Automatic Synchronization of Music Data in Score-, MIDI- and PCM-Format

Vlora Arifi, Michael Clausen, Frank Kurth, and Meinard Müller
University of Bonn, Germany – Department of Computer Science III
{vlora, clausen, frank, meinard}@cs.bonn.edu, http://www.mmmdb.iai.uni-bonn.de

1 Problem Setting: Synchronization

Synchronization task: Given a (time-) position in some representation of a piece of music (e.g., in score or MIDI format), determine the corresponding position within some other representation (e.g., given in PCM-format).

Example: Consider scores, PCM (waveform), and MIDI versions of the first 16 measures of the Aria cur Varievorsam by J.S. Bach, BWV 988.

In this poster, we shall only consider Score-to-PCM (SP-) synchronization.

Applications:
- Automatic annotation of a piece of music available in different data formats as a basis for content-based retrieval.
- Usage of link structures to access PCM audio pieces accurately after score-based music retrieval.
- Investigation of agogic and tempo studies.
- Automatic tracking of the score positions during a performance.

2 Overview: Synchronization Framework

Synchronization proceeds in three steps:
1. Extract note parameters from the PCM (Section 3).
2. Preprocess score to normalized representation (Section 4).
3. Synchronize extracted note parameters and score (Section 5.3) w.r.t. a suitable cost function (Section 5.2).

3 Extraction of Note Parameters

Onset detection:
- Track changes of signal’s frequency contents over time using novelty curves.
- Define time-resolution of resulting estimated onset positions using linear prediction (following Fuster et al. (1992)).

Pitch extraction:
- Subband analysis using tree-structured multi-rate filterbank adapted to musical scale (at most one note per subband) following Bobosh et al. (1998).
- Establish note positions in subbands using detected onset positions.
- Estimate pitches using template-matching.

Example: Novelty curve recovered from the PCM version of the Aria:

In the following diagram, the red arrows link the corresponding events of the different versions. Based on these three typical music representations, we may consider the following types of synchronization tasks:

Applications:
- Automatic annotation of a piece of music available in different data formats as a basis for content-based retrieval.
- Usage of link structures to access PCM audio pieces accurately after score-based music retrieval.
- Investigation of agogic and tempo studies.
- Automatic tracking of the score positions during a performance.

4 Data Modeling

After suitable (time-) quantization, we distinguish two types of score-based note-objects:
1. Explicit notes: all time- and pitch parameters are given explicitly.
2. Implicit notes: notes with special properties, e.g., trill or appoggio

Example: Two implicit notes, appoggio and trill (left), possible realizations (center); implicit notes modeled by fuzzy notes (right).

5 Synchronization Algorithm

5.1 Score-PCM Matches

Assume that score and \( \Delta \)-quantized extracted PCM-data are given by the sets:
\[
S = \{s_1, s_2, s_3, \ldots, s_m\}
\]
and
\[
\Delta \text{PCM} = \{p_1, p_2, \ldots, p_n\}
\]
where \( S \) contains all notes, \( \Delta \text{PCM} \) the \( \Delta \)-quantized notes (MP=0), \( i \in [m] \), \( j \in [n] \) and \( \Delta = 50 \) (also for \( \Delta \)-quantized notes).

Cost function:
\[
C_P^\text{SP} (s_i, p_j) = \begin{cases} 
0 & \text{if } (s_i, p_j) \in \text{match} \\
\text{cost}(s_i, p_j) & \text{otherwise}
\end{cases}
\]

Determine cost-minimizing SP-match using dynamic programming.

5.3 Cost-Optimal SP-Matches

Determine cost-minimizing SP-match using dynamic programming. Recursively define a matrix \( C = (c_{ij}) \) with \( i \in [0, m] \) and \( j \in [0, n] \):
1. Initial condition:
   \[
   c_{0j} = \begin{cases} 
   0 & \text{if } j \leq 1 \\
   \infty & \text{otherwise}
   \end{cases}
   \]
2. Recursion:
   \[
   c_{ij} = \min\left\{ c_{i-1,j-1} + \text{cost}(s_i, p_j), c_{i-1,j}, c_{i,j-1} \right\}
   \]

5.2 Cost Functions For SP-Matches

Definition 5.2. Let \( \rho = (\alpha, \beta, \gamma, \delta, \Delta) \in \mathbb{R}^5_+ \) be a parameter vector. Then:
\[
\text{if } e_i \text{ is an } \alpha \text{-note in } S \text{ and } e_j \text{ is an } \alpha \text{-note in } \Delta \text{PCM},
\]
\[
C^\text{SP} (s_i, \Delta \text{PCM}) = \begin{cases} 
\alpha & \text{if } e_i = e_j \\
\infty & \text{otherwise}
\end{cases}
\]

6 An Example

Note-scores for the Aria (left) and quantized note parameters \( \Delta \text{PCM} \) extracted from a PCM version (right).

Corresponding cost matrices:

Cost matrix \( C \) and a cost-minimizing SP-match (for first 16 measures):

7 Conclusions

- Tests on a variety of classical polyphonic piano pieces (lengths 10 – 60 s) played on various instruments yield good results.
- Crucial choice of parameter vector \( \rho \) in cost function.