A Unified Approach to Content-Based and Fault Tolerant Music Identification

Michael Clausen
Frank Kurth
University of Bonn

Proceedings of the Second International Conference on WEB Delivering of Music
2002 IEEE

A Full-Text Retrieval Approach to Content-Based Audio Identification

Andreas Ribbrock
Frank Kurth
University of Bonn

Overview

- Introduction
- Data Modeling
- Fault Tolerance
- Content-based Search in Scores
- Content-based Search in Audio Data
- Our Project
- Article critics

Introduction

- The two articles deal with indexing and searching of polyphonic and PCM audio
- When dealing with polyphonic audio searching is done using pitches
- When searching in PCM audio some massive data reduction needs to be done
- Searching in PCM audio is accomplished by creating feature extractors

Data Modeling

- Much related work use string-based representation
- U represent all possible objects and D is a document
  $D \subseteq U$
- Polyphonic music is represented by

  $$U := Z \times P$$

- Where Z is onset time, and P is the set of admissible pitches

Data Modeling

- A query is a set of notes $Q \subset Z \times P$
  and a query is represented: $Q = \{[t_1, p_1], \ldots, [t_n, p_n]\}$

- A hit on a query Q in a database $D = (D_1, \ldots, D_N)$
  is a pair $(t, i) \in Z \times [1:N]$ such that $Q + t := \{[t_1 + t, p_1], \ldots, [t_n + t, p_n]\} \subseteq D_i$

- All exact hits are given by $H_D(Q) := \{(t, i) \mid Q + t \subseteq D_i\}$

Data Modeling

- When modeling PCM audio we use a feature extractor $F[x](n) = \ell$
- For a fixed feature extractor F and signal x we obtain a document consisting of all nonzero features along with there positions

$$D_f(x) := \{[n, \ell] \mid F[x](n) = \ell \neq 0\} \subseteq Z \times [1:c]$$

- The set of all hits is defined by:

$$H_{D_F}(Q) := \{(t, i) \mid D_F(Q) + t \subseteq D_F(x_i)\}$$

Fault Tolerance

- In real scenarios users may not remember nodes are so some fault tolerance is needed
- Two ways to deal with Fault Tolerance
  - k-Mismatches
  - Fuzzy Search

Fault Tolerance k-Mismatches

- k-mismatches is defined by $H_{D,k}(Q)$ which is all the matches to a query Q containing at most k non matching objects

$$\{(t,i) \mid \exists Q' \subseteq Q, |Q'| \geq |Q| - k \; such\,that \; Q' + t \subseteq D_i\}$$

- This can be used to create a ranked list if the output of $H_{D,k}(Q)$ is sorted in decreasing order

Fault Tolerance Fuzzy Search

- Fuzzy search is used when there is doubt about certain parts of the query
- For each $q \in Q$ there is a set of alternatives $\mathbf{F}_q \subseteq U$ and is called a fuzzy query $\mathbf{F}_Q$. If there is no doubt about a specific $q \in Q$ one would choose $\mathbf{F}_q = \{q\}$
- An elementary query of $\mathbf{F}_Q$ is if there for each $q \in Q$ exist exactly one alternative.
- The hit of the fuzzy query is then<
  $\{(t, j) \mid P + t \subseteq D_j \text{ for an elementary query } P \text{ of } \mathbf{F}_Q\}$

Content-based Search in Scores
Searching Polyphonic Scores

Example of a search Document D1with two queries

$$Q_1 := \{[0, 74], [4, 70]\}, Q_2 := \{[4, 74], [8, 70]\}$$

$$D1 := \{[8, 74], [11, 77], [11, 69], [12, 77], [12, 72], [16, 74], [16, 65], [20, 70], [23, 74], [23, 66], [24, 74], [24, 69], [28, 70], [28, 62]\} \subset U$$

Then the set of all t such that is for $Q_1 + t \subseteq D_1$ *and* $Q_2 + t \subseteq D_1$ is

$$Q_1 = \{(16,1), (24,1)\} \text{ and } Q_2 = \{(12,1), (20,1)\}$$

Content-based Search in Scores
Searching Polyphonic Scores

- If we include knowledge of metrical position we can reduce the exact hit of our queries
- Our Universe is modified and takes nodes from the set

  $V := Z \times [0 : \ell\text{-}1] \times P \qquad \ell := \frac{br}{u} => \ell = \frac{3 \cdot 16}{4} = 12 \qquad H_D([0, \lambda, p]) := \{(t,i) \mid [t, \lambda, p] \in D_i\}$

- Our Document transforms to

  $D1 := \{[0,8,74],[0,11,77],[0,11,69],[1,0,77],[1,0,72],$
  $[1,4,74],[1,4,65],[1,8,70],[1,11,74],[1,11,66],$
  $[2,0,74],[2,0,69],[2,4,70],[2,4,62]\} \subset U$

- The queries transform to
  $Q_1 = \{[0,0,74],[0,4,70]\}$ and $Q_2 = \{[0,4,74],[0,8,70]\}$
- For Q_1 the exact hit is $(2,1)$ and for Q_2 the exact hit is $(1,1)$

Content-based Search in Scores
Search results

- MIDI database with 12000 songs and 327 MB in size.
- Search index consist of the sets $H_D([0, \lambda, p)$
- Hardware is Pentium II, 333 MHz, 256 MB RAM, Windows NT 4.0

a	4	8	12	16	18	20	30	50	100
b	51	86	92	97	96	100	107	125	159
c	1	5	7	10	11	12	19	31	64

- Row a - Number of nodes in a query
- Row b - Total system response
- Row c - Time to fetch inverted lists

Searching in Melody Data Bases: notify!-bywhistling

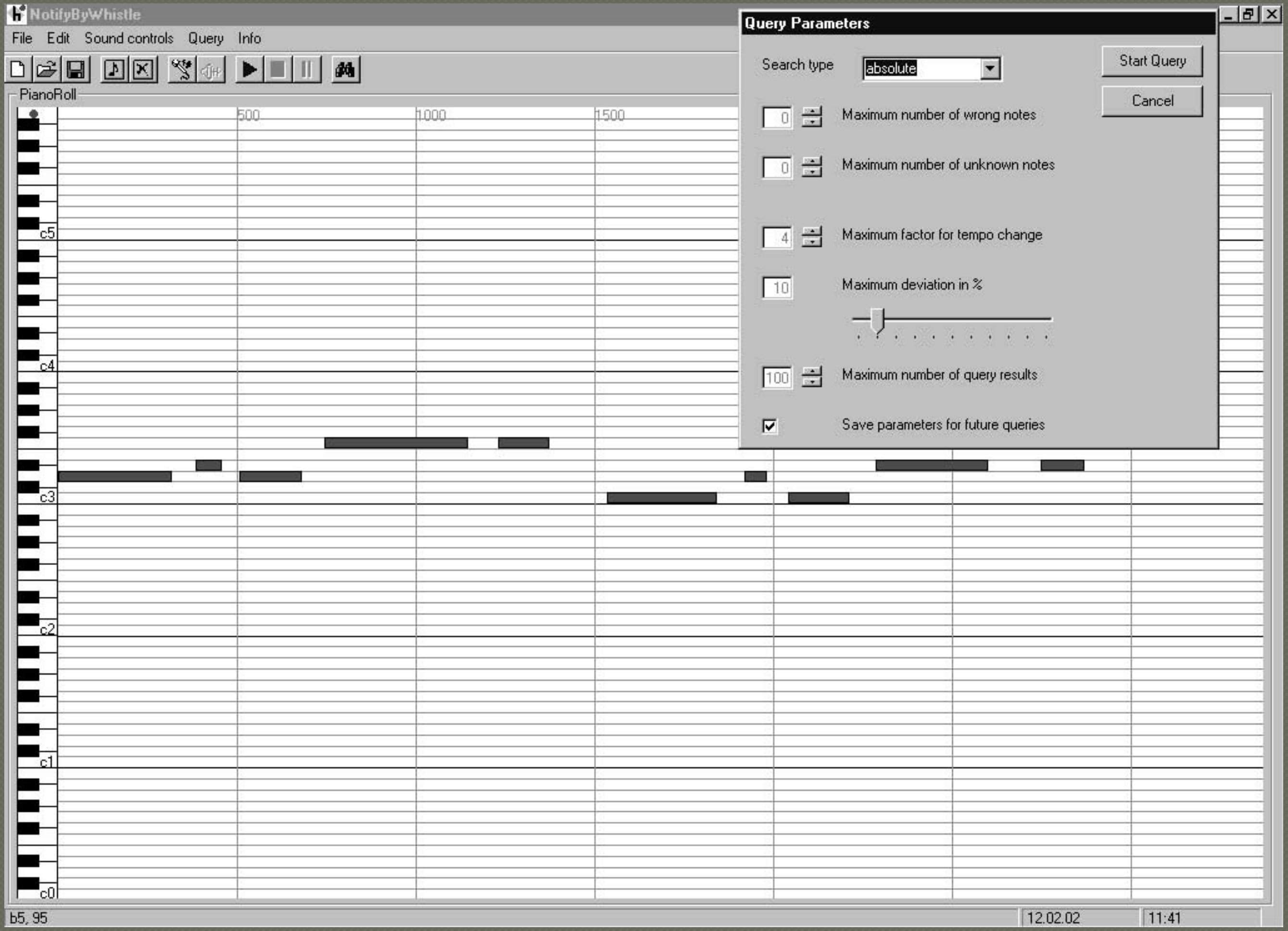
- The whistled song from a user normally have a different tempo than the original

- The whistled tempo curve changes over time so rather than static s-times value, the changes lie between $s_\ell \leq s \leq s_u$

- The user whistles a song to an algorithm which outputs a sequence of MIDI-notes which can be edited in a program

- A search for "Yellow Submarine" in the database with a rhythm tolerance of 10% 23 were found

Searching in Melody Data Bases: notify!-bywhistling



Content–based Search in Audio Data: The audentify!–System

- The audentify System is designed identify short excerpts (1-5 sek)
- It takes use of feature extractors $D_F(x)$ for a given base signal x and a feature extractor F
- Feature density of a feature extractor is defined as $\delta = \frac{k}{n}$ if each interval of length n taken from $F[X]$ contains k features

Content–based Search in Audio Data: Max Feature

- First a input signal is prefiltered, $C_f[x] := f * x$ with a FIR filter f
- $M_m[x]$ denotes m-significant local maxima of x
- $M'_m[x]$ denotes local maxima on non-zero elements of x
- Then a γ operator is defined as a sequence that contains at the position of each significant maximum, the distance to the next significant maximum
- Then a linear quantizer Q_c reduces the extracted distances to c feature classes

$$F_{Max} = Q_C \circ \delta \circ M'_K \circ C_f$$

Content–based Search in Audio Data: Volume Feature

- A more robust Feature Extractor than the one showed before is based on the volume of the signal
- First volume for a given signal is analyzed using Hamming-window
- Then the smoothed by a low pass filter
- The local maxima and minima is extracted using operator $M_K^{''}$
- Then the difference between the local maxima is found

$$F_{Vol} := \delta'_{O_1,O_2} \circ M_K^{''} \circ C_f \circ V_{s,w}$$

Content–based Search in Audio Data: WFT-Feature

- Both F_{Max} and F_{Vol} are feature extractors which are working in the time domain where the WFT-Feature is extracted from the frequency domain
- A signal x is transformed into the frequency domain using a windowed Fourier transform
- Then using an operator S the frequency centroid is calculated
- Then a low pass filter is used, the local maxima are extracted and the distance is between the two consecutive local maxima are calculated

$$F_{wft} := Q_c \circ \delta \circ M_K \circ C_f \circ S \circ w_{g,s}$$

Content–based Search in Audio Data: Code Feature

- A problem with the feature extractors presented before is that two signals with different signal quality can different features
- To solve this problem a rough binary quantizer is used on the signal
- Then a string over a finite alphabet approximating the signal x is then produced using code. Two signals with different signal quality should then have the same string
- Then the nearest codebook entry is denoted to a bit vector

$$F_{code}^{C} := \varepsilon_C \circ C_{n,m} \circ P[x]$$

Content–based Search in Audio Data: audentify!–mobile

5 types of query signals is considered

- Short parts of a track taken (cropped) from an arbitrary position within the track
- MP3 re–encoded and decoded versions of a track were MP3–compression is performed at 96 kbps
- Tracks recorded by placing microphone in front of a loudspeaker
- Tracks recorded by placing a cellular phone (GSM) in front of a loudspeaker

Content–based Search in Audio Data: audentify!–mobile

- Tracks recorded by a cellular phone with the incomming audio signal recorded by placing a microphone in front of the loudspeaker of a receiving phone
- For signals 1-3 only a very short sample was needed to find a match. For signal 4-5 at least a sample of 15-20 seconds is needed before a match could be found

Our Project

In our project we try to recognize PCM audio recorded from a mobile phone.

We can use the knowledge about the different feature extractors and which ones are good to use when working with highly distored audio material

Article critics

- Positive:
  - Many things from the two articles are relevant for our project
  - First half of the first article is easy to understand

- Negative:
  - Requires some background knowledge to fully understand what is going on
  - Could use more examples and illustrations, there is a lot of text
  - Last half of the first article is hard to understand
  - The second article is very short and compressed