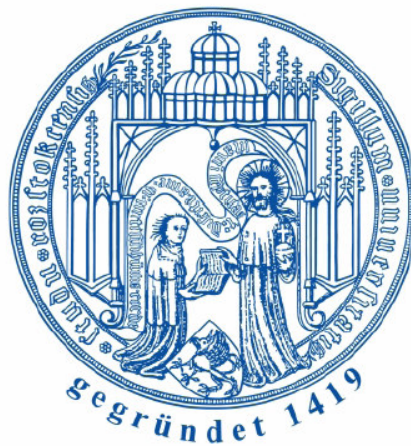


Melodie-Suche in einem Archiv von historischen Notenhandschriften



Studienarbeit

Universität Rostock
Institut für Informatik

angefertigt von: Katrin Ziehe
geboren am: 05.02.1982 in Lutherstadt Wittenberg
Gutachter: Prof. Dr. Andreas Heuer
Betreuer: Dipl.-Ing. Andreas Finger
Dipl.-Ing. Temenushka Ignatova
Starttermin: 15.04.2005

Zusammenfassung

In den letzten Jahren entstand in der Musikwissenschaft der Wunsch, Musikkompositionen auch auf digitalem Wege zu archivieren. Bisher ließen sich Musikfragmente nur mühsam und zeitaufwendig einem Komponisten per Hand zuordnen. Mithilfe von Music Information Retrieval - Systemen (MIR - Systeme) soll diese Aufgabe nun vom Computer übernommen werden.

In dieser Studienarbeit wurden zum einen drei unterschiedliche MIR - Konzepte untersucht und vorgestellt. Zum Anderen sollte bei einem Vergleich dieser drei Systeme ihre Vor- und Nachteile aufgedeckt werden. Letztendliches Ziel war es ein geeignetes Konzept für das eNoteHistory - Projekt zu entwickeln.

Abstract

Over the last years in Music-Science the wish arosed, to archive musical compositions in a digital way. Until now musical fragments could only be allocated to a composer by hand in a tedious and time-consuming manner. With help of Music Information Retrieval - systems (MIR - systems) this task has to be done by the computer.

At first three different MIR - concepts have been researched and introduced to the reader in this seminar paper. In a comparison of these three systems their advantages and disadvantages had to be discovered. The final aim was the development of an adequate concept for the eNoteHistory - project.

Inhaltsverzeichnis

1 Einleitung	7
1.1 Thema der Studienarbeit	7
1.2 eNoteHistory	7
2 Einführung in die Musiktheorie und Problembetrachtung.....	8
2.1 Kurzvorstellung der 4. Hauptsuchmethoden.....	8
2.2 Parameter eines Musikstückes	9
2.3 Notensystem/ Tonsystem	10
2.4 Symbolische Repräsentation von Musikstücken.....	11
2.5 Monophonie und Polyphonie	14
3 Konkrete Konzepte.....	15
3.1 C-Brahms	15
3.1.1 Aufbau.....	15
3.1.2 Die Suchmethoden	16
3.1.3 Methode P1	18
3.1.4 Zusammenfassung.....	18
3.2 Themefinder	19
3.2.1 Aufbau.....	19
3.2.2 Humdrum-Code.....	20
3.2.3 Zusammenfassung.....	23
3.3 Musipedia	24
3.3.1 Aufbau.....	24
3.3.2 Parsons-Code.....	25
3.3.3 Parsons-Code-Editierdistanz	25
3.3.4 Zusammenfassung.....	26
4 Vergleich der Konzepte.....	29
5 Konzept für eNoteHistory	32
6 Zusammenfassung und Ausblick	38

1 Einleitung

1.1 Thema der Studienarbeit

Aufgabe dieser Studienarbeit ist es, ein passendes Music-Information-Retrieval Konzept für das eNoteHistory-Projekt zu finden. Das eNoteHistory-Projekt dient der Speicherung historischer Notenhandschriften. Mit Hilfe eines Music-Information-Retrieval (MIR) Programms soll nun auch die Suche auf den Incipits der Musikstücke ermöglicht werden.

Zunächst wird in dieser Arbeit auf die allgemeinen Probleme bei MIR Konzepten eingegangen. Anschließend wird ein kurzer Überblick über die vier Hauptsuchmethoden gegeben.

Um ein passendes Konzept für das Projekt eNoteHistory zu finden, ist es ein Teil der Studienarbeit, vorab bestehende Music Information Retrieval Programme zu untersuchen.

Es gibt eine Reihe von web-basierten und frei zugängigen Programmen. Diese Arbeit beschränkt sich auf drei verschiedene Programme und ihre jeweiligen Konzepte.

Das erste hier vorgestellte Konzept, ist das von der Universität Helsinki zur Suche von Musikstücken entwickelte Programm: „C-Brahms“.

Als zweites wird das Programm „Themefinder“, ein Projekt der Universitäten von Stanford und Ohio, betrachtet.

Abschließend wird auf das freizugängliche Programm „Musipedia“ und seine Konzepte eingegangen.

Im anschließenden Kapitel werden die drei Konzepte miteinander verglichen. Es wird versucht, die Frage zu beantworten, welches der vorgestellten Konzepte am Besten geeignet ist, Musikstücke zu suchen.

Abschließend wird der Entwurf eines Konzeptes zum Suchen von Musikdaten im Projekt eNoteHistory vorgestellt.

1.2 eNoteHistory

Das eNoteHistory-Projekt ist eine gemeinschaftliche Arbeit des Fraunhofer Instituts für graphische Datenverarbeitung in Rostock, dem Institut für Musikwissenschaften und dem Lehrstuhl für Datenbank- und Informationssysteme der Universität Rostock. Ziel ist es, Musikstücke des 18. Jahrhunderts in einer dafür vorgesehenen Datenbank digital zu archivieren und den Zugriff auf die Informationen mit verschiedenen Werkzeugen zu ermöglichen.

Für den Retrieval-Prozess werden die einzelnen Incipits der Musikstücke verwendet. Ein Incipit umfasst die ersten 4-5 Takte einer Komposition, welche im eNoteHistory-Projekt im Midi-Format abgespeichert werden. Zurzeit liegen dem Projekt die Incipits für 6000 historische Musikstücke vor.

Weitere Informationen sind unter der Projektseite: www.enotehistory.de zu finden.

2 Einführung in die Musiktheorie und Problembetrachtung

2.1 Kurzvorstellung der 4. Hauptsuchmethoden

Man unterscheidet vier verschiedene Hauptsuchmethoden. Für die eigentliche Aufgabe sind sie weniger von Interesse, aber für ein besseres Verständnis der Thematik werden sie hier kurz vorgestellt. [4]

1. Suchen auf Metadaten
2. Suche auf akustischen Merkmalen
3. Suche in gesprochenem Text
4. Suche mit Noten

1. Suche auf Metadaten

Bei dieser Suchmethode handelt es sich nicht im eigentlichen Sinne um ein Music Information Retrieval Konzept. Die Suche auf Metadaten basiert im Wesentlichen auf dem Testretrievalverfahren. Die eigentlichen Audiodaten werden bei dieser Art der Suche nicht berücksichtigt. In einer Datenbank werden neben den Audiodaten noch zusätzliche Attribute, wie Informationen über den Titel, den Interpreten und andere wichtige Daten abgelegt. Nach diesen, beziehungsweise auf diesen Informationen wird die Suchanfrage letztendlich gestartet. Beispiele für die Suche auf Metadaten gibt es an vielen Stellen, z.B. file-sharing Programme basieren auf dieser Art der Suche.

2. Suche auf akustischen Merkmalen

Hierbei erfolgt die Suche auf den Rohdaten des Audiosignals. Die Signalinformationen werden mittels charakteristischen Merkmalen, wie Tonhöhe, Lautstärke und Intensität beschrieben. Dafür wird das Audiosignal in Intervalle aufgeteilt und zu jedem Intervall die gewünschten Merkmale gespeichert. Je nach Programm erhält man einen bis zu 12-dimensionalen Vektor, der die akustischen Eigenschaften der Melodie repräsentiert. Mit Hilfe der Fast Fourier Transformation werden die Werte in zeitabhängigen Vektorfunktionen dargestellt, auf denen dann die Suchfunktionen angewendet werden.

3. Suche in gesprochenem Text

Da die Suche im gesprochenen Text in der Studienarbeit keine Rolle spielt, wird sie an dieser Stelle nur kurz umrissen. Die Suche in gesprochenem Text wird unterschieden, in die Suche mit Phonemen und in die Suche mit Wörtern. Bevor auf der Audiodatei aber gesucht werden kann, müssen zuerst alle unwichtigen Nebengeräusche herausgefiltert werden. Bei der Suche mit Phonemen wird die Anfrage getippt oder gesprochen. Die eingetippten Wörter müssen umgewandelt werden. Die Suche mit Wörtern erfordert die Benutzung eines kontrollierten Wörterbuches. Wortgrenzen, Firmen-, Eigen- und Ländernamen werden bei dieser Art der Suche nur schwer oder gar nicht erkannt.

4. Suche mit Noten

Bei dieser Suchmethode wird direkt nach den Noten gesucht, aber auch nach dem Takt oder dem Tempo. Man hat eine Melodie im Kopf, weiß aber nicht zu welchem Musikstück sie gehört. Ein Ansatz bei der Suche nach solchen Melodien ist das Query by Humming. Es gibt im Internet Programme, in denen man die Melodie in ein Mikrophon summt und das Programm die Datenbank nach dem passenden Musikstück durchsucht.

Andere Programme verwenden für die Suche den Parsons-Code, eine Folge von „D“, „U“ und „R“ für die Suche nach dem passenden Musikstück.

D - Down, U - Up, R - Repeat

Dabei muss der Suchende angeben, ob der nächste Ton in seiner Melodie höher, tiefer oder gleich des vorangegangenen Tones ist. Meist wird das Musikstück bei dieser Art des MIR schon als Folge D-R-U abgespeichert, was die Suche darauf wesentlich vereinfacht. (siehe Kapitel 3.3.2)

Da die Suche mit Noten die Suchmethode ist, die hauptsächlich Anwendung in der Studienarbeit findet, werden im Laufe der Ausarbeitung noch andere Ansätze vorgestellt und die Methoden anhand von konkreten Programmen genauer erläutert.

2.2 Parameter eines Musikstückes

Ein Musikstück besteht aus verschiedenen Parametern, die unterschiedlich wichtig für das Erkennen von Musikstücken sind. [1]

50% Tonhöhe
40% Rhythmus + Tondauer
10% Klangfarbe + Dynamik

Die Tonhöhe ist die wichtigste Eigenschaft eines Tones. Die Veränderung eines Tons lässt sich durch Abstraktion von anderen Toneigenschaften wie Dauer, Lautstärke oder Klangfarbe als „lineare“ oder „zyklische“ Veränderung wahrnehmen. Töne werden als „höher“ empfunden, wenn ihre Klangfarbe „heller“ ist. Die „Höhe“ eines Tones ist demnach eine relativ willkürliche Bezeichnung. Das Tonhöhenempfinden ist umso höher, je größer die Frequenz einer Schwingung ist. Des Weiteren wird die Empfindung der Tonhöhe durch die Schallintensität und die Dauer beeinflusst. [2]

Neben den allgemeinen Tonhöhenangaben ist der Rhythmus, ein grundlegendes musikalisches Strukturelement. Mit Rhythmus bezeichnet man Ordnung, Gliederung und sinnfällige Gestaltung des zeitlichen Verlaufs von Melodien. Er wird häufig mit Metrum oder Takt verwechselt. [2]

Die Klangfarbe eines Tones ist abhängig von seinem Klangspektrum, welches sich aus Art, Zahl und Intensität der Teiltöne zusammensetzt. [2]

Die Dynamik ist die Differenzierung der Tonstärke. Sie ist entweder eine stufenweise Veränderung (z.B. forte, mezzoforte, piano) oder eine allmähliche (crescendo, decrescendo). [2]

Die Klangfarbe und die Dynamik machen zusammen nur 10% der Parameter eines Musikstückes aus und werden darum im weiteren Verlauf nicht näher betrachtet.

2.3 Notensystem/ Tonsystem

Zum besseren Verständnis der nachfolgenden Kapitel sollen an dieser Stelle, die Begriffe Noten-, bzw. Tonsystem, Ornamentik und Transposition erläutert werden.

Das Notensystem ist ein System, dass aus 5 Linien besteht. In diesen Linien werden die Noten des Tonsystems angeordnet. Man verwendet für die Anordnung der Notenzeichen sowohl die Linien, als auch die Zwischenräume. Sollten die Zeilen nicht ausreichen, so verwendet man Hilfslinien. Zur Festlegung der Tonhöhe beginnt jedes Notensystem mit einem Notenschlüssel. Man unterscheidet zwischen verschiedenen Arten von Schlüsseln.

Die für diese Arbeit relevanten Notenschlüssel sind der Violin- und der Bassschlüssel.



Abbildung 2.1: Violinschlüssel und Bassschlüssel

Grundlage des Tonsystems ist die Tonleiter. Ein Beispiel dafür ist die C-Dur-Tonleiter, sie setzt sich aus den Tönen: c-d-e-f-g-a-h zusammen.

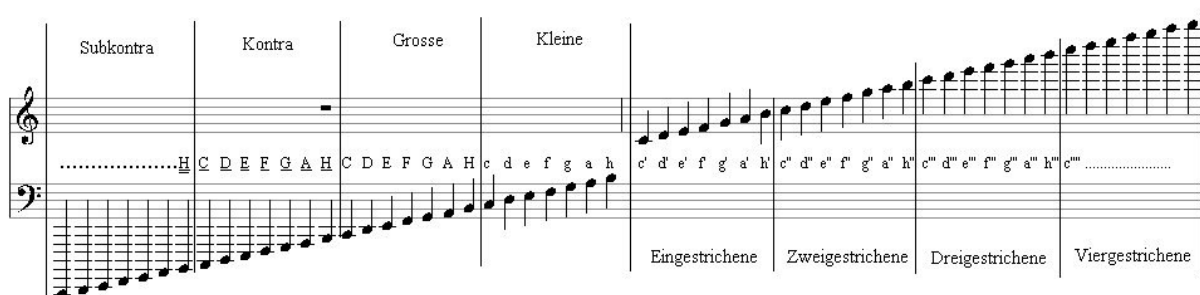


Abbildung 2.2: Tonsystem mit Oktaveneinteilung

Die Abbildung 2.2 zeigt das Tonsystem, welches in Oktavenräume eingeteilt ist. Diese Räume gehen jeweils vom Ton c bis zum Ton h, und haben alle unterschiedliche Namen (z.B.: Kleine Oktave). Als Oktave bezeichnet man ein musikalisches Intervall. Nimmt man einen festen Ton und halbiert, bzw. verdoppelt dessen Frequenz, so erhält man (subjektiv empfunden) einen gleich klingenden Ton, nur eine Oktave tiefer oder höher. Ein Beispiel für solche Töne wäre c' und c''. In der Regel entspricht die Länge der Tonleiter genau einer Oktave und wird dann im Oktavenabstand wiederholt.

Ein anderes Wort für Ornamentik ist Verzierung und bezeichnet in der europäischen Musik die Ausschmückung einer Melodie. Diese Ausschmückungen werden meistens durch besondere Zeichen oder kleinere Noten angedeutet.

Ein Musikstück zu transponieren (Transposition) heißt, es intervallgetreu in eine andere Tonart zu versetzen. Zum Beispiel kann man ein Musikstück von D-Dur nach F-Dur transponieren. Eine andere Art der Transposition ist die Oktavierung, wobei die Töne gleich bleiben, nur eine Oktave höher oder tiefer gespielt werden.

Genauere Definitionen der oben beschriebenen Begriffe findet man im Schülerduden Musik, des Dudenverlages [2].

2.4 Symbolische Repräsentation von Musikstücken

Die symbolische Repräsentation von Musikstücken bietet einige Schwierigkeiten. Oftmals werden bei der Repräsentation wichtige Informationen weggelassen. Es werden zum Beispiel nur die Tonhöhenangaben codiert, aber Angaben über die Einsatzzeit und die Dauer einer Note werden weggelassen. Einsatzzeit und Dauer eines Signals sind aber wichtige Bestandteile des Rhythmus.

Somit sollten auch die Einsatzzeit und die Dauer der Note in die symbolischen Repräsentation eingehen.

Die nachfolgend beschriebenen Erläuterungen beziehen sich auf das Vorlesungsskript von Prof. Dr. Michael Clausen. [3]

Die klassische Methode zur Datenmodellierung ist es, dem Musikstück eine Zeichenkette zuzuordnen. Diese Zeichenkette kann zum Beispiel aus der Angabe der Tonhöhen bestehen.



Abbildung 2.3: monophone Notenfolge

Nach der klassischen Methode würde man der Notenfolge aus der Abbildung 2.3 folgende Zeichenkette zuordnen: **fagff**.

Was bei monophonen Stücken noch nicht so große Probleme bereitet, ist bei polyphonen Musikstücken schon recht schwierig.

Die beiden Noten **a** und **c**, in Abbildung 2.2 sind zeitlich gleichberechtigt und müssen dementsprechend dargestellt werden. Modelliert man sie wie oben gezeigt (z.B: {a,c}g), werden aber wichtige Musikinformationen, wie Tondauer und Einsatzzeit weggelassen.



Abbildung 2.4: polyphone Notenfolge

Die Lösung für dieses Problem ist, statt das Musikstück als Folge von Tonhöhenangaben zu codieren, Mengen elementarer Notenobjekte zu betrachten. Es werden zu jeder Note im Musikstück, neben der Tonhöhe, auch noch explizit alle wichtigen Parameter angegeben. Für die Konzepte in dieser Arbeit ist besonders die Einsatzzeit ein wichtiger Punkt und soll somit hier in die Codierung eingehen.

Um diese Codierung zu erreichen, werden die Noten eines Audiosignals als Tupel moduliert, bestehend aus Tonhöhe p und Einsatzzeit t . [3]

$$[t, p] \in \mathbb{Z} \times [0 : 127] =: M$$

Anhand dieser Repräsentation ist es möglich, neben der Tonhöhe auch die Einsatzzeit zu berücksichtigen. Neben der Repräsentation der Audiosignale ist es auch nötig, bestimmte Anforderungen an die Anfrage zu stellen. Die einzelnen Programme stellen ganz unterschiedliche Anfragen. Allgemein wollen wir hier festlegen, dass eine Anfrage ein Notenfragment oder Ausschnitt aus einem Musikstück ist und dieses Musikstück durch unsere Anfrage lokalisiert werden soll.

Da die Anfrage also ein Teil eines Musikstückes ist, definieren wir:

$$Q = \{[t_1, p_1], \dots, [t_n, p_n]\}. [3]$$

Eine andere symbolische Repräsentation ist die Klavierwalzendarstellung, oder auch Piano-Roll-Notation. Dabei werden die Noten einer Melodie als Rechtecke im 2-dimensionalen Raum dargestellt. Auf diese Weise lassen sich die wichtigsten Eigenschaften einer Note darstellen, das sind Tonhöhe, Tondauer und Einsatzzeit. Wie oben beschrieben, sind dies die besonders zu berücksichtigenden Eigenschaften einer Note.

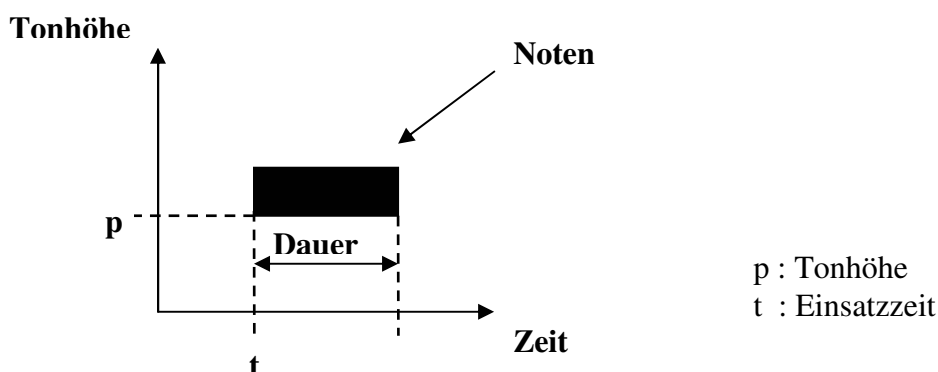


Abbildung 2.5: Klavierwalzenton

Das Beispiel zeigt wie die Klavierwalzendarstellung für ein Melodiefragment aussieht.



Abbildung 2.6: Notenbeispiel Anfrage

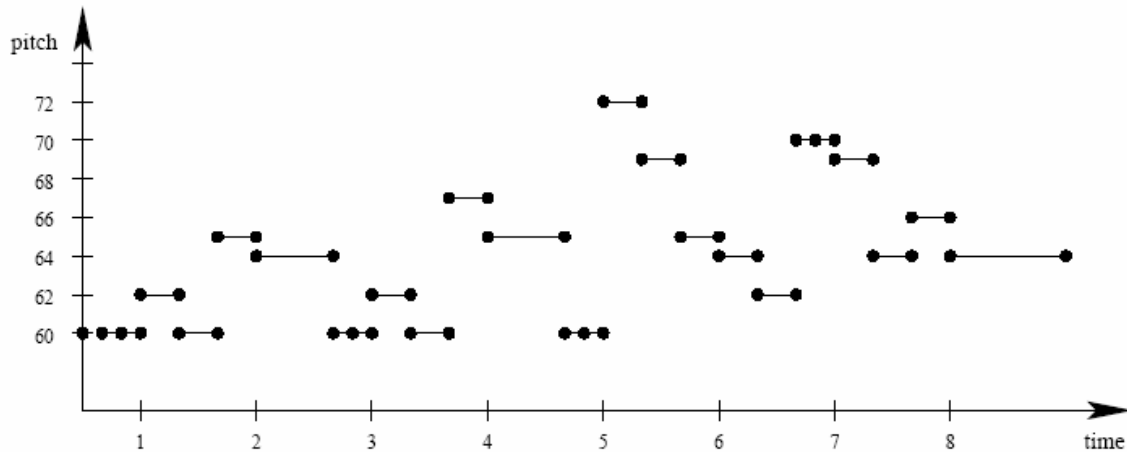


Abbildung 2.7: Klavierwalzendarstellung Anfrage

Das Beispiel zeigt eine etwas andere Darstellung der Noten. Die Note besteht hier aus einem Start- und Endpunkt und die Dauer der Note wird durch die Länge der Linie wiedergegeben. Die Muster der einzelnen Incipits in der Datenbank werden mit dem Anfragemuster verglichen. Angenommen das Melodiefragment aus Abbildung 2.6 ist die Anfrage, so wird ein MIR-System folgendes Ergebnis liefern.

Abbildung 2.8: Notenbeispiel Resultat



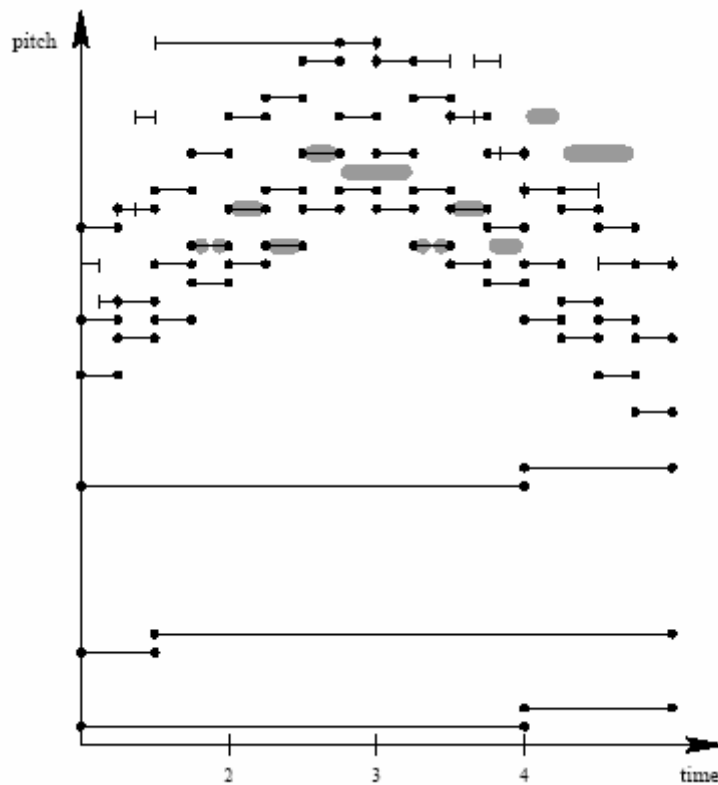


Abbildung 2.9.: Klavierwalzendarstellung Resultat

Die grauen Flecken in Abbildung 3.6 repräsentieren die ersten 11 Noten aus der Anfrage. Es ist zu sehen, dass nicht alle Elemente der Anfrage mit den Elementen des gefundenen Musikstückes übereinstimmen. In Kapitel 3 werden Methoden vorgestellt, die Incipits auch dann als Treffer identifizieren, wenn sie nicht genau mit der Anfrage übereinstimmen.

2.5 Monophonie und Polyphonie

Wie das Beispiel im vorherigen Abschnitt zeigt, lassen sich mit der Klavierwalzendarstellung auch polyphone Musikstücke wiedergeben. Die Noten, welche zur gleichen Zeit gespielt werden, sind im gleichen Zeitintervall dargestellt.

Beim eNoteHistory-Projekt handelt es sich um polyphone Notenhandschriften, die mit Hilfe der Klavierwalzendarstellung repräsentiert werden könnten. Probleme entstehen allerdings bei der Suchanfrage, es handelt sich um eine monophone Anfrage.

Wie finden wir ein monophones Musikfragment in einem polyphonen Musikstück?

Eine Lösung ist bei polyphonen Incipits, die einzelnen Stimmen jeweils mittels eines eigenen Incipits abzuspeichern. Diesen Ansatz verwenden auch die im nächsten Kapitel vorgestellten Systeme. Nur bei C-Brahms gibt es Suchmethoden, die eine polyphone Anfrage erlauben und auch auf polyphonen Incipits suchen. Näheres dazu in Kapitel 3.1.

Eine genauere Beschreibung der Probleme von Music Information Retrieval liefert die Veröffentlichung von Tim Crawford und Donald Byrd. [13]

3 Konkrete Konzepte

Wie bereits in der Einleitung erwähnt, sollen an dieser Stelle drei konkrete Konzepte vorgestellt werden. Es gibt eine große Anzahl von Musiksuchmaschinen im Internet. Die drei hier vorgestellten Systeme sind bekannte Konzepte und verfügen über eine Datenbank, die uns bekannte Musikstücke enthält. Des Weiteren haben die drei Systeme unterschiedliche Methoden entwickelt, um die Suche nach Incipits zu ermöglichen.

Die hier vorgestellten Konzepte speichern nicht das komplette Musikstück und führen die Suchanfrage darauf aus. In der Datenbank werden einzelne Themen oder die Anfangstakte eines Musikstückes gespeichert. Es ist also nicht möglich auf dem gesamten Musikstück zu suchen, sondern die Suche beschränkt sich auf einzelne Abschnitte.

3.1 C-Brahms

Das Projekt C-Brahms wurde in Finnland entwickelt und ist eine frei zur Verfügung stehende Musiksuchmaschine. Sie wurde im Januar 2002 eingeführt und ist ein Gemeinschaftsprojekt der Universität Helsinki und der Technologischen Universität Helsinki

Nach eigenen Angaben ist das Ziel von C-Brahms, polyphone Musikstücke in großen Musikdatenbanken zu finden. Die Musik soll dabei in symbolischer Codierung vorliegen. [5]

3.1.1 Aufbau

Bei C-Brahms hat man die Auswahl zwischen 9 verschiedenen Suchmethoden, sowohl für monophone als auch für polyphone Musikstücke. Die Methoden unterscheiden sich sehr voneinander. So werden bei einigen Algorithmen mögliche Eingabefehler mitbetrachtet, bei anderen muss die gesuchte Melodie genau angegeben werden, um korrekte Ergebnisse zu erzielen. (siehe Kapitel 3.1.2)

Die Anfrage bei C-Brahms erfolgt über die Klaviertastendarstellung oder über die Eingabe der Tonhöhe (Notenname, z.B.: C, E, ...) mittels Computertastatur. Bei der Eingabe der Melodie über die Klaviertasten, wird der Rhythmus der gesuchten Melodie soweit vernachlässigt, dass bei der Tondauer immer von 1/4 Noten ausgegangen wird. Mehr Einfluss auf den Rhythmus hat der Nutzer, wenn er die Notennamen mit der Computertastatur eingibt. Dann kann er durch das Angeben einer Zahl vor dem Notennamen bestimmen ob es sich zum Beispiel um eine 1/2, eine 1/4 oder eine 1/16 Note handelt. (z.B.: 4C, 8C, 16C, ...)

Die ganze Eingabe des Rhythmus ist sinnlos, wenn man die falsche Suchmethode auswählt. (siehe Kapitel 3.1.2)

Des Weiteren hat der Nutzer die Möglichkeit, neben der Suchmethode noch andere Einstellungen vorzunehmen, bevor er eine Anfrage tätigt. So kann er die Sortiermethode und die Anzahl der Resultate wählen, aber auch bestimmen, ob die gesuchte Melodie am Anfang eines Musikstückes oder überall vorkommen kann.

Hat man unter C-Brahms eine Anfrage gestellt, werden die gefundenen Melodien je nach Sortiermethode aufgelistet. Der Nutzer hat jetzt die Möglichkeit sich einen längeren Auszug aus dem passenden Musikstück anzuhören. Außerdem erhält er umfangreiche Informationen zum Musikstück, zum Beispiel kann er sich die Notenschrift des Musikstückes ansehen.

Bei C-Brahms besteht die Möglichkeit, eine polyphone Anfrage zu stellen. Dabei werden die polyphonen Noten mit einem „+“ verbunden. (z.B.: 4C5+4D5)
Wichtig hierbei ist, dass die richtige Suchmethode angegeben wird, denn nicht jede Methode eignet sich zum Suchen mit polyphoner Anfrage. (siehe Kapitel 3.1.2)

3.1.2 Die Suchmethoden

Der Nutzer hat bei C-Brahms eine große Auswahl an verschiedenen Suchmethoden. Ihm stehen 9 verschiedene Methoden zur Verfügung, die alle Vor- und Nachteile haben. Die nachfolgende Tabelle gibt einen Überblick über die Suchmethoden und ihre speziellen Eigenschaften.

Name	Transpositions- beständigkeit	Monophone Muster	Polyphone Muster	Rhythmus	Ornamentik	Fehler- zulassung
1. P1	x	x	x	x	x	
2. P2	x	x	x	x	x	x
3. P3	x	x	x	x	x	x
4. MonoPoly	x	x				
5. IntervalMatching	x	x				
6. ShiftOrAnd		x				
7. PolyCheck		x	x			
8. Splitting	x	x			x	
9. LCTS	x	x			x	x

Tabelle 3.1: Suchmethoden

Die ersten drei Suchmethoden liefern gute Trefferergebnisse. Allerdings sind besonders P1 und P2 sehr schwer zu handhaben. Diese Suchmethoden sollten nur eingesetzt werden, wenn der Rhythmus der gesuchten Melodie exakt bekannt ist. Sonst kann es sein, dass die gewünschte Melodie nicht in der Trefferliste aufgeführt wird. Die drei Suchmethoden unterscheiden sich insofern, dass man bei P1 sehr genau Tonhöhe und Tondauer der gesuchten Melodie angeben muss. Alle Elemente der Anfrage, müssen mit den Elementen der gefundenen Melodie übereinstimmen. Wenn dieses Wissen nicht vorhanden ist, kann der Nutzer die Methoden P2 oder P3 verwenden. Bei der Methode P2 müssen nicht mehr alle Elemente der Anfrage im Resultat auftauchen. Die höchstmögliche Übereinstimmung soll erzielt werden. Die Methode P3 geht noch einen Schritt weiter, da bei ihr mögliche Fehler in der Anfrage toleriert werden. Wenn der Nutzer also eine Tonhöhe falsch angibt, kann die passende Melodie durchaus noch gefunden werden. Allerdings wird die Trefferliste sehr lang sein, so dass die gewünschte Melodie nicht zwangsläufig in den angegebenen Treffern zu finden sein wird. Für die Berechnung der Treffer verwenden diese Methoden die Klavierwalzendarstellung und die dazugehörigen Algorithmen. (siehe Kapitel 3.1.3)

Die nächsten drei Methoden (4-6) bauen aufeinander auf. Sie alle benutzen für die Berechnung der Suchanfrage Funktionen der Vektormathematik. Wobei die MonoPoly-Methode, die am weitesten entwickelte der drei ist. Die Entwicklung begann mit dem ShiftOrAnd-Algorithmus, welcher die Operationen And und Or verwendet, um die Anfrage mit der Datenbank zu vergleichen. Da der ShiftOrAnd-Algorithmus keine Transpositionsbeständigkeit aufweisen konnte, wurde er zum IntervalMatching-Algorithmus weiterentwickelt. Zusätzlich zum ShiftOrAnd-Algorithmus kommt der DirectCheck-Algorithmus hinzu. Dieser gleicht ab, ob an einer gegebenen Position die jeweils gesuchte Note vorhanden ist oder nicht.

Eine Verbesserung hinsichtlich der Laufzeit bietet der MonoPoly-Algorithmus. Da die Suche auf einer statischen Datenbank betrieben wird, kann zusätzlich zur Filterphase eine Vorverarbeitungsphase eingebunden werden.

Zur Methode Nummer 7, dem PolyCheck, bietet C-Brahms keine Erklärungen an.

Das Splitting (Methode 8) ist eher theoretischer Natur. Es wird versucht, versteckte Melodien zu finden. Hierbei wird das Muster der Anfrage Stimmen übergreifend, wie unten im Bild dargestellt, gesucht. Man kann sowohl die Anzahl der Lücken im Muster, als auch die maximale Anzahl der Sprünge zwischen den Stimmen definieren. Im Beispiel unten taucht das Anfragemuster zweimal im Suchergebnis mit jeweils einem Stimmwechsel auf.



Fig. 1. A monophonic query melody.

Selim Palmgren: Serenade

Fig. 2. An excerpt of a four-track vocal composition. The query melody in Fig. 1 has two transposed occurrences such that $\kappa = 2$ and $\alpha = 0$.

Abbildung 3.1: Splitting

Die LCTS-Methode ist transpositionsbeständig, außerdem lässt die Methode Ornamentik und Fehler in der Anfrage zu.

Genauere Informationen, bzw. die mathematischen Grundlagen der verschiedenen Methoden, auf Internetseite von C-Brahms zu finden. [6]

3.1.3 Methode P1

In diesem Abschnitt wird die P1 Methode von C-Brahms als Vertreter für die 9 verschiedenen Suchmethoden, etwas näher erläutert.

Die Methode P1 verwendet zur symbolischen Repräsentation ihrer Musikstücke, die im Kapitel 2 vorgestellte Klavierwalzendarstellung. Eine Note liegt bei dieser Art der Repräsentation im euklidischen Raum \mathfrak{R}^2 . Sie verfügt über einen Startpunkt $s = (s_x, s_y) \in \mathfrak{R}^2$ und Endpunkt $s' = (s'_x, s'_y) \in \mathfrak{R}^2$. Der Abstand zwischen diesen Punkten wird als Segment $[s, s']$ bezeichnet und repräsentiert die Dauer der Note. Die Tonhöhe lässt sich an der y-Achse ablesen.

Eine Anfrage wird als „P“ bezeichnet und setzt sich aus folgenden m Segmenten zusammen: $[p_1, p'_1], \dots, [p_m, p'_m]$. Ein Stück aus der Datenbank wird definiert als T und setzt sich wiederum aus n Segmenten zusammen: $[t_1, t'_1], \dots, [t_n, t'_n]$. Des weiteren geht man davon aus, dass $m \ll n$ sein soll. Die Segmente der Anfrage und der Datenbank sollen in lexikografischer Reihenfolge vorliegen.

Bei „exact matching“ = P1 werden zuerst alle Startpunkte der Anfrage $\overline{P} = (p_1, p_2, \dots, p_m)$ mit den Startpunkten der Datenbank $\overline{T} = (t_1, t_2, \dots, t_n)$ verglichen. Da die gesuchte Anfrage nicht unbedingt am Anfang des Musikstückes zu finden sein muss, wird zum Beginn der Suche der Punkt p_1 in allen Dokumenten ermittelt. Für die Dokumente in denen der Punkt p_1 auftaucht, wird nun jeweils die zeitliche Distanz vom Start des Stückes bis zu diesem Punkt als Wert f gespeichert. Dieser f-Wert wird zu jedem p_i hinzuaddiert. In den bisher gefundenen Stücken wird jetzt nach den Punkten p_2+f bis p_m+f gesucht. Analog wird mit den Endpunkten p'_1+f bis p'_m+f verfahren. Sollten alle Punkte exakt zutreffen, so ist $\overline{P}+f \subseteq \overline{T}$ und man hat eine gewisse Anzahl von zutreffenden Musikstücken gefunden.

Nähere Informationen und genaue mathematische Berechnungen zum Algorithmus P1 befinden sich auf der Internetseite von C-Brahms, algorithm references [1-3] in Abschnitt: „3 Exact matching“. [6]

3.1.4 Zusammenfassung

Das Programm C-Brahms ist vor allem eine Musiksuchmaschine für Nutzer, die sich gut mit Musik auskennen. Die 9 verschiedenen Suchmethoden haben alle ihre Vor- und Nachteile. Ein Anwender, der sich mit ihnen auseinandersetzt, kann für jede Anfrage die effektivste Methode auswählen. Je nach Methode werden unterschiedliche Algorithmen für die Berechnung der Ergebnisse benutzt.

Hat der Nutzer das gesuchte Musikstück gefunden, so kann er sich eine längere Midi-Datei anhören und sich die Notenschrift für die Melodie ansehen.

Das Programm C-Brahms findet man auf der Internetseite:

<http://www.cs.helsinki.fi/group/cbrahms/demoengine/>

3.2 Themefinder

Themefinder entstand als nicht-kommerzielles Gemeinschaftsprojekt der Universitäten Ohio State und Stanford. Die Datenbank und Suchmaschine wurden von Prof. David Huron entwickelt, bei Datenbankorganisation und -management wurde er zusätzlich von Doktor Craig Stuart Sapp unterstützt. [7]

3.2.1 Aufbau

Das Programm benutzt zur Anfragfragestellung ein web-basiertes Interface. Man hat die Möglichkeit die Datenbank nach so genannten Themes oder Incipits zu durchsuchen. Das Incipit umfasst die ersten Takte eines Musikstückes. Im Gegensatz dazu muss das Theme oder Thema eines Musikstücks nicht zwangsläufig am Anfang auftauchen.

Die bei Themefinder verfügbaren Stücke sind im „**kern“ Musikformat codiert, welche auf dem Humdrum-Code basiert. (siehe Kapitel 3.2.2)

Das Web-Interface greift auf CGI-Perl-Skripte zu, welche dem Benutzer basierend auf seiner Anfrage eine Seite mit dem Ergebnis zusammenstellt. [7] Welche Algorithmen Themefinder zur Suche verwenden ist nicht bekannt. Letztendlich arbeitet die Suche wahrscheinlich mittels eines Textvergleiches zwischen den gespeicherten Humdrum-Stücken, und der in den Humdrum-Code übersetzten Anfrage.

Themefinder bietet mehrere unterschiedliche Möglichkeiten die Suchanfrage zu formulieren. [8]

- | | |
|---------------------|---|
| 1. Repertoire: | hier kann man eine Anfrage nach klassischer Musik, Folklore, etc. einteilen |
| 2. Pitch: | direkte Tonhöhenangaben (z.B.: C, E, F#) |
| 3. Interval: | Suche nach musikalischen Intervallen (z.B.: +m9, -M3) |
| 4. Scale Degree: | Tonleitersuche (z.B.: do=1, re=2, mi=3, ...) |
| 5. Gross Contour: | auf und ab der Melodie (z.B.: / = hoch, \ =runter, - = gleich) |
| 6. Refined Contour: | verfeinerte Gross Contour (z.B.: u = kleiner Schritt nach oben, U = großer Schritt nach oben, ...) |
| 7. Location: | hier kann man angeben, ob sich die Anfrage am Beginn eines Themas befindet oder an jeder andern beliebigen Stelle |
| 8. Key: | Notenschlüssel (z.B.: C) |
| 8.1 Mode: | minor oder major (Moll oder Dur) |
| 9. Meter: | Taktangabe (z.B.: 1/4, 4/4) |

Hat man sich für eine oder mehrere Möglichkeiten der Anfrage entschieden, wird die Suche gestartet. Als Resultat wird eine Seite generiert, welche die passenden Musikstücke in alphabetischer Reihenfolge auflistet.


Die Resultatliste ist eine Reihe von Incipits verschiedenster Musikstücke. Ist die gesuchte Melodie in der Liste, besteht die Möglichkeit sich weitere Informationen anzeigen zu lassen. Man kann sich diese Incipits anhören, genauere Informationen zum Musikstück abrufen und sich den Humdrum-Code ansehen.






Beispiel:

Stellt man bei Themefinder die Anfrage nach der Notensequenz „A G E“ so bekommt man eine Trefferliste mit 10 Musikstücken, die dieses Thema enthalten.


Matches = 10

Brahms, Johannes (1833-1897), Concerto in A Minor, Op. 102, Vn., Vcl. & Orch., 1st Movement, Theme



1.     

Brahms, Johannes (1833-1897), Quintet in F Minor, Op. 34, Pft. & Str., 3rd Movement, 3rd Theme













2.     

Abbildung 3.2: Auszug Trefferliste Themefinder

Angenommen die gesuchte Melodie entspricht dem ersten Stück in der Liste, also Johannes Brahms Konzert in A Moll, dann werden dem Nutzer folgende Optionen angezeigt:

-  genaue Informationen über das Musikstück (z.B.: Erscheinungsjahr, Satznummer)
-  anhören der Mididatei
-  zusätzliche Seite, mit Informationen zu Humdrum-Code
-  andere Themes im selben Satz
-  andere Themes in der selben Arbeit

3.2.2 Humdrum-Code

Bei Themefinder liegen die Musikstücke in der „**kern“-Notation vor. Diese erlaubt nicht nur die Codierung von Tonhöhe und Dauer, sondern auch spezielle Eigenheiten eines Musikstückes, wie z.B. Ornamentik oder Notenvorzeichen. Man kann also auch ein Musikstück nach solchen Kriterien analysieren.

Neben „**kern“ existieren noch viele andere Notationen, welche auf dem Humdrum-Code basieren.

Der Humdrum-Code selbst repräsentiert die Syntax, bzw. Grammatik der „**kern“-Notation.

Zur besseren Erläuterung wird nachfolgend der Humdrum-Code für Johannes Brahms Konzert in A Moll betrachtet.

```

1.      !!!ref: COM CDT, OTL, OPS, OMV, thm
2.      !!!COM: Brahms, Johannes
3.      !!!CDT: 1833/05/07/-1897/04/03/
4.      !!!CNT: German
5.      !!!OTL: Concerto in A Minor for Violin, Cello, and Orhcestra
6.      !!!OPS: Op. 102
7.      !!!ODT: 1888///
8.      !!!OCY: &Ouml;sterreich
9.      !!!OPC: Vienna
10.     !!!PDT: 1879///
11.     !!!MRD: 1887///
12.     !!!OMV: Mov. 1
13.     !!!thm: A
14.     !!!AIN: cello orchestra violn
15.     !!!ENC: David Huron & Walter Hewlett
16.     !!!EEV: Version 1.1
17.     !!!OCL: RSC
18.     !!!ACO: Classical; RSC
19.     !!!YEM: Copyright 1999 Center for Computer Assisted Research in the
           Humanities & David Huron
20.     !!!YEN: United States of America
21.     **kern
22.     *clefG2
23.     *k[]
24.     *a:
25.     *M4/4
26.     =
27.     4.aa\
28.     8gg\
29.     4ee\
30.     4r
31.     =
32.     4.ffa\
33.     8ee\
34.     4a/
35.     4r
36.     =
37.     6b\
38.     6cc\
39.     6dd\
40.     4.cc\
41.     8a/
42.     =
43.     4.g/
44.     8f/
45.     4.e/
46.     8cc\
47.     =
48.     2b\
49.     2a/
50.     =
51.     2ee\
52.     *-

```

Abbildung 3.3: Humdrum-Code

Der Humdrum-Code ist in der Regel in drei Teile aufgeteilt. Die ersten 20 Zeilen, welche durch „!!!“ gekennzeichnet werden, sind reine Kommentare. Hier können Angaben zum Komponisten, Erscheinungsdatum und vielem mehr gemacht werden.

Der zweite Teil besteht im Beispiel aus den Zeilen 21 bis 25, hier werden generelle Angaben zum Musikstück gemacht. Die erste Angabe „**kern“ gibt an, dass es sich hierbei um die „**kern“-Interpretation des Humdrum-Codes handelt. Die Humdrum-Informationen werden als Token bezeichnet. In der „**kern“-Interpretation dürfen diese Token nur drei Datentypen entsprechen, welche Noten (notes), Pausen (rest) und Taktstrich (bar line) umfassen.

Die nächsten 4 Zeilen sind so genannte Tandem-Interpretationen. Sie werden durch einen einfachen * gekennzeichnet. Tandem-Interpretationen sind ergänzende oder zusätzliche Informationen zum Musikstück, *clefG2 gibt z.B. an, dass dieses Thema einen G-Schlüssel hat, k[] bedeutet, dass keine Note in der Melodie mit einem Vorzeichen behaftet ist und M4/4 gibt den Takt an.

Die Zeilen 26 bis 52 machen den letzten Teil des Codes aus. Durch das = werden Taktstriche, die in der Notenschrift beschrieben sind, wiedergegeben. Die Melodie beginnt mit einer punktierten Viertelnote mit der Tonhöhe „A5“. Alle diese Angaben lassen sich mittels Humdrum-Code darstellen:

4 . aa \

Die 4 gibt an, dass es sich um eine Viertelnote handelt, wohingegen der nachfolgende Punkt diese Note als punktiert repräsentiert. Mit dem „aa“ wird die Tonhöhe beschrieben. Hierbei handelt es sich um ein A, welches eine Oktave höher ist als das mittlere A. (siehe Kapitel 2.2) Im Humdrum-Code wird mit dem letzten Zeichen beschrieben, ob der Notenhals nach rechts oben oder links unten geführt wird.

Abgeschlossen wird die Humdrum-Codierung einer Melodie immer mit dem Zeichen: *-.

Mehrstimmigkeit wird im Humdrum-Code ganz einfach durch zusätzliche Spalten dargestellt. Leider nutzt die Themefinder-Suchmaschine diese Möglichkeit der Darstellung derzeit noch nicht aus. Ein Beispiel wie es repräsentiert sein könnte, wäre:

```

**kern      **kern      **kern
*I:Violino I *I:Violino II *I:Bassi
*clefG2     *clefG2     *clefF4
*k[f#c#g#]  *k[f#c#g#]  *k[f#c#g#]
...         ...         ...
[4cc\      [4cc\      8f/
4ee\      4ee\      8g/
...         ...         ...

```

Mit Hilfe von einigen weiteren Zeichen lassen sich alle symbolischen Darstellungen mittels Humdrum-Code darstellen. So werden z.B.: Haltebögen durch [] repräsentiert. Genauere Informationen zur Darstellung der einzelnen symbolischen Zeichen im Humdrum-Code sind unter: <http://www.music-cog.ohio-state.edu/Humdrum/representations/kern.html> [9] zu finden.

3.2.3 Zusammenfassung

Das Programm Themefinder bietet nicht nur Musikprofis eine Möglichkeit nach Melodien zu suchen. Sollte dem Nutzer die genaue Tonhöhe einmal nicht bekannt sein, so bietet das Programm noch eine Vielzahl von anderen Arten der Suchanfrage an.

Die Codierung der Musikstücke erfolgt über den Humdrum-Code, mit welchem nicht nur die Noten einer Melodie codiert werden können. Es werden alle symbolischen Zeichen, ob sie nun für das Spielen des Themas relevant sind oder nicht, in den Code aufgenommen. Das Programm Themefinder liefert dem Anwender zusätzliche Informationen zu den gefundenen Melodien. So kann man sich die Melodie anhören und andere Themen desselben Satzes ansehen, wenn diese in der Datenbank vorhanden sind.

Das Programm Themefinder und weitere Informationen finden Sie auf der Internetseite:

<http://www.themefinder.org>

3.3 Musipedia

Musipedia ist eine offene Musik-Enzyklopädie. Sie wurde 1997 von Rainer Typke entwickelt. Ursprünglich orientierte sich das Programm am Buch von Denys Parsons „The Directory of the Tunes and Musical Themes“.

3.3.1 Aufbau

Seit dem Jahr 1997 wurde das Programm ständig weiterentwickelt und verwendet heute für den Suchalgorithmus eine angepasste Form der Vantage-Point-Methode. Die Vantage-Point-Methode wurde von ihren Entwicklern Vleugels und Veltkamp. Für die effiziente Suche nach Bildern beschrieben. Um diese Methode für ein effizienteres Suchen bei Musipedia einzusetzen zu können, musste sie noch angepasst werden. Musipedia benutzt für die Suche die Parsons-Code-Editierdistanzen (siehe Kapitel 3.3.2), welche die gemeinsamen Anfänge zweier Zeichenketten betrachtet. Die Vantage-Point-Methode verwendet dagegen für das Suchen von Bildern die Dreiecksungleichung, welche in der Regel für die Editierdistanzen nicht gilt. [10]

Die Enzyklopädie kann heute von jedem erweitert und editiert werden. Dabei ist die Form des Eintrages nicht von Bedeutung. Es kann sich um ein Notenbild, eine Midi-Datei, einen Text über einen Komponisten oder Parsons-Code handeln.

Man hat mehrere Möglichkeiten um nach einer Melodie zu suchen. Zum Einen kann man den Parsons-Code eingeben (siehe Kapitel 3.3.2), die Klaviatur verwenden um eine Melodie einzugeben, oder man singt und pfeift dem Computer die gesuchte Melodie einfach vor (Search by Humming).

Bei der Klaviatur benutzt man die Tasten einer Klavierdarstellung, bei der auch die Möglichkeit besteht die Tonhöhen und den Rhythmus mit anzugeben. Allerdings berücksichtigt die Suchmaschine derzeit nur den Parsons-Code. Angaben über die Tonhöhe und die Dauer werden beim Parsons-Code nicht aufgeführt. (siehe Kapitel 3.3.2) Deshalb müssen die Angaben zu Tonhöhe und Rhythmus über die Klaviatur nicht korrekt sein, solange die Melodiekontur einigermaßen stimmt.

Nach Eingabe der gesuchten Melodie wandelt das Programm die Noten in den Parsons-Code um. Auf Basis des Codes und der Parsons-Code-Editierdistanzen wird in der Datenbank nach dem passenden Musikstück gesucht. Um ein gutes Ergebnis zu erreichen, sollte man laut einer Empfehlung von Musipedia zwischen 10 und 20 Noten eingeben. [11]

Die passenden Musikstücke werden, sortiert nach der Genauigkeit der Übereinstimmungen, in einer Liste angegeben. Man hat nun die Möglichkeit die Suche einzugrenzen, sich die gefundenen Musikstücke anzuhören, oder sich genauere Informationen zu den einzelnen Musikstücken anzeigen zu lassen.

Sollten wenige passende Musikstücke gefunden werden, berücksichtigt das Programm mögliche Fehler bei der Eingabe und gibt alternative Vorschläge an.

3.3.2 Parsons-Code

Der Code wurde von Denys Parsons entwickelt und ist besonders gut für die Suche nach Melodien geeignet. Auch ohne viel Musikverständnis kann man anhand des Parsons-Code nach einer Melodie suchen. Hierbei wird die Melodie eines Musikstückes durch eine einfache Buchstabenkette dargestellt. Mit dem Parsons-Code wird das auf und ab der Melodie wiedergegeben. Der Code betrachtet weder die Tonhöhe, noch den Rhythmus eines Musikstückes, aber auch ohne diese scheinbar wichtigen Informationen ist es möglich viele verschiedene Melodien zu unterscheiden.

Der Code basiert auf vier Zeichen, welche durch Aneinanderreihung die Melodie repräsentieren.

- U (up) -> die nachfolgende Note ist höher
- D (down) -> die nachfolgende Note ist tiefer
- R (repeat) -> sie ist gleich

- * -> die erste Note dient als Bezugspunkt. Sie wird als Stern dargestellt.

Beispiel:



Abbildung 3.4: Notenbeispiel Musipedia

Parsons-Code: *UDDR

visualisierter Parsons-Code:

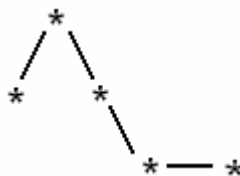


Abbildung 3.5: visualisierter Parsons-Code

Von allen Musikstücken in der Datenbank Musipedia's wurde der Parsons-Code abgespeichert. Bei der Suche nach Melodien wird der Parsons-Code dieser Musikstücke mit dem Parsons-Code der gesuchten Melodie verglichen und unter Zuhilfenahme der Parsons-Code-Editierdistanz eine Übereinstimmung berechnet.

3.3.3 Parsons-Code-Editierdistanz

Mit der Editierdistanz lässt sich die Ähnlichkeit zwischen zwei Zeichenketten bestimmen. Zwei Zeichenketten sind sich besonders ähnlich, wenn nur wenige Editieroperationen nötig sind, um die eine Zeichenkette in die andere umzuwandeln. Dafür gibt es drei verschiedene

Editieroperationen. Diese sind: das Löschen von Zeichen, das Einfügen von Zeichen und das Ersetzen von einem Zeichen x durch ein Zeichen y .

Um zu vermeiden, dass zwischen jeder Anfrage und den Melodien in der Datenbank die Editierdistanz berechnet werden muss, setzt Musipedia die Vantage-Point-Methode ein. Für alle Melodien, die sich in der Datenbank befinden, wird vorab der Abstand zum Vantage-Objekt berechnet. Die Vantage-Point-Methode nutzt für ihre Berechnungen die Dreiecksungleichung aus.

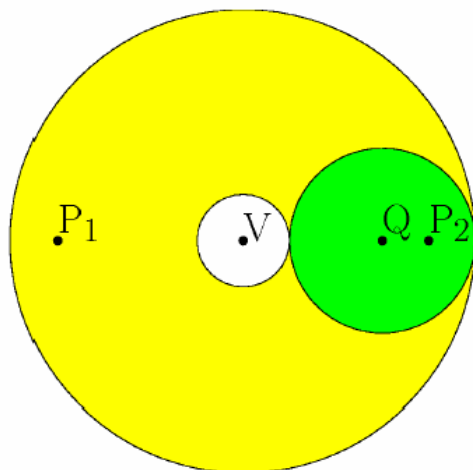


Abbildung 3.6: Vantage-Objekt

Vantage-Objekts V :

Gesucht werden alle Objekte P , die sich innerhalb des grünen Kreises mit dem Radius r befinden. Den Mittelpunkt des grünen Kreises bestimmt die Anfrage Q . Für alle in der Datenbank vorkommenden Objekte P_i wurde der Abstand zum Vantage-Objekt bereits vorab berechnet. Wenn jetzt eine Anfrage gestellt wird, muss nur noch der Abstand der Anfrage Q zu V berechnet werden. Jetzt kann man alle Objekte außerhalb des gelben Ringes um V schnell ausschließen. [10]

Die Melodien mit ähnlichem Abstand zum Vantage-Objekt sind der Suchanfrage ähnlich und werden als Treffer ausgegeben. Der Einsatz der Vantage-Point-Methode steigert somit die Effektivität des Programms.

Um alle Objekte außerhalb der Schnittmenge entsprechender Ringe auszuschließen, kann man mehrere Vantage-Objekte angeben. [10]

3.3.4 Zusammenfassung

Musipedia ist eine Musiksuchmaschine, die vor allem Musikamateuren die Suche nach einer Melodie ermöglicht. Der Nutzer benötigt keine Kenntnisse über Noten oder Tonhöhen. Die Suchmaschine verwendet für die Suche den einfachen Parsons-Code und vergleicht dann mittels Editierdistanz die Zeichenkette der Anfrage mit den Zeichenketten der in der Datenbank vorhandenen Musikstücke.

Der Anwender hat die Möglichkeit sich die gefundene Melodie anzuhören und sich nähere Information über das Musikstück, zu welchem die Melodie gehört, anzusehen.

Das Programm, sowie nähere Informationen lassen sich auf der Internetseite:

<http://www.musipedia.org>
finden.

4 Vergleich der Konzepte

In den vorangegangenen Kapiteln wurden die drei Systeme C-Brahms, Themefinder und Musipedia kurz vorgestellt. Vergleicht man die drei Konzepte miteinander, so stellt man einige Unterschiede fest. Des Weiteren soll in diesem Kapitel ermittelt werden, wie genau die vorgestellten Systeme die Anforderungen für das eNoteHistory-Projekt erfüllen.

Verglichen wird in den Bereichen Anfrage, Anfragefehler, Ranking, Suchmethoden und Speicherung.

Anforderungen des eNoteHistory-Projektes:

Zur symbolischen Repräsentation werden n-grams verwendet. Die Länge n soll generell variabel gehalten werden und im Bereich zwischen 5 und 9 liegen. Die n-grams sollen sowohl den Tonhöhenverlauf, als auch die Tondauer repräsentieren. Dabei kann die Tonhöhe auf der absoluten als auch auf der relativen Tonhöhe abgebildet werden.

Anfrage:

Die Unterschiede der Programme beginnen schon bei der Anfrage. Bei allen drei Systemen kann der Nutzer die Anfrage über Tonhöheangaben stellen. Die Systeme C-Brahms und Musipedia bieten die Möglichkeit der Eingabe der Töne über die Klaviertastendarstellung, Themefinder dagegen nur über die Computertastatur. Die einfachste Anfrage bietet Musipedia dem Anwender. Da im Moment die Melodien nur in der Parsons-Code-Notation in der Datenbank vorliegen, benötigt der Anwender weder genaue Kenntnisse über Tonhöhen, noch Rhythmus. Ihm werden 3 Methoden zur Anfragestellung zur Verfügung gestellt. Auch bei der Angabe der Tonhöhen über die Klaviertastendstellung, wird nur die melodische Kontur zur Berechnung des Ergebnisses betrachtet. Der Nachteil dabei ist, dass der Anwender eine ausreichend große Anzahl von Noten beschreiben muss, um ein gutes Ergebnis zu erzielen. Das System bietet vor allem Anwendern mit wenigen Kenntnissen über Musiknotation, eine Möglichkeit nach Melodien zu suchen. Ganz anders ist das Konzept von C-Brahms. Bevor eine Anfrage starten kann, sollte man sich mit den verschiedenen Suchmethoden beschäftigen. Denn nur wenn der Anwender die passende Suchmethode für seine Anfrage gewählt hat, ist die Anfrage auch erfolgreich. Ein Vorteil gegenüber den anderen Programmen ist aber, dass C-Brahms unter anderem Suchmethoden verwendet, die die Möglichkeit bieten auch nach polyphonen Noten zu suchen. Der Nutzer kann mit C-Brahms zusätzlich zur Tonhöhe auch auf den Rhythmus eingehen.

Eine gute Alternative zu den beiden anderen Programmen ist Themefinder. Es bietet dem Nutzer mehr Methoden der Anfragestellung, aber anders als bei C-Brahms muss er nicht eine Entscheidung über verschiedene Suchmethoden treffen. Themefinder ist sowohl für Anwender mit Musikkenntnissen als auch für Anwender mit wenig Kenntnissen nutzbar. Der Anwender kann die Art der Anfragestellung wählen, die ihm zusagt.

Fazit: Das System Themefinder erfüllt die Anforderungen des eNoteHistory-Projektes. Es bietet dem Anwender die Anfragestellung nach der relativen und der absoluten Tonhöhe an. Musipedia bietet zwar auch beide Arten der Anfrage an, aber gesucht wird nur mittels relativer Tonhöhe.

Anfragefehler:

Alle drei Systeme berücksichtigen mögliche Fehler in der Anfrage. Die vorgestellten Systeme arbeiten auch in diesen Bereich unterschiedlich. C-Brahms bietet drei Suchmethoden an, die Fehler in der Anfrage berücksichtigen. Da Melodien in Musipedia nur im Parsons-Code in der Datenbank vorhanden sind, kann der Nutzer bei der Angabe der Tonhöhe Fehler machen. Sollten nicht genug Ergebnisse für die gestellte Anfrage zu finden sein, gibt Musipedia auch Melodien an, die mit ihrem Parsons-Code leicht von der Anfrage abweichen. Themefinder geht auch hier einen anderen Weg. Hier kann der Nutzer schon bei der Anfrage mögliche Fehler eingeben bzw. kennzeichnen. Sollte er sich an einer Stelle nicht sicher sein ob der angegebene Ton der Richtige ist, kennzeichnet er die Stelle durch einen „.“, ein „?“ oder einen „*“.

Fazit: Auch das System von eNoteHistory soll Fehler, die dem Anwender bei der Anfragestellung möglicherweise unterlaufen, berücksichtigen. Das Konzept von Musipedia erfüllt am Besten die Anforderungen des Projektes. Es werden die n-grams ausgegeben, die dem Anfrage-n-gram am Ähnlichsten sind.

Ranking:

Bei den Systemen C-Brahms und Musipedia wird die Trefferliste nach der Genauigkeit sortiert, d.h. wie genau stimmen Anfrage und das gefundene Incipit überein. Bei C-Brahms werden die Incipits zuerst angegeben die keine Abweichung zur Anfrage aufweisen, sollten es mehrere sein, werden sie nach Alphabet sortiert. Anschließend werden die Incipits aufgelistet, die leicht von der Anfrage abweichen, z.B. transponierte Incipits. Das System Musipedia sortiert nach der Größe der Editierdistanz. Das Incipit, dessen Editierdistanz am Wenigsten zur Editierdistanz (immer im Bezug auf das Vantage-Objekt) der Anfrage abweicht, steht an erster Stelle. Themefinder sortiert seine Ergebnisliste nach dem Alphabet.

Alle drei Systeme bieten Informationen über den Komponisten und das Musikstück an. Außerdem kann man sich bei allen dreien die Melodien anhören. C-Brahms hat in diesem Punkt einen Vorteil. Anders als Themefinder und Musipedia, deren Midi-Datei aus 20-30 Noten besteht, kann der Nutzer bei C-Brahms sich einen größeren Ausschnitt aus dem Musikstück anhören. So bekommt er einen besseren Eindruck vom ganzen Musikstück.

Positiv bei Themefinder ist, dass der Anwender sich andere Themen aus den Musikstücken, die in der Datenbank gespeichert sind, anzeigen lassen kann.

Das geringste Angebot an Zusatzinformationen bietet Musipedia. Man kann sich das Thema anhören, bekommt kurze Informationen zum Musikstück und sieht den Parsons-Code für die Melodie.

Suchalgorithmen und Speicherung:

Wie schon in Kapitel 3 erwähnt, verwenden die drei Systeme unterschiedliche Arten der Speicherung. Musipedia greift auf die Speicherung mit Hilfe des Parsons-Codes zurück und verzichtet damit bewusst auf die Betrachtung von Tonhöhe und Rhythmus. Der Code ist sehr einfach und mit Hilfe der Editierdistanzen lässt sich berechnen, ob zwei Zeichenketten gleich sind.

Bei Themefinder erfolgt die symbolische Repräsentation mittels Humdrum-Code. Mit seiner Hilfe lassen sich nicht nur Tonhöhenangaben codieren, sondern auch alle anderen musikalischen Merkmale.

Auch hier bildet C-Brahms wieder eine Ausnahme. Das Programm hat für die unterschiedlichen Suchmethoden auch unterschiedliche Algorithmen. Diese werden unterschiedlich gespeichert, was im Kapitel 3.1 nachzulesen ist.

Fazit: Wie im Kapitel 5 vorgestellt, verwendet das Konzept von eNoteHistory zur Berechnung der Ähnlichkeit zwischen Anfrage und den Incipits in der Datenbank die Editierdistanz. Aber anders als bei Musipedia wird die Editierdistanz nicht auf dem Parsons-Code angewendet. Die n-grams im Konzept von eNoteHistory enthalten nicht nur die relative Tonhöhe. Nähere Informationen in Kapitel 5.

Vergleichsbeispiel:

Stellt man an alle drei Programme die gleiche Anfrage, so liefern alle die passenden Ergebnisse. Im Test wurde nach Ludwig van Beethovens Stück „Für Elise“ gesucht. Die elf Töne zu Beginn des Stückes lauten: „E´ D´# E´ D´# E´ B´ D´ C´ A C E A B E“.

Bei der Anfrage fallen einem die ersten Unterschiede auf. Die Töne oben sind so aufgeschrieben, wie die Anfrage mittels Klaviatur unter Musipedia aussieht. Unter C-Brahms ist die Notation anders, hier ist „E´= E6“, „D´# = D6#“ und so weiter. Bei Themefinder lässt man die Angaben zur Oktave ganz weg und schreibt nur: „E G# ...“.

Zuerst wird getestet, wie das Ergebnis mit der von den Programmen angegebenen Mindestanfragegröße aussieht. (siehe Tabelle 4.1)

Die Tabelle zeigt, dass die angegebene Mindestgröße der Anfrage keine guten Ergebnisse liefert. Das beste Ergebnis liefert C-Brahms, aber auch nur wenn man die richtige Suchmethode wählt. (im Beispiel die MonoPoly-Methode)

	min. Größe	Übereinstimmungen	Für Elise	Fazit
C-Brahms	2	18135 Treffer in 277 Titeln	12. Stelle	mit richtigen Algorithmus, in Ordnung
Themefinder	1	1051 Treffer	65. Stelle	da nach Alphabet geordnet schon an 65 Stelle
Musipedia	5	100 angegebene Treffer	nicht gefunden	nicht unter den max. 100 angegebenen Treffern

Tabelle 4.1: Auswertung Mindestanfragegröße

Da im eNoteHistory-Projekt die n-grams eine Länge von 5 bis 9 Tönen haben werden, sollte auch die Anfrage in diesem Bereich liegen. Der zweite Test soll zeigen, wie genau die vorgestellten Systeme mit einer Anfragelänge von 7 das gesuchte Incipit (Für Elise) finden.

	Anfragegröße	Übereinstimmungen	Für Elise
C-Brahms	7	1. Treffer	1. Stelle
Themefinder	7	1. Treffer	1. Stelle
Musipedia	7	100 angegebene Treffer	41. Stelle

Tabelle 4.2: Auswertung Anfragegröße 7

Der Test zeigt, dass die Systeme C-Brahms und Themefinder zumindest bei der Suche nach dem Incipit des Stückes „Für Elise“ ein sehr gutes Resultat liefern. Musipedia benötigt weit mehr Angaben, als die beiden anderen Konzepte. Erst bei einer Anfragelänge von 14 steht „Für Elise“ an erster Stelle.

Gesamtfazit:

Keines der vorgestellten Konzepte kann man eins zu eins für das eNoteHistory-Projekt übernehmen. In jedem der drei Programme gibt es jedoch gute Ansätze, die man in das eNoteHistory-Projekt mit einbeziehen kann.

Musipedia hat zwar die leichteste Anfrage, aber die Suchmaschine ist nicht wirklich dazu geeignet professionell nach Musikstücken zu suchen. Das Programm kann dem Anwender helfen nach einer Melodie zu suchen, die ihm durch den Kopf geht und von der er keine genaueren Angaben hat. Aber dies ist nicht effektiv bei der Suche nach einer Melodie, die in Notenschrift vorliegt. Dazu eignet sich besser das Programm C-Brahms, da der Nutzer hier genauere Angaben zur Melodie machen kann. Allerdings ist die Auswahl der passenden Suchalgorithmen sehr schwer und der Anwender sollte sowohl Kenntnisse in der Musiktheorie, als auch in der Mathematik besitzen, um dieses Programm effektiv nutzen zu können.

Eine gute Mischung ist das Programm Themefinder. Die Anfrage ist nicht so kompliziert wie bei C-Brahms und die Speicherung ist genauer als bei Musipedia. Durch die Humdrum-Codierung ist es möglich, alle relevanten Eigenschaften eines Musikstückes zu codieren. Außerdem benötigt man dadurch weniger Anfrageelemente, um die Ergebnisliste einzuschränken.

Auf der nächsten Seite findet sich noch eine tabellarische Zusammenfassung der Unterschiede der drei vorgestellten Programme. (siehe Tabelle 4.3)

	C-Brahms	Themefinder	Musipedia
Anzahl der Melodien		über 18.000 Incipits	ca. 30.405 (am 15.11.05)
Anfragemethode	1 Klaviertastatur	9 verschiedene	3 verschiedene
Suchmethoden	9 verschiedene	1	1

Größe der Anfrage	min. 2 / max. 30	min.1 / für gute Ergebnisse mehr	min.5 / besser 10-20
Anfragefehler	einige Suchmethoden	Eingabe während der Anfrage	wenn zu wenig Treffer, Alternativen
Speicherung	verschiedene je nach Suchmethode	Humdrum-Code	Parsons-Code
Ergebnis	viele Informationen	viele Informationen	wenig Information
Besonderheiten	lange Midi-Datei zum Anhören	andere Themen der gleichen Arbeit	pfeifen und summen der Anfrage

Tabelle 4.3: Konzept-Vergleich

5 Konzept für eNoteHistory

Die nachfolgende Zeichnung ist eine grobe Darstellung der Musiksuchmaschine von eNoteHistory. Am Ende des Kapitels befindet sich eine detaillierte Darstellung der Suchmaschine.

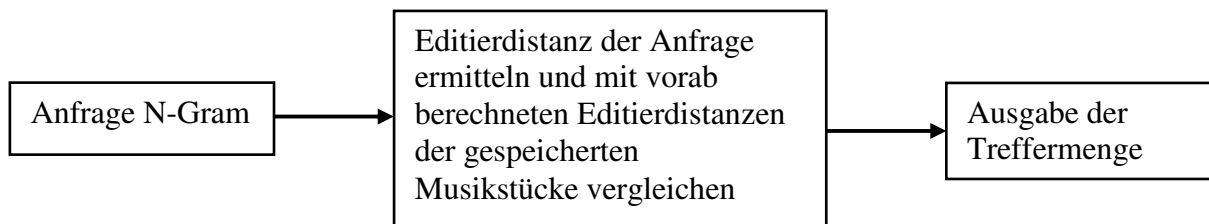


Abbildung 5.1: erste Konzept-Grafik

Im Projekt eNoteHistory erfolgt die Speicherung der Incipits der einzelnen Musikstücke durch n-grams. Ein n-gram hat die Länge n, welche im Projekt generell variabel sein, aber zwischen 5 und 9 liegen soll. Da ein Incipit länger als 9 Noten lang sein kann, wird das Incipit in Segmente aufgeteilt und zu jedem Segment ein n-gram abgespeichert. Das zu einem Incipit gehörende n-gram gibt also den Tonhöhenverlauf wieder. Da es sich bei den gespeicherten Stücken meist um polyphone Musikstücke handelt, wird zu jeder Stimme ein eigenes n-Gram abgespeichert. Es gibt verschiedene Arten der Tonhöhenabbildung. Zum einen die Abbildung der absoluten Tonhöhe und zum andern die Abbildung der relativen Tonhöhe.

Befassen wir uns hier zuerst mit der relativen Tonhöhenabbildung. Die Idee die dahinter steckt ist die Gross Contour, auch bekannt als Parson's Code. Diese Art der Abbildung wird auch bei dem Programm Musipedia angewendet. Die Idee ist gut, aber sie hat den entscheidenden Nachteil, dass die Differenzierung der einzelnen Stücke eine große Anzahl von Tonhöhenzeichen erfordert. Da nur das auf und ab der Melodie gespeichert wird, sind sich viele Melodien zumindest am Anfang sehr ähnlich. Um dieser Gleichheit ein wenig entgegenzuwirken, kann man das auf und ab einer Melodie noch etwas genauer beschreiben. Bei Themefinder heißt die Idee Refined Contour, hierbei wird auch die Größe der Auf- und Absprünge der Melodie beschrieben.

Notenbeispiel	Gross Contour	Refined Contour
	UDDR	UddR
	UDDR	uddR

Tabelle 5.1: Vergleich relative Tonhöhenangabe

Wie man in der Tabelle sieht, unterscheiden sich die beiden Notenbeispiele in der Gross Contour nicht voneinander. Durch die Angabe der Abbildung mittels Refined Contour ist die Wiedergabe genauer.

Die relative Abbildung der Tonhöhe ist leicht zu realisieren und auch der Vergleich der Anfrage mit den Musikstücken, die in der Datenbank vorhanden sind, lässt sich mittels Editierdistanz gut bewerkstelligen.

Da die Anfrage bei eNoteHistory aber auch nur 5-9 Noten lang sein soll, macht sich der Nachteil der relativen Tonhöhenabbildung zu stark bemerkbar.

Es ist besser die Tonhöhenabbildung mit Hilfe der absoluten Tonhöhe anzugeben. Hierbei stellt sich allerdings die Frage, ob wir die Oktavierung der Noten mitbeachten oder nicht.



Notenbeispiel	ohne Oktavierung	mit Oktavierung
	c f g c a	c1 f1 g1 c1 a1
	c f g c a	c1 f1 g1 c2 a1

Tabelle 5.2: Vergleich absolute Tonhöhenangabe

Genauer wird die Speicherung der Incipits natürlich mit der Angabe, in welchem Oktavenbereich sich die einzelne Note befindet. In welcher Oktave die Note gespielt wird, soll hier durch die Groß- und Kleinschreibung und die Angabe einer Zahl nach dem Notennamen angegeben werden.

Ein anderer wichtiger Bestandteil der Musik ist die Angabe des Notenwertes, also die Dauer der Note. Auch diese Angabe lässt sich in einem n-gram wiedergeben. Man schreibt vor die Tonhöhe die Angabe für den Notenwert.

Zeichen	Name
1c	ganze Note (c in der kleinen Oktave)
2c	halbe Note
4c	Viertelnote
8c	Achtelnote
16c	Sechzehntelnote
32c	Zweiunddreißigstelnote
64c	Vierundsechzigstelnote

Tabelle 5.3: Auflistung Notenwertangaben

Ein Musikstück besteht nicht nur aus Tonhöhenangaben. Es gibt noch eine Vielzahl von anderen Zeichen, wie z.B.: Pausen, punktierte Noten, Haltebögen u.v.m. Diese sollen in diesem Konzept bei der Speicherung der n-grams erst einmal nicht berücksichtigt werden.

Laut Anforderung an das eNoteHistory-Projekt sollen die Incipits sowohl durch n-grams mit relativer, als auch n-grams mit absoluter Tonhöhe repräsentiert werden. Dies ermöglicht eine relativ genaue Abbildung der Notenschrift einer Melodie mittels n-grams.

Notenbeispiel	N-gram für das Beispiel
	4a2 8g2 4e2 4f2 8e2 4a1
	U U D U U

Tabelle 5.4: Beispiel n-gram

Die in der Datenbank gespeicherten n-grams werden mit der Anfrage verglichen. Auch die Anfrage hat die Form von n-grams.

Wie bei der Speicherung der Incipits in der Datenbank, enthalten die n-grams auch bei der Anfrage Informationen über Notenwert und Tonhöhe.

Damit die Treffermenge möglichst klein und genau ist, sollte die Anfrage eine Länge von 5 bis 9 Noten haben. Das Ergebnis wird natürlich exakter, wenn die Länge der Anfrage genau der Länge des n-grams entspricht, welches es zu suchen gilt.

Auch die Suchanfrage mit nur einer Note sollte rein theoretisch möglich sein, wobei allerdings die Trefferliste zu lang wäre, um sie vollständig auszugeben. Sie sollte nach einem bestimmten k-Wert abgebrochen und der Anwender darauf hingewiesen werden, dass für eine genauere Suche mehr Noten nötig wären. Sollte die Suchanfrage eine Länge von 9 überschreiten, so wird der überschüssige Teil abgeschnitten.

Nachdem wir festgelegt haben, wie die Anfrage und die gespeicherten Incipits auszusehen haben, kommt jetzt das wichtigste einer Suchmaschine, nämlich das Suchen der Anfrage in der Datenbank.

Für den Vergleich zwischen Anfrage und den Daten in der Datenbank wird die Editierdistanz verwendet. Dieses Verfahren verwendet auch die Suchmaschine von Musipedia. Aber anders als bei Musipedia können die n-grams bei eNoteHistory mehr Informationen enthalten. Es sind differenziertere Vergleiche nötig als bei Musipedia.

Bei der Editierfunktion gibt es drei Arten von Operationen. Diese sind das Löschen eines Zeichens, das Einfügen eines Zeichens und das Ersetzen eines Zeichens.

Ziel der Editierdistanz ist es, die minimalen Kosten zu ermitteln, die benötigt werden, um die Datenbankzeichenketten in die Anfragezeichenkette umzuwandeln. Jeder Operation wird ein Gewicht, bzw. Kosten $w(x, y)$ zugeordnet. („-,“ steht für eine Lücke; solche entstehen zum Beispiel, wenn die Anfrage kürzer als die n-grams in der Datenbank ist)[12]

(-, y) Einfügen von y in die Datenbankzeichenkette gegenüber der Anfrage
 (x, -) Löschen von x in der Anfrage
 (x, y) Ersetzen von x durch y

(x, x) Matchsituation: es sind keine Änderungen nötig, da das Zeichen der Anfrage mit dem Zeichen der Melodie in der Datenbank übereinstimmt

Die Kosten w wären also: $w(-, y) = w(x, -) = w(x, y) = 1$ und $w(x, x) = 0$

Bei jeder Mismatch-Situation, also jedem Anwenden einer Operation, werden die Kosten für die Editierdistanz D erhöht. [12] Damit ist auch klar, dass umso geringer die Editierdistanz, desto ähnlicher sich die Anfrage und das n -gram der Melodie in der Datenbank sind.

Die Frage ist nun, wie groß die Editierdistanz werden darf, damit die Melodie noch in der Trefferliste auftaucht. Da es von verschiedenen Musikstücken unterschiedliche Versionen gibt und man sich beim Notenwert nicht der richtigen Angabe sicher sein kann, sollten auch Fehler in der Anfrage berücksichtigt werden. Ein anderer Grund für die Berücksichtigung von Fehlern kann sein, dass keine oder zuwenig genaue Treffer gefunden wurden. Fehler können falsche Notenwertangabe oder falsche Tonhöhenangaben sein. Natürlich darf die Anzahl der Fehler eine bestimmte Grenze nicht überschreiten. Allerdings sollten Fehler in der Notenwertangabe geringer gewichtet sein, als Fehler in der Tonhöhenangabe. Beziehungsweise sollten sie ganz vernachlässigt werden, wenn nur die Notenwerte nicht stimmen, aber die Tonhöhe exakt angegeben wurde.

Das Berechnen der Editierdistanz einer Anfrage mit jedem Eintrag in der Datenbank, ist sehr umständlich und zeitaufwendig. Um die Effizienz der Suchmaschine zu steigern, setzt Musipedia die Vantage-Point-Methode ein. Der Vantage-Index nutzt die Dreiecksungleichung aus. Durch ihren Einsatz wird vermieden, dass die Distanz zwischen der Anfrage und jedem Eintrag in der Datenbank berechnet werden muss.

Zur Verdeutlichung des Vantage-Objektes an dieser Stelle noch einmal die Abbildung aus dem Abschnitt 3.3.3.

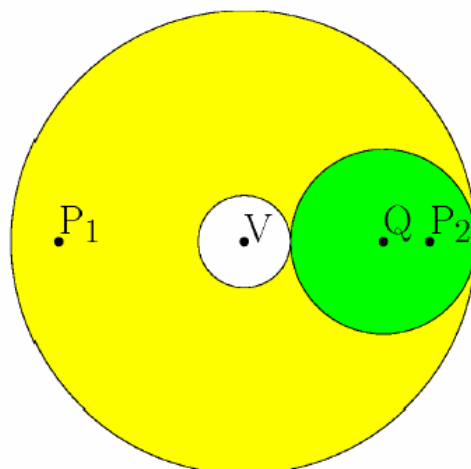


Abbildung 5.2: Vantage-Objekt

Wichtig für unseren Vergleich sind die Objekte im grünen Kreis, denn diese sind unserer Anfrage am Ähnlichsten, da sie einen annähernd gleichen Abstand zum Vantage-Objekt haben. Der Radius des grünen Kreises hängt mit der Anzahl der zugelassenen Fehler zusammen und diese Fehler bestimmen schließlich die Größe des Kreises.

Als Vantage-Objekt kann man z.B.: ein n-gram verwenden, das der Länge 9 entspricht und nur Nullwerte enthält.

Ist das n-gram einer gespeicherten Melodie kürzer als das Vantage-Objekt, werden die fehlenden Stellen mit Nullwerten aufgefüllt. Das gleiche passiert mit einer Anfrage, die nicht der Länge des Vantage-Objektes entspricht. Die Situation, das das n-gram der Anfrage oder einer Melodie länger als das Vantage-Objekt ist, kann nicht eintreten, da wir oben festgelegt haben, dass die n-grams höchstens die Länge 9 haben dürfen.

Bis entspricht der Vergleich der Anfrage und den Daten in der Datenbank noch mit der Musiksuchmaschine von Musipedia überein. Die Schwierigkeit bei eNoteHistory ist unsere n-grams komplexer sind. Jedes Tupel des n-grams besteht aus drei Teilen, dem Notenwert, der Tonhöhe und der Oktaveneinteilung. Alle drei Teile müssen gesondert mit den entsprechenden Teilen des n-grams in der Datenbank verglichen werden.

Anfrage: Die Anfrage besteht aus einem n-gram der Länge 6.

16 a 2 | 8 g 2 | 4 e 2 | 4 **fis** 2 | 8 e 2 | 4 a 1

Melodie aus der Datenbank:

4 a 2 | 8 g 2 | 4 e 2 | 4 **f** 2 | 8 e 2 | 4 a 1

Beim Vergleich der Anfrage mit der Melodie aus der Datenbank ist es wichtig, dass die richtigen Teile der Anfrage mit der der zu suchenden Melodie verglichen werden. Die im Beispiel rot gekennzeichneten Stellen, sind die Stellen, bei denen besonders darauf geachtet werden muss, dass die richtigen Teile miteinander verglichen werden.

Nachdem die zutreffenden Musikstücke aus der Datenbank gefiltert wurden, werden sie jetzt in einer Ergebnisliste präsentiert. Die Ergebnisse werden nicht wahllos angezeigt, sondern unterliegen einem bestimmten Ranking. Als Erstes werden die Musikstücke nach der Genauigkeit ihrer Übereinstimmung mit der Anfrage aufgelistet. Melodien die in Länge, Tonhöhe und Tonwert mit der Anfrage genau übereinstimmen kommen zuerst. Als nächstes werden die Melodien aufgelistet, bei denen die Editierdistanz nicht gleich der Editierdistanz der Anfrage entspricht. Incipits mit gleicher Editierdistanz werden alphabetisch sortiert.

Diese Art der Angabe der Treffer wird solange fortgesetzt, bis alle Melodien aufgelistet wurden, die die maximale Anzahl an Fehlern nicht überschreiten. Bei zu vielen Treffern wird die Auflistung nach 50 Treffern abgebrochen. Diese Art des Ranking entspricht in etwa dem des Programms von C-Brahms.

Die anschließende Abbildung 5.3 soll die Arbeitsweise des Konzeptes verdeutlichen.

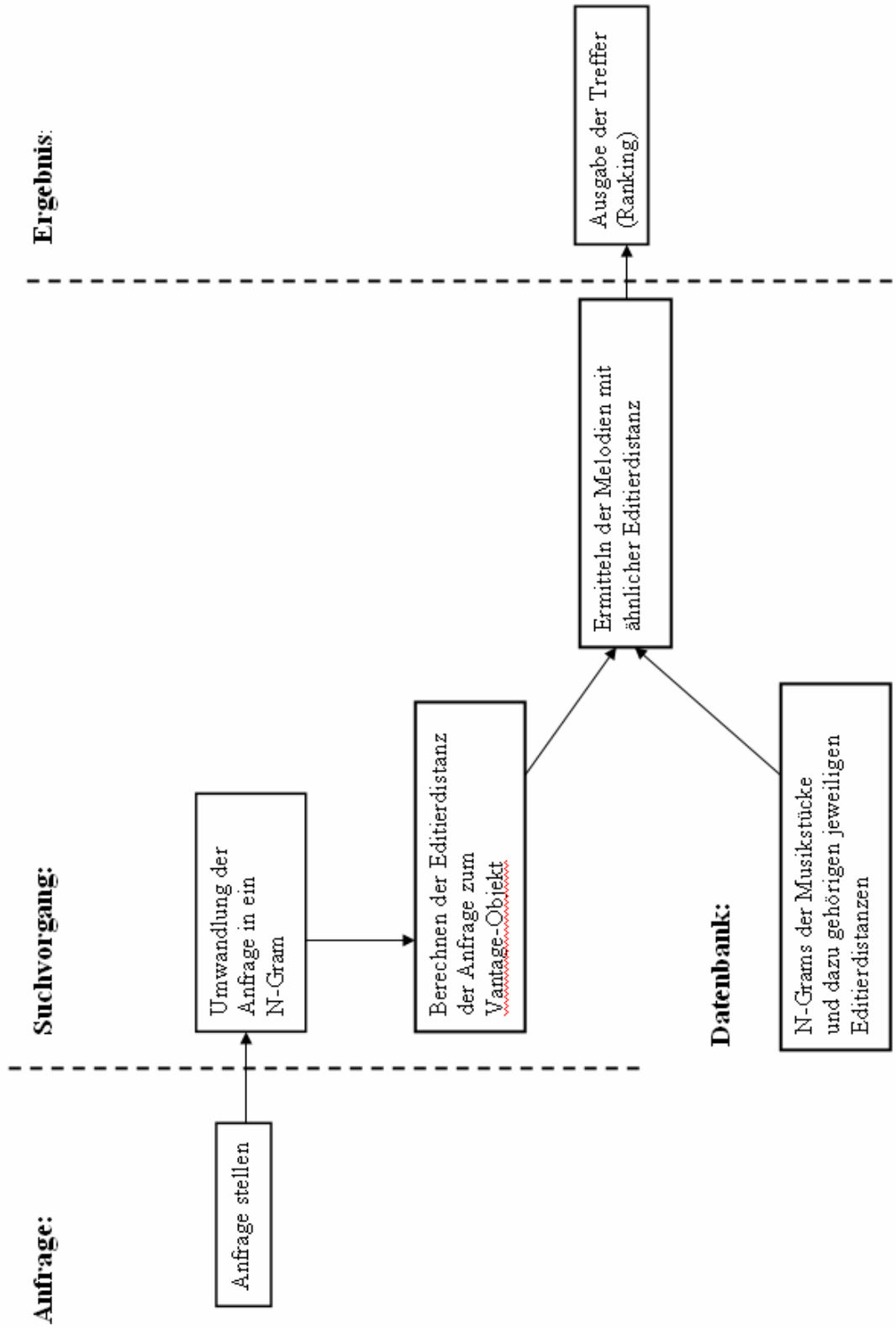


Abbildung 5.3: Konzept-Grafik

6 Zusammenfassung und Ausblick

In dieser Studienarbeit wurden Konzepte zur Suche von Musikstücken mit Hilfe von Suchmaschinen vorgestellt. Das Ziel von Music Information Retrieval ist es, Melodien aus einem Musikbestand herauszufiltern, die einer Anfrage melodisch ähnlich sind. Die Verwendung von Musiksuchmaschinen kann für die unterschiedlichsten Nutzer von Interesse sein. Sie können zum Auffinden von Noten und Aufnahmen in einer Musikbibliothek oder einem Musikladen von Nutzen sein. In der Forschung können Musiksuchmaschinen helfen Notenschriften einem Komponisten und einem Musikstück zuzuordnen.

Die in der Studienarbeit vorgestellten Konzepte sind auf diese unterschiedlichen Anwender zugeschnitten. Das Programm Musipedia, mit ihrem einfachen Konzept, ermöglicht auch Anwendern mit geringen Notenkenntnissen nach einer Melodie zu suchen.

C-Brahms ist dagegen schwieriger zu handhaben, da es Notenwert und Tonhöhe voraussetzt. Das Programm gibt aber einen guten Überblick verschiedener Suchmethoden. Der Anwender kann eine Suchmethode wählen, die melodieübergreifend sucht, was bei polyphoner Musik Vorteile haben kann.

Die Suchmaschine von Themefinder legt viel Wert auf unterschiedliche Möglichkeiten der Anfragestellung. Der Anwender hat die Wahl zwischen 9 verschiedenen Arten seine Anfrage zu stellen.

Das Konzept, welches für eNoteHistory im Rahmen dieser Studienarbeit erstellt werden sollte, enthält Einflüsse von allen drei vorgestellten Programmen. Hauptsächlich ist es aber eine Weiterentwicklung von Musipedia. Anders als bei Musipedia werden die Melodieeigenschaften genauer abgespeichert und auch in die Suche mit einbezogen.

An einigen Stellen ist das Konzept noch erweiterbar. So wäre es denkbar noch weitere musikspezifische Details abzubilden, z.B. punktierte Noten oder auch Haltebögen, die in der Notenschrift vorkommen. Der Suchalgorithmus könnte auch stimmenübergreifend programmiert werden. Es liegt also noch Ausbaupotential in diesem Konzept.

Die Entwicklung der letzten Jahre hat gezeigt, dass große Fortschritte in vielversprechenden Projekten im Bereich Music Information Retrieval erzielt wurden. Es ist abzusehen, dass in Zukunft Musikbibliotheken auf diese Systeme zurückgreifen werden.

Literaturverzeichnis

- [1] Frans Wiering. Melodic Features and Retrieval, Musicology 3-4
ISMIR Graduate School, Barcelona 2004
- [2] Prof. Dr. Bernd Enders, Jürgen Frey, Dr. Ralph Nickles, Thomas Welke. Schülerduden
Musik, 3. Auflage, Dudenverlag, Mannheim, 2000
- [3] Prof. Dr. Michael Clausen. Grundlagen des Multimediaretrievals I, Kapitel 3,
Universität Bonn, WS 2003/2004
<http://www-mmdb.iai.uni-bonn.de/lehre/materialMMRws0304/skriptMMRI.pdf>
- [4] Prof. Dr. Härder. Multimediale Informationssysteme, Audio / Video Retrieval,
Universität Kaiserslautern, SS 2002
<http://ahr.informatik.uni-kl.de/courses/seminar/SS2002/ausarbeitung12.pdf>
- [5] Department of Computer Science. <http://www.cs.helsinki.fi/group/cbrahms/index.html>
- [6] C-Brahms. Algorithm references [1-3], [4-6],[8],[9],
<http://www.cs.helsinki.fi/group/cbrahms/demoengine/>
- [7] about Themefinder. <http://www.themefinder.org/help/about/>
- [8] Themefinder. <http://www.themefinder.org/>
- [9] The Humdrum „**kern“ Repräsentation. Ohio State University, School of Music,
<http://www.music-cog.ohio-state.edu/Humdrum/representations/kern.html>
- [10] Rainer Typke. Music Information Retrieval mit Transportdistanzen, 2005
<http://www-mmdb.iai.uni-bonn.de/konferenzen/gi-symposium2005/Typke-MIRmitTransportdistanzen.pdf>
- [11] Der Parsons-Code. <http://de.musipedia.org/pcnop.0.html?&L=2>
- [12] Prof. Dr. E. Rahm. Algorithmen und Datenstruktur 2, Kapitel 4, Seite 14, Universität
Leipzig, SS 2002, <http://dbs.uni-leipzig.de/de/skripte/ADS2/HTML/kap4-14.html>
- [13] Donald Byrd, Tim Crawford. Problems of Music Information Retrieval in the Real
World, 2002

Abbildungsverzeichnis

2.1	Violinschlüssel und Bassschlüssel	10
2.2	Tonsystem mit Oktaveneinteilung	10
2.3	monophone Notenfolge	11
2.4	polyphone Notenfolge	12
2.5	Klavierwalzenton	12
2.6	Notenbeispiel Anfrage	13
2.7	Klavierwalzendarstellung Anfrage	13
2.8	Notenbeispiel Resultat	13
2.9	Klavierwalzendarstellung Resultat	14
3.1	Splitting	17
3.2	Auszug Trefferliste Themefinder	20
3.3	Humdrum-Code	21
3.4	Notenbeispiel Musipedia	25
3.5	visualisierter Parsons-Code	25
3.6	Vantage-Objekt	26
5.1	erste Konzept-Grafik	32
5.2	Vantage-Objekt	35
5.3	Konzept-Grafik	37

[2.2] Tonsystem mit Oktaveneinteilung.
URL:<http://www.mvkuenten.ch/Inhalte/Musiktheorie/Bezeichnung%20der%20Toene.html>

[2.6] Notenbeispiel Anfrage.
<http://www.cs.helsinki.fi/group/cbrahms/demoengine>,
Algorithm references [1-3], Seite 2

[2.7] Klavierwalzendarstellung Anfrage.
<http://www.cs.helsinki.fi/group/cbrahms/demoengine>
Algorithm references [1-3], Seite 3

[2.8] Notenbeispiel Resultat.
<http://www.cs.helsinki.fi/group/cbrahms/demoengine>,
Algorithm references [1-3], Seite 2

- [2.9] Klavierwalzendarstellung Resultat.
<http://www.cs.helsinki.fi/group/cbrahms/demoengine/>,
Algorithm references [1-3], Seite 3
- [3.1] Splitting.
<http://www.cs.helsinki.fi/group/cbrahms/demoengine/>
Algorithm references [8], Seite 3
- [3.2] Auszug Trefferliste Themefinder.
Auszug aus der Themefinder-Resultatenliste bei der Anfrage: „A G E“,
URL: <http://www.themefinder.org/cgi-bin/themerresults?session=05418422&page=1>
- [3.3] Humdrum-Code.
Humdrum-Code: Johannes Brahms Konzert in A Moll
URL: <http://www.themefinder.org/cgi-bin/themeinfo?istn=572916879661&type=humdrum>
- [3.6] Vantage-Objekt.
URL: <http://www-mmdb.iai.uni-bonn.de/konferenzen/gi-symposium2005/Typke-MIRmitTransportdistanzen.pdf>, Seite 18
- [5.2] Vantage-Objekt.
URL: <http://www-mmdb.iai.uni-bonn.de/konferenzen/gi-symposium2005/Typke-MIRmitTransportdistanzen.pdf>, Seite 18

Tabellenverzeichnis

3.1	Suchmethoden	16
4.1	Auswertung Mindestanfragegröße	29
4.2	Auswertung genaue Treffer	30
4.3	Konzept-Vergleich	30
5.1	Vergleich relative Tonhöhenangabe	32
5.2	Vergleich absolute Tonhöhenangabe	33
5.3	Auflistung Notenwertangaben	33
5.4	Beispiel n-gram	34

[3.1] Suchmethode. übersetzt nach der Tabelle auf der Internetseite
URL: <http://www.cs.helsinki.fi/group/cbrahms/demoengine/>

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Vorlage der angegebenen Literatur und Hilfsmittel angefertigt habe.

Rostock, 22. Dezember 2005