

# Musipedia, eine offene Musiksuchmaschine

Rainer Typke\*

Auf Musipedia.org arbeiten seit 1997 einige tausend Benutzer an einer Sammlung von Melodien und musikalischen Themen. Der Hauptzweck dieser Sammlung ist das Identifizieren von Melodien, d. h. das Herausfinden von Merkmalen wie Titel und Komponist, anhand rein musikalischer Merkmale wie Melodie und/oder Rhythmus. Wenn man einen Ohrwurm im Kopf hat, aber nicht weiß, worum es sich genau handelt, kann man die Melodie Musipedia vorpfeifen, auf einer virtuellen Klaviatur spielen oder den Rhythmus klopfen, und die Suchmaschine wird in der Melodiesammlung die ähnlichsten Einträge finden und anzeigen.

Musipedia basiert auf Offenheit, nicht nur weil die Melodiesammlung in Wikipedia-Manier von jedermann bearbeitet werden kann, sondern auch, weil sehr viel Open Source-Software in die Implementierung eingeflossen ist. Nicht zuletzt steht Musipedia über eine SOAP-Schnittstelle anderen Benutzern offen, die beispielsweise in ihre eigene Website eine inhaltsbasierte Musiksuche einbauen möchten.

Dieser Vortrag gibt einen Überblick über die Funktionsprinzipien von Musipedia und zeigt, wie das Vorhandensein von Open Source-Software (Linux, MySQL, Lilypond, Red5, CGAL, diverse Sound-Bibliotheken) und Daten mit Creative Commons-Lizenz (Freesound.org) neue Entwicklungen wie diese inhaltsbasierte Musiksuchmaschine möglich macht.

## 1 Einleitung

Denys Parsons zitiert in seinem “Directory of Tunes and Musical Themes” [Par75] Bernard Levins Beschreibung eines alten Problems:

“...what if we cannot read, or write, a single note of music? What if we think that A flat major is an army officer who has had the misfortune to be run over by a tank? Are we to long in vain for the ability to discover what we are trilling, or the trombone-playing busker in the street is tootling, or the maiden at the piano glimpsed behind the lace curtains is tinkling?”

---

\*Der Autor dank dem FWF für die Finanzierung von Teilen der in diesem Artikel vorgestellten Ergebnisse (Indexierung, Audioanalyse) im Rahmen eines Lise-Meitner-Projekts (Projektnr. M1027-N15).

Mit anderen Worten: was kann man tun, wenn man einen Ohrwurm im Kopf hat, für den man gerne den Titel und Komponisten wüsste? Die wenigsten Menschen haben ein absolutes Gehör, mit dessen Hilfe sie Metadaten wie die Tonart bestimmen könnten; auch das Aufschreiben der Noten ist nicht jedem ohne weiteres möglich. Eine denkbare Lösung ist Parsons' Buch [Par75], in dem alphabetisch sortiert für tausende von Melodien die melodische Kontur verzeichnet ist. Dabei werden Zeichenketten aus einem 3-Buchstaben-Alphabet verwendet, wobei "R" für eine Tonwiederholung, "D" für einen Schritt nach unten und "U" für einen Schritt nach oben steht. Jedes Paar aufeinanderfolgender Noten wird mit einem Buchstaben codiert. Die Verwendung dieses Buches ist allerdings nicht ganz unproblematisch, weil jeder noch so kleine Fehler beim Bestimmen der melodischen Kontur das Auffinden im Verzeichnis scheitern lassen kann, besonders wenn er nah am Beginn der Zeichenkette auftritt. Außerdem benötigt man Parsons' Buch, das schon seit Jahrzehnten vergriffen ist, und nicht zuletzt handelt es sich bei dem Buch um eine statische Sammlung. Andere Werke wie Barlow und Morgenstern's "Dictionary of Musical Themes" [BM49] haben ähnliche Nachteile.

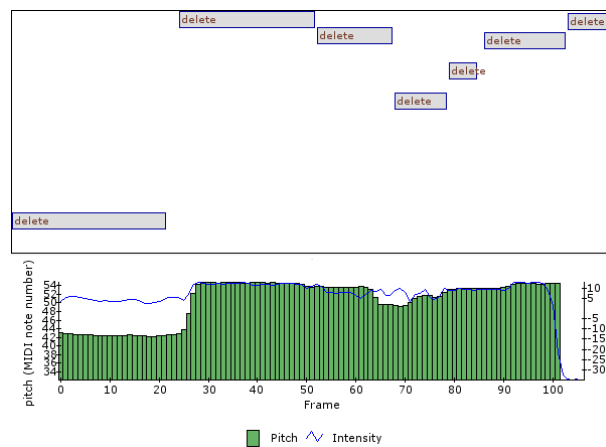


Abb. 1: Eine gesungene Suchanfrage. Im oberen Bereich werden die erkannten Noten dargestellt, darunter sieht man die Tonhöhenkurve (Balken) und die Lautstärke-Daten (Kurve), aus denen diese Noten errechnet wurden.

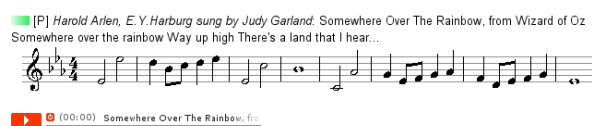


Abb. 2: Diese Melodie ist unter den ca. 30.000 Melodien in der Musipedia-Sammlung der Suchanfrage aus Abbildung 1 am ähnlichsten.

Ein weiterer Ansatz für die Identifikation von Musik, der in den letzten Jahren Verbreitung gefunden hat, ist Audio Fingerprinting [Wan03]. Die britische Firma Shazam stellt einen

Dienst für die Identifikation von Tonaufnahmen über Mobiltelefone zur Verfügung. Ein wenige Sekunden langer Ausschnitt reicht aus, um aus Millionen von Aufnahmen die richtige zu finden. Allerdings scheitert dieses Verfahren, wenn man eine Melodie identifizieren will, von der man gerade keine kommerzielle Aufnahme zur Hand hat, die sich in Shazams Datenbank befindet.

Midomi ([www.midomi.com](http://www.midomi.com)) ist ein kommerzieller Anbieter eines Musikidentifizier-Dienstes im Internet. Midomi versucht, gesungene Melodien zu identifizieren, indem sie direkt mit anderen gesungenen Melodien verglichen werden. Dadurch wird das Problem umgangen, zunächst Noten zu erkennen, also eine symbolische Repräsentation zu finden. Das ist ein offensichtlicher Vorteil, da die Notenerkennung aus gesungenen Melodien nicht immer zuverlässig funktioniert, aber gleichzeitig auch eine große Einschränkung, weil so keine der großen existierenden Sammlungen von symbolisch repräsentierter Musik (MIDI-Dateien, Partituren) durchsucht werden können. Nur für Melodien, die zuvor von einem anderen Midomi-Benutzer gesungen wurden, besteht die Hoffnung, dass sie identifiziert werden können.

Sherlock Hums ([www.sherlockhums.com](http://www.sherlockhums.com)) ist ein junges Unternehmen, das seit einigen Jahren ankündigt, bald einen Dienst für das Identifizieren gesungener Melodien anzubieten. Der Dienst steht zwar noch nicht bereit, aber die Ankündigung ist gut gemacht und unterhaltsam und unterstreicht zumindest, dass das Problem der Ohrwurmidentifizierung weit verbreitet und relevant ist. Man kann das Phänomen, dass ein Unternehmen jahrelang einen solchen Dienst ankündigt und zumindest in diese Ankündigung, aber wahrscheinlich auch in die Implementierung einen nicht ganz unwesentlichen Aufwand steckt, sicherlich dahingehend interpretieren, dass ein gut funktionierender Internetdienst fürs Identifizieren gepfiffener oder gesungener Melodien ein anspruchsvolles Ziel ist.

Musipedia löst einige der Probleme, die bei der Verwendung von Parsons' Buch bestehen:

- **Dynamische Sammlung:** Musipedia bietet eine Sammlung von Melodien, die von jedermann editiert und erweitert werden kann. Jeder Eintrag enthält neben Metadaten in Textform auch eine Melodie in Notenschrift und als MIDI-Datei. Der Notensatz erfolgt mit Lilypond [NN03], die Eingabe kann aber ohne Lilypond-Kenntnisse erfolgen.
- **Fehlertolerante Suche:** Musipedia enthält eine Suchmaschine, die die Sammlung aufgrund rein musikalischer Merkmale wie Melodie und Rhythmus durchsuchen kann. Suchanfragen können gesungen oder gepfiffen werden oder über die Computertastatur oder eine Klaviatur auf dem Bildschirm eingegeben werden. Bei der Ähnlichkeitssuche werden kleinere Ungenauigkeiten oder Fehler toleriert, sie führen nicht sofort dazu, dass eine Suche fehlschlägt.

Im nächsten Abschnitt wird Musipedia näher beschrieben. In Abschnitt 3 wird gezeigt, wie die Probleme gelöst wurden, für die es keine fertige quelloffene Software gab, insbesondere die Notenerkennung in gesungenen oder gepfiffenen Melodien und die inhaltsbasierte Musiksuche. Schließlich werden in Abschnitt 4 die quelloffenen Softwarepakete aufgelistet, die dazu verwendet wurden, Teilprobleme wie z. B. die Tonhöhenenerkennung zu lösen.

## 2 Die Musipedia-Musiksuchmaschine: Leistungsmerkmale

Musipedias wichtigste Leistungsmerkmale sind die Editierbarkeit der Sammlung durch jedermann und die inhaltsbasierte Musiksuche.

Musipedia ist in mehrerer Hinsicht offen: die Daten in der wikiartigen Melodiesammlung stehen unter recht großzügigen Lizenzen. Bei der Eingabe einer neuen Melodie wählt man zwischen “Public Domain” und diversen Creative Commons-Lizenzen aus. Außerdem gibt es eine SOAP-Schnittstelle, mit der die Musipedia-Suchmaschine als Webservice in andere Programme oder Websites eingebaut werden kann.

### 2.1 Editierbarkeit der Sammlung

Einerseits versucht Musipedia, so viele Möglichkeiten wie möglich auszunutzen, die das Notensatzsystem Lilypond bietet. Andererseits sollte es auch ohne Vorkenntnisse jedermann möglich sein, die Sammlung zu bearbeiten. Daher gibt es beim Eingeben oder Bearbeiten von Lilypond-Quelltext die Wahl zwischen der Möglichkeit, Noten direkt in Lilypond-Syntax einzugeben, oder aber ein kleines Zusatzfenster zu nutzen, in dem eine Note durch das Klicken entsprechender Symbole editiert werden kann.

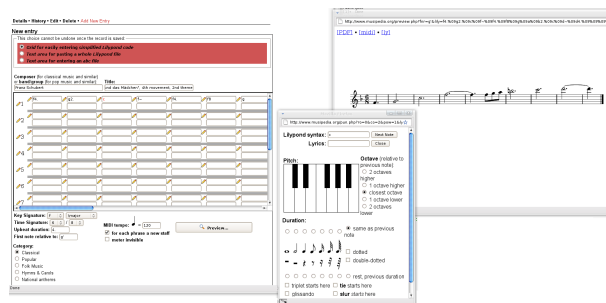


Abb. 3: Im Hauptfenster (links) gibt es ein Gitter mit zwei Feldern für jede Note – eines für die Note selbst und darunter eines für eventuellen Text. In einem kleinen Zusatzfenster kann mit einigen Mausklicks eine Note bearbeitet oder neu eingegeben werden (Mitte). Der aktuelle Zustand kann in einer Vorschau angezeigt werden (rechts).

Eine neue Melodie kann wie folgt der Sammlung hinzugefügt werden:

1. Sicherstellen, dass die Melodie wirklich noch nicht in der Sammlung ist (zur Suchmaschine siehe Abschnitt 2.2). Unten im Suchergebnis gibt es einen Link zum Hinzufügen einer fehlenden Melodie.
2. Es erscheint ein leeres Gitter (siehe Abbildung 3). Wer Lilypond schon beherrscht, kann einfach in jedes obere Kästchen eines Kästchenpaars eine Note eintragen, beispielsweise “f4.” für ein F, punktierte Viertelnote, selbe Oktave wie die vorige Note. Wer diese Syntax nicht beherrscht, klickt stattdessen auf das kleine Stiftsymbol (siehe Abb. 4) und wählt die Option, das betreffende Kästchen zu editieren. Es erscheint dann ein

- Zusatzfenster (Mitte), in dem man durch einen Klick auf die Klaviatur die Tonhöhe bestimmt und ggf. mit weiteren Klicks Eigenschaften wie Notendauer, Oktave, Bindebogen etc. eingeben kann, ohne irgendetwas über Lilypond zu wissen. Es erscheint dann automatisch der entsprechende Lilypond-Code im Kästchen.
3. Ebenfalls wichtig ist die Eingabe der Tonart, der Taktart etc. – dies kann unterhalb des Gitters mit einigen Drop-Down-Feldern erledigt werden.
  4. Um ein Zwischenergebnis zu sehen, wie beispielsweise rechts in Abbildung 3, klickt man einfach auf den Preview-Knopf.
  5. Wenn die Melodie korrekt aussieht, wählt man unten auf der Seite noch eine Lizenz aus – es steht neben “Public Domain” auch eine Auswahl von Creative Commons-Lizenzen und die Mutopia-BSD-Lizenz zur Auswahl – und klickt schließlich auf den Speichern-Knopf.
  6. Nun steht die neue Melodie in der Datenbank und kann sofort von der Suchmaschine gefunden werden.

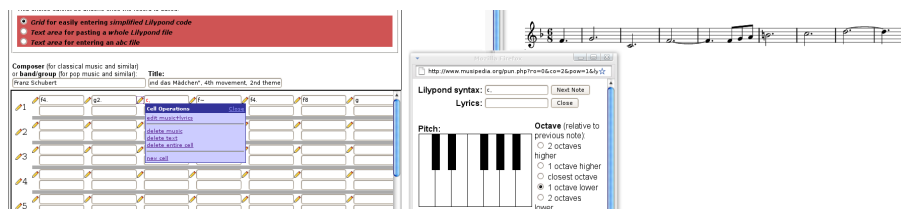


Abb. 4: Der Oktaven-Fehler von Abb. 3 ist schnell korrigiert: Stift-Symbol für die Note anklicken und im Popup-Fenster (Mitte) die richtige Oktave auswählen.

## 2.2 Suchmaschine

### 2.2.1 Suchen anhand gesungener oder gepfiffener Suchanfragen

Mit einem kleinen Flash-Applet kann man ein paar gesungene oder gepfiffene Noten aufnehmen und die Aufnahme an den Musipedia-Server schicken. Dort werden die darin enthaltenen Noten erkannt; diese Noten werden im Browser des Benutzers dargestellt, wo sie abgespielt und ggf. korrigiert werden können. Dann kann der Benutzer die Musipedia-Sammlung nach den Einträgen durchsuchen, die seiner Suchanfrage am ähnlichsten sind. Dabei werden sowohl der Rhythmus als auch die Intervalle berücksichtigt. Zur Illustration siehe Abb. 1, 2, 7 und 9.

#### 2.2.2 Suchen mittels Klaviatur

Wer kein Mikrofon zur Hand hat oder meint, nicht singen oder pfeifen zu können, kann eine von zwei Klaviaturen für die Eingabe einer Melodie verwenden. Es steht ein Flash-Klavier

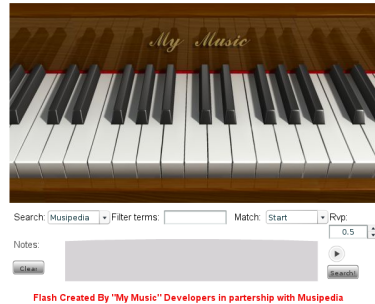


Abb. 5: Durch Klicken auf die Klaviertasten kann eine Suchanfrage eingegeben werden.

zur Verfügung, das u. a. von Joshua Koo entwickelt wurde (siehe Abb. 5), und außerdem ein JavaScript-Klavier (siehe Abb. 6). Beim Flash-Klavier muss man im korrekten Rhythmus auf die Tasten klicken, die Bedienung entspricht also weitgehend der eines wirklichen Klaviers. In der JavaScript-Variante dagegen kommt es nicht darauf an, die Tasten zum richtigen Zeitpunkt zu treffen. Notendauern werden stattdessen dadurch bestimmt, dass man das gewünschte Symbol (von der Achtelnote bis zur halben Note) anklickt.

Der Suchalgorithmus ist derselbe wie für gesungene Suchanfragen, es werden also sowohl die Intervalle als auch der Rhythmus berücksichtigt.

### 2.2.3 Suchen anhand der groben melodischen Kontur

Auch wenn man stark von den Melodien abstrahiert, indem man den Rhythmus vollkommen ignoriert und die Intervallfolge auf eine Folge von “Hoch”, “Herunter” und “Wiederholung” reduziert, sind erstaunlich wenig Noten nötig, um eine Melodie in einer großen Sammlung zu finden. Musipedia erlaubt es, diese grobe Kontur als Zeichenfolge einzugeben und dann die ähnlichsten Einträge in der Datenbank zu finden.

Die Ähnlichkeitssuche mildert das Problem ab, das bei der Verwendung des Buches von D. Parsons [Par75] entsteht, sobald sich auch nur ein kleiner Fehler in die Zeichenkette einschleicht.

### 2.2.4 Rhythmus-Suche

Die Eingabe eines Rhythmus durch rhythmisches Antippen einer Taste ist wahrscheinlich die einfachste Art, eine Suchanfrage einzugeben. Allerdings ist der Rhythmus meist auch am wenigsten charakteristisch, so dass besonders bei oft auftretenden Rhythmen die Gefahr besteht, viele Stücke zu finden, die man nicht wirklich sucht.

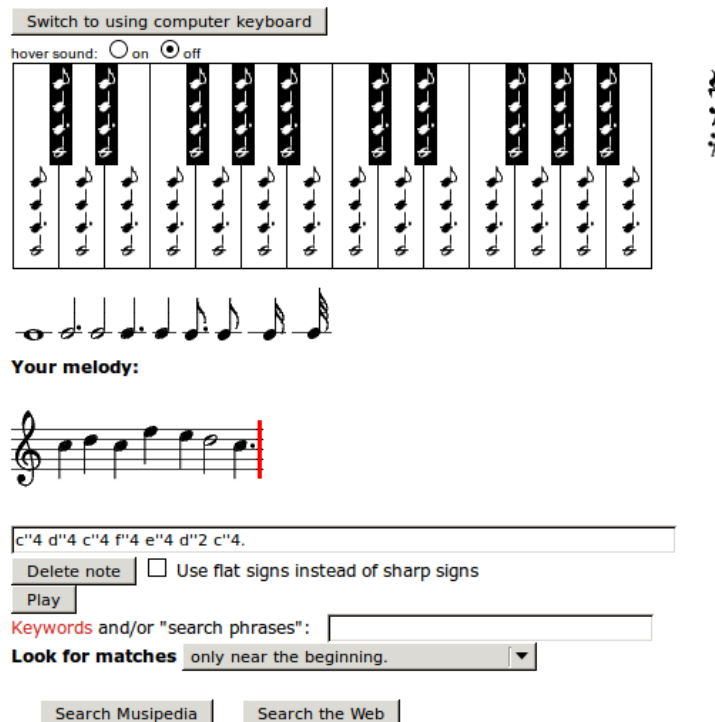


Abb. 6: Bei dieser Klaviatur kommt es nicht auf den genauen Klickzeitpunkt an. Stattdessen kontrolliert man die Notendauern über die Notensymbole.

## 3 Funktionsweise

### 3.1 Eingabe einer Suchanfrage

#### 3.1.1 Singen oder Pfeifen: Melodie

Am interessantesten und, wenn die Sache nicht an Hintergrundgeräuschen oder mangelhaften Sangeskünsten scheitert, auch am einfachsten zu benutzen, ist die Sing- oder Pfeif-Benutzerschnittstelle. Man kann eine Melodie über das Computermikrofon und ein kleines Flash-Applet, das in Abbildung 8 oben sichtbar ist, dem Musipedia-Server vorsingen oder -pfeifen. Dieser extrahiert dann zunächst mit Hilfe von Praat [BW08] eine Tonhöhenkurve und eine Lautstärkekurve. Diese Daten sind beispielhaft in Abbildung 7 gezeigt – hier handelt es sich um eine gesungene Suchanfrage nach Beethovens fünfter Sinfonie, erster Satz, erstes Thema.

Wenn diese beiden Kurven bekannt sind, muss noch festgestellt werden, wo die Grenzen der Noten liegen, die der Benutzer gemeint hat. Bei der Notenerkennung werden drei Kriterien angewandt:

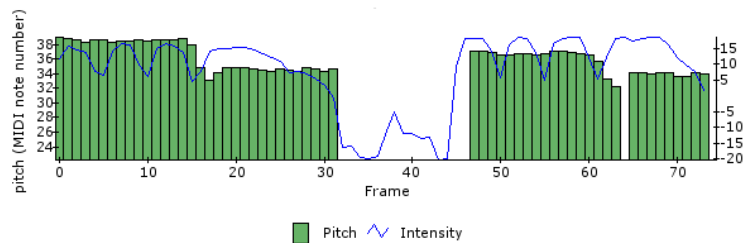


Abb. 7: Erkannte Tonhöhen (Balken) und Intensitäten (Kurve) in einer gesungenen Suchanfrage. Dieses Zwischenergebnis wird mittels Praat [BW08] aus der Tonaufnahme extrahiert.

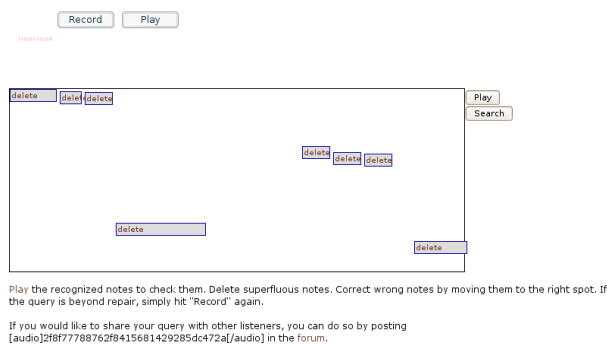


Abb. 8: Aus den in Abbildung 7 gezeigten Daten extrahiert Musipedia diese acht Noten.

**1. Notentrennkriterium: keine definierte Tonhöhe.** Offensichtlich kann man ohne weiteres annehmen, dass eine Note auf jeden Fall dann aufhört, wenn kein Klang mit definierter Tonhöhe mehr vorhanden ist. Das ist beispielsweise zwischen den Frames 32 und 47 in Abbildung 7 der Fall.

**2. Notentrennkriterium: Lautstärkeschwankungen.** In diesem Beispiel sieht man aber auch schön, dass die Tonhöhenkurve oft über viele Noten hinweg nicht unterbrochen wird – es kann sogar die Tonhöhe dabei gleich bleiben. Es ist dann nötig, anhand der Lautstärkekurve zu entscheiden, wo weitere Notengrenzen eingezeichnet werden müssen. Hier ist beispielsweise in den ersten 14 Frames eine Dreiergruppe von etwa gleich hohen Noten enthalten, und in den Frames 47 bis 60 eine weitere Dreiergruppe.

**3. Notentrennkriterium: Tonhöheschwankungen.** Normalerweise kann man an einer Änderung der Tonhöhe, gefolgt von einer gewissen Zeit mit wiederum stabiler Tonhöhe, ebenfalls erkennen, dass eine neue Note beginnt. Dieses Kriterium ist beispielsweise für das in Abbildung 1 gezeigte Beispiel wichtig. Dort wird das hohe Es, die zweite Note, von Frame 27 bis



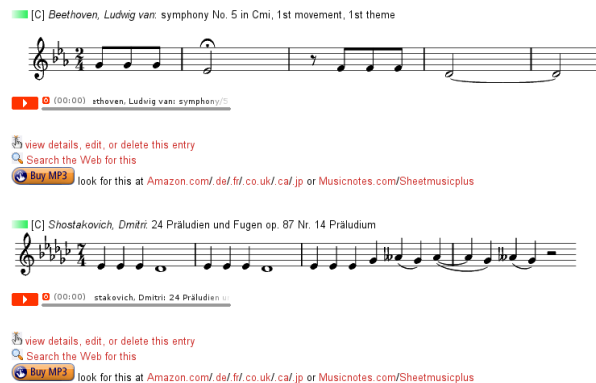


Abb. 9: Diese zwei Melodien aus der Musipedia-Sammlung sind der Gruppe von 8 Noten (Abbildung 8) am ähnlichsten.

Frame 48 gehalten, gefolgt von einem D, das von Frame 49 bis Frame 61 gehalten wird, ohne dass sich die Lautstärke dazwischen deutlich ändern würde.

**Problem: Keine allgemein optimale Gewichtung der Kriterien möglich.** Je nach dem individuellen Sing- oder Pfeifstil kann die Deutlichkeit der Lautstärkeschwankungen von Sänger zu Sänger stark variieren. Bei Legatogesang werden kaum Schwankungen der Lautstärke feststellbar sein, während bei gepfiffenen Staccatomelodien extrem starke Lautstärkeschwankungen zu beobachten sind. Um das daraus resultierende Problem, dass man die Gewichtung des Lautstärke- und des Tonhöhenkriteriums nicht für alle Suchanfragen gleichermaßen optimal fest einstellen kann, zu umgehen, versucht Musipedia nicht direkt Notenanfänge zu erkennen.

**Lösung: Clustering mit Minimierung der Intra-Cluster-Abstände zwischen Frames.** Statt einer direkten “Note Onset Detection” verfolgt Musipedia einen Clustering-Ansatz. Zunächst werden die Frames anhand des ersten, sehr klaren Kriteriums (keine Note enthält einen Frame ohne definierte Tonhöhe) in zusammenhängende Frame-Gruppen geclustert. Danach wird wiederholt für jeden Cluster berechnet, wie hoch seine Kosten sind; der Preis eines Clusters entspricht dabei der Summe der Abstände für jedes enthaltene Paar von Frames. Der Abstand zwischen zwei Frames wiederum ist die Summe zweier Komponenten: der größte Lautstärkeanstieg, der zwischen den beiden Frames beobachtet wurde, und die Tonhöhendifferenz zwischen den beiden Frames. Beide Komponenten werden normiert, also in eine Zahl in einem festen Intervall umgerechnet, wobei der Umrechnungsfaktor auf dem insgesamt größten beobachteten Vorkommen basiert. Durch die Normierung werden individuelle Unterschiede in der Deutlichkeit der Lautstärkeschwankungen zwischen verschiedenen Benutzern neutralisiert.

In jeder Iteration wird der teuerste Cluster in zwei neue Cluster aufgespalten. Die Grenze wird dabei so gewählt, dass die Summe der Kosten der beiden neuen Cluster minimal wird.

Durch diese Vorgehensweise wird kein Cluster explizit nur wegen einer Tonhöhenschwankung oder nur wegen einer Lautstärkeschwankung geteilt, sondern es werden immer beide Kriterien zugleich angewandt. Je nach Datenlage wird automatisch eines oder das andere Kriterium, oder auch beide gleichzeitig, die Position des neuen Notenbeginns bestimmen. Außerdem wird, falls beispielsweise eine Suchanfrage mit tremolierender Stimme gesungen wird, automatisch berücksichtigt, dass überall Lautstärkeschwankungen vorhanden sind, nicht nur an Notengrenzen. Solange die Schwankungen an Notengrenzen noch höher sind als in den Noten, ist das schon hilfreich für die Notenerkennung. Mit anderen Worten: der Kostenminimierungsansatz vermeidet explizite Schwellenwerte für Note Onset Detection-Funktionen, die selbst nach einer Normierung der Intensitäts- und Tonhöheschwankungen noch immer problematisch wären.

Der Notenerkennungsalgorithmus ist in [Typ09] genauer beschrieben.

## 3.2 Suchalgorithmus

### 3.2.1 Abstandsmaße

Die inhaltsbasierte Musiksuche beruht auf drei verschiedenen Merkmalen: Rhythmus (damit meinen wir Folgen von Anfangszeitpunkten von Noten), feine melodische Kontur (codiert als Folgen von Intervallen) und grobe melodische Kontur (codiert als Folge von Tonhöhe-Wiederholungen oder Auf- oder Abwärtsschritten, wie in [Par75]).

Für die Rhythmussuche (“Query by Tapping”) wird ausschließlich die Rhythmuskomponente verwendet, für die Kontursuche ausschließlich die grobe melodische Kontur, und für die diversen Varianten der Melodiesuche (Singen/Pfeifen, Klaviatur) eine Kombination aus Rhythmus- und feiner Intervall-Suche. Die Gewichtung der Rhythmus- und der Intervallkomponente kann dabei vom Benutzer kontrolliert werden.

#### Kontursuche: Editierdistanz für Zeichenketten

Konturen werden mittels einer Editierdistanz verglichen, um eine gewisse Fehlertoleranz zu erreichen. Beim Vergleich zweier Zeichenketten wird eine optimale Kombination aus Einfüge-, Ersetzen- und Löschoptionen bestimmt, die die eine Zeichenkette in die andere Zeichenkette überführt. Die Kosten für diese Operationen, die bei der Optimierung minimiert werden, können variieren, und zwar abhängig sowohl vom Typ der Operation (Einfügen, Löschen, Ersetzen) als auch davon, welche Zeichen behandelt werden. Beispielsweise verursacht das Ersetzen eines “nach unten” durch ein “nach oben” höhere Kosten als das Ersetzen eines “selbe Tonhöhe” durch ein “nach oben”. Das Abstandsmaß und die Methode fürs Finden geeigneter Gewichte sind in [PT01] beschrieben.

#### Rhythmussuche: “Manhattan-EMD”

Die “Earth Mover’s Distance” (EMD) ist ein stetiges Abstandsmaß für gewichtete Punktmengen, das partielles Vergleichen möglich macht. Ausgangspunkt sind zwei Gruppen von gewichtsbehafteten Punkten, die in einem mehrdimensionalen Raum angeordnet sind. Um eine Punktmenge mit der anderen zu vergleichen, berechnet man einen optimalen Gewichtsfluss, der die eine Punktmenge in die andere überführt. Sollte die Summe der Gewichte

der beiden Punktmengen nicht gleich sein, reicht es, die leichtere Punktmenge komplett auf Positionen der anderen Punktmenge zu verschieben (also die eine Punktmenge nur in einen Teil der anderen Punktmenge zu überführen). Bei diesem partiellen Vergleich wird die am besten passende Untermenge der schwereren Punktmenge bestimmt.

Die Stetigkeit dieses Abstandsmaßes ist nützlich, um kleinere Abweichungen bei der Eingabe einer Suchanfrage zu tolerieren – diese schlagen sich nämlich in entsprechend kleinen Änderungen beim berechneten Abstand nieder.

Partielles Vergleichen ist nützlich, um größere Abweichungen bei der Eingabe wie beispielsweise das komplette Weglassen einer Note (bei gesungenen Suchanfragen werde nicht immer alle Noten richtig erkannt) ausgleichen zu können. Solange die übrigen Noten ihren Entsprechungen in einer verglichenen Melodie noch ähneln, kann dank der automatischen Begrenzung des Vergleichs auf diese Noten das völlige Fehlen einer Note toleriert werden.

Für den Rhythmusvergleich arbeitet Musipedia mit Punkten in einem eindimensionalen Raum, wobei jeder Punkt dasselbe Gewicht besitzt. Die einzige Dimension ist der Anfangszeitpunkt einer Note. Da die EMD für solche eindimensionale Punktmengen metrische Unterräume hat, in denen sie genau der  $l_1$ -Norm entspricht (auch “Manhattan-Norm” genannt), nennen wir diese Spielart der EMD “Manhattan-EMD”. Eine genauere Beschreibung des Abstandsmaßes und auch der Indexierungstechnik findet sich in [TWT08].

### **Intervallsuche: “Manhattan-EMD”**

Für den Vergleich zweier Folgen von Intervallen wird dasselbe Abstandsmaß verwendet wie für den Vergleich zweier Folgen von Noten-Startzeitpunkten (siehe den vorigen Abschnitt). Das bedeutet: in den metrischen Unterräumen, d. h. falls gleich lange Folgen von Intervallen verglichen werden, ist der Abstand gleich der Summe der Differenzen der Intervalle, die in den Folgen jeweils an derselben Stelle stehen. Falls Folgen verglichen werden, die nicht gleich viele Intervalle enthalten, wird automatisch in der längeren Folge die ähnlichste Unterfolge zum Vergleich herangezogen. Da das Weglassen eines Intervalls aus einer Intervallfolge einen Einfluss auf alle nachfolgenden Noten hätte, werden die Intervallfolgen vor der Anwendung der Manhattan-EMD in normierte absolute Werte umgerechnet. Das heißt, die mit den Intervallfolgen codierten Noten werden so transponiert, dass die ersten Noten gleich sind, und dann wird mit Folgen von Tonhöhen statt mit Folgen von Intervallen gerechnet, so dass das Weglassen eines Elements der Folge nicht den Rest der Folge verfälscht.

### **3.2.2 Suchindex**

Das Berechnen der Abstandsmaße zwischen einer Suchanfrage und jedem einzelnen Eintrag in der Datenbank wäre zwar für 31.000 kurze Melodien machbar, aber nicht sehr elegant. Für die 100.000 kompletten MIDI-Dateien, die ebenfalls mit Musipedia durchsucht werden können, würde das die Suche schon deutlich langsamer machen.

Daher verwendet Musipedia die Vantage-Indexing-Methode [Typ07], bei der für jedes Objekt in der Datenbank die Abstände zu einer kleinen Gruppe von Vantage-Objekten vorausberechnet werden. Wenn dann eine Suchanfrage beantwortet werden soll, werden auch für diese die Abstände zu denselben Vantage-Objekten berechnet. Dann werden nur die Objekte aus der Datenbank in die Suche mit einbezogen, die ähnliche Abstände zu den Vantage-Objekten

haben. Falls die Dreiecksungleichung für das verwendete Abstandsmaß gilt, wird dadurch das Suchergebnis nicht verfälscht.

Als “Objekt in der Datenbank”, das mit Vantage-Objekten verglichen wird, nimmt Musipedia nicht komplette Einträge, sondern nur Gruppen von Noten, wobei die Gruppengröße in einem bestimmten Bereich liegt.

Trotzdem ist die Dreiecksungleichung nicht notwendigerweise erfüllt. Die “Manhattan EMD” beispielsweise ist unter anderem Wegen der Verletzung der Dreiecksungleichung keine Metrik. Daher müssen noch weitere Tricks angewandt werden, um sicherzustellen, dass die Verwendung des Suchindexes das Ergebnis nicht verfälscht. Diese Methode ist in [TWT08] beschrieben.

Auch für die Editierdistanz, die für die grobe Kontursuche eingesetzt wird, wird Vantage Indexing verwendet. Hier gibt es allerdings noch offene Probleme. Die Fragestellung ist nicht, die laut Editierdistanz ähnlichsten Zeichenketten zu finden, sondern diejenigen Paare von Zeichenketten, bei denen der Abstand der kürzeren Zeichenkette zu einem beliebigen Präfix der längeren Zeichenkette besonders gering ist. In dieser Form ist das Abstandsmaß keine Metrik, und eine ideale Lösung des Indexproblems steht noch aus. Musipedia umgeht das Problem derzeit dadurch, dass aufgrund der ersten fünf Zeichen der verglichenen Zeichenketten mit der Vantage-Methode der Suchraum eingeschränkt wird. Beim Vergleich von Zeichenketten mit fester Länge ist die verwendete Editierdistanz eine Metrik. Die dadurch verursachte Einschränkung, dass die Suche nur für mindestens 5 Noten funktioniert, ist kein großes Problem, da kürzere Suchanfragen ohnehin kein nützliches Ergebnis liefern würden (viel zu viele Melodien haben für die ersten 5 Noten dieselbe grobe melodische Kontur).

## 4 Open Source-Komponenten

Ohne quelloffene Komponenten könnte Musipedia nicht in dieser Form existieren. Nicht nur grundlegende Dinge wie das Server-Betriebssystem (Debian), die Datenbank (MySQL), das Content Management System (Typo3) und die Skriptsprache PHP, sondern auch Komponenten, die mehr mit der musikalischen Anwendung zu tun haben, sind Open Source-Software.

### 4.1 Red5: Wird für das Aufnehmen gesungener oder gepiffener Suchanfragen verwendet.

Red5 (<http://www.red5.org>) ist ein in Java implementierter, quelloffener Flash-Server, der es unter anderem möglich macht, über ein kleines, im Browser ausgeführtes Flash-Programm Tonaufnahmen an den Server zu schicken. Dort steht die Aufnahme dann im FLV-Format zur Verfügung, einem proprietären Adobe-Format.

### 4.2 Nelly2PCM: Umwandlung vom FLV- ins Wave-Format

Nelly2PCM ist ein quelloffenes Programm, das bis Sommer 2008 von [code.google.com](http://code.google.com) heruntergeladen werden konnte. Der Zweck dieses Programms ist die Umwandlung von Daten aus dem proprietären Flash Video-Format (FLV) in Formate, die besser dokumentiert sind und von mehr Programmen gelesen werden können. Leider wurde Google im Sommer 2008 durch

eine einstweilige Verfügung, die sich auf den US-amerikanischen “Digital Millenium Copyright Act” (DMCA) beruft, dazu bewegt, dieses Projekt nicht weiter verfügbar zu machen. Möglicherweise spielte der Gedanke, dass das Programm fürs Umgehen von Kopierschutz-Mechanismen verwendet werden könnte, dabei eine Rolle<sup>1</sup>. Leider werden dadurch auch Anwendungen, die wie Musipedia ihren Benutzern erlauben wollen, über das Internet eine kleine Tonaufnahme anzufertigen und zu verarbeiten, in Zukunft erschwert, obwohl dabei keinerlei Kopierschutz umgangen wird.

### **4.3 Praat: “Doing Phonetics by Computer” - wird fürs Erkennen der Tonhöhe und Intensität verwendet.**

Paul Boersma und David Weenink vom Institute of Phonetic Sciences, Universiteit van Amsterdam, stellen mit Praat (<http://www.praat.org>) ein sehr leistungsfähiges Programm zur Analyse von Klang zur Verfügung. Wie der Name suggeriert (“praten” ist das niederländische Wort für “Sprechen”, “praat” ist der Imperativ oder auch die erste, zweite oder dritte Person Singular), ist der Hauptzweck des Programms die Sprachanalyse, aber nach kurzen Experimenten mit Praat und einigen anderen Algorithmen, die beispielsweise von Paul Brossiers “Aubio”<sup>2</sup> bereitgestellt werden (Yin [dCK02], mcomb, fcomb und Schmitt), wurde schnell klar, dass Praat die Anforderungen von Musipedia sehr gut erfüllt. Musipedia verwendet Praat, um für gepfiffene oder gesungene Suchanfragen eine Tonhöhen- und eine Intensitäts-(Lautstärke-)Kurve zu erhalten. Die Tonhöhe (Fundamental Frequency) wird durch Praat mit einem Algorithmus bestimmt, der mittels Autokorrelation Periodizität erkennt und recht robust ist gegen Rauschen und Störgeräusche, wie sie bei einer Anwendung wie Musipedia oft auftreten. Details dieses Algorithmus sind in [Boe93] beschrieben.

### **4.4 JavaScript Drag & Drop von Walter Zorn**

Nach der Notenerkennung kann der Benutzer mittels Drag & Drop noch die genaue Position jeder Note, d. h. ihre Tonhöhe und den Anfangszeitpunkt, korrigieren. Dazu wird die JavaScript-Drag & Drop-Bibliothek von Walter Zorn (<http://www.walterzorn.de/dragdrop/dragdrop.htm>) benutzt.

### **4.5 Lilypond: Notensatz**

Han-Wen Nienhuys und Jan Nieuwenhuizen haben das Notensatzprogramm “Lilypond” [NN03] geschrieben (<http://www.lilypond.org>), das von Musipedia für die grafische Darstellung der Melodiensammlung und die Repräsentation der Melodien als MIDI-Dateien verwendet wird.

### **4.6 Image\_Graph: Visualisierung von Tonhöhen und Intensität**

Image.Graph ([http://pear.veggerby.dk/wiki/image\\_graph](http://pear.veggerby.dk/wiki/image_graph)) ist eine PHP-Bibliothek von Neufeind und Veggerby, mit der man Balkendiagramme und andere Diagramme erstellen

---

<sup>1</sup>siehe <http://chillingeffects.org/anticircumvention/>

<sup>2</sup><http://www.aubio.org/>

kann. Gepliffene und gesungene Suchanfragen werden damit visualisiert, um dem Benutzer eine Vorstellung davon zu geben, ob er deutlich genug gesungen hat, um eine Umwandlung in Noten überhaupt möglich zu machen.

## 4.7 Freesound: Samples für die Klaviatur

Auf <http://www.freesound.org/> stehen zahlreiche Tonaufnahmen zur Verfügung, die mit Creative Commons-Lizenzen verwendet werden können. Musipedia verwendet einige dieser Aufnahmen, um das JavaScript-Klavier und den Editor für aus gesungenen Suchanfragen extrahierte Noten mit Klang zu versorgen.

## 4.8 CGAL: Range Trees

Die “Computational Geometry Algorithms Library” (<http://www.cgal.org>) enthält die Implementierung eines verschachtelten “Range Tree” (Bereichs-Baumes). Damit ist es möglich, in einem mehrdimensionalen Raum effizient alle Elemente zu finden, die sich in einem Hyperquader befinden. Musipedia nutzt diese Datenstruktur, um Gruppen von Noten in der Datenbank zu finden, deren Abstände zu Vantage-Objekten in bestimmten Bereichen liegen (nah an den Abständen zwischen einer Suchanfrage und denselben Vantage-Objekten).

# 5 Zusammenfassung

Der Musipedia-Internetdienst fürs gemeinsame Editieren einer Musiksammlung und fürs Identifizieren gepliffener, gesungener oder eingetippter Melodiefragmente ist ein gutes Beispiel für das “Standing on the Shoulders of Giants”, das möglich gemacht wird durch Open Source-Software und Daten, die unter einer großzügigen Lizenz stehen wie beispielsweise einer Creative Commons-Lizenz. Natürlich ist das Grundprinzip des Verwendens von Bibliotheken, die Teilprobleme lösen, auch ohne Offenheit möglich, aber zumindest Musipedia wäre in ihrer derzeitigen offenen und kostenlos angebotenen Form nicht möglich, wenn die verwendeten Softwarekomponenten signifikante Kosten verursachen würden.

## Offene Probleme

Es gibt noch einige offene Probleme: eines der verwendeten nicht metrischen Abstandsmaße, die Editierdistanz für Präfixe mit partiellem Vergleich, wird noch nicht perfekt durch eine Indexmethode unterstützt. Noch besser wäre die Unterstützung einer Editierdistanz, die eine kurze Suchanfrage mit einem Teil einer längeren Zeichenkette vergleicht, der irgendwo auftreten kann, nicht notwendigerweise am Anfang. Aus der Bioinformatik, die mit Gensequenzen ähnliche Probleme hat, sind n-gram-Ansätze vertraut, die in abgewandelter Form möglicherweise nützlich sein könnten.

Die von Musipedia verwendete Methode fürs Erkennen von Noten in gesungenen oder gepliffenen monophonen Tonaufnahmen könnte noch verbessert werden, indem das Abbruchkriterium fürs Aufspalten von Noten besser formuliert wird. Auch durch einen noch besseren Input in Form korrekterer Tonhöhe- und Lautstärkekurven als sie derzeit von Praat geliefert werden, ließe sich die Leistung noch steigern.

## 6 Danksagung

Der Autor dank dem FWF für die Finanzierung von Teilen der in diesem Artikel vorgestellten Ergebnisse (Indexierung, Audioanalyse) im Rahmen eines Lise-Meitner-Projekts (Projektnr. M1027-N15). Dank gebührt auch den Autoren der zahlreichen quelloffenen Programme, die Musipedia mit möglich gemacht haben (siehe Abschnitt 4), und nicht zuletzt den Musipedia-Benutzern, die die Sammlung pflegen.

## Literatur

- [BM49] H. Barlow and S. Morgenstern. *A Dictionary of Musical Themes*. Ernest Benn Limited, 1949.
- [Boe93] Paul Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *IFA Proceedings*, 17:97–110, 1993.
- [BW08] P. Boersma and D. Weenink. Praat: doing phonetics by computer (version 5.0.29) [computer program], 2008. Retrieved August 4, 2008, from <http://www.praat.org/>.
- [dCK02] Alain de Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.
- [NN03] H.-W. Nienhuys and J. Nieuwenhuizen. Lilypond, a system for automated music engraving. In *Proceedings of the XIV. Colloquium on Musical Informatics (XIV CIM 2003)*, 2003. See <http://www.lilypond.org>.
- [Par75] Denys Parsons. *The directory of tunes and musical themes*. Spencer Brown, 1975.
- [PT01] L. Prechelt and R. Typke. An interface for melody input. *ACM Transactions on Computer-Human Interaction*, 8(2):133–149, 2001.
- [TWT08] R. Typke and A. Walczak-Typke. A combined vantage indexing and tunneling method. 2008. 9th International Conference on Music Information Retrieval.
- [Typ07] R. Typke. *Music Retrieval based on Melodic Similarity*. Doctoral Thesis, Universiteit Utrecht, <http://rainer.typke.org/books.html>, 2007.
- [Typ09] R. Typke. Note recognition from monophonic audio: a clustering approach. 2009. 7th Workshop on Adaptive Multimedia Retrieval, Madrid.
- [Wan03] A. Wang. An industrial strength audio search algorithm. In *ISMIR Proceedings*, Baltimore, 2003.