Proceedings of the Second Vienna Talk, Sept. 19–21, 2010, University of Music and Performing Arts Vienna, Austria

ROBUST REAL-TIME MUSIC TRACKING

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 paper describes our 'any-time' real-time music tracking system which 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). A unique feature of our system is the ability to cope with arbitrary structural deviations (e.g. jumps, re-starts) online during a live performance.

1. INTRODUCTION

In this paper we describe the current state of our real-time music tracking system. The task of a music tracking system is to follow a musical live performance on-line and to output at any time the current position in the score.

While most real-time music tracking systems are based on statistical approaches (e.g. the well-known systems by Raphael [8] and Cont [4]), our alignment algorithm is based on an on-line version of the Dynamic Time Warping algorithm first presented by Dixon in [5]. We subsequently proposed various improvements and additional features to this algorithm (e.g. a tempo model and the ability to cope with 'jumps' and 're-starts' by the performer), which are the topic of this paper (see also Figure 1 for an overview of our system).

2. DATA REPRESENTATION

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 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.

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 [5].

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

Originally proposed by Dixon in [5], 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

We presented some improvements to this algorithm, focusing both on increasing the precision and the robust-

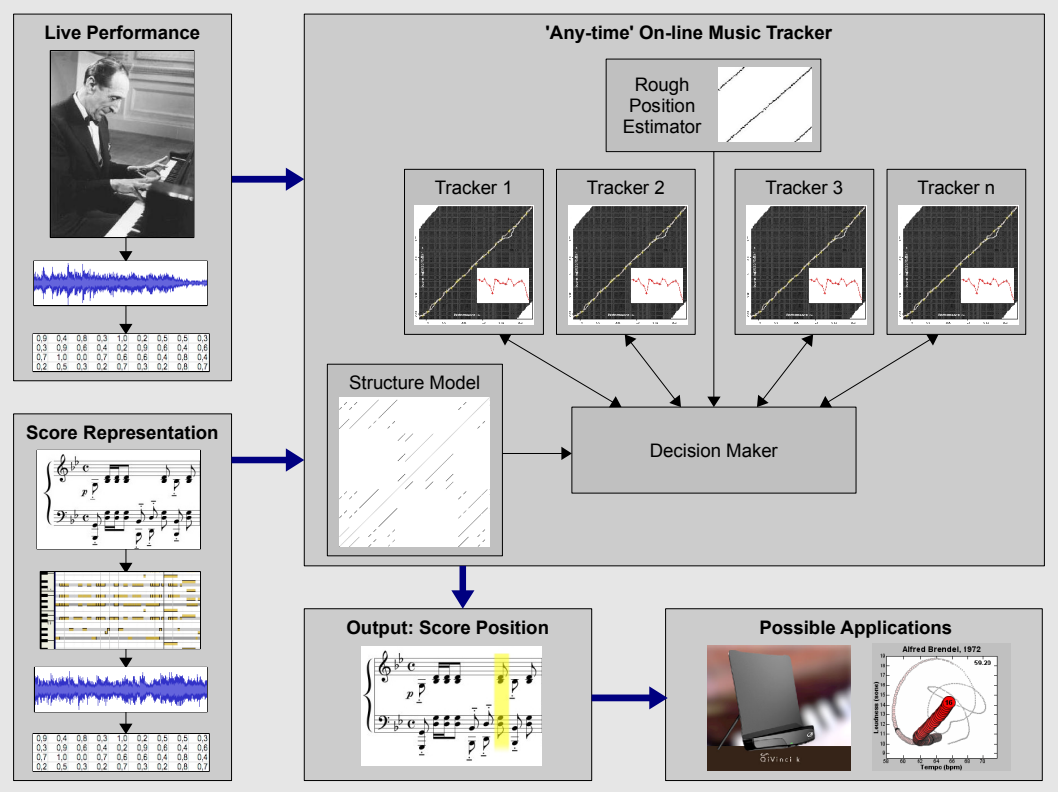

Figure 1: Overview of our 'Any-time' On-line Music Tracking System

ness of the algorithm in [3]. 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 [6]. 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

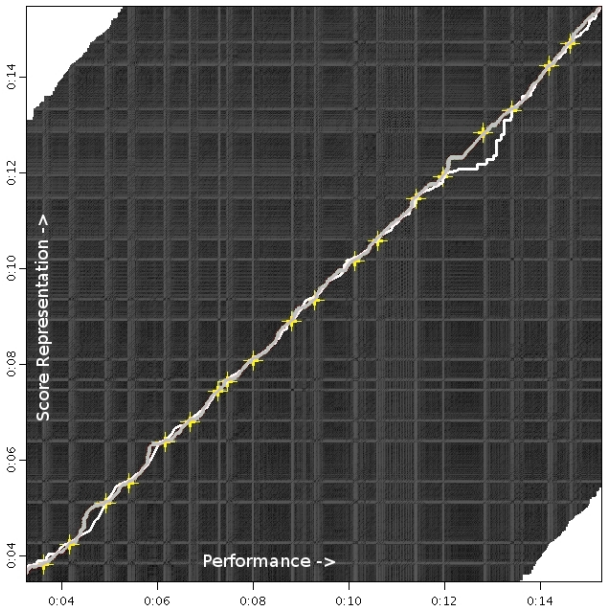

Figure 2: Illustration of the ODTW algorithm, showing the iteratively computed forward path (white), the much more accurate backward path (grey, also catching the one onset that the forward path misaligned), and the correct note onsets (yellow crosses, annotated beforehand). In the background the local 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.

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.

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

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.

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.

6. 'LEARNING' TEMPO DEVIATIONS FROM DIFFERENT PERFORMERS

The introduction of this very simple tempo model already leads to considerably improved tracking results. But especially at phrase boundaries with huge changes in tempo the above-mentioned delay in the recognition of tempo changes still results in large alignment errors. Furthermore, such tempo changes are very hard to catch instantly, even with more reactive tempo models. To cope with this problem we came up with an automatic and very general way to provide the system with information about possible ways in which a performer might shape the tempo of the piece.

First we extract tempo curves from various different performances (audio recordings) of the piece in question. Again, as for the real-time tempo estimation, this is done completely automatically using the method described in [6] (see Section 5.1), but as the whole performance is known beforehand and the tempo analysis can be done off-line there is now no need for further smoothing of the tempo computation. These tempo curves are directly imported into our real-time tracking system.

More precisely, as before, after every iteration of the path computation algorithm the vector v_t containing tempo information at note onsets is updated based on the backward path and the above-mentioned local tempo computation method. But now the tempo curve of the live performance over the last $w = 50$ onsets, again located at least 1 second in the past, is compared to the previously stored tempo curves at the same position. To do this all n tempo curves are first normalized to represent the same mean tempo over these w onsets as the live performance. The Euclidean distances between the curve of the live performance and the stored curves are computed. These distances are inverted and normalized to sum up to 1, thus now representing the similarity to the tempo curve of the live performance.

Based on the stored tempo curves our system can now estimate the tempo at the current position. As the current position should be somewhere between the last aligned onset o_j and the onset o_{j+1} to be aligned next, we compute the current tempo t according to Formula 2, where t_{i,o_j} and $t_{i,o_{j+1}}$ represent the (scaled) tempo information of curve i at onset o_j and o_{j+1} respectively, and s_i is the similarity value of tempo curve i.

$$t = \frac{\sum_{i=1}^{n}[(t_{i,o_j} + t_{i,o_{j+1}})s_i]}{2} \tag{2}$$

Intuitively, the tempo is estimated as the mean of the tempo estimates at these 2 onsets, which in turn are computed as a weighted sum of the (scaled) tempi in the stored performance curves, with each curve contributing according to its local similarity to the current performance. Please note that this approach somewhat differs from typical ways of training a score follower to follow a particular performance. We are not feeding the system with 'rehearsal data' by a particular musician, but with many different

ways of how to perform the piece in question, as the analysed performances may be by different performers and differ heavily in their interpretation style. The system then decides on-line at every iteration how to weigh the curves, effectively selecting a mixture of the curves which represents the current performance best.

7. 'ANY-TIME' MUSIC TRACKING

Our system (see Figure 1 for an overview) 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.

At the core is a process (the 'Rough Position Estimator', see Figure 1) that continually updates and evaluates high-level hypotheses about possible current positions in the score, which are then verified or rejected by multiple instances of the basic alignment algorithm ('Tracker 1–n', each including its own tempo model) described above. To guide our system in the face of possible repetitions and to avoid random jumps between identical parts in the score, we also introduced automatically computed information about the structure of the piece to be tracked. We chose to call our new approach *'Any-time Music Tracking'*, as the system is continuously ready to receive input and find out what the performers are doing, and where they are in the piece. For more details we refer the reader to [2].

8. EVALUATION

All above-mentioned improvements to the original algorithm were extensively evaluated. We have shown that the resulting system is both more robust and more accurate than the original system. Furthermore we already demonstrated the systems accuracy and reliability live on stage during a real piano performance. For detailed evaluation results we refer the reader to [3], [1] and [2].

9. FUTURE 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.

A possible future scenario would be to extend our 'any-time' algorithm to operate on a whole database of musical pieces, automatically recognising both the piece being played, and the current position. An off-line matching/retrieval scenario related to this has been described in [7]. Practically this will require a clever indexing scheme based on musically relevant high-level features to quickly find those pieces and time points most likely to match the ongoing sound stream.

10. 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 Research Fund (FWF) under project number TRP 109-N23.

11. 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] Arshia Cont. A coupled duration-focused architecture for realtime music to score alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6), 2010.

[5] 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.

[6] 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.

[7] Meinard Mueller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *Proc. of the 5th International Conference on Music Information Retrieval (ISMIR)*, London, 2005.

[8] Christopher Raphael. Current directions with music plus one. In *Proc. of the Sound and Music Computing Conference (SMC)*, Porto, Portugal, 2009.