Inhaltsbasierte Musiksuche: Konzepte und Anwendungen

Dmytro Shaykhit
shaykhit@ifi.lmu.de

Universität München
Medieninformatik
Amalienstrasse 17, 80333 München, Deutschland

Zusammenfassung Die bibliographische Information (z.B. Titel, Name des Komponisten) stellt nur einen Parameter dar, der oft bei Musiksuche verwendet wird. Es existieren aber noch weitere Parameter, die aus dem Inhalt (*content*) des Musikwerkes gewonnen und bei der Suche implizit oder auch explizit angegeben werden können. In dieser Arbeit befasse ich mich mit den Merkmalen (sogenannten Features) einer Melodie (wie z.B. Tonhöhe, Rhytmus, Klangfarbe), dem Suchprozess selbst und der praktischen Anwendung der inhaltsbasierten Suche. Den Suchprozess über den Inhalt habe ich in mehrere Schritte aufgeteilt: a) Vorsummen (*query by humming*), ein Musikfragment oder Features werden als Angaben benutzt. b) Die eingegebenen Daten werden in eine passende Darstellung umgewandelt, somit entsteht die Anfrage (*query*). c) *Query object* wird mit der Information aus Musikdatenbanken gematcht. Die weitverbreiteten Matchingalgorithmen sind zeichenkettenbasierte Methoden für monophonische Musik, mengenbasierte Methoden und *probabilistic matching*. d) eine Liste der „ähnlichen" Musikwerke wird zurückgegeben. Ferner stelle ich einige MIR Systeme und Ziele die sie verfolgen vor.
Stichwörter: *music information retrieval*, *music information retrieval system*

1 Einleitung

Für die Suche nach einem Musikwerk wird heutzutage meistens die präzise Eingabe der textuellen (bibliographischen) Information benutzt. Das ist aber nur dann möglich, wenn man genau weiß, was man sucht. Mit jedem Jahr erscheint eine Vielzahl der neuen Melodien, Musikwerke in verschiedener Ausführungen, Remix-Versionen, entstehen neue Bands etc., so dass sich in naher Zukunft die Suche per Text sich als sehr schwierig oder sogar unmöglich erweisen wird. Deshalb findet der Musik-Content (Die Information die die Musik selbst beschreibt, z.B. Noten) immer mehr Beachtung und wird schon bei einigen Musik-Datenbanken als ein weiterer Eingabeparameter benutzt. DFKI (Deutsches Forschungszentrum für Künstliche Intelligenz) definiert den Begriff „MIR" folgendermaßen: „Musik Information Retrieval stellt ein interdisziplinares Forschungsgebiet dar, das bereits seit Ende 1960 existiert und sich mit der zunehmenden Verbreitung des Internet und entsprechenden digitaler Formate (bspw. MP3) zunehmend etablieren konnte. Die zu lösenden Kernprobleme umfassen Technologien, die dem Menschen einen effizienten Zugriff auf umfangreiche Musikkollektionen ermöglichen." [10]
Bei der inhaltsbasierten Suche geht man davon aus, dass der Content der Musik aus vielen nützlichen Komponenten besteht, wie z.B. Tonhöhe, Rhythmus

und Klangfarbe. Die Musik-Komponenten versucht man so genau wie möglich aus dem Inhalt zu extrahieren, so dass die Kombination aus diesen Merkmalen die Musik eindeutig darstellen kann. Normalerweise speichern die Musikdatenbanken und MIR-Systeme, die die inhaltsbasierte Suche unterstützen, für jedes ihrer Musikstücke noch zusätzlich eine entsprechende Darstellung, und stellen ein Tool zu Verfügung, das die Anfrage (ferner auch als *query* bezeichnet) in ein passendes Format konvertiert, sodass das Matching möglich wird.
Es gibt viele Anwendungen, die es ermöglichen, eine Anfrage in Form von Vorsummen zu stellen. Z.B. wird in [1] die Anwendung *query by humming* benutzt um das Vorsummen zunächst ins MPEG7-Format umzuwandeln und dann als eine gewöhnliche Anfrage beim Matching zu benutzen.
Es stellt sich die Frage: „wie funktioniert Matching und was bedeutet es eigentlich?“. Matching bedeutet eine oder mehrere Musikwerke zu finden, die mit der Anfrage mit einer gewissen Toleranz änlich sind. Es gibt dazu zahlreiche Matching-Algorithmen und die Auswahl eines geeigneten Algorithmus hängt von der Art ab, wie die Musikdaten dargestellt und gespeichert werden.
Zunächst stelle ich die Merkmalen vor, die aus dem Musik-Content gewonnen werden können. Dann beschreibe ich den Suchprozess und am Ende gebe ich einen kürzen Überblick über einige existierenden MIR-Systeme und deren Aufgaben.
In dieser Arbeit verwende ich viele musikalische Begriffe, die nicht jedem verständlich sein können. Für eine detaillierte Erklärung verweise ich den Leser auf die Musiklehre (siehe z.B. [9]).

2 Verwandte Arbeiten

Das Fach befindet sich heutzutage in einer Entwicklungsphase und derzeit gibt es fast kein einziges Buch, das die Prinzipien, die hinter MIR stecken, ausführlich behandelt und eine gute theoretische Grundlage darstellt. Es existieren allerdings zahlreiche Studien, Konferenzen, technische Berichte, Seminare, die sich mit dem Thema auseinander setzen. Hier ist wichtig die Arbeit von Downie [6] und das Buch „Introduction to MPEG-7“ [1] zu erwähnen. [6] gibt einen guten Überblick über den Stand der Entwicklung im MIR-Bereich im Jahr 2003 und klassifiziert MIR-Systeme in 2 Typen: *Locating MIRS* und *analytic/production MIRS*. [1] spezifiziert ein Standard für die Audiodarstellung im MPEG-7-Audio-Format und gibt einen Überblick über solche Anwendungen wie z.B. *query by humming*, die schon ein Teil des Standards geworden sind. Auf diese Quellen werde ich oft in meiner Arbeit verweisen.

3 Musikfeatures

Die Musikinformation (Content) kann man grundlegend in 7 Komponenten (Features) zerlegen [6]: Tonhöhe, Tempo, Harmonie, Klangfarbe, redaktionelle, textuelle und bibliographische Information, wobei die einzelnen Features sich nicht gegenseitig ausschließen, zudem ist es schwierig, wegen des ständigen Zusammenspiel der Features, eine ausführliche Analyse für jedes musikalische Merkmal

durchzuführen. Z.B kann bei der Analyse der durch Noten dargestellten musikalischen Daten der Satz „adagio“ in einer Partitur sowohl dem Tempo als auch der redaktionellen Information zugeordnet werden [6].

3.1 Tonhöhe

Die Tonhöhe ist ist die Funktion der fundamentalen Frequenz des gespielten Tons, d.h. der Anzahl der Oscillationen pro Sekunde. [6]. In der Musikwissenschaft gibt es dafür viele Darstellungen. Eine der bekanntesten ist die graphische Darstellung, wo jede Note gemäß ihrer Tonhöhe vertikal positioniert wird. Die Differenz zwischen zwei Tonhöhen wird als Intervall bezeichnet.
Dieses musikalische Merkmal wird meistens aus der akustisch dargestellten Musik mithilfe von Fourier-Transformationen extrahiert. In [2] werden z.B. die Frequenzamplituden mit der Anwendung von FFT (Fast Fourier Transform) an einem 16 Sekunden langen Musikstück bei der Samplingsrate von 10 Samples pro Sekunde berechnet und entsprechend normalisiert.

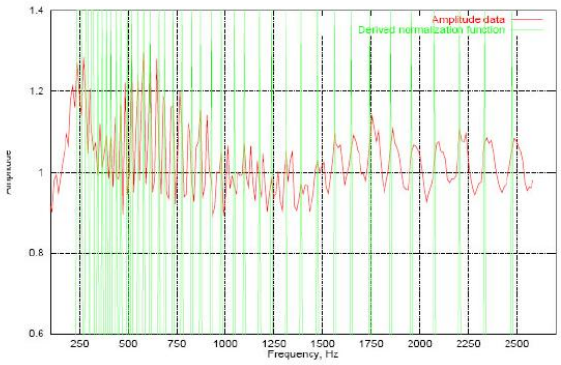

Abbildung 1. Normalized frequency amplitudes and the notes of the western scale [2]

3.2 Tempo (Rhythmus)

Das Tempo betrifft die Dauer der musikalischen Ereignisse. Meter, Notendauer, Harmoniedauer und Akzente setzten die Rhythmuskomponente des Musikstückes zusammen [6]. Spezielle Eigenschaften von Rhythmus werden durch die Wahrnehmungsart der Menschen beeinflusst: z.B. werden die schnellen Melodien energisch wahrgenommen während die langsamen eher friedlich sind [2]. Die Rhythmen einiger Genres unterscheiden sich gravierend voneinander (vgl. Jazz und Hard-Rock). Es gibt viele Möglichkeiten die Tempoinformation darzustellen: absolut(z.B. Metronom-Angabe, MM=80), allgemein(z.B. adagio, presto, fermata) oder relativ(z.B. schneller, langsamer) [6]. Bei der Identifizierung der Musik spielt der Rhythmus eine wichtige Rolle, weil er eine Dimension darstellt, in der die Melodien im allgemeinen nicht transformiert oder verzerrt werden können [1]. D.h., es kann z.B. ein Musikwerk mit verschiedenen Instrumenten

gespielt werden und dadurch verschiedene Klangfarben und eine mögliche Verzerrung der Noten als Folge haben, wobei der Rhythmus gleich behalten wird. Jeder normale Mensch ist somit imstande zu begreifen dass es sich um das gleiche Musikwerk handelt.

3.3 Harmonie (Polyphonie)

Harmonie tritt auf, wenn zwei oder mehr Tönen simultan, also gleichzeitig klingen. In der Literatur wird oft dazu die Bezeichnung „Polyphonie“ benutzt [6]. Wenn gleichzeitig nur ein Ton klingt, so heißt es „Monophonie“. Standardgemäß wird Harmonie in der Musiknotation durch die verschiedenen vertikalen Positionen der Noten auf eine Stelle beschrieben. Außerdem können die einzelnen harmonischen Ereignisse durch eine Kombination von Tonhöhen oder Intervallen und der Position des Haupttons (bzw. fundamentalen Tons) auf der Tonleiter ausgedrückt werden. Der Akkord ist ein Beispiel für Harmonie und besteht in der Regel aus drei oder mehreren Noten, die simultan klingen. Die Akkorde können Gruppen bilden und ihre eigene Bezeichnungen haben [9].

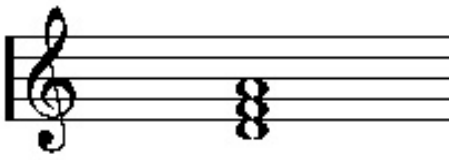

Abbildung 2. Akkord aus drei Noten [9]

Der Zugang zu den Aspekten der harmonishen Information kann auch für die durch Noten dargestellten Melodien problematisch sein, da sie meistens nicht explizit in den Musikwerken erwähnt werden, obwohl schon in der Partitur vorhanden sind. Die Ausnahme ist das Einfügen von Akkordbezeichnungen oder Akkordsymbolen direkt in die Noten der meist bekannten (klassischen) Musik. [6]

3.4 Klangfarbe

Die Klangfarbe ist das wahrgenommene Merkmal, das den Klang zweier Töne mit der gleichen Tonhöhe und Lautstärke unterschiedlich macht [1]. Wegen des Unterschiedes in der Klangfarbe klingt eine Note mit Piano gespielt anders als mit Flöte. Im allgemeinen hat jedes musikalische Instrument eine eigene Klangfarbe. Die Zuordnung eines Instruments zu einem Musikfragment oder dem ganzen Musikwerk wird unter Orchesterinformation gespeichert und meistens in die bibliographische Information in Form einer Mapping-Liste von Instrumenten und den dazugehörigen Fragmenten hinzugefügt. Deshalb kann die Orchesterinformation sowohl ein Teil des Klangfarbe-Features als auch der bibliographischen Information sein [6].

3.5 Redaktionelle Information

Die redaktionelle Information bezieht sich auf die Verfeinerung eines Musikstückes. Das sind Aspekte der Musikverschönerung, Artikulationen, *staccati*, dynamische Instruktionen, Haltebögen, Bogenführungen, usw. [6,9]. Die redaktionelle Information kann symbolisch (z.B. -,3,!), textuell (z.B. crescendo, diminuendo) oder in den beiden Varianten dargestellt werden. Ferner können die einzelnen Musikfragmente hier auch enthalten sein [6]. Somit macht die Identifikation der richtigen Version des Musikwerkes für MIR-Systemen beim Vorhandensein einer großen Diskrepanz im redaktionellen Teil große Schwierigkeiten.

3.6 Textuelle Information

Die textuelle Information schließt die Lyrik der Lieder, Hymnen, Sinfonien, Chören, etc. ein. Die Texte der Lyrik sind meistens unabhängig von der Musik selbst. Die Begründung dafür ist die Tatsache, dass z.B. ein Lied mehrere Texte und umgekehrt ein Text mehrere musikalische Varianten haben kann. Deshalb ist ein gegebenes Lyrikfragment meistens nicht informativ genug um eine gewünschte Melodie aus einem MIR-System zu gewinnen [6]. Man darf auch nicht vergessen, dass es eine große Anzahl von Musikwerken ohne irgendwelchen Text gibt [6].

3.7 Bibliographische Information

Titel des Werkes, Name des Komponisten, Autor des Textes, Veröffentlichungsdatum, der Interpret sind Beispiele für bibliographische Information. Diese Information ist von dem Inhalt des Musikwerkes unabhängig und gehört eher zu den Metadaten der Musik. Alle Schwierigkeiten, die die Art und Weise der bibliographischen Beschreibung und Zugang betreffen, haben hier ihre Relevanz [6]. Heutzutage ist dies die am meisten benutzte Methode, wenn man nach einem Musikwerk im Internet sucht oder eine Anfrage an Musikdatenbanken stellt.

4 Suchprozess

Den Suchprozess nach einem Musikwerk in einer Multimediadatenbank kann man in folgenden Schritten aufteilen:

- Eine oder mehrere Komponenten der Musikinformation werden entweder direkt oder in Form von Vorsummen, Musikfragment oder *beatboxing* (Pulsieren der Lippen und Gurgel) eingegeben.
- Umwandlung der eingegebenen Daten in eine passende Darstellung und Erstellung eines Query-Objektes.
- Das Query-Objekt wird mit der Information aus Musikdatenbanken gematcht. In diesem Schritt finden die Matching-Algorithmen ihre Anwendung.
- Rückgabe einer Liste mit den ähnlichen Musikwerken.

Im folgenden werden einige Aspekte der Musiksuche detailliert vorgestellt.

4.1 Eingabemöglichkeiten

Es existieren MIR-Systeme, die sich mehr an Musik-Wissenschaftler orientieren und für eine Melodiedarstellung alle früher beschriebene Musikfeatures verwenden. Die Entwickler solcher Systeme haben zum Ziel einen Zugang zu allen Musikkomponenten bereitzustellen und eine Möglichkeit für eine ausführliche Analyse zu geben. Dabei ist es möglich die Musikfeatures direkt einzugeben um nach einer erwünschten Melodie zu suchen. Z.B. werden in der RISM-Datenbank spezielle Indizes, sogenannte „*music incipit*“, gespeichert. *Incipits* werden in alphanumerischer Notation mit dem von Brook entwickelten *plaine and easie code* kodiert und enthalten Tonhöhen- und Tempoinformation [6]. Die Anfragen müssen dann auch mit *plaine and easie code* in Form von Zeichenketten formuliert werden um bestimmte Tonhöhen-Tempi-Kombination als Suchkriterien für *incipits* anzugeben. Hier ist ein Beispiel für eine solche Anfragen:

```
%F-4$bB@3/8#'8C.6.3$,B'C&/,8A'D6(-)D/,8G'8.C,6B/8F
```

Diese Zeichenkette entspricht dem *incipit* für Mozarts *II core vi dono* aus *Cosi fan tutte* in der RISM Datenbank [6].

Ein anderes Beispiel ist David Hurons Humdrum Toolkit, der aus mehr als 50 in Wechselbeziehung stehenden Programmen für die Konstrutkion aller möglichen Anfragen im UNIX-Format besteht [6]. Die Anfragen im Humdrum erfordern aber sehr gute Kenntnisse im UNIX-Bereich. Z.B für die Suche nach den Stellen, wo ein gegebenes Motiv auftaucht, könnnen folgende Befehle gebraucht werden:

```
extract -i'**kern' HG.kern | semits -x | xdelta -x | xdelta -s = |
patt -t Motive1 -s = -fMotive1.pat | extract --i**'pat' |
assemble HG.krn
```

Eine weitere Möglichkeit der Suche nach einem gewünschten Musikwerk über seinen Inhalt liefern uns die QBH (*query by humming*) Anwendungen. Hier wird die Suche durch Vorsummen oder Singen der Melodie des gewünschten Musikwerks ermöglicht. Die meisten Menschen können durch das Vorsummen ganz leicht das ganze Werk ohne zusätzliche Information richtig identifizieren. Das Vorsummen einer Melodie kann sogar fehlerhaft sein und trotzdem stört es uns nicht das richtige Ergebnis zu finden. Diese Aufgabe scheint aber nicht so trivial zu sein, da sie viele Features der Musik, wie z.B. Lyrik, Klangfarbe, Tempo und Rhythmus umfasst.

[1] gibt einen Vorschlag für das QBH-System: Es werden 2 Datenbanken verwendet - die erste zur Speicherung von komprimierten MPEG4-Audio-Dateien und die zweite zur Speicherung der MPEG7-Darstellung und eines Links auf die jeweilige Datei in der ersten Datenbank. Der Benutzer kann an einem Computer, der an einen QBH-Server angeschlossen ist, die Query-Taste drücken und eine Melodie vorsummen. Das Signal wird dem Query-Server übergeben, wo die notwendigen MPEG7-Metadaten extrahiert werden. Diese Metadaten werden dann als Anfrage benutzt, um in der MPEG7-Datenbank die ähnlichen Daten zu finden. Die Top-Liste der ersten Treffer wird an den Benutzer geschickt, wo er einen davon auswählen und das Streaming ins MPEG4-Format beginnen kann.

In [3] wird das Vorsummen als Eingabe benutzt und durch ein Anfrageverarbeitungsmodul in eine passende Darstellung umgewandelt. Diese Daten werden dann mit den MIDI-Dateien aus der Musikdatenbank gematcht. Die Abbildung 3 zeigt die Komponenten dieses Systems.

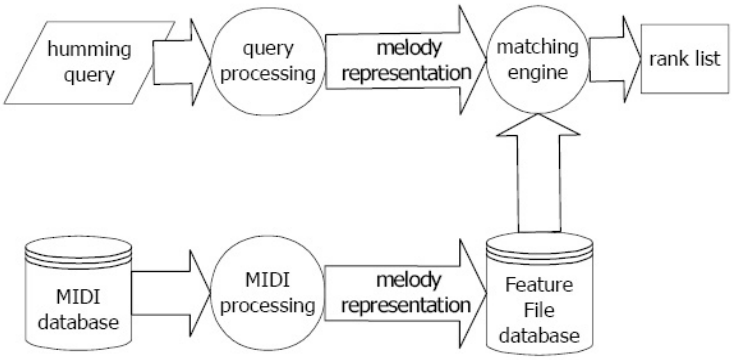

Abbildung 3. QBH System Flowchart [3]

Als Eingabe können auch Musikfragmente benutzt werden, um z.B. das gleiche Genre zu finden. In [4] wird gezeigt, wie man für die polyphonischen Musikwerke des gleichen Genres eine Ähnlichkeit findet. Ein Benutzer eines QBE(*query by example*)-Systems kann eine eigene Spielliste nach seinem Geschmack und Bedürfnissen aufbauen.

Außer Vorsummen, verschiedener Musikfeatures und Musikfragmente für die inhaltsbasierte Suche, wird auch in letzter Zeit das sogenannte *beatboxing* für bestimmte Genres verwendet. *beatboxing* ist die Simulation eines Schlaginstrumentes, indem man mit Hilfe von Lippen, Gurgel und Mund die Töne (künstlich) erzeugt [7]. Diese Art und Weise Töne zu erzeugen ist in den 1980-er Jahren entstanden, als die Schlaginstrumente sehr teuer waren und die Jugend sich eine solche Technik noch nicht leisten konnte. Es gibt im allgemeinen keine Grenze für die Bandbreite der imitierenden Musikinstrumente und somit kann eine Vielzahl von verschiedenen Tönen, Melodien u.ä. erzeugt werden [7]. Der Musiker deckt z.B. oft seinen Mund mit einer Hand zu, um lautere, tiefere Töne zu erzeugen.

Bionic beatboxing voice processor in [7] ermöglicht es dem Benutzer in sein Mikrophon zu beatboxen um mit Hilfe von Interface und vorher aufgenommener Audio-Samples ein Drum-Loop(stark rhythmische Töne die durch verschiedene Schlaginstrumenten erzeugt wurden) höherer Qualität zu erzeugen. Für DJs ist das von großem Interesse - es eröffnen sich damit neue Perspektiven für ein innovatives Umgehen mit der Musik.

4.2 Erstellung einer Anfrage. Die Anfragevearbeitung

Es ist meistens notwendig die eingegebene Musikinformation in eine passende Darstellung umzuwandeln oder im Fall der akustischen rohen Daten, als eine Menge von Feature-Vektoren darzustellen. Die somit angefertigte Anfrage

werde ich ferne als Query-Objekt bezeichnen. Vor der eigentlichen Anfrageverarbeitung muss auch die äquivalente Formatdarstellung für alle Musikwerke in der Datenbank vorhanden sein. wie ich schon früher erwähnte, gibt es z.B. im Vorschlag von [1] für ein QBH-System zwei Datenbanken: die erste für die eigentliche Speicherung der Musikwerke in MPEG4 Format und die zweite zur Speicherung der Musikdarstellung ins MPEG-7-Format.

Query by humming Das Query-Objekt im Vorschlag von [1] für ein QBH-System wird durch die Umwandlung des eingegangenen Signals ins MPEG-7-Audio-Format erzeugt. Der MPEG-7-Audio-Standard bietet dafür das spezielle Beschreibungsschema - *MelodyContour*, das sich nur auf die Repräsentation der monophonischen Musik beschränkt. Die Tonhöhen selbst werden nicht für die Darstellung benutzt, weil bei der Veränderung der Tonart die Identifikation eines Musikstücks nicht mehr möglich ist. Stattdessen gibt die *MelodyContour* das Tonhöhenverhalten für die ganze Melodie von Note zu Note an: ob z.B. die nächste Note höher als die vorherige ist oder auf dem gleichen Niveau bleibt [1]. In der Tabelle 1 sind fünf Stufen der *MelodyContour* aufgelistet, die für die Beschreibung des Tonhöhenverhaltens benutzt werden.

Tabelle 1. The five levels of contour information in MelodyContour Description Scheme [1]

Contour value	Change in interval
-2	Descent of a minor-third or greater
-1	Descent of a half step or whole step
0	No change
1	Ascent of a half step or whole step
2	Ascent of a minor-third or greater

Außerdem wird in der *MelodyContour* die Rhythmusinformation gepeichert, d.h. Zeitsignaturen und Takt der Melodie.

In [3] werden drei Parameter für die Melodiedarstellung benutzt: Tonhöhenkontur (beinhaltet Tonhöheninformation), Tonhöhenintervall und die Dauer. Die Tonhöhenkontur wird durch Zeichen U oder D dargestellt, wobei U für eine höhere und D für eine niedrigere Note im Vergleich mit der vorherigen steht. Das Tonhöhenintervall ist die Differenz zwischen der Frequenzen von zwei aufeinanderfolgenden Noten. Und die Dauer bedeutet wie lange eine Note gespielt bzw. vorgesummt wird. Z.B. wird, für die in der Abbildung 4 gezeigte Melodie,

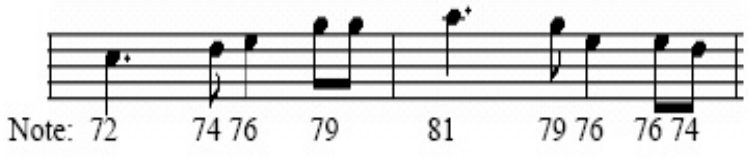

Abbildung 4. Ein Teil der Partitur eines Lieds [3]

folgende Daten abgespeichert:

```
(*,*,*)(U,64.1,1)(U,71.9,2)(U,124.7,2)(U,96.0,3)
(D,-96.0,1)(D,-124.7,3)(D,-64.1,1)
```

Die Melodiedarstellung kann aus dem Vorsummen-Signal z.B. mit dem in [3] vorgestellten Verfahren gewonnen werden. In diesem Verfahren werden zunächst die Stille und Geräusche entfernt, dann wird das Vorsummen-Signal in Segmente aufgeteilt und durch diese Segmente die Notenänderungen festgestellt. Ferner wird die Melodiedarstellung aus den Notenänderungen in das oben vorgestellte Format konvertiert (siehe die Abbildung 5). Für die genaue Beschreibung des

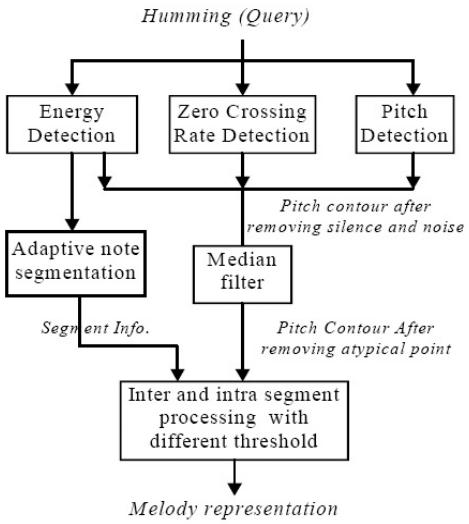

Abbildung 5. Query processing algorithm [3]

Algorithmen siehe [3].

Query by beatboxing Die Töne die durch *beatboxing* erzeugt werden, sind von kurzer Dauer (ca. 0.25 sec) [7] und aufgrund der polyphonischen Natur der gesuchten Melodien (*drum loops*) wird normalerweise ein Feature-Vektor für die Dauer des Tons berechnet. In [7] werden die folgenden Features für die Konstruktion des Query-Objekts vorgeschlagen:

- Zeitdomain Features: *ZeroCrossings*, *Root Mean Squared Energy* (RMS) und *Ramp Time*
- Spektraldomain Features: *Centroid, Rollof und Flux*
- *Mel Frequency Cepstral Coefficients* (MFCC)
- *Linear Predictive Coefficients* (LPC)
- Wavelet-basierte Features

Der Umfang dieser Arbeit erlaubt mir nicht die einzelne Features zu beschreiben, im allgemeinen stellen sie aber mathematische Konstrukte zur Signalanalyse dar. Die Autoren von [7] haben jedoch für die Analyse der *drum loops* ein

anderes Verfahren gewählt - *beat histogram* (Taktenhistogramm). Laut diesem Verfahren wird das Signal durch DWT(Discrete Wavelet Transform) in zwei separate Frequenzbänder aufgeteilt. *beat histrogram* (BH) zeigt die Verteilung der verschiedenen Taktenperiodizitäten des Signals. In der Abbildung 6 wird das Diagramm für ein Stück von Rhythm-und-Blues-Musik dargestellt. Die Hauptspitze im BH wird als Tempo für ein Signal ausgewählt. Zusätzlich zum BH kann jedes Frequenzband separat verarbeitet werden um die einzelnen Spuren des Musikstücks zu identifizieren.

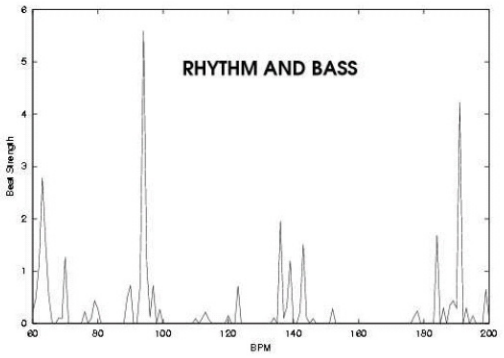

Abbildung 6. Beat Histogram [7]

Query by example Das Ziel der QBE-Anwendungen ist nicht die Suche nach einem ähnlichen Musikwerk sondern die Suche nach Musikstücken des gleichen Genres. In [4] wird zwischen zwei Ähnlichkeitsarten unterschieden: Lokale Ähnlichkeit - die Ähnlichkeit, die ein Mensch nach dem Hören von 1 oder 2 Sekunden zweier Musikstücke bemerkt, und Globale Ähnlichkeit - die Ähnlichkeit, die ein Mensch nach dem Hören von 10 oder 20 Sekunden der Exzerpte (ein Ausschnitt des Musikwerkes) bemerkt. Das Query-Objekt wird aus dem ein- bis zweisekündigen Musiksegment für die lokale Ähnlichkeit gebildet, indem man z.B., wie in [4] durch STFT(Short Term Fourier Transform), die Spektralvektoren aus diesem Segment extrahiert. Für die globale Ähnlichkeit z.B. für zwei zwanzigsekündige Segmente, werden die Segmente einfach auf 20 einsekündige Segmente aufgeteilt.

4.3 Matching-Algorithmen und Effizienz der Suche

Nachdem ein Query-Objekt erzeugt wurde und die Musikwerke in der Musikdatenbank eine passende Darstellung haben muss jetzt die Query mit den Daten der Datenbank gematcht werden. Abhängig von der Art der Musikdarstellung (symbolisch oder akustisch) und dem Typ der Suchanfrage (Exact-Match, approximiertes Matching, usw.) kann man die Matching-Algorithmen in 3 Klassen aufteilen: Zeichenkettenbasierte Methoden für monophonische Melodien, mengenbasierte Methoden für polyphonische Musik und wahrscheinlichkeitsorientierte Methoden (*probabilistic matching*).

Zeichenkettenbasierte Methoden Die monophonische Musik wird meistens durch Zeichenketten dargestellt, wo jedes Zeichen eine Note oder ein paar der nacheinander folgenden Noten beschreibt. Die Zeichenketten können auch die Tonhöhenintervalle, wie in [3] durch Angabe des Höhenzuwachs jeder Note im Vergleich mit der vorherigen, die bestimmte Folge der Töne oder ähnliches darstellen. Hier finden die Algorithmen für die Berechnung der „Stringdistanzen", die Suche nach der größten gemeinsamen Teilsequenz oder die Suche nach der bestimmten Stellen, wo eine Zeichensequenz in einer anderer Sequenz vorhanden ist, ihre Anwendung [5].
Beim approximierten Matching wird oft die Stringdistanz durch dynamische Programmierung berechnet, d.h. inwiefern die Zeichenketten sich voneinander unterscheiden. Die einfache Berechnung der Stringdistanz zwischen einem Query-Objekt und jedem Eintrag in der Datenbanken ist allerdings nicht ausreichend, da die Musikstücke eine unterschiedliche Länge haben können. Deshalb müssen zuerst die passende Zeichenketten konstruiert und dann die Distanzen überprüft werden [5]. Für *exact match query* (EMQ) können die speziellen Index-Strukturen, wie z.B. invertierte Listen, B-Bäume, B*-Bäume etc., für die Speicherung und Indizierung der Daten benutzt werden [5].
In [3] wird *hierarchical matching* benutzt. Diese Methode ist im Wesentlichen das approximierte Stringmatching mit einigen Adoptionen. Der Algorithmus wird in drei Schritten aufgeteilt:

1. Es wird ein approximiertes Matching und eine dynamische Programmierung benutzt um die Tonhöhenkontur zwischen der Anfrage und den Kandidaten abzugleichen.
2. Ferner wird die Ähnlichkeit im Tonhöhenintervall und in der Rhythmuskomponente zwischen der Anfrage und den Kandidaten berechnet.
3. Das Ergebnis wird aus der Kombination der vorherigen Schritte nach dem folgenden Schema berechnet:

$$Rank = (\alpha \cdot \texttt{rank\_i} + (1 - \alpha) \cdot \texttt{rank\_r}) \cdot \frac{N_{mis}}{N}, \text{ wobei}$$

`rank_i` - Ähnlichkeit im Tonhäheninterval
`rank_r` - rhytmische Ähnnlichkeit
α - Gewicht (0.7)
N - die Länge der Zeichensequenz

Die Abbildung 7 zeigt den Ablauf des Algorithmus.

Mengenbasierte Methoden Bei diesem Verfahren wird die Musik als eine Menge (Set) von Ereignissen bzw. Features mit bestimmten Eigenschaften, wie z.B. Tempo, Rhythmus, Tonhöhe etc., angesehen. In [2] wird jedes Musikstück als eine Menge von Feature-Vektoren dargestellt, wobei für jeden Vektor die Daten als Gleitkommazahlen in einem separaten Array gespeichert werden. Somit ist ein Musikstück ein Punkt in einem n-dimensionalen euklidischen Raum und zwei Musikstücke klingen ähnlich wenn ihre Mengen der Feature-Vektoren v1 und v2 innerhalb einer Distanz ε liegen (approximiertes Matching). Für jedes Paar wird die euklidische Distanz berechnet:

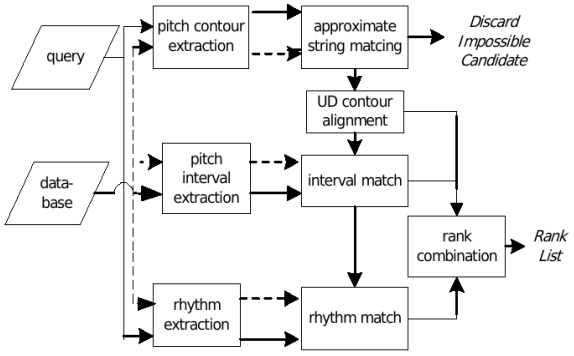

Abbildung 7. Hierarchical matching algorithm [3]

$$d = \sqrt{\Sigma(v_1^i - v_2^i)^2}, \quad d \leq \varepsilon$$

Es gibt auch Methoden für symbolisch dargestellte Daten, die die Musik als Menge von Noten ansehen [5]. Exaktes Matching wird dann durch die Suche nach der Obermenge des Query-Objekts und approximiertes Matching durch die Suche nach Obermengen der Untermengen des Query-Objektes ermöglicht. Für die Indizierung werden invertierte Listen und Dreiecksungleichung benutzt.

Probabilistic Matching Beim *probabilistic matching* werden probabilistische Eigenschaften der Kandidaten untersucht und mit der jeweiligen Eigenschaften des Query-Objektes verglichen. Z.B. berechnet GUIDO-System die Markov-Modelle, die die Wahrscheinlichkeiten der Zustandsübergänge in Musikstücken beschreiben [5]. Laut [5] werden die Musik-Features benutzt um die Markov-Ketten zu berechnen, wo die Zustände den Features, wie bestimmte Tonhöhe, Intervall oder Notendauer, entsprechen. Die Ähnlichkeit zwischen einem Query-Objekt und einem Kandidat in der Datenbank kann dann durch die Berechnung des Produktes der Wahrscheinlichkeiten für Zustandsübergänge basierend auf der Übergangsmatrix des Kandidaten für jedes Paar der nacheinander folgenden Zustände im Query-Objekt bestimmt werden [5]. Für die Indizierung wird *hierarchical clustering* benutzt, d.h. es wird eine Baumstruktur zur Speicherung der Übergangsmatrizen verwendet [5].

Effizienz der Suche Obwohl die Matching-Algorithmen für eine präzise und erfolgreiche Suche eine große Rolle spielen, verlangen sie viel Speicher und CPU-Zeit. Es gibt einige Arbeiten, die versuchen, die Suche durch eine bestimmte Indexstruktur für die Verwaltung der musikalischen Daten zu beschleunigen. In [8] werden z.B. die minimal umgebende Rechtecke (MUR) für die Verwaltung der akustischen Daten vorgeschlagen. Die Extrakt-Features sind dabei die ersten DFT(Diskrete Fourier Transformation)-Koeffizienten einer Audiodatei, bzw. -Sequenz. Die durch die Extrakt-Features gewonnenen Daten (die Mengen der multidimensionalen Punkten) werden in der raumorganisierten Datenstruktur (R*-Baum) gespeichert um die Suchzeit zu verringern. Für das eigentliche Matching wird hier eine mengenbasierte Methode verwendet, nämlich die Berech-

nung der euklidischen Distanz. Die in [8] durchgeführten Experimente zeigen, dass die Gruppierung der Daten in MURs und ein geschickter Zugang zu diesen Daten (*false alarm resolution*) die Suchzeit im Vergleich zu anderen existierenden Methoden gravierend verringern. Diese Methode unterstützt allerdings nur die Bereichsanfragen und außerdem können, wie die Autoren selbst behaupten, noch viele Parameter verbessert werden.

4.4 Suchergebnisse

Im Vorschlag von [1] für ein QBH-System werden die ähnlichen Musikstücke in Form einer Liste mit Information über dieser Musikstücke an den Benutzer geschickt. Der Benutzer kann ein Element aus dieser Liste auswählen und somit noch eine Anfrage (diesmal textuelle) an die Datenbank mit MPEG4-Dateien schicken. Nach der erfolgreichen Suche nach der gewünschten Datei fängt das Streaming an.

Für die QBE-Anwendungen sieht die Ergebnisliste aber anders aus: es werden nicht ähnliche Melodien sondern Musikstücke des gleichen Genres, die in der Datenbank vorhanden sind, an den Benutzer zurückgeschickt. Somit wird seine Spielliste konstruiert.

Die Beatboxing-Anwendungen orientieren sich mehr an DJs und die Suchergebnisse können sowohl ähnliche *drum loops* als auch ganz neue, durch *beatboxing* generierte Audiodateien sein. Die Beatboxer können mit verschiedenen Audiodateien manipulieren und mit einem Programm wie z.B. *bionic beatBoxing voice processor* [7] ein neues *drum loop* in einem gegebenen Takt erstellen.

5 MIR Systeme

Heutzutage existieren zahlreiche MIR-Systeme mit den unterschiedlichsten Aufgabenbereichen und den unterschiedlichsten Benutzerkreisen. Die Tabelle 1 aus [5] gibt einen guten Überblick über 17 MIR-Systeme. Hier beschränke ich mich nur auf die kurze Beschreibung einiger der bekanntesten Systeme.

5.1 CUIDADO Music Browser

Der Music-Browser wurde im Rahmen des CUIDADO-Projekts entwickelt. CUIDADO orientiert sich an die Entwicklung der inhaltsbasierten Audioanwendungen mit Hilfe von MPEG-7-Standard. Der Music Browser hat folgende Funktionalitäten[1]:

- *Tonähnlichkeit*: Bei der Angabe eines Tons werden die vorgeschlagenen Treffer ausgegeben.
- *Musikähnlichkeit*: Suche nach Musikdateien, die die gleichen „high-level“ Features wie z.B. Genre, Melodie oder Instrument haben.
- *Suche nach Musikähnlichkeiten*: Suche nach den ähnlichen Musikwerken mit Hilfe von einer vorher aufgenommenen Audiodatei
- *Mehrsprachiges Interface für die Stichwörtereingabe*: Verlinkung der individuellen Stichwörter in einer beliebigen Sprache zu einem gegebenen Musikstück. Diese Information wird dann für die Suche benutzt.

– *Andere Funktionen*: Es werden Benutzerprofile zur Speicherung der individuellen Einstellungen benutzt. Außerdem sind hier auch solche Tools wie *fast listening* und *constraint-based music selection* implementiert.

5.2 Meldex/Greenstone

In der Datenbank von Meldex werden die Volkslieder, basierend auf die Sammlung von „Essen“ und „Digital Tradition“, gespeichert. Meldex benutzt 2 Matching-Methoden: Stringdistanzen mit dynamischer Programmierung und Zustand-matching-Algorithmen (*state matching*) für approximiertes Matching [5]. Im Internet gibt es ein Interface, das die Suche durch die Angabe von bestimmten Noten bzw. Notensequenzen oder des Titels ermöglicht.

5.3 GUIDO/MIR

In diesem System werden die Query-Objekte als eine Kombination der melodischen (absolute Tonhöhe, Intervalle, etc.) und rhythmischen Information (absolute Dauer, relative Dauer, Trend) dargestellt. Die Markov-Ketten des ersten Grades werden für die Modellierung der melodischen und rhythmischen Konturs der monophonischen Musik verwendet. Für jedes Musikstück und jeden Anfragentyp gibt es eine Markov-Kette. Die Übergangsmatrizen sind in einer Baumstruktur organisiert (Blätter: Musikstücke; innere Knoten: Übergangsmatrizen) [5].

In [5] unterscheidet man zwischen 3 Benutzergruppen, die durch MIR-Systeme adressiert werden können: Industrie, Amateure und Profis (Interpreten, Lehrer, Musiker). Außerdem werden hier auch die Aufgabenklassen (Suche nach einem Werk oder Genre) kurz dargestellt und eine Analyse der MIR Systeme im Aufgaben- und Benutzerbereich durchgeführt. Die Abbildung 8 zeigt die Zuordnung der einzelnen MIR-Systeme zu den jeweiligen Suchaufgaben.

6 Fazit

Die inhaltsbasierte Musiksuche basiert auf den musikalischen Merkmalen (Features) wie z.B. Tonhöhe, Rhythmus, Klangfarbe, etc., die aus dem Musik-Content extrahiert werden können. Die Musikwerke stellen entweder rohe akustische (mp3, wav) oder symbolische (midi) Daten. Im Falle der akustischen Daten sind bestimmte Extrakt-Features wie z.B. DFT, Tonhistogramm oder *centroid* notwendig, um die Musik entspechend darstellen zu können. Die symbolischen Daten werden normalerweise als Zeichenketten dargestellt, die bestimmte musikalische Merkmale beschreiben.
In dieser Arbeit konzentrierte ich mich auf 4 Typen von Anfragen: die Suche über direkte Angaben von Musik-Features (mehr für Spezialisten geeignet), *query by humming*, *query by example* und *query by beatboxing*. Die Effizienz und Genauigkeit der Suche hängt von der Auswahl einer geeigneten Matching-Methode und einer Indexstruktur zur Speicherung der musikalischen Daten ab. Die am

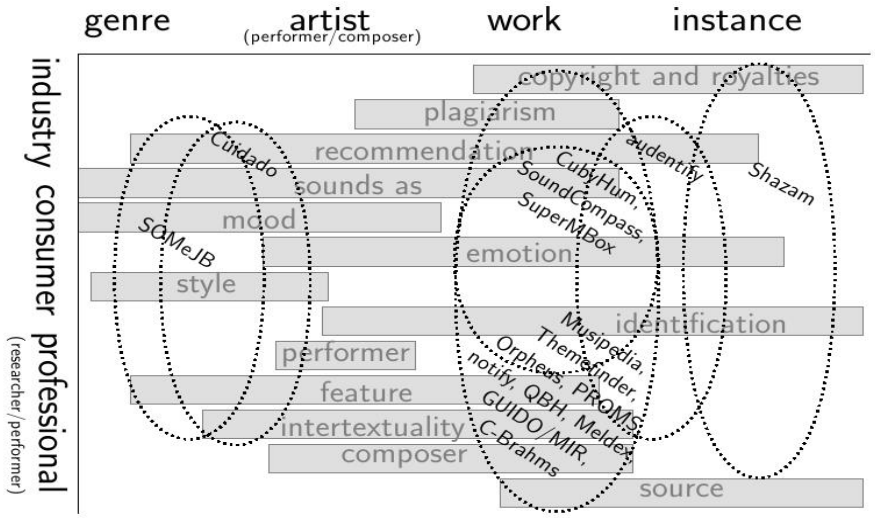

Abbildung 8. A mapping of MIR systems to retrieval tasks [5]

häufigsten verwendeten Indexstrukturen sind B(*)-Bäume, R(*)-Bäume und invertierte Listen. Die Matching-Algorithmen hängen stark mit Musikrepräsentation zusammen (z.B. Berechnung der Stringdistanz für Zeichenketten oder der euklidischen Distanz für mehrdimensionale Vektoren) und werden benutzt, um die Query-Objekte mit der Information aus der Datenbank zu matchen.
In den letzten Jahren wurde eine Vielzahl von MIR-Systemen entwickelt, die sich an unterschiedlichen Benutzergruppen (z.B. Amateure oder Profis) orientieren und bestimmte Funktionalitäten zur Verfügung stellen. Einige der bekanntesten MIR-Systeme sind CUIDADO-Music-Browser, Meldex/Greenstone und GUIDO/MIR. Die MIR-Anwendungen können sich auch in ihrem Aufgabenbereich (Suche nach Genre oder Musikstück) unterscheiden. Z.B. orientieren sich die QBE-Anwendungen nur auf die Suche nach Musikwerken im gleichen Genre.
Die meisten MIR-Anwendungen beschränken sich aber nur auf die Suche der monophonischen Musik und können manchmal die Anfrage falsch interpretieren. Der größte Teil der modernen Musik ist allerdings polyphonisch und wird im MP3- oder WAV-Format gespeichert. Ein weiteres Problem sind die Autorenrechte, weil die meisten Musikwerke nicht kostenlos sind und die Musik-Datenbanken sie generell nicht speichern dürfen. Es besteht somit einen starker Bedarf an einer weiteren Forschung im MIR-Bereich.

Literatur

1. B. S. Manjunath, P. Salembier, T. Sikora: Introduction to MPEG-7: Multimedia Content Description Interface. John Wiley & Sons Inc (2002) 283–297; 339–344

2. M. Welsh, N. Borisov et. al.: Querying large collections of music for similarity. Technical report UC Berkeley Computer Science Division (1999)

3. L. Lu, H You, HJ Zhang: A new approach to query by humming in music retrieval. In Proc. of the IEEE ICME01, Tokyo, Japan (2001)

4. H. Harb, L. Chen: A query by example music retrieval algorythm. In Proc. of the 4th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '03), Apr. (2003)

5. . Typke, F. Wiering, R. C. Veltkamp: A survey of music information retrieval systems. Technical report Universiteit Utrecht, the Netherlands (2004)

6. J. S. Downie: Music information retrieval (Chapter 7). In Annual Review of Information Science and Technology 37, ed. Blaise Cronin, 295-340. Medford, NJ: Information Today,(2003). Available from `http://music-ir.org/downie_mir_arist37.pdf`

7. A. Kapur, M. Benning, G. Tzanetakis: Query-by-beat-boxing: Music retrieval for the DJ. In the 5th international conference on Music Information Retrieval. (2004)

8. I. Karydis, A. Nanopoulos et. al.: Audio indexing for efficient music information retrieval. In Proc. of the 11th International Multimedia Modelling Conference (2005) 22–29

9. L. Gurulev, D. Nizaev Musiklehre in russischer Sprache: `http://www.7not.ru/theory`

10. Deutsches Forschungszentrum für Künstliche Intelligenz: `http://www.dfki.uni-kl.de/KM/content/e13/e109/index_ger.html`