Ähnlichkeitssuche in Musik-Datenbanken mit Hilfe von Visualisierungen

Dirk Habich
Informatik, Uni Halle, Deutschland
habich@informatik.uni-halle.de

Alexander Hinneburg
Informatik, Uni Halle, Deutschland
hinneburg@informatik.uni-halle.de

Abstract: Ähnlichkeitssuche in Datenbanken wurde bisher in vielen Bereichen erfolgreich angewendet. Jedoch gibt es vergleichsweise wenige Arbeiten, die sich mit Ähnlichkeitssuche in Musikdaten beschäftigen. Da derzeit immer mehr Musik über das Internet verfügbar wird, stößt dieser Bereich zunehmend bei vielen Anwendungsgruppen auf reges Interesse. Bisherige Suchverfahren lassen sich aber nur im beschränkten Maße an die vielfältigen Anwendungsszenarien anpassen. Meist läßt sich dies nur über die Auswahl einer abstrakten Metrik realisieren. Jedoch ist es ein ungelöstes Problem, wie der Benutzer dem Suchsystem mitteilen kann, welche Aspekte bei der Suche für eine Aufgabe relevant sind.

Diese Arbeit stellt einen Ansatz vor, der versucht, die sogenannte *semantische Lücke* zwischen Benutzer und System durch einer Kombination aus konventioneller Ähnlichkeitssuche mit interaktiven Visualisierungen zu überbrücken. Dafür wurde eine neue Feature-Extraktionsmethode für Musikdaten entwickelt, die gleichzeitig für eine Visualisierung geeignet ist. Die abgeleitete Visualisierung beschreibt statistische Eigenschaften des Musikstücks. Mit Hilfe der Visualisierung kann die Ähnlichkeit zweier Musikstücke auf herkömmliche Weise akustisch, aber auch visuell bewertet werden. Der visuelle Weg ist viel schneller als das akustische Durchhören verschiedener Resultate und ermöglicht so die Verwendung von *Relevance Feedback*, mittels dessen das System sich iterativ an die Vorstellungen des Benutzers anpassen kann. Wir haben unseren Ansatz mit einer bekannten Methode für Musik-Ähnlichkeitssuche verglichen und demonstrieren die Effektivität anhand von Anwendungsbeispielen.

1 Einführung

Ähnlichkeitssuche in großen Datenbanken ist ein wichtiges Forschungsgebiet mit vielfältigen Anwendungsgebieten. Hier ist unter anderem Ähnlichkeitssuche in Bild-Datenbanken, in Datenbanken mit geometrischen, geographischen und CAD-Objekten und in Dokument-Datenbanken zu nennen. Jedoch relative wenige Arbeiten wurden bisher über Musikähnlichkeitssuche veröffentlicht.

1.1 Musik Retrieval

Die meisten Arbeiten über Audiodaten kommen aus dem Bereich Spracherkennung. Es gibt einige Versuche, die dort entwickelten Verfahren auf Musikähnlichkeitssuche zu übertragen. Eine einfache Übertragung ist jedoch nur beschränkt möglich, da sich Sprache und Musik in ihren Eigenschaften (Frequenzen, Rhythmus, usw.) stark unterscheiden [Sch00].

Publizierte Arbeiten über Musikdaten umfassen unter anderem die Bereiche Analyse, Instrumenterkennung, Klassifikation und inhaltsbasierte Suche. Musikanalyse beschäftigt sich mit Rhythmus-, Melodie- und Harmonieerkennung [Rap01, DC01, BB01] ebenso wie mit der Frage, welche Eigenschaften für Musik-Ähnlichkeit wichtig sind [YK99]. Instrumenterkennung hat das Ziel in einem Musikstück die verwendeten Instrumente zu isolieren [HBABS00].

Musik-Klassifikation untersucht Methoden, die das automatische Einordnen von elektronisch gespeicherten Musikstücken nach verschiedenen Genres erlauben. Hier wurden verschiedene Trainingsansätze vorgeschlagen, bei denen die verschiedenen Klassen von den Verfahren selbst bestimmt werden [WBKW96]. Pampalk, Rauber und Merkl [PRM02] beschreiben eine Methode, die nach dem Lernschritt die Beziehungen zwischen den Klassen mittels einer abstrakten Karte visualisiert.

Ein wichtiger Aspekt der inhaltsbasierten Ähnlichkeitssuche auf Musikdaten ist das Problem der Anfragegenerierung. Die beiden vorgeschlagenen Ansätze sind erstens: die Anfrage wird vom Benutzer ins Mikrofon gesummt [KNS+00]; zweitens: es wird ein Anfragemusikstück in elektronischer Form dem Suchsystem präsentiert. Der erste Ansatz eignete sich besser für Musikstücke mit einfachem Melodieverlauf, wie zum Beispiel Volkslieder und ist oft auf Musikstücke im MIDI-Format (Musical Instrument Digital Interface) beschränkt, bei dem die verschiedenen Tonsequenzen separat in symbolischer Form abgespeichert werden. Der Großteil der elektronisch verfügbaren Musik liegt jedoch in digitalen Audio-Formaten wie WAV oder MP3 vor, die mittels Sampling aus analogen Aufnahmen gewonnen werden und nur die überlagerten Signale repräsentieren. Eine befriedigende Umwandlung von stark überlagerten Musiksignalen in einem Audio-Format in das MIDI-Format ist trotz sehr hohem Aufwand kaum möglich.

Um Musikstücke im Audioformat miteinander zu vergleichen, werden die Audiodaten in mehrdimensionale Vektoren transformiert, in denen Eigenschaften des Stücks zusammengefaßt sind. Für die Umwandlung können die Audio-Daten von zwei Seiten betrachtet werden. Die erste Sichtweise nutzt meßbare physikalischen Größen, wie Amplitude oder Frequenz. Die als Fourier-Transformation [Bri74] bekannte Zerlegung des Amplituden-Signals in Sinus-Wellen verschiedener Amplitude und Wellenlänge ist ein wichtiger Analyseschritt, der in ähnlicher Art und Weise auch im inneren Ohr des Menschen abläuft [Roe79]. Die andere Sichtweise bezieht sich auf Eigenschaften der menschlichen Wahrnehmung wie Lautstärke oder Harmonieempfinden. Ausgehend von psychoakustischen Untersuchungen wurden Modelle der menschlichen Wahrnehmung von Musik und Sprache entwickelt. Diese Modelle wurden auf den Computer übertragen, um das menschliche Hörempfinden zu simulieren. Die Ausgangswerte diese Modelle können ebenfalls zu einem Eigenschaftsvektor kombiniert werden, der ein Musikstück beschreibt. Eine detai-

lierte Beschreibung von Möglichkeiten der Audiodatenanalyse ist in [PFE96] zu finden.

1.2 Ähnlichkeitssuche in Datenbanken

Für Ähnlichkeitssuche in Datenbanken werden in der Regel nur die Eigenschaftsvektoren verwendet. Im Datenbankbereich wurden verschiedene Indexstrukturen beschrieben, welche Nächsten-Nachbarnsuche auf mehrdimensionalen Vektoren unterstützen. Ein Überblick über den aktuellen Forschungsstand ist in [BBK01] zu finden. Neben dem dort behandelten Problem der Effizienzsteigerung der Nächsten-Nachbarnsuche wurde in den letzten Jahren auch das Effektivitätsproblem behandelt. Hier wird untersucht, wie aussagekräftig die Ergebnisse einer Anfrage auf hochdimensionalen Daten sind. Die Fragestellung wurde in [BGRS99, AHK01] theoretisch untersucht und für verschiedene Datenverteilungen gezeigt, daß mit steigender Dimensionalität der Abstand von einem beliebigen Anfragepunkt zum nächsten Nachbarn schneller steigt als die Differenz zwischen den Abständen zum nächsten bzw. zum weitesten Nachbarn, d.h. daß der Kontrast zwischen dem nächsten und weitesten Nachbarn mit steigender Dimensionalität abnimmt. In [HAK00] wurde vorgeschlagen, statt im hochdimensionalen Datenraum in Unterräumen mit niedrigerer Dimensionalität nach aussagekräftigen nächsten Nachbarn zu suchen. Das Problem der Nächsten-Nachbarnsuche in einer Projektion des Datenraumes ist viel komplexer als eine einfache Nächsten-Nachbarnanfrage, da die Anzahl der potentiell interessanten Projektionen sehr groß ist. In [HAK00] wurde das Problem mittels eines genetischen Algorithmus gelöst, dessen Fitnessfunktion bewertet, wie stark die Datenpunkte um den Anfragepunkt geclustert sind. In [HK00] wurde eine allgemeine interaktive Variante diese Methode vorgeschlagen. Anstelle der automatischen Fitnessfunktion entscheidet der Benutzer interaktiv, welche der vom System vorgeschlagenen Projektionen eine aussagekräftige Nächsten-Nachbarnsuche erlaubt. Innerhalb weniger Iterationen paßt sich das System dem Ähnlichkeitsbegriff des Benutzers an, soweit die notwendige Information in den Eigenschaftsvektoren kodiert ist. Wichtig für diese Art der Ähnlichkeitssuche ist eine geeignete Visualisierung für die Objekte. In [HK00] wurden unter anderem Anwendungen für interaktive Bildähnlichkeitssuche untersucht, bei denen eine Visualisierung schon in natürlicher Art und Weise durch die jeweiligen Bilder gegeben ist.

1.3 Überblick

In diesem Artikel wird eine Methode zur interaktiven, visuellen Musikähnlichkeitssuche beschrieben. In Kapitel 2 werden Grundlagen zur Musikverarbeitung eingeführt. In Kapitel 3 wird ein einfaches Verfahren zur Berechnung von zeitunabhängigen Eigenschaftsvektoren aus beliebig langen Musikstücken entwickelt und darauf aufbauend eine neue Visualisierungstechnik vorgestellt, die es erlaubt, beliebig lange Musikstücke visuell miteinander zu vergleichen. Beide Komponenten werden in *MusicOpt*, einem datenbankgestütztem, interaktiven Retrieval-System, integriert. Abschließend wird in Kapitel 4 die neue interaktive Methode mit einem bekannten Musik-Retrieval-Verfahren verglichen.

2 Grundlagen der Musikverarbeitung

Zur einfachen Charakterisierung der Audiodaten kann die Fourier-Transformation herangezogen werden. Der Grundgedanke der Fourier-Transformation ist, daß beliebig komplexe Signale aus einer Summe von Sinusschwingungen unterschiedlicher Frequenz, Amplitude und Phase zusammengesetzt werden können. Die Aufgabe der Signalanalyse ist also die Zerlegung eines Signals in seine einzelnen harmonischen sinusförmigen Teilschwingungen.

Die Audiodaten liegen grundsätzlich als kontinuierliche Signale vor, die zur rechnergestützten Bearbeitung erst digitalisiert werden müssen. Für die Digitalisierung – auch Sampling genannt – wird das analoge Signal zu diskreten Zeitpunkten abgetastet und per Analog / Digital-Wandler in diskrete Werte umgewandelt. Auf der Zeitachse spricht man von Abtastrate(-frequenz), auf der Amplitudenachse von Quantisierung. Die Abtastfrequenz sollte nach dem Nyquist-Theorem bestimmt werden, d.h die Abstastfrequenz muß mindestens doppelt so groß sein, wie die größte im Signal vorkommende Frequenz. Dies stellt sicher, daß das analoge Signal aus der digitalen Form wieder zurückgewonnen werden kann. Da das menschliche Gehör Frequenzen von 20Hz bis 20kHz wahrnehmen kann, werden Audiodaten in der Regel mit einer Abtastfrequenz von 44,1kHz gesampelt.

Die Quantisierung hat auch großen Einfluß auf die Klangqualität des digitalen Signals. Der Amplitudenbereich wird in eine Anzahl von Intervallen unterteilt, so daß für jeden Abtastwert nur der Index des betroffenen Bereichs gespeichert werden muß.

Das Resultat der Digitalisierung ist ein zeitdiskretes Signal s, auf das die diskrete Fourier-Transformation angewendet werden kann. Falls $s(n)$ für $n \in \{0, 1, 2, \ldots, N-1\}$ definiert ist, so ist die N-Punkt diskrete Fourier-Transformation (N-Punkt DFT) definiert als:

$$S(f) = \sum_{n=0}^{N-1} s(n) e^{-j\frac{2\pi}{N} \cdot n \cdot f} \quad \texttt{für} \quad f = 0, 1, \ldots, N-1 \quad \texttt{und} \quad j = \sqrt{-1} \tag{1}$$

Die schnelle Fourier-Transformation (Fast-Fourier-Transformation / FFT) ist ein komplexer Algorithmus, der die Berechnungskomplexität von $O(n^2)$ auf $O(n \cdot \log n)$ verringert [Bri74].

Da für die weitere Verarbeitung eine Fourier-Transformation eines Musikstückes als Ganzes nur die durchschnittliche Frequenzverteilung liefert und die FFT des gesamten Musikstückes sehr viel Rechenzeit in Anspruch nimmt, läuft die Frequenzzerlegung der Audiodaten folgendermaßen ab. Das Audiosignal wird in gleich große, sich überlappende Fenster (typischerweise 16 ms) zerlegt. Auf jedem Fenster wird eine diskrete Fourier-Transformation (N-Punkt DFT) durchgeführt, um das Signal in seine Frequenzanteile zu zerlegen.

Da es durch ein diskretes Abschneiden des Signals an den Rändern des betrachteten Fensters zu Verfälschungen kommen kann, wird das Signal an den Rändern durch eine sogenannte Fensterfunktion ein- und ausgeblendet. Zur Fensterbildung wird in dieser Arbeit die Hamming-Funktion verwendet [Sch00].

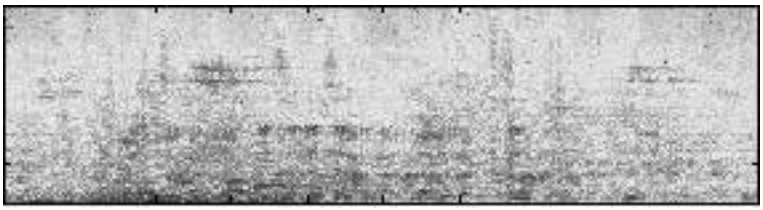


Abbildung 1: Die Abbildung zeigt eine 2-dimensionale zeitabhängige Frequenzintensitätsfunktion, bei der die Zeit auf der x-Achse und die Frequenz auf der y-Achse liegen. Der Intensitätswert ist als Grauwert dargestellt (hell entspricht geringer, dunkel hoher Intensität).

Durch die Fensterunterteilung entsteht als Resultat der Fourier-Transformationen eine zweidimensionale Frequenzintensitätsfunktion, die für jedes Zeitintervall (Fenster) ein Frequenzspektrum liefert. Abbildung 1 zeigt ein Beispiel für eine zeitabhängige Frequenzintensitätsfunktion. Die Grauwerte der Pixel zeigen, mit welcher Intensität die jeweiligen Frequenzen auftreten.

3 Interaktive visuelle Musikähnlichkeitssuche

In diesem Kapitel wird ein einfaches Verfahren zur Berechnung von zeitunabhängigen Eigenschaftsvektoren aus beliebig langen Musikstücken entwickelt und darauf aufbauend eine neue Visualisierungstechnik vorgestellt, die es erlaubt, beliebig lange Musikstücke visuell miteinander zu vergleichen. Beide Komponenten werden in *MusicOpt*, einem Datenbank gestütztem, interaktiven Retrieval-System integriert.

3.1 Eine effiziente Musikähnlichkeitsmetrik

Die FFT der überlappenden Zeitfenster liefert ein zeitliche Reihe von Frequenzspektren mit Intensitätswerten. Diese sehr genaue Darstellung ist für effiziente Ähnlichkeitsvergleiche aufgrund ihrer Größe ungeeignet. Für die Ähnlichkeitssuche ist eine kompakte, zeitunabhängige Darstellung sinnvoll. Um eine Beschreibung mit den erforderlichen Eigenschaften zu erhalten, wird ein Histogramm berechnet, in welchem gezählt wird, wie oft die jeweiligen Frequenz-Intensitätskombinationen in dem Musikstück auftreten. In Abbildung 2 wird ein Überblick über diese neue Transformation dargestellt. Die Transformation wird im folgenden etwas genauer beleuchtet.

Wie in Kapitel 2 beschrieben, stellt die Frequenzintensitätsfunktion eine sehr genaue, zeitabhängige Beschreibung des Audiosignals dar. Für jedes Zeitfenster wird eine zweidimensionale Ausgabe bestehend aus Frequenzbereich und Intensität erzeugt. Die Größen Frequenz und Intensität werden in Abbildung 1 auf y-Achse und Grauwert gelegt. Um ein kompakteres und zeitunabhängiges Histogramm zu erhalten, werden die zeitlich verschiedenen 2D-Ausgaben der FFT für die Zeitfenster erstens in grobere Bereiche unterteilt und

zweitens über die Zeit aggregiert.

Für die Vergröberung wird der Frequenzbereich von 20Hz-22kHz logarithmisch in k Frequenzbereiche eingeteilt. Die logarithmische Unterteilung wurde gewählt, da der Mensch die Frequenzen nahezu logarithmisch und nicht linear wahrnimmt [Roe79]. Die Intensitäten werden in Dezibel umgerechnet und der Intensitätsbereich wird ebenfalls in l Bereiche unterteilt. In den beschriebenen Anwendungen erstreckt sich der genutzte Intensitätsbereich von 0-60 Dezibel. Das Histogramm besteht also aus k Frequenzbereichen mit jeweils l Intensitätsbereichen. Typische Werte für die Anzahl der Frequenz- bzw. Intensitätsbereiche sind $k = 40$ und $l = 20$. Anschließend wird die Frequenzintensitätsfunktion nach der Zeit durchlaufen, wobei die Häufigkeiten der auftretenden Frequenzintensitätskombinationen gezählt werden.

Diese Frequenz-Histogramme können als Grundlage einer effizienten Vergleichsmetrik für Musikstücke dienen. Zwei Frequenz-Histogramme a, b werden mittels euklidischer Metrik wie folgt verglichen:

$$dist(a,b) = \sqrt{\sum_{i=1}^{k}\sum_{j=1}^{l}(a_{i,j} - b_{i,j})^2} \tag{2}$$

Diese Metrik kann für die Nächsten-Nachbarnsuche verwendet werden. Eine einfache Verallgemeinerung auf Nächsten-Nachbarnsuche in Projektionen [HAK00] ist zwar möglich, aber die genetische Suche bleibt wegen der hohen Dimensionalität ($d = 20 \cdot 40 = 800$) uneffektiv. Eine sinnvolle Einschränkung des Projektionssuchraumes wird durch die Forderung erreicht, daß alle Intensitätsbereiche derselben Frequenz immer vollständig in einer Projektion enthalten sein müssen.

3.2 Musik-Visualisierung

In diesem Abschnitt wird ein Verfahren beschrieben, wie aus den ermittelten Frequenz-Histogrammen der Musikstücke Visualisierungen generiert werden können. Diese werden für das interaktive genetische Verfahren benötigt, bei dem der Benutzer entscheidet, welche Projektion eine relevante Ähnlichkeitssuche erlaubt. Die in [HAK00] vorgeschlagene automatische Fitnessbewertung ist hier nicht sinnvoll anwendbar, da oft nur sehr wenige relevante Ergebnisse in einer Datenmenge vorkommen können. Somit ist die für die automatische Fitnessfunktion wichtige Annahme nicht mehr gültig, daß die relevanten Daten um den Anfragepunkt geclustert sind.

Um die zwei-dimensionalen Frequenz-Histogramme platzsparend darzustellen, wurden die Häufigkeitswerte auf Graustufen abgebildet. Jedoch eine einfache Abbildung reichte bei diesen Daten nicht aus. Da wenige Histogrammzellen sehr stark belegt sind, ergeben die Histogramme nach der Abbildung auf Grauwerte ein kaum strukturierte Visualisierung (Abb. 3(a)). Um den Kontrast zu erhöhen wurde in den weiteren Teilabbildungen (b-d) nicht die Häufigkeit h sondern $h^{1/2}$, $h^{1/4}$ bzw. $h^{1/8}$ dargestellt. Abbildung 3(c) weist den besten visuellen Kontrast auf und daher wurde $h^{1/4}$ für die Visualisierungen verwendet.

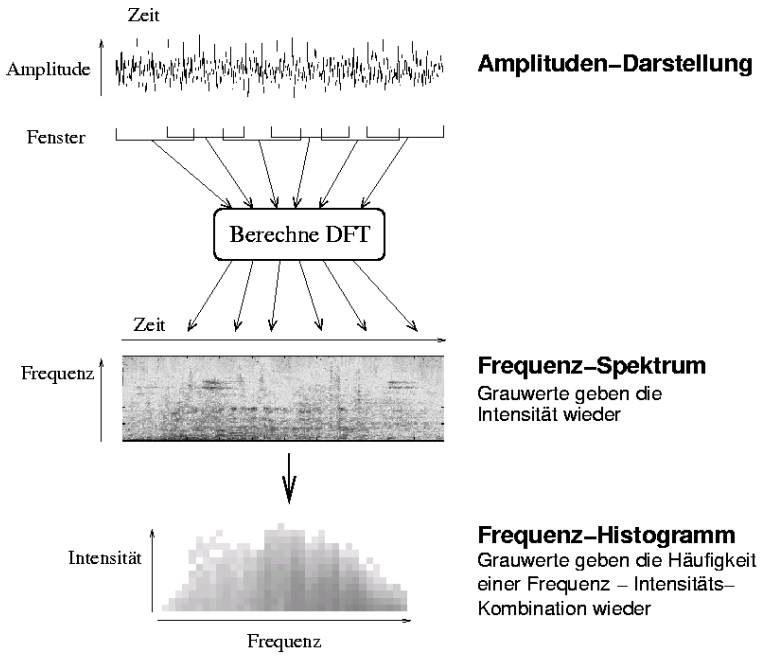


Abbildung 2: Die Abbildung zeigt eine Übersichtsskizze der Prozedur zur Transformation der Audiodaten in eine zeitunabhänige Vektordarstellung. Das Ergebnis der Umwandlung ist ein Histogramm, das die Häufigkeiten von Frequenzintensitätskombinationen beschreibt. Das Histogramm entsteht durch Zählung der auftretenden Kombinationen über die Dauer des Musikstückes (oder eines Teils des Stückes).

Die Kontrastqualität wurde noch auf weiteren Beispielen geprüft und immer ergab $h^{1/4}$ den besten visuellen Eindruck. Die Suche nach einer befriedigenden Erklärung für diesen Sachverhalt ist Gegenstand weiterer Forschungen.

3.3 Das MusicOpt-System

In dem Musik-Retrieval-System *MusicOpt* wurde das angepaßte interaktive Verfahren zur Ähnlichkeitssuche in Unterräumen mit den Musikvisualisierungen kombiniert. Mittels interaktiver Auswahl der relevanten Projektionen kann der Benutzer dem System sein Ähnlichkeitsmaß mitteilen, ohne den Transformationsprozess der Musikdaten zu den Frequenz-Histogrammen im Detail verstehen zu müssen. Dieser interaktive Prozess führt in den meisten Fällen zu besseren Suchergebnissen.

Abbildung 5 zeigt ein Bildschirmfoto des *MusicOpt*-Systems. Ein Zeile der Bildmatrix zeigt jeweils das Ergebnis einer Nächsten-Nachbarnsuche in einem bestimmen Unterraum. In der ersten farbig hinterlegten Zeile ist das Ergebnis der Nächsten-Nachbarnsuche, bei der alle Dimensionen berücksichtigt werden, dargestellt. Bei den übrigen sind die Fre-

(a) (b) (c) (d)

Abbildung 3: Die Abbildungen (a-d) zeigen das selbe Frequenz-Histogramm mit unterschiedlichen Häufigkeitstransformationen. Für Abbildung (a) wurde die Häufigkeit linear auf die Grauwerte abgebildet, bei den Abbildungen (b-d) wurde $h^{1/2}$, $h^{1/4}$ bzw. $h^{1/8}$ verwendet. Ein visueller Vergleich ergibt, daß für $h^{1/4}$ die Grauwertskala am besten ausgeschöpft wurde.

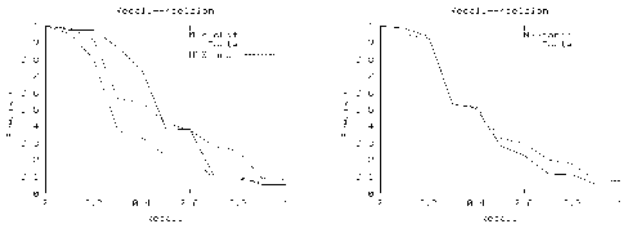
quenzbereiche, die den genutzten Unterraum definieren, mit einem schwarzen Punkt markiert. Um die Relevanz der Ergebnisse zu beurteilen, kann der untrainierte Benutzer zuerst einige Stücke akustisch durchhören, um so Musik und Visualisierung miteinander in Verbindung zu bringen. Später kann der Benutzer die für ihn relevanten Ergebnisse auf visueller Basis auswählen. Das System erzeugt mittels einfacher Cross-Over Operatoren neue Varianten der gewählten Projektionen. Falls das Ähnlichkeitsverständnis des Benutzers mittels einer Projektion der zugrundeliegenden Frequenz-Histogramme ausdrückbar ist, konvergiert das System in der Regel nach drei bis vier Iterationen.

4 Evaluierung

In diesem Abschnitt wird der interaktive Ansatz zur Musikähnlichkeitssuche mit dem Verfahren von Foote [Foo97] verglichen. Zum diesem Ansatz war das Suchsystem und die Datenbasis unter `http://www.fxpal.com/people/foote/musicr/doc0.html` verfügbar. Die Datenbasis besteht aus 255 Musikstücken, die auf jeweils 7 Sekunden gekürzt wurden. Die Musikstücke stammen von 40 verschiedenen Bands und Interpreten. Die Ergebnisse des Ansatzes von Foote sind vorberechnet in den Webseiten gespeichert. Eine wichtige Frage bei der Evaluierung ist die Definition der relevanten Ergebnisse zu einer Anfrage. Für diese Arbeit wurde die oft genutzte Annahme verwendet, daß zu einem Anfragestück die Stücke der gleichen Musikgruppe relevant sind. Für die Auswertung wurden die üblichen Maße wie Precision und Recall genutzt, ebenso wie die Standard-Auswertungssoftware der Information-Retrieval Konferenz TREC `http://trec.nist.gov/`. Precision und Recall sind wie folgt definiert:

$$Precision = \frac{Number\_Retrieved\_Relevant}{Number\_Total\_Retieved}, Recall = \frac{Number\_Retrieved\_Relevant}{Number\_Possible\_Relevant} \quad (3)$$

Ziel eines Suchsystem ist es, daß beide Werte möglichst groß sind. Die Abbildung 4(a) zeigt die durchschnittliche Recall-Precision-Kurve über 10 Anfragen. Es ist zu sehen, daß mit dem interaktiven *MusicOpt*-Ansatz mit höherer Genauigkeit deutlich mehr relevante Treffer erzielt werden können als mit dem automatischen Ansatz von Foote. Besonders wichtig ist der folgende Bereich: Recall[0.2:0.5], Precision[0.6:1], weil mit einer Einstellung aus diesem Bereich die Ergebnisse dem Benutzer zuerst präsentiert werden. Da entgegen der Annahme, daß alle Musikstücke einer Gruppe relevant sind, bei den meisten Gruppen starke Inhomogenitäten in ihren Stücken bestehen, spiegelt ein Recall größer als 0.5 oft nicht die gewünschten Ergebnisse wieder. In Abbildung 4(b) verglichen wir die



(a) (b)

Abbildung 4: Teil (a) Vergleich Foote, Nearest-Neighbour-Search und MusicOpt für 10 Anfragen; Teil (b) Vergleich Foote, Nearest-Neighbour-Search für 40 Anfragen, (Werte rechts oben sind besser)

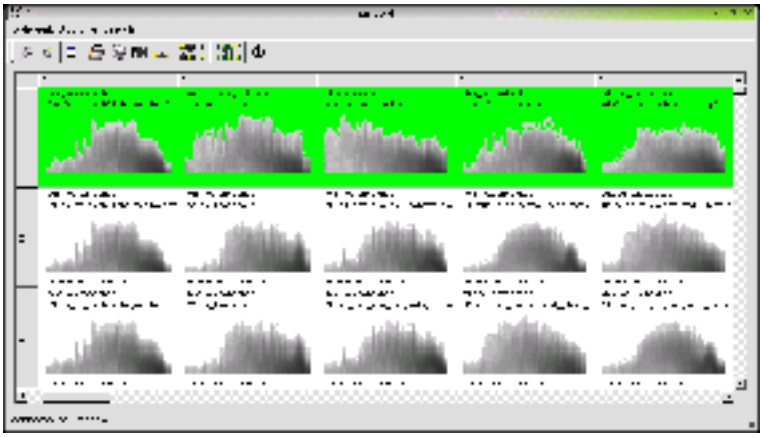


Abbildung 5: Die Abbildung zeigt ein Bildschirmfoto des *MusicOpt*-Systems.

Standard-Nächste-Nachbarnsuche mit den Ergebnissen von Foote auf 40 Anfragen. Beide Verfahren arbeiten etwa gleich gut, so daß der Effektivitätsgewinn in Abbildung 4(a) allein der interaktiven *MusicOpt*-Methode zuzurechnen ist.

In Abbildung 5 ist das Potential der Effektivitätssteigerung an einem Beispiel gezeigt. Als Anfragepunkt würde ein Lied eines Adventschors gewählt. Die Standard-Nächste-Nachbarnsuche liefert nur ein weiteres Lied dieses Chors zurück. Mit Hilfe der *MusicOpt*-Methode läßt sich diese Anzahl auf drei erhöhen, wobei nicht-relevante Stücke nicht mehr im Ergebnis auftauchen. Die verbesserte Qualität läßt sich auch anhand der Visualisierungen verifizieren.

In weiteren Forschungen sollen die Möglichkeiten der Rhythmuserkennung und abstrakte, allegorische Visualisierungen zur Verbesserung der Musikähnlichkeitssuche untersucht werden.

Literaturverzeichnis

[AHK01] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *ICDT 2001 (International*

Conference on Database Theory), London, UK, 2001. Springer, 2001.

[BB01] Jerome Barthélemy and Alain Bonardi. Figured bass and tonality recognition. In *Proceedings of the Second Annual International Symposium on Music Information Retrieval: ISMIR 2001*, pages 129–136. Indiana University, 2001.

[BBK01] Christian Böhm, Stefan Berchtold, and Daniel A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373, 2001.

[BGRS99] Kevin S. Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When Is "Nearest Neighbor" Meaningful? In *Database Theory - ICDT '99, 7th International Conference, Jerusalem, Israel, January 10-12, 1999, Proceedings*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235. Springer, 1999.

[Bri74] E. O. Brigham. *The Fast Fourier Transform*. Prentice-Hall Inc., 1974.

[DC01] Adriane Swalm Durey and Mark A. Clements. Melody spotting using hidden Markov models. In *Proceedings of the Second Annual International Symposium on Music Information Retrieval: ISMIR 2001*, pages 109–117. Indiana University, 2001.

[Foo97] J. Foote. Content-based retrieval of music and audio. In *Multimedia Storage and Archiving Systems II, Proceedings of SPIE*, pages 138–147, 1997.

[HAK00] Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What Is the Nearest Neighbor in High Dimensional Spaces? In *VLDB'2000, Proceedings of 26th International Conference on Very Large Data Bases, Cairo, Egypt*. Morgan Kaufmann, 2000.

[HBABS00] Perfecto Herrera-Boyer, Xavier Amatriain, Eloi Batlle, and Xavier Serra. Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In *Proceedings of the Second Annual International Symposium on Music Information Retrieval: ISMIR 2000*. University of Massachusetts at Amherst, 2000.

[HK00] Alexander Hinneburg and Daniel A. Keim. Using Visual Interaction to solve Complex Optimization Problems. In *Dagstuhl seminar on Scientific Visualization*. Kluwer, May 2000.

[KNS+00] Naoko Kosugi, Yuichi Nishihara, Tetsuo Sakata, Masashi Yamamuro, and Kazuhiko Kushima. A practical query-by-humming system for a large music database. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 333–342. ACM Press, 2000.

[PFE96] Silvia Pfeiffer, Stephan Fischer, and Wolfgang Effelsberg. Automatic audio content analysis. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 21–30. ACM Press, 1996.

[PRM02] E. Pampalk, A. Rauber, and D. Merkl. Content-based Organization and Visualization of Music Archives. In *Proceedings of ACM Multimedia 2002*, France, 2002. ACM.

[Rap01] Christopher Raphael. Automated rhythm transcription. In *Proceedings of the Second Annual International Symposium on Music Information Retrieval: ISMIR 2001*, pages 99–107. Indiana University, 2001.

[Roe79] J. G. Roederer. *Introduction to the Physics and Psychophysics of Music*. Springer, New York, 1979.

[Sch00] Carsten Schäfer. Entwicklung einer Audio-Retrieval-Komponente für ein multimediales IR-System. Master's thesis, University Dortmund, 2000.

[WBKW96] Erling Wold, Thom Blum, Douglas Keislar, and James Wheaton. Content-Based Classification, Search, and Retrieval of Audio. *IEEE MultiMedia*, 3(3):27–36, 1996.

[YK99] Chi Lap Yip and Ben Kao. A Study of Musical Features for Melody Databases. In *Proceedings l0th International Conference on Database and Expert Systems Applications*, pages 724–733, 1999.