Inhaltsverzeichnis

[Abstract 3](#_Toc26184355)

[Danksagung 3](#_Toc26184356)

[1. Einleitung 3](#_Toc26184357)

[2. Arten von Pattern 3](#_Toc26184358)

[3. Die Fourier Transformation 3](#_Toc26184359)

[4. Die Spektralanalyse 3](#_Toc26184360)

[5. Pattern Erkennung mithilfe der Fourier Transformation 3](#_Toc26184361)

[6. Pattern Erkennung mithilfe von Rhythmus und Melodie 3](#_Toc26184362)

[6.1. String basierte Pattern suche 3](#_Toc26184363)

[6.2. Geometrische Pattern suche 3](#_Toc26184364)

[6.2.1. Fünf-Dimensionales Punkt Set 3](#_Toc26184365)

[6.2.1.1. SIA 3](#_Toc26184366)

[6.2.1.2. SIATEC 3](#_Toc26184367)

[6.2.1.3. COSIATEC 3](#_Toc26184368)

[6.3. Pattern Erkennung mithilfe von Matrizen 3](#_Toc26184369)

[7. Pattern Erkennung mithilfe von Neuronalen Netzen 4](#_Toc26184370)

[Abbildungsverzeichnis 5](#_Toc26184371)

[Tabellenverzeichnis 5](#_Toc26184372)

[Quellenverzeichnis 5](#_Toc26184373)

[Eidesstattliche Erklärung 5](#_Toc26184374)

# Abstract

# Danksagung

# 1. Einleitung

# 2. Arten von Pattern

# 3. Die Fourier Transformation

# 4. Die Spektralanalyse

# 5. Pattern Erkennung mithilfe der Fourier Transformation

# 6. Pattern Erkennung mithilfe von Rhythmus und Melodie

# 6.1. String basierte Pattern suche

# 6.2. Geometrische Pattern suche

# 6.2.1. Fünf-Dimensionales Punkt Set

(Erst ab drittens Interessant!! Zweidimensional Aufbereiten <http://drops.dagstuhl.de/opus/volltexte/2006/652/pdf/06171.MeredithDavid.Paper.652.pdf> Plus Quellen, Chromatische Scala für Musik(Wert 2 der Fünf Dimensionales Punktset)

Algorithmen dafür unter anderen: SIA (MTPs), SIATEC, COSIATEC)

# 6.2.1.1. SIA

# 6.2.1.2. SIATEC

# 6.2.1.3. COSIATEC

# 6.3. Pattern Erkennung mithilfe von Matrizen

<http://hss.ulb.uni-bonn.de/2013/3271/3271.pdf> (darunter alle von dem Link ab 6.2 interessant)

Pattern können auch in Subpattern unterteilt sein

Pattern ranking function?

Ganze Musikstück muss nicht unbedingt ein Pattern sein es können auch „sinnlose“ Filler dabei sein

Arten von Pattern (Seite 98)

Methode: Melody extraction (nimmt an das Melodie in der höchsten Note enthalten ist)

Stringbasierte Erkennung und geometrische Erkennung (Seite 101 Quellen anschauen) und beides zusammen (102 oben)

<https://www.researchgate.net/profile/Jia_Lien_Hsu/publication/221615538_Efficient_Repeating_Pattern_Finding_in_Music_Databases/links/54e614280cf2bff5a4f29302.pdf>

correlative Matrix Verfahren

<http://pdfs.semanticscholar.org/f6ce/8e49f1987d927be91c99e82458c415266c89.pdf>

Baumstruktur (suffix tree) beim auswerten (auch schon in anderen Werken benutzt)

# 7. Pattern Erkennung mithilfe von Neuronalen Netzen

Noten erforderlich -> wenn dies nicht gegeben dann mp3 in MIDI und MIDI in Noten umformen -> MIDI-Spuren einzeln eingeben, damit keine Interinstrumentaren Fehler auftreten

Einkürzen sodass nur Punkte angezeigt werden

Mit Bildoperation aus Computergrafik bearbeiten

Motif Viewer mal anschauen

# 8. Realisierung der Pattern Erkennung

Nach dem in den vorherigen Kapiteln Grundlagen der einzelnen Grundbausteine der Pattern Erkennung erklärt wurden, geht es in diesem Abschnitt um die Realisierung der einzelnen Zwischenaufgaben. Die Umsetzung des Projektes wird in Python geschehen.

Zu aller erst müssen auf das gewünschte Musikstück verschiedene Filter gelegt werden. Die benötigten Filter sind ein Low-Pass-Filter, ein Band-Pass-Filter und ein High-Pass-Filter. Als nächstes werden die Patternstreams der Beats per minute (BPM) und des Rhythmus des Musikstücks extrahiert. Danach werden die Spektrumbilder des Low- und High-Pass-Filter und die Patternstreams der BPM und des Rhythmus an ein neuronales Netz gegeben, um die Pattern des Basses, der Drum und der Clap zu erkennen.

Der darauffolgende Schritt besteht darin, dass die Audiodatei des Band-Pass-Filters so manipuliert wird, sodass die Konvertierung der .wav Datei in ein MIDI-File möglich ist. Wenn dies geschehen ist werden die im Kapitel [am Schluss aktualisieren] verschiedene Algorithmen zur Pattern Erkennung implementiert und bewertet.

Zum Schluss müssen die erkannten Pattern angezeigt werden.

# 8.1. Realisierung des Low-/Band-/High-Pass-Filter

Für die Umsetzung der drei Filter wird das aus der Pythonbibliothek stammende Packet aubio verwendet. Dieses ist zum einlesen der .wav zuständig. Da es in diesem Packet keine Funktionen für einen Low-/Band-/High-Pass-Filter gibt, muss für dessen Umsetzung ein Biquad-Filter benutzt werden.

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Quellenverzeichnis

# Eidesstattliche Erklärung

Ich erkläre hiermit, dass ich diese Masterarbeit selbstständig ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Alle den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen sind als solche einzeln kenntlich gemacht.

Diese Arbeit ist bislang keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht worden.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Leipzig, den