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

Vlora Arifi, Michael Clausen, Frank Kurth, and Meinard Müller
Universität Bonn, Institut für Informatik III
Römerstr. 164, D-53117 Bonn, Germany
{vlora, clausen, kurth, meinard}@cs.uni-bonn.de

Abstract

In this paper we present algorithms for the automatic time-synchronization of score-, MIDI- or PCM-data streams representing the same polyphonic piano piece.

1 Introduction

Modern digital music libraries consist of large collections of documents containing music data of diverse characteristics and formats. For example, for one and the same piece of music, the library may contain the corresponding score in Capella or Score format, some MIDI-files, and several interpretations in form of CD recordings. Inhomogeneity and complexity of such music data make content-based browsing and retrieval in digital music libraries a difficult task with many yet unsolved problems. One important step towards a solution are synchronization algorithms which automatically link data streams of different data formats representing a similar kind of information. In particular, in the framework of audio by *synchronization* we mean some procedure which, for a given position in some representation of a given piece of music (e.g., given in score format), determines the corresponding position within some other representation (e.g., given in PCM-format).

Such synchronization algorithms have applications in many different scenarios: following some score-based music retrieval, linking structures can be used to access some suitable audio CD accurately to listen to the desired part of the interpretation. A further application is the automatic annotation of a piece of music in different data formats as a basis for content-based retrieval. As another example, musicologists can use synchronization algorithms for the investigation of agogic and tempo studies. Furthermore, temporal linking of score and audio data can be useful for automatic tracking of the score positions during a performance.

In our work, we concentrate on three representative data formats used for music data: the symbolic *score format*, the physical *PCM-format* and the *MIDI-format* which may be thought of as a hybrid of the last two data formats. We have developed synchronization algorithms for two data streams in any of these three data formats which will be referred to as Score-to-MIDI (SM) synchronization, Score-to-PCM (SP) synchronization, and MIDI-to-PCM (MP) synchronization. Especially, SP- and MP-synchronization constitute a difficult problem since the waveform-based PCM-format does not contain any explicit information on the notes. Therefore, in these cases some score-like parameters such as onset times and pitches have to be extracted from the PCM-data prior to the actual synchronization.

Due to space limitations, we are not able to give an overview of the related work. Links to the relevant literature can be found in the full paper version Arifi et al. (2003a) available on our website or in Arifi et al. (2003b).

2 Feature Extraction

In this section we summarize our system for extracting note parameters from the PCM-data stream. Using several established tools from audio signal processing, our main contributions are a refined template matching algorithm for polyphonic pitch extraction and a two-step algorithm for note onset detection.

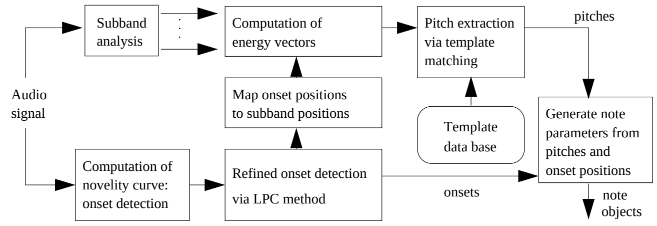

Figure 1: Diagram of the feature extraction algorithm.

Figure 1 shows the overall feature extraction algorithm. Similar to Bobrek et al. (1998) an input PCM-signal is transformed to a subband representation using a multirate filter bank. Simultaneously, a two-stage peak-picking algorithm detects probable note onset positions. According to those onset positions, the subband representation is split into time intervals. For each interval, we calculate an energy vector with components corresponding to the subbands: each component contains the total energy within the interval of the respective subband. Then, for each energy vector a pitch extraction based on a template matching algorithm is performed. The pitch extraction yields a set of notes for the corresponding time interval. The feature extraction algorithm outputs a note object for each note in each time interval, where a note object consists of an onset time and a pitch information. For implementation details of the overall system we refer to Arifi (2002).

3 Synchronization Algorithms

In this section we describe the actual synchronization algorithms. Here, we only consider the case of a score- and a PCM-data stream (SP-synchronization). The other cases such as SM- or MP-synchronization are even easier or can be done in a similar fashion (see Arifi (2002)).

We first preprocess the score data stream where we distinguish between two kinds of note objects: *explicit* and *implicit* notes. For *explicit* objects all note parameters such as measure, beat, duration, and pitch are given explicitly. In view of the synchronization algorithm we only use the musical onset time and pitch. We represent each explicit note object by a tuple $(e, p) \in \mathbb{Q} \times [0 : 127]$, where the rational number e encodes the onset time and the integer p the pitch of the corresponding note. By an *implicit* note object we understand notes or a group of notes with some additional specification such as a trill, an arpeggio or grace notes. Implicit objects allow different realizations. To get this ambiguity under control we introduce the concept of a *fuzzy note*, which is defined to be a tuple (e, H) consisting of some onset time $e \in \mathbb{Q}$ and some set of alternative pitches $H \subset [0 : 127]$. In this model, the preprocessed score is given by some subset $S \subset \mathbb{Q} \times 2^{[0:127]} \times 2^{[0:127]}$, where $2^{[0:127]}$ denotes the set of all subsets of $[0 : 127]$. Here, in a triple $(e, H_0, H_1) \in S$ the subset $H_0 \subset 2^{[0:127]}$ consists of all pitches of explicit note objects having musical onset time e and similarly the subset $H_1 \subset 2^{[0:127]}$ consists of all pitches of implicit note objects.

Next, the extracted note parameters from the PCM data stream (see Section 2) are further preprocessed by quantizing the onset time candidates with some suitable chosen quantizer resolution $\Delta > 0$. The resulting data stream is denoted by P_Δ which can be regarded as a subset of $\mathbb{Q} \times 2^{[0:127]}$. Note that in the PCM-case there are only explicit note objects.

Altogether, we may assume that the score and the Δ-quantized extracted PCM-data are given by the sets $S = [(s_1, S_{01}, S_{11}), \ldots, (s_s, S_{0s}, S_{1s})]$ and $P_\Delta = [(p_1, P_{01}), \ldots, (p_p, P_{0p})]$. Here, the s_i, $1 \leq i \leq s$, denote the musical onset times and the p_j, $1 \leq j \leq p$, the quantized physical onset times. Furthermore, $S_{0i}, S_{1i}, P_{0j} \subset [0 : 127]$ are the respective sets of pitches for the explicit and implicit objects.

We now accomplish the SP-synchronization by matching the sets S and P_Δ in the following sense.

Definition 3.1. A *score-PCM-match* (SP-match) of S and P_Δ is defined to be a partial map $\mu \colon [1 : s] \to [1 : p]$, which is strictly monotonously increasing on its domain satisfying $(S_{0i} \cup S_{1i}) \cap P_{0\mu(i)} \neq \emptyset$ for all $i \in \mathrm{Domain}(\mu)$.

This definition needs some explanations. The fact that objects in S or P_Δ may not have a counterpart in the other data stream is modeled by the requirement that μ is only a partial function and not a total one. The monotony of μ reflects the requirement of faithful timing: if a note in S precedes a second one this also should hold for the μ-images of these notes. Finally, the requirement $(S_{0i} \cup S_{1i}) \cap P_{0\mu(i)} \neq \emptyset$ prevents that onset times are linked which are completely unrelated with respect to their pitches.

Obviously, there are many possible SP-matches between S and P_Δ. However, introducing a cost function makes different matches comparable. For an explicit formula of our cost function, which is crucial in view of the quality of the solution for the synchronization problems, we have to refer to Arifi et al. (2003a). Summarizing, our cost function penalizes non-matched explicit and implicit note objects in S (where the weighting of the penalty depends on the type of note objects), non-matched note-objects in P_Δ, as well as large relative onset time deviations thus preventing large global deviations in the synchronization.

The solution of the SP-synchronization is then realized by an optimal SP-match minimizing the cost function. This optimal match can be efficiently computed by means of dynamic programming.

4 Experimental Results and Conclusions

We have implemented a prototype of the extraction algorithms from Section 2 and the synchronization algorithms in the MATLAB programming language and tested our algorithms for SM-, SP-, and MP-synchronization on a variety of classical polyphonic piano pieces of different complexity and length (ranging from 10 to 60 seconds) played on various instruments. Furthermore, we have systematically generated a library of more than one hundred test pieces both in MIDI- and PCM-format played on a MIDI-piano, a Steinway grand piano, and a Schimmel piano. In some of those pieces our performer has deliberately built in excessive accelerandi, ritartandi, rhythmic distortions, and wrong notes. Even in these extreme situations, where one unsurprisingly has many “erroneously” extracted note objects which considerably differ from the score-data, our SP-synchronization algorithm resulted in good overall global matches which are sufficient for the applications mentioned in the introduction. Even more, in case of rather accurate extracted note parameters our synchronization algorithms could resolve subtle local time variations in some interpreted version of the piano piece. For further details and results of our experiments we refer to Arifi (2002).

References

Arifi, V. *Algorithmen zur Synchronisation von Musikdaten im Paritur-, MIDI- und PCM-Format*. PhD thesis, Universität Bonn, Institut für Informatik (2002). Retrievable from `http://hss.ulb.uni-bonn.de:90/ulb_bonn/diss_online/math_nat_fak/2002/arifi_vlora/`

Arifi, V., Clausen, M., Kurth, F., Müller, M. (2003a). *Automatic Synchronization of Music Data in Score-, MIDI-, and PCM-Format*. Technical Report, Universität Bonn, Institut für Informatik. Retrievable from `http://www-mmdb.iai.uni-bonn.de/eng-public.html`

Arifi, V., Clausen, M., Kurth, F., Müller, M. (2003b). Automatic Synchronization of Music Data. To appear in *Computing in Musicology*, MIT Press.

Bobrek, M., Koch, D. (1998). Music Signal Segmentation Using Tree-Structured Filter Banks. *Journal of Audio Engineering Society*, 46(5), 412–427.