Hochschule Düsseldorf
University of Applied Sciences
HSD

Fachbereich Medien
Faculty of Media

WISSENSCHAFTLICHE VERTIEFUNG

Autmoatic Music Transcription

Autor:
Benedikt Kolodziej
878007
benedikt.kolodziej@study.hs-duesseldorf.de
Medieninformatik (B. Sc.)

Betreuender Professor:
Prof. Dr. Dennis Müller
dennis.mueller@hs-duesseldorf.de

Zeitraum
28.05.2025 - xy

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

1.1 Automatic Music Transcription

Musik ist seit Jahrtausenden ein zentraler Bestandteil unserer Gesellschaft. Während etwa 2000 bis 0 v.Chr. musikalische Werke meist mündlich überliefert wurden, entwickelte sich in diesem Zeitraum auch eine Notenschrift. Diese Notenschrift ermöglichte es, Musikstücke einfacher zu erlernen und einem breiteren Publikum zugänglich zu machen. Durch die Digitalisierung erhielten Digital Audio Workstations zunehmend Einzug in die Musikproduktion, wodurch Notenblätter oft nicht mehr notwendig waren und es weniger Bedarf gab diese Lieder zu übersetzen in Notenschrift.

An dieser Stelle setzt Automatic Music Transcription (AMT) an. AMT ist ein Prozess, bei dem eine Audiospur als Input gegeben wird und diese durch Computerprogramme Notenblätter oder, was weiter verbreitet ist, MIDI-Dateien als Output wiedergeben. Dabei werden durch mehrere Prozesse die Eigenschaften der Noten, zum Beispiel Frequenz oder Lautstärke, analysiert und im Kontext des Musikstückes analysiert.

1.2 Herausforderungen und Hindernisse

Anstatt das man selber diese Lieder, alleine durchs Gehör, in Notenschrift überträgt würde diese Aufgabe eine Software für einen erledigen. Dieses Ziel ist jedoch schwer zu erreichen, da Musik mehrdimensional ist durch zum Beispiel Zeit, Tonhöhe und Polyphonie. Vor allem bei polyphonen Musikstücken haben herkömmliche Algorithmen viele Schwierigkeiten. In diesen Fällen müssen sie nämlich viele verschiedene Stimmen gleichzeitig analysieren und im späteren auch die jeweiligen Töne voneinander differenzieren und eindeutig einem Instrument zuordnen. Ein weiteres Problem ist die Individualität jedes Musikstückes. In realen Aufnahmen können leichtes Rauschen, kleine Spielfehler oder stilistische Mittel wie Vibrato auftreten, die je nach Interpreten unterschiedlich klingen. Zudem sind die meisten AMT-Modelle auf westliche Tonleiter trainiert. Dies kann zu Problemen führen, wenn man zum Beispiel arabische oder indische Musikstücke transkribieren möchte.

1.3 AMT und Künstliche Intelligenz,

Um diese Vielfalt zu bewältigen, ist ein neuer, oft genutzter Ansatz, die Nutzung von künstlicher Intelligenz und Machine Learning.

Im Gegensatz zu Algorithmen ist KI flexibler und kann sich besser einstellen auf kleine Abweichungen in Musikstücken. In vorherigen Modellen wurden meist direkt elektronische Audioaufnahmen oder MIDI-Dateien verwendet, da realitätsnahe Audioaufnahmen meist zu viele Störfaktoren haben. Durch KI kann man nun mehr auf reale Audioaufnahmen zurückgreifen und das Modell somit besser anpassen für einen realistischen Gebrauch.

Auch die Mehrdimensionalität von Musik kann KI deutlich besser bewältigen als Algorithmen. Neuronale Netze besitzen eine mehrdimensionale Struktur, die es ihnen ermöglicht, verschiedene Muster, Stimmen und Eigenschaften zu erlernen. Auf der anderen Seite müssen klassische Algorithmen diese verschiedenen Dimensionen explizit modellieren und sind nicht in der Lage, Muster selbstständig zu erkennen. Sie folgen nur dem, was zuvor vom Menschen fest einprogrammiert wurde.

Um ein AMT-Modell mit KI zu kreieren, muss man sich auch für ein KI-Modell entscheiden. Hier werden meistens Recurrent Neural Networks (RNN) oder Convolutional Neural Networks (CNN), als einzelne Module, benutzt. [1] Diese Module bilden keine eigenständigen KIs, sondern lassen sich flexibel innerhalb eines Systems kombinieren. Da es keine objektiv richtigen oder falschen Notenfolgen gibt, ist der Einsatz von Reinforcement Learning nicht sinnvoll. Dementsprechend braucht man auch ein zuverlässiges Datenset aus Audiodateien und deren zugehörigen MIDI-Dateien.

RNNs sind spezialisiert, um zeitliche Abläufe besser im Kontext zu verstehen. In der Musik werden viele Noten hintereinander gespielt, diese müssen harmonisch im Stück übereinstimmen. Das RNN verarbeitet die jeweiligen Sequenzen und merkt sich die Informationen der schon gespielten Noten,

um die darauffolgenden Noten besser einordnen zu können. So lassen sich Tonfolgen harmonischen Strukturen zuordnen oder der Rhythmus des Stücks analysieren. [1]

CNNs hingegen können gut räumliche Strukturen erkennen. Das hifft uns bei der Analyse von Spektrogrammen. Meist werden die verschiedenen Frequenzen der Noten, die gespielt wurden, nach der Verarbeitung des Musikstückes in Spektrogrammen wiedergegeben. Durch die Analyse von dem Spektrogramm können gewisse Frequenzmuster erkannt werden, die dann einem bestimmten Instrument zugeordnet werden können. Das ist vor allem hilfreich dabei verschiedene Stimmen der jeweiligen Instrumente voneinander zu differenzieren.

RNNs und CNNs werden auch häufig kombiniert in AMT-Modellen. Meist in folgender Reihenfolge:

1. **CNN:** Extrahiert folgende Merkmale aus dem Spektrogramm:
    - Frequenzverteilungen und spektrale Muster
    - Tonhöhenlage und damit verbundene Obertöne
    - Klangfarbe einzelner Instrumente
    - Energieverteilung, unter anderem zur Erkennung von Toneinsätzen
    - Harmonische Strukturen wie Akkordfolgen
2. **RNN:** Verarbeitet auf Basis dieser Merkmale die zeitliche Abfolge und erkennt dabei folgende Eigenschaften:
    - Reihenfolge und Übergänge musikalischer Ereignisse
    - Beginn und Ende einzelner Töne zur Bestimmung der Notendauer
    - Rhythmische Muster und zeitliche Gruppierungen
    - Musikalische Phrasen mit zusammenhängender Struktur
    - Wiederholungen, Themen oder längere Abhängigkeiten im Verlauf
3. **Output:** Gibt das transkribierte Musikstück in strukturierter Form aus:
    - Als MIDI-Datei mit exakten Noteninformationen

Nachdem man die KI diesesen Prozess durchlaufen hat kann man mit einem eigenen oder externen Programm diese MIDI-Dateien zu standardisierter Notenschrift transkribieren.

Es gibt aber auch Projekte, wo nur ein bestimmter Teil dieser Kette mit KI-Modulen realisiert wird, oder andere KI-Module verwendet werden.

1.4 praktische Anwendungsfelder und Vorteile von AMT

AMT kann auch bei vielen anderen Problemen helfen oder in vielen Bereichen Quality-of-Life-Changes bringen. Zum einen kann der Musikunterricht spannender und interaktiver gestaltet werden. Es gibt eine breitere Auswahl von Musikstücken, die man den Schülern anbieten kann, wodurch diese durch individuell angepasste Musikstücke mehr Spaß und Ehrgeiz beim lernen haben könnten. Zudem kann man die gespielten Musikstücke der Schüler direkt beim Spielen transkribieren und gezielt erkennen, wo der jeweilige Schüler noch Verbesserungsmöglichkeiten hat. Grundsätzlich können deutlich mehr Musikstücke transkribiert werden, wodurch sich große Archive aufbauen lassen. Ein größeres Interesse an Musik wird geweckt, da Musikstücke von beliebten Serien, Filmen oder Spielen leichter für deren Musikbegeisterte Zielgruppe zugänglich sind. Allein dadurch, dass Computerprogramme Musikstücke besser verstehen, können darauf aufbauend weitere Tools für die Musikproduktion entwickelt werden. Auch KI würde davon stark profitieren. KI-generierte Musik würde verbessert werden, da die KI selber ein besseres Verständnis der Musik entwickelt. Audiobasierte Suchmaschinen könnten gewünschte Musikstücke oder bestimmte Videos präziser finden.

Musik könnte barrierefreier gestaltet werden, indem gehörlose Menschen sie lesen können und Musiker beim Spielen direktes Feedback erhalten, ob sie die Noten korrekt gespielt haben.

1.5 Motivation und Zielsetzung dieser Arbeit

(Noch nicht angefangen!) Was wird in der Arbeit behandelt, worauf liegt der Fokus (z.B. KI-Methoden für AMT), warum ist das Thema relevant (z.B. für Musiker, KI-Forschung, Musikpädagogik)?

2 Geschichte

2.1 Moorer und eines der ersten Musiktranskriptionssysteme

2.1.1 Grundlegende Probleme bei AMTs

Einer der ersten Papers über Automatic Music Transcription wurde von James A. Moorer im Jahr 1977 geschrieben. [3] In diesem beschreibt Moorer seinen Ansatz, polyphone Musik Audiospuren direkt über Computerprogramme in Notenschrift zu übertragen. Während dieses Prozesses fallen ihm schon sehr viele schwierigkeiten auf, die auch in späteren Papern und Arbeiten eine ausschlaggebende Rolle spielen werden.

Eines dieser Probleme wird von ihm als das "Cocktail-Party-Problem" bezeichnet. Dieses stellt die Schwierigkeit dar, auf einer Party bestimmten Stimmen zu folgen, während viele verschiedene Stimmen gleichzeitig erklingen. Das gleiche Problem liegt in der Noten transkription. Die meisten Musikstücke haben mehrere Instrumente, welche gleichzeitig spielen. Öfter gibt es auch Musikstücke wo es für ein Instrument, wie zum Beispiel Violine, mehrere verschiedene Stimmen gibt. Dies erschwert die Zuordnung bestimmter Noten zu einer gewählten Stimme. Schon viele Menschen scheitern deshalb daran große Musikstücke richtig zu transkribieren. Noch schwieriger wird es hier für Computerprogramme. Anfangs identifizieren diese bestimmte Töne anhand der Frequenz des Tons. Leider reicht das, wie Moorer feststellt, nicht aus um genaustens zu bestimmen, welche Töne genau momentan gespielt werden.

Jeder Ton hat Obertöne. Diese Obertöne sind jeweils das Vielfache von dem Grundton, den man spielt. Heißt, wenn ich auf einem Klavier den Ton C3 mit 130,81 Hz spiele dann hat dieser die Obertöne C4 (261,62 Hz), G4 (392,42 Hz) usw. Wenn man nur C3 spielt erklingen für das Computerprogramm auch die jeweiligen Obertöne, was man Frequenzüberlagerung nennt. Diese Frequenzüberlagerung sorgt dafür, das zum Beispiel ein Klavier anders klingt als eine Violine. Daraus resultiert dann die Klangfarbe (Timbre) eines bestimmten Instrumentes. Leider konnte Moorer zu dieser Zeit noch nicht die Klangfarbe eines Instruments erkennen. Er konnte auch noch nicht das Problem der Obertöne lösen, da die damaligen Algorithmen und Verfahren noch nicht in der Lage waren die Grundfrequenz von den Obertönen zu trennen, weshalb er sich ausschließlich auf zweistimmige polyphone Musikstücke fokussierte.

Ein weiteres Problem war Rauschen in realistischen Audiospuren und Stilistische mittel in der Musik, wie zum Beispiel Vibrato. In real aufgenommenen Audiospuren gibt es immer ein gewisses Hintergrundrauschen. Dieses kann von einem Computerprogramm auch als Note erkannt werden oder verhindern, das bestimmte Noten richtig vom Computerprogramm erkannt werden. Moorer hat ein Musikstück analog aufgenommen und dieses dann mit einem 14-Bit converter digitalisiert. Dadurch war das Rauschen nicht weg, aber da er das Musikstück selber aufgenommen hat und dieses konvertiert hat sorgte es für insgesamt geringeres Rauschen. Zudem konnte er so die Musiker davon abhalten bestimmte Stilistische mittel zu verwenden, um bessere Daten zur Transkription zu erhalten. Stilistische mittel, wie Vibrato, konnten nicht genutzt werden, da diese eine kleine aber kontinuierliche Veränderung der Frequenz verursachen. Dadurch kann das Computerprogramm nicht korrekt erkennen, das eigentlich eine einzelne Note gespielt wurde. Somit war der Onset und Offset der Note komplett falsch.

Das letzte Problem, was Moorer angesprochen hat, ist das Nutzen von nicht harmonischen Instru-

menten wie Trommeln oder einem Schlagzeug. Diese Instrumente haben keinen eindeutigen Pitch für deren Töne, sie sind eher abhängig von Rhythms und Lautstärke. Da Moorers AMT sich jedoch auf das Frequenzmuster der Noten fokussiert, können diese Musikinstrumente nicht berücksichtigt werden.

2.1.2 Der Aufbau von Moorers AMT-Systems

Moorers automatische Musiktranskriptionssystem war eins der ersten seiner Art. Viele weiteren AMT-System leiten sich von diesen ab.

Zunächst wird ein analoges Musiksignal mit einem 14-Bit converter digitalisiert. Dieses digitale Musiksignal wurden dann genutzt um mithilfe von Bandpassfiltern, ein Filter welcher nur bestimmte Frequenzen durchlässt, bestimmte Frequenzbereiche zu isolieren. Dadurch konnte Moorer die gespielte Note und deren Dauer, also zugleich auch deren Onset und Offset, feststellen.

Nun mussten die bestimmten Noten einer gewählten Stimme zugeordnet werden. Dies wurde durch melodische Gruppierung verwirklicht. Zunächst wurden Inseln gebildet. Inseln sind Noten die sich Zeitlich vollständig überlappen. Wir gehen davon aus das jede Stimme nur eine Note gleichzeitig spielt, wodurch diese Noten nicht der gleichen Stimme zugeordnet werden können. Als Nächstes müssen die anderen Noten auf verschiedene Kombinationen getestet werden. Desto kleiner die Frequenz sprünge je Note sind, desto wahrscheinlicher gehören sie einer Stimme zu. Zudem werden Gruppierungen von Noten erstellt, welche am wahrscheinlichsten harmonisch nacheinander gespielt worden.

Zum Schluss ließ Moorer die gewonnenen Daten durch ein Programm laufen, um diese dann mithilfe eines Plotters in eine Notenschrift umzuwandeln.

2.2 MIDI-Dateien

Notenschrift als Input für ein Computerprogramm, Synthesizer oder ähnliches ist unhandlich. Zunächst müsste man die gespielten Noten immer wieder zu Notenschrift konvertieren und danach diese auch noch in anderen Programmen analysieren, was sehr aufwändig werden würde. Eine bessere lösung dafür wäre eine Datenschreibweise, bei der alle wichtigen Informationen bestimmter Noten übersichtlich aufgeschrieben sind. MIDI-Dateien sind dafür perfekt geeignet.

MIDI ist ein Standartprotokoll zur kommunikation zwischen elektronischen Musikinstrumenten, Computern und anderen Geräten wie zum Beispiel Synthesizer. Dieses Protokoll wurde 1983 erstmals eingeführt und wurde schnell zu einem Standard in der digitalen Musikindustrie. [2] In MIDI-Dateien werden Daten von Tönen gelagert, welche zum Beispiel zuvor von einem elektronischen Instrument gespielt wurden oder durch AMTs erfasst wurden.

In MIDI-Dateien werden folgenden Daten gespeichert:

1. **MIDI Header-Chunk (MThd):** Enthält grundlegende Informationen zur Struktur der Datei:
    - Formattyp (0 = eine Spur, 1 = mehrere synchrone Spuren, 2 = unabhängige Spuren)
    - Anzahl der folgenden MTrk-Blöcke (Tracks)
    - Zeitauflösung (Ticks pro Viertelnote)
2. **MIDI Track-Chunks (MTrk):** Jede Spur enthält eine zeitlich sortierte Liste von MIDI-Events:
    - **MIDI-Events:**
        - Note Onset / Offset
        - Control Change (Lautstärke)
        - Program Change (gibt das spielende Instrument an)
        - Pitch Bend (verändert die Tonhöhe)

  - Aftertouch / Polyphonic Key Pressure (Druckstärke pro Taste)
- **Meta-Events:**
  - Set Tempo (Tempo in Mikrosekunden pro Viertelnote)
  - Time Signature (Taktart zum Beispiel 4/4 oder 3/4)
  - Key Signature (Tonart zum Beispiel C-Dur oder A-Moll)
  - Track Name
  - Lyrics
  - Markers
  - End of Track (Ende einer Spur)
- **System Exclusive Events (SysEx):**
  - Herstellerspezifische Daten wie Synthesizer-Presets oder Spezialbefehle

3. **Delta-Time:** Gibt die Zeit (in Ticks) an, die seit dem letzten Event vergangen ist:
   - Ermöglicht die genaue zeitliche Positionierung jedes MIDI-Events
   - Grundlage für das Timing und die rhythmische Struktur der Datei

Am wichtigsten sind dabei die MTrk-Blöcke, in denen die Daten der einzelnen Noten gespeichert werden. Dabei stellt ein Track die Eventliste einer ganzen Stimme dar, wie zum Beispiel die Melodiestimme, eine Violine, die Pedalsteuerung eines Klaviers oder Metadaten. Es fällt auf das diese vier Beispiele alle sehr unterschiedliche Aufgaben und bedeutungen haben. Das liegt daran, dass in MIDI-Dateien eher zusammenhängende Funktionen gespeichert werden und nicht nur Musiknoten.

MIDI-Dateien kamen auch der Forschung für AMT-Systemen sehr gelegen, da man nun ein standardisiertes output Format für diese Programme besaß. Später werden diese zudem sehr Essenziell bei dem Training KI basierter AMT-Systeme.

2.3

3 Fazit

Abschließende Bemerkungen, Reflexion und Ausblick.

Literaturverzeichnis

[1] Sebastian Böck und Markus Schedl. “Polyphonic Piano Note Transcription with Recurrent Neural Networks”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Zugriff am 19.05.2025. 2012, S. 121–124. URL: https://www.cp.jku.at/people/schedl/Research/Publications/pdf/boeck_schedl_icassp_2012.pdf.

[2] smith dave und wood chet. “the ’usi’, or universal synthesizer interface”. In: *journal of the audio engineering society* 1845 (Okt. 1981).

[3] James A. Moorer. “On the Transcription of Musical Sound by Computer”. In: *Computer Music Journal* 1.4 (1977). Zugriff am 19.05.2025, S. 32–38. URL: https://www.jamminpower.org/PDF/JAM/Transcription%20of%20Musical%20Sound%20-%20CMJ%201977.pdf.