Querying Improvised Music

Do you sound like yourself?

Michael Jewell, Christophe Rhodes and Mark d'Inverno

Introduction

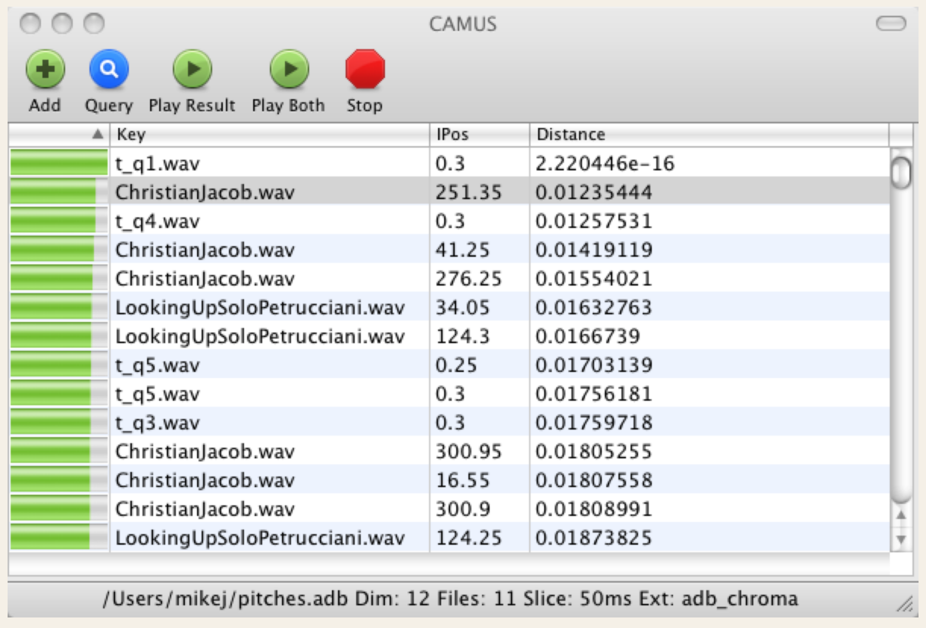
This work describes and investigates a practical use-case for searching in an exploratory fashion for fragments of audio within a small collection, in the manner of **Query-by-Example**. Our motivation for this is to help performers, especially of improvised music, to critically assess their own **improvisation style** and how it relates to others'; we aim to support investigation of questions such as how performance characteristics

- develop through a single performance;
- change over the course of a career;
- compare with those of great, professionally-recorded musicians;
- are affected by performance or recording context.

In addition, because many musical performers are not highly technical users, this has acted as a demonstrator for a **high-level user interface** to the audioDB search engine [8] developed by the OMRAS2 project, as well as a proof-of-concept to an integrated workflow using other **semantically-enriched** tools [1], such as Vamp plugins for audio feature extraction.

CAMUS

CAMUS (previously iAudioDB) provides a **friendly and familiar interface** to audioDB's functionality for users wishing to investigate and understand their own collections. It does not require knowledge of the workings of feature extractors, as it is able to use RDF descriptions of Vamp plugins to perform extraction; nor does it require knowledge of the low-level audioDB functionality – it automatically constructs databases, and presents query results in a tabular form, ordered by distance from the query.



CAMUS is not the only interface to audioDB's functionality; others, specialized to **different tasks and user communities**, are in development or have already been described (*e.g.* [6,7]).

Results

Ranked retrieval lists from CAMUS were compared against manually-annotated ground truth for each query to establish **precision** and **recall** performance of the system on this retrieval task, summarized in the table below.

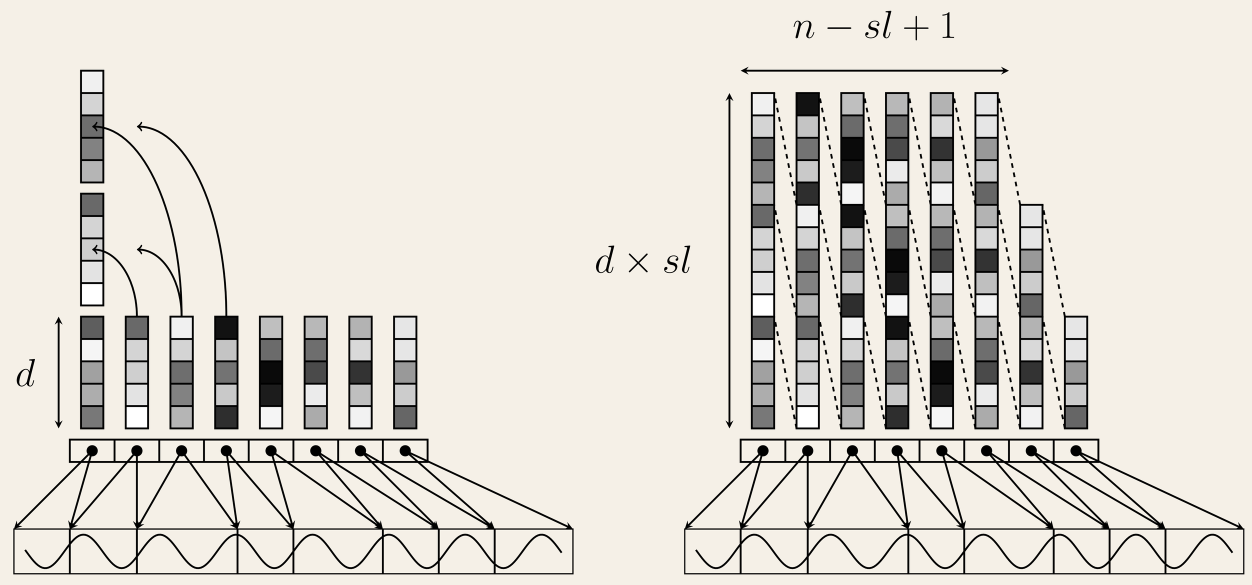
Feature	Precision	Recall	F-Score
MFCC	0.89	0.29	0.44
Constant-Q	**1.00**	0.57	0.73
Chromagram	0.97	**0.83**	**0.89**

Perhaps unsurprisingly given the invariances that we wish to abstract over [5] in this search task, even very simple (FFT-based) chromagram features perform well, reaching a balanced F-Score of 89%. Note that because the audioDB engine is agnostic to the feature being searched over, **better-designed feature variants** can be used to improve retrieval performance if the variant is applicable to a particular task [4].

The principal cause of failure to retrieve relevant results in our corpus arose because of the use of the **sustain pedal** in the query fragment. This relationship of retrieval failure to performance characteristic pointed to a distinction in performance style previously unknown to our amateur improvising pianist.

audioDB

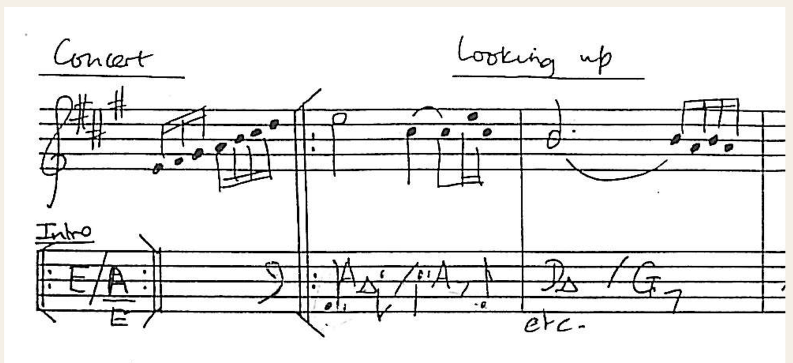
audioDB is a **content-based similarity search** engine, capable of matching on **time series** [3] of arbitrary content-related features, provided a suitable distance measure can be defined. Sequences of features are concatenated to form features in a higher-dimensional space:

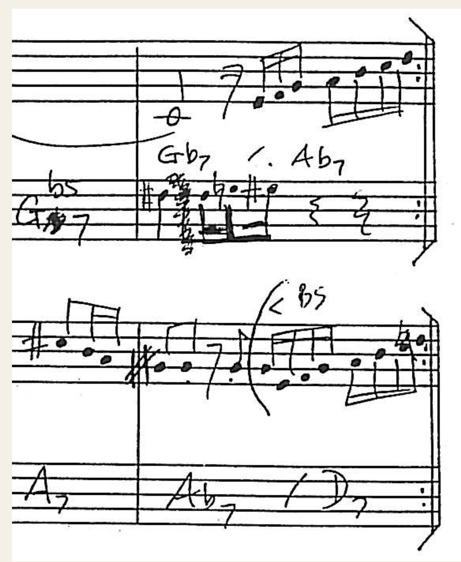


For certain distance measures (*e.g.* Euclidean, Manhattan), we can **probabilistically index** the database of feature vectors in such a way that investigations can scale to very large commercial-scale or **Internet-scale collections** of audio. We have also developed a simple statistical model for thresholding judgments [2], allowing both **automatic relevance cut-off** at small collection scales and **automatic selection of indexing parameters** at large collection scales [8].

Experiment

Our corpus contained several recordings of *Looking Up* by Michel Petrucciani, performed in a variety of ways: in professional and amateur contexts, as a solo performance or in ensemble, on electric and acoustic pianos, and recorded in studio or by a laptop's internal microphone. In addition, the corpus contained recordings of other tracks (*Ambleside Days* by John Taylor and *My Romance* by Rogers and Hart).





Audio Features

We extracted a number of simple audio features for each track in our corpus using multiple feature extractors: both **fftExtract**, intended to cover MPEG-7 Audio features, and the Chromagrams and MFCCs from the **QM Vamp plugins**. Using these features, we then searched for motifs present in *Looking Up* within our database.

References

[1] C. Cannam and others. Linked Data and You: Bringing music research software into the Semantic Web. *Journal of New Music Research*, accepted.

[2] M. Casey, C. Rhodes and M. Slaney. Analysis of Minimum Distances in High-Dimensional Musical Spaces. *IEEE TASLP*, 16(5):1015–1028, 2008.

[3] M. Casey and M. Slaney. The Importance of Sequences in Music Similarity. In *Proc. ICASSP*, vol. V, pp. 5–8, 2006.

[4] T. Crawford, M. Mauch and C. Rhodes. Recognizing Classical Works in Historical Recordings. To appear in *Proc. ISMIR*, 2010.

[5] K. Lemström and G.A. Wiggins. Formalizing invariances for content-based music retrieval. In *Proc. ISMIR*, pp. 591–596, 2009.

[6] M. Magas, M. Casey and C. Rhodes. mHashup: fast visual music discovery via locality-sensitive hashing. In *ACM SIGGRAPH 2008 New Tech Demos*, p. 1, Los Angeles, 2008.

[7] M. Magas and J. Wood. A More User-Centric Approach to the Retrieval of Music Data. *Journal of New Music Research*, in review.

[8] C. Rhodes and others. Investigating Music Collections at Different Scales with AudioDB. *Journal of New Music Research*, accepted.

EPSRC
Engineering and Physical Sciences Research Council

Goldsmiths
UNIVERSITY OF LONDON