

Melodiesuche

Ansätze zur
Analyse und Indizierung von Melodien
in einer durchsuchbaren Datenbank
unter Verwendung der MUSITECH-Umgebung.

Referent: Christian Datzko, Universität Osnabrück, datzko@t-online.de

Grundüberlegungen

- Suchmaschine für Melodien?
 - Für Begriffe und Namen gibt es unter anderem Enzyklopädien und Fach-Lexika, aber auch Suchmaschinen wie Google
 - Wer heute nach einer Melodie sucht, sucht meist nach Suchbegriffen wie dem Namen eines Komponisten, der Werkbezeichnung oder ähnlichem.
 - Die Suche nach Melodien direkt ist derzeit nur sehr eingeschränkt möglich.

Existierende Lösungen

- BARLOW und MORGENSTERN (1948): *A Dictionary of Musical Themes*.
 - Nach C-Dur/-Moll transponiert
 - Nur Tonhöhe wird wiedergegeben
 - Keine Fehlertoleranz

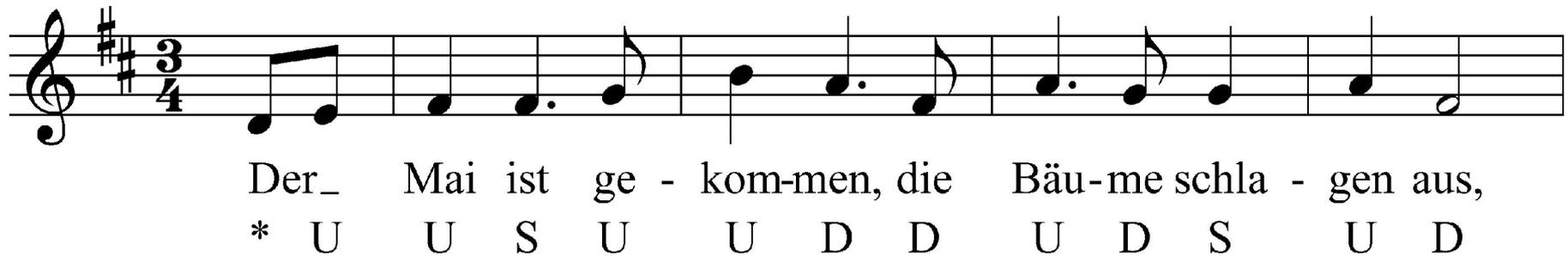


Der_ Mai ist ge - kom-men, die Bäu-me schla - gen aus,
C D E E F A G E G F F G E

The image shows a musical staff in treble clef with a key signature of one sharp (F#) and a 3/4 time signature. The melody consists of the following notes: C4 (quarter), D4 (quarter), E4 (quarter), E4 (quarter), F4 (quarter), A4 (quarter), G4 (quarter), E4 (quarter), G4 (quarter), F4 (quarter), F4 (quarter), G4 (quarter), E4 (quarter). The lyrics are written below the staff, with a line under 'Der' and hyphens in 'ge - kom-men' and 'schla - gen'.

Existierende Lösungen

- PARSONS, DENYS (1975): *Directory of Tunes and Musical Themes*.
 - Tonartunabhängig
 - Nur Richtung der Melodie (Down, Same, Up)
 - Geringe Fehlertoleranz



Der_ Mai ist ge - kom-men, die Bäu-me schla - gen aus,
* U U S U U D D U D S U D

Existierende Lösungen

- PAUWS, STEFFEN (2002): *CubyHum: A Fully Operational Query by Humming System*.
 - Komplett fehlertolerant
 - Berücksichtigt Rhythmus
 - Nur mit Computern sinnvoll einsetzbar
 - Vergleicht je zwei Melodien auf ihre Ähnlichkeit, gibt einen Ähnlichkeitswert $\in [0..1]$ an
 - Formel zur Berechnung beruht auf dem Prinzip der „Edit Distance“

Formel für CubyHum

$$\begin{array}{c}
 \text{Zu durchsuchende Melodie (} i \text{)} \\
 D = \begin{array}{c} \text{Zu} \\ \text{suchende} \\ \text{Melodie} \\ \text{(} j \text{)} \end{array} \begin{pmatrix} 0 & D_{1,1}+1+\frac{K \cdot L(q_2)}{L(q_1)} & \dots & D_{i-1,1}+1+\frac{K \cdot L(q_i)}{L(q_{i-1})} & \dots & D_{N-1,1}+1+\frac{K \cdot L(q_N)}{L(q_{N-1})} \\ \vdots & ? & & \dots & & ? \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & ? & & \dots & & ? \end{pmatrix}
 \end{array}$$

$$D_{i,j} = \min \left\{ \begin{array}{l}
 D_{i-1,j} + 1 + K \cdot \left| \frac{L(q_i)}{L(q_{i-1})} \right| \quad \text{(Intervall-Löschung)} \\
 D_{i-2,j-1} + 1 + K \cdot \left| \frac{L(q_{i-1}) + L(q_i)}{L(q_{i-2})} - \frac{L(s_j)}{L(s_{j-1})} \right|, \text{ wenn } q_{i-1} + q_i = s_j, i > 2 \quad \text{(Noten-Löschung)} \\
 D_{i-1,j-1} + \frac{C}{\sigma} \cdot |q_i - s_j| + K \cdot \left| \frac{L(q_i)}{L(q_{i-1})} - \frac{L(s_j)}{L(s_{j-1})} \right| \quad \text{(Rückung)} \\
 D_{i-1,j-2} + 1 + K \cdot \left| \frac{L(q_i)}{L(q_{i-1})} - \frac{L(s_{j-1}) + L(s_j)}{L(s_{j-2})} \right|, \text{ wenn } q_i = s_{j-1} + s_j, i > 2 \quad \text{(Noten-Einfügung)} \\
 D_{i,j-1} + 1 + K \cdot \left| \frac{L(s_j)}{L(s_{j-1})} \right| \quad \text{(Intervall-Einfügung)}
 \end{array} \right.$$

Neuer Algorithmus

- Motiv-Melodie-Suche
 - Melodien werden in ihre Motive segmentiert
 - Es werden Stücken mit ähnlichen Motiv-Konturen gesucht (Grobsuche)
 - Je häufiger ein Motiv in einer Melodie auftritt, als desto charakteristischer wird es bewertet
 - Eher seltene Motive werden stärker gewichtet, eher häufige Motive (z.B. typische Wendungen) werden geringer gewichtet
 - Bei Bedarf kann ein genauerer Vergleich nachgeschaltet werden (Feinsuche)

Segmentierung

- Ausnutzen der MUSITECH-Umgebung
 - Ein Segmentierer ist schon implementiert!
 - „Größere Einsatzabstände bewirken bei ansonsten gleichen Parametern eine Gruppengrenze.“
 - Aufgrund des modularen Konzepts von MUSITECH kann jedoch jeder andere Segmentierer verwendet werden

Der_ Mai ist ge - kom-men, die Bäu-me schla - gen aus,

1 1 1 1 2 2 2 3 3 4 4 4 4

Abstrahierung der Motive

- Ähnliche Idee wie bei Denys Parsons
 - Nur für die Grobsuche relevant
 - Abstraktion der Melodie in Down, Equal Height, Up
 - Abstraktion des Rhythmus in Longer, Equal Length, Shorter
- Vergleich der Melodie-Rhythmus-Konturen im Rahmen der Suche zusätzlich noch mit einer Bewertung der Ähnlichkeit wie bei Edit Distance

Grobsuche

$$0 \leq l = \frac{1}{d(S, Q) + 1} \cdot \frac{n_1}{n_2} \leq 1$$

- Bedeutung der Symbole:
 - $d(S, Q)$: Edit Distance zwischen Suchmotiv und Fragemotiv
 - n_1, n_2 : Anzahl der Segmente in der aktuellen Segmentklasse, Anzahl der Segmente in der Melodie insgesamt
 - l : lokaler Wert

Gewichtung und Feinsuche

- Gewichtung:

$$l_{\text{neu}} = l \cdot \left\{ \begin{array}{ll} 1 & \text{für: die Segmentklasse kommt selten vor} \\ 0.75 & \text{für: die Segmentklasse kommt durchschnittlich vor} \\ 0.5 & \text{für: die Segmentklasse kommt häufig vor} \end{array} \right\}$$

- Feinsuche:

- Verwendet eine der bisher vorgestellten Methoden um ein Ähnlichkeiten $\in [0..1]$ zwischen den Motiven der Segmentklasse zu berechnen und multipliziert diese jeweils mit l_{neu} .

Lokales Maximum und Gesamtsumme

- Die Maxima der verschiedenen durch die Feinsuche ermittelten Werte werden für alle Segmentklassen aufsummiert.
- Insgesamt hat man so einen Wert $\in [0..1]$, der ein Maß für die Ähnlichkeit zwischen den beiden verglichenen Melodien ist.

Vergleich der Methoden (Effizienz)

	Zeitbedarf Indizierung	Zeitbedarf Suche	Platzbedarf Index
Parsons	$O(1)$	$O(1)$	$O(L)$
CubyHum	$O(N)$	$O(M \cdot N \cdot L)$	$O(N \cdot L)$
Motiv-Suche	$O(N \cdot L)$	$O(N^2 \cdot M^2 \cdot L)$	$O(M + N)$

M = Länge der Suchmelodie, N = (maximale) Länge der zu durchsuchenden Melodie, L = Anzahl der Melodien in der Datenbank

- Parsons: sehr schnell und wenig Platzbedarf
- CubyHum: recht langsam, normaler Platzbedarf
- Motiv-Suche: extrem langsam, wenig Platzbedarf

Vergleich der Methoden (Qualität)

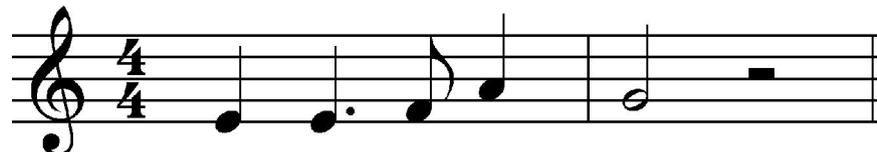
- Parsons
 - Nicht fehlertolerant
 - Nicht für große Datenbanken geeignet, weil zu grob
- CubyHum
 - Ergebnisse mit guter Qualität
 - Teile der Suchmelodie sind nicht zu finden
- Motiv-Suche
 - Geht am stärksten auf den musikalische Aufbau ein
 - Ergebnisse sind stark von der Qualität des Segmentierers abhängig

Demonstration der Implementation

- 1. Suchmelodie (MIDI-Datei, Es-Dur):



- Datenbank:
 - 500 MIDI-Dateien der Digital Tradition-Datenbank
 - Suchmelodien (zur Überprüfung)
- 2. Suchmelodie (Ausschnitt der vorherigen MIDI-Datei, in C-Dur):



Gefundene Melodien zum Vergleich

- The Battle of Harlaw (2) (BATHRL2.MID)



As I cam in by Den- nie- dier And doon by Ne- ther- ha', There was



fif- ty thou- sand Hie- land men Cam mair- ching tae Har- law, Wi' a



dree dree dra- die drum- tie dree, A dree dree drum- tie dra.

Gefundene Melodien zum Vergleich

- Bessie of Ballydubray (BESSBAL.MID)



One night as this young man lay down for to sleep, Young Bessie came



to him and o'er him did weep, Saying, "You are the young man who



led me astray, Away from my friends from sweet Ballydubray.

Literatur

BARLOW, HAROLD und MORGENSTERN, SAM (1948): *A Dictionary of Musical Themes*. Crown Publishers, New York.

GREENHAUS, DICK (2004): *The Digital Tradition*.

URL: <http://www.mudcat.org/DigiTrad-blurb.cfm>.

PARSONS, DENYS (1975): *Directory of Tunes and Musical Themes*. Spencer Brown, Cambridge.

PAUWS, STEFFEN (2002): *CubyHum: A Fully Operational Query by Humming System*. In: MICHAEL FINGERHUT, Hg., *ISMIR 2002 Conference Proceedings – Third International Conference on Music Information Retrieval*. IRCAM – Centre Pompidou, Paris.

URL: <http://ismir2002.ismir.net/proceedings/02-FP06-2.pdf>.

Weiterführende Informationen

- Sprechen Sie mich gerne an!
- URL zu dieser Präsentation:
`http://christian.datzko.ch/musik-wissenschaft/melodiesuche_vortrag.pdf`
- E-Mail: `datzko@t-online.de`
- Über dieses Thema habe ich meine Examensarbeit geschrieben, ich schicke sie Ihnen auf Anfrage gerne zu.