. Aus PSA erhaltene unabhängige Prior-Amplituden-Basis-Funktionen, aus [Fitzgerald, 2003 - 2]

3.3.4 PSA in Zusammenhang mit harmonischen Instrumenten

Als Voraussetzung für ISA wurde erwähnt, dass nur nach Signal-Quellen mit gleichbleibender Tonhöhe gesucht werden kann. Diese Regel kann im Zusammenhang mit PSA dahingegend gelockert werden, als dass dies nur für jene Signal-Quellen zutreffen muss, nach denen auch gesucht wird. Nachdem Drums eben dieses Kriterium erfüllen, ist lt. Fitzgerald und Kollegen ihre Klassifikation im Umfeld von harmonischen Instrumenten besonders für die Prior-Subspace-Analysis geeignet.

Da Akkorde bzw. Töne von harmonischen Instrumenten sehr wohl den Amplitudenverlauf des Signals über der Zeit verändern, wird die Trennung/Kategorisierung der Drums etwas erschwert. Es ist jedoch zu erwähnen, dass harmonische Instrumente, im Gegensatz zu Drums die Spitzen des spektralen Signals eher nur an den Positionen der Oberwellen aufweisen, und die restlichen Frequenzen, die dazwischen liegen, eher eine geringere Amplitude aufweisen. Da auch bei Akkorden die gespielten Töne meist in einem harmonischen Verhältnis auftreten, überlappen sich auch hier die spektralen Spitzen eher an den entsprechenden Oberwellen. Wird nun ein Akkord oder Ton gespielt, so gibt es über weite Stellen des Frequenz-Spektrums Bereiche, die nur wenig Signalenergie aufweisen, sogenannte „Täler“. Nur bei den bereits erwähnten Oberwellen findet man entsprechende Spitzen. Wählt man nun eine höhere Auflösung im Zusammenhang mit der durchgeführten FFT, so erkennt man, dass Spektren mit höherer Frequenz-Auflösung weniger Störspitzen zwischen den Oberwellen des Tones bzw. des Akkordes aufweisen, als Spektren mit niedrigerer Auflösung. Ein gleichzeitig durchgeführter Snare-Schlag kommt somit, auch in Begleitung mit harmonischen Komponenten trotzdem gut zur Geltung. Dies ist aus Abb. 18 ersichtlich.

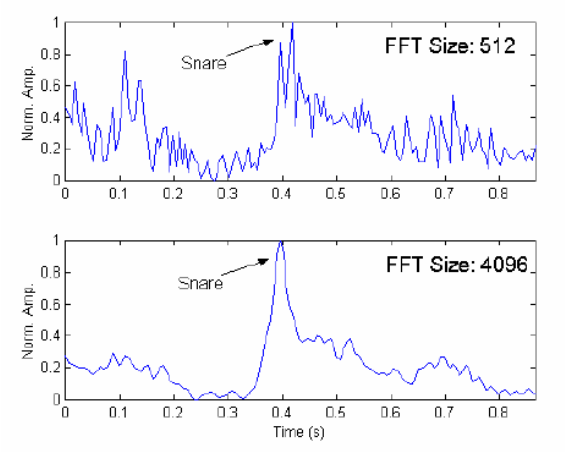

Abb. 18. FFT einer Mischung aus Snare-Schlag mit harmonischer Begleitung. Die durch geringere Auflösung enstandenen Störeinflüsse (oben) sind deutlich stärker als bei Verwendung einer höheren Auflösung (unten) [Fitzgerald, 2003 - 1]

Für eine adäquate Hihat-Erkennung ist der Störeinfluss der harmonischen Komponenten jedoch weitaus größer, weshalb zusätzliche Schritte unternommen werden müssen. Dies ist wohl darauf zurückzuführen, dass Hihat-Anschläge eine eher konstante spektrale Dichte über den gesamten Frequenzbereich aufweisen und dadurch anfälliger für harmonische Störeinflüsse sind, als zb. Bass-Drums, deren Hauptenergie im unteren Bereich liegt.

Abhilfe kann hier die Einbeziehung der Power Spectral Density (PSD) schaffen. Diese gibt Auskunft über die durchschnittliche Energie in den verschiedenen Regionen des Spektrums. In Pop-Songs befindet sich beispielsweise der Großteil der Signal-Energie in den unteren spektralen Bereichen, also eher im Bass und in den unteren Mitten. Dies kann man sich nun zu Nutze machen, in dem man die Spektrogramm-Werte durch die jeweilige PSD dividiert, was zur Folge hat, dass höher-frequente Anschläge, also jene der Becken und der Hihat besser hervortreten und somit erkennbar sind.

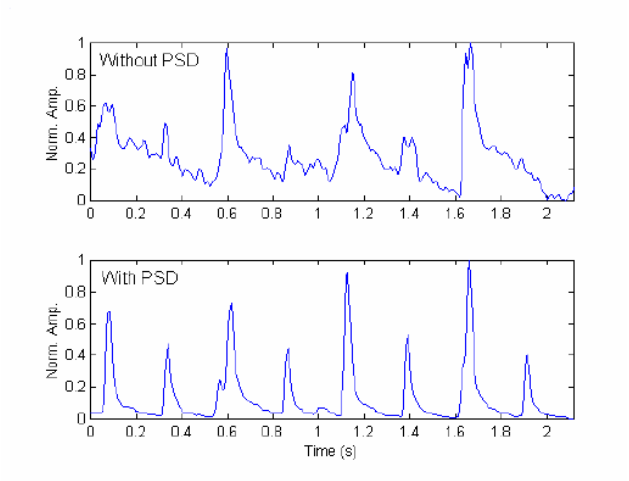

Abb. 19. Hihat-Amplitude über Zeit, ohne Einbeziehung der PSD (oben) und mit Einbeziehung der PSD (unten), aus [Fitzgerald, 2003 - 1]

Zur Berrechnung der Energiedichte wird eine Eigenvektor-Methode verwendet, die nur eine geringe Anzahl an Eigenvektoren mit einbezieht, da nur die „grobe“ Energieverteilung in den signifikanten, energiereichen Regionen nötig ist. In Abb. 19. lässt sich der Unterschied des Hihat-Signals im Zeitbereich deutlich erkennen. Die Anschläge treten im unteren Bild deutlicher hervor und lassen sich dadurch auch besser erkennen.

Da sich der Störeinfluss der tonalen Instrumente im Erkennungsprozess von Bass-Drum und Snare-Drum jedoch immer noch negativ bemerkbar macht, wird eine einfache Threshold-Regel eingeführt. Diese setzt alle Amplitudenwerte des Zeitbereichs unter dem normalisierten Pegel von 0.4 auf 0. Für die Hihat ist diese Regel nicht mehr nötig, da hier die ungewünschten Störeinflüsse bereits durch die Division mit dem PSD-Wert eliminiert werden.

Das Einführen der Threshold-Regel hat jedoch einen Nebeneffekt auf den Erkennungsprozess. Durch die 0-Setzung von weiten Teilen des Signals treten bei Drum-Anschlägen „scharfe“, eher unnatürliche Spitzen im Bass-Drum und Snare-Drum Signal auf. Im Gegensatz dazu sind die Hihat-Anschläge aber eher natürlich, da diese durch die 0-Setzungs-Regel nicht beeinflusst werden. Werden jetzt die

gegensätzlichen Signalformen in den ICA-Algorithmus hineingeschickt, so enhalten die resultierenden unabhängigen Signale ungewünschte Artefakte, wie beispielsweise starke Amplitunden-Modulationen, in Bereichen, wo eigentlich keine hohen Signal-Stärken auftreten sollten. Um dem entgegenzuwirken, wird der ICA-Algorithmus nur mehr auf Bass-Drum- und Snare-Drum-Signale angewandt, da diese beiden von der Signal-Form eher ähnlich sind und daher keine Artefakt-Bildung auslösen. Das Hihat-Amplituden-Signal wird nun direkt zur Onset-Detection weitergereicht, ohne den ICA-Algorithmus zu durchlaufen. Dies hat zur Folge, dass in seltenen Fällen die Hihat-Anschläge nicht mehr so gut erkannt werden, zusätzlich ist nun auch die gleichzeitige Erkennung von mehreren Events, beispielsweise eines Snare- und Hihat-Anschlages nicht mehr möglich. Da jedoch die Hihat in Songs meist zur gleichen Zeit in Kombination mit einer Snare-Drum vorkommt, schmälert dies die Effizienz des Algorithmus lt. Fitzgerald und Kollegen nur marginal.

3.3.5 Ergebnisse/Evaluation

Zu Beginn wurde Fitzgeralds PSA-Methode in [Fitzgerald, 2003 - 1] an 15 Drum-Loops, die keine musikalische Begleitung beinhalteten getestet. Klassifiziert wurden dabei Bass-Drum, Snare-Drum sowie Hihat. Die durchschnittliche Erkennungsrate des Systems lag dabei bei 92.5%. Im Vergleich dazu, konnte mit der ISA respektive der Sub-Band-ISA lediglich eine Quote von 89.5% erreicht werden. Der PSA-Algorithmus zeigte sich aus performanter, als die ISA-Methode und war im Vergleich 10 mal (Sub-Band-ISA) bzw. 5 mal (ISA) schneller. Dies ist darauf zurückzuführen, dass im Rahmen der PSA die Reihung der Komponenten mit Hilfe der PCA weg fällt.

Eine genauere Evaluation des PSA-Systems wurde jedoch an Songs mit harmonischer Begleitung durchgeführt. Dabei wurde ebenfalls nach Bass-Drum, Snare-Drum, sowie Hihat klassifiziert. Zwanzig Exzerpte aus allen gängigen populären Sparten wurden analysiert, wobei alle Drum-Patterns zuvor von Experten transkribiert wurden.

Type	Total	Missing	Incorrect	%
Snare	57	1	9	82.5
Kick	84	4	7	86.9
Hi-hats	238	14	30	81.5
Overall	379	19	46	82.8

Abb. 20. Ergebnisse des PSA-Systems aus [Fitzgerald, 2003 - 1]

Es ist zu erkennen, dass, obwohl das System eine robuste Leistung erbringt, die Ergebnisse an Songs mit harmonischer Begleitung etwas schlechter ausfallen, als an reinen Drum-Loops.

Einmal wurde eine Bass-Drum mit dem Anschlag einer Bass-Gitarre verwechselt, die restlichen 4 nicht erkannten Bass-Drum Anschläge konnten zwar im Bass-Drum-Signal ausgemacht werden, die Erkennungs-Threshold war jedoch zu hoch angesetzt. 6 mal innerhalb Genres, und zwar innerhalb von Disco-Songs wurde die Snare falsch klassifiziert, was darauf zurückzuführen ist, dass in diesem Genre der Snare-Sound generell weniger Höhen als in anderen Genres aufweist. 5 der inkorrekt erkannten Snares traten gleichzeitig mit Hihat oder Bass-Drum auf, und wurden fälschlicherweise der jeweiligen anderen Drum-Klasse zugeordnet. Hihat-Fehler traten entweder auf Grund einer zu hoch angelegten Threshold oder aber durch nicht eliminierte Störungen der PSD-Division auf.

4. Implementierung eines einfachen Drum-Transcription-Systems

Im Rahmen des Bakkelaureats-Projektes soll die Implementierung eines eigenen, simplen Drum-Transcription-Systems vorgestellt werden. Dieses soll drei Klassen von Drums, nämlich Bassdrum, Snaredrum sowie Overheads erkennen. Das System wurde in Matlab implementiert und es wurde eine einfache GUI zum Trainieren bzw.

Evaluieren von Audio-Beispielen erstellt. Zum Trainieren und Evaluieren der Systems wurde das Machine-Learning-Tool WEKA aus [WEKA-Web] benutzt.

Nach grundlegendem Studium der derzeitig gängigen Verfahren, von denen einige bereits im Vorfeld dieser Arbeit vorgestellt wurden, wurde zu Beginn entschieden, einen Ansatz zu wählen, der dem von Richard und Gillet aus [Richard, 2005 - 1] ähnlich ist. Aus diesem Grund wurde die in [Richard, 2006] vorgestellte ENST-Audio Datenbank für Trainings – und Evaluierungszwecke verwendet. Nachdem ursprünglich eine Trennung der harmonischen und stochastischen Signal-Anteile mit Hilfe der Noise-Subspace-Projection geplant war, wurde dieser Ansatz im weiteren Verlauf des Projektes verworfen, da sich in der Sample-Datenbank lediglich Aufnahmen von reinen Drum-Sounds befanden, also keine adäquaten Mixes von Drums und begleitender Musik zu finden waren und auch die Implementierung der Noise-Subspace-Projection nach eingehender Auseinandersetzung mit [Richard, 2005 - 1], sowie [Richard, 2005 - 2] nicht ersichtlich war. Es wurde daher ein einfacher Ansatz auf Basis eines Oktav-Band-Spektrogramms gewählt, der in den folgenden Abschnitten näher erläutert wird.

4.1 Grundlegender Aufbau des Systems

Mit Hilfe einer FFT wird das Spektrogramm des Eingangssignals gebildet, welches im Wav-Format vorliegt. Das Signal wird in je 1 Oktav-breite Frequenz-Bereiche unterteilt, um die Frequenz-Information zu reduzieren sowie eine gezieltere Unterscheidung der an den Anschlägen beteiligten Spektralbereichen zu ermöglichen. Um unabhängig(er) von der Lautstärke des Eingangssignals bzw. der Anschlagstärke der einzelnen Onsets zu sein, wird eine Pegel/Lautheitsanpassung des Spektrogramms durchgeführt.

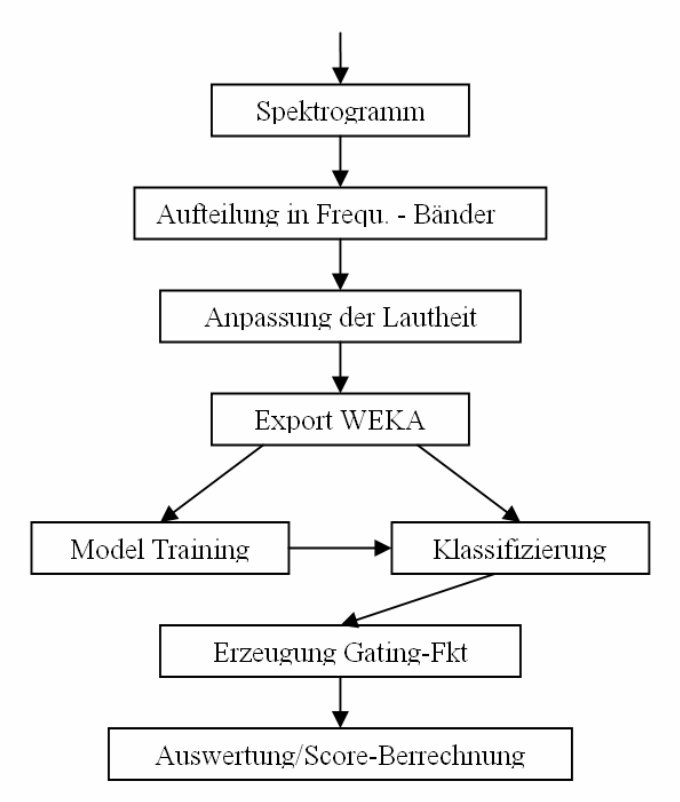

Abb. 21. Ablauf-Diagramm des implementierten Systems

Um im Machine-Learning-Tool WEKA Files als Trainings/Evaluierungs-Objekte eines Klassifizierers verwenden zu können, müssen die Daten als *.ARFF – File exportiert werden. Dies geschieht im nächsten Schritt. Danach können die so exportieren Informationen zum Trainieren eins Klassifizierers oder aber zum Klassifizieren von Test-Daten an zuvor trainierten Modellen verwendet werden. Die vom WEKA erhaltenen vorhergesagten Anschlags-Werte werden nun mit einem einfachen Algorithmus in Gate-Funktionen für die drei Drum-Klassen, also Bassdrum, Snaredrum sowie Overheads umgewandelt. Zuletzt werden die so erzeugten Anschlags-Werte mit den eigentlichen, korrekten, aus der ENST-Datenbank extrahierten Anschlagswerten verglichen.

4.2 Oktavband/Spektrogramm

Eingangsseitig wird das Audio-Signal mit 22050 Hz abgetastet. Diese Sampling-Rate wurde gewählt, um den Rechenaufwand im Vergleich zu 44100 Hz geringer zu halten. Die Fenstergröße beträgt 2048 Samples, wobei es zwischen den Fenstern zu einer 25%-igen Überlappung kommt. Als Fensterfunktion wird ein Hanning-Fenster benutzt. Anschließend wird das Spektrogramm auf 8 je eine Oktav umfassende Frequenz-Bänder reduziert

.

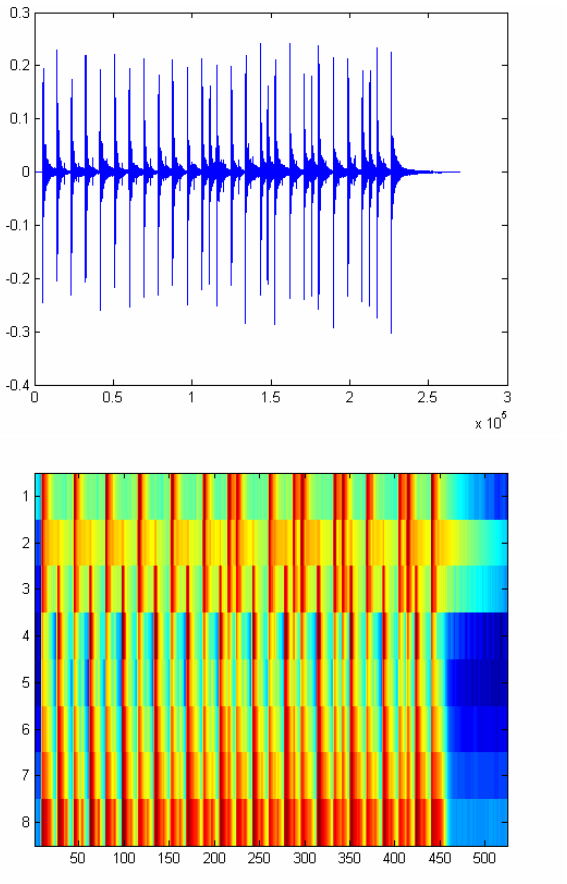

Abb. 22. Eingangssignal im Zeitbereich (oben), dazugehöriges Spektrogramm (unten), wobei die Spektralwerte bereits in [db] angegeben sind.

4.3 Anpassung der Lautstärke

Um den Einfluss der Onset-Lautstärke bzw. der Lautstärke des Gesamtsignals zu verringern, wird im nächsten Schritt eine entsprechende Anpassung durchgeführt. Dabei wird für jedes Zeit-Frame die durchschnittliche Lautstärke über die 8 Frequenz-Kanäle errechnet. Nun wird von jedem Frame seine durchschnittliche Lauststärke subtrahiert. Die Subtraktion wird durchgeführt, da es sich bei den Spektrogramm-Values bereits um db-Werte handelt. Wird zb. eine Hihat stärker angeschlagen, so haben die entsprechenden Frames im Spektrogramm mehr Energie und es wird eine stärkere Anpassung durchgeführt, als wenn es sich um einen leiseren Anschlag handelt.

$$Avg_k = \frac{1}{M} * \sum_{i=1,k}^{M} A_{i,k} \qquad (45)$$

Wobei

Avg_k	durchschnittlicher Wert des k-ten Zeitfensters,
M	Anzahl der Frequenzkanäle, in unserem Fall also 8,
$A_{i,k}$	Spektrogramm-Wert, i-tes Freuquenzband, k-tes Zeitfenster

Jeder Spektralwert wird dann wie folgt berrechnet:

$$B_{i,k} = A_{i,k} - Avg_k \qquad (46)$$

Wobei $B_{i,k}$ dem angepassten Spektrogramm Wert zum *i*-ten Frequenzband und zum *k*-ten Zeitfenster entspricht.

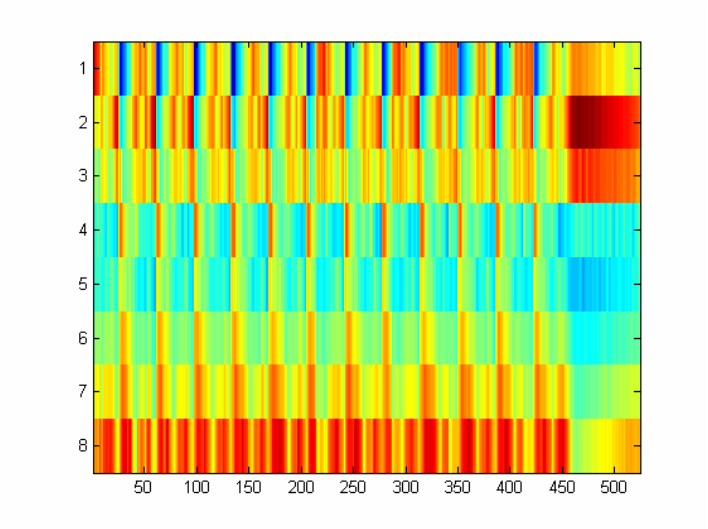

Abb. 23. Angepasstes Spektrogramm aus Abb. 22 in [db]

4.4 Export WEKA

Im diesem Schritt werden die Signal-Informationen entsprechend aufbereitet, um sie im Anschluss mit Hilfe des Machine-Learning-Tools WEKA weiter verarbeiten zu können. Trainings- und Testset können dabei bequem über die graphische Benutzeroberfläche festgelegt werden (siehe Abb. 25).

WEKA benutzt das sogenannte „Attribute-Relation File Format“ oder kurz ARFF um die Ein-/Ausgabedaten zu repräsentieren. Wie in [ARFF-Web] ersichtlich werden in ARFF Files sogenannte Relations definiert. Im Header werden die Attribute festgelegt, die den „Spalten“ der Daten entsprechen. Zu jedem Attribute Set können nun ein oder mehrere Klassen festgelegt werden. Natürlich muss man die entsprechenden Datentypen eines jeden Attributes (zb. NUMERIC, STRING…usw)

festlegen. Klassen-Werte können ebenfalls als numerische Werte oder verbale Ausdrücke definiert werden. In der DATA-Section des Files befinden sich die eigentlichen Daten. Pro Zeile findet man die entsprechende Anzahl an definierten Attributen sowie eine oder mehrere Klassen-Labels vor. Eine Zeile, auch „Instanz“ genannt entspricht also einem Messwert und dessen Klassifizierung.

Im Rahmen dieses Projektes werden die zu exportierenden Daten so gewählt, dass jeweils 4 Zeit-Frames pro Frequenzband, also je 32 Werte (= 4 Zeit-Frames x 8 Frequ.-Bänder) eine Instanz bilden. Die Klasseneinteilung wurde so gewählt, dass pro zu identifizierender Klasse, also Bassdrum, Snaredrum und Overheads ein eigenes ARFF-Daten-Set erstellt wird. Somit wird zum Training bzw. zum Evaluieren der Daten je ein File für Bassdrum, eines für Snaredrum und eines für die Overheads exportiert.

Da ein Zeit-Sample ungefähr einem Wert von ca. 23 ms entspricht, werden je 4 Samples, also ein Zeitbereich von ca 92ms herangezogen, da ein Drum-Event vom Anschlag bis zum Ausklang ebenfalls eine ungefähre Länge von 90ms aufweist. Befindet sich ein Anschlag der Trainingsdaten in diesem Zeitfenster, so wird als Klassen-Wert dieser Instanz eine 1 gesetzt. Findet kein Anschlag in diesem Zeitbereich statt, so wird die Instanz als 0 klassifiziert. Die Zeitwerte der Anschläge aus dem Trainingsset sind innerhalb der ENST-Datenbank als Text-File angegeben. Diese Files werden eingelesen und Anschlagswerte von Bassdrum, Snaredrum und Overheads werden mit dem Zeitbereich der einzelnen Instanzen verglichen, um diesen danach ein entsprechendes Klassen-Label zuordnen zu können. Anfangs wurde das Spektrogramm in Schritten von je 4 Sample-Werten durchgegangen. Es wurde dann jedoch beschlossen, den Zeitsprung zwischen den Instanzen auf je 1 Sample zu setzen, um die 4-fache Menge an Trainingsdaten zu erhalten. Pro Anschlag ergibt sich dadurch im ARFF-File in 4 hintereinander folgenden Zeilen als Klassen-Label eine 1.

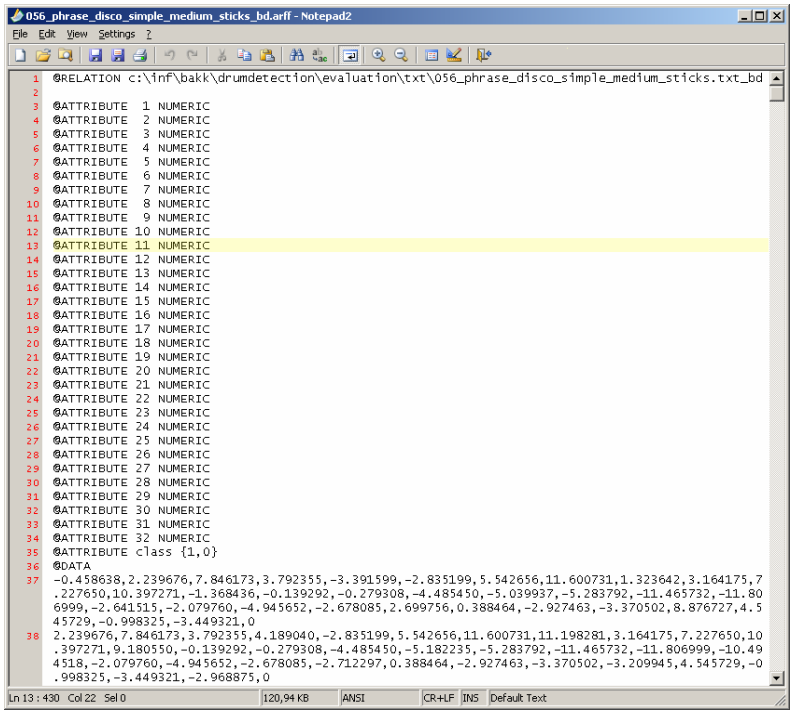

Abb. 24. Beginn eines generierten ARFF-Files zum WEKA-Export für Bassdrum-Klassifizierer

4.5 Model-Training

Um einen Klassifizierer zu trainieren wird ein einziges ARFF-File mit den Trainingsdaten für alle Files aus dem Trainingsset pro Drum-Klasse erstellt. Es ergeben sich daher pro Training 3 ARFF-Files, je eines für Bassdrum, Snaredrum sowie Overheads.

Der WEKA-Aufruf erfolgt direkt über Matlab, in dem über die Konsole das entsprechende WEKA-Command ausgeführt wird.

Wählt man in der grafischen Test-Umgebung des Systems ein oder mehrere Trainings-Files (Musik-Samples der ENST-Datenbank), so wird je ein ARFF-File für Bassdrum, eines für Snare sowie eines für Overheads erstellt. Mit diesem File, dessen Größe je nach Anzahl und Länge der Test-Samples variiert wird dann je ein Klassifizierer für Bassdrum, Snaredrum und Overheads trainiert. Diese drei trainierten

Klassifizierer werden jeweils als Bassdrum-, Snaredrum-, und Overhead-Model gespeichert und können nun zur Klassifikation eingesetzt werden.

4.6 Klassifizierung

Um nun Anschläge aus einem Test-Set zu ermitteln, wird ein für dieses Musik-Sample erzeugtes ARFF-File an den zuvor trainierten Klassifizierern mit Hilfe von WEKA ermittelt. Dabei wird die ARFF-Datei wie vorhin beschrieben erzeugt. Die so vorhergesagten Anschläge werden direkt mit den richtigen Anschlägen, die aus der ENST-Datenbank extrahiert wurden, verglichen. Auch hier erfolgt der Aufruf von WEKA über die Konsole direkt aus Matlab und wird „dunkel“ im Hintergrund ausgeführt.

4.7 Erzeugen einer Gating-Funktion

WEKA returniert die erzeugten Ergebnisse im Text-Format. Dieser Text wird in einer eigenen Funktion „geparsed“. Das Ergebnis ist eine Gating-Funktion, bestehend aus 1en im Bereich eines vermuteten Anschlags und 0en in allen anderen Regionen.

Bei der Erzeugung selbst wird zwischen Bassdrum, Snare und Overheads differenzert und eine für die Drum-Klassen unterschiedliche „Smoothing-Funktion“ angewandt. Kommt es nun zu einem Anschlag, so wird in der Ausgabe-Gate-Funktion ein Zeitfentser mit 4 Samplewerten erzeugt. Für jede Klasse wird also pro Anschlag in der Ausgabe-Funktion ein ca. 92ms langes Fenster generiert. Befindet sich ein eigentlicher Anschlag innerhalb dieses Fensters wird der vermutete Onset als richtig, ansonsten als falsch gewertet.

4.8 Grafisches Interface

Wird das System an einem File evaluiert, so werden entsprechend aufbereitete grafische Ausgaben erzeugt. Werden in der Testumgebung mehrere Files gleichzeitig markiert und evaluiert, so erhält man lediglich die aufsummierten richtigen, falschen sowie eigentlichen Anschläge.

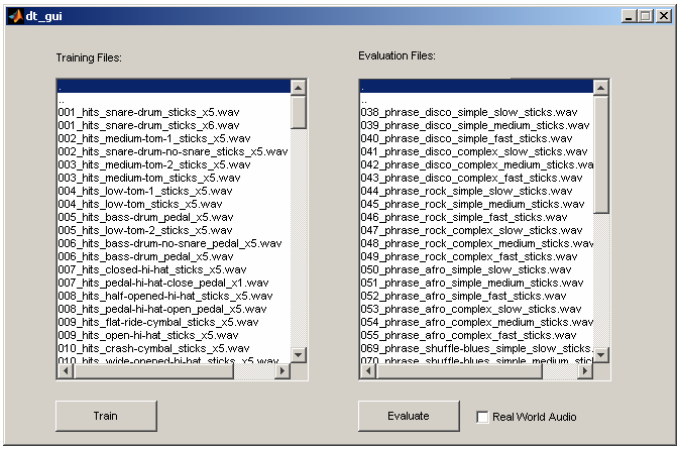

Abb. 25. Grafisches Interface der Test-Umgebung, Training im linken Bereich, Evaluierung im rechten Bereich

4.9 Ergebnisse/Evaluation

Im Rahmen der Auswertung werden, pro Evaluierung an einem oder auch mehreren Files die Anzahl der eigentlichen Anschläge, die Anzahl der gesamt detektierten Anschläge sowie die Anzahl der richtig detektierten Anschläge erfasst. Mit diesen ermittelten Werten, werden dieselben Qualitätskriterien wie in Formel 14, 15 und 16 berechnet.

Innerhalb der ENST-Datenbank wurden 15-20 Sekunden-Exzerpte aus vielfältigen Genres von 3 Drummern unabhängig von einander eingespielt und aufgenommen. Die Anschläge wurden entsprechend erfasst und transkribiert. Die Genres umfassen zum Beispiel Songs aus dem Bereich Hardrock, Funk, Disco..uvm.

Von jedem Genre gibt es komplexe sowie simple Ausschnitte, außerdem findet man auch pro Genre Samples mit verschiedenen Geschwindigkeiten (Slow, Medium, Fast). In der Datenbank sollten eigentlich auch instrumentale Begleitspuren vorhanden sein, jedoch waren diese großteils gemutet oder nicht auffindbar. Daher wurde das System an reinen Drum-Spuren trainiert und auch evaluiert.

Als Klassifizierer für alle drei Klassen, also Bassdrum, Snaredrum und Overheads wurde je ein Random-Forest mit den Parametern I = 10, K = 0, S = 1 verwendet. Dieser wurde gewählt, nachdem an verschiedenen Testdaten (66% Percentage-Split) folgender Vergleich mit einem SVM- und einem KNN-Klassifierer durchgeführt wurde:

Exzerpt	**Drum-Klasse**	**Random Forest**	**SVM**	**KNN**
Rock	BD	95.8%	92.2%	95.8%
Rock	SN	96.4%	97.0%	92.6%
Rock	OH	85.5%	84.3%	84.9%
Afro	BD	95.4%	95.3%	94.2%
Afro	SN	76.7%	77.9%	77.9%
Afro	OH	87.2%	87.2%	74.4%
Disco	BD	87.4%	85.0%	83.8%
Disco	SN	85.6%	81.2%	84.7%
Disco	OH	80.6%	74.0%	75.3%

Die Klassifizierer wurden in ihren Parametern mit den WEKA Default-Werten getestet. (**RandomForest:** I10, K0, S1,

SVM: -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 –K,

KNN: IBK- K1, W 0 – A).

Der getestete Random Forest – Klassifizierer lieferte die solideste Performance in allen 3 Drum Klassen und wurde daher als Klassifizierer für Bassdrum, Snaredrum und Overheads gewählt.

Trainiert wurde an einer vielfältigen Auswahl an Files und auch an verschiedenen Drummern, wobei immer nur Files von jenem Drummer evaluiert wurden, von dem sich keine Samples im Trainingsset befanden.

Das System wurde zu erst an fast allen vorhandenen Files von Drummer 1 und Drummer 3 evaluiert. Im Anschluss daran wurde an fast allen Phrasen von Drummer 2 evaluiert, wobei nur Exzerpte klassifiziert wurden, die mit Sticks eingespielt waren. Samples, die mit Besen oder Sonstigem aufgenommen wurden, waren nicht innerhalb des Evaluierungssets vorhanden. Folgende Ergebnisse konnten in dieser Konfiguration erzielt werden:

	recall	**precision**	**F-measure**
Bassdrum	70.3%	76.0%	73.0%
Snaredrum	60.9%	79.2%	68.9%
Overheads	51.7%	61.5%	56.2%

An den selben Modellen wurde ein weiterer Test, jedoch mit einer geringeren Anzahl an Files von Drummer 2 durchgeführt. In diesem Fall wurden 11 Exzerpte evaluiert, wobei von jedem Genre jeweils ein komplexes sowie ein simples Sample verwendet wurde. Die Geschwindigkeit aller Test-Exzerpte war medium. Folgende Ergebnisse wurden erzielt:

	recall	**precision**	**F-measure**
Bassdrum	78.5%	79.6%	79.1%
Snaredrum	62.8.9%	78.9%	70.0%
Overheads	52.0%	64.1%	57.4%

Wie aus dem zweiten Test ersichtlich ist, performt das System besser, wenn sich

lediglich Songs in moderatem Tempo im Evaluierungsset befinden. Ausserdem wurden in den 11 Songs „ausgefallene“ Genres wie zum Beispiel Afro (ein Stil in dem so gut wie keine hörbaren Snare-Schläge stattfinden) weg gelassen.

Wurde die Toleranzgrenze der erkannten Onsets testweise von 4 auf 3 Samples zurückgesetzt, so konnte man ebenfalls eine Verschlechterung der Performance verzeichnen:

	recall	**precision**	**F-measure**
Bassdrum	69.2%	70.2%	69.7%
Snaredrum	53.5%	67.2%	59.6%
Overheads	43.4%	53.4%	47.8%

In diesem Test musste ein Onset also innerhalb eines ca. 70ms großen Zeitfensters liegen, um als richtig eingestuft zu werden.

Anzumerken ist hier, dass ein direkter Vergleich mit den in Kapitel 3 beschriebenen Systemen nicht möglich ist, da ein einheitliches Testszenario mit einheitlichen Bedingungen nicht vorhanden ist. Auch gibt es Unterschiede in der Auswahl der Testdaten, da diese zum Teil Real-World-Audio-Samples, zum Teil jedoch synthetisch generierte Dateien beinhalten.

Als Beispiel-Szenario soll hier die Evaluierung eines Exzerptes aus dem Bereich Hardrock veranschaulicht werden:

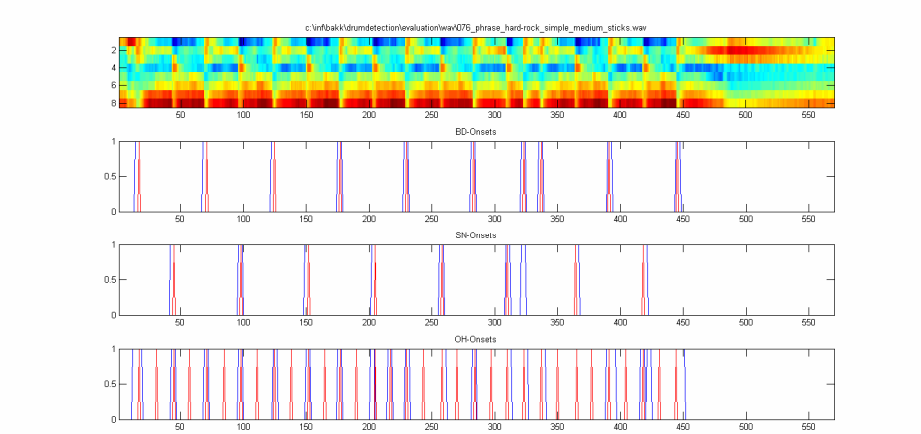

Abb. 26. Evaluierung des Systems an Hardrock-Sample: Spektrogramm (oben), eigentliche und erkannte BD-Onsets (2. v. o.), eigentliche und erkannte SN-Onsets (3. v. o.), eigentliche und erkannte OH-Onsets (unten). Gating-Fkt. in blauer Farbe, Anschläge in roter Farbe.

Wie aus Beispielszenario 1 ersichtlich wird, wird ein Großteil der Anschläge erkannt. Das hier dargestellte Exzerpt ist eine simple Hardrock-Phrase mit Tempo medium. Im nachfolgenden Beispielszenario 2, bei dem versucht wurde, eine Afro-Phrase zu klassifizieren, ist die Anzahl der richtig erkannten Onsets um ein Vielfaches geringer. Als Grund für die schlechtere Performance ist wohl die hohe Anzahl an Ghostnotes zu nennen. Ghostnotes sind marginal erfolgende Anschläge mit sehr niedriger Lautstärke. In diesem Sample kommen sie vor allem auf der Snare sehr häufig vor. Auch das höhere Tempo der Anschläge wirkt sich wohl negativ auf die Performance des Systems aus.

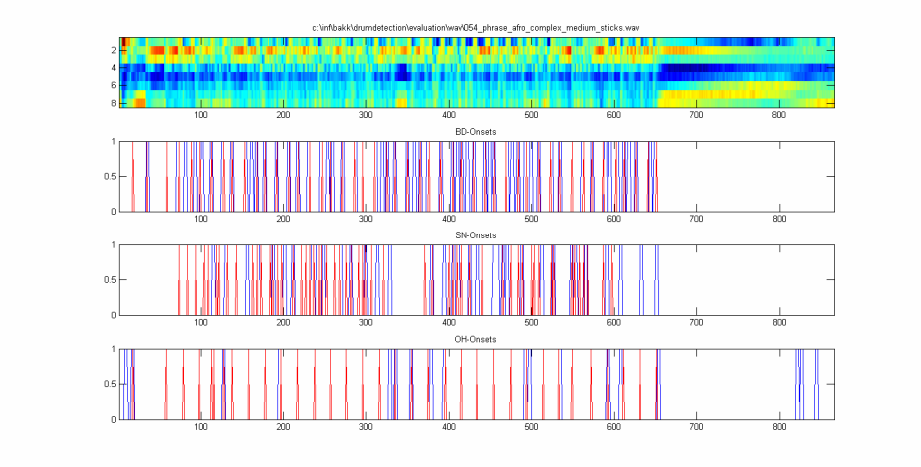

Abb. 27. Evaluierung des Systems an Afro-Sample: Spektrogramm (oben), eigentliche und erkannte BD-Onsets (2. v. o.), eigentliche und erkannte SN-Onsets (3. v. o.), eigentliche und erkannte OH-Onsets (unten). Gating-Fkt. in blauer Farbe, Anschläge in roter Farbe.

Es ist ausserdem möglich, dass System an „Real-World-Audio“-Samples zu evaluieren. Da jedoch in sämtlichen Trainingsdaten keine Begleitmusik vorhanden ist, sind auch die Resultate entsprechend dürftig. Im folgenden der Versuch einer Klassifizierung eines Song-Exzerptes mit musikalischer Begleitung.

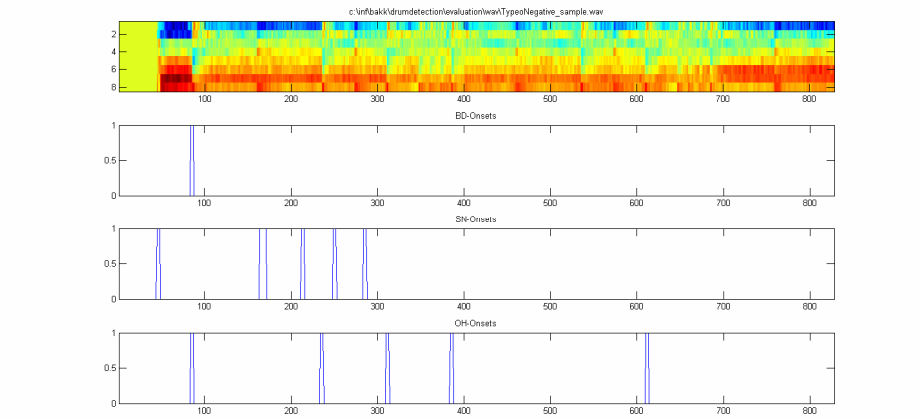

Abb. 28. Evaluierung des Systems an Audio-Sample mit Drums und Begleitmusik: Spektrogramm (oben), eigentliche und erkannte BD-Onsets (2. v. o.), eigentliche und erkannte SN-Onsets (3. v. o.), eigentliche und erkannte OH-Onsets (unten). Gating-Fkt. in blauer Farbe, Anschläge in roter Farbe.

Wie aus Abbildung 28 ersichtlich werden nur wenige Snare-Onsets erkannt, wobei die erkannten Anschläge meistens auch falsch sind. Zu Beginn wird beispielsweise eine Snare vermutet, obwohl hier eigentlich die Begleitmusik einsetzt.

4.10 Verbesserungen

Obwohl das System an reinen Drum-Sounds recht passable Ergebnisse erzielt, gibt es eine Vielzahl an Verbesserungen, die in Zukunft durchgeführt werden können. Zum ersten sollte es möglich sein, Drum-Sounds in Begleitung mit harmonischen Instrumenten zu klassifizieren. Dies könnte man erreichen, in dem man ähnliche Ansätze wie jene von Richard und Gillet oder Fitzgerald aus Kapitel 3 bzw. 5 verfolgt. Auf eine Trennung von harmonischen und stochastischen Teilen könnte man eventuell zur Gänze verzichten, indem man mit einer vielfältigen, entsprechend großen Trainings-Datenbank trainiert, die eine breite Palette musikalischer Genres

und neben reinen Drum-Sounds auch bereits transkribierte Samples mit harmonischer Begleitung beinhaltet.

Es ist auch zielführend, eine genauere Erkennung der Onsets anzustreben, also den Toleranzbereich der Ausgabe-Gate-Funktionen zu verringern, ohne jedoch die Robustheit des Systems zu vermindern.

Zu guter letzt wäre auch eine gänzliche Lautstärke-Unabhängigkeit wünschenswert. Zum einen, weil sehr leise gespielte Ghost-Notes nur schwer zu erkennen sind, zum anderen weil jedes Musikstück, in verschiedener Lautstärke gemischt wurde und auch die Lautstärke-Verhältnisse der einzelnen Instrumente zueinander je nach Genre, Geschmack sowie Fähigkeit des Ton-Ingenieurs variieren.

5. Zusammenfassung

Im Rahmen dieser Arbeit wurden verschiedene gängige Methoden zur Drum-Transcription näher beleuchtet. Ergebnisse dieser Methoden wurden verglichen und diskutiert und es wurde in weiterer Folge ein eigenes, simples System zur Schlagzeug-Erkennung vorgestellt. Die Funktionalität und der Ablauf dieses Systems wurden erläutert. Zuletzt wurden die Ergebnisse der Evaluierung diskutiert und weitere Verbesserungsvorschläge eingebracht.

6. Literaturverzeichnis

[Badeau, 2002]
Badeau R., Boyer R., David B. (2002) EDS Parametric Modeling and Tracking of Audio Signals. Proceedings of the 5th International Conference on Digital Audio Effects (DAFX 02), Hamburg, Germany.

[Fitzgerald, 2003 – 1]
Fitzgerald D., Lawlor B., Coyle E. (2003) Drum Transcription in the presence of pitched instruments using Prior Subspace Analysis. Irish Signals and Systems Conference, Limerick, Ireland.

[Fitzgerald, 2003 – 2]
Fitzgerald D., Lawlor B., Coyle E. (2003) Prior Subspace Analysis for Drum Transcription. 114th AES convention, Amsterdam, The Netherlands.

[Fitzgerald, 2004]
Fitzgerald D. (2004) Automatic Drum Transcription and Source Separation. Thesis presented to Dublin Institute of Technology, Faculty of Engineering and Faculty of Applied Arts in Dublin, Ireland.

[Goto, 2004]
Yoshii K., Goto M., Okuno H. (2004) Automatic Drum Sound Description for Real-World Music using Template Adaptation and Matching Methods. Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005) in London, England. Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004) in Barcelona, Spain.

[Goto, 2005]
Yoshii K., Goto M., Okuno H. (2005) AdaMast: A Drum Sound Recognizer based on Adaptation and Matching of Spectrogram Templates. Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX 2005) in London, England.

[Richard, 2005 - 1]
Richard G., Gillet O. (2005) Drum Track Transcription of Polyphonic Music Using Noise Subspace Projection. Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005) in London, England.

[Richard, 2005 – 2]
Richard G., Gillet O. (2005) Extraction and Remixing of Drum Tracks from Polyphonic Music Signals. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics in New Paltz, NY.

[Richard, 2006]
Richard G., Gillet O. (2006) ENST-Drums: An extensive Audio-Visual Database for Drum Signals Processing. Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006) in Victoria, Canada.

[Fletcher, 1998]
Fletcher N.H., Rossing T.D. (1998) The Physics of Musical Instruments. Springer, Berlin, 3. Auflage, 2000, Kapitel 12, 18, 20.

[WEKA-Web]
Web-Quelle: http://www.cs.waikato.ac.nz/ml/weka/ vom 27.10.2009

[ARFF-Web] :
Web-Quelle: http://www.cs.waikato.ac.nz/~ml/weka/arff.html vom 27.10.2009