Automatic Synchronisation Between Audio and Score Musical Description Layers

Antonello D'Aguanno and Giancarlo Vercellesi

Laboratorio di Informatica Musicale (LIM)
Dipartimento di Informatica e Comunicazione (DICo)
Universitá degli Studi di Milano,
Via Comelico 39,
I-20135 Milano, Italy
{daguanno,vercellesi}@dico.unimi.it
http://www.lim.dico.unimi.it

Abstract. This work describes algorithms dedicated to score and audio alignment using the MX / IEEE P1599 format. The format allows description of the score, and management of synchronisation points, linking them with different versions of the performed music. An algorithm is proposed here that allows alignment of an MX score and its execution, coded in PCM format, which produces an output for the MX Spine that contains synchronisation between notes and audio signal. The proposed architecture is based on two different steps: the first deals with the audio level and the extraction of features like pitch, onset and the like, while the second determines the alignment between the features extracted in the first step and the notes present in the MX score. For each step, different algorithms are proposed and discussed, and analysis and comparison of synchronisation capabilities are provided.

Keywords: IEEE P1599, MX, synchronisation, PCM, MIDI.

1 Introduction

Contemporary digital music archives consist of huge collections of heterogeneous documents. For a music piece, an archive may contain corresponding scores in different versions, for example, voice and piano or orchestral, as well as several interpretations, e.g. played by different performers and recorded in diverse formats (CD recordings, MP3, FLAC and so on). The heterogeneity of music information makes retrieval hard to accomplish [1] and as a consequence many problems remain unsolved. One important problem that needs a solution is synchronisation, which requires the implementation of algorithms that automatically link different audio streams of the same piece to symbolic data formats representing the different scores.

In particular, for the Notational and Audio layers in MX, synchronisation means that some algorithms, for a given event in some representation of the music score, are capable of determining the timing of the corresponding audio

B. Falcidieno et al. (Eds.): SAMT 2007, LNCS 4816, pp. 200–210, 2007.

events within an audio representation in some format. After execution of a synchronisation process in MX it is possible, for a given audio event, to find the correct position in the score, in the *Notational* and the *Audio* layers.

As stated in [2]: "'Such synchronisation 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."'This possibility represents a useful tool for music students, who can listen to music audio and, at the same time, see the corresponding notes. The MX spine and MX Music Events are the linking structure described in [2]. Furthermore, the MX linking structure can be useful for musicologists who can use synchronising algorithms to link the interpretation layer and the score layer, and then use MX for the investigation of agogics and tempo. In addition, temporal linking of score and audio data can be useful for automatic tracking of score positions during a performance.

This article is organised as follows. Section 2 presents an overview of score following and synchronisation algorithms proposed in literature. Section 3 describes the ComSi algorithm. Section 4 describes the power of MX for representing audio synchronisation results. Section 5 provides the analysis of results and, finally, conclusions will be summarised in 6.

2 Related Works

Many algorithms have been proposed in literature that deal with synchronisation. The majority of them can be subdivided into two groups: in the first one, audio and score have to be analysed, while the second one requires the realisation of correct links between these two layers ([3][2][4][5][6][7][8]). The algorithms proposed in the literature use several different systems to implement audio analysis, with well-known tools from audio signal processing. For example, [7] uses a Short Time Fourier Transform, in [2] proposes an onset detection followed by pitch detection. In [6] the feature extraction procedure performs these operations: decomposition of the audio signal into spectral bands corresponding to the fundamental pitches and harmonics, followed by the computation of the positions of significant energy increases for each band - such positions are candidates for note onsets. The most popular solution proposed in the literature to select the correct links between audio and score is based on the *template matching technique* ([7][3][8]). Such algorithms build a MIDI score to obtain a template of the real execution, which is compared to the real audio using a DTW[1] programming technique [9]. The correct synchronisation is then obtained from the difference between the agogics of the real execution and that of MIDI. The algorithm described here uses a different approach based on recursive research, as explained in see section 3.3).

Actually, the unsolved problems of synchronisation are not limited to finding a suitable algorithm. There is at least one more open question: once the correct

[1] Dynamic Time Warping is a technique for aligning time series that has been well known in the speech recognition community since the 1970s.

synchronisation has been obtained, how can this result generalised to use it in other applications? The answers proposed in the literature are trivial and not sufficient to allow interoperability or future utilisation. For example, the last MIREX[2] task, focused on synchronisation, used a text file to bridge audio and score, by letting every line in its text file connect a MIDI event with an audio instant. While this solution meets experimental needs, it is not suitable for commercial applications. The algorithm proposed in section 3 uses the features of MX to solve this problem (See section 4).

3 The COMSI Algorithm

This algorithm proposed here is able to synchronise a MX music score with one or more PCM (Pulse Code Modulation) audio executions. It is recursive, and consists of three different phases (figure 1):

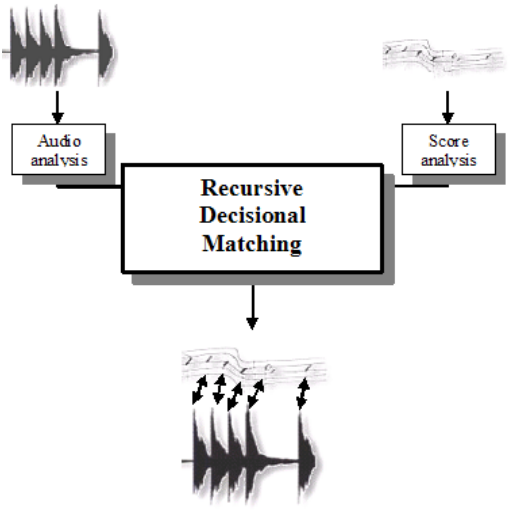

Fig. 1. The general overview of the COMSI algorithm

- in the first phase, the MX score is read in order to extract all relevant musical events. This algorithm is also able to read MIDI files.
- in the second phase, the PCM audio signal is analysed to identify all possible musical notes
- in the third and last phase, a decisional matching is performed to put relate the event at the score level with the same event at the audio level. The decisional method is based on the attack time of notes extracted at audio level.

Each phase is described in the following sections.

[2] Music Information Retrieval Evaluation eXchange is a community-based endeavor which involves many researchers active in MIR field.

3.1 The MX Score Analysis

In this phase, the score is read to extract the musical events that are related to the audio signal by means of decisional matching. For each measure, only the notes with strong accent are selected, because it is easier to recognise them at audio level, when the audio signal is analysed. For every strong accent, the verticalisation of the score at that point is computed. An example is shown in the table below (1) where fundamental frequencies are referred to notes in the score (i.e. 440 Hz corresponds to A):

Table 1. This table contains the time described as measures and events and the fundamental frequencies related to the note known at that time. The algorithm is able to perform these operations on MIDI scores as well.

Measures	Events	Fundamental Frequencies (Hz)
1	1	1047, 311, 415, 523, 1109, 988
1	2	784, 1175, 1568, 294, 440, 466, 698, 1397
2	1	523, 156, 208, 262, 554, 494
2	2	392, 587, 784, 147, 220, 233, 349, 698

3.2 The PCM Audio Analysis

A recursive, second-order notch filter bank is used to perform audio signal analysis. The aim of this phase is the detection of each attack-time note represented at the symbolic level in the score. For each verticalised chord extracted from the score (see table (1)), the audio signal is filtered with a notch filter centred on the frequency of the note examined, as shown in figure 2.

Consider note A at 440Hz. While (a) shows its waveform signal, this signal is filtered with the notch filter with center frequency 440Hz (b), which yields a filtered signal with high energy. Note that if the signal is filtered at different centre frequencies, the signal will contain less energy: see the cases for centre frequency at 220Hz (c) and 880Hz (d).

The bandwidth of the notch filter is 10 Hz. This value has been selected because it covers the whole frequency bandwidth, starting from the MI at 82,406889 Hz (first octave) and avoiding superposition between adjacent semitones (figure 3).

The audio signal in time-windows is split at 100ms. Thanks to the notch filter bank, the energy of each time-window can be computed. For each note, a possible attack-time is considered to be the audio segment which has an energy value above a convenient relative threshold, obtained from the average energy of the filtered signal.

When two or more notes are present in a single musical event, the energies of each single note that composes the analysed musical event are multiplied together. This allows the creation of a pseudo-score that contains, for each note extracted at the score level, the attack-time hypothesis at the audio level. This attack-time hypothesis gets defined as the execution event.

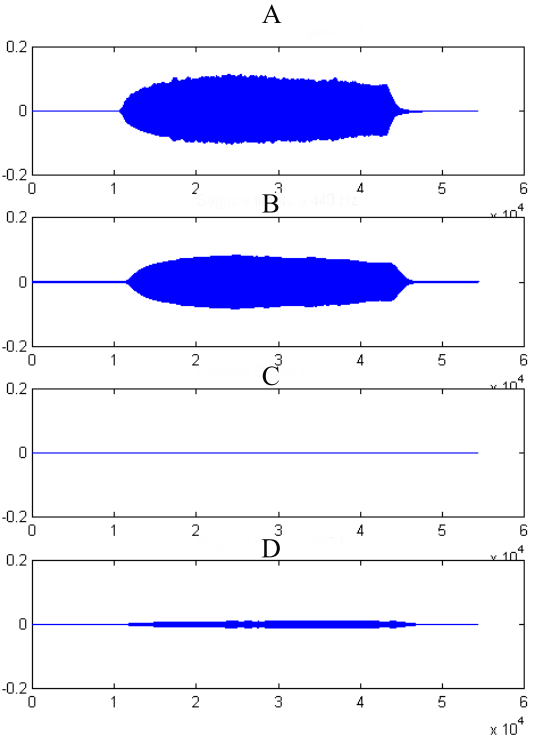

Fig. 2. An example of a 440Hz signal (a) filtered with a notch filter having its centre frequency at 440Hz (b), 220Hz (c) and 880Hz (d)

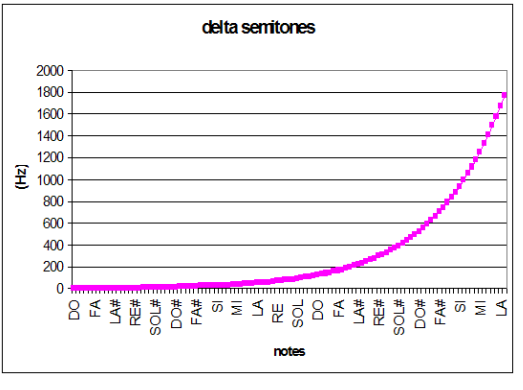

Fig. 3. Differences between two adjacent semitones

3.3 The Decisional Matching

In this phase, the musical event at score level is related to the same one found at audio level. In other words, the attack-time note for each note contained in the score has to be found. It is based on (figure 4):

- the expected attack-time note,
- the relative position into the symbolic score
- the duration of audio signal

This decisional matching is performed in two steps:

1. recursive research of attack-time related to the musical event that corresponds to the first event of each measure in the score. The average duration of each measure is computed as the ratio between the whole duration of the audio signal and the number of measures at score level. Hence, for every event, the expected attack-time is computed and the nearest execution event at audio level is examined. The adjacent measure average duration (right and left intervals) is computed with respect to the expected attack-time. This attack-time is considered valid if the differences between the average time of the left and the right intervals are less than 30% of the expected average duration. Then, all the steps listed above are applied recursively to detect all attack-times.
2. Sequential research of each musical event contained in a measure. The algorithm is similar to the previous one, but in this case the algorithm is sequential. The average duration of quarter notes is examined, with a tolerance of 30% of every average duration, further reduced to 20%.

4 The Linking Structure in MX

In this section, the MX capabilities to represent the synchronisation anchors between audio and score are described (more information about MX can be found in [10][11][12]). In MX, synchronization anchors are saved in three different layers: *Audio*, *Notational*, *Logic*. The `track_event` tag in the *Audio* layer has two attributes: `event_ref`, which contains the identifier for the current event, and `start_time`, which contains the related audio time. In this example this means that in the audio file, event named `vno1_001_01` is present after 30sec.

Similarly, in the *Notational* layer, `graphic_event` tag is present and has five different attributes: `event_ref` the name of the event we considered and the other four attributes which represent the position of the event in a graphic image. The measurement unit of the position is specified by the `measurement_unit` tag (expressed in pixels in this example).

The origin of the axis is in the upper left corner. Every event is represented by a bounding box, and in the Notational layer the co-ordinates of the upper-left corner and lower-right are annotated. Looking further at this example, it appears that the upper-left corner of the event *vno100101* is positioned 529 pixels to the

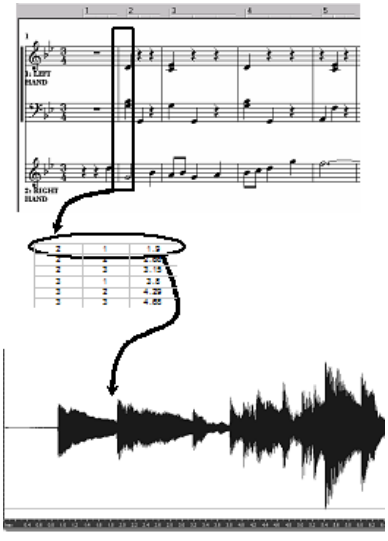

Fig. 4. The schema of the decisional matching phase

right and 109 pixels below the axis origin, while the lower right corner is 555 pixels to the right and 172 pixels below the axis origin. The element spine in the *Logic* layer links graphic event and track event, which have the same *eventref* attribute. Attributes timings and *hpos* have the following meaning:

- `timing` represents the temporal co-ordinate of the event. It describes the time offset from the occurrence of the preceding event. Time is expressed in *Virtual Time Units* (VTU). The measurement unit can be defined according to user needs, but all the numeric values must be integer. Since 0 means simultaneity in this relative system, each event is simultaneous to all those with 0 timing. Besides, if the event itself presents has a timing equal to 0, it is also simultaneous to all the preceding events up to the first event with a non-zero value, that event included.
- `hpos` refers to a virtual horizontal dimension. The value is expressed in a relative spatial unit called *VirtualPiXels* (VPX), which provides reference points for the vertical alignment of different symbols in different voices, parts or staves. The score is viewed in this case from an abstract vantage point (real scores are treated in the Notational layer). As a consequence, staff systems never generate new lines: there is only one staff system, including all symbols on a single line, so that only the horizontal axis is used in establishing vertical alignment.

Example of the MX potential to represent the results of the synchronisation algorithm - AudioLayer

```
<audio>
  <track file_name="nina.wav" encoding_format="audio_wav"
```

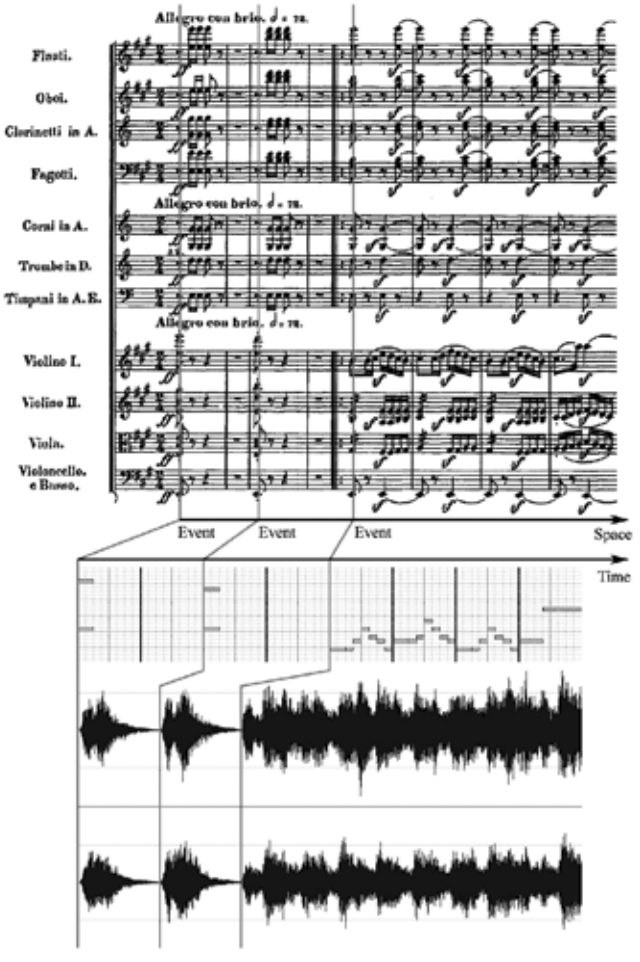

Fig. 5. The role of Spine in MX. In this image a score is shown with the related MIDI sequence and the performance file. The Spine links them all.

```
                                    file_format="audio_wav">
    <track_indexing timing_type="seconds">
      ...
      <track_event start_time="0.30" event_ref="vno1_001_01"/>
      <track_event start_time="0.30" event_ref="vno1_001_02"/>
      ...
      <track_event start_time="12.00" event_ref="vno1_15_03"/>
      <track_event start_time="12.00" event_ref="vno2_15_03"/>
      ...
    </track_indexing>
  </track>
</audio>
```

Example of the MX potential to represent the results of the synchronisation algorithm - Notational Layer

```
<notational>
  <graphic_instance_group description="Manuscript">
    <graphic_instance file_name="M2_0004.jpg" file_format="image_jpeg"
                    encoding_format="image_jpeg" position_in_group="1"
                                          measurement_unit="pixels">
      ...
      <graphic_event event_ref="vno1_001_01" upper_left_x="529"
            upper_left_y="109" lower_right_x="555" lower_right_y="172"/>
      <graphic_event event_ref="vno1_001_02" upper_left_x="557"
            upper_left_y="105" lower_right_x="580" lower_right_y="165"/>
      ...
      <graphic_event event_ref="vno1_15_03" upper_left_x="627"
            upper_left_y="693" lower_right_x="656" lower_right_y="719"/>
      <graphic_event event_ref="vno2_15_03" upper_left_x="846"
            upper_left_y="689" lower_right_x="883" lower_right_y="713"/>
      ...
    </graphic_instance>
  </graphic_instance_group>
</notational>
```

Example of the MX potential to represent the results of the synchronisation algorithm - Spine

```
<logic>
  <spine>
    ...
    <event id="vno1_001_01" timing="12" hpos="12"/>
    <event id="vno1_001_02" timing="0" hpos="0"/>
    ...
    <event id="vno1_15_03" timing="48" hpos="48"/>
    <event id="vno2_15_03" timing="0" hpos="0"/>
    ...
  </spine>
</logic>
```

5 Analysis of Results

Different tests have been performed on the ComSi algorithm in regard to synchronisation of several polyphonic music tracks and MX scores. In these tests, scores have been considered with different time signatures. Synchronisation is considered correct when the ComSi-time of the analysed musical event lies between the real-time of the previous musical event and the next one, as in the formula:

$$previous_{real-timeevent} < ComSi_{synchtime} < next_{real-timeevent} \quad (1)$$

Only the error related to the first event of each measure are reported. This approach has been chosen to simplify the analysis of the results.

The table 2 shows the results: the error columns report the percentage of measure for which the first event of the measure had been confused with the first events of another measure. This two error columns refer to a measure which has a distance of |1| or |2|, respectively from the considered measure. Presently, a new test infrastructure is being developed which will use the MX capabilities to host many different synchronisation methods between the Audio and the Notational layer. This new system will allow intensive tests for many different synchronisation algorithms and will take into account every musical event.

Table 2. Results of ComSi test

Track	Time Signature	Correct Measure	One Error Measure Event	Twice Error Measure Event
Chopin	3/4	66%	24%	10%
Beethoven	4/4	71%	18%	11%
Beethoven (faster)	4/4	69%	16%	15%
Fiore di Maggio	4/4	70%	17%	13%
Cavatina	6/8	43%	48%	9%
Aria	6/8	61%	31%	8%

6 Conclusions

In this paper we have described algorithms dedicated to score and audio alignment using the MX / IEEE P1599 format. This algorithm allows alignment of an MX score and its execution, coded in PCM format, and produces an output for the MX Spine that contains synchronisation between notes and audio signal. The analysis of results 5 show the reliability of our algorithm. Furthermore, the MX standard will allow intensive tests for many different synchronisation algorithms and will take into account every musical event.

References

1. Stephen, J.: Music information retrieval. In: Annual Review of Information Science and technology, ch. 7, Blaise Cronin, Medford, NJ, USA, vol. 37, pp. 295–340 (2003)
2. Arifi, V., Clausen, M., Kurth, F., Muller, M.: Automatic synchronization of music data in score-, midi-and pcm-format. In: ISMIR. Proc. of Intl. Symp. on Music Info. Retrieval (2003)
3. Dannenberg, R.B., Hu, N.: Polyphonic audio matching for score following and intelligent audio editors. In: Proceedings of the 2003 International Computer Music Conference, Singapore, October 2003, pp. 27–33 (2003)

4. Clausen, M., Kurth, F., Müller, M., Arifi, V.: Automatic Synchronization of Musical Data: A Mathematical Approach. MIT Press, Cambridge (2004)
5. Dixon, S., Widmer, G.: Match: A music alignement tool chest. In: 6th International Conference on Music Information Retrieval (2005)
6. Müller, M., Kurth, F., Röder, T.: Towards an efficient algorithm for automatic score-to-audio synchronization. In: ISMIR 2004. 5th International Conference on Music Information Retrieval, Barcellona, Spain (October 2004)
7. Soulez, F., Rodet, X., Schwarz, D.: Improving polyphonic and poly-instrumental music to score alignment. In: 4th International Conference on Music Information Retrieval, pp. 143–148 (2003)
8. Turetsky, R.J., Ellis, D.: Ground-truth transcriptions of real music from force-aligned midi syntheses. In: Proc. Int. Conf. on Music Info. Retrieval ISMIR, vol. 3 (2003)
9. Rabiner, L.R., Juang, B.H.: Fundamentals of Speech Recognition. Prentice, Englewood Cliffs, NJ (1993)
10. Haus, G., Longari, M.: A multi-layered, timebased music description approach based on xml. Computer Music Journal 29(1), 70–85 (2005)
11. Haus, G., Longari, M.: Towards a symbolic/time-based music language based on xml. In: MAX 2002. Proceedings of the First International IEEE Conference on Musical Applications Using XML (2002)
12. Baraté, A., Ludovico, L.A.: An xml-based synchronization of audio and graphical representations of music scores. In: WIAMIS 2007. Proceedings of the 8th International Workshop on Image Analysis for Multimedia Interactive Services, Santorini, Greece (2007)