REAL-TIME MUSIC TRACKING USING TEMPO-AWARE ON-LINE DYNAMIC TIME WARPING

Andreas Arzt
Department of Computational Perception
Johannes Kepler University Linz

Gerhard Widmer
Department of Computational Perception
Johannes Kepler University Linz
The Austrian Research Institute
for Artificial Intelligence (OFAI)

ABSTRACT

This extended abstract describes our real-time music tracking system, which was submitted to the MIREX 2010 Score Following task. Our system is based on an on-line version of the well-known Dynamic Time Warping (DTW) algorithm and includes some extensions to improve both the precision and the robustness of the alignment (e.g. a tempo model and the ability to reconsider past decisions).

1. SYSTEM OVERVIEW

Rather than trying to transcribe the incoming audio stream into discrete notes and align the transcription to the score, we first convert a MIDI version of the given score into a sound file by using a software synthesizer. Due to the information stored in the MIDI file, we know the time of every event (e.g. note onsets) in this 'machine-like', low-quality rendition of the piece and can treat the problem as a real-time audio-to-audio alignment task.

The alignment algorithm itself is based on an on-line version of the Dynamic Time Warping algorithm and supported by a tempo model based on the alignment results during the last couple of seconds. The output of the system is at any time the current position in the score. See Figure 1 for an overview of our system.

2. DATA REPRESENTATION

The score audio stream and the live input stream to be aligned are represented as sequences of analysis frames, computed via a windowed FFT of the signal with a hamming window of size 46ms and a hop size of 20ms. The data is mapped into 84 frequency bins, spread linearly up to 370Hz and logarithmically above, with semitone spacing. In order to emphasize note onsets, which are the most important indicators of musical timing, only the increase in energy in each bin relative to the previous frame is stored.

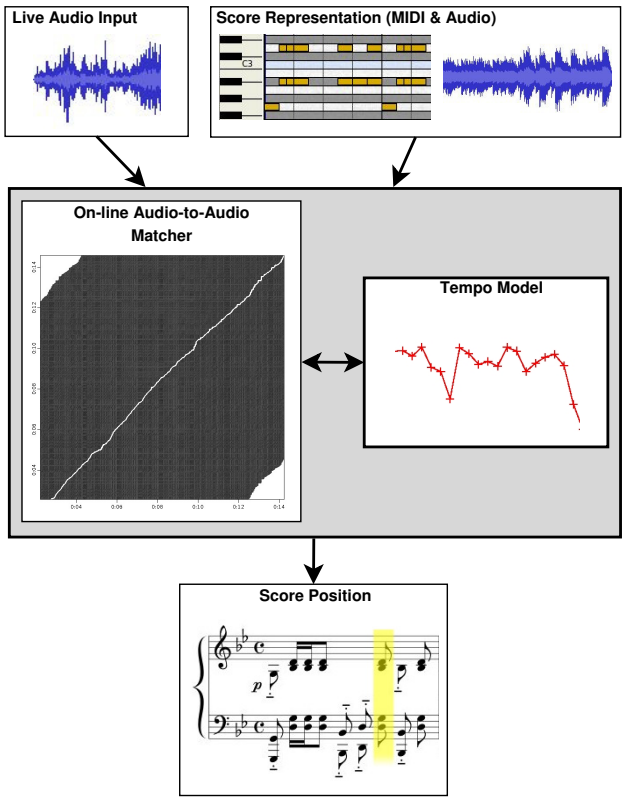

Figure 1. Overview of our Real-time Music Tracking System

3. ON-LINE DYNAMIC TIME WARPING

This algorithm is the core of our real-time music tracking system. ODTW takes two time series describing the audio signals – one known completely beforehand (the score) and one coming in in real time (the live performance) –, computes an on-line alignment, and at any time returns the current position in the score. In the following we only give a short intuitive description of this algorithm, for further details we refer the reader to [4].

Dynamic Time Warping (DTW) is an off-line alignment method for two time series based on a local cost measure and an alignment cost matrix computed using dynamic programming, where each cell contains the costs of the optimal alignment up to this cell. After the matrix computation is completed the optimal alignment path is obtained

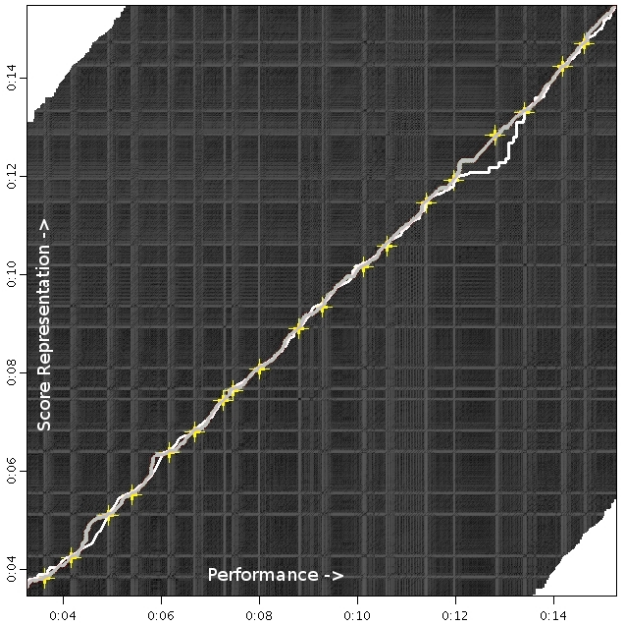

Figure 2. Illustration of the ODTW algorithm, showing the iteratively computed forward path (white), the much more accurate backward path (grey, also catching the one onset that the forward path misaligned), and the correct note onsets (yellow crosses, annotated beforehand). In the background the local alignment costs for all pairs of cells are displayed. Also note the white areas in the upper left and lower right corners, illustrating the constrained path computation around the forward path.

by tracing the dynamic programming recursion backwards (*backward path*).

Originally proposed by Dixon in [4], the ODTW algorithm is based on the standard DTW algorithm, but has two important properties making it useable in real-time systems: the alignment is computed incrementally by always expanding the matrix into the direction (row or column) containing the minimal costs (*forward path)*, and it has linear time and space complexity, as only a fixed number of cells around the forward path is computed.

At any time during the alignment it is also possible to compute a *backward path* starting at the current position, producing an off-line alignment of the two time series which generally is much more accurate. This constantly updated, very accurate alignment of the last couple of seconds is used heavily in our system to improve the alignment accuracy (see Section 4). See also Figure 2 for an illustration of the above-mentioned concepts.

4. THE FORWARD-BACKWARD STRATEGY

Some improvements to this algorithm, focusing both on increasing the precision and the robustness of the algorithm, were presented in [3] and are incorporated in our system. Most importantly, this includes the 'forward-backward strategy', which reconsiders past decisions (using the backward path) and tries to improve the precision of the current score position hypothesis.

More precisely, the method works as follows: After every frame of the live input a smoothed backward path is computed, starting at the current position (i, j) of the forward path. By following this path $b = 100$ steps backwards on the y-axis (the score) one gets a new point which lies with a high probability nearer to the globally optimal alignment than the corresponding point of the forward path.

Starting at this new point another forward path is computed until a border of the current matrix (either column i or row j) is reached. If this new path ends in (i, j) again, this can be seen as a confirmation of the current position. If the path ends in a column $k < i$, new rows are calculated until the current column i is reached again. If the path ends in a row $l < j$, the calculation of new rows is stopped until the current row j is reached.

5. A SIMPLE TEMPO MODEL

We also introduced a tempo model to our system [1], a feature which has so far been neglected by music trackers based on DTW.

5.1 Computation of the Current Tempo

The computation of the current tempo of the performance (relative to the score representation) is based on a constantly updated backward path starting in the current position of the forward calculation. Intuitively, the slope of such a backward path represents the relative tempo differences between the score representation and the actual performance. Given a perfect alignment, the slope between the last two onsets would give a very good estimation about the current tempo. But as the correctness of the alignment of these last onsets generally is quite uncertain, one has to discard the last few onsets and use a larger window over more note onsets to come up with a reliable tempo estimation.

In particular, our tempo computation algorithm uses a method described in [5]. It is based on a rectified version of the backward alignment path, where the path between note onsets is discarded and the onsets (known from the score representation) are instead linearly connected. In this way, possible instabilities of the alignment path between onsets (as, e.g., between the 2^{nd} and 3^{rd} onset in the lower left corner in Fig.2) are smoothed away.

After computing this path, the $n = 20$ most recent note onsets which lie at least 1 second in the past are selected, and the local tempo for each onset is computed by considering the slope of the rectified path in a window with size 3 seconds centered on the onset. This results in a vector v_t of length n of relative tempo deviations from the score representation. Finally, an estimate of the current relative tempo t is computed using Eq.1, which emphasizes more recent tempo developments while not discarding older tempo information completely, for robustness considerations.

$$t = \frac{\sum_{i=1}^{n}(t_i * i)}{\sum_{i=1}^{n} i} \quad (1)$$

Of course, due to the simplicity of the procedure and especially the fact that only information older than 1 second is used, this tempo estimation can recognize tempo changes only with some delay. However, the computation is very fast, which is important for real-time applications.

In [1] we also introduced a more sophisticated tempo model, which is based on a prior analysis of other performances of the piece in question. This feature was deactivated for the MIREX submission.

5.2 Feeding Tempo Information to the ODTW

Based on the observation that both the alignment precision and the robustness directly depend on the similarity between the tempo of the performance and the score representation, we now use the current tempo estimate to alter the score representation on the fly, stretching or compressing it to match the tempo of the performance as closely as possible. This is done by altering the sequence of feature vectors representing the score audio. The relative tempo is directly used as the probability to compress or extend the sequence by either adding new vectors or removing vectors.

More precisely, after every incoming frame from the live performance, and before the actual path computation, the current relative tempo t is computed as given above, where $t = 1$ means that the live performance and the score representation currently are in the exact same tempo and $t > 1$ means that the performance is faster than the score representation. The current position in the score p_s is given by the forward path and thus coincides with the index of the last processed frame of the score representation. If a newly computed random number r between 0 and 1 is larger than t (or $\frac{1}{t}$ if $t > 1$) an alteration step takes place. If $t > 1$, a feature vector is removed from the score representation by replacing p_s+1 and p_s+2 with a mean vector of $p_s + 1$ and $p_s + 2$. And if $t < 1$, a new feature vector, computed as the mean of p_s and p_s+1 is inserted next into the sequence between p_s and p_s+1. As our system is based on features emphasizing note onsets, score feature vectors representing onsets (which are known from the score) are not duplicated, as more (and wrong) onsets would be introduced to the score representation. In such cases the alteration process is postponed until the next frame. Furthermore, to avoid that the system could get stuck at one frame, alterations may take place at most 3 times in a row.

6. 'ANY-TIME' MUSIC TRACKING

Our system also includes a unique feature, namely the ability to cope with arbitrary structural deviations (e.g. large omission, (re-)starts in the middle of a piece) during a live performance. While previous systems – if they did deal at all with serious deviations from the score – had to rely on explicitly provided information about the structure of a piece of music and points of possible deviation (e.g., notated repeats, which a performer might or might not obey), our system does without any such information and continuously checks all (!) time points in the score as alternatives to the currently assumed score position, thus theoretically being able to react to arbitrary deviations (jumps etc.) by the performer.

As the data for this MIREX task does not include such deviations from the score, this feature was deactivated for the evaluation runs. For further information about our 'any-time' music tracking system we refer the reader to [2].

7. RESULTS

Coming soon ...

8. FURTURE WORK

An important direction for future work is the introduction of explicit event detection into our system, based on both an estimation of the timing and an analysis of the incoming audio frames. This would increase the alignment precision especially for sparse and/or monophonic music.

9. ACKNOWLEDGEMENTS

This research is supported by the City of Linz, the Federal State of Upper Austria, and the Austrian Federal Ministry for Transport, Innovation and Technology, and the Austrian Science Fund (FWF) under project number TRP 109-N23.

10. REFERENCES

[1] Andreas Arzt and Gerhard Widmer. Simple tempo models for real-time music tracking. In *Proc. of the Sound and Music Computing Conference (SMC)*, Barcelona, Spain, 2010.

[2] Andreas Arzt and Gerhard Widmer. Towards effective 'any-time' music tracking. In *Proc. of the Starting AI Researchers' Symposium (STAIRS), European Conference on Artificial Intelligence (ECAI)*, Lisbon, Portugal, 2010.

[3] Andreas Arzt, Gerhard Widmer, and Simon Dixon. Automatic page turning for musicians via real-time machine listening. In *Proc. of the 18th European Conference on Artificial Intelligence (ECAI)*, Patras, Greece, 2008.

[4] Simon Dixon. An on-line time warping algorithm for tracking musical performances. In *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, Edinburgh, Scotland, 2005.

[5] Meinard Mueller, Verena Konz, Andi Scharfstein, Sebastian Ewert, and Michael Clausen. Towards automated extraction of tempo parameters from expressive music recordings. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, 2009.