HTWK Hochschule für Technik, Wirtschaft und Kultur Leipzig

Masterprojekt

im Studiengang Informatik

Thema: Entwicklung und Evaluation von Ähnlichkeitsmerkmalen für eine Organisation von Audiodateien

eingereicht von: Lars Hamann | lars.hamann@stud.htwk-leipzig.de | hamann671@protonmail.com |

eingereicht am: 30.09.2020

Betreuer: Prof. Dr. rer. nat. Johannes Waldmann

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Themenvorstellung

Die Organisation von Audiodateien beziehungsweise Audiosamples ist für jeden Computermusiker eine nicht endenden wollende Aufgabe. Nach welchen Kriterien sollen Samples organisiert werden? Sollen sie nach Hersteller und Erzeuger einer Samplegruppe (z.B. TR-808), nach Art eines Audiosamples (z.B. Kick-Drum) oder gar nach Gefühl, was man dabei verspürt, unterteilt werden? Es wäre doch praktisch, wenn Audiosamples automatisch nach Ähnlichkeit organisiert werden können oder automatisch ähnliche Audiosamples, anhand eines bereits ausgesuchten, gefunden werden können.

Dieses Problem ist der Kern dieser Ausarbeitung. Die Organisation einer unorganisierten bzw. unübersichtlichen Sammlung von Audiodateien. Dabei werden verschiedene Ähnlichkeitsmerkmale zwischen Audiodateien untersucht. Ferner werden Ansätze aus der Bildverarbeitung auf Audio übertragen. Ähnlichkeit wird in dem Sinne definiert, dass der Abstand zweier Audiodateien anhand ihrer Merkmale möglichst klein ist.

Als Sammlung von unorganisierten Audiodateien wird die *Tidal Cycles Super Dirt-Sample*-Bibliothek verwendet [1]. Es handelt sich dabei um Audiosamples für die digitale Musikproduktion mit Tidal Cycles, einer Live-Coding Umgebung. Eine zweidimensionale Landkarte soll die Ähnlichkeit visualisieren, wobei auch eine Relation von Abstand auf der Landkarte und Abstand der Ähnlichkeit wünschenswert ist. Es könnten damit ähnliche Audiosamples gefunden werden, welche davor durch den unorganisierten Datensatz nicht ersichtlich sind oder bisher nicht verwendete Audiosamples entdeckt werden.

Zunächst werden Darstellungen und Transformationen von zeitdiskreten Audiosignalen vorgestellt um anschließend aus den verschiedenen Darstellungsformen Merkmale extrahieren zu können. Es werden anschließend die Organisationsvoraussetzungen für den *Super Dirt Sample*-Datensatz definiert und der Analyseprozess der zuvor erläuterten Merkmale erklärt. Zuletzt werden die Ergebnisse vorgestellt und visualisiert, gefolgt von einer abschließenden Zusammenfassung und Bewertung des Projekts.

2. Digitale Repräsentation von Audiosignalen

Um aus digitalen Audiosignalen Merkmale zu extrahieren, muss sich zunächst mit den verschiedenen digitalen Repräsentationen und ihren Transformationen beschäftigt werden. Diese sind essentiell, um zum Einen Merkmale zu extrahieren und zum Anderen neue Merkmale zu entwickeln.
Generell handelt es sich bei einem digital kodierten Audiosignal um eine zeit- und wertdiskrete Funktion, die einen Zeitpunkt auf einen Amplitudenwert abbildet.

$$f : \{1, 2, ..., N\}^k \rightarrow \{0, 1, ..., m\}^k$$

Das gängige Audioformat WAVE hat oft eine Samplerate (Abtastrate) SR von 44100 Hz, was bei einer gegebenen Länge T der Audiodatei zu $N = SR * T$ Abtastwerten (Samples) führt. Ein Sample ist in diesem Bezug ein einzelner Wert des Zeitsignals. Ein Sample, als kurze Audiodatei wird in dieser Arbeit Audiosample genannt. Die Anzahl an diskreten Amplitudenwerten beträgt meist $m = 2^{16} - 1$. Die resultierende zeitliche Auflösung entspricht $\Delta t = \frac{1}{SR}$. Bei der soeben genannten Samplerate wird etwa alle 0.023 ms ein Wert abgespeichert. Die Anzahl an Audiokanälen wird durch k repräsentiert und ist bei einem monophonen Signal 1 und bei einem stereophonen Signal 2 [2]. Die Wahl von 44100 Abtastwerte pro Sekunde ist zum Einen auf das Abtasttheorem und zum Anderen auf den beschränkten, vom Menschen wahrgenommenen Frequenzbereich zurückzuführen. Menschen können Schallwellen von etwa 16 Hz bis 20 kHz auditiv wahrnehmen [3, S.37]. Um in Kapitel 2.1 das Signal verlustfrei in den Frequenzbereich zu transformieren, muss die Abtastrate SR größer als das Doppelte der maximalen Frequenz f_{max} des Signals sein:

$$SR > 2 * f_{max}$$

Das diskrete Audiosignal kann zur Visualisierung durch ein Zeit-Amplituden-Diagramm werden (s. Abb: 1). [3]

2.1. Diskretes Frequenzspektrum

Eine weitere Darstellung ist das Amplitudenspektrum oder meist Spektrum genannt. Dazu wird ein zeitdiskretes Signal mittels einer Diskreten Fourier-Transformation (DFT) in den komplexen Frequenzraum transformiert. Zur zweidimensionalen Darstellung wird anschließend der Betrag zu jedem komplexen Amplitudenwert berechnet. Die Diskrete Fourier-Transformation ist beschrieben durch:

$$\underline{X}(k) = \sum_{n=0}^{N-1} x(n) * exp(-i * 2\pi * \frac{k * n}{SR}) \mid k \in K, K \in \mathbb{N}$$

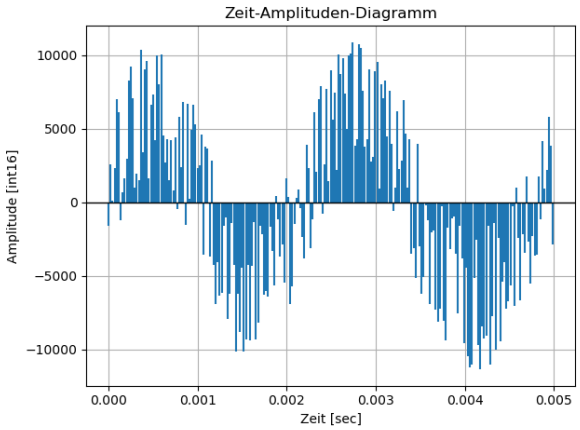


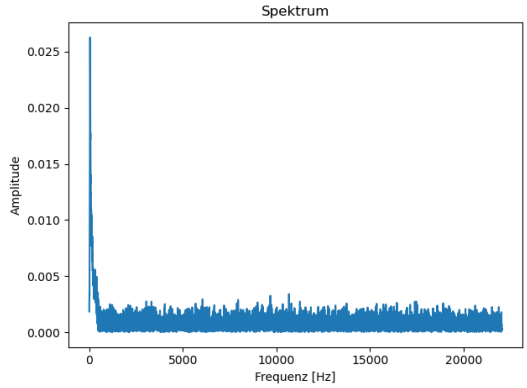
Abbildung 1: Zeit-Amplitudendiagramm einer Snare-Drum (Abgeschnitten ab 5 ms)

Dabei entspricht N der Anzahl an Samples und n der des Zählindexes. Das Amplitudenspektrum besteht aus K Spektrallinien von 0 bis $K-1$. Diese repräsentieren jeweils eine Frequenz, wobei k selbst keine Frequenz ist, sondern einem Fourier-Koeffizienten entspricht. Für eine verlustfreie Rücktransformation in den zeitdiskreten Bereich wird $K = N$ vorausgesetzt. Die Frequenzauflösung, also der Abstand zwischen den diskreten Spektrallinen beträgt $\Delta f = \frac{SR}{N}$. Bei einem Signal mit Länge 8000 Samples und der Samplerate 44100 Hz ist die Frequenzauflösung $\approx 5.51Hz$. Die Null-Frequenz entspricht keiner realen Frequenz, sondern ist die Summe des zeitdiskreten Signals. Die Diskrete Fourier-Transformierte ist periodisch und symmetrisch an der Spektralkomponente $\frac{f_S}{2}$. Das heißt für alle Fourier-Koeffizienten k gilt

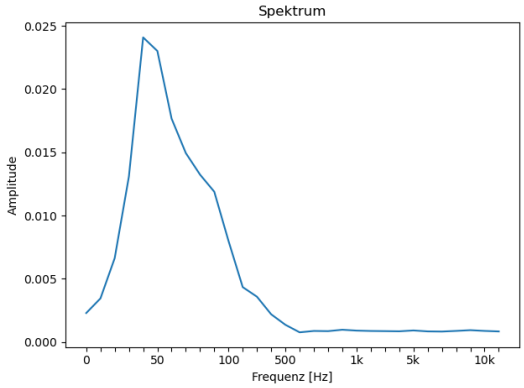
$$k \in \{\lfloor \frac{f_S}{2} \rfloor + 1, ..., f_S\} \mid X(k) = \overline{\underline{X}(f_S - k)}$$

Von den K Frequenzen sind also nur $\frac{K}{2}$ Frequenzen „unabhängig“. Die entsprechenden Paarungen verhalten sich konjugiert komplex zueinander. Für eine korrekte Darstellung des komplexen Frequenzraums, werden in der Regel zwei Spektren gezeichnet. Das Amplitudenspektrum und das Phasenspektrum. In dieser Arbeit, liegt der Fokus auf der vereinfachte Form, also das reine Amplitudenspektrum. Dazu wird der Betrag $X(k) = |\underline{X}(k)|$ aus der Fourier-Transformierten berechnet und dadurch die Dimensionalität reduziert.
[4, S. 40-57], [5, S.279-325]

In Abb. 2a ist der Frequenzverlauf der in Abb. 1 angeschnittenen Snare-Drum zu sehen. Außer einem deutlichen Ausschlag nahe der Null-Frequenz lässt sich allerdings nicht wirklich viel erkennen. Eine bessere Darstellung liefert eine logarithmische Skalierung des Frequenzbereichs (siehe Abb. 2b). Dadurch ist der Ausschlag besser einzuordnen



(a) lineares Spektrum



(b) Logarithmisches Spektrum

Abbildung 2: Spektren der Snare-Drum

- zwischen 40 und 50 Hz. Auf logarithmische Frequenzdarstellung wird in Kapitel 2.4 näher eingegangen.

2.2. Spektrogramm

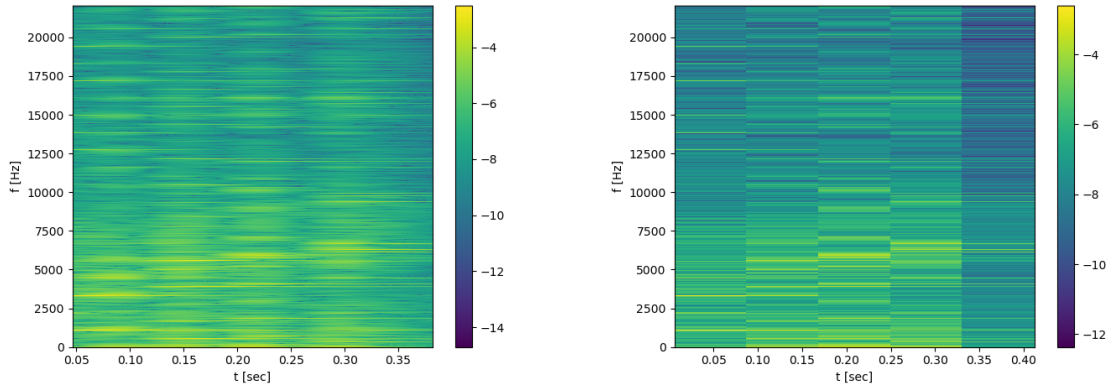
Der Nachteil bei der Transformation eines zeitdiskreten Audiosignals in den Frequenzbereich ist der Verlust der zeitlichen Auflösung. Bei kurzen Audiodateien mag es nebensächlich sein, doch erklingen mehrere Töne oder Geräusche nacheinander, ist eine zeitlose Frequenzdarstellung eher suboptimal. Abhilfe schafft ein Spektrogramm, welches durch die Diskrete Kurzzeit-Fourier-Transformation (STFT) berechnet wird. Dabei wird ein Spektrum in $m \in \mathbb{Z}$ gleich große Zeitabschnitt unterteilt und für jeden Zeitabschnitt die DFT berechnet. Die Unterteilung entspricht dabei einer Faltung des diskreten Zeitsignals mit einer diskreten, symmetrischen und periodischen Fensterfunktion. Für jedes Fenster werden dann die Fourier-Koeffizienten $k \in [0, K]$ berechnet. Dabei sollte eine Fensterfunktion $w : [0, N-1] \to \mathbb{N}$ mit Fensterlänge $N \in \mathbb{N}$ mindestens folgende Eigenschaften besitzen: Periodisch, symmetrisch und diskret. Für alle n innerhalb eines Fensters gilt: $w(n) = w(N-n)$. Die wohl einfachste Funktion ist dabei ein Rechteckfenster, was zu jedem Zeitpunkt n innerhalb des Fensters die Identität ausgibt.

Die Fensterfunktion wird zur Berechnung jedes Zeitfensters m verschoben und mit dem Zeitsignal multipliziert. Dabei bietet es sich an, die Schrittlänge durch einen festen Faktor $H \in \mathbb{R}$ (Sprunggröße) zu erweitern. Dies führt zu einer höheren Fehlerreduktion bei den Transformationen. Zunächst beschreiben wir die Diskrete Kurzzeit-Fourier-

Transformation mit:

$$\chi(m,k) = \sum_{n=0}^{N-1} x(n + m * H) * \hat{w}(n) * exp(-2\pi * i * \frac{kn}{N})$$

[4, S 40-57, S.93-109], [6, S. 7-19]
Die Wahl der Zeit- und Frequenzauflösung, also H und K können nun relativ unabhängig von einander gewählt werden. In der Regel wird aber H stark mit N gekoppelt und das Verhältnis zwischen N und K ist wieder durch das Abtasttheorem begrenzt. Theoretisch könnte man dennoch eine Fenstersprungweite von $H = 3$ wählen und mit einer Fenstergröße von 44100 Samples eine 1 Hz breite Frequenzauflösung mit $\Delta t = \frac{H}{SR} = \frac{3}{44100} \approx 68\mu$s erreichen [4, S. 54 - 56]. Für jedes der 14700 Spektren pro Sekunde wird aber die gesamte Fensterbreite von 1s mit einbezogen. In Abb. 3 sind zwei Spektren mit unterschiedlicher Fensterschrittweite dargestellt. Auf der linken Seite ist die Schrittweite nahe an dem soeben konstruierten Beispiel. Es sind lediglich 6 Samples pro Zeitpunkt m die nicht mit dem nächsten Zeiteinheit überlappen. Auf der rechten Seite ist die Sprungweite gleich der Fenstergröße. Dort ist sehr gut die Blockbildung im Zeitbereich zu erkennen.



(a) H = 6 und N = 4096 (b) H = 0 und N = 4096

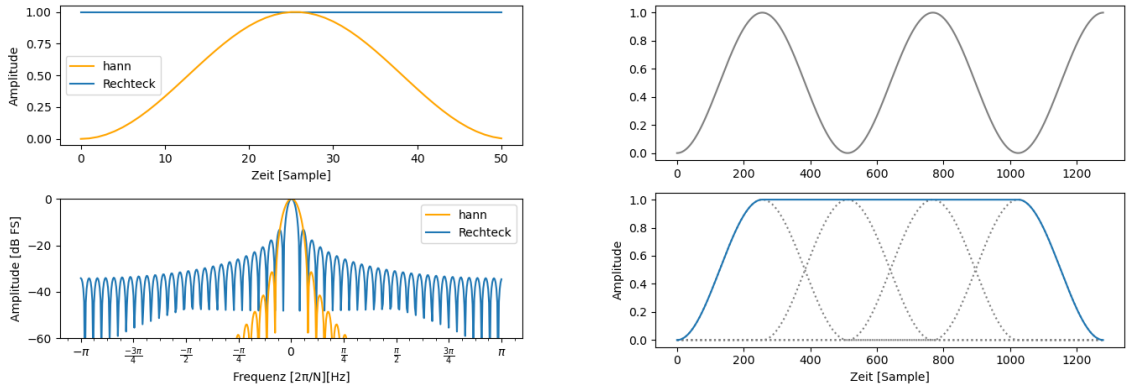
Abbildung 3: Spektrogramme

2.3. Fensterfunktionen

Als Beispiel für eine Fensterfunktion wurde bereits das Rechteckfenster erwähnt und sinnvolle Eigenschaften an Fensterfunktionen gerichtet. Dabei ist das Rechteckfenster freilich nicht die einzige Fensterfunktion und auch nicht für alle Situationen gleich gut geeignet. Ein populäres Beispiel ist das Hann-Fenster - beschrieben durch

$$w(n) = \alpha + (1 - \alpha) * cos\left[\left(\frac{2\pi}{N}\right) * n\right] \quad | \, \alpha = 0.5$$

Ein wesentlicher Unterschied der beiden Funktionen ist in Abb: 4a dargestellt. Im oberen Bereich sind die zeitlichen Funktionen abgebildet und in der unteren Hälfte ihre Fourier-Transformierte. Dort ist zu entnehmen wie stark der „Leckeffekt“ auftritt. Also wie stark der Wert einer Frequenz die anderen beeinflusst. Dabei streut ein Rechecktfenster beim Hauptausschlag gar nicht, allerdings streuen die Nebenkeulen sehr stark. Beim Hann-Fenster sind die Nebenkeulen stark gedämpft, wofür dann die Hauptkeule streut. Es muss letztendlich ein Kompromiss gefunden werden, wobei das Hann-Fenster in den meisten Fällen diesen Kompromiss trifft. [7, S. 204-208], [8, S. 258-285]



(a) Vergleich Rechteck und Hann-Fenster

(b) Beeinflussung des Hann-Fensters

Abbildung 4: Beeinflussung der Fensterfunktion

Der eigentliche Grund, warum überhaupt Fensterfunktionen verwendet werden ist der folgende: Die DFT ist eigentlich für zeitdiskrete und periodische Signale ausgelegt. Bei der normalen DFT, also dem analysieren eines kompletten Signals, kann, zumindest gedanklich, das gleiche Signal (oder auch Lied) nach Abschluss wieder von vorne beginnen. Zumindest in der Musik wurde zwischen einzelnen Liedern auf einer CD oft eine kurze Stille eingefügt. Eine ähnliche Idee liegen den Fensterfunktionen zu Grunde. Es wird damit versucht, ein nicht periodisches Signal in ein Periodisches zu verwandeln. Dadurch wird das Problem mit suboptimalen Abtastzeitpunkten und nicht periodischen Signalen verbessert; doch nun wird in periodischen Abständen Stille dem Zeitsignal hinzugefügt. Um dieses Problem wiederum zu entwerten, wird bei der STFT eine flexible Fensterschrittweite gewählt. Dadurch fließt jedes Sample im zeitdiskreten Signal mit unterschiedlichen Gewichtungen bei der DFT ein. Die Summe der Gewichte für ein Sample kann die Repräsentation im Frequenzbereich verbessern. In Abbildung 4b ist ein Beispiel für die Summe der Gewichte jedes Samples mit dem Hann-Fenster dargestellt. In der oberen Hälfte ist die Schrittweite gleich der Fenstergröße und in der unteren Hälfte $H = \frac{N}{2}$. Dabei ist zu erkennen, das im unteren Teil für die meisten Samples ein Summengewicht von 1 vorliegt und im oberen Bereich viele Samples gedämpft werden.[6, S. 7-9]

2.4. Weitere Reduktions- und Skalierungsmöglichkeiten

In Abb. 2 wurde bereits ein Spektrogramm mit einer linearen und einer logarithmischen Skalierung berechnet und dargestellt. Dabei ist die lineare Darstellung nicht die Optimale für den Menschen. Verschiedene Frequenzgruppen, aus dem etwa 20 kHz wahrnehmbaren Frequenzbereich werden unterschiedlich stark wahrgenommen. Dies betrifft sowohl Lautheit, Unterscheidbarkeit, Wahrnehmbarkeit, Lautstärke, als auch die Phase einer Frequenz sowie komplexere Klänge. Es gibt diverse Methoden um eine bessere Abbildung sowohl des Frequenzbereichs als auch des Wertebereichs für den Menschen zu schaffen. [3, S.27-45]

2.4.1. Bark

Durch die Bark-Skala wird, je nach Literatur, das Spektrum in 24 - 28 nicht überlappende Frequenzbänder unterteilt. Diese haben sich aus verschiedenen psychoakustischen Untersuchungen herauskristallisiert. Es wurde versucht, das wahrgenommene Frequenzspektrum in disjunkte Gruppen zu unterteilen, welche vom Menschen unterscheidbare Eigenschaften besitzen. So finden wir beispielsweise

> die Verdeckung reiner Töne durch schmalbandiges Rauschen, die Sensitivität gegenüber der Phasenverschiebung und die Integration der Schallintensität von Klängen oder Schmallbandrauschen [...] ausschließlich innerhalb von Frequenzgruppen. [3, S.41]

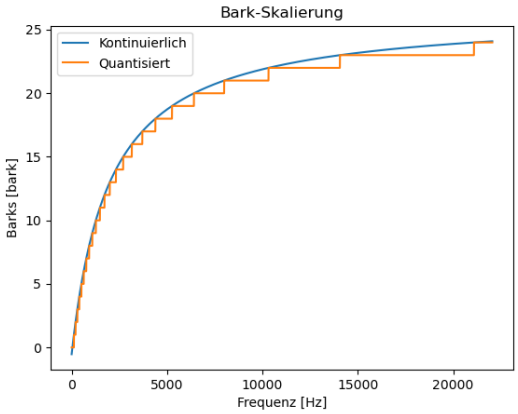
Nach Zwicker nach [9] ist die Umrechnung von Hz zu Bark beschrieben durch:

$$z = 13 * atn(0.00076 * f) + 3.5 * atn((\frac{f}{7500})^2)$$

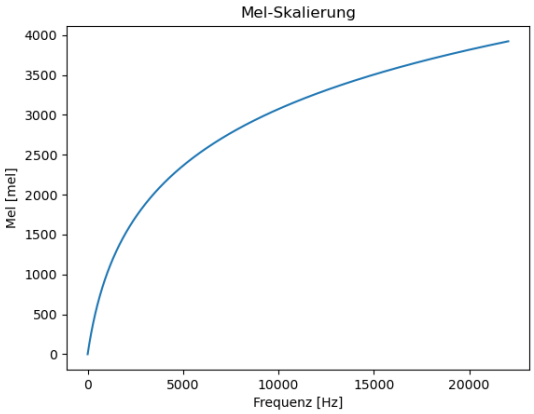
Es findet sich in der Literatur dazu viele verschiedene Formulierungen und Optimierungen mit etwaigen unbekannten Rundungen und anderen Unbekannten [9],[10]. Die in Abb. 5a dargestellte Frequenz-Bark-Relation resultiert aus eben genannter Formel (gerundet auf 10^0). Für die Zuordnung in die 24 disjunkte Gruppen, werden die Werte auf die nächste ganze Zahl abgerundet, was auch mit einer Rechteckfensterfunktion im Frequenzbereich umgesetzt werden kann.

2.4.2. Mel

Eine weitere psychoakustische Skalierung bzw Einteilung der Frequenzen durch ihre wahrgenommene Tonhöhenänderung ist die Mel-Skalierung. Der Versuchsaufbau ist dem der Bark-Skalierung recht ähnlich. Menschen sollen mit Hilfe eines Referenztons eine



(a) Bark-Skalierung mit aus Literatur abweichenden Werten



(b) Mel-Skalierung

Oktave einschätzen. Das Ziel der Mel-Skala ist allerdings nicht das Unterteilen in Frequenzgruppen, sondern $\frac{1}{10}$ Oktave der menschlichen Wahrnehmung zu quantifizieren. Die Mel-Skalierung findet oft Einsatz in der Spracherkennung und verhält sich, ähnlich wie die Bark-Skalierung, bis etwa 700 Hz linear und danach eher logarithmisch. Auch hierbei finden sich mehrere Definitionen, je nachdem mit welchem Ziel diese berechnet werden [11], [12], [13, S.66-70], [14]. Für die Repräsentation in 5b wurde die Formel nach [11, Formel 4] verwendet:

$$\hat{f}_{mel} = 2595 * log_{10}\left(1 + \frac{f_{lin}}{700}\right)$$

Für Merkmalsextraktionen wird die Mel-Skalierung allerdings meist für die Einteilungen in Frequenzbänder unterteilt. Die Einteilung dazu wird mit Hilfe von Filterfunktionen (Fensterfunktionen) bewerkstelligt, wobei die Art des Filters, die Anzahl an Bändern und ihre entsprechenden Grenz- und Zentrumsfrequenzen erneut nicht uniform definiert sind [11].

2.4.3. Decibel Full Scale

Decibel Full Scale ist eine Decibel Skalierung für Amplitudenwerte in digitalen Audiosignalen. Da Decibel selbst eine referenzwertabhängige „Einheit“ ist wurde diese im AES Standard definiert [15]. Diese beträgt generell den maximal möglichen Wert eines Codeworts. Liegen Werte als int16 vor, ist der Referenzwert $P_0 = 32.767$; bei einer float32 Repräsentation der int16 Werte ist $P_0 \approx 1$. Die float32 Darstellung erlaubt eine Darstel-

lung von -1 bis 1 (1 ausgeschlossen). Die Umrechnung in db FS ist entsprechend:

$$\texttt{Verhältnis}_{dBFS} = 20 * log\left(\frac{P_1}{P_0}\right)$$

2.5. Zusammenfassung

Dieses Kapitel hat die wesentlichen Elemente der digitalen Signalverarbeitung beleuchtet, mit dem Ziel Audiodateien für eine Analyse von Merkmalen möglichst sinnvoll in andere Räume zu transformieren. Dabei wurden drei wesentliche Darstellung aufgezeigt:

- zeitdiskrete Darstellung
- frequenzdiskrete Darstellung
- zeit- und frequenzdiskrete Darstellung

Für die Transformationen in die jeweiligen Räume wurden Abhängigkeiten aufgezeigt, welche die Qualität und Güte beeinflussen können. Diese sind:

- Größe des Fensters einer Fensterfunktion
- Wahl der Fensterfunktion
- Sprungweite der Fensterfunktion
- Skalierung des Frequenzbereichs
- Skalierung der Werte

Es gilt entsprechend geeignete Werte in Bezug auf den zu analysierenden Datensatz zu wählen.

3. Ähnlichkeitsmerkmale von Audio

Unter Ähnlichkeit wird im Allgemeinen eine Relation zwischen zwei oder mehreren Dingen verstanden, die in mindestens einem, aber nicht in allen Eigenschaften (Merkmalen) übereinstimmen und nicht stark voneinander abweichen.
Eine Eigenschaft kann beispielsweise die Tonhöhe sein. So sind sich zwei Töne ähnlich, wenn sie eine (fast) gleiche Grundfrequenz besitzen. Dabei ist die Klangfarbe, also welches Instrument den Ton erzeugt, zweitrangig. Anders hingegen ist eine ähnliche Klangfarbe, also die Art des Instruments eher unabhängig von der Tonhöhe. Zum Beispiel sollte eine Geige bei jedem normal erzeugten Ton den sie generiert, als Geige erkennbar sein.
In diesem Projekt wird sich auf die Ähnlichkeit der Klangfarbe, bzw des Instruments fokussiert. Dies ist im Bereich des Music Information Retrieval (MIR) eine nicht triviale Aufgabenstellung, was schwierig zu messen und quantifizieren ist. [4, S.26-30].
Generell lassen sich Merkmale für Music Information Retrieval als ein Spektrum von Low-Level- über Mid-Level- bis hin zu High-Level-Features unterteilen. Low-Level-Features sind Merkmale, die sich meist direkt aus der Wellen- oder Spektralrepräsentation berechnen lassen und wenig semantische Bedeutung beinhalten. Zum Beispiel ist der Mittelwert der Signalenergie, oder die Varianz des Spektrums ein Low-Level-Feature.
High-Level-Features sind vor allem symbolisch mit hoher Semantik. Beispiele für High-Level-Features sind die Instrumentenfamilie, Klangfarbe, das Musik-Genre, die Partitur oder auch die Stimmung.
Mid-Level-Features schließen die semantische Lücke zwischen Low-Level- und High-Level-Features. Sie repräsentieren musikalische Semantik, sind selbst aber nicht symbolisch. Ein Beispiel ist die sogenannte Piano-Rolloff-Repräsentation. Diese versucht die Obertöne aus einem Spektrogramm zu reduzieren, wodurch anschließend Partituren erzeugt werden können.
[16] [17] [18]
In diesem Projekt wird sich vor allem auf Low-Level-Merkmale konzentriert. Die meisten Merkmale und Filterfunktionen werden mit bereits implementierten Algorithmen der Open-Source-Bibliothek Essentia extrahiert [19]. Hinzu kommen noch Merkmale mit Ursprung aus der Digitalen Bildverarbeitung. Für diese wird auch eine neue Transformation vorgestellt, wodurch diese Merkmale Audiosignale besser beschreiben. Im folgenden werden einige Merkmale beschrieben, wobei eine Unterteilung in Bezug auf ihre Eingabe stattfindet. Also ob sie direkt auf dem zeitdiskreten Signal, einem Spektrum oder Spektrogramm berechnet werden.

3.1. Merkmale aus einem zeitdiskretem Signal

Zero Crossing Rate bzw. Nulldurchgangsrate beschreibt die Häufigkeit an Vorzeichenwechsel eines zeitdiskreten Signals. Dabei tendieren periodische Signale zu kleineren und verrauschte Signale zu größeren Werten. Es sollte damit Aufschluss geben, ob eine Audiodatei Rauschen enthält, oder das Instrument selbst aus einer Rauschquelle besteht (z.B. Snare oder Rasseln). [18]

Auto Correlation - Autokorrelation - beschreibt das Verhältnis eines Eingangsignals mit sich selbst zu einem späteren Zeitpunkt. Dabei wird der Grad des späteren Zeitpunkts durch lag-x beschrieben. lag-1 ist somit der Vergleich eines Signals $x(t)$ mit $x(t+1)$. Auch hier können damit Aussagen zu der Zufälligkeit eines Signals getätigt werden.[20]

Effective Duration bzw. effektive Länge ist die Zeit in der die Energie eines zeitdiskreten Signals kontinuierlich über einem bestimmten Schwellenwert liegt. Es wird zumindest bei der Implementierung in Essentia davon ausgegangen, dass ein Signal in einer Hülle (ADSR-Envelope) vorliegt. Damit könnten perkusive von nicht perkusiven Instrumenten unterschieden werden. [18] [21]

3.2. Merkmale aus einem Spektrum

Spectral Centroid oder spektraler Mittelwert ist das Barycenter eines Spektrums. Dabei wird die Frequenz als Wert aufgefasst und die Wahrscheinlichkeit für das Auftreten jeder Frequenz ist ihre normalisierte Amplitude. Es gibt Aussagen über die „Helligkeit“ und Form eines Klangs. [18]

Spread ist die Varianz eines Spektrums. [18]

Skewness - Schiefe - gibt Auskunft über die Asymmetrie einer Verteilung um den Mittelwert. Also ein Indikator, ob eine Verteilung symmetrisch ist, oder mehr Energie links bzw. rechts des Mittelwerts überwiegt [18]. Damit können sich eventuell Instrumente mit einem hohen Obertonanteil von Instrumenten mit einer kleinen Frequenzbandbreite unterscheiden.

Kurtosis bzw. Wölbung beschreibt die Flachheit einer Verteilung um den Mittelwert. Bei weißem Rauschen ist die Wölbung zum Beispiel sehr flach, ein einzelner Sinuston ist hingegen sehr spitz. [18]

Mel Frequency Cepstral Coefficients (MFCC) werden häufig in der Spracherkennung eingesetzt und werden allgemein wie folgt berechnet:

- Erstellen von n überlappenden Filtern auf Basis der Mel-Skala. Dies entspricht einer Dreiecksfensterfunktion im Spektrum.
- Berechnung der Energie der jeweiligen Frequenzbänder
- Logarithmierung der Amplitudenwerte
- Inverse Diskrete Kosinus Transformation für den Erhalt der cepstralen Koeffizienten
- Auswahl von m Koeffizienten

Die Wahl an ursprünglichen Mel-Bändern n und die jeweilige Reduktion auf m Koeffizienten ist nicht uniform definiert. Es zeigt sich, dass die MFCC als eigenständige unabhängige Werte betrachtet werden können und eigenen sich sowohl für Spracherkennung, Musikgenreerkennung sowie für Klangfarbenerkennung [11] [4, S.175-178]

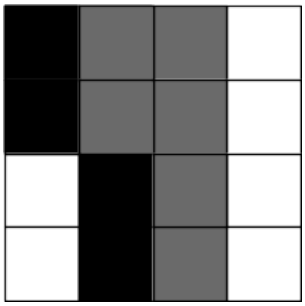
3.3. Merkmale aus einem Spektrogramm

Prinzipiell lassen sich alle Merkmale aus einem Spektrum auch auf Spektrogramme anwenden und erhalten in der Regal dadurch lediglich eine zeitliche Komponente. Darüber hinaus wird eine Methode aus der digitalen Bildverarbeitung verwendet und modifiziert.

Haralick Merkmale kommen eigentlich aus der digitalen Bildverarbeitung und beschreiben 13 statistische Werte die globale Textur von Bildern beschreiben. Diese eignen sich zum Klassifizieren von Bildern. Für Audio wird diese Methode auf ein Spektrogramm angewendet, welches als Grauwertbild interpretiert wird [22], [23]. Zunächst wird ein Bild in eine Gray-Level-Cooccurence-Matrix (GLCM) transformiert. Als Beispiel dient Abb. 6. Diese gibt Auskunft über die Häufigkeit von benachbarten Grauwerten. Im Beispiel handelt es ich um eine zyklische rechte Nachbarschaft. Daraus werden dann folgende 13 Werte berechnet:

Zweites Drehmoment	Kontrast	Korrelation	Varianz
Inverses Differenzmoment	Mittelwertsumme	Varianzsumme	Entropysumme
Korrelationsinformation (1)	Varianzdifferenz	Entropydifferenz	Entropy
Korrelationsinformation (2)			

Tabelle 1: Haralick Merkmale



(a) Bild mit 3 Grauwerten

	0	0	4
	4	2	0
	0	4	2

(b) GLCM

Abbildung 6: Beispiel einer GLCM

Haralick Merkmale mit Frequenz Aus Tabelle 1 ist zu entnehmen, dass die Haralick-Merkmale keine speziell auf Bilder angepasste Merkmale sind. Lediglich die GLCM basiert auf Bildern. In [22] und [23] wird das Spektrogramm in ein 8 Bit Grauwertbild konvertiert und dann in die GLCM überführt. Im Folgenden wird eine andere Nachbarschaftsrelationmatrix vorgestellt, die die Nachbarschaft von Frequenzen in einem Spektrogramm wiedergeben soll: Frequency-Adjacence-Cooccurence-Matrix (FACM)

Ein Spektrum besteht aus S_F diskreten Frequenzblöcken mit $L_f = \{0, 1, .., S_F - 1\}$, mehreren zeitlich diskreten Spektren S_T mit $L_t = \{0, 1, ..., S_T - 1\}$ und den diskreten Amplitudenwerten $S_M = (0, m)$. So ist ein Spektrogramm eine Funktion, welche ein Frequenz-Zeit-Paar auf einen Amplitudenwert abbildet.
Also $S : L_t \times L_f \rightarrow S_M$.
Für die Beschreibung der Frequency-Adjacence-Cooccurence-Matrizen (FACM) wird zunächst der Begriff der Adjazenz einer Frequenz im musikalischen Sinne definiert. Diese kann entweder parallel oder sequentiell ausfallen. Parallele Adjazenz einer Frequenz sind alle Frequenzen die zur selben Zeiteinheit erklingen. In einem Spektrum ist jede Frequenz parallel adjazent zueinander. Sequentielle Adjazenz sind hingegen alle Frequenzen die zeitlich nach einer Frequenz erklingen. In einem Spektrogramm sind das zwei zeitlich versetze Spektren. Es wird davon ausgegangen, dass durch die Kombination der sequentiellen und parallelen Adjazenz mittels elementweiser Mittelwertbildung eine ähnliche Matrix entsteht, wie aus der ursprünglichen Idee von Haralick [24]. Dort werden meist auch mehrere GLCMs mit anderen Nachbarschaftsrelationen erzeugt (rechts, diagonal, oben, etc.) und mit elementweiser Mittelwertbildung kombiniert. Es wird angenommen, dass mit FACM ebenfalls Texturmerkmale berechnen werden können.
Bei einer parallelen FACM sei jedes P_{ij} die gewichtete Summe des Amplitudenwerts der Frequenz i zur Zeiteinheit t_n, multipliziert mit dem Amplitudenwert der Frequenz j zur selben Zeiteinheit t_n. Bei einer sequentiellen FACM entspricht jedes P_{ij} der gewichteten Summe des Amplitudenwerts der Frequenz i zur Zeiteinheit t_n mit dem Amplitudenwert der Frequenz j zur Zeiteinheit t_{n+d}, mit d als variablen zeitlichen Versatz. Formal lässt es sich wie folgt beschreiben:

$$P(f_1, f_2) = \sum \{S(t, f_1) * S(t, f_2) \mid t \in L_t, f_1, f_2 \in L_f\}$$

$$P(f_1, f_2) = \sum \{S(t, f_1) * S(t + d, f_2) \mid f_1, f_2 \in L_f, d \in \mathbb{N}, t + d \; mod \; S_T \in L_t\}$$

Ein detailliertes Beispiel findet sich in Anhang A.
Aus der sequentiellen und parallelen FACM selbst, könnten auch Merkmale gewonnen werden. Darauf wird in dieser Ausarbeitung nicht näher eingegangen, doch als Visualisierung dient Abb. 7. Dort sind zwei Audiosamples mit paralleler (oben) und sequentieller FACM dargestellt.

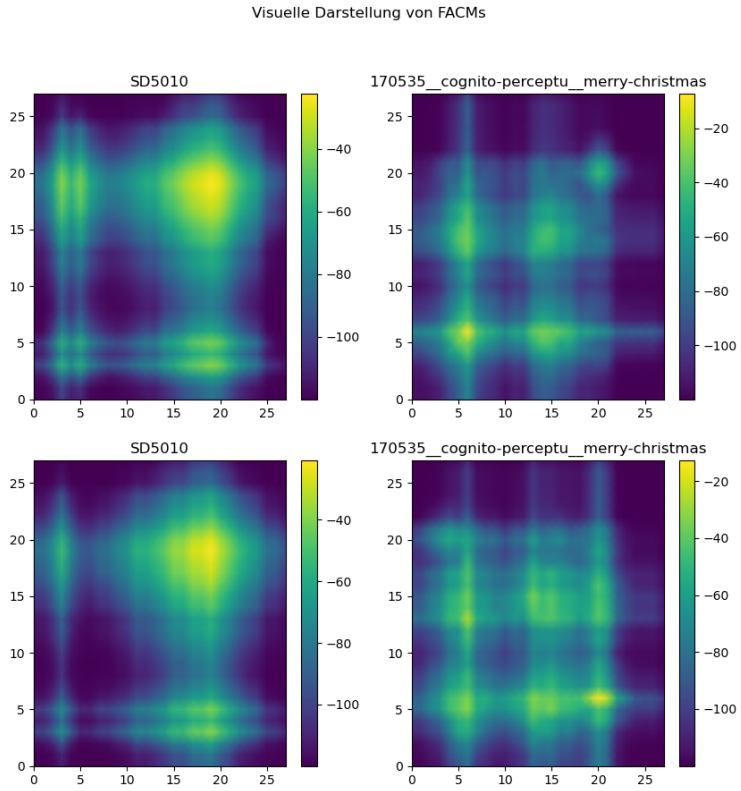


Abbildung 7: Vergleich von FACM aus zwei Super-Dirt-Audiosamples. Oberer Bereich entspricht paralleler FACM, untere Hälfte sequentieller FACM. Angewandt auf Bark-Bänder und Amplitudenwerte in db FS

4. Organisation von Audiodateien

In diesem Kapitel wird der Ablauf erklärt, wie geeignete Ähnlichkeitsmerkmale für eine Audiosample-Bibliothek gefunden werden können. Allgemein werden dazu Merkmale ausgewählt, gruppiert und mit Vergleichswerten überprüft um die Merkmale beurteilen zu können. Die durch diesen Prozess extrahierten Merkmale werden für die spätere Auswertung und Visualisierung in Kapitel 5 verwendet.

4.1. Dirt-Samples

Die Dirt-Samples sind die Standard-Audiosample-Bibliothek für den SuperDirt-Sampler [25] [1]. Diese Audiosample-Bibliothek hat das offene Problem der Rationalisierung, Neugruppierung und Kategorisierung der Bibliothek [26]. Sie ist zwar frei zugänglich, allerdings gibt es keine freie Lizenz dazu. Diese Bibliothek eignet sich für eine Analyse durch dieses Projekts. Es gibt insgesamt 2046 Audiodateien die uneinheitlich benannt und in nicht immer aufschlussreichen Ordnernamen kategorisiert sind. Beispiele hierfür sind:

Ordner	Dateiname	„Beschreibung“ durch Autor
mp3	[0-9]*mp*3[0-9].*wav*	diverses Rauschen
kurt	[0-9]*kurt*[0-9].*wav*	verzerrte Geräusche (von Kurt?)
battles	[0-9]*explo*[0-9].*wav*	explosionsartiges Rauschen
speakspell	[0-9].*wav*	verzerrte Mundgeräusche/Sprache
Blue	*	Autotune Sprache?
bev	*mono*, *stereo*	16 s Ausschnitt mit Gesang und Instrumenten
yeah	[0-9]*Sound*[0-9].*wav*	sehr kurze perkussive Geräusche
bird[1-2]	bird[0-9].*wav*	magisches Vogelzwitschern
bird3	bird3[0-9].*wav*	ächzendes KlirrKlarr (vllt. verzerrte Vögel)

Tabelle 2: Beispiele für Benennung der Dirt-Samples

Selbstverständlich gibt es auch sinnvolle Order und Dateienamen. Um später geeignete STFT-Parameter zu wählen wird die Länge der Audiosamples betrachtet (vlg. Abb. 8). Der Großteil der Dateien ist zwischen 0,1 und 0,5 Sekunden lang und etwa 30 Dateien sind unter 1 ms.

4.2. Vergleichswerte

Um Ergebnisse standardisiert überprüfen und Ähnlichkeitsmerkmale qualitativ messen zu können, ist es vorteilhaft Vergleichswerte in Form von Annotationen für den Datensatz zu besitzen. Diese wären bei den Dirt-Samples eine Zuordnung jedes Samples zu

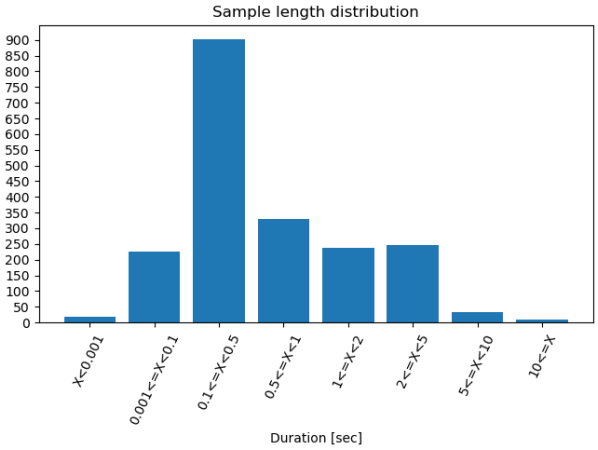


Abbildung 8: Verteilung der Audiosamplelänge

einem standardisierten klangbeschreibenden Wort. Die Dirt-Samples selbst sind nicht annotiert und die Namensgebung ist, wie aus Tab. 2 zu entnehmen, nicht immer ersichtlich. Es gibt also keine Grundwahrheit. Die über 2.000 Samples selbst zu annotieren ist zeitaufwändig und bei alleiniger Annotation auch fehleranfällig (vgl. „Beschreibung“ aus Tab. 2). Deshalb wurde auf ein automatisches Annotierprogramm zurückgegriffen. Es handelt sich dabei um das Neuronale Netz Cochlear.ai, das im Rahmen der „General-purpose audio tagging of Freesound content with AudioSet labels“ die besten Ergebnisse liefert [27] [28]. Ziel dieses Wettbewerbs war es, ein Neuronales Netz zu entwickeln, welches jedem Audiosample, aus einem Datensatz mit knapp 10000 Audiosamples, einer Kategorie zuordnet. Die Kategorien bestehen aus 41 Begriffen der „Googles AudioSet Ontology“ (vlg. Tab. 3).

Tearing	Bus	Shatter	Gunshot, gunfire	Fireworks
Writing	Oboe	Scissors	Microwave oven	Keys jangling
Drawer open or close	Squeak	Knock	Telephone	Saxophone
Computer keyboard	Flute	Clarinet	Acoustic guitar	Tambourine
Glockenspiel	Gong	Snare drum	Bass drum	Hi-hat
Electric piano	Harmonica	Trumpet	Violin, fiddle	Double bass
Cello	Chime	Cough	Laughter	Applause
Finger snapping	Fart	Meow	Cowbell	Bark
Burping, eructation				

Tabelle 3: 41 Begriffe aus Googles AudioSet Ontology

Diese Begriffe passen eigentlich nur begrenzt zu dem Datenmaterial mit oft synthetischen Klängen der Dirt-Samples. Bei sporadischen Wiedergaben konnte bisher weder ein „Bus“ noch eine „Mikrowelle“ wahrgenommen werden. Es wird davon ausgegangen, dass gleich klassifiziert Audiosamples sich auch ähnlich anhören, selbst wenn sie nicht

der entsprechenden Klassifizierung entsprechen. Insgesamt wurden 2017 von den insgesamt 2046 Dateien annotiert.
Bei manueller Überprüfung einiger Gruppen lässt sich feststellen, dass die Klassifizierung durchaus akzeptabel ist und vor allem unter Hi-Hat, Snare Drum und Bass Drum viele Dirt-Samples richtig zugeordnet werden.

4.3. Merkmalsgenerierung

Für die Generierung der Merkmale wurde sich für eine STFT-Fenstergröße $N = 4096$ mit einer Schrittweite von $H = 256$ und der der Hann-Fensterfunktion entschieden. Daraus ergibt sich eine Auflösung $\Delta t \approx 5,8$ ms und $\Delta f \approx 10,76$ Hz. Das Hann-Fenster eignet sich gut für unbekannte und variable Audiosignale. Die optimale Schrittweite beträgt eigentlich $H = \frac{N}{2}$, wodurch eine nahezu einheitliche Gewichtung aller zeitdiskreten Werte resultiert (vgl. Abb. 4a). Da viele der Audiosamples allerdings recht kurz sind wird mit einer sehr kleinen Schrittweite versucht, die zeitliche Auflösung zu verbessern [6].
Jedes Audiosample wird als Monosignal interpretiert, wodurch ihr Seitenanteil, falls vorhanden, ausgelöscht wird. Alle spektralen Merkmale werden mit 6 verschieden Frequenzskalierungen bzw. Frequenzbändern berechnet. Diese sind:

- Lineares Spektrum
- Logaritmisches Spektrum [29]
- Mel-Band mit 96 Frequenzbändern
- Bark-Band mit 28 Frequenzbändern
- 99 Chromatisch basierte Frequenzbänder (mit Hilfe von Rechteckfenstern)
- 198 Chromatisch basierte Frequenzbänder (mit Hilfe von Dreiecksfenstern)

Insgesamt werden pro Audiodatei 212 Merkmale generiert und gespeichert, sodass diese nicht für jede Analyse erneut berechnet werden müssen. Die unterschiedliche Anzahl an Bark-Bändern liegt an der von Essentia abweichenden Definition [30]. Es werden zwei niedrig-Bark-Bänder halbiert. Die Chromatisch basierte Frequenzbandeinteilung unterteilt das Spektrum in diesem Projekt zunächst auf 99 Bänder, wobei 1 Band die Breite einer Musiknote besitzt. Es umspannt den Frequenzbereich von $440Hz{*}2^{-32/12} \approx 69.29Hz$ bis $440Hz * 2^{67/12} \approx 21096Hz$. Dabei wurden die Frequenzen mit Hilfe eines Rechteckfensters zusammengefasst. Die 198 Noten-Variante fügt jeweils den Mittelwerte zwischen zwei Noten hinzu mit einer überlappenden Dreiecksfensterfunktion. Die wesentlichen Merkmale die damit berechnet wurden sind in Kapitel 3.1,3.2 und 3.3 beschrieben. Diese werden fast konsequent auf alle Spektren und Spektrogramme angewendet. Lediglich die Haralickmerkmale werden werder mit der GLCM noch mit der FACM für das Lineare Spektrum berechnet. Alle FACMs und GLCMs werden mit Nachbarschaftsabstand

$d = 1$ berechnet und normalisiert, sodass die Summe aller Werte 1 entspricht. Für die GLCMs wird ein Spektrogramm nicht als Grauwertbild exportiert, sondern der Wertebereich zwischen Minimum und Maximum innerhalb eines Spektrums in den Wertebereich $(0, 1, ..., 255)$ überführt.

4.4. Clustering

Es wurden in Abschnitt 4.2 41 Vergleichsklassen vorgestellt, welche standardisierte Klangnamen zu Klängen und Geräuschen zuordnen. Dadurch wurden die Dirt-Samples in 40 Klassen eingeordnet. Es soll nun versucht werden, über eine andere Methode die Audiosamples in 40 Klassen einzuteilen um die Zuweisung mit den Vergleichswerten zu überprüfen und daraus Aussagen über die Qualität der Merkmale zu treffen. Die Klassen- bzw. Gruppeneinteilung wird mit Hilfe eines Clusterverfahrens erstellt. Ein nachvollziehbares Verfahren bietet das hierarchische Clustern, bzw. das Agglomerative Clustern. Dabei werden einzelne Audiosamples nach und nach zu Clustern vereint. Das heißt zu Beginn repräsentiert jedes Audiosample jeweils ein Cluster. In jedem Schritt werden dann die Cluster mit dem geringsten Abstand vereint, bis die gesamte Menge in einer Gruppe enthalten ist. Anschließend kann daraus eine beliebige Anzahl an Gruppen extrahiert werden. Als Vereinigungsoperation kann dabei Durchschnittsabstand, Minimalabstand, Maximalabstand oder auch die Minimierung der Varianz verwendet werden. Nachteil dieses Verfahrens ist, dass je nach Vereinigungsoperation kleine bzw. einelementige Cluster entstehen können. Die Laufzeit beträgt je nach Implementation $\mathcal{O}(n^2)$ bis $\mathcal{O}(n^3)$ mit n als Anzahl an Elementen, was bei etwa 2000 Dateien keinerlei Hürde darstellen sollte [31] [32] [33].
Aus den Ergebnissen des Agglomerativen Clusterns lässt sich ein Dendrogramm erzeugen, was als Visualisierung, oder auch Landkarte zur Ähnlichkeit der Audiosamples dienen kann (s. Abb. 10). Darauf wird in Kapitel 5 näher eingegangen.

4.5. Merkmalsauswahl

Es gibt nun eine vorhandene Audiosample-Bibliothek, welche automatisch annotiert wurde, ein Clusterverfahren, dessen Ergebnisse mit der Annotation verglichen werden können und eine Menge von Merkmalen zu den Audiosamples. Auf dieser Basis baut die Merkmalsauswahl auf. Es wird eine Teilmenge aller erzeugten Merkmale generiert, mit Hilfe dieser werden die Audiosamples geclustert und anschließend überprüft, ob die Clusterung mit der Vergleichsannotation korreliert. Im Folgenden werden zwei Varianten beschrieben um Teilmengen aus allen Merkmalen auszuwählen.

Zufallsbasiert Hierbei wird zunächst eine Grundmenge selektiert, geclustert und überprüft wie stark das Ergebnis mit der Annotation korreliert. Anschließend wird zufällig eine kleine Anzahl an Merkmalen hinzugefügt und/oder entfernt. Mit diesen Merkmalen wird dann wieder geclustert und die Korrelation mit der Annotation berechnet. Wenn das Ergebnis besser ist, wird die Teilmenge übernommen, ansonsten wird die vorherige Teilmenge erneut modifiziert. Dies wird für eine bestimmte Anzahl an Iterationen durchgeführt und das beste Ergebnis ausgegeben und gespeichert.

Korrelationsbasiert Es wird eine Korrelationsmatrix zwischen allen Merkmalen und der Annotation erstellt. Anschließend werden die Merkmale ausgewählt, die selbst stark mit der Vergleichsannotierung und schwach mit allen anderen Merkmalen korrelieren. Damit soll sicher gestellt werden, dass die ausgewählten Merkmale möglichst unabhängig voneinander sind und dadurch möglichst viel Information tragen. Dies ist ein deterministisches Verfahren.

Im allgemein werden alle Merkmale normalisiert, sodass der höchste Wert 1 und der niedrigste Wert 0 innerhalb eines Merkmals beträgt. So soll verhindert werden, dass einzelne Merkmale bei einer Abstandsberechnung dominieren. Des weiteren werden alle Merkmale, die negativ mit der Annotation korrelieren, negiert. Ziel der Merkmalsauswahl ist es, eine Teilmenge aller erzeugten Merkmale zu finden, die aussagekräftig die Audiodateien gruppieren können und somit als Indikator für ein Ähnlichkeitmaß in Form einer Abstandsfunktion fungiert.

5. Ergebnisse und Visualisierung

Aus den verschiedenen Kombinationen der generierten Merkmale korreliert die folgende Teilmenge mit einem Wert von -0.497351 am stärksten mit der Vergleichsannotation:

Name	Korrelation zur Annotation
Haralick:LogSpektrogramm:FACM:Summe der Mittelwerte	0.5338620705935813
MFCC-4	0.4210362890893467
MFCC-7	0.4089776354173794
Haralick:99Noten:FACM:Summe der Varianz	0.38427288419484157
199Noten:Spektraler Mittelwert	0.36561708654078906
ZeroCrossingRate	0.3555859090827954
MFCC-8	0.34931501759443606

Tabelle 4: Bestes Korrelationsergebnis

In Tab. 9 in Anhang C sind die besten 20 Merkmale in Bezug auf ihre absolute Korrelation mit den Vergleichsannotationen gelistet. Es fällt auf, dass zwei Haralick-Merkmale (Summe der Mittelwerte und Summe der Varianz), für fast alle gewählten Spektren mit FACM-Transformation, relativ stark mit den Annotationen korrelieren und teilweise bessere Werte als MFCC erzielen. Vor allem die Mittelwert-Summe der Haralick Merkmale sind von allen Spektren, für die sie berechnet wurden, unter den besten 13 Merkmalen. Sie scheinen stabil gegenüber verschiedenen Frequenzskalierungen zu sein. Auch ist kein über die GLCM generiertes Merkmal dort gelistet. Dies lässt den Schluss zu, dass die FACM eher eine Nachbarschaftsrelation eines Audiosignals repräsentiert als die GLCM, oder Implementationsfehler bei der GLCM existieren.

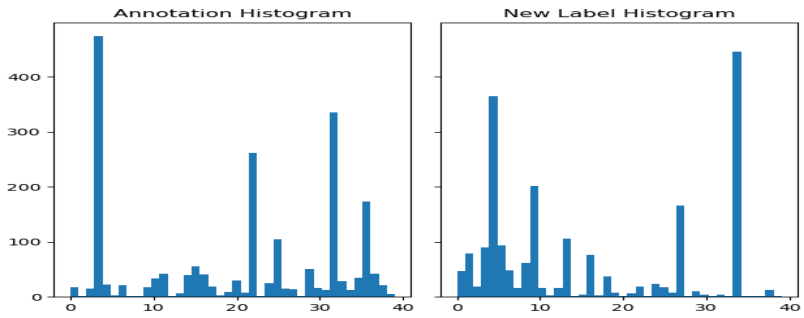


Abbildung 9: Histogramm der Gruppenzuweisung

Aus Abb. 9 ist das Histogramm der Gruppeneinteilung der Vergleichsannotation (links) und die durch das Clustering entstandene Zuweisung (rechts) zu sehen. Die Klassennamen sind durch Zahlen ersetzt. Die Vermutung war, dass auch bei einer nicht deckungsgleichen Gruppenzuweisung, die einzelnen Gruppen in sich Ähnlichkeiten aufweisen. Die

Einteilungen sind nach manueller Begutachtung durchaus akzeptabel. Beispielsweise befinden sich in der größten Gruppe vor allem Bass Drums, Kick Drums und diverse Trommeln mit wenig völlig unpassenden Audiosamples. Aus der Agglomerativen Clusterung kann ebenfalls das Dendrogramm zur Visualisierung dienen (vgl. Abb. 10). Dabei ist der Abstand zwischen den Clustern auf der unteren Achse eingezeichnet. Da die Vereinigungsoperation als mittleren Abstand aller beinhaltenden Elemente verwendet wurde, ist nicht sofort ersichtlich, wie groß der Ähnlichkeitsabstand zwischen einzelnen Audiosamples ist. Darüber hinaus wurden für eine lesbare Darstellung bereits Cluster als Knoten zusammengefasst (eingeklammerte Zahlen auf der linken Seite), was das soeben genannte Problem verstärkt.
Eine weitere Darstellung erfolgt als minimaler Spannbaum aller Audiosamples mit dem Visualisierungstool GraphViz [34]. In Abb. 11, 12 und 13 in Anhang B ist der minimale Spannbaum über alle Audiosamples dargestellt (Dateinamen wurden verkürzt). Der Vorteil daran ist, dass damit die Abstände zwischen den Audiosamples in Relation zu den Ähnlichkeitsabständen stehen. Der wesentliche Nachteil daran ist, dass die resultierende Landkarte nicht sonderlich leserlich und gleichzeitig vollständig abbildbar ist. Für eine Webansicht könnte diese Landkarte allerdings durchaus Verwendung finden und eine eventuelle Segmentierung würde dabei ebenso helfen.

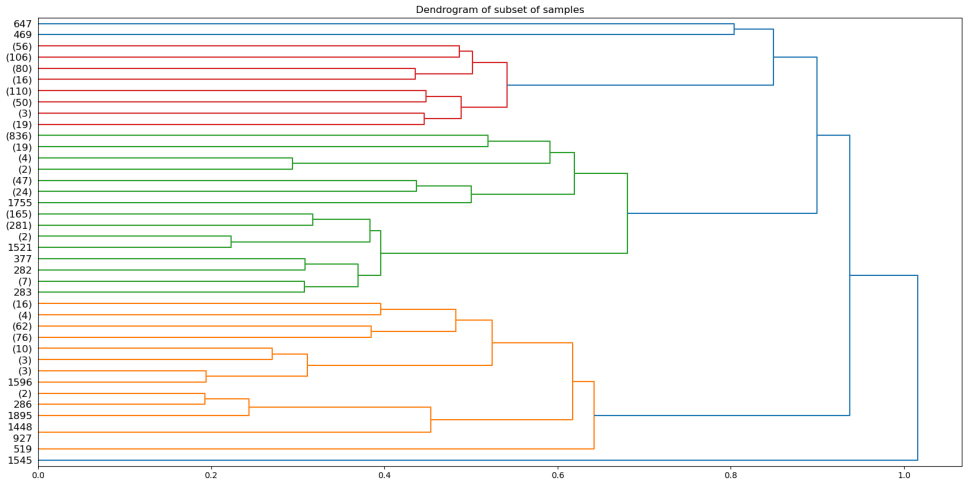


Abbildung 10: Dendrogramm der Clusterzuweisung

6. Fazit

Das Ziel dieses Projekts war es, Ähnlichkeitsmerkmale für Audio zu entwickeln, zu evaluieren und eine passende graphische Repräsentation des Ähnlichkeitsabstands zu finden. Ähnlichkeit wurde dabei im Sinne der Klangfarbe von Audio definiert und das Ähnlichkeitsmaß ist der kleinste Abstand der Merkmale zwischen Audiosamples. Es wurde zunächst auf die Digitale Signalverarbeitung von Audio eingegangen, um daraus Merkmale zu entwickeln, bzw. bereits implementierte Algorithmen zu verstehen und einzusetzen. Da die Dirt-Samples keine Annotationen besitzen, wurden diese über ein automatisches Annotationsprogramm erstellt, um qualitative Aussagen über die verwendeten Merkmale zu tätigen. Die aus diesem Vergleichsprozess herauskristallisierten Merkmale werden schlussendlich verwendet um:

- die Audiosamples in Gruppen unterteilen zu können
- die n ähnlichsten Audiosamples zu suchen
- Ähnlichkeitabstände durch einen minimalen Spannbaum zu visualisieren

Die in dieser Ausarbeitung entwickelte Nachbarschaftsrelationsmatrix für Audio (FACM) korreliert sowohl für die Mittelwertsumme als auch für die Varianzsumme am stärksten zur Vergleichsannotation bei fast allen Frequenzskalierungen (vgl. Tab. 9). Diese sollten mit offiziell vergleichbaren Daten überprüft werden. Die Gruppeneinteilung durch die besten Merkmale wurden manuell überprüft und der subjektive Eindruck dazu ist zufrieden stellend - mit Verbesserungsmöglichkeiten. Es werden generell perkussive Klänge gruppiert, in denen das ein oder andere Instrumentensample wie z.B. Gitarre, Trompete, etc. mit zugewiesen wird. Es gibt auch völlig bunt gemischte Gruppen. Die Visualisierung des Ähnlichkeitabstands mit einem minimalen Spannbaum ist möglich, doch für eine praktische Relevanz sollte die Lesbarkeit verbessert werden.
Es gibt diverse Fehlerquellen, welche die Ergebnisse dieser Ausarbeitung beeinflussen. Zunächst sind alle Merkmale von den gewählten Parametern der Digitalen Signalverarbeitung abhängig. Änderungen der Fensterfunktion, Sprungweite und Fenstergröße haben einen starken Einfluss auf alle Merkmale. Die automatische Annotation, welche zur Evaluation dient, ist nicht akkurat und die 41 Klangfarbenbegriffe eignen sich nur bedingt für synthetische Klänge. Des Weiteren ist die Optimierung der Merkmale mittels Korrelation zur Vergleichsannotation nicht optimal. Korrelation repräsentiert lediglich den paarweisen linearer Zusammenhang. Es können auch nichtlineare Zusammenhänge zwischen mehr als einem Paar vorherrschen, die somit nicht erkannt werden. Auch die Gruppierung durch ein Agglomeratives Clustering kann verbessert, oder durch ein geeigneteres Verfahren ausgetauscht werden. Zuletzt könnte der Vergleich der hier vorgestellten FACM mit der ursprünglichen GLCM auch durch eine kognitive Verzerrung des Autors erklärt werden, wodurch Fehler der FACM eher untersucht und optimiert und Fehler der GLCM als inhärent angenommen wurden.

Zusammenfassend zeigt diese Ausarbeitung das generelle Vorgehen für eine Merkmalsextraktion aus Audiodateien für eine Neuorganisation der selbigen und Visualisierung der Ähnlichkeit. Die schlussendlichen Ergebnisse sind aus subjektiven Empfinden zufriedenstellend, wenn auch verbesserungsfähig.

A. Berechnung der FACM

Hier wird ein detailliertes Beispiel einer FACM beschrieben. Tab 5 definiert ein Beispielproblem. Durch Berechnung 1 wird eine FACM mit paraller Adjazenz gebildet (Ergebnis in Tabelle 8a). Berechnung 2 zeigt den Rechenweg für eine zyklisch-sequentielle Adjazenz mit Abstand $d = 1$(Ergebnis in Tabele 8b). Schlussendlich ist in Tabelle 8c die Kombination der beiden FACM mit elementweiser Mittelwertbildung.

Tabelle 5

(a) Indices eines Spektogramms mit $L_t = \{1, 2, 3, 4\}, L_f = \{1, 2, 3\}$

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)

(b) Spektrogramm

0.9	0.2	0.1	0.0
0.3	0.8	0.8	0.2
0.9	0.9	0.8	0.4

$$
\begin{aligned}
P(1,1) &= (1,1) * (1,1) + (2,1) * (2,1) + (3,1) * (3,1) + (4,1) * (4,1) \\
&= 0.9 * 0.9 + 0.2 * 0.2 + 0.1 * 0.1 + 0.0 * 0.0 \\
P(1,2) &= (1,1) * (1,2) + (2,1) * (2,2) + (3,1) * (3,2) + (4,1) * (4,2) \\
&= 0.9 * 0.3 + 0.2 * 0.8 + 0.1 * 0.8 + 0.0 * 0.2 \\
P(1,3) &= (1,1) * (1,3) + (2,1) * (2,3) + (3,1) * (3,3) + (4,1) * (4,4) \\
&= 0.9 * 0.9 + 0.2 * 0.9 + 0.1 * 0.8 + 0.0 * 0.4 \\
P(2,2) &= (1,2) * (1,2) + (2,2) * (2,2) + (3,2) * (3,2) + (4,2) * (4,2) \\
&= 0.3 * 0.3 + 0.8 * 0.8 + 0.8 * 0.8 + 0.2 * 0.2 \\
P(2,3) &= (1,2) * (1,3) + (2,2) * (2,3) + (3,2) * (3,3) + (4,2) * (4,3) \\
&= 0.3 * 0.9 + 0.8 * 0.9 + 0.8 * 0.8 + 0.2 * 0.4 \\
P(3,3) &= (1,3) * (1,3) + (2,3) * (2,3) + (3,3) * (3,3) + (4,3) * (4,3) \\
&= 0.9 * 0.9 + 0.9 * 0.9 + 0.8 * 0.8 + 0.4 * 0.4 \\
P(2,1) &= P(1,2) \\
P(3,1) &= P(1,3) \\
P(3,2) &= P(2,3)
\end{aligned}
$$

1.: Berechnung für parallele Adjazenz

$$
\begin{aligned}
P(1,1) &= (1,1) * (2,1) + (2,1) * (3,1) + (3,1) * (4,1) + (4,1) * (1,1) \\
&= 0.9 * 0.2 + 0.2 * 0.1 + 0.1 * 0.0 + 0.0 * 0.9 \\
P(1,2) &= (1,1) * (2,2) + (2,1) * (3,2) + (3,1) * (4,2) + (4,1) * (1,2) \\
&= 0.9 * 0.8 + 0.2 * 0.8 + 0.1 * 0.2 + 0.0 * 0.3 \\
P(1,3) &= (1,1) * (2,3) + (2,1) * (3,3) + (3,1) * (4,3) + (4,1) * (1,3) \\
&= 0.9 * 0.9 + 0.2 * 0.8 + 0.1 * 0.4 + 0.0 * 0.9 \\
P(2,1) &= (1,2) * (2,1) + (2,2) * (3,1) + (3,2) * (4,1) + (4,2) * (1,1) \\
&= 0.3 * 0.2 + 0.8 * 0.1 + 0.8 * 0.0 + 0.2 * 0.9 \\
P(2,2) &= (1,2) * (2,2) + (2,2) * (3,2) + (3,2) * (4,2) + (4,2) * (1,2) \\
&= 0.3 * 0.8 + 0.8 * 0.8 + 0.8 * 0.2 + 0.2 * 0.3 \\
P(2,3) &= (1,2) * (2,3) + (2,2) * (3,3) + (3,2) * (4,3) + (4,2) * (1,3)) \\
&= 0.3 * 0.9 + 0.8 * 0.8 + 0.8 * 0.4 + 0.2 * 0.9 \\
P(3,1) &= (1,3) * (2,1) + (2,3) * (3,1) + (3,3) * (4,1) + (4,3) * (1,1) \\
&= 0.9 * 0.2 + 0.9 * 0.1 + 0.8 * 0.0 + 0.4 * 0.9 \\
P(3,2) &= (1,3) * (2,2) + (2,3) * (3,2) + (3,3) * (4,2) + (4,3) * (1,2)) \\
&= 0.9 * 0.8 + 0.9 * 0.8 + 0.8 * 0.2 + 0.4 * 0.3 \\
P(3,3) &= (1,3) * (2,3) + (2,3) * (3,3) + (3,3) * (4,3) + (4,3) * (1,3) \\
&= 0.9 * 0.9 + 0.9 * 0.8 + 0.8 * 0.4 + 0.4 * 0.9
\end{aligned}
$$

2.: Berechnung einer zyklisch-sequentiellen Adjazenz

Tabelle 7

0.2	0.32	0.63
0.9	1.1	1.72
1.01	1.41	2.21

(a) Zyklisch-Sequentielle FACM

0.85	0.51	1.07
0.51	1.41	1.71
1.07	1.71	2.42

(b) Parallele FACM

0.525	0.415	0.85
0.705	1.255	1.715
1.04	1.56	2.315

(c) Zellweise Mittelwertbildung aus Tabelle 8a und 8b

B. Landkarte der Dirt-Samples

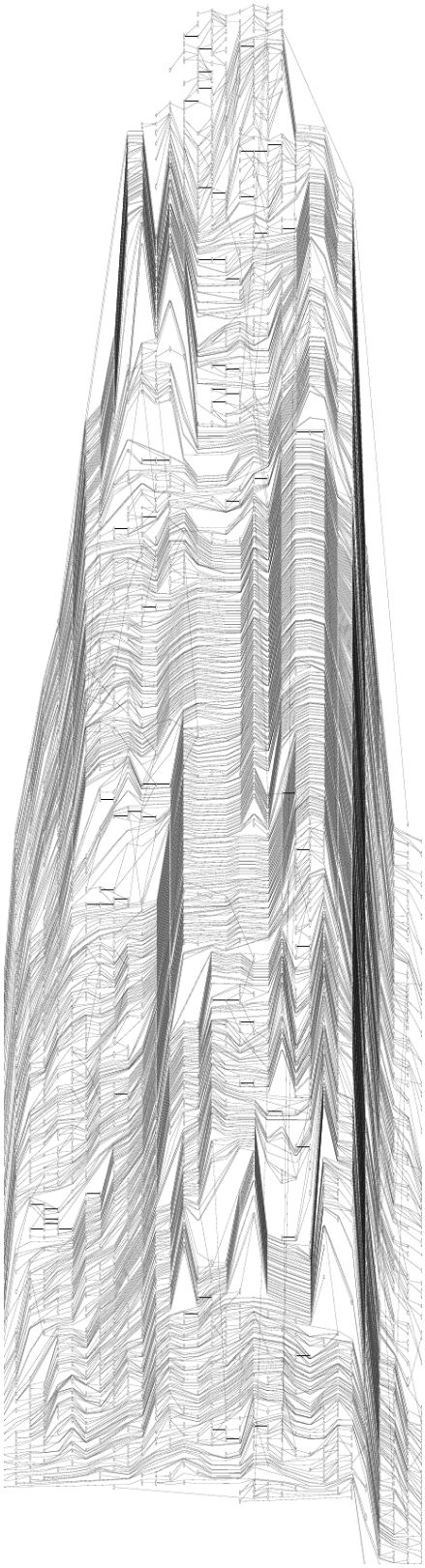


Abbildung 11: Vollständige Landkarte (90 Grad rotiert)

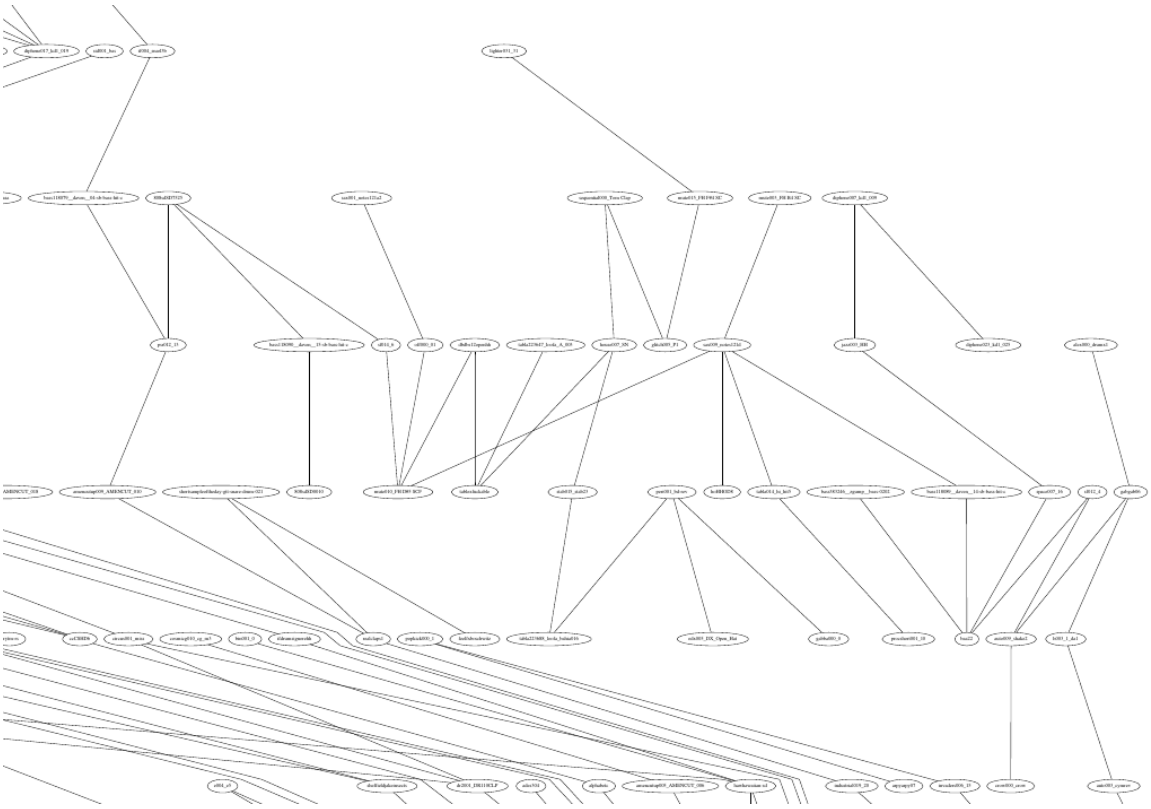


Abbildung 12: Kleiner Ausschnitt der Landkarte

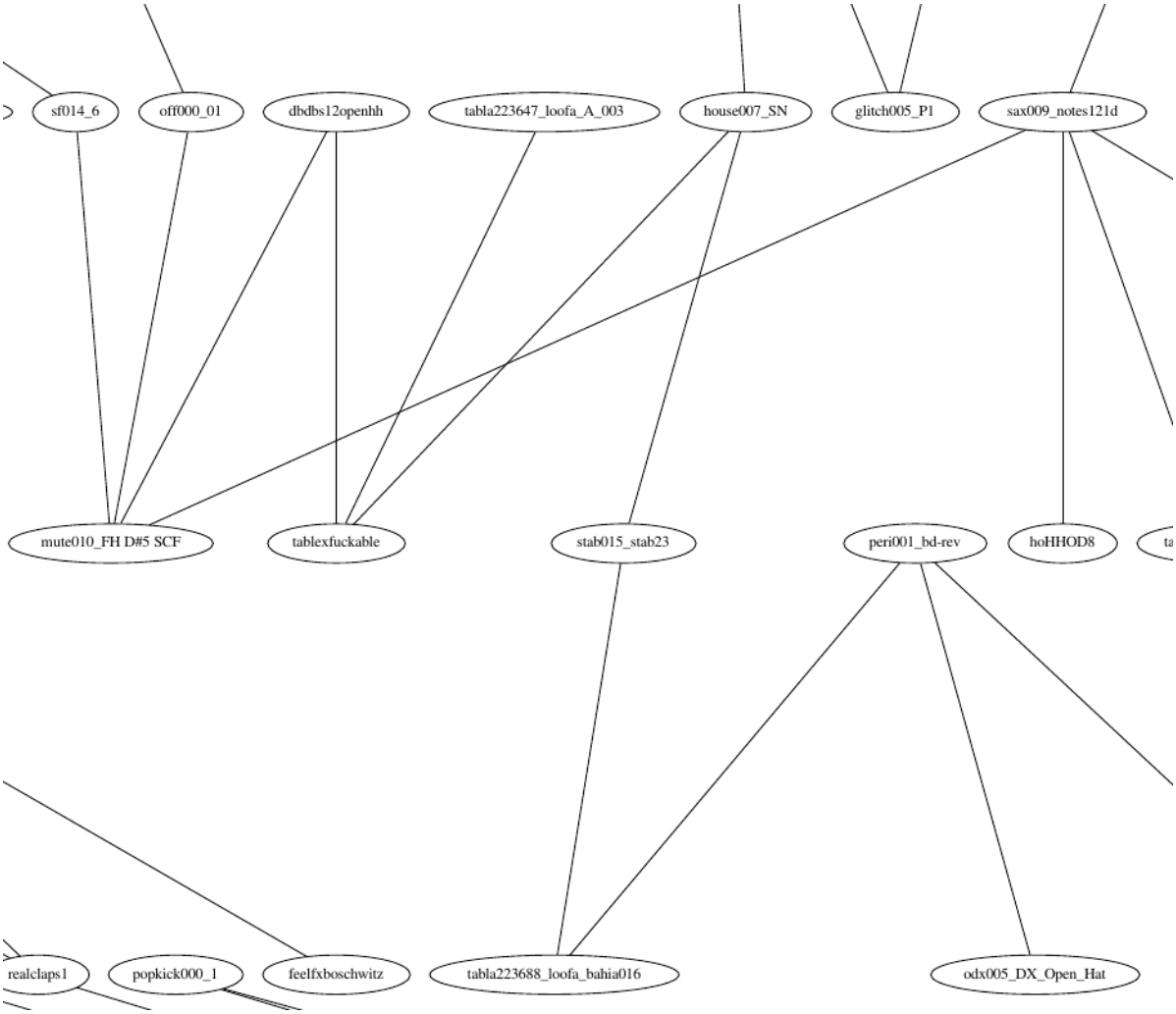


Abbildung 13: Noch kleinerer Ausschnitt der Landkarte

C. Die besten 20 Merkmale

Name	Korrelation zur Annotation
Har-Log_-FACM_10	0.533862
Har-Log_-FACM_12	0.481785
RecChr_Centroid	0.481289
MFCC-3	0.474201
Bark28_Centroid	0.458489
Har-RecChr_-FACM_10	0.447002
Lin_Centroid	0.446877
Har-TriChr_-FACM_10	0.441772
MFCC-4	0.421036
Har-Bark28_-FACM_10	0.417385
MFCC-7	0.408978
Har-RecChr_-FACM_12	0.384273
Har-Mel96_-FACM_10	0.373660
TriChr_Centroid	0.365617
ZeroCrossingRate	0.355586
Har-TriChr_-FACM_12	0.352475
MFCC-8	0.349315
Har-Bark28_-FACM_12	0.344520
MFCC-11	0.337367
Lin_EssEntropy	0.329154

Tabelle 9: Besten 20 Merkmale
Har = Haralick, Log = LogSpektrum, FACM_10 = Haralick:Summe der Mittelwerte, FACM_12 = Haralick:Summe der Varianz, TriChr = 198-Chromatische-Frequenzbänder, RecChr = 99-Chromatische-Frequenzbänder, Lin = Lineares Spektrum.

Literatur

[1] A. M. Julian Rohrhuber and contributors, "Dirt-Samples." https://github.com/musikinformatik/Dirt-Samples(2020-09-05), commit: 1d67800, 2019.

[2] P. Kabal, "Audio File Format Specifications." http://www-mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/WAVE.html (2020-08-30).

[3] E. Günter and Schukat-Talamazzini, *Automatische Spracherkennung - Grundlagen, statistische Modelle und effiziente Algorithmen.* Vieweg+Teubner Verlag, Wiesbaden, 1995.

[4] M. Müller, *Fundamentals of Music Processing.* Springer International Publishing, 01 2015.

[5] T. Kuttner, *Freie Schwingungen.* Wiesbaden: Springer Fachmedien Wiesbaden, 2015.

[6] G. Heinzel and A. Rüdiger, "Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows," *Max Plank Inst*, vol. 12, 01 2002.

[7] D. C. von Grünigen, *Digitale Signalverarbeitung - mit einer Einführung in die kontinuierlichen Signale und Systeme.* Hanser Fachbuchverlag, 2014.

[8] D. F. Elliott, *Handbook of Digital Signal Processing: Engineering Applications.* Academic Press, 1988.

[9] H. Traunmüller, "Analytical Expressions for the tonotopic sensory scale," *The Journal of the Acoustical Society of America*, vol. 88, pp. 97–100, 07 1990.

[10] J. O. S. III, "The Bark and ERB Bilinear Transforms," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 6, pp. 697–708, 1999.

[11] T. Ganchev, N. Fakotakis, and K. George, "Comparative evaluation of various MFCC implementations on the speaker verification task," *Proceedings of the SPECOM*, vol. 1, 01 2005.

[12] T. Kamm, H. Hermansky, and A. Andreou, "Learning the Mel-scale and Optimal VTN Mapping," 10 2011.

[13] S. J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4.* Cambridge University Press, 2006.

[14] T. Kamm, H. Hermansky, and A. G. Andreou, "Learning the Mel-scale and Optimal VTN Mapping," 2011.

[15] A. E. Society, "AES standard method for digital audio engineering - Measurement of digital audio equipment," tech. rep., Audio Engineering Society, Inc., 1998.

[16] T. Kitahara, "Mid-level Representations of Musical Audio Signals for Music Information Retrieval," in *Advances in Music Information Retrieval*, pp. 65–91, Springer Berlin Heidelberg, 2010.

[17] D. Bogdanov, J. Serrà, N. Wack, P. Herrera, and X. Serra, "Unifying low-level and high-level music similarity measures," *IEEE Transactions on Multimedia*, pp. 687–701, 2011.

[18] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," tech. rep., Icram, 01 2004.

[19] M. T. Group, "Essentia - Open-source library and tools for audio and music analysis, description and synthesis." https://essentia.upf.edu (2020-06-03).

[20] NIST, "Autocorrelation." https://www.itl.nist.gov/div898/handbook/eda/section3/eda35c.htm (2020-08-20).

[21] M. T. Group, "Essentia - Algorithm Reference - Effective Duration." https://essentia.upf.edu/reference/std_EffectiveDuration.html (2020-08-20).

[22] S. Sawada, Y. Takegawa, and K. Hirata, "On Hierarchical Clustering of Spectrogram," in *Music Technology with Swing* (M. Aramaki, M. E. P. Davies, R. Kronland-Martinet, and S. Ystad, eds.), (Cham), pp. 226–237, Springer International Publishing, 2018.

[23] Y. Costa, L. Soares de Oliveira, A. Koerich, and F. Gouyon, "Comparing textural features for music genre classification," *International Joint Conference on Neural Networks*, pp. 1867–1872, 01 2012.

[24] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[25] A. M. Julian Rohrhuber and contributors, "SuperDirt." https://github.com/musikinformatik/SuperDirt (2020-09-05), commit: d92384f, 2020.

[26] A. M. Julian Rohrhuber and contributors, "Dirt-Samples." https://github.com/tidalcycles/Dirt-Samples/issues/15 (2020-09-05), commit: 1d67800, 2019.

[27] I.-Y. Jeong and H. Lim, "AUDIO TAGGING SYSTEM FOR DCASE 2018: FOCUSING ON LABEL NOISE, DATA AUGMENTATION AND ITS EFFICIENT LEARNING," tech. rep., DCASE2018 Challenge, September 2018.

[28] E. Fonseca, M. Plakal, F. Font, D. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose Tagging of Freesound Audio with AudioSet Labels: Task Description, Dataset, and Baseline," *ArXiv*, vol. abs/1807.09902, 2018.

[29] M. T. Group, "Essentia - Algorithm Reference - LogSpectrum." https://essentia.upf.edu/reference/std_LogSpectrum.html (2020-08-20).

[30] M. T. Group, "Essentia - Algorithm Reference - Bark Bands." https://essentia.upf.edu/reference/std_BarkBands.html (2020-08-20).

[31] F. Wysotzki, "Steinhausen, D. / Langer, K., Clusteranalyse. Einführung in die Methoden und Verfahren der automatischen Klassifikation. Algorithmen, Fortran-Programme, Anwendungsbeispiele. Berlin-New York. Walter de Gruyter. 1977. 206 S., 63 Abb., DM 34,–," *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 59, no. 2, pp. 142–143, 1979.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[33] F. e. a. Pedregosa, "Agglomerative Clustering." https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html (2020-08-20), 2020.

[34] E. R. Gansner and S. C. North, "An open graph visualization system and its applications to software engineering," *SOFTWARE - PRACTICE AND EXPERIENCE*, vol. 30, no. 11, pp. 1203–1233, 2000.