A PROTOTYPICAL SERVICE FOR REAL-TIME ACCESS TO LOCAL CONTEXT-BASED MUSIC INFORMATION

Frank Kurth, Meinard Müller, Andreas Ribbrock, Tido Röder, David Damm, and Christian Fremerey
University of Bonn, Germany
Department of Computer Science III

ABSTRACT

In this contribution we propose a generic service for real-time access to context-based music information such as lyrics or score data. In our web-based client-server scenario, a client application plays back a particular (waveform) audio recording. During playback, the client connects to a server which in turn identifies the particular piece of audio as well as the current playback position. Subsequently, the server delivers local, i.e., position specific, context-based information on the audio piece to the client. The client then synchronously displays the received information during acoustic playback. We demonstrate how such a service can be established using recent MIR (Music Information Retrieval) techniques such as *audio identification* and *synchronization* and present two particular application scenarios.

Keywords: Music services, context-based information, fingerprinting, synchronization.

1. INTRODUCTION

The last years have seen the development of several fundamental MIR techniques such as audio fingerprinting [3, 4], audio identification [1], score-based retrieval, or synchronization of music in different formats [2, 6, 7]. Besides the development of tools for basic retrieval tasks, the importance of using feature-based representations for exchanging *content-based music information* (i.e., any information related to the content of the raw music data) over the internet has been recognized recently [8]. As an important example, compact noise-robust audio fingerprints may be used to precisely specify a playback position within a particular piece of PCM audio [4].

As the online distribution of audio documents evolves, there is an increasing demand for advanced MIR services which are able to provide content-based information as well as metadata related to particular music documents. While there are already various services and resources on the internet providing *global* information such as lyrics or scores for a particular piece of music, there is still a lack of services providing *local* information for small excerpts of a given piece of music. However, such local *context-based information* (e.g., information related to a local time interval) can be of great value to a user while listening to a piece of music. Examples of applications incorporating local context-based information include score following, lyrics following, karaoke, or the online-display of translations or commentaries.

In this contribution, we propose a generic framework which allows users to access and exchange context-based (local) information related to particular pieces of audio. We demonstrate the feasibility of our framework by presenting two services, one providing context-based lyrics, the other providing score information.

2. GENERIC FRAMEWORK

The generic scenario of the proposed service consists of a *preprocessing phase* and the *runtime environment*.

In the preprocessing phase, we start with a given data collection of PCM audio pieces. For each of these audio pieces, we assume the existence of a particular type of additional, context-based information such as the lyrics in case of pop-songs or score information in case of classical music. The preprocessing phase consists of two parts. In the first part, we create a fingerprint database (FPDB) using the raw audio material. Employing fingerprinting techniques such as [4], the FPDB allows us to precisely identify a particular (short) excerpt taken from any audio piece within the collection. The identification provides us with the respective song ID and the current position of the excerpt within that song. The second part of preprocessing consists of linking the context-based information for each audio piece to the actual time-line of that piece. This amounts to assigning a particular starting position and duration to each basic component, e.g., a single word in the lyrics scenario or a single note in the score scenario. Fig. 1 shows a score-, PCM-, and MIDI-version of the first measures of J.S. Bach's *Aria con variazioni* (BWV 988). The upper part of the figure illustrates the concept of score-PCM synchronization where a link between a symbolic note event and its corresponding physical realization is indicated by an arrow. Below, a corresponding illustration is given for a MIDI-PCM synchronization. Technically, the linking may be performed by specialized *synchroniza-*

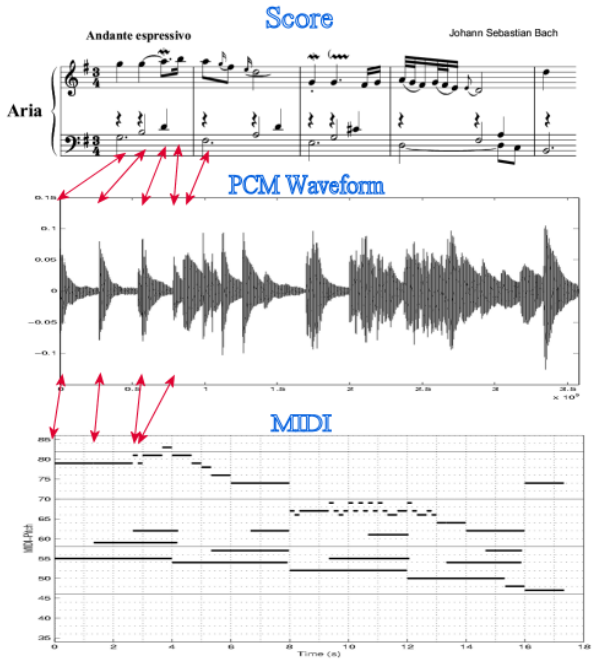

Figure 1. Synchronizing score to PCM (top) and MIDI to PCM (bottom). Corresponding events are linked by red arrows.

tion algorithms such as [2] in the score-PCM scenario. As a result we obtain a so-called *sync file* which allows us to access the available context-based information for each time interval of a particular piece of audio. Both the FPDB and sync files are stored on the server.

At runtime, we assume that a user is equipped with a client application which basically consists of an extended audio player. The user selects a particular audio piece for playback, which then undergoes a local fingerprint extraction at the current playback position. The extracted fingerprint is transmitted to the server, which then tries to identify the audio piece and current playback position. If successful, the server retrieves the context-based information from the sync file that is available for the current playback position. Using a suitable communication protocol, the information is then transferred to the client. During playback the client displays the information synchronously with the acoustic signal. Note that the actual visualization depends on the particular type of application. For example, in a karaoke-like scenario one would probably want to display entire lines of lyrics in advance and additionally highlight single words while they are actually sung.

During subsequent playback, the client may request any further context-based information on the identified piece of audio by simply specifying a target time interval. The server then retrieves all available data for that time interval from the sync file. Note that when using a suitable synchronization protocol between client and server, a repeated fingerprint-based identification of the current playback position is not necessary.

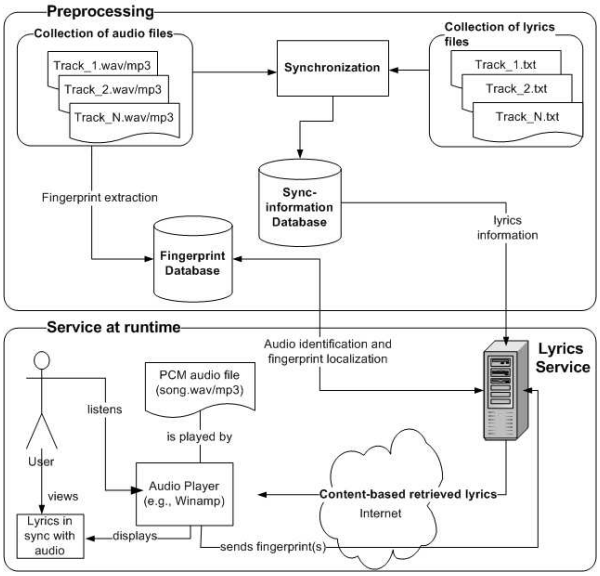

Figure 2. Service offering context-based lyrics.

3. APPLICATIONS: PROTOTYPICAL SERVICES

3.1. A Lyrics Service

Fig. 2 shows an overview of a corresponding service for real-time delivery of context-based lyrics information. In our test setting the FPDB is created using a previously proposed audio fingerprinting technique [4]. Sync files are created from a corresponding collection of lyrics for all of the songs contained in our audio collection. As there are currently no general approaches to automatic PCM-lyrics synchronization available, this step is carried out manually. For the client application we implemented an enhanced audio player capable of fingerprint extraction, client-server communication, and synchronous display of lyrics during playback.

In the following we discuss some issues on the implemented system. Our client-server implementation is `Java`-based using a `C++`-library for fingerprint extraction, index creation and audio identification. The server implements a scheduler component which assigns a task ID to each incoming client request, hence facilitating multiple concurrent requests. During playback, the client performs fingerprint extraction and transfers the fingerprints to the server which in turn performs the task of audio identification. Following a successful identification, the client is allowed to query the server for the desired context-based metadata by specifying a target time interval. Note again that our audio identification technique [4] yields the exact offset between the queried fragment and the original audio signal, hence allowing us to precisely synchronize time-offsets between the client- (query) and server-side (original) audio material. The communication between client and server is performed using `Java`'s RMI (remote method invocation) mechanism and data access is based on the assigned task IDs.

The client system consists of the generic `SyncPlayer`,

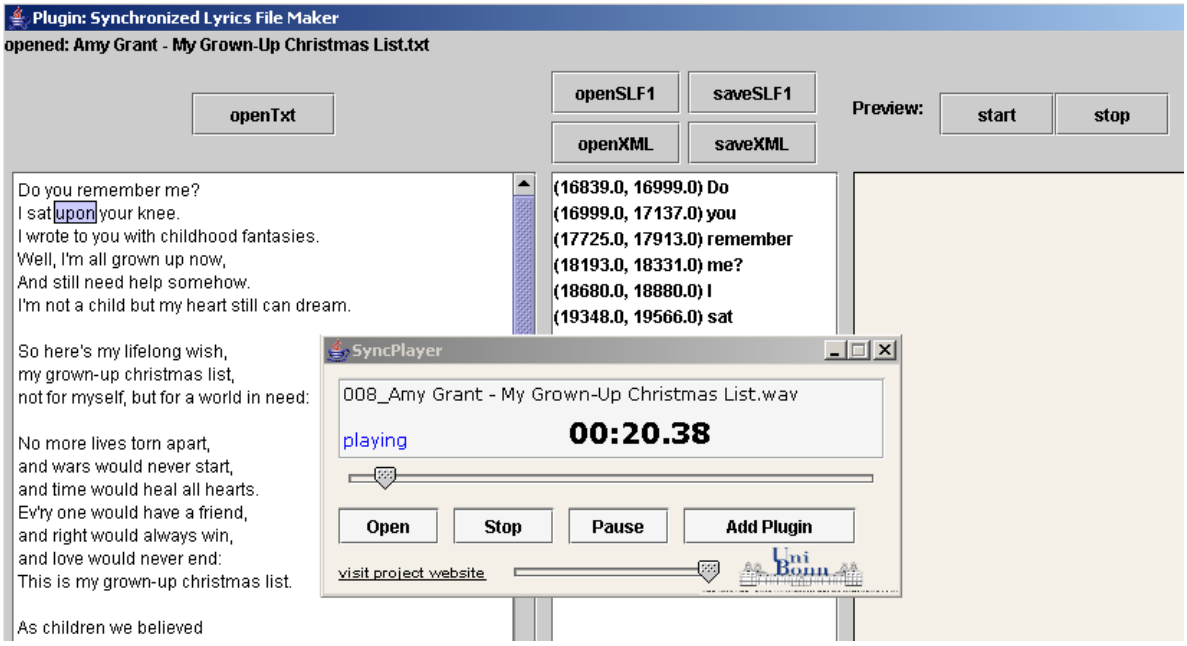

Figure 4. `SyncFileMaker` plug-in during manual synchronization of lyrics (left column, the current word position is highlighted) to a simultaneously played audio track. The center column contains manually specified word positions, the right column is used for previewing the results of the manual annotation.

Figure 3. Generic `SyncPlayer` (top) during playback of *My Grown-Up Christmas List* (by Amy Grant). The bottom part of the figure shows the `Lyrics Display` plug-in highlighting current text lines and word positions simultaneously to the audio playback.

```
<?xml version="1.0" ?>
<SyncFile>
  <Header>
    <Type>Lyrics</Type>
    <SamplesPerUnit>500.0</SamplesPerUnit>
    <SamplingFrequency>44100</SamplingFrequency>
    <Tracks>1</Tracks>
  </Header>
  <Body>
    <Track>
      <Description>Track Number One</Description>
      <Events>
        <Event>
          <Start>1509</Start>
          <Duration>17</Duration>
          <Data>Do</Data>
        </Event>
        ...
      </Events>
    </Track>
  </Body>
</SyncFile>
```

Figure 5. Skeleton of a sync file. The actual synchronization information is contained in the `<Event>`-Tags.

which currently supports basic playback of `WAV`- (44.1 kHz, 16 bit, stereo) and `MP3` audio files.

The `SyncPlayer` offers a plug-in concept for an application specific display of metadata. This concept is realized as a `Java` interface. To display lyrics, we implemented a `Lyrics Display` plug-in which is depicted in Fig. 3 (bottom) along with the `SyncPlayer` (top). During playback, the context-based metadata is transferred to all of the activated plug-ins, which are in turn responsible for an appropriate visualization. As an example, the `Lyrics Display` visualizes a local section of lyrics and highlights the current rows of text as well as the currently sung words (Fig. 3, bottom).

To facilitate manual creation of sync files in the lyrics scenario, we implemented a `SyncFileMaker` plug-in, see Fig. 4. This plug-in may be used to generate sync files by simply pressing a key on a computer keyboard at each word position during playback of a song, thus generating a list of time stamps which may then be exported to a sync file. Sync files are realized using a simple XML-style format, see Fig. 5 for a small example. Note that our implementation and sync file format conceptually supports multiple tracks, which may be used to store different types of metadata. Time stamps within the sync files are related to sample positions using the `<SamplesPerUnit>` tag. This allows us to employ the same time resolution as used by the fingerprinting algorithm: fingerprinting is usually performed by a sliding window technique, resulting in a decrease in time resolution by a certain factor.

The `SyncPlayer` and the `LyricsPlayer` plug-in are available for download at our website

`http://www-mmdb.iai.uni-bonn.de/research.php`

by following the corresponding link in the *Demos* section. Currently, there is a small collection of about ten songs available for demonstrating the lyrics service.

While the time for fingerprint extraction may be neglected, the delay between start of audio playback and display of context-based lyrics mainly depends on the quality of the internet connection, the server load, and the time required for querying the FPDB. As we currently transfer all of the available lyrics information for one piece of audio to the client directly after identification, there is no additional latency besides this initial delay, even when skipping through the audio piece during playback. Note that additional delays may occur when playing back `MP3` audio tracks which are due to the used `MP3` software library.

3.2. A Service for Score-Based Information

As a second scenario we propose a prototypical service for providing context-based score information during playback. In contrast to the former lyrics service, in this scenario we use a novel algorithm for score-PCM synchronization [5] to *automatically* generate the sync files. Furthermore, the karaoke-like visualization is replaced by a score-following type of display, highlighting the current score positions during playback. A corresponding plug-in for our `SyncPlayer` is currently under development and will be available from our website upon completion.

4. CONCLUSIONS AND ONGOING WORK

We presented a web-based client-server scenario for real-time access to context-based local music information such as lyrics or score-related data. The two proposed prototypical services can be realized using recent MIR techniques and are a first step towards bridging the gap between widely available global music information and the increasing demand for selective access to local context-based information. Obviously, the proposed generic framework has various further applications such as synchronous display of translations, commentaries, or instrument specific music information like guitar chords.

Part of our future work will be concerned with completing, refining and evaluating the proposed services. In this we are particularly interested in possible application scenarios within existing as well as emerging digital libraries. Another important challenge will be the design of new methods for automatically synchronizing metadata to audio signals. While there have been recent advances allowing for resonable synchronizations of score-like data to polyphonic recordings for a limited class of instruments, those techniques are not yet applicable to arbitrary pieces of music. Furthermore, the automatic synchronization of lyrics to audio recordings is a challenging field of research and will require, e.g., advanced algorithms for detecting vocal passages in audio recordings.

5. REFERENCES

[1] Eric Allamanche, Jürgen Herre, Bernhard Fröba, and Markus Cremer. AudioID: Towards Content-Based Identification of Audio Material. In *Proc. 110th AES Convention, Amsterdam, NL*, 2001.

[2] Vlora Arifi, Michael Clausen, Frank Kurth, and Meinard Müller. Automatic Synchronization of Musical Data: A Mathematical Approach. In Walter B. Hewlett and Eleanor Selfridge-Fields, editors, *Computing in Musicology*. MIT Press, in press, 2004.

[3] Pedro Cano, Eloi Battle, Ton Kalker, and Jaap Haitsma. A Review of Audio Fingerprinting. In *Proc. 5. IEEE Workshop on MMSP, St. Thomas, Virgin Islands, USA*, 2002.

[4] Frank Kurth, Michael Clausen, and Andreas Ribbrock. Identification of Highly Distorted Audio Material for Querying Large Scale Data Bases. In *Proc. 112th AES Convention, Munich, Germany*, 2002.

[5] Meinard Müller, Frank Kurth, and Tido Röder. Towards an Efficient Algorithm for Automatic Score-to-Audio Synchronization. In *International Conference on Music Information Retrieval, Barcelona, Spain*, 2004.

[6] Ferréol Soulez, Xavier Rodet, and Diemo Schwarz. Improving polyphonic and poly-instrumental music to score alignment. In *International Conference on Music Information Retrieval, Baltimore*, 2003.

[7] Robert J. Turetsky and Daniel P.W. Ellis. Force-Aligning MIDI Syntheses for Polyphonic Music Transcription Generation. In *International Conference on Music Information Retrieval, Baltimore, USA*, 2003.

[8] George Tzanetakis, Jun Gao, and Peter Steenkiste. A Scalable Peer-to-Peer System for Music Content and Information Retrieval. In *International Conference on Music Information Retrieval, Baltimore*, 2003.