Proseminar Computer und Musik
Strukturelle Analyse von Musik

Imke Haverkämper

26. Februar 2018

1 Musikalische Struktur

Musik zu segmentieren und analysieren ist weitgehend unproblematisch, wenn die zugehörige Notation vorliegt- vor eine Herausforderung wird man erst dann gestellt, wenn nur eine Audioaufnahme zur Verfügung steht. Im Folgenden wird die automatisierte Herangehensweise an dieses Problem beschrieben.

Zunächst ist es allerdings wichtig zu klären, welche Strukturen hierbei untersucht werden sollen. Musik ist per se hierarchisch aufgebaut, wobei einzelne Noten, die durch ihren Klang definiert sind, das niedrigste strukturelle Level bilden. Diese formen in Kombination größere Strukturen wie Sätze und Motive, die wiederum das gröbste strukturelle Level, nämlich das allgemeine Layout eines Musikstücks modellieren. Dieses kann in Abhängigkeit von Genre, Stil, kulturellem Hintergrund und einer Vielzahl weiterer Faktoren variieren. Moderne Popsongs sind beispielsweise einteilbar in Intro, Refrain, Vers und gegebenenfalls Outro. Jenes allgemeine Layout wird als musikalische Struktur bezeichnet und steht im Fokus der Analyse.

In westlichen Musikstücken bezeichnet man diese strukturelle Ebene auch als musikalische Form, die üblicher Weise von Kontrast und Vielfalt geprägt ist. Formal ist hierbei immer zu unterscheiden zwischen einem Musikstück im abstrakten Sinne, also einem Stück in notierter Form und einer Repräsentation, also einer Audioaufzeichnung des Stückes. Im Allgemeinen bezeichnet man einen Abschnitt einer solchen Audiodatei als Segment und einen Abschnitt aus der Notation eines Musikstückes bezeichnet man als Part. Der Begriff Sektion kann für Parts verwendet werden, ebenso gut allerdings auch für ein Segment. Bei einer strukturellen Analyse werden Parts gewöhnlich mit Großbuchstaben *A, B, C*, ... in Reihenfolge ihres Auftretens benannt. Wiederholungen eines Parts werden mit gleichem Buchstaben und aufsteigenden Indizes versehen. Es kann vorkommen, dass ein Part

nicht eindeutig abgrenzbar ist, bzw. aus mehreren kleineren Parts bestehen kann. Diese kleineren Subparts werden dann nach dem selben Prinzip mit Kleinbuchstaben und Indizes versehen. Ziel der automatisierten Analyse ist es, den Segmenten der Audiodatei diese abstrakten Parts zuzuordnen. [1, S. 167, S. 172]

2 Analyse- und Segmentierungsverfahren

Um die Audiorepräsentation eines Musikstücks verarbeiten zu können, muss zunächst eine Segmentierung vorgenommen werden. Dadurch wird die Aufnahme in kleinere, aussagekräftigere Einheiten unterteilt, die leichter zu analysieren sind, als die gesamte Datei. Ein Segment ist dabei ein ununterbrochenes Intervall, das von einem eindeutigen Start- und einem Endpunkt begrenzt wird. Segmentiert werden kann zum Beispiel auf Grundlage von Tempo, Tonart oder Timbre.

Daraufhin wird nach Beziehungen zwischen Segmenten gesucht, um Segmentgruppen zu bilden. Am Beispiel eines aktuellen Popsongs könnte eine dieser Gruppen alle Segmente eines Stückes enthalten, die Wiederholungen des Refrains darstellen. Ebenso wird erforscht, welcher zeitliche Abschnitt einer Aufnahme welcher strukturellen Einheit entspricht. Eine solche Analyse kann mithilfe verschiedener Methoden durchgeführt werden, die sich in drei Kategorien einteilen lassen:

- **Repetition-basiert:** Suche nach sich wiederholenden Mustern.
- **Homogeneity-basiert:** Suche nach Eigenschaften, die konstant bleiben.
- **Novelty-basiert:** Suche nach Eigenschaften, die sich ändern.

Hierbei ist zu erwähnen, dass homogeneity-basierte und novelty-basierte Ansätze beide eine Eigenschaft im Hinblick auf Veränderung beobachten, dabei unterscheidet sich lediglich die jeweilige Interpretation der Ergebnisse. Welche speziellen Verfahren sich zur Analyse und Segmentierung eines bestimmten Stückes eignen, muss unter Betrachtung seiner individuellen Struktur und der musikalischen Eigenschaften der Tonaufnahme entschieden werden. Im Folgenden liegt der Schwerpunkt auf dem repetition-basierten und homogeneity-basierten Verfahren mittels Self Similarity Matrizen.[1, S. 170-172]

3 Self Similarity Matrizen

Das Erstellen einer Self Similarity Matrix (SSM) ermöglicht das Visualisieren von Beziehungen zwischen einzelnen Segmenten. Damit eine solche SSM generiert werden kann, muss die zu analysierende Sequenz vorab in eine geeignete Darstellungs-

form bezüglich einer musikalischen Eigenschaft umgewandelt werden. Hierbei bietet es sich an, ein Chroma Feature zu benutzen.

Dann wird jedes Element dieser Sequenz mit jedem anderen darin liegenden verglichen und anhand eines Similarity Measures, d.h. eines Gleichheitsmaßes, in Relation gesetzt. Daraus ergibt sich bei einer Anzahl von n Elementen eine Matrix der Größe $n \times n$. Aus Gründen der Einfachheit kann ohne Beschränkung der Allgemeinheit angenommen werden, dass der Raum, in dem die Elemente liegen, ein Euklidischer Raum und das Similarity Measure der Betrag des Skalarproduktes zweier Elemente ist. Dadurch ergeben sich Werte zwischen 0 und 1, wobei eine maximale Ähnlichkeit, also eine exakte Übereinstimmung durch den Wert 1 und dementsprechend eine minimale Ähnlichkeit, also keinerlei Übereinstimmung durch den Wert 0 repräsentiert wird. Da ein Element keinem anderen so ähnlich ist, wie sich selbst, hat die Matrix auf ihrer Diagonalen den maximalen Wert 1.

Wie bereits angedeutet, wird im folgenden die Repräsentation mittels Chroma Features verwendet. In diesem Fall symbolisiert eine intensivere Einfärbung einer Zelle einen höheren Similarity-Wert. Zur Veranschaulichung zeigt Abbildung 1 eine SSM, wie sie in idealsten Verhältnissen entstehen würde. Hierbei liegt sowohl senkrecht, als auch waagerecht ein und dieselbe Sequenz an. Spricht man von einem Abschnitt entlang der Horizontalen, so bezeichnet man diesen als Segment α [s:t] mit dem Startpunkt s und Endpunkt t. Steht ein solches α in Beziehung zu einem anderen Segment, so wird dieses Segment entlang der Senkrechten als das von α induzierte Segment bezeichnet und auch π_1 genannt. Analog kann α auch mit π_2 gekennzeichnet werden. [1, S. 178, S. 184]

3.1 Block- und Pfadstrukturen

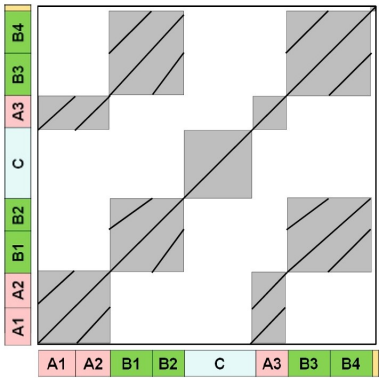

Abbildung 1: Eine ideale SSM [1, S. 180]

Wie bereits erwähnt, visualisiert eine SSM Beziehungen zwischen Segmenten. Solche Beziehungen können anhand der Block- und Pfadstrukturen wie sie in Abbildung 1 zu erkennen sind, sichtbar gemacht werden. Als Block bezeichnet man eine Untermatrix der SSM, deren Zellen alle hohe Similarity-Werte aufweisen. Eine solche Blockstruktur steht hierbei für hohe Homogenität bezüglich einer Eigenschaft innerhalb des entsprechenden Segments. Jeder Block einen zugehörigen Score σ, der sich aus der Summe der Similarity-Werte aller im Block liegenden Zellen ergibt. Blöcke können durch

die Wahl einer passenden Feature Repräsentation hervorgehoben werden.

Feature Repräsentationen entstehen in Abhängigkeit zweier Parameter l und d, wobei l ein Längenparameter ist, der die Featurewerte über l Frames mittelt, und d einen Downsamplingparameter darstellt, der die Featurerate um den Faktor d reduziert und somit die Recheneffizienz weiterer Schritte verbessert. Durch Variation dieser Parameter können zum einen Blöcke näher aneinander rutschen und dadurch deutlicher erkennbar werden, zum anderen können aber Pfade dadurch verschwinden.

Als Pfad wird eine Sequenz von Zellen mit hohen Similarity-Werten bezeichnet, auch ein Pfad hat einen Score σ, der sich analog zum Score eines Blockes durch die Summe aller Similarity-Werte der im Pfad liegenden Zellen berechnen lässt. Pfade repräsentieren Wiederholungen eines Abschnittes und verlaufen, sofern sie exakt sind, entlang der Diagonalen. Da Wiederholungen in der Musik häufig durch Änderung der Instrumentation, des Tempos oder anderen Merkmalen charakterisiert sind, kommt es vor, dass Pfade steiler oder flacher verlaufen. Für das menschliche Auge sind diese häufig trotzdem leicht erkennbar, eine maschinelle Verarbeitung wird dadurch allerdings erschwert. Diese Problematik kann durch die Nutzung von Bildbearbeitungstechniken eingedämmt werden. [1, S. 181-184]

3.2 Glättung von Pfaden mittels Multiple-Filtering-Ansatz

Um Pfade maschinell deutlicher erkennbar zu machen, können verschiedene Filterungstechniken auf die erhaltene SSM angewendet werden. Ein Beispiel dafür ist der Averaging-Filter, auch Low-Pass-Filter genannt. Dieser kann, entlang der Diagonalen angewandt, Strukturen betonen, die parallel zu dieser verlaufen und nicht-diagonale Informationen verwischen. Um Tempovariationen maschinell erfassbar zu machen und keine wichtigen Teile der Pfadstrukturen zu zerstören, die nicht exakt diagonal verlaufen, kann dieser Filter in mehrere Richtungen in der Umgebung der Diagonalen verwendet werden. Dafür wird ein endliches Set Θ erstellt, das aus Tempoparametern θ besteht, die die jeweiligen Tempounterschiede repräsentieren. Diese Strategie bezeichnet man als Multiple-Filtering-Ansatz. Dabei erzeugt jeder Tempoparameter θ eine eigene neue SSM.

In der Realität weicht eine Wiederholung normalerweise nur selten mehr als 50% von der Originalgeschwindigkeit ab. Deshalb bietet es sich an, Θ so zu wählen, dass Abwandlungen von -50% bis +50% durch verhältnismäßig wenige Parameter berücksichtigt werden können. Eine typische Wahl dafür wäre also zum Beispiel $\Theta = \{0.66, 0.81, 1.00, 1.22, 1.50\}$. Da die Glättung mit diesem Vorgehen vorwärts verläuft, kann ein Ausbluten der Pfade nur durch erneutes Anwenden des Fil-

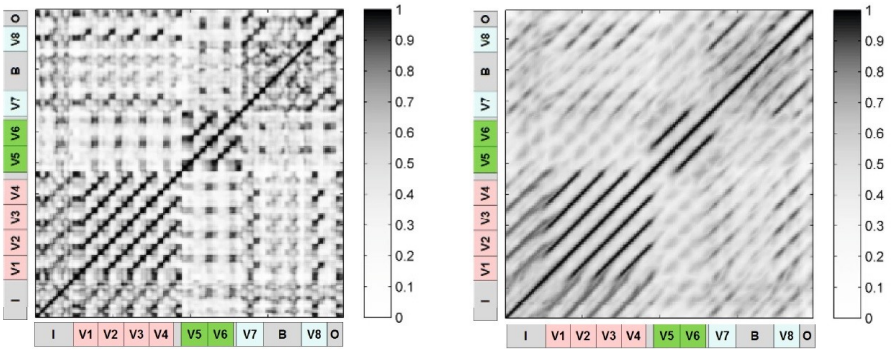

Abbildung 2: Matrix vor und nach Pfadglättung [1, S. 191]

ters in entgegengesetzter Richtung verhindert werden. Auch dieses Filtern ergibt wiederum für jeden Tempoparameter eine neue SSM. All diese Matrizen werden dann miteinander verglichen und durch ihr zellenweises Maximum entsteht eine verbesserte SSM. In Abbildung 2 wird der Effekt dieses Verfahrens anhand einer Self Similarity Matrix vor und nach dem Filtern sichtbar. Während sie vorher unübersichtlich und verrauscht ist, erkennt man danach deutlich sämtliche Pfade. Somit ist offensichtlich, dass es mittels des Multiple-Filtering-Ansatz möglich ist, Tempounterschiede effizient auszugleichen. [1, S. 186-190]

3.3 Transpositionsinvarianz

Wie bereits festgestellt, weichen Wiederholungen in der westlichen Musik nicht nur vom ursprünglichen Tempo ab. Oft wird die Melodie eines Abschnittes in der Tonlage etwas nach oben oder unten verschoben. Diese Veränderungen müssen ebenso wie Tempounterschiede sichtbar gemacht werden, damit eine SSM erfolgreich interpretiert werden kann. Dies ist durch zyklische Verschiebung der Chroma-Werte anhand eines Verschiebungsoperators $\rho : \mathbb{R}^{12} \to \mathbb{R}^{12}$ möglich. Damit ergibt sich durch das Verschieben der Werte mittels ρ um $i \in \mathbb{Z}$ Halbtöne eine i-transpositionierte Self Similarity Matrix bezeichnet mit $\rho^i(S)$, wobei $\rho^{12}(S) = S$. Nach Durchführung dieser Verschiebung erhält man 12 verschiedene Matrizen, aus denen man daraufhin, wie bei der Pfadglättung mittels Multi Filtering auch, durch Vergleichen und Extrahieren der zellenweisen Maxima eine neue, verbesserte SSM generiert.

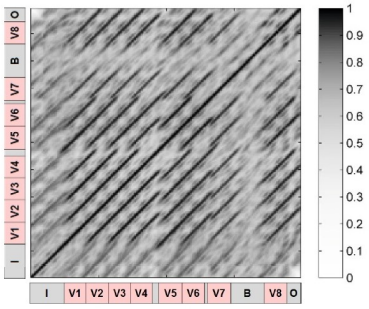

Abbildung 3: Eine S^{TI} [1, S. 191]

Jene Matrix nennt sich **transpositionsinvariante Self Similarity Matrix** $\mathbf{S}^{TI}$. Dieses Verfahren sollte allerdings erst nach Vereinfachung der zu analysierenden SSM durchgeführt werden, denn wie in Abbildung 3 zu sehen ist, erhöht sie den Informationsgehalt enorm und kann dadurch den Einfluss von Störvariablen ungewollt steigern. [1, S. 190-192]

3.4 Schwellenwertverfahren

Das Erkennen von Blöcken und Pfaden bei der automatisierten Analyse einer SSM ist nicht nur durch relative Tempounterschiede erschwert. Diese Matrizen sind meist sehr detailreich und enthalten viele irrelevante Informationen, die eine Analyse verkomplizieren. Deshalb ist es sinnvoll, einige unwichtige Informationen zu entfernen, bevor die SSM weiter verarbeitet wird. Eine Möglichkeit dies zu erreichen ist das Thresholding, bzw. Schwellenwertverfahren. Dabei wird ein Schwellenwert für Similarity-Werte festgelegt, der überschritten werden muss, damit der Wert einer Zelle auch in einer neuen, vereinfachten Matrix beibehalten wird. Das Schwellenwertverfahren kann entweder global oder lokal vorgenommen werden.

Beim **globalen Thresholding** wird ein Schwellenwert τ mit allen Similarity-Werten in der Matrix verglichen und jene, die darunter liegen werden alle auf 0 gesetzt. Ein globaler Threshold kann hierbei zum Beispiel relativ gewählt werden, indem man $\rho \cdot 100\%$ der Zellen mit den höchsten Similarity-Werten beibehält, wobei $\rho \in [0,1]$ der relative Threshold ist. Wenn ein noch intensiverer Kontrast zwischen den Werten von Nöten ist, kann man als nächstes alle Werte über τ auf 1 setzen, um die SSM zu binärisieren. Da diese Vorgehensweise sehr viele Informationen eliminiert, ist es meistens von Vorteil stattdessen eine lineare Skalierung aller Werte von $[\tau, \mu]$, wobei μ der maximale Similarity-Wert innerhalb der Matrix ist, zu [0,1] vorzunehmen. In manchen Fällen kann es sogar sinnvoll sein, alle Werte unterhalb des Schwellenwertes auf einen Parameter δ zu setzen, statt diese auf 0 zu setzen.

Beim **lokalen Thresholding** bleibt der Similarity-Wert einer Zelle genau dann erhalten, wenn er unter den $\rho \cdot 100\%$ höchsten der Zeile und gleichzeitig auch der Spalte liegt, ansonsten wird er auch hier auf 0 gesetzt. Daraus ergibt sich, dass beim Schwellenwertverfahren im Allgemeinen nur die stärksten Signale zuverlässig übernommen werden. Indes können aber auch schwächere Signale relevant sein und durch die Anwendung verloren gehen. Deshalb ist eine adäquate Wahl des Schwellenwertes von enormer Wichtigkeit für den Erfolg oder Misserfolg der automatisierten strukturellen Analyse von Musik. [1, S. 192-194]

Somit wurde die Analyse von musikalischen Strukturen mittels SSM weitläufig erklärt, ohne bisher eine praktische Anwendung zu diskutieren. Den wohl größten Anwendungsbereich einer strukturellen Analyse von Musikstücken stellt das Audio Thumbnailing dar.

4 Audio Thumbnailing mittels Fitness Measure

Ein Audio Thumbnail stellt eine Art Vorschau für ein Musikstück dar. Das ist beispielsweise beim Durchsuchen einer großen Musiksammlung hilfreich, da ein Zuhörer so schnell entscheiden kann, ob das entsprechende Stück seinen Anforderungen entspricht. Deshalb soll ein Thumbnail soll automatisch erstellt werden, indem ein Ausschnitt aus einer Aufnahme ausgesucht wird, der diese am besten repräsentiert. Meist ist der Refrain eines Songs ein guter Kandidat für einen solchen Ausschnitt, weil dieser im Allgemeinen mehrmals wiederholt wird und damit einen großen Teil des Stückes ausmacht. Darum wird mittels Algorithmen oft versucht, eine Sequenz zu finden, die zum einen möglichst kurz ist und zum anderen besonders häufig wiederholt wird. Hierbei ist ein Vorgehen mithilfe von Self Similarity Matrizen, wie es oben beschrieben ist, geeignet.

Um entscheiden zu können, welches der Segmente, die in der SSM sichtbar gemacht wurden, das repräsentativste ist, führt man ein **Fitness Measure**, also ein Eignungsmaß, ein. Dieses Eignungsmaß erklärt einerseits, wie gut ein Ausschnitt den Rest des Stückes beschreibt und andererseits, wie groß der Anteil des Stückes ist, der von diesem Ausschnitt und dessen Wiederholungen abgedeckt wird. Das kann erreicht werden, indem alle Relationen zwischen einem Segment und seinen Wiederholungen betrachtet werden. Hierfür wiederum führt man **Pfadfamilien** ein. Bevor man diese näher beschreiben kann, muss man zunächst eine Segmentfamilie A deifnieren. Diese ist ein Set, das die Segmente α_1 bis α_n enthält, wobei alle Elemente paarweise disjunkt sind. Jede Segmentfamilie hat eine bestimmte Coverage $\gamma(A)$, die angibt, wie groß der zeitliche Umfang der Segmentfamilie ist. Dies erreicht man durch Addition aller Längen der darin liegenden Elemente. Eine Pfadfamilie ist nun ein Set P, das Pfade enthält, die über ein Segment α verlaufen, analog zur Segmentfamilie sind auch hier alle Elemente disjunkt. Auch eine Pfadfamilie hat einen Score σ, der sich mit der Summe aller Scores der darin liegenden Pfade berechnen lässt.

Des weiteren gibt es eine optimale Pfadfamilie P^*, die alle Wiederholungen eines Abschnittes enhält und somit einen maximalen Score hat. Eine Möglichkeit unser Fitness Measure zu wählen wäre demnach der Score dieser optimalen Pfadfamilie. Dieser Score ist allerdings stark abhängig von der Länge der Sequenz und

enthält viele Selbsterklärungen. Deshalb ist es sinnvoll, zuerst die Länge von α zu subtrahieren und den score dann hinsichtlich der Länge der Pfade der Familie zu normalisieren. Dadurch ergibt sich als normalisierten Score

$$\overline{\sigma}(\alpha) := \frac{\sigma(P^*) - |\alpha|}{\sum_k^K = L_k}.$$

Damit gibt $\overline{\sigma}$ intuitiv den durchschnittlichen Score der optimalen Pfadfamilie P^* abzüglich aller Selbsterklärungen an und wird dadurch zu einem längenunabhängigen Vergleichsmaß. Somit ist erklärt, wie ein Segment, das eine Aufnahme möglichst gut beschreibt, ausgewählt werden kann.

Um den zeitlichen Umfang zu bewerten, führt man nun ein **Coverage Measure** für ein gegebenes Segment α ein. Hierfür wird die durch die optimale Pfadfamilie induzierte Segmentfamilie A^* und die zugehörige Coverage $\gamma(A^*)$ betrachtet. Analog zum Score wird auch diese normalisiert zu

$$\overline{\gamma}(\alpha) := \frac{\gamma(A^*) - |\alpha|}{N}.$$

Der Wert des Coverage Measures gibt also das Verhältnis der Länge der gesamten Segmentfamilie zur absoluten Länge des Stückes abzüglich der Selbsterklärungen wieder.

Wie bereits erwähnt strebt man an, dass das Fitness Measure sowohl einen hohen Score, als auch eine hohe Coverage hat, was schwierig zu kombinieren ist, da kürzere Segmente häufiger einen höheren Score, aber eine geringere Coverage haben. Umgekehrt verhält es sich bei längeren Segmenten. Dieser Effekt kann durch das Mitteln der beiden Werte ausgeglichen werden. Deshalb definiert man ihr harmonisches Mittel als **Fitness Measure** φ mit

$$\varphi(\alpha) := 2 \cdot \frac{\overline{\sigma}(\alpha) \cdot \overline{\gamma}(\alpha)}{\overline{\sigma}(\alpha) + \overline{\gamma}(\alpha)},$$

wobei kürzere Segmente leicht bevorzugt werden. Nun wird der **Audio Thumbnail** definiert als das Segment maximaler Fitness. Darüber hinaus kann es Sinn ergeben, eine Mindestlänge für den Thumbnail einzuführen, wenn man den Einfluss von Störfaktoren in der zugrundeliegenden SSM reduzieren möchte. [1, S. 195-202]

5 Zusammenfassung und Kritik der Methodik

Bisher wurde hier ein beispielhafter Verlauf einer strukturellen Analyse von Beginn bis zur Anwendung beschrieben. Dabei sollte die Repräsentation zunächst segmentiert, Self Similarity Matrizen generiert und diese anhand von Multi-Filtering-Ansatz, Transpositionsinvarianz und Schwellenwertverfahren bearbeitet werden, um Relationen zwischen Segmenten erkennen zu können. Zuletzt sollte schließlich ein Audiothumbnail berechnet werden. An dieser Stelle sollte gesagt sein, dass die hier genannten Verfahren nicht die einzig erprobten Möglichkeiten sind, eine strukturelle Analyse zu gestalten. All diese Verfahren haben allerdings einen gemeinsamen Nachteil: Sie manipulieren die originalen Daten der Repräsentation und sind dabei anfällig für Fehler, die das Ergebnis maßgeblich beeinflussen und verfälschen können.

Hierbei ist anzumerken, dass selbst die Entscheidung über Erfolg oder Misserfolg einer automatisierten Analyse ein Problem darstellt, das hier bislang keine Erwähnung fand. Ein intuitiver Ansatz die Ergebnisse auf Richtigkeit zu prüfen ist es, diese mit einer von einem Experten manuell erstellten Analyse zu vergleichen. Dabei wird vernachlässigt, dass Musik per se nicht eindeutig ist, es also durchaus vorkommen kann, dass zwei Experten bei ihren Untersuchungen zu unterschiedlichen Ergebnissen kommen. Einer könnte ein Segment als Einheit beschreiben, während ein anderer dieses in mehrere kleine Subsegmente separieren würde. Die Entscheidung, ob ein automatisiertes Verfahren gelungen ist oder nicht, hängt also direkt davon ab, welche Präferenzen bezüglich der Strukturierung der Experte hat, der die Vergleichsbasis erstellt hat. Damit ist eine objektive Untersuchung auf Richtigkeit erschwert. Zusammenfassend ist demnach zu sagen, dass das Ergebnis einer automatisierten strukturellen Analyse sensibel für Verarbeitungsfehler und, ebenso wie bei einem manuellen Vorgehen, nicht eindeutig ist.

Literatur

[1] M Müller. *Fundamentals of Music Processing*. 2015.