Strategies towards the Automatic Annotation of Classical Piano Music

Bernhard Niedermayer[1]
[1]Department for Computational Perception
Johannes Kepler University Linz, Austria
music@jku.at

Gerhard Widmer[1,2]
[2]Austrian Research Institute for
Artificial Intelligence, Vienna, Austria
music@ofai.at

ABSTRACT

Analysis and description of musical expression is a substantial field within musicology. However, manual annotation of large corpora of music, a prerequisite for describing and comparing different artists' styles, is very labor-intensive. Therefore, computer systems are needed that can annotate recordings of different performances automatically, requiring only minimal corrections by the user. In this paper, we apply Dynamic Time Warping for audio-to-score alignment to extract the onset times of all individual notes within an audio recording, and compare two strategies for improving accuracy. The first one is based on increasing the temporal resolution of the features used. To cope with constraints in terms of computational costs, we apply a divide and conquer pattern. The second strategy is introducing a post-processing step in which the onset time of each individual note is revised. The advantage of this method is that, in contrast to default algorithms, arpeggios and asynchronies can be resolved as well.

1. INTRODUCTION

An important subfield of musicology is the analysis and description of musical style and expression. However, large corpora of annotated pieces of music played by several performers are needed to extract meaningful patterns or to support previously developed hypotheses. Such data can be acquired by performing pieces on computer-monitored instruments.

Despite the advantage of providing accurate and extensive data, using computer-monitored instruments for data acquisition has several substantial shortcomings. First of all, one can assume that music students might be persuaded quite easily to take part in such a project, but it will be hard to persuade top-class artists to do so. Secondly, it is not possible to analyze an artist's expressive evolution over long periods. And finally, research could not include artists who, although dead, remain famous and whose music is enjoyed by a broad audience.

Another source of data are audio recordings, which are not only cheap but also available in an extensive variety. However, the raw audio signal must be annotated before any high-level analysis can be performed, and manual annotation is very labor-intensive. In order to carry out research on large corpora of music, automatic – or at least semi-automatic – methods for data acquisition are needed.

The most general approach of collecting symbolic data from audio recordings would be Automatic Music Transcription. However, accuracy and robustness of state-of-the-art transcription methods do not meet the requirements of applications such as musical performance analysis. Especially in the context of classical music, where it can be assumed that the piece played is known in advance, using an audio file in combination with additional information given by the score has therefore become a common practice. Since the notes played are known a priori, the task is to extract the exact parameters of each note from the audio recording.

Such parameters do not only include the timing and the loudness of a note, but also characteristics such as its articulation or, when considering piano, pedal pressure. However, since knowing at which exact point in an audio signal a note is played is a prerequisite for estimating further properties, most current research is focused solely on this. The task is to temporally align or synchronize the notes given by the score to an audio recording - a process known as audio-to-score alignment.

In doing so, features are calculated from individual time frames of the audio signal. There are two main state-of-the art approaches to incorporating the score information: The score, which is by default given in MIDI-format, is used to either build a graphical model [1], such as an HMM, or it is used to compute a sequence of the same features as extracted from the audio signal [2]. Score and audio representations are then related to each other using the Viterbi algorithm or Dynamic Time Warping.

We use Dynamic Time Warping to compute this alignment. Since the algorithm is of quadratic complexity in both time and space, the temporal resolution of the features extracted cannot be increased arbitrarily without encountering limitations in terms of computational cost. One method of reducing the complexity is to apply a *divide and conquer* pattern splitting a piece into several sections using anchor notes, for which the timing is known. Dynamic Time Warping can then be performed on these individual sections without losing generality.

In [3], which originally introduced this approach, anchor notes were selected by the user. We propose a method for extracting such anchor notes automatically. To this end, Dynamic Time Warping is performed once, using a coarse

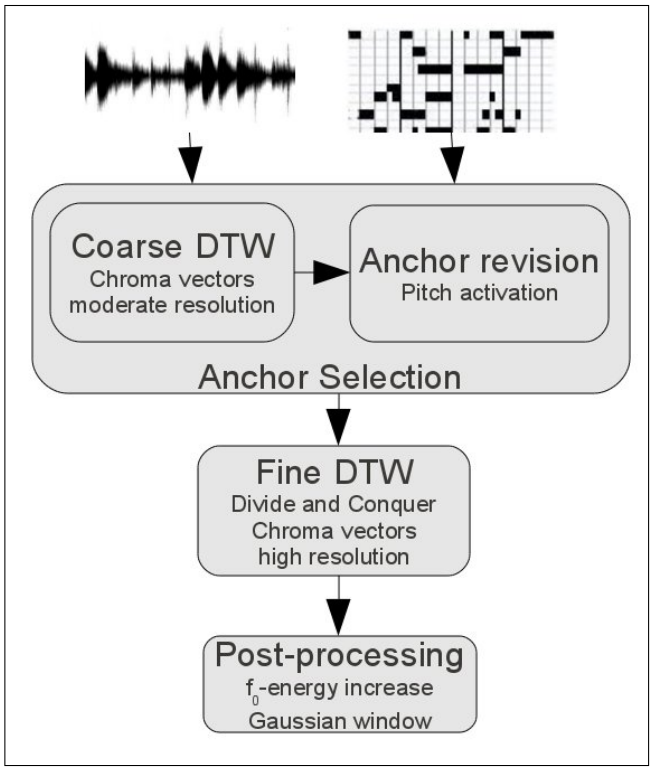


Figure 1. System overview

temporal resolution. Based on this initial estimate, anchor notes for which the timing can be extracted with relatively high confidence are identified and their onset times are revised.

Another shortcoming Dynamic Time Warping shares with the approach based upon graphical models is that several notes that occur simultaneously in the score, such as the individual notes of a chord, are always aligned to the same time frame of the audio signal. This is probably not relevant to applications such as augmented audio players. In performance analysis, however, this precludes the handling of arpeggiations or asynchronies. Therefore, we propose a post-processing step in which the onset time of each individual note is revised.

This post-processing step resembles the one described in [4] in both methodology and results. However, [4] used a beta-distribution to model the expectation strength of a note occurring at a certain point between two anchor notes. The beta-distribution was chosen because of its restriction to a fixed interval and the flexibility of its shape, but its use lacks probability-theoretic justification. In this paper, we show that comparable results can be obtained, by applying a weighting which reflects the normally distributed errors made by Dynamic Time Warping.

Figure 1 shows the system architecture, which is further described below as follows. First, we give an overview of related work in Section 2. Then, we explain the audio-to-midi alignment using coarse and fine Dynamic Time Warping in Section 3 and the extraction of the anchor notes based on the coarse alignment in Section 4. Section 5 describes the post-processing step. In Section 6, an evaluation of the system is presented, followed by conclusions in Section 7.

2. RELATED WORK

In offline audio-to-score alignment, a major group of approaches is based on chroma vectors in combination with Dynamic Time Warping (DTW) [2, 3, 4, 5, 6, 7]. This method has proven to yield robust global alignments. However, it cannot compete with onset detection algorithms concerning local accuracy. This was shown in [8], where, as a consequence, the idea of chroma vectors was combined with (pitch-wise) spectral flux – a feature used in onset detection.

A way of applying machine learning techniques to refine music alignments was shown in [5]: A neural network that detects note boundaries is trained on the result of an alignment. In an iterative process, the alignment can then be improved using the neural network's output, and the training is repeated.

Another approach of improving accuracy is to increase temporal resolution. Since DTW is of the order $O(n^2)$, this method is constrained by computational costs. The divide and conquer principle aside, [6] uses a multi-scale algorithm where in each iteration the resolution is increased and, at the same time, the area searched for an optimal alignment is narrowed down.

[7] and [9] combined those two strategies in an efficient way: Both compute an alignment based on DTW and then refine the note onsets within a search window around the initial estimates. Since the size of these search windows is small, a relatively high temporal resolution can be chosen. The additional features used in this post-processing step emphasize onsets of individual pitches. In doing so, the DTW algorithm's problem of unresolved arpeggiations or asynchronies becomes irrelevant. However, in contrast to the system presented here, the method in [9] relies on manual path initialization in the DTW step, and in [7], potentially conflicting notes are not revised, only marked for further processing.

3. AUDIO-TO-MIDI ALIGNMENT

3.1 Chroma Vectors

Chroma vectors are the feature used in most alignment systems because of their robustness to several common phenomena in music, such as changing timbre or different degrees of polyphony. In [2], chroma vectors were shown to outperform several other features in the context of audio matching and alignment. They consist of a 12-dimensional vector per time frame, in which each element represents one pitch class (C, C#, D, ...).

When calculating chroma vectors from a midi file, the energies (in midi terminology *velocities*) of all pitches belonging to the same pitch class are summed up. Additionally, it is beneficial to also consider harmonics by adding decreasing contributions of energies to the corresponding pitch classes. In contrast, when considering an audio signal, the pitches of notes sounding within a certain time frame are not known a priori. In this case, the values are

computed based on an STFT spectrogram by summing up the energies of those frequency bins which are mapped to the same pitch class. The mapping is done by choosing the pitch (and the corresponding pitch class with its index i) with the smallest frequency deviation from a bin's center frequency f_k, according to

$$i = \left[\text{round}\left(12\log_2\left(\frac{f_k}{440}\right)\right)\right] \mod 12 \qquad (1)$$

Within the work reported here, we use two STFT configurations: (i) a window size of 4096 samples and a hop size of 1024 samples, referred to as *moderate resolution*; and (ii) a window size 512 and a hop size 128, referred to as *high resolution*.

3.2 Dynamic Time Warping

After the feature extraction step, the score and the audio signal are both represented by a sequence of feature vectors. To evaluate an alignment, a cost function must be defined which measures the error made when aligning a specific frame of the first sequence to the corresponding frame of the second one. Preliminary experiments showed that the Euclidean distance yields better results within our framework than other functions, such as the cosine distance or the symmetric Kullback-Leibler divergence.

Using this cost function, a similarity matrix S can be calculated. The rows of S represent the time frames of the audio recording, while the columns represent the time frames of the score. Each value S_{ij} gives the cost of aligning frame i of the audio signal to frame j of the score. All continuous and monotonic paths through S which begin and end at the two end-points of the main diagonal represent valid alignments. The sum of all S_{ij} along an alignment path is the respective global alignment cost.

DTW computes an optimal alignment, i.e., the one minimizing the global cost, in two steps. In the first one, the optimal cost C_{ij} of each partial alignment, ending with frame i of the audio signal being aligned to frame j of the score representation, is calculated according to

$$C(i,j) = \min\begin{cases} C(i-1,j-1) + S_{ij} \\ C(i-1,j) + S_{ij} \\ C(i,j-1) + S_{ij} \end{cases} \qquad (2)$$

By starting at $C_{0,0} = S_{0,0}$ and storing all intermediary results in a matrix C, this recursion can be calculated efficiently.

$C_{N-1,M-1}$ is the minimal global alignment cost. However, in this application, the cost itself is not as important as the alignment path corresponding to this optimum. This path is obtained in the second step by backtracking based on knowledge of which of the three options in equation 2 was used in each step. This information can easily be stored during the forward step. For a more detailed description of the basic DTW algorithm, we refer the interested reader to [10].

3.3 Efficiency Considerations

Given two sequences of lengths N and M, DTW is of complexity $O(N * M)$ in both time and space. This resolves to $O(N^2)$ under the assumption that the score is stretched to the length of the audio signal prior to the feature extraction step. This precludes aligning arbitrarily long feature sequences and therefore limits both the lengths of pieces to be aligned and temporal resolution.

A classical method of improving the efficiency of DTW is to constrain the search for an optimal alignment path to a certain area within the similarity matrix S, such as the Itakura parallelogram or the Sakoe-Chiba band [10]. This is based on the assumption that expressive tempo changes will not exceed certain limits, or that the alternation of speeding up and slowing down will prevent the alignment path from deviating from the main diagonal by more than a maximum offset. These approaches can reduce computational costs to the order of $O(2N)$. However, there is the risk that, at some point, the assumptions do not hold and the true alignment path leaves the search area.

Other methods which share similar strengths and weaknesses are Path Pruning, in which only the most promising partial paths with costs below an adaptive threshold are further expanded, Shortcut Path, where only the alignment of frames corresponding to note on- and offsets are considered, and multi-scale DTW [6].

A completely different approach is to perform an online alignment – also known as score following [11]. This algorithm does not consider a piece as a whole, but advances through the audio signal incrementally. This works for arbitrarily long pieces and, leaving the real-time aspect out of consideration, arbitrarily high feature resolutions. The drawback is that the method can only extract instantaneous optima at each step and cannot guarantee that a global optimum is found.

3.4 Divide and Conquer Approach

[3] introduced a divide and conquer approach to improve the efficiency of DTW. Given a set of anchor notes for which the exact timing is known, solving the alignment problem for the whole piece can be reduced to finding optimal alignments between each pair of consecutive anchor notes. Given a maximal interval c between two anchors, the sub-DTWs are computed in $O(c^2)$ in both time and space. When considering the whole piece, the space complexity of $O(c^2)\widehat{=}O(1)$ does not change. Time complexity, however, increases to $O(c^2 * N/c) = O(c * N)\widehat{=}O(N)$. Compared to the original algorithm of order $O(N * M)$, this approach reduces complexity and guarantees that a globally optimal alignment is found.

This increase in efficiency is countered by the additional problem of how to identify suitable anchor notes and how to extract their respective onset times. [3] proposed an approach in which the user selects an anchor configuration manually or verifies suggestions made by the algorithm. These suggestions are established based on cues such as pauses, long isolated fortissimo chords, or notes with salient fundamental pitches, i.e., pitches that do not overlap with harmonics of concurrently played notes.

4. ANCHOR EXTRACTION

In this paper, we show how anchor notes can be determined automatically. The selection is based on a coarse DTW computed as described above. Within a search window around the first onset estimate of each note, a revised candidate is then extracted using the Pitch Activation feature as described in [7]. Finally, all notes for which ambiguities arise are dropped from the list of anchors.

Doing this has two implications. First, basing anchor selection on an initial alignment alters the method, shifting it away from the original divide and conquer approach and towards a special multi-scale DTW. Second, the algorithm is not guaranteed to find the global optimum, since errors in the anchor selection result in inaccurate alignments.

4.1 Pitch Activation

The feature used for revising onset candidates is pitch activation, which is calculated by applying a modification of non-negative matrix factorization (NMF) to audio data in the frequency domain. NMF is the decomposition of an input matrix V of size $n \times m$ into two output matrices W and H of sizes $m \times r$ and $r \times n$ such that the elements of all these matrices are non-negative and

$$V \approx WH \tag{3}$$

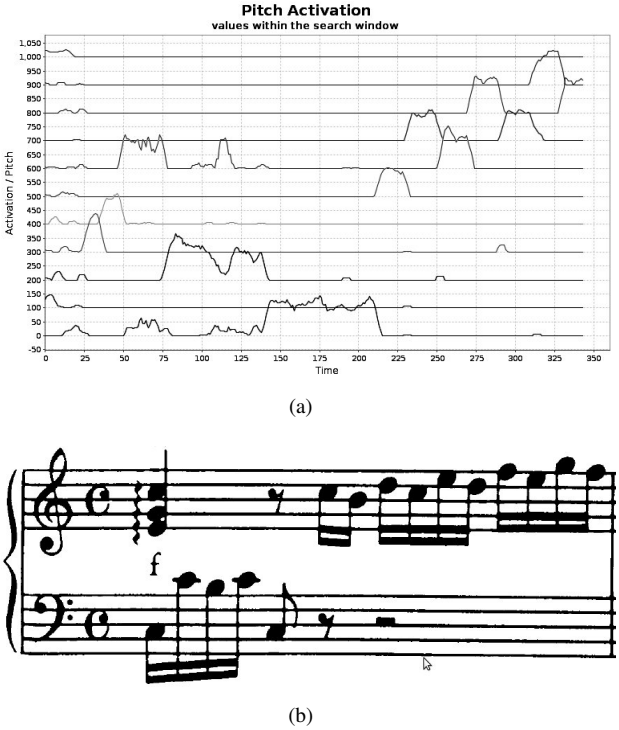
The reconstruction error, i.e., the deviation of WH from V, can be measured with several cost functions such as the Euclidean distance or the Kullback-Leibler divergence. An optimal factorization is calculated by minimizing this cost.

Applied to audio processing, NMF can be used to factorize a spectrogram into a dictionary W of weighted frequency groups and the corresponding activation energies H of these frequency groups over time. Depending on V and the parameter r, the base components in W can represent models of single tones or chords. But since the NMF algorithm, as originally introduced in [12], is unsupervised, it is more likely that some of the components also describe single partials, special patterns during an attack, sustain, or decay phase of a note, or even just noise. However, in the context of audio-to-score alignment, where the piece played is known a priori, we assume the instrument or set of instruments playing to be known as well. Thus, tone models can also be trained using supervised methods.

Based on this assumption, a dictionary W of tone models is trained in advance ([7, 4, 13]). The training data comprises recordings of single tones played at several degrees of loudness on the instrument under consideration. A short-time Fourier transform is calculated and factorized while exploiting knowledge of the tone samples. Since only one tone is played in each sample, the number of components r is set to one. The activation energy of this component w over time is fixed to $\overline{h}$ and assumed to be equal to the amplitude envelope. Equation 3 then resolves to

$$V \approx w\overline{h} \tag{4}$$

By minimizing the reconstruction error, a tone model is learned from each training sample. Since the relative energy of the harmonics depends on the intensity, preliminary models are trained using several degrees of loudness, and the final model is then obtained by taking the average weight for each frequency.



(a)

(b)

Figure 2. Example of activation patterns: (b) shows the first bar of Mozart's piano sonata k279. In (a) the activation patterns of the single pitches are plotted in ascending order.

Given this fixed dictionary $\overline{W}$, equation 3 can be rewritten as

$$v \approx \overline{W}h \tag{5}$$

where v and h are single columns of V and H that can now be processed independently. Since in both equation 5 and equation 4 there is only one variable left, a non-negative least squares optimization minimizing the mean square criterion

$$f = \frac{1}{2} \parallel \overline{W}h - v \parallel^2 \tag{6}$$

can be applied instead of the original NMF methods. This not only reduces computational costs but also, due to the independence of individual frames, makes the pitch activation h – a frame-wise f_0 estimation – a feature suitable for online algorithms.

Figure 2 shows an example of such activation patterns. The dictionary used for the factorization consisted of the models describing those pitches which are expected to be played within the time range shown only.

4.2 Anchor selection

For the anchor selection, the pitch activation feature is used to revise the onset estimates obtained by DTW. $\overline{W}$ is composed from the tone models of those pitches, expected to

be played within the search window and an additional component modeling white noise. The new onset candidate is set to the frame with the maximal increase of h_p, where p is the index of the component corresponding to the pitch of the note under consideration. In cases in which the onset is unambiguous, these onset candidates have proven to be very accurate. But in cases in which the same note is repeated several times within the search window, this method is too simple and very likely to fail.

To solve this dilemma, notes that are expected to be played more often than once within the search window are disregarded as potential anchors. Also, all notes for which the onset is ambiguous, i.e., for which the difference between the onset estimate obtained by the initial coarse DTW and the revised onset time exceeds a certain threshold, are dropped from the list of anchor candidates. In doing so, the anchor selection benefits from the robustness of the DTW as well as from the accuracy of the pitch activation-based onset revision.

Although this approach is very simple, our evaluation in Section 6 shows that by adjusting the onset times of these anchor notes only, the overall result is improved significantly.

5. POST PROCESSING FOR POLYPHONIC PIECES

As pointed out before, both DTW and alignment methods based on graphical models suffer from the shortcoming that notes which are indicated in the score as being played simultaneously simultaneously will inherently be aligned to the same time frame within the audio signal. Hence, increasing the temporal feature resolution to arbitrary dimensions as described in Section 3 benefits monophonic pieces for which the onsets of individual notes are extracted and pieces which are too long to be processed as a whole, even when using moderate resolutions. However, when considering polyphonic pieces, notes which are indicated in the score as being played simultaneously will never be played precisely simultaneously by the performer due to arpeggiations or asynchronies. Therefore, using a resolution high enough to break a chord down into several notes and their individual onsets results in an ambiguous onset time for the chord as a whole. It is not clear if the estimate obtained by DTW or the Viterbi algorithm represents the note which is played first, the one which is played last, or some time in between where the cumulative energy of all chord notes has exceeded a certain threshold.

To overcome this problem, we apply a post-processing step in which the onset times of all non-anchor notes are revised as well. We assume that the high-resolution DTW computed by our system yields relatively accurate estimations and that deviations from the real onset times follow a normal distribution. Therefore, on the one hand, a search window of length $2l$ centered around the initial estimate is considered. On the other hand, feature values computed to refine the onset time are weighted using a Gaussian window.

However, the choice of features is not trivial. Pure onset detection functions, such as spectral flux, are not sufficient, since, when dealing with polyphonic pieces, the onsets of other chord notes must be expected to occur within the immediate vicinity of a note. Also, the pitch activation feature used for anchor selection is not suitable, since it performs poorly in situations of repeated notes.

Preliminary experiments showed that, considering the remaining non-anchor notes, the increase in the energy of the fundamental frequency of a note is the most reliable and accurate onset estimate. We obtain this information from a constant Q transform with a frequency resolution of one bin per semitone and set the revised onset candidate to the time frame at which the maximal increase occurs.

Parameter values of around 100 ms for the search radius l and 0.4 for the standard deviation σ of the Gaussian window have proven to yield good results. A detailed evaluation can be found in the next section.

6. EVALUATION

6.1 Evaluation Method

The evaluation was done using the first movements of 11 Mozart sonatas played by a professional pianist. The performances were recorded on a computer-monitored Bösendorfer SE290 grand piano, logging the exact onset times of all notes. The data comprises more than 30.000 notes with an overall performance time of more than one hour. Scores were presented to the system in midi format.

The absolute temporal displacement between aligned notes and the ground truth served as the main evaluation criterion. We investigated the median absolute displacement, the 75^{th}, and the 95^{th} percentile. In our opinion, this shows a clearer view of a system's performance than the mean and variance of absolute displacements, since these values are more sensitive to outliers. When considering only mean and variance, it is difficult to distinguish systems that yield accurate estimates for most notes but produce a few outliers with relatively large temporal displacement, from systems which are more robust but less accurate.

In the evaluation of the whole system including the post-processing, we include two other criteria. The long-term goal of our research is to provide an annotation system that detects onset times as accurately as a human. Only a very small number of notes for which manual correction is needed should remain. [14] showed that humans do not perceive timing deviations smaller than 10 ms. Therefore, we also investigated the proportion of notes which are aligned with a timing deviation below this threshold.

Furthermore, we determined the percentage of the notes aligned with a displacement of less than 50 ms. This criterion is well known from the field of onset detection. Here, it reflects the ratio of reasonably well aligned notes to outliers.

6.2 Evaluation Results

Table 1 shows the accuracy of the selected anchor notes in comparison to the non-anchor notes before and after performing the fine resolution DTW. One can clearly see that the anchor nodes are indeed more accurate than the

piece	duration	# notes	# anchors	$50\% < x$[ms]			$75\% < x$[ms]			$95\% < x$[ms]		
				anch.	orig.	new	anch.	orig.	new	anch.	orig.	new
K.279-1	4:55	2803	885	6.0	15.7	15.8	13.3	29.6	29.8	43.7	127	128
K.280-1	4:48	2491	987	5.7	23.2	22.9	12.1	44.6	44.6	43.5	165	165
K.281-1	4:29	2648	954	6.6	25.2	25.1	13.3	47.3	47.2	47.8	137	138
K.282-1	7:35	1907	513	7.8	26.8	26.7	14.5	64.0	64.2	80.8	388	389
K.283-1	5:22	3304	875	8.1	15.5	15.4	14.4	27.8	27.8	40.9	67.8	68.2
K.284-1	5:17	3700	853	7.0	15.2	15.3	15.9	30.6	30.7	62.3	108	107
K.330-1	6:14	3160	888	6.0	16.0	15.9	10.5	29.4	29.3	37.9	148	146
K.332-1	6:02	3470	844	11.5	22.8	22.9	19.0	42.1	42.3	61.3	167	168
K.333-1	6:44	3774	1122	8.0	17.1	17.1	14.4	30.3	30.4	42.1	105	105
K.457-1	6:15	2993	885	9.2	21.3	21.3	16.3	40.5	40.3	59.8	267	267
K.475-1	4:58	1284	371	15.4	36.3	36.3	23.7	92.0	92.5	79.4	270	270

Table 1. Comparison between accuracy (median, 75^{th} percentile, and 95^{th} percentile) of the anchor notes (anch.), the non-anchor notes computed by the coarse DTW (orig.), and the non-anchor notes after performing the fine DTW implementing the divide and conquer pattern (new)

piece	# notes	$50\% < x$[ms]			$75\% < x$[ms]			$95\% < x$[ms]		
		orig.	anch.	new	orig.	anch.	new	orig.	anch.	new
K.279-1	2803	15.7	11.2	11.8	30.0	24.5	25.7	103	101	103
K.280-1	2491	23.6	12.7	13.4	41.9	32.0	32.8	126	127	126
K.281-1	2648	24.2	15.2	16.1	42.4	36.5	36.9	114	114	114
K.282-1	1907	23.5	18.7	19.6	53.7	47.2	48.1	354	354	354
K.283-1	3304	14.6	12.7	12.8	27.1	24.5	24.8	62.0	60.8	60.9
K.284-1	3700	15.4	12.5	13.1	31.0	26.9	27.4	96.8	98.0	98.0
K.330-1	3160	14.9	11.4	11.8	27.7	24.0	24.7	118	118	115
K.332-1	3470	20.5	18.4	18.6	38.6	35.6	36.3	138	140	138
K.333-1	3774	16.2	12.9	13.4	29.3	25.8	26.4	79.6	75.8	76.4
K.457-1	2993	19.4	15.7	16.2	36.9	33.5	34.2	204	203	202
K.475-1	1284	29.7	24.5	25.0	68.4	65.5	65.9	224	224	224
all	31534	18.4	14.1	14.7	35.2	30.1	30.8	130	131	131

Table 2. Overall accuracy (median, 75^{th} percentile, and 95^{th} percentile) of the divide and conquer DTW (new) compared to the coarse DTW with the anchor notes revised (anch.) and the coarse DTW without anchor note revisions (orig.)

remaining notes. However, it is remarkable that the high-resolution DTW implementing the divide and conquer principle does not improve the results obtained by the original implementation using a moderate temporal resolution. A discussion on this issue is given in the next section.

It is worth mentioning that, due to the semi-automatic nature of the anchor selection, only a very small number of anchors was used in [3]. In our approach, the number of anchors is much larger, as shown in Table 1. Qualitative analysis of single passages showed that there are "easy" sections, in which no ambiguities occur and almost every single note is chosen to serve as anchor, whereas there are "difficult" sections within a piece in which only few anchors are found. Although not required, the high number of anchor notes is desirable, since, in contrast to [3], the objective here was not only efficiency, but also to investigate accuracy aspects. This approach to anchor selection clearly outperforms the DTW variant in terms of accuracy.

Recalling the whole system's architecture, as depicted in Figure 1, Table 2 compares the results after the individual stages - the coarse DTW, the coarse DTW with revised anchor notes, and the high-resolution DTW exploiting these anchor notes. It is even more apparent that, while the revision of anchor notes improves the result significantly, the additional high-resolution DTW even decreases the overall accuracy very slightly.

The overall accuracy of the whole system including the post-processing step is listed in Table 3. According to our evaluation criteria, more than 90% of all notes were aligned reasonably well, i.e., such that evaluation frameworks used in onset detection would classify them as correct. Almost half of the notes were aligned with an error small enough not to be perceived by a human listener. Comparing the percentiles to the ones given in Table 2 clearly proves the benefit of the post-processing step.

Since the high-resolution DTW did not improve the results, the question arises if the system performed better without this step. Applying the post-processing method directly to the results of the anchor selection yielded similar results as the whole system. The overall number of notes with a temporal displacement of less than 10 ms increased slightly to 49.2%, while the number of notes with an alignment error of less than 50 ms decreased to 88.9%.

piece	# notes	$50\% < x$[ms]	$75\% < x$[ms]	$95\% < x$[ms]	$x < 10\,ms$	$x < 50\,ms$
K.279-1	2803	7.7	20.3	93.3	59.4%	90.7%
K.280-1	2491	7.3	16.0	79.0	62.0%	91.5%
K.281-1	2648	9.1	21.8	112	53.4%	89.8%
K.282-1	1907	11.4	22.0	258	44.3%	85.9%
K.283-1	3304	10.1	17.6	51.7	49.3%	94.8%
K.284-1	3700	8.1	20.1	78.5	57.7%	90.4%
K.330-1	3160	8.0	16.0	66.3	58.8%	93.5%
K.332-1	3470	15.8	25.8	106	31.6%	90.0%
K.333-1	3774	10.4	19.0	60.3	48.5%	93.3%
K.457-1	2993	13.4	25.1	164	37.6%	86.1%
K.475-1	1284	19.0	30.0	359	24.7%	85.6%
all	31534	10.3	21.3	92.6	49.0%	90.7%

Table 3. Overall accuracy after post-processing

7. CONCLUSIONS

We have described two strategies to improve the accuracy of offline audio-to-score alignments. One is to apply a higher feature resolution. In order not to be constrained by computational costs, a divide and conquer approach exploiting selected anchor notes was used.

The second strategy is to include a post-processing step which works on the level of individual notes. Here, we have proposed an approach that combines the robustness of DTW with the accuracy of a special onset feature by weighting the feature values using a Gaussian window centered around the onset estimate obtained by DTW. In [4], which introduced a very similar post-processing method, the analog weighting of feature values was done based on a beta-distribution, which was used for pragmatic reasons only. In contrast, the Gaussian window applied in our approach is justified by the actual data.

Our evaluation showed that the largest improvement is due to the revision of the anchor notes. Based on this step, increasing the temporal resolution does, remarkably, not yield higher, but even slightly lower overall accuracy. A possible explanation is that, on the one hand, errors caused by arpeggiations or asynchronies cannot be eliminated by DTW or related algorithms, independently of the temporal resolution. On the other hand, the same features – chroma vectors – were used for the initial coarse alignment and the high-resolution DTW. In cases in which chroma vectors, despite their advantages, are not powerful enough to represent all information that would be needed, the feature resolution becomes irrelevant.

Also, only notes for which the revised onset obtained by the pitch activation feature was near the initial estimate were chosen as anchors. This was necessary to exclude ambiguous cases. However, the revised anchors themselves never deviate from the initial alignment path by more than a small threshold. Therefore, the additional information produced by theses corrections is limited.

The post-processing step, in contrast, improved the result of the DTW including the revision of anchor notes significantly. We attribute this to the same factors as mentioned above. First, a new feature which is independent of the ones used previously is introduced to the system and therefore adds new information. Also, since the post-processing steps work at the level of independent notes, asynchronies can now be resolved.

We conclude that the DTW algorithm using features of moderate resolution works with high robustness and satisfactory accuracy. Improvements of the algorithm which can exploit features with higher temporal resolution did not improve the overall results. We will therefore concentrate our future work on more advanced post-processing methods, since this is the area where we see the largest potential for improvements.

8. ACKNOWLEDGEMENTS

This research is supported by the Austrian Federal Ministry for Transport, Innovation and Technology, and the Austrian Science Fund (FWF) under project numbers TRP 109-N23, P19349-N15, and Z159.

9. REFERENCES

[1] C. Raphael: "Aligning Music Audio with Symbolic Scores Using a Hybrid Graphical Model", *Machine Learning*, Vol. 65 (2-3), pp. 389–409, 2006.

[2] N. Hu, R. B. Dannenberg, and G. Tzanetakis: "Polyphonic Audio Matching and Alignment for Music Retrieval", *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, 2003.

[3] M.Müller, F. Kurth and T. Röder: "Towards an Efficient Algorithm for Automatic Score-to-Audio Synchronization", *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, Barcelona, 2004.

[4] B. Niedermayer and G. Widmer: "A Multi-Pass Algorithm for Accurate Audio-to-Score Alignment", *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, Utrecht, 2010.

[5] N. Hu and R. B. Dannenberg: "A Bootstrap Method for Training an Accurate Audio Segmenter", *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, London, 2005.

[6] M. Müller, H. Mattes, and F. Kurth: "An Efficient Multiscale Approach to Audio Synchronization", *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, Victoria, 2006.

[7] B. Niedermayer: "Improving Accuracy of Polyphonic Music-to-Score Alignment", *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, Kobe, 2009.

[8] S. Ewert and M. Müller: "Refinement Strategies for Music Synchronization", *Proceedings of the 5th International Symposium on Computer Music Modeling and Retrieval (CMMR 2008)* Copenhagen, 2008.

[9] Y. Meron and K. Hirose: "Automatic alignment of a musical score to performed music", *Acoustical Science and Technology*, Vol. 22, No. 3, pp. 189–198, 2001.

[10] Rabiner, L. R. and Juang, B.-H. "Fundamentals of speech recognition". Prentice Hall, Englewood Cliffs, NJ, 1993.

[11] S. Dixon: "Live Tracking of Musical Performances Using On-Line Time Warping", *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx)*, Madrid, 2005.

[12] Lee, D. D. and Seung, H. S. "Algorithms for Non-Negative Matrix Factorization", *Neural Information Processing Systems*. 2000.

[13] F. Sha and L. Saul: "Real-time pitch determination of one or more voices by nonnegative matrix factorization", *Advances in Neural Information Processing Systems 17*, K. Saul, Y. Weiss, and L. Bottou (eds.), MIT Press, Cambridge, MA, 2005.

[14] A. Friberg and J. Sundberg: "Perception of just noticeable time displacement of a tone presented in a metrical sequence at different tempos", *Proceedings of the Stockholm Music Acoustics Conference*, pp. 39–43, Stockholm, 1993.